



Bluemix Hands-On Workshop

Section 3 - First Deploy Exercises

Version:	5
Last modification date:	7-Jul-15
Owner:	IBM Ecosystem Development

Table of Contents

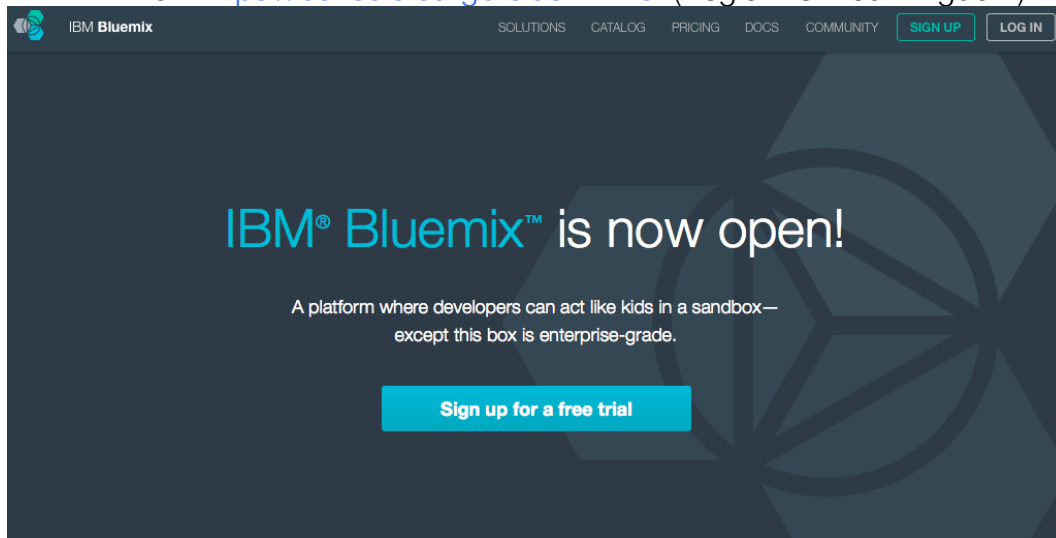
Bluemix Hands-On Workshop.....	1
Section 3 - First Deploy Exercises.....	1
Exercise 3.a – Deploy your first application.....	3
Exercise 3.b - Deploy then update an application using the CLI	6
Exercise 3.c – Working with Eclipse and Bluemix	13

Exercise 3.a – Deploy your first application

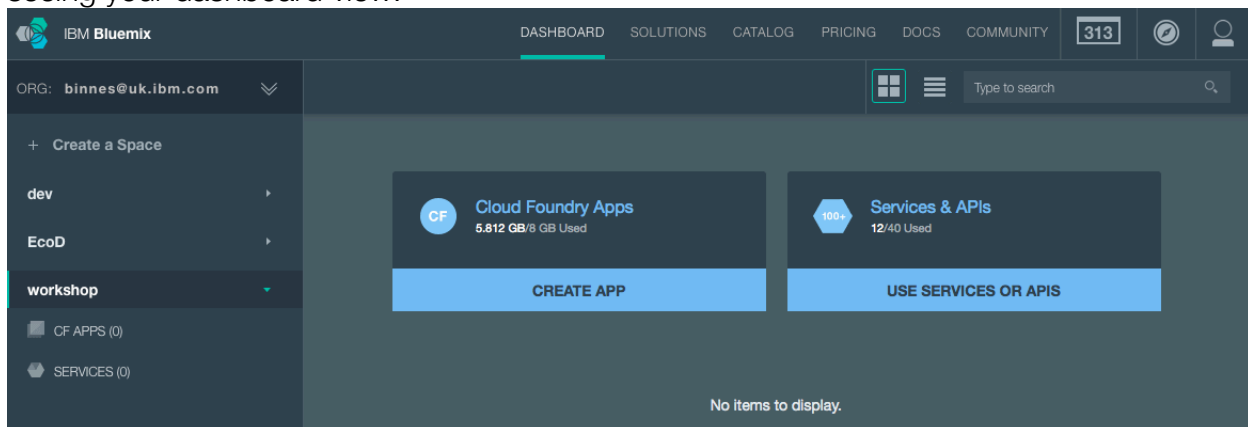
This exercise will show you how to deploy an application from the IBM Bluemix Web User Interface. Once deployed you will download the code for the application, make a change to the code then redeploy using the command line.

In a browser navigate to one of the following:

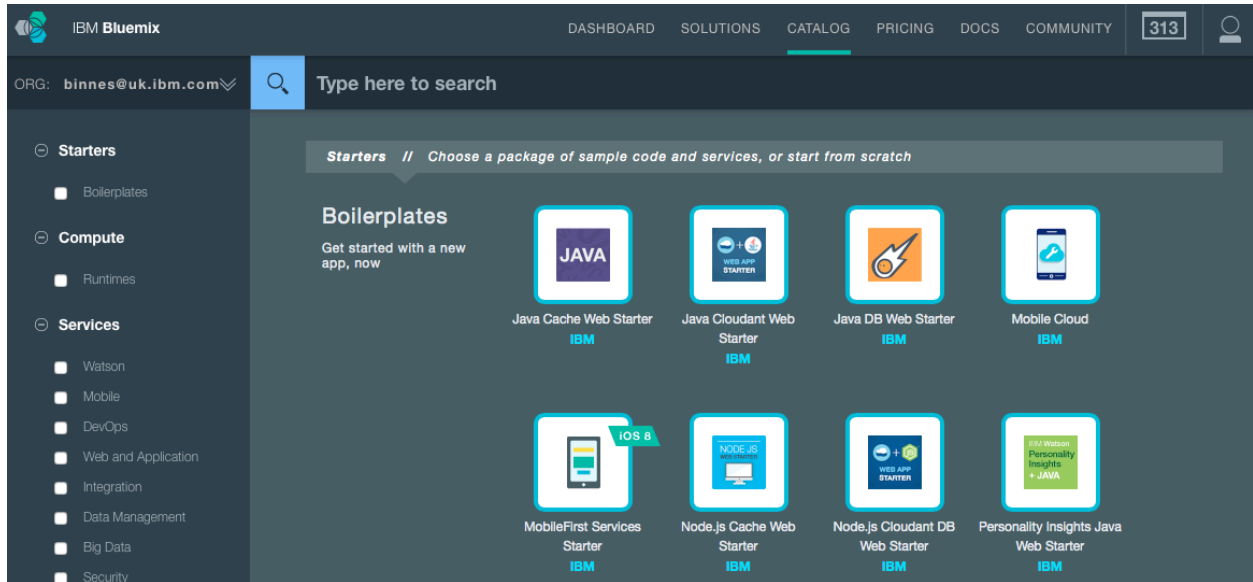
- <https://bluemix.net> - this link should take you to your 'default' location
 - <https://console.ng.bluemix.net> (Region: US South)
 - <https://console.eu-gb.bluemix.net> (Region: United Kingdom)



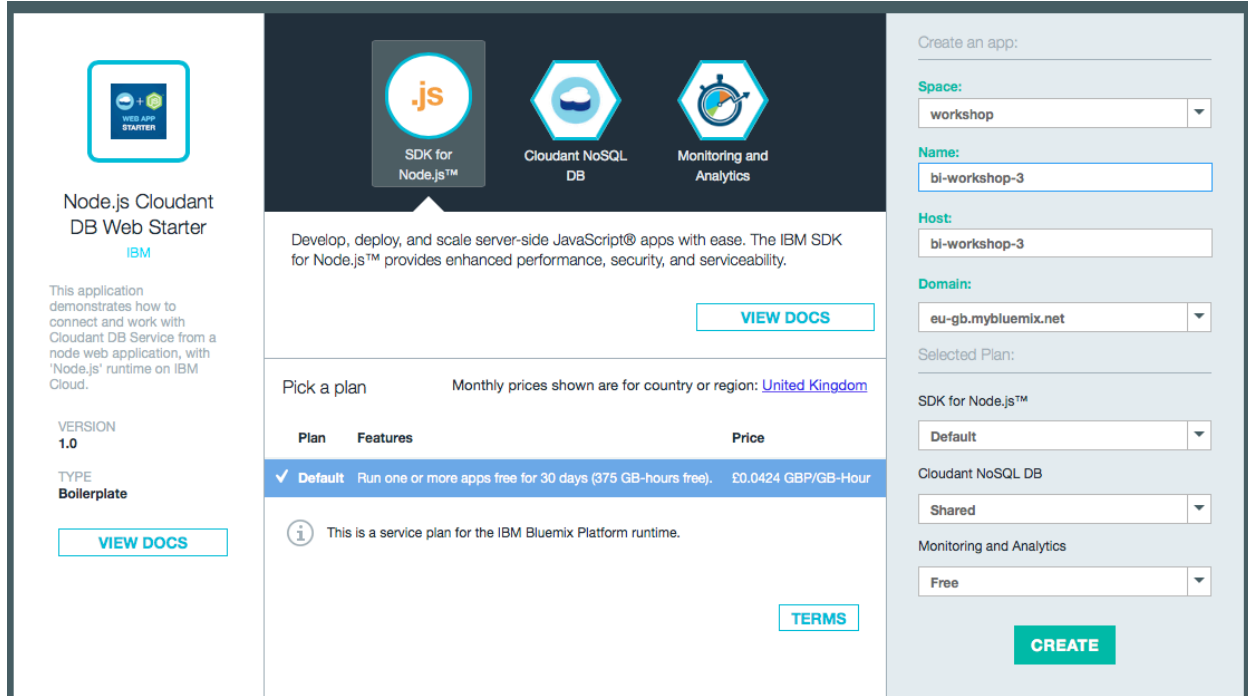
Select 'LOG IN' then enter your log in information and press 'Sign In'. You should be seeing your dashboard view:



Select the 'CATALOG' view :



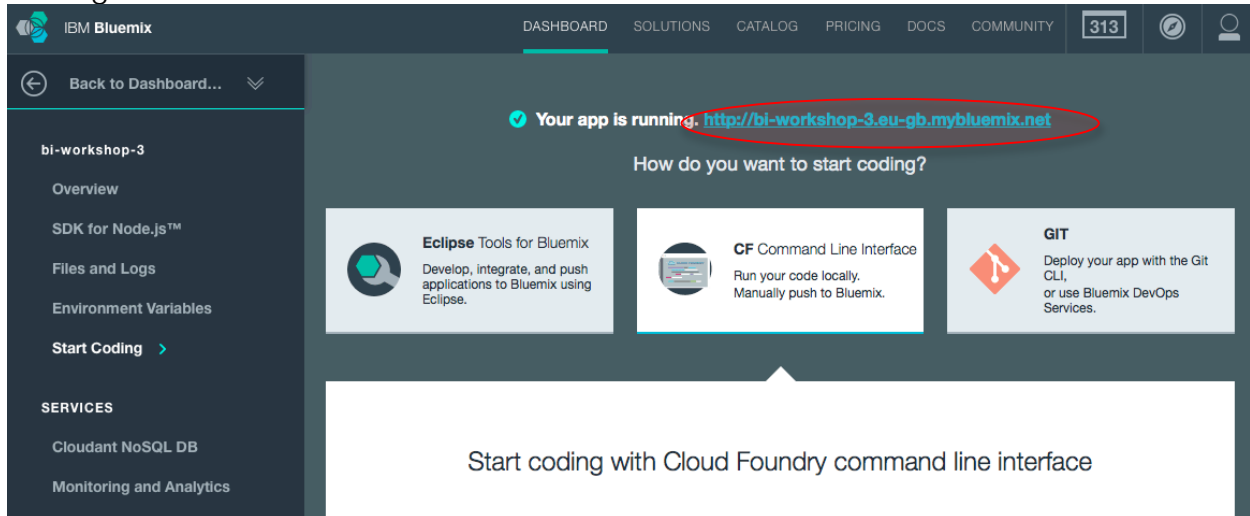
Select the Node.js Cloudant DB Web Starter from the Boilerplates section:



Enter a name for your application, as shown above (host will automatically be completed). The host name must be unique on Bluemix, so please choose a name with your company name or initials to try to make a unique name.

Press 'CREATE'

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



Exercise 3.b - Deploy then update an application using the CLI

Download the starter code to your workstation by selecting 'Download Starter Code'

How do you want to start coding?

Eclipse Tools for Bluemix
Develop, integrate, and push applications to Bluemix using Eclipse.

CF Command Line Interface
Run your code locally. Manually push to Bluemix.

GIT
Deploy your app with the Git CLI, or use Bluemix DevOps Services.

Start coding with Cloud Foundry command line interface

You can use the Cloud Foundry command line interface to deploy and modify applications and service instances.

S Setup: Before you begin, install the cf command line interface.

[Download CF Command Line Interface](#)

Restriction: The Cloud Foundry command line interface is not supported by Cygwin. Use the Cloud Foundry command line interface in a command line window other than the Cygwin command line window.

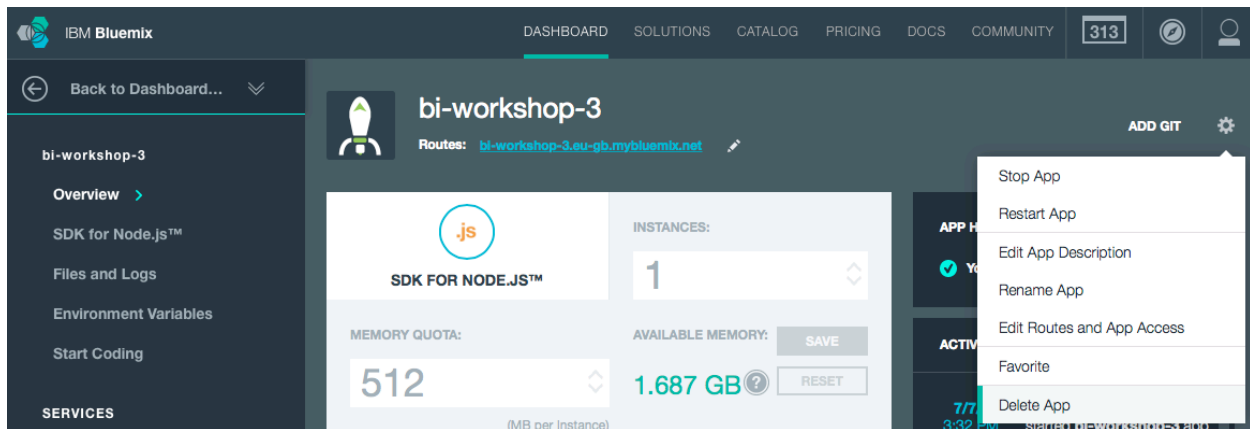
After the cf command line interface is installed, you can get started:

- Download your starter code.
[Download Starter Code](#)
- Extract the package to a new directory to set up your development environment.
- Change to your new directory.

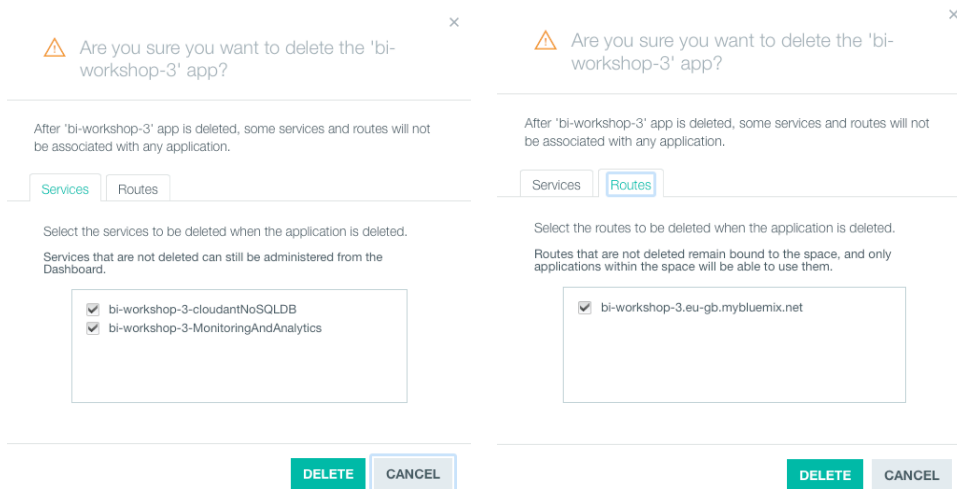
```
cd your_new_directory
```

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. (package.json should be in your current directory)

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

You 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 repeatable 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 document layer that's always on. This service is compatible with CouchDB, and accessible through a simple to use HTTP interface for mobile and web applications
elephantsql	turtle	PostgreSQL as a Service
erservice-beta1	free	IBM Embeddable Reporting for Bluemix provides a mechanism to connect to relational data sources, create reports/dashboards, and embed this service within your application.
loadimpact	1free	Automated and on-demand performance testing
memcachedcloud	25mb	Enterprise-Class Memcached for Developers
mongodb	100	MongoDB NoSQL database
mongolab	sandbox	Fully-managed cloud MongoDB
mlight	Default	Develop responsive, scalable applications with a fully-managed messaging provider in the cloud. Quickly integrate with application frameworks through easy-to-use APIs.
mysql	100	MySQL database
newrelic	standard	Monitor and monitor your apps

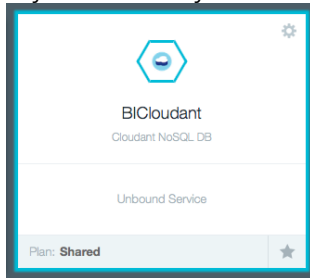
to create the service use command

```
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.

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

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

where:

- BI-myFirstDeploy-3 will be the application name and hostname
- -c specifies the command to start the application
- -m specifies the amount of memory to allocate each application
- --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:

- BI-MyFirstDeploy-3 is the application name used when deploying 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.

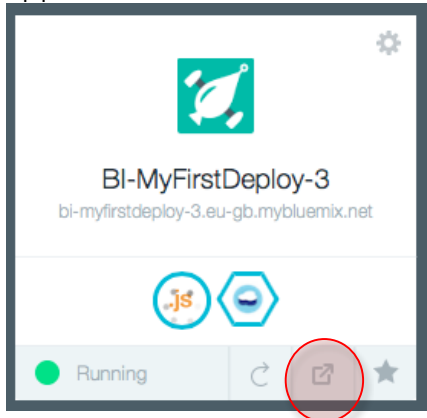
To start an application use the following command – substitute the name of your application:

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

where:

- BlmyFirstDeploy 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 app.js and modify the name of the file and file description (line 306 and 307)

```
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'
```

Change the name from 'sample_doc' to 'test_doc' and the desction from 'A sample Document' to 'A test Document'. Ensure you save the file

We need to remove the sample file from the database to allow it to be populated again, so in the Bluemix Web UI switch into the Dashboard then select your applications.

IBM Bluemix

Dashboard SOLUTIONS CATALOG PRICING DOCS COMMUNITY 313

Back to Dashboard...

BI-MyFirstDeploy-3

Overview >

SDK for Node.js™

Files and Logs

Environment Variables

Start Coding

SERVICES

Cloudant NoSQL DB

Routes: bi-myfirstdeploy-3.eu-gb.mybluemix.net

INSTANCES: 1

MEMORY QUOTA: 128 (MB per Instance)

AVAILABLE MEMORY: 2.062 GB

SAVE RESET

ADD A SERVICE OR API

BIND A SERVICE OR API

Cloudant NoSQL DB

BiCloudant

Shared

Show Credentials

APP HEALTH

RESTART

STOP

ACTIVITY LOG

7/7/15 3:45 PM binnes@uk.ibm.com started BI-MyFirstDeploy-3

7/7/15 3:42 PM binnes@uk.ibm.com updated BI-MyFirstDeploy-3 • changed routes

7/7/15 3:42 PM binnes@uk.ibm.com created BI-MyFirstDeploy-3

Estimate the cost of this app

Select the Cloudant Service instance then launch the Cloudant Dashboard.

Cloudant NoSQL DB

LAUNCH

You should see a single DB:

Databases

Database name

Add New Database

API URL

Your Databases

Name	Size	# of Docs	Update Seq	Actions
my_sample_db	89 bytes	1	1	

select it:

my_sample_db ▾

Permissions

Changes

All Documents +

All Design Docs +

id "b883fb81574144747569faace49f8a76"

```
{
  "_id": "b883fb81574144747569faace49f8a76",
  "_rev": "1-68bf69a09284f2fc7c4d3ede47e59d01",
  "value": {
    "rev": "1-68bf69a09284f2fc7c4d3ede47e59d01"
  },
  "key": "b883fb81574144747569faace49f8a76"
}
```

End of results

then select to delete the document:

b883fb81574144747569faace49f8a76

Save Cancel

Upload Attachment Clone Document Delete

```
1 {
2   "_id": "b883fb81574144747569faace49f8a76",
3   "_rev": "1-68bf69a09284f2fc7c4d3ede47e59d01",
4   "name": "sample_doc",
5   "value": "A sample Document"
6 }
```

confirm the delete when prompted.

Please confirm

Are you sure you want to delete this document?

Cancel Okay

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

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

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

Favorites Organizer

A test Document

test_doc

Choose File no file selected Upload

Click the Add button to add a new category.

ADD

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

```
cf d BI-MyFirstDeploy-3 -r
```

where:

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

```
cf ds BICloudant
```

where

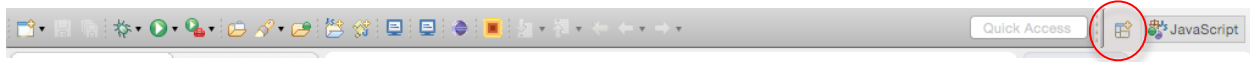
- BICloudant is the name of the service instance to be deleted

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

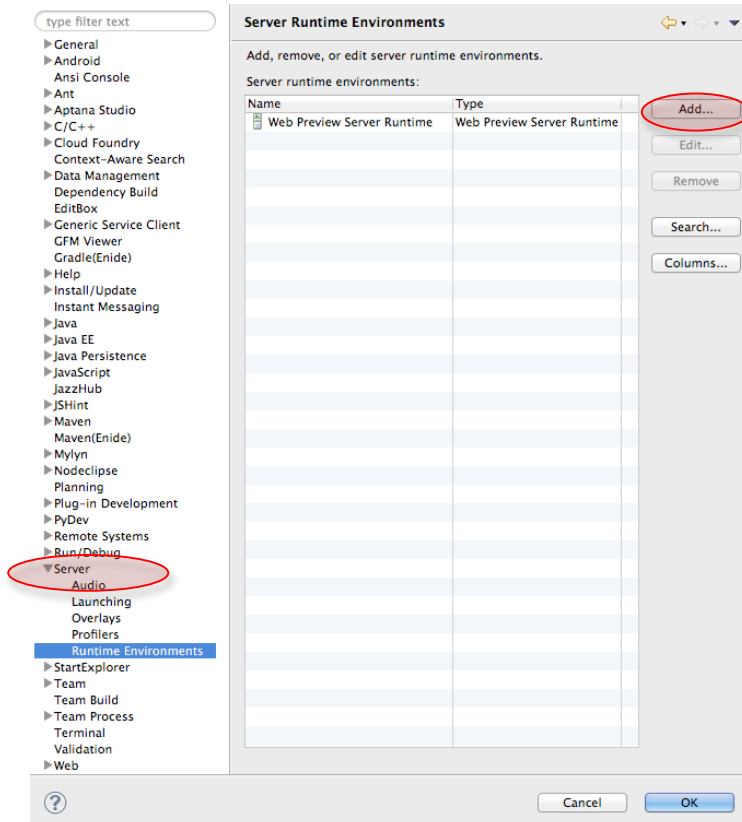
Exercise 3.c – Working with Eclipse and Bluemix

This exercise will show you how to work with Eclipse and Bluemix.

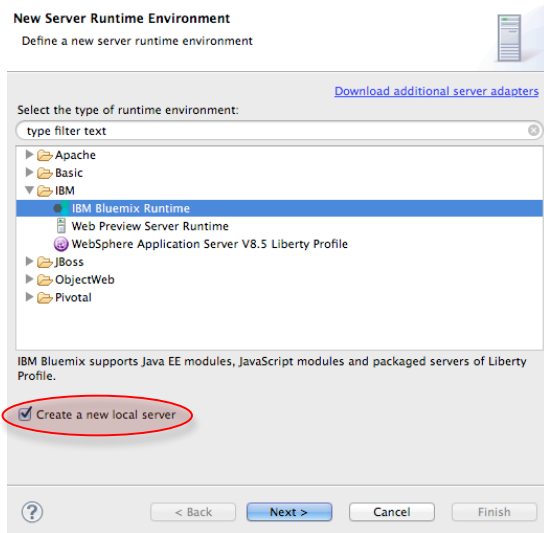
Launch Eclipse - suggest use a new workspace, but not essential. Switch to JavaScript perspective.



First task is to configure Eclipse to use Bluemix as a server, so open up Eclipse Preferences and select Server -> Runtime Environments



select 'Add...' then select IBM Bluemix Runtime and select 'Create a new local server checkbox'

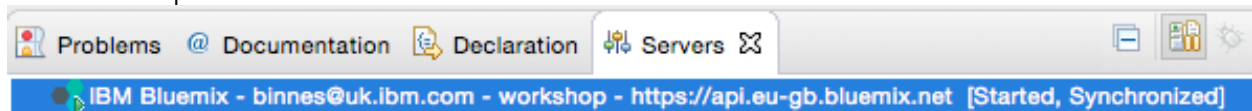


select 'Next >'

Enter your Email and Password – these are the credentials you use to log into Bluemix. Select the correct Bluemix Region from the URL dropdown then select 'Validate Account' to ensure all details are OK then select 'next >'. Choose the organization and space you want to use then select 'Finish'. Select OK to close the preferences page.

Note: If you want to work with multiple spaces in Eclipse you need to create multiple server configurations.

Open the Servers view in Eclipse – if it is not showing use the menu items -> Window -> Show View -> Others. Select the Server folder in the popup that appears then select Servers to open the view:

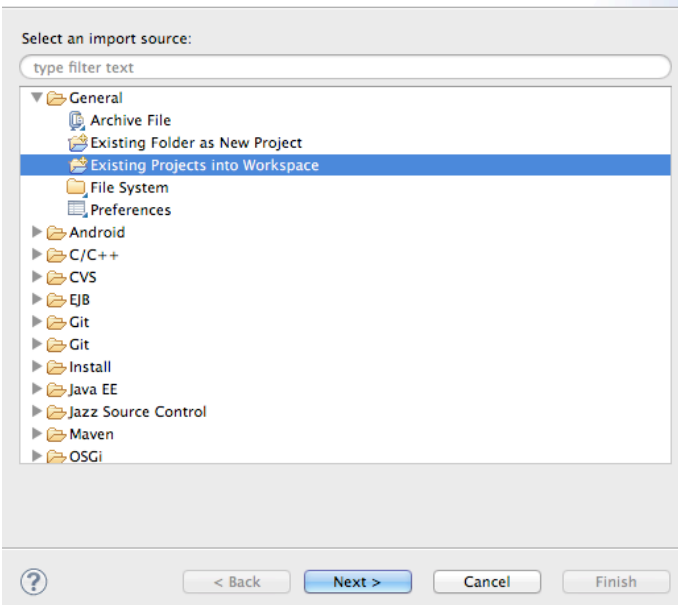


We will use the same application as we used in the last session, so you should have the archive file you downloaded in the last exercise already on your workstation – if not, deploy the Node.js Cloudant Web Starter application from the Bluemix UI, and download the starter application as you did in part a of this exercise.

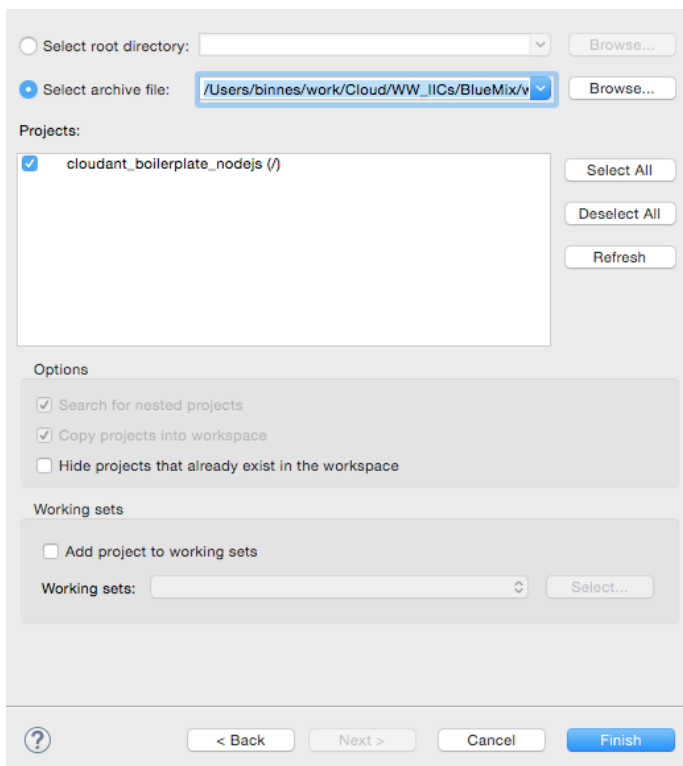
We need to import the starter application package to Eclipse – File -> Import
This will open up the Import dialog. Select General -> Existing Projects into Workspace

Select

Create new projects from an archive file or directory.



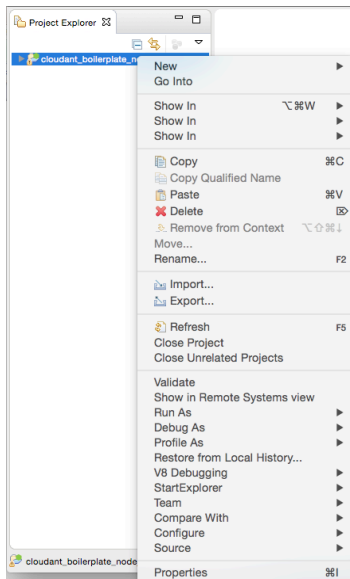
select 'Next >' then select the downloaded zip file and select 'Finish'



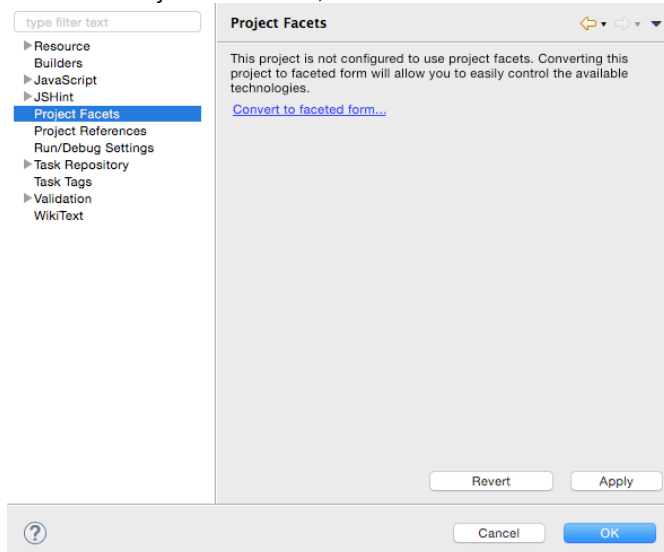
The project will be created

Before JavaScript applications can be deployed to Bluemix by the plugin they need to be identified as a project suitable for Bluemix deployment by assigning a Facet to the project.

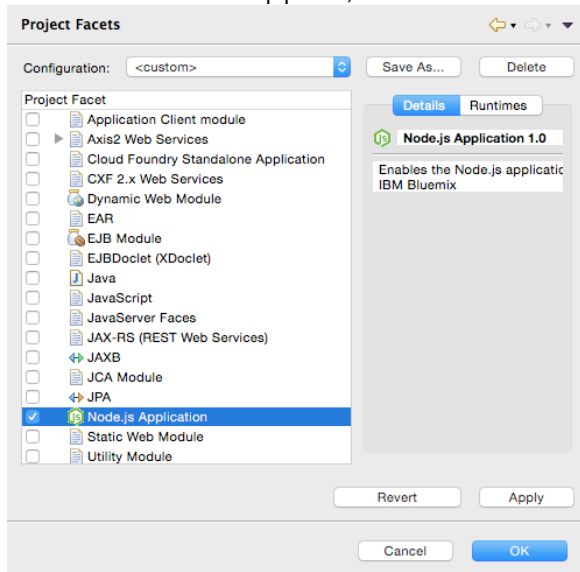
Right-click the project in the 'Project Explorer' view and select Properties from the menu:



Select 'Project Facets', then 'Convert to faceted form...'

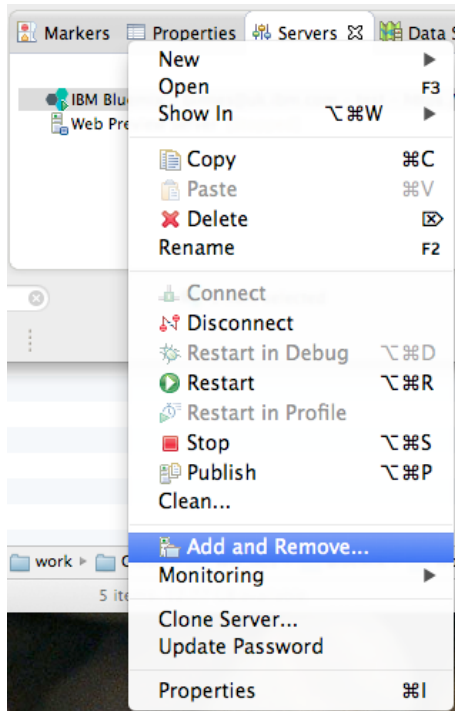


When the Facets appear, check the Node.js Application facet, then OK to close the dialog.

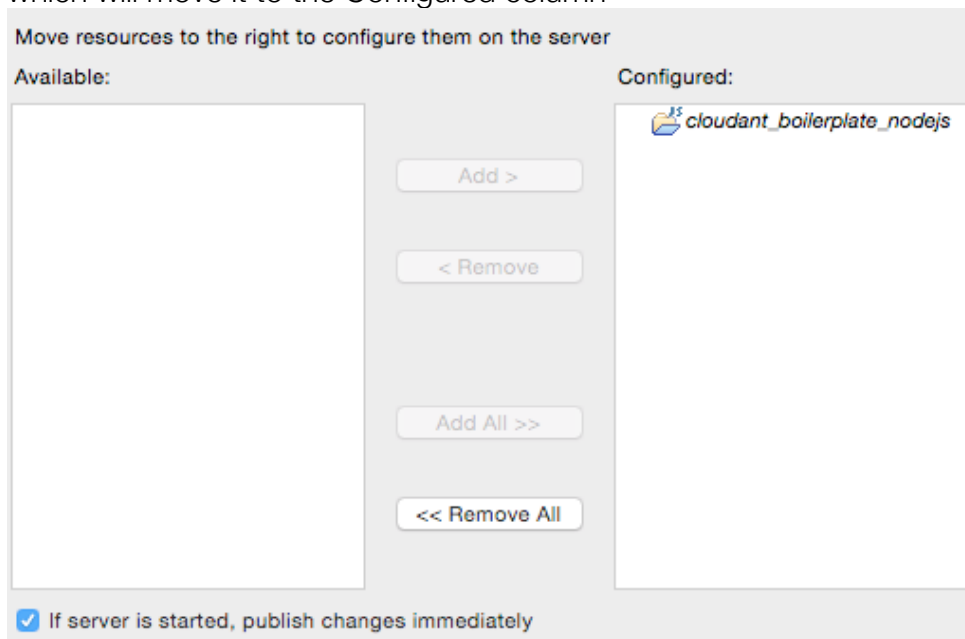



Before deploying the application delete the manifest.yml file by right-clicking on it then selecting delete.

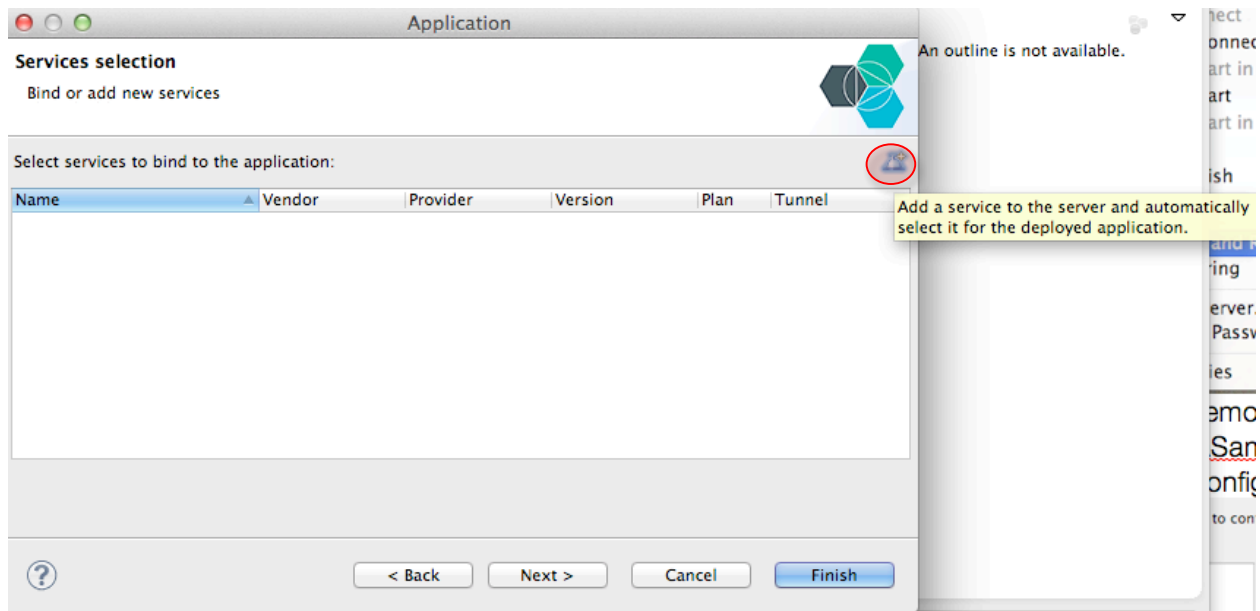
To deploy the application right-click on IBM Bluemix in the 'Servers' view – select the server definition for the space you want to deploy the application to if you have multiple definitions in the Servers view



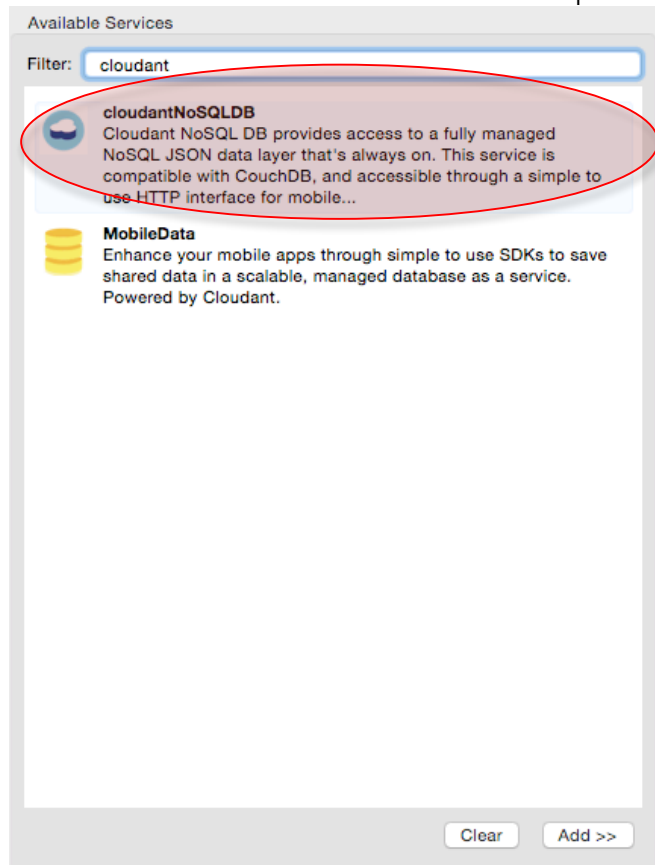
then Add and Remove
select the 'cloudant_boilerplate_nodejs' in the Available column and then select 'Add >', which will move it to the Configured column



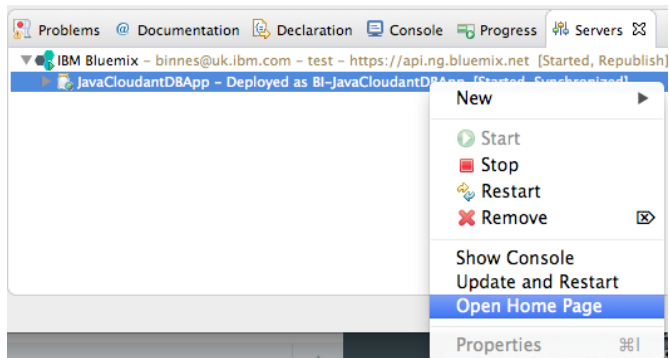
Press finish – this will then launch the deploy dialog. Change the Application name to something unique then press 'Next', check the URL contains something that you think will be unique then press 'Next >'. On the Services selection screen press the  icon to add a service



when the list of available services come up select the CloudantNoSQLDB service



then 'Add >>' to move Cloudant to the right panel. Press Finish and Finish again to close the application Deploy dialog. The application is now getting deployed to Bluemix. Eclipse should automatically switch to the Console view where you will see details of the deploy.



Once the application is running – switch back to the server view and expand the Bluemix server – you should see the application listed. You can launch the application from Eclipse by right-clicking on the application in the servers view and selecting ‘Open Home Page’.

Note: In Eclipse you can alter the Browser used to launch applications from the menu -> Window -> Web Browser - then select the preferred option.

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

Cloudant NoSQL DB

LAUNCH

You should see a single DB

Databases				
Database name <input type="text"/>			Add New Database	API URL
Your Databases				
Name	Size	# of Docs	Update Seq	Actions
my_sample_db	89 bytes	1	1	

select it, then select to delete the document

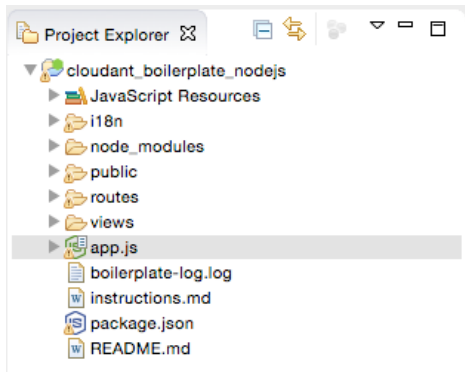
☒ Save

```

1 {
2   "_id": "f3c4f286df4ce443e5934065c84bf412",
3   "_rev": "1-68bf69a09284f2fc7c4d3ede47e59d01",
4   "name": "sample_doc",
5   "value": "A sample Document"
6 }
  
```

confirm the delete when prompted.

Back in Eclipse in the Project Explorer view double click the app.js file, which will open it in an editor view.



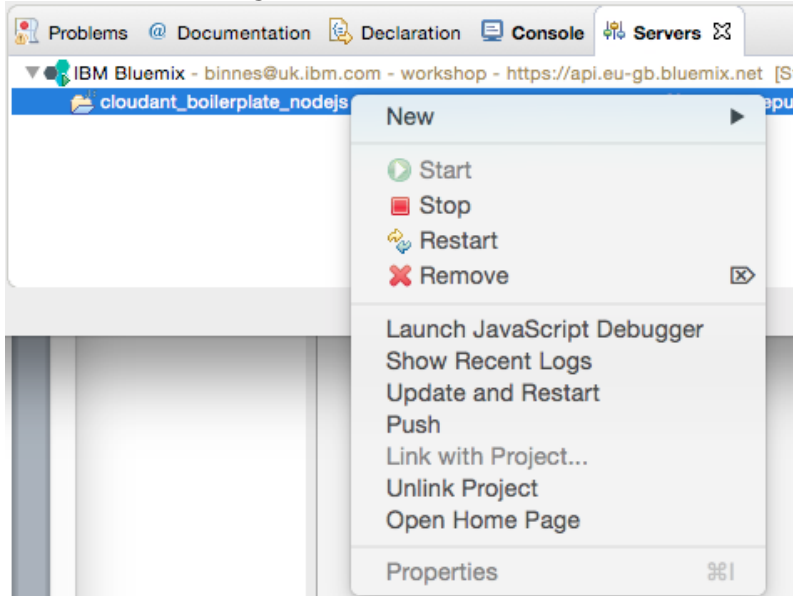
Modify the name of the file and file description, as we did in part a (line 306 and 307)

```

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 sample Document'
311             }, '', function(err, doc) {

```

save the change File->Save then notice in the Server view the state of the Bluemix server has changed to republish – telling us that an application has changed, which is not yet been published to Bluemix. Select the 'cloudant_boilerplate_nodejs' application in the Server view and right-click then select 'Update and Restart'



Once the application has restarted test the application by opening the home page. Check that the change is now live.

To finish this exercise right-click the project in the Explorer view and select Delete to delete the project. Select to delete project contents on disk and then press OK. You will then be asked if you want to delete the cloudantNoSQLDB service – deleting a project deployed via Eclipse will also delete it from Bluemix! Select the checkbox to delete the service. In the Bluemix Web UI confirm that the application and service have been deleted.