Getting Started with IBM Bluemix

Hands-On Workshop

Module 3: First Deploy Exercises

Table of Contents

Workshop overview

In this workshop, you will:

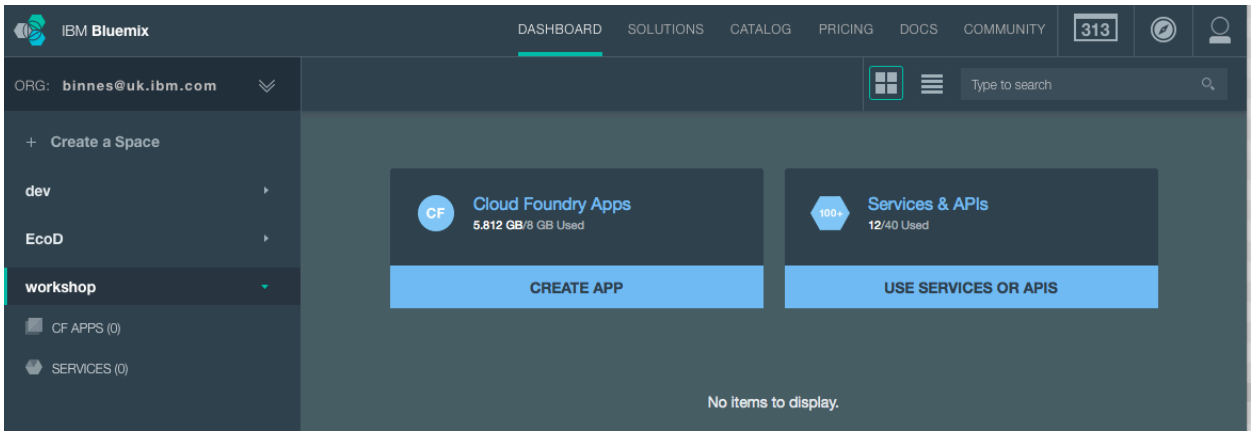
- Deploy a simple application from the Bluemix web interface.
- Use the cf command line to modify and deploy the application.
- Configure Eclipse so that you can create IBM Bluemix as a runtime in Eclipse

Exercise 3a: Deploy your first application

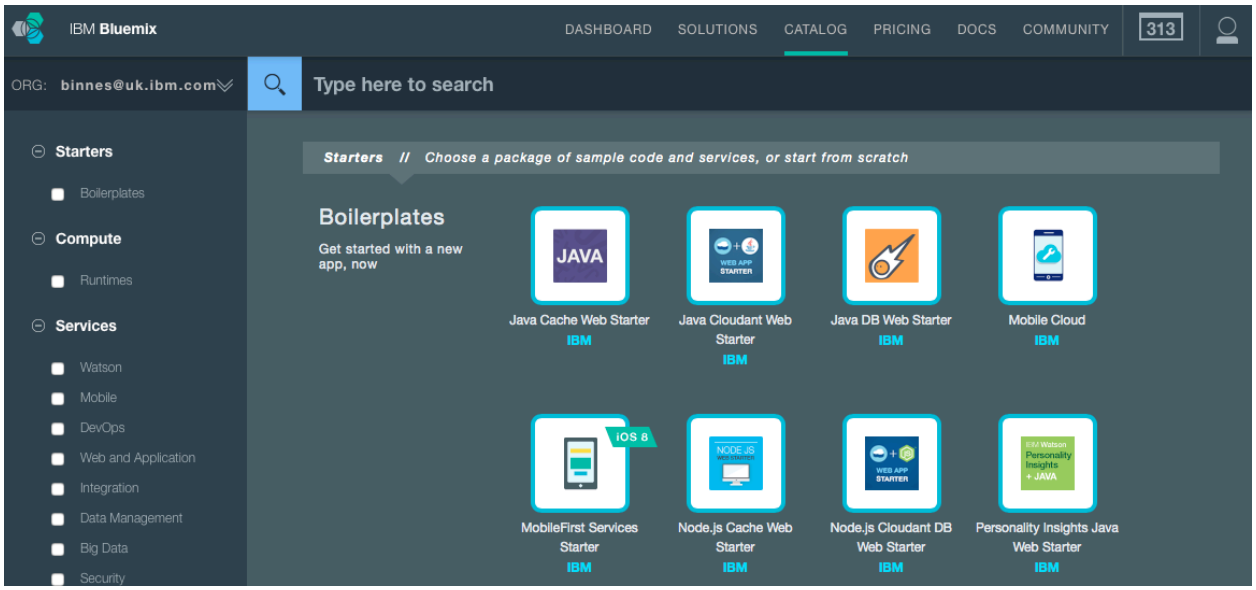
This exercise will show you how to deploy a simple application from the IBM Bluemix web interface.

1. In a browser, navigate to one of the following regions:
   - https://bluemix.net: This link should take you to your default location.
     - https://console.ng.bluemix.net (Region: US South)
     - https://console.eu-gb.bluemix.net (Region: United Kingdom)

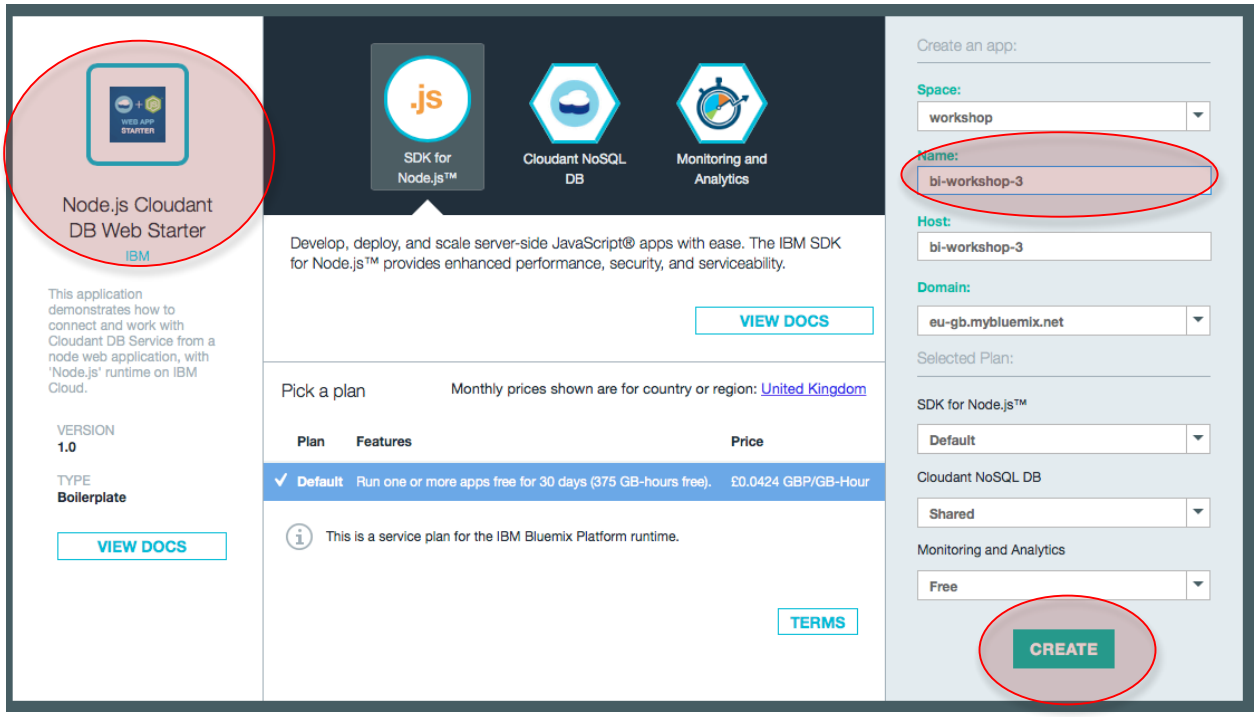
2. Click **LOG IN** and then enter your login information on the IBM id page and click **Sign in**. You should see your dashboard view:



3. Click **CATALOG**.



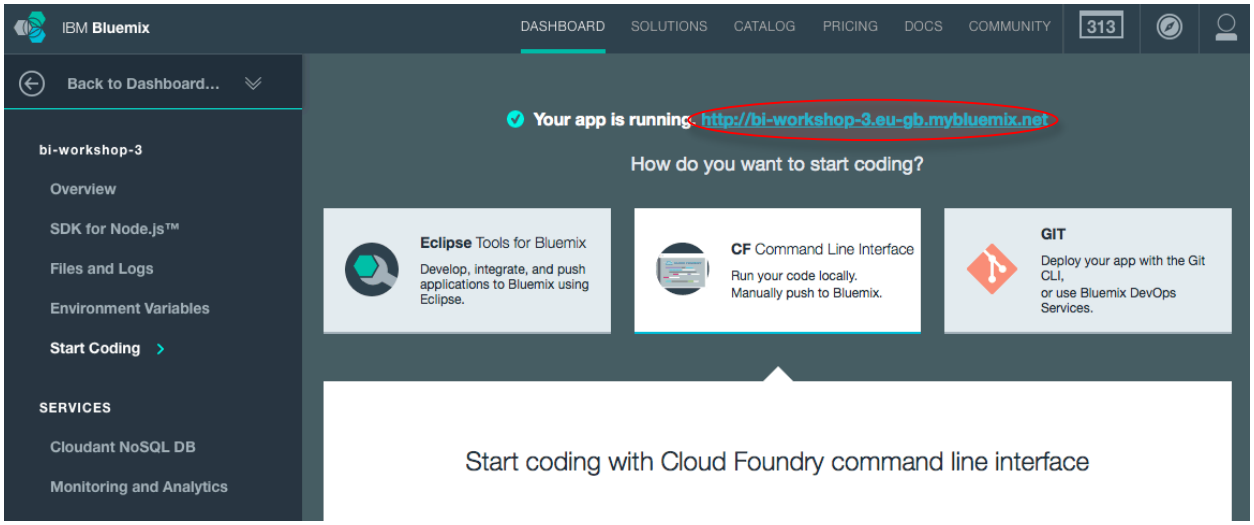
4. Select the **Node.js Cloudant DB Web Starter** from the **Boilerplates** section.



5. Enter a name for your application as shown above. The host name information is automatically entered. The host name must be unique on Bluemix, so enter a name with your company name or initials to make the name unique.

6. Click **CREATE**.

After a short while your application should be running. You can launch the application by clicking on the route.

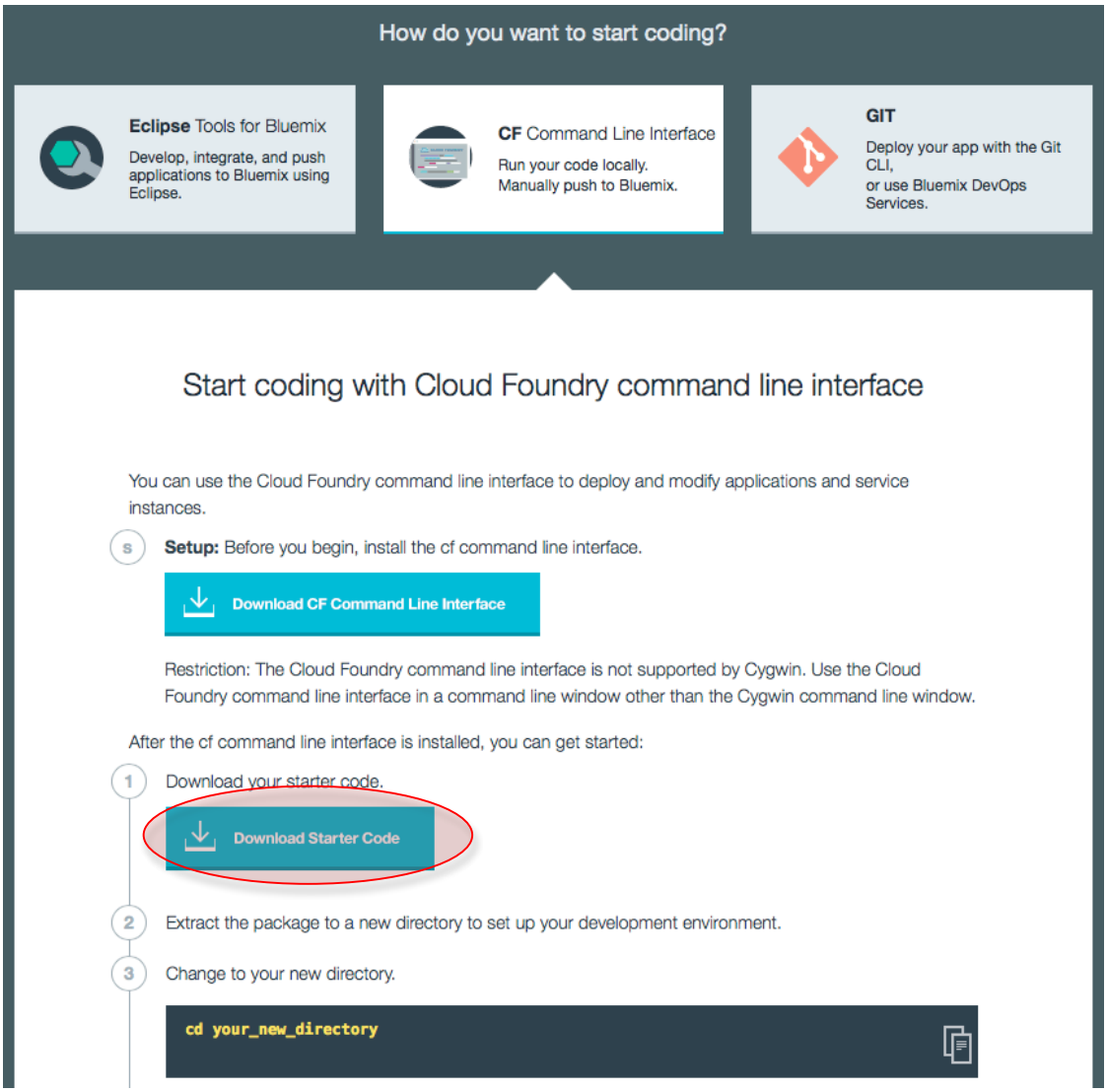


Exercise 3b: Deploy and then update the application by using the CLI

In this exercise, you use the cf command-line interface (CLI) to work with Bluemix. You use this tool in a terminal or command window on your workstation.

Use the same sample application that was used in exercise 3a.

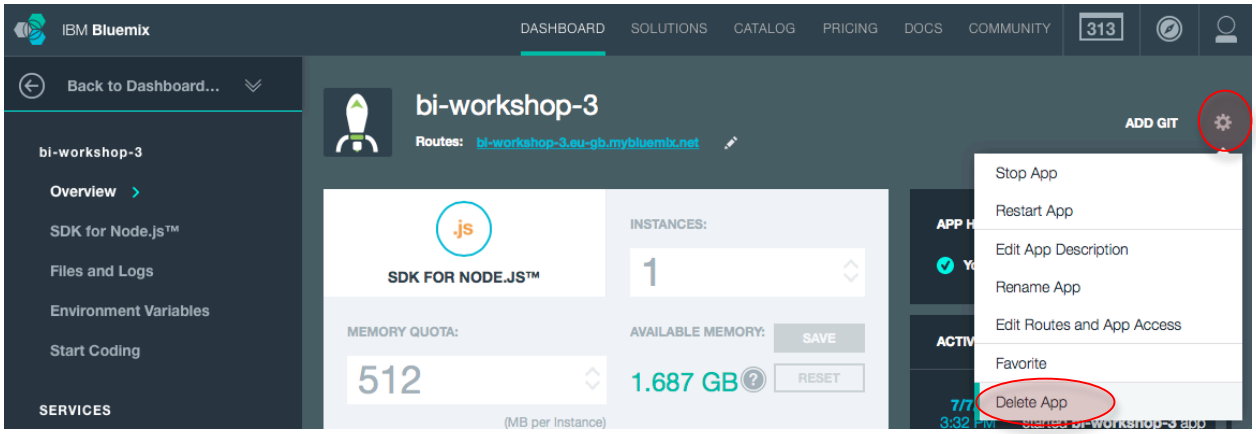
1. Click **Start Coding** and then click **Download Starter Code**.



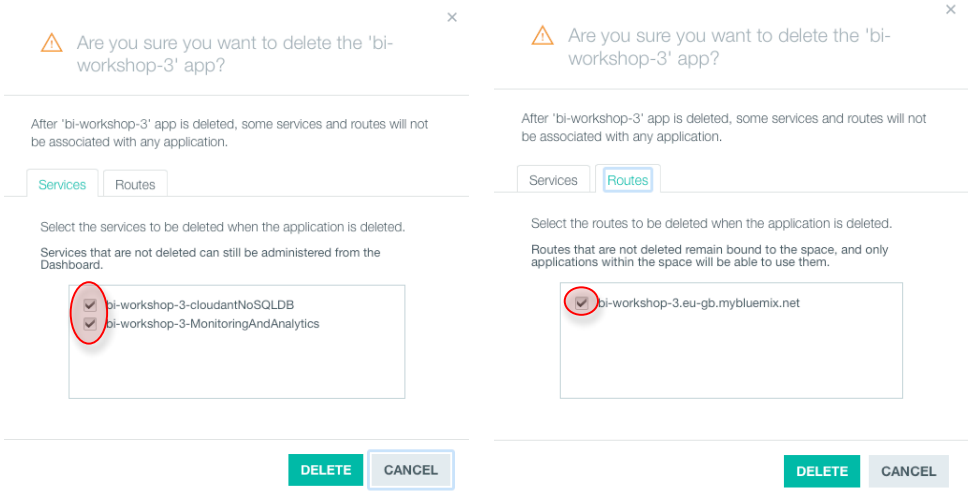
2. After the starter package is downloaded, move it to a directory on your workstation where you want to work, such as the `Bluemix` directory in your `Documents` folder.

3. Extract the package by double-clicking or right-clicking and click **Extract** or **Unarchive**. Do *not* delete the .zip file: you will need it in Exercise 3c.

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. Delete the service and the route with the application by selecting the checkbox in the **Services** tab and the **Routes** tab:



6. Click **OK** 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.)

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)*

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:

    cf marketplace

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

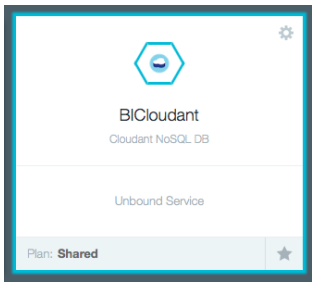
```
WorkloadScheduler          free                           Use the Workload Scheduler service to create and schedule repeata
ble business processes to make applications production ready. Trigger your processes to run based on an event or according to a
 schedule
blazemeter                 free-tier                      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 the 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, which will be explained later.
- `--no-start` instructs to CLI tool not to automatically start the application.

Don't want to allow the application to automatically start because it needs a database to run. You must link the Cloudant database instance to the application before you start the application.

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 application name used when the application is deployed.
    - `BICloudant` is the service instance name used when the service is deployed.

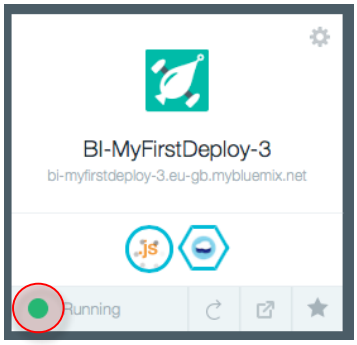
    If you refresh the web UI, you see that the application and service are linked, but the application is still stopped.

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

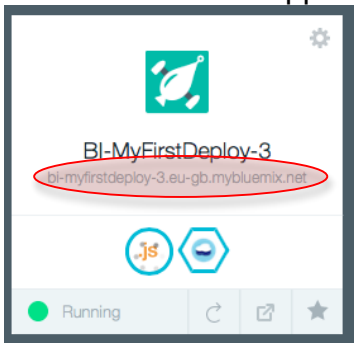
    cf start BI-MyFirstDeploy-3

    - `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 on 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 306, 307 and 310):

- Line 306: Change the docName from `'sample_doc'` to `'test_doc'`
- Line 307: Change the docDesc from `'A sample Document'` to `'A test Document'`
- Line 310: Change the value from `'A sample Document'` to `'A test Document'`

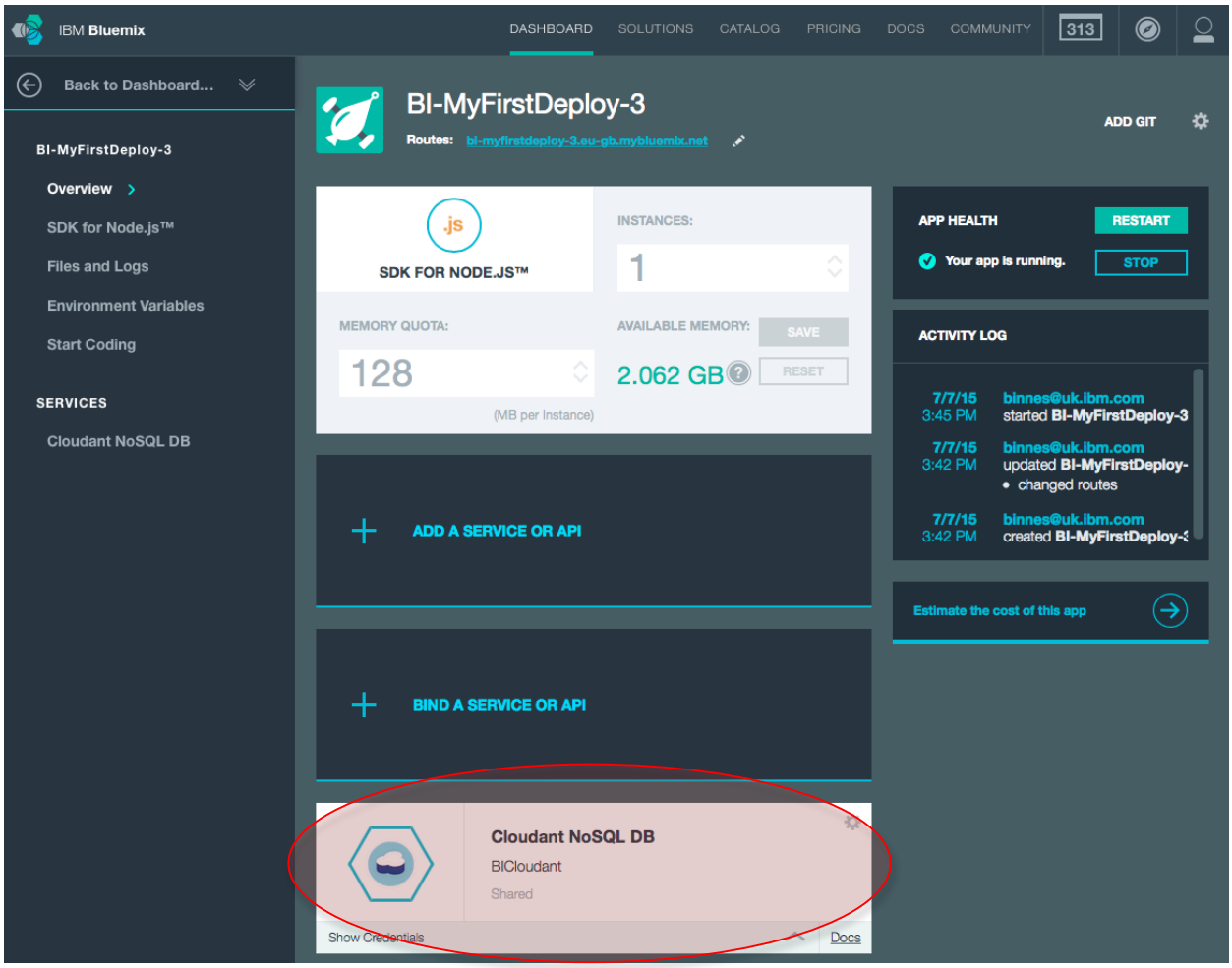
Save the file when you're finished editing.

```
292 app.get('/api/favorites', function(request, response) {
293
294     console.log("Get method invoked.. ")
295
296     db = cloudant.use(dbCredentials.dbName);
297     var docList = [];
298     var i = 0;
299     db.list(function(err, body) {
300       if (!err) {
301         var len = body.rows.length;
302         console.log('total # of docs -> '+len);
303         if(len == 0) {
304           //push sample data
305           // save doc
306           var docName = 'sample_doc';
307           var docDesc = 'A sample Document';
308           db.insert({
309             name : docName,
310             value : 'A sample Document'
```

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

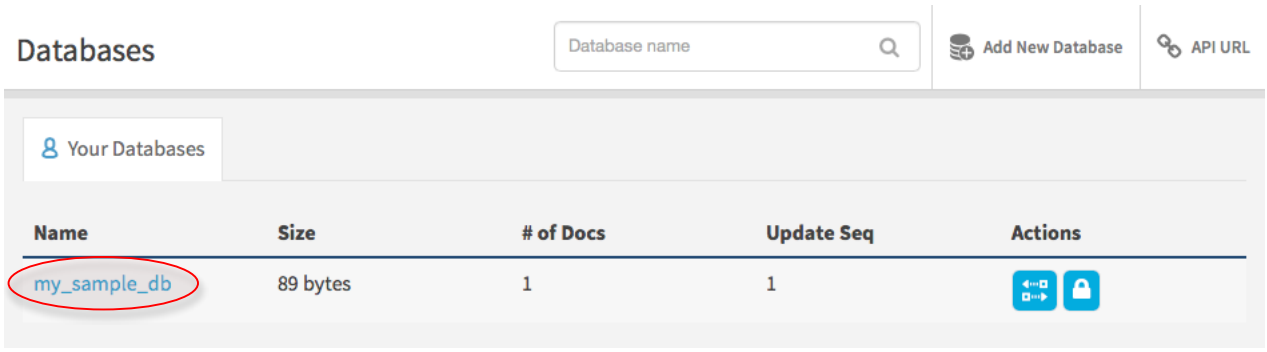
We have just modified the code that creates the sample document in the database. The document must be deleted from the database before you restart the application to allow the database to be populated again.

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

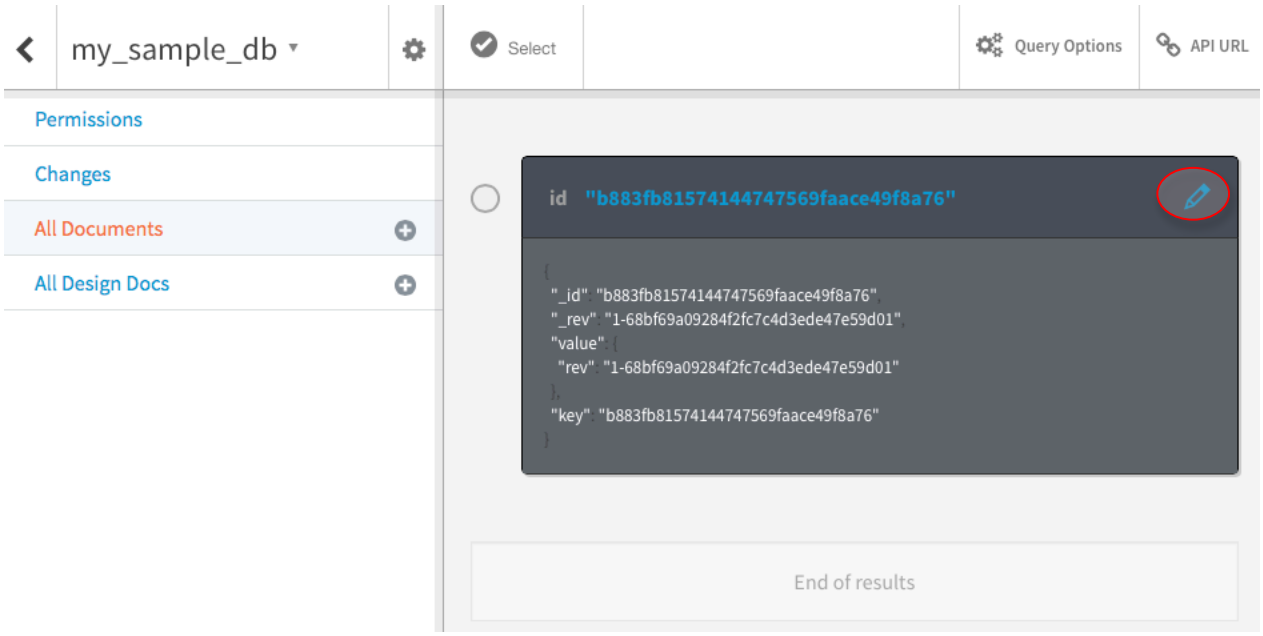


20. Launch the Cloudant console.

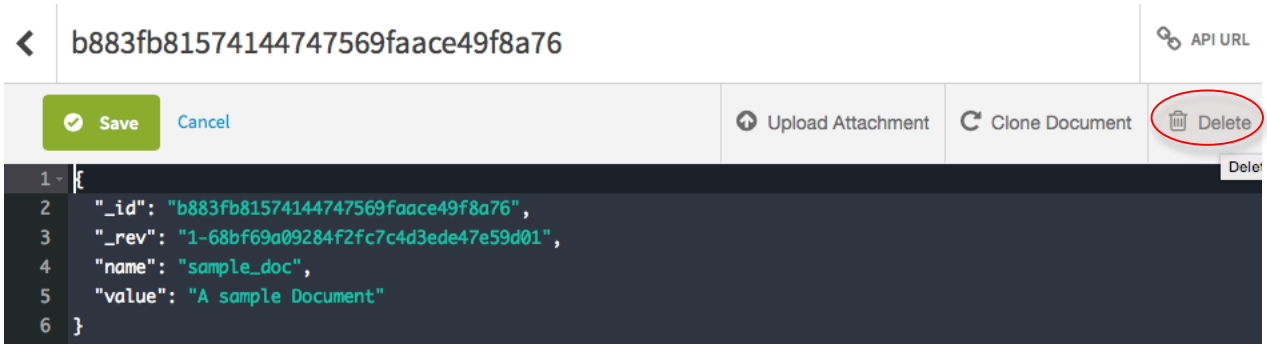
You should see a single database. Select the database:



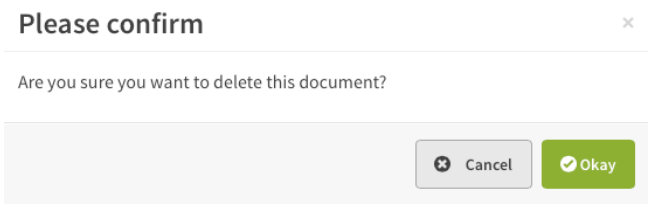
21. Edit the database document.



22. Delete the document.



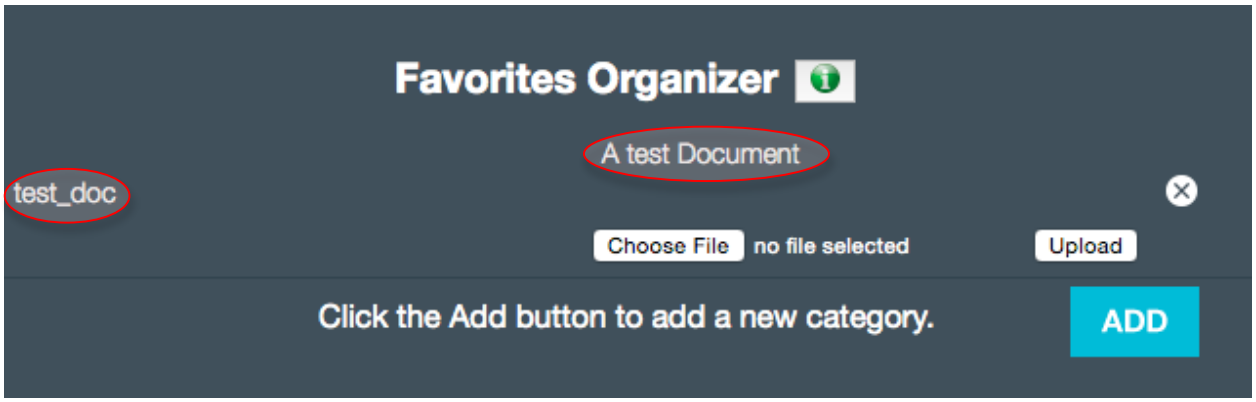
23. Confirm the deletion when prompted.



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

```
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 exercise.

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.

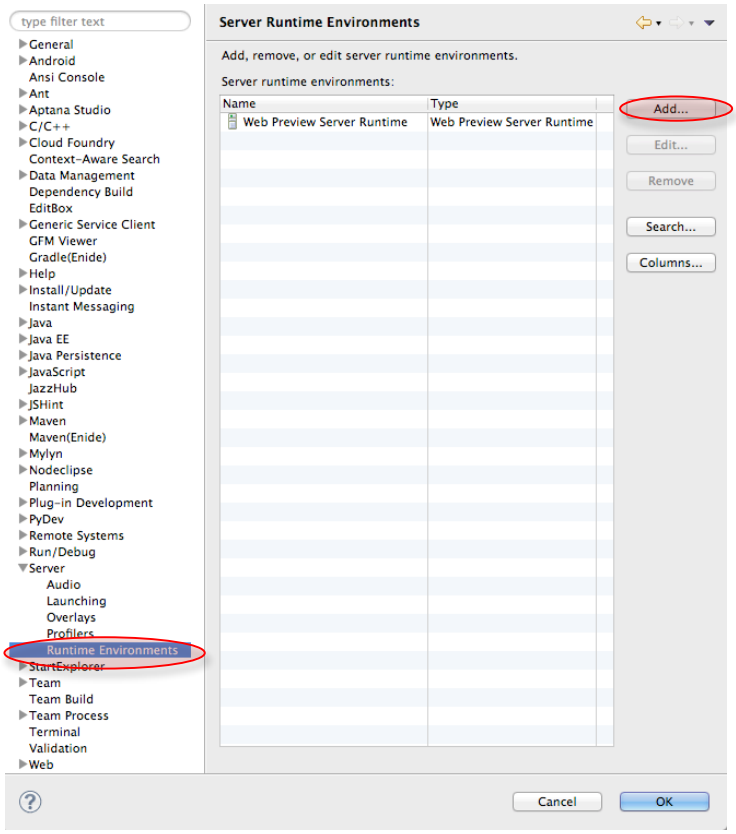
Confirm the deletion of the application and service.

Exercise 3c: Working with Eclipse and Bluemix

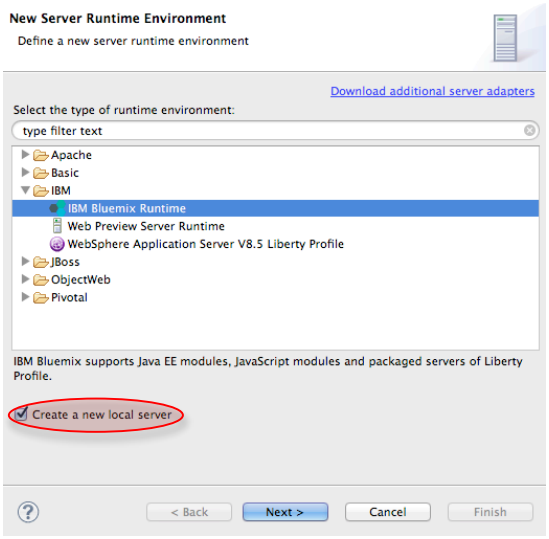
This exercise shows you how to work with Eclipse and Bluemix.

1. Launch Eclipse and switch to the JavaScript perspective. You should use a new workspace, but this is not essential.

2. Configure Eclipse to use Bluemix as a server by opening **Eclipse Preferences** and clicking **Server > Runtime Environments**.

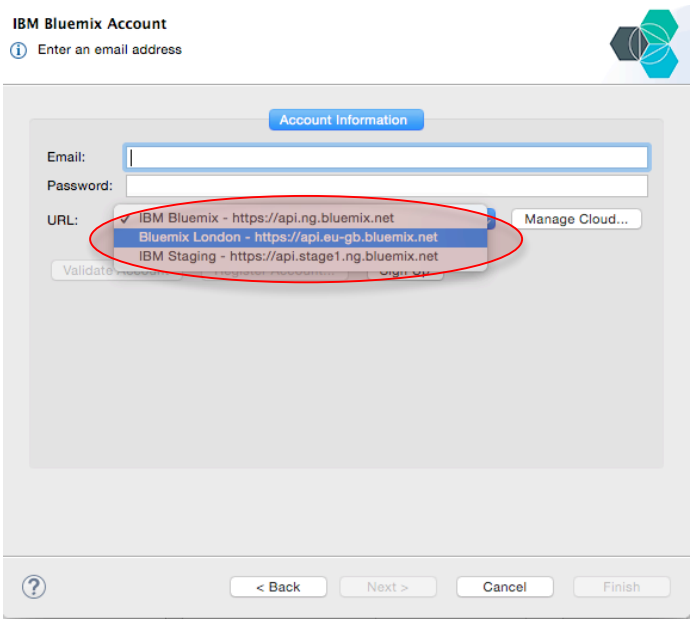


3. Click **Add**. Then, select **IBM Bluemix Runtime** and select the **Create a new local server** checkbox.



4. Click **Next**.

5. Select the appropriate URL to match the region that you have been using.

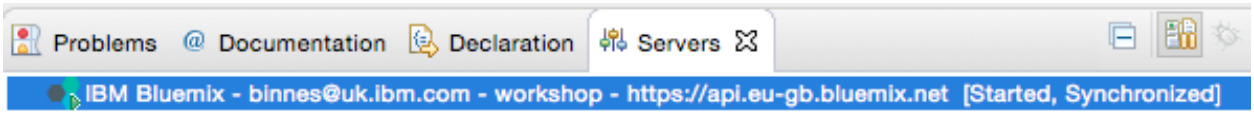


6. Enter your email and password, which are the credentials that you used to log into Bluemix. Click **Validate Account** to ensure that all details are valid and then click **Next**.

7. Select the organization and space that you want to use and then click **Finish**. Click **OK** to close the Preferences page.

If you want to work with multiple spaces in Eclipse, you must create multiple server configurations.

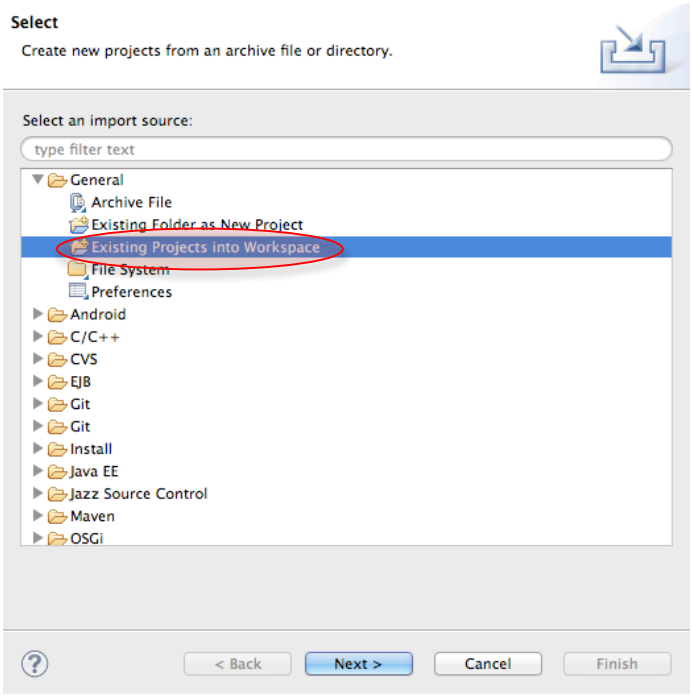
8. Open the Servers view in Eclipse. If it's not showing, click **Window > Show View > Servers**.



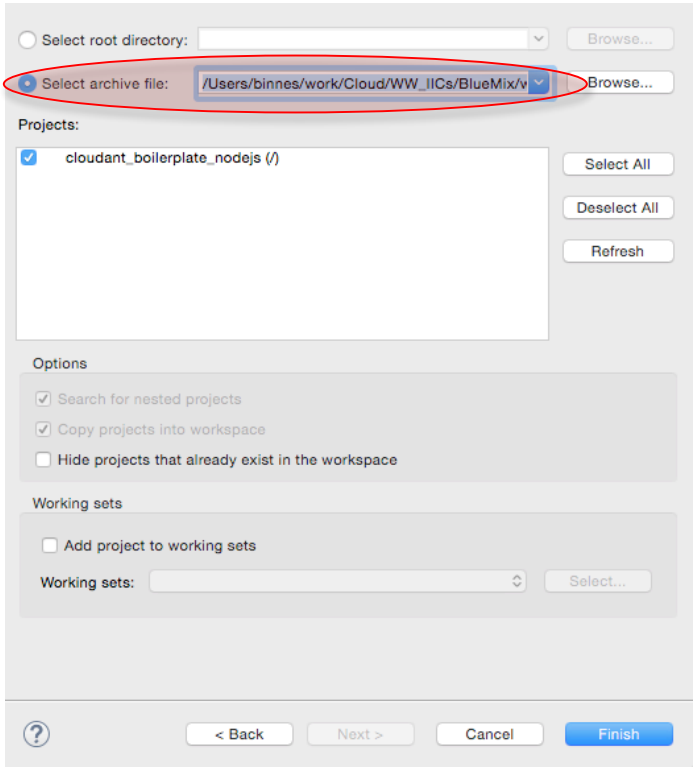
Use the same application that you used in the last session. You should have the archive file that you downloaded in the last exercise on your workstation. If not, deploy the Node.js Cloudant Web Starter application from the Bluemix web UI, click **View Guide,** and then download the starter application.

9. Import the starter application package to Eclipse by clicking **File > Import**.

10. In the Import dialog, click **General > Existing Projects into Workspace**.



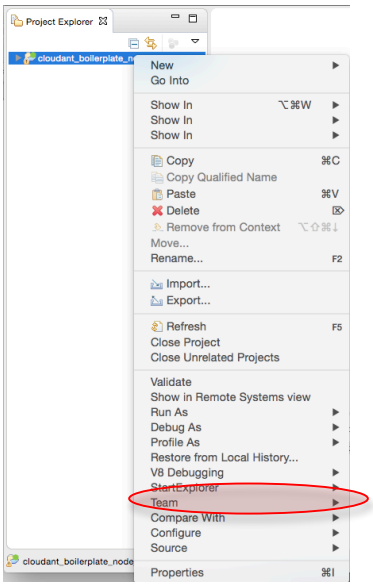
11. Click **Next**. Then, select the downloaded Zip file and click **Finish**.



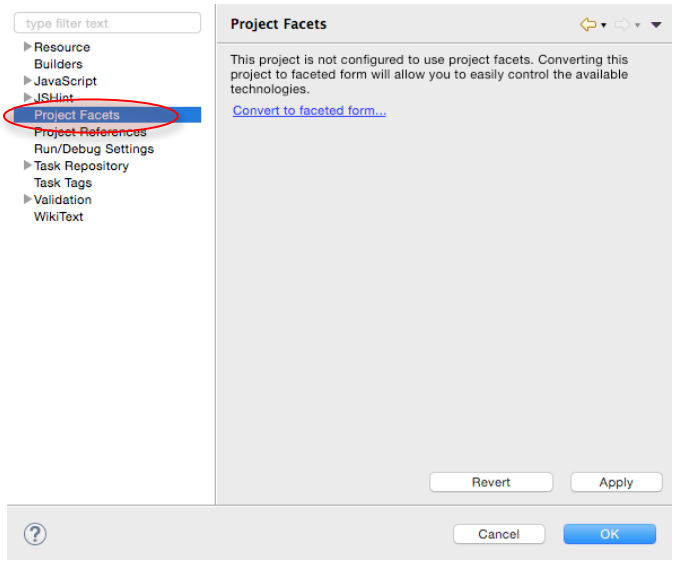
The project will be created.

Before JavaScript applications can be deployed to Bluemix by the plug-in, those applications must be identified as a project suitable for Bluemix deployment by assigning a facet to the project.

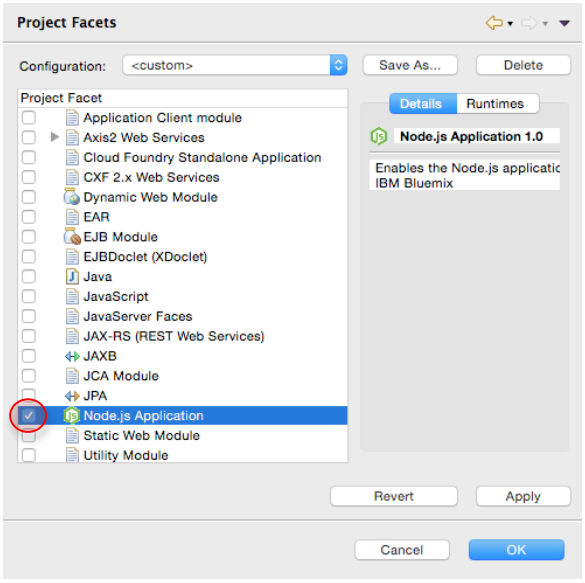
12. Right-click the project in the Project Explorer view and click **Properties**.



13. Click **Project Facets > Convert to faceted form**.

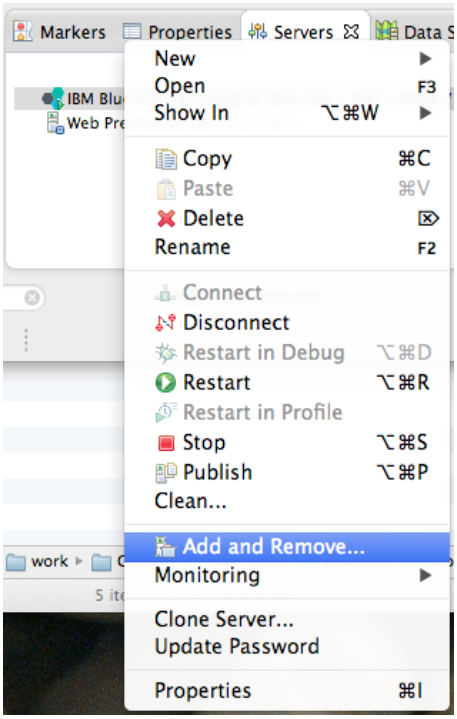


14. When you see the facets, select the **Node.js Application** facet. Click **OK** to close the dialog.



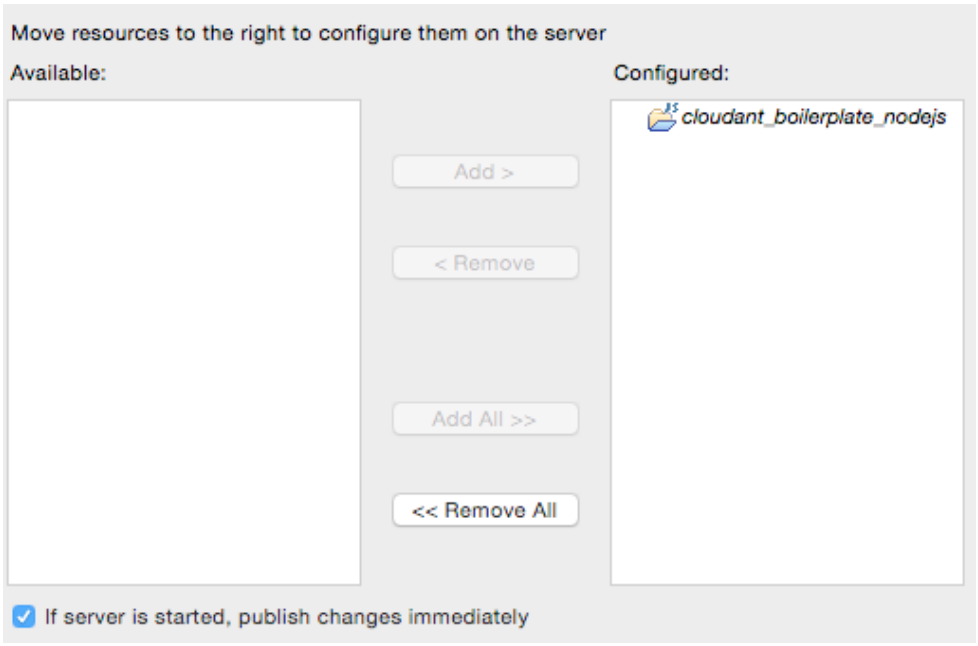
15. Delete the `manifest.yml` file by right-clicking it and clicking **Delete**.

16. Deploy the application. Right-click **IBM Bluemix**. If you have multiple definitions in the Servers view, select the server definition for the space that you want to deploy the application to.



17. Click **Add and Remove**.

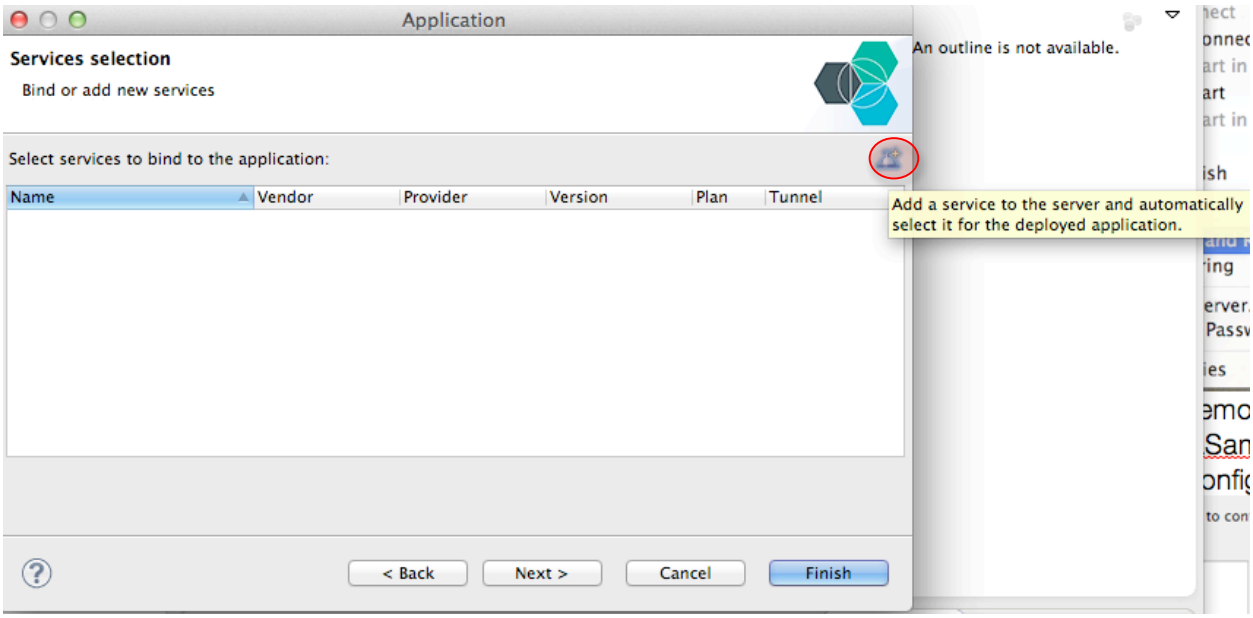
18. Select the **cloudant_boilerplate_nodejs** in the **Available** column and click **Add** to move it to the Configured column.



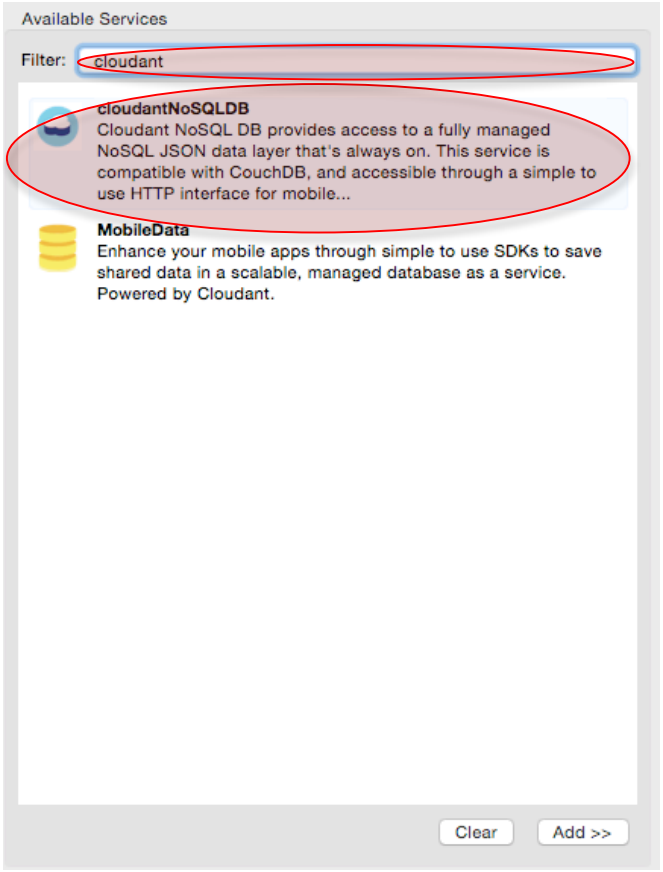
19. Click **Finish**.

20. In the Deploy dialog, change the application name to something unique. Click **Next**. Ensure that the URL contains a string that will be unique and then click **Next**.

21. On the Services selection window, click the **Add a Service** ( ) icon to add a service.

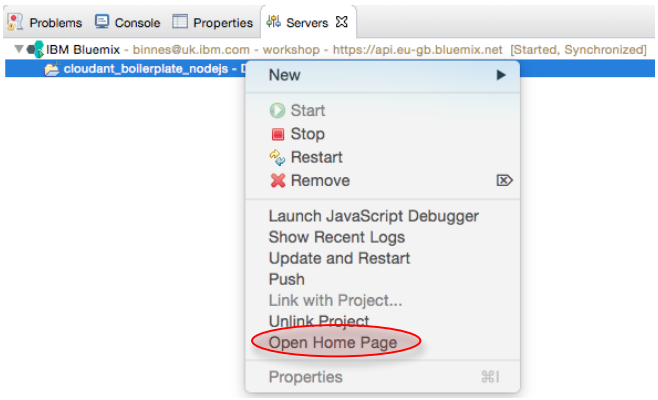


22. When the list of available services is shown, select the **CloudantNoSQLDB** service.



23. Click **Finish** and **Finish** again to close the application Deploy dialog.

The application is now being deployed to Bluemix. Eclipse should automatically switch to the Console view where you see details of the deployment.

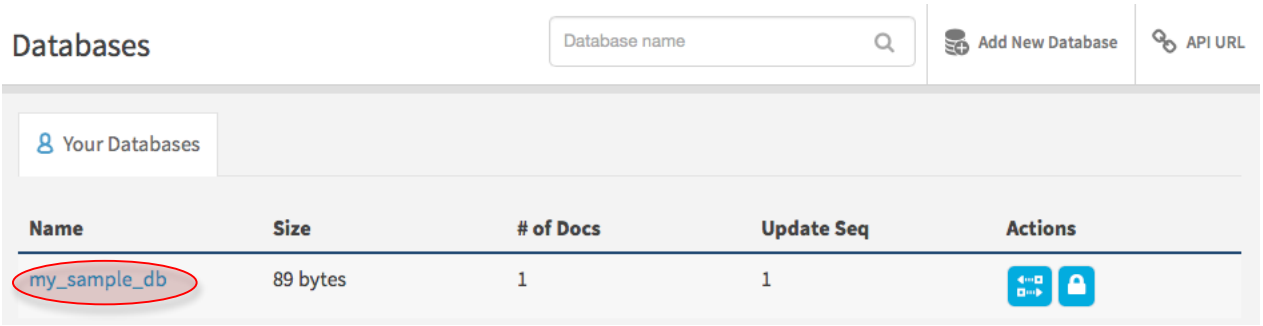


24. After the application is running, switch to the Server view and expand the **Bluemix** server. You should see the application in the list. Start the application from Eclipse by right-clicking the application in the Servers view and clicking **Open Home Page**.

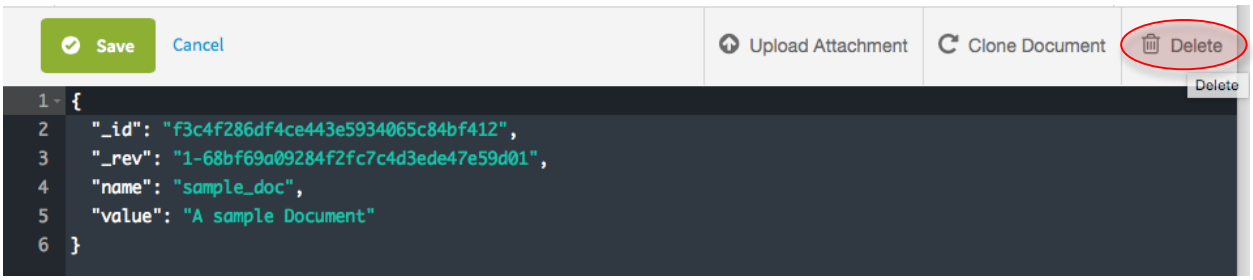
Tip: In Eclipse, you can change the browser that's used to start applications by clicking **Window > Web Browser** and selecting your preferred browser.

25. Remove the sample document from the database to allow the application to create it when the database launches. In the Bluemix web UI, select the Cloudant Service instance and then launch the Cloudant Dashboard.

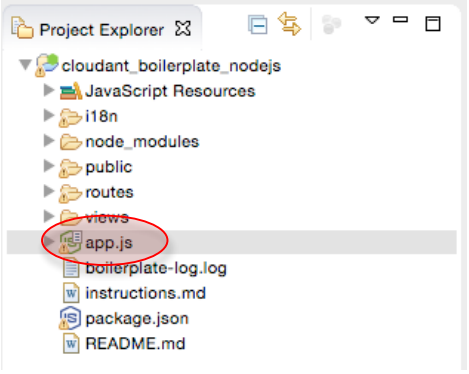
You should see a single database:



26. Select the database and then delete the document. Confirm the deletion.



27. In a text editor, open the file `app.js`:



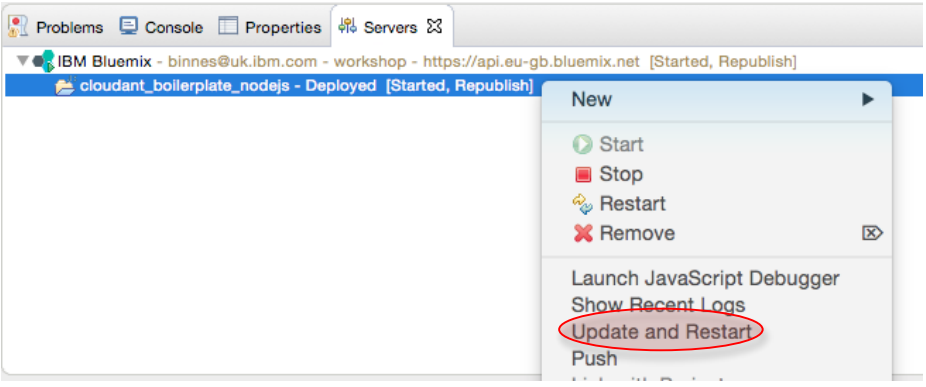
28. Modify the name of the file, the file description and value (lines 306, 307, and 310) to replace `sample` with `test`:

```
303       if(len == 0) {
304         //push sample data
305         // save doc
306         var docName = 'test_doc';
307         var docDesc = 'A test Document';
308         db.insert({
309           name : docName,
310           value : 'A test Document'
311         }, '', function(err, doc) {
```

29. Save the changes by clicking **File > Save**.

Notice in the Server view that the state of the Bluemix server has changed to **republish**, which means that an application has changed. However, the application has not yet been published to Bluemix.

30. Select the **cloudant_boilerplate_nodejs** application in the Server view and right-click and click **Update and Restart**.



31. After the application is restarted, test the application to ensure that the change is now live.

After the project has been tested, the application can be deleted to release resources for future exercises.

32. Right-click the project in the Explorer view and click **Delete** to delete the project. Delete project contents on disk and then click **OK**.

You are asked whether you want to delete the cloudantNoSQLDB service. Select the checkbox to delete the service. In the Bluemix web UI, confirm that the application and server have been deleted.

Important: Deleting a project that is deployed through Eclipse also deletes it from Bluemix.