



## **Lab: working with Eclipse and Bluemix**

## Overview

This lab introduces you to using Eclipse to manage the development of projects that can be deployed to Bluemix. It will show how a sample Node.js application can be imported as a project to Eclipse, updated by using the Eclipse workbench, and published to IBM® Bluemix®.

## Prerequisites

You need the following account and software:

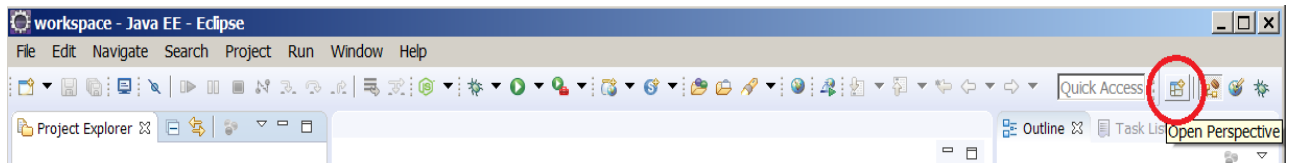
- An [IBM Bluemix account](#)
- Eclipse (Neon, Mars, or Luna versions) with the IBM Bluemix tools for Eclipse and a Node.js plugin (either IBM Node.js Tools for Neon or Enide.p2f for Mars and Luna)
- An Internet Explorer, Firefox, or Chrome web browser

See the prerequisite installation videos at the start of this course for more information.

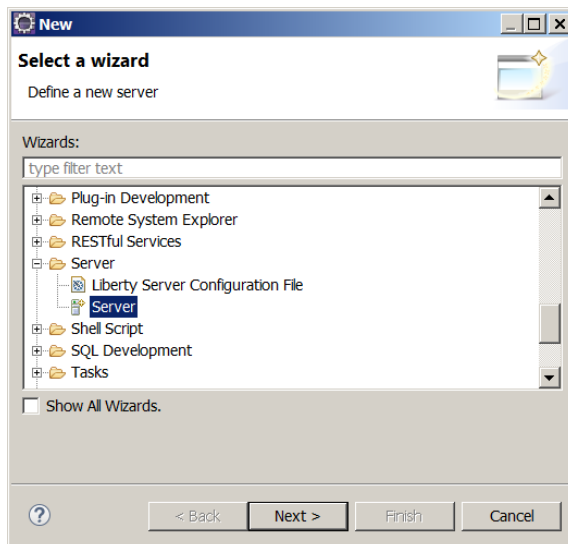
## Step 1. Add a Bluemix server to your Eclipse workspace

Before you publish an Eclipse project to Bluemix, you need to configure one or more servers in Eclipse. A server is needed for each organization and space where you will manage applications in Bluemix.

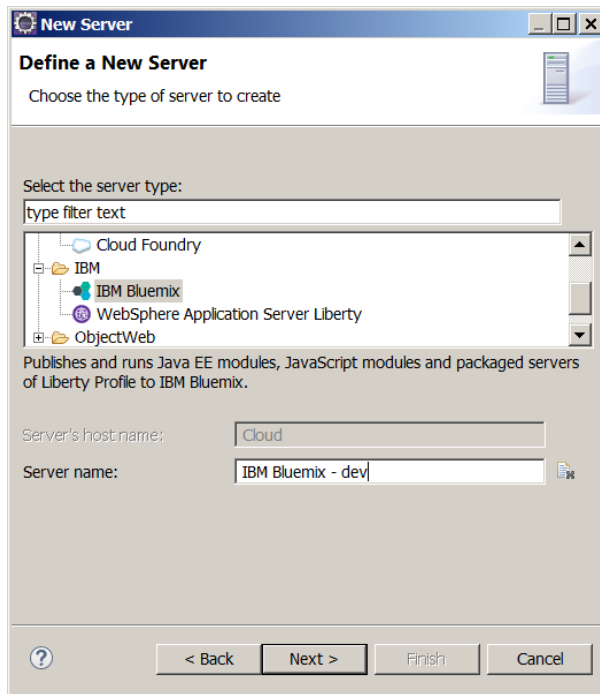
1. Launch Eclipse. You might want to use a new workspace, but this is not required. Go to the workbench if you're not already there and then click **Open Perspective** and select the **Node** perspective.



2. Configure Eclipse to use Bluemix as a server by clicking **File > New > Other** and select the **Server** wizard from the list. Then, click **Next**.

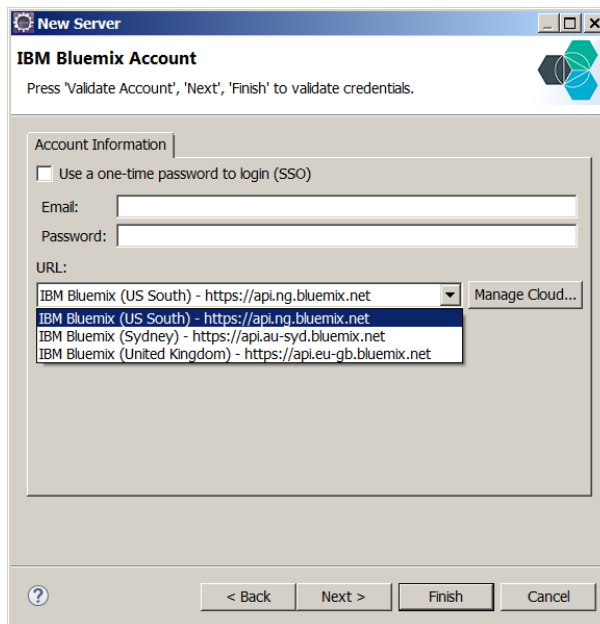


3. Expand **IBM**, select **IBM Bluemix** as the server type, and edit the name for the server. You might want to add the Bluemix space that is associated with this Eclipse server entry.

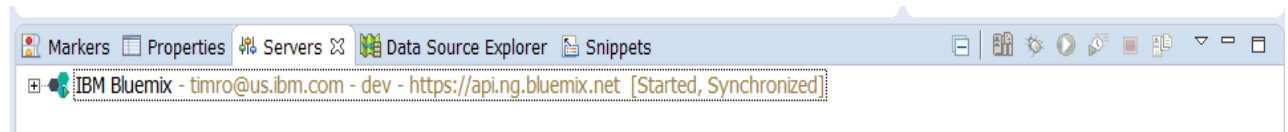


4. Click **Next**.

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



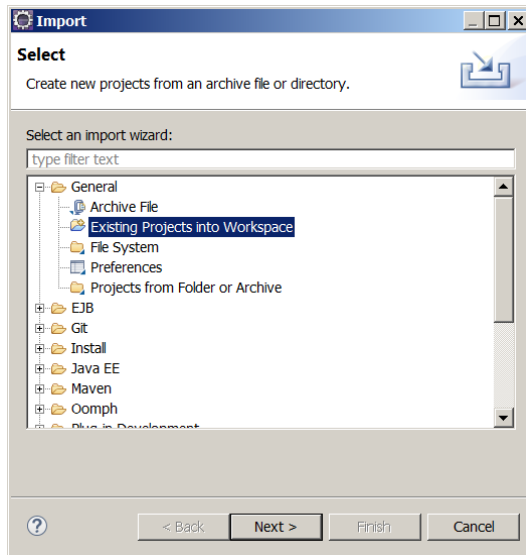
6. Enter your email and password, which are the credentials that you used to log into Bluemix. Or if your organization uses enterprise federation for IBM Web id, select **Use a one-time password**, follow the link, and enter the code provided. Click **Validate Account** to ensure that all details are 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**.



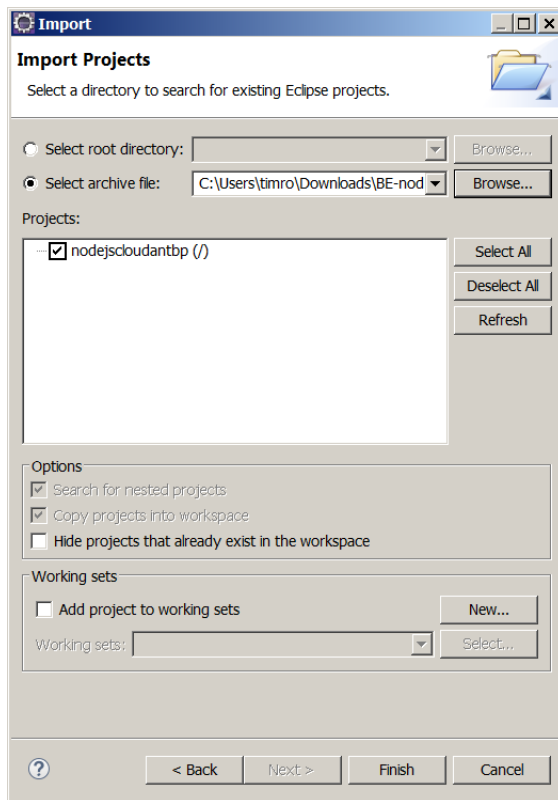
## Step 2. Add an application to your Bluemix workspace

For this step, you will use the same application that you used in the last section. You should have the archive file that you downloaded in the last lab on your workstation. If not, deploy the Node.js Cloudant Web Starter application from the Bluemix web UI, click **Getting Started**, and then download the starter application.

1. Import the starter application package to Eclipse by clicking **File > Import**.
2. In the Import dialog, click **General > Existing Projects into Workspace**.



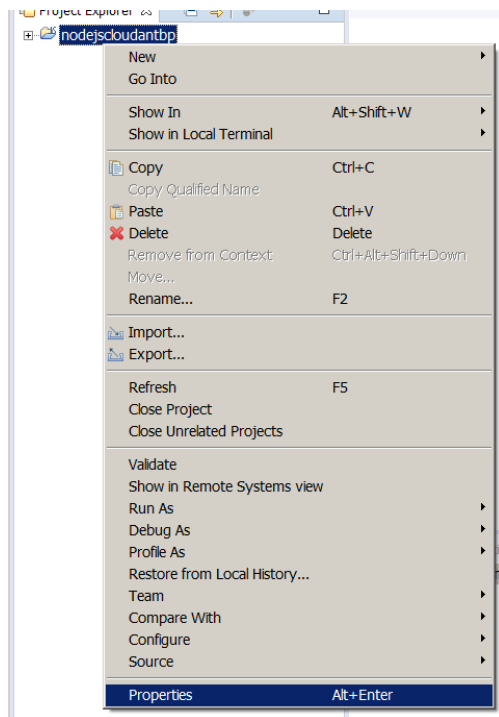
3. Click **Next**. Choose **Select archive file** and then select the downloaded Zip file and click **Finish**. Note that the project name shown might differ from the one shown, but it does not need to be changed to match the following image.



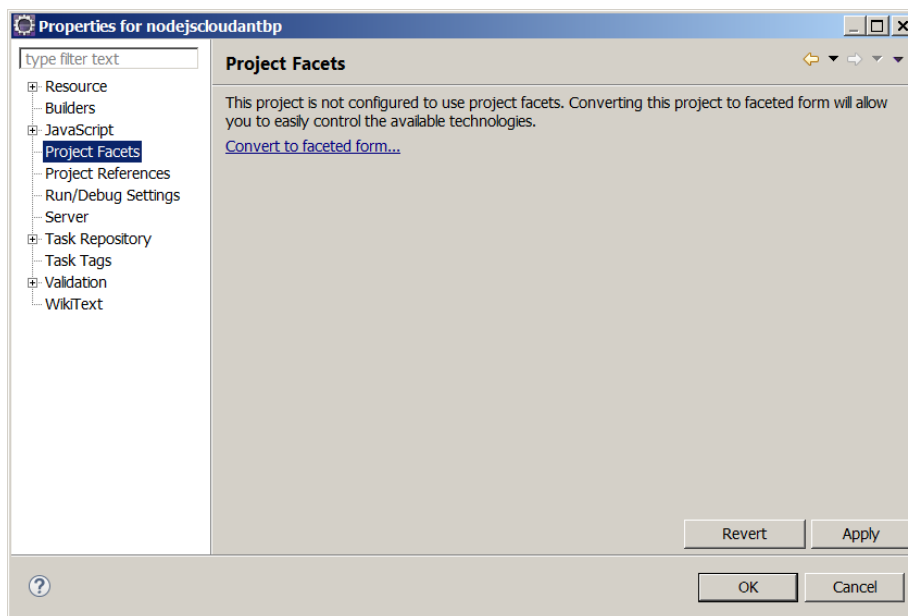
The project will be created.

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

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

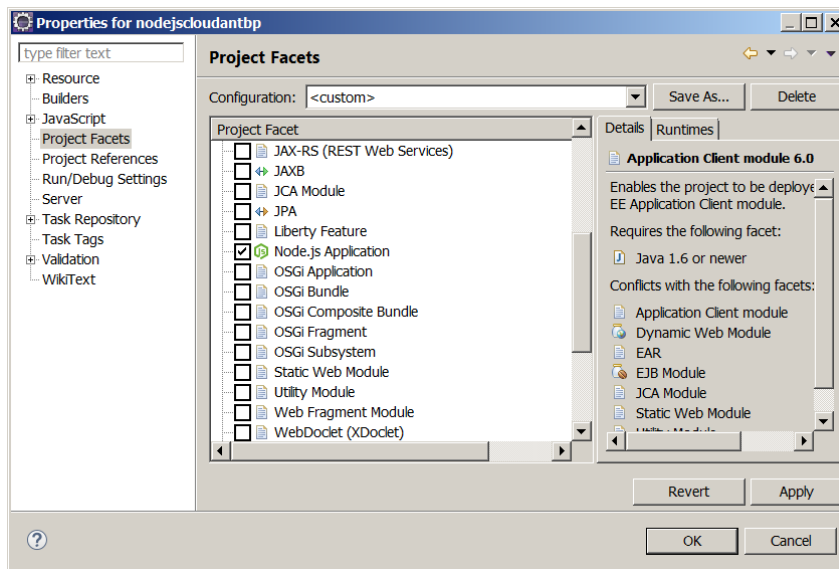


5. Click **Project Facets > Convert to faceted form**.



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



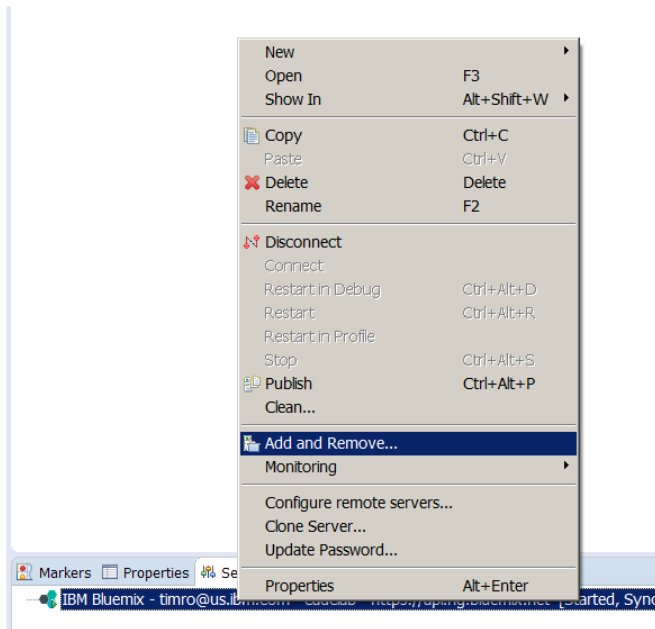


7. Expand the project, find and remove the `manifest.yml` file by right-clicking it and clicking **Delete**.

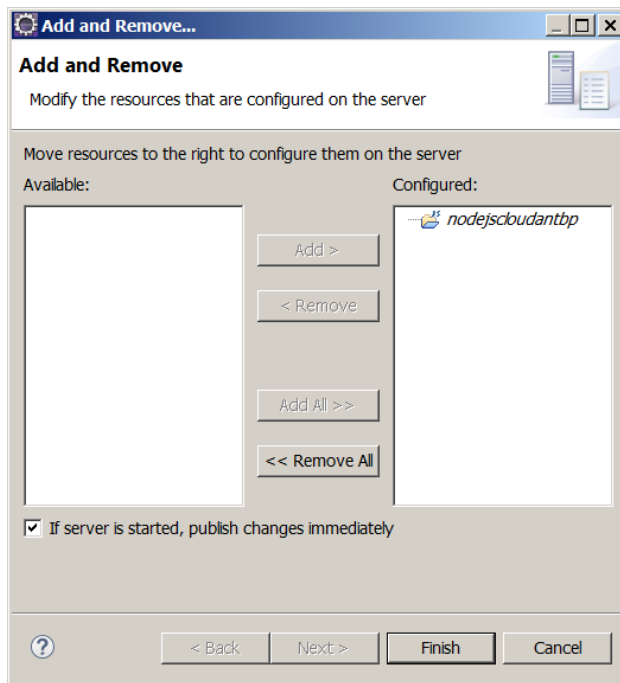
## Step 3. Deploy the application to Bluemix


In this step, you will deploy the application to Bluemix.

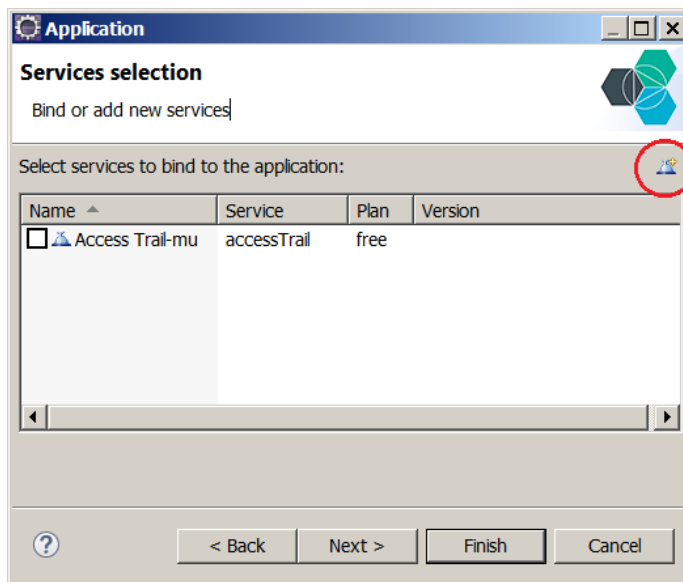
1. In the **Servers** tab right-click **IBM Bluemix** and select **Add and Remove**. If you have multiple definitions in the Servers view, right-click the server definition for the space that you want to deploy the application to.



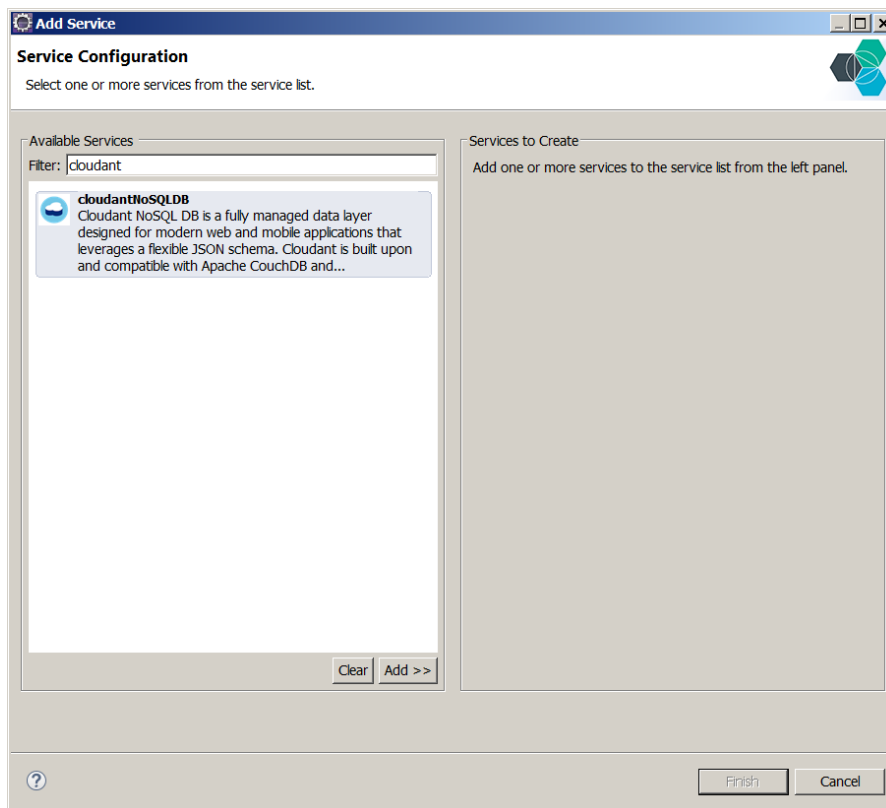
2. Select the project, which might differ from the one below, in the **Available** column and click **Add** to move it to the **Configured** column.



3. Click **Finish**.
4. In the Deploy dialog, change the application name to something unique. Click **Next**. Ensure that the Deployed URL contains a string that will be unique and then click **Next**.
5. In the Services selection window, click the **Add a Service** (  ) icon to add a service.

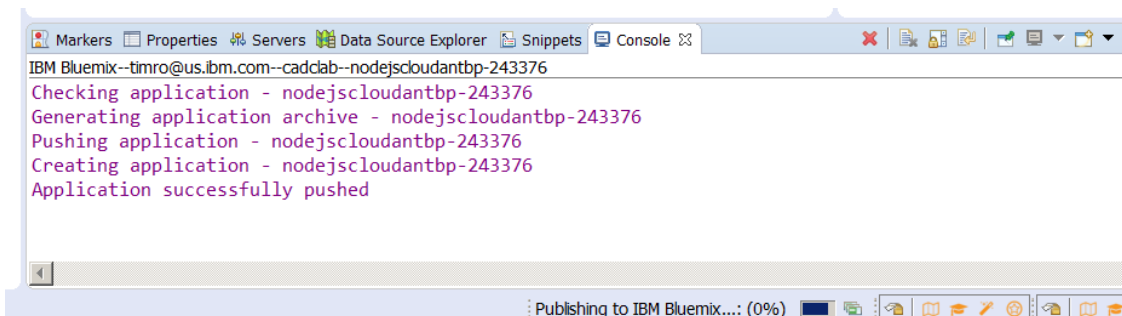


6. When the list of available services is shown, select the **CloudantNoSQLDB** service. To make it easier to find, type `cloudant` into the filter.



7. Update the name if desired and take the default (Lite) plan. Click **Finish** and **Finish** again to close the service selection dialog and begin the deploy.

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



8. 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.

## Step 4. Modify the application and republish to Bluemix

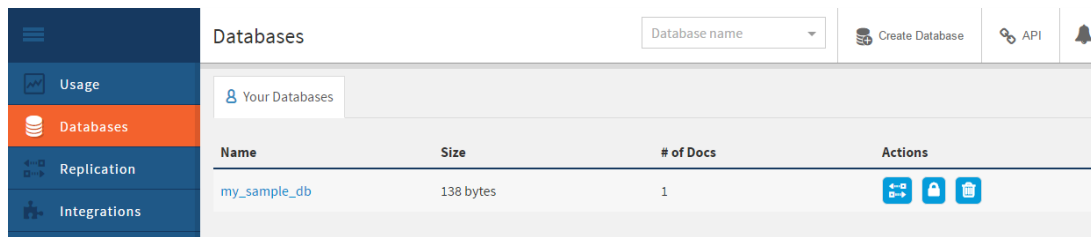
Use these steps to remove the sample document (repeating the steps from the cf CLI lab) from the database to allow the application to create it when the application launches.

1. In the Bluemix web UI, select the Cloudant Service instance and then launch the Cloudant Dashboard.

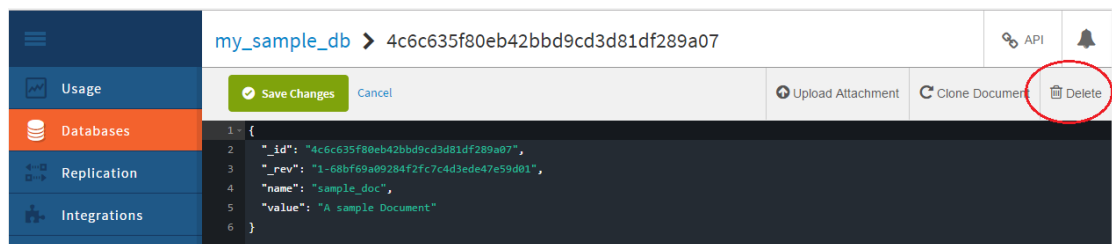
Cloudant NoSQL DB

LAUNCH

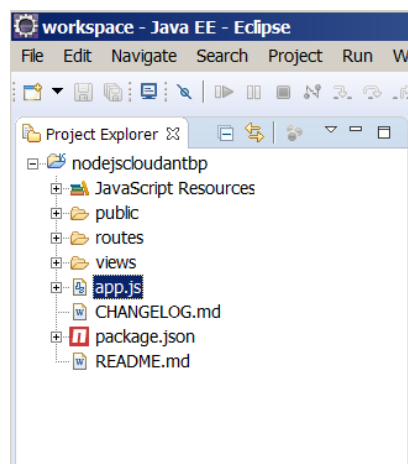
You should see a single database.



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



3. Open the file `app.js` in an editor window:



4. Modify the name of the file, the file description, and value (lines 344, 345, and 348) to replace `sample` with `test`:

```

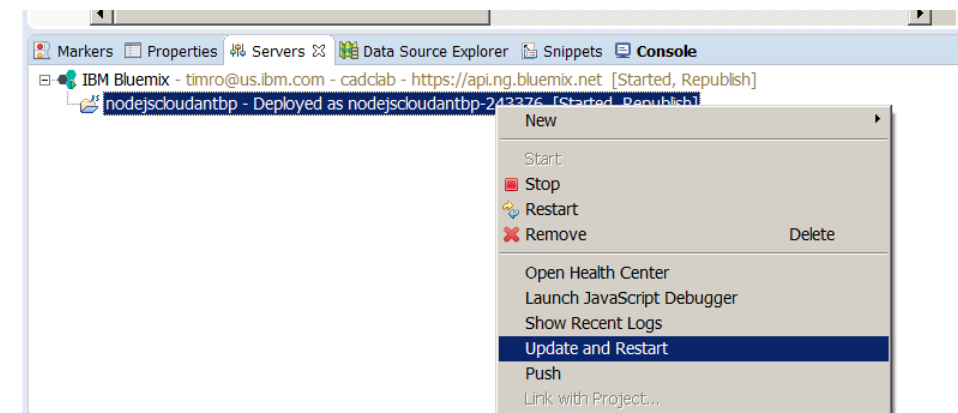
332 console.log("Get method invoked.. ")
333
334 db = cloudant.use(dbCredentials.dbName);
335 var docList = [];
336 var i = 0;
337 db.list(function(err, body) {
338     if (!err) {
339         var len = body.rows.length;
340         console.log('total # of docs -> '+len);
341         if(len == 0) {
342             // push sample data
343             // save doc
344             var docName = 'test_doc';
345             var docDesc = 'A test Document';
346             db.insert({
347                 name : docName,
348                 value : 'A test Document'
349             }, function(err, doc) {
350                 if(err) {
351                     console.log(err);

```

5. 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.

6. Select the deployed application in the Server view and right-click and click **Update and Restart**. (The name shown here might differ).



7. 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 from your Bluemix organization.

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

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

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

## Summary

In this lab, you now know how to configure Eclipse to support development of projects that are published in Bluemix. The Eclipse tools for Bluemix support the development and deployment of Node.js and Java Liberty applications.