

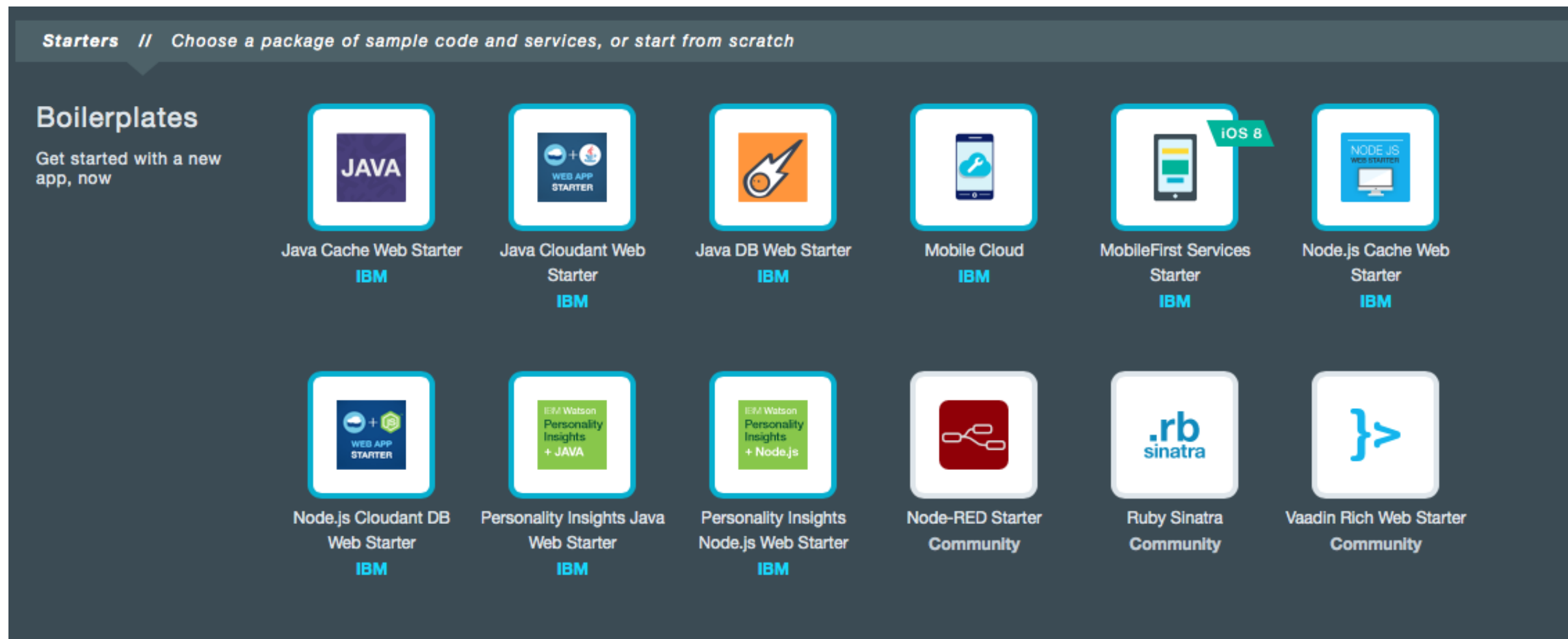


Module 3: Boilerplate applications

Deploying your first application

Boilerplates

- Provide a fast way to get an application started
- Package of sample application code and services



Environment Variables

- Bluemix sets environment variables when deploying an application to inform the application about the environment it is running in.
 - Application may need to use this information to set configuration – such as the hostname and port to listen for incoming requests.
 - VCAP_APP_HOST – specifies the hostname on which your application should listen
 - VCAP_APP_PORT – specifies the port number on which your application should listen
- You can also define custom environment variables to provide configuration to your application

Manifest file

- Allows you to specify the parameters for an application deployment
- manifest.yml file is used at deploy time if found in directory application is being pushed from

applications:

- name: bluemix-todo-node
host: bluemix-todo-node-`\${random-word}`
command: node app.js
memory: 128M
services:
 - todo-db

applications:

- name: Myphpmyadmin
memory: 128M
instances: 1
host: Myphpmyadmin
path: .
buildpack: https://github.com/dmikusa-pivotal/cf-php-build-pack.git
services:
 - mysql_BlueMixLab

<http://docs.cloudfoundry.org/devguide/deploy-apps/manifest.html>

.cfignore

- When you push an application all content of the current directory and all sub directories are pushed to the server – this is not the behavior you always want
- .cfignore allows control of what is sent to the server by listing the files and directories you do not want to send to the server
- Sample .cfignore file:
 - .git
 - node_modules
 - tmp
 - lib-src

Accessing sample code

- Sample code for the application is available to download after a runtime or boilerplate has been deployed.

The screenshot shows the IBM Bluemix dashboard for a project named 'Bltest123'. The left sidebar contains navigation links: Overview, SDK for Node.js™, Files and Logs, Environment Variables, Start Coding (highlighted with a red circle and a callout '1.Click'), SERVICES, Cloudant NoSQL DB, and Monitoring and Analytics. The main content area shows a status message 'Your app is running. http://Bltest123.binnes.me.uk' and a section titled 'How do you want to start coding?' with three options: Eclipse Tools for Bluemix, CF Command Line Interface, and GIT. Below this, the 'Start coding with Cloud Foundry command line interface' section provides instructions. A 'Download CF Command Line Interface' button is shown. A restriction note states: '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.' A numbered list of steps follows: 1. Download your starter code. (The 'Download Starter Code' button is highlighted with a red box and a callout '2.Click'), 2. Extract the package to a new directory to set up your development environment., and 3. Change to your new directory. A terminal snippet shows the command `cd your_new_directory`.

Command-line interface

- Bluemix uses the Cloud Foundry command-line interface (CLI): `cf`
 - `cf help` provides help page showing all the commands
 - `cf help <command>` provides help for specific command

<code>cf l</code>	login	<code>cf ds</code>	delete service
<code>cf t</code>	target space or organization	<code>cf bs</code>	bind service to application
<code>cf a</code>	list apps in current space	<code>cf st</code>	start app
<code>cf app</code>	display status for specific app in current space	<code>cf sp</code>	stop app
<code>cf p</code>	push (deploy or update) app	<code>cf d</code>	delete app
<code>cf s</code>	show service info	<code>cf scale</code>	scale app
<code>cf cs</code>	create service	<code>cf logs</code>	tail or show logs for app

Sample cf commands

To log in to Bluemix:

```
cf l -a https://api.ng.bluemix.net -u <email> -p <password> -o <email> -s dev
cf l -a https://api.eu-gb.bluemix.net -u <email> -p <password> -o <email> -s dev
```

- This command will log in to Bluemix, set the organization to the user's own organization, and the space to dev

To check what space you are logged into or to change the space:

```
cf t
cf t -s test
```

- The first option prints the current target organization and space.
- The second option switches to the test space.

To check what spaces exist in an organization:

```
cf spaces
```

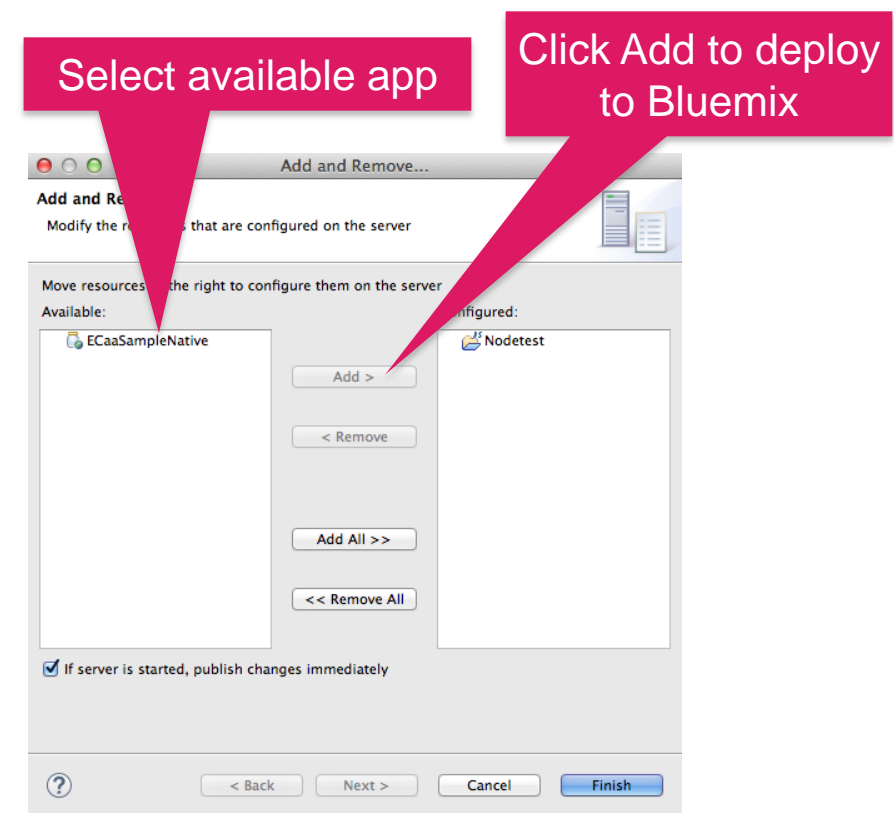
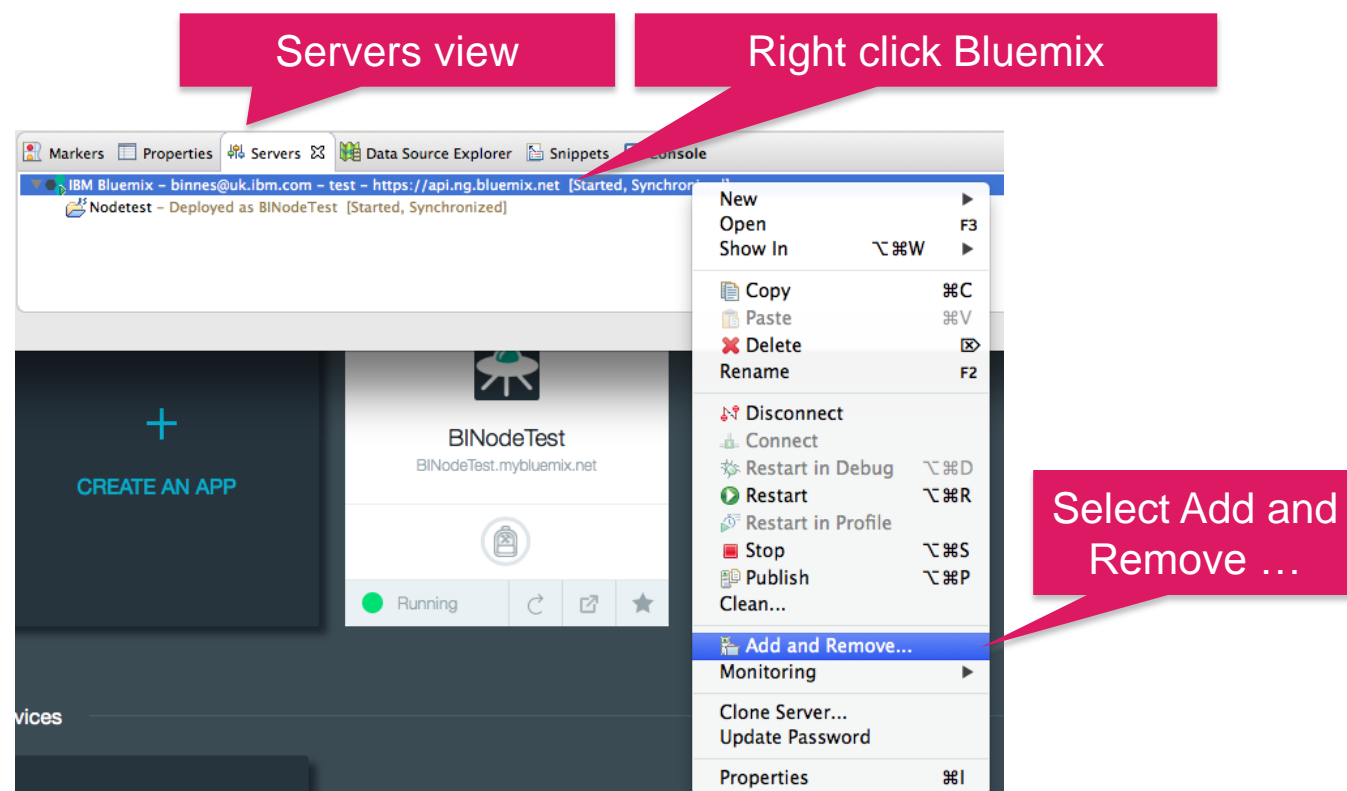
- Displays the spaces available in the current organization.

```
cf space dev
```

- Displays information about the dev space in the current organization.

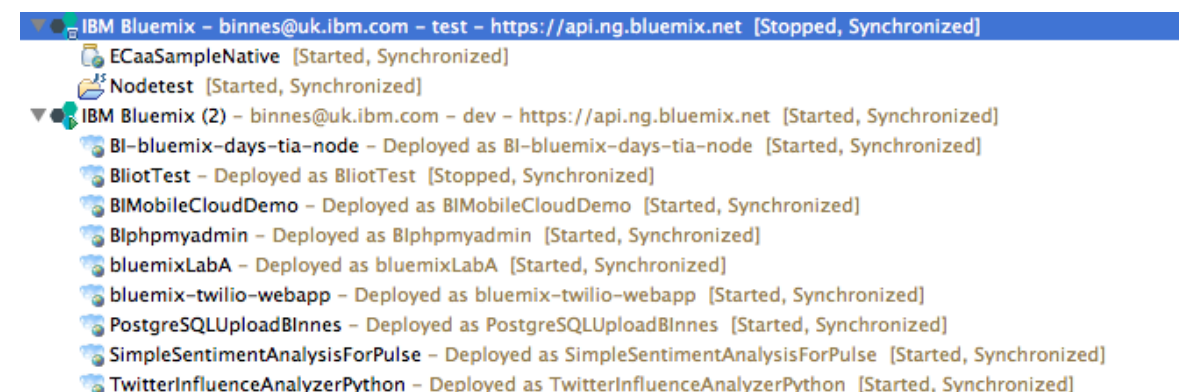
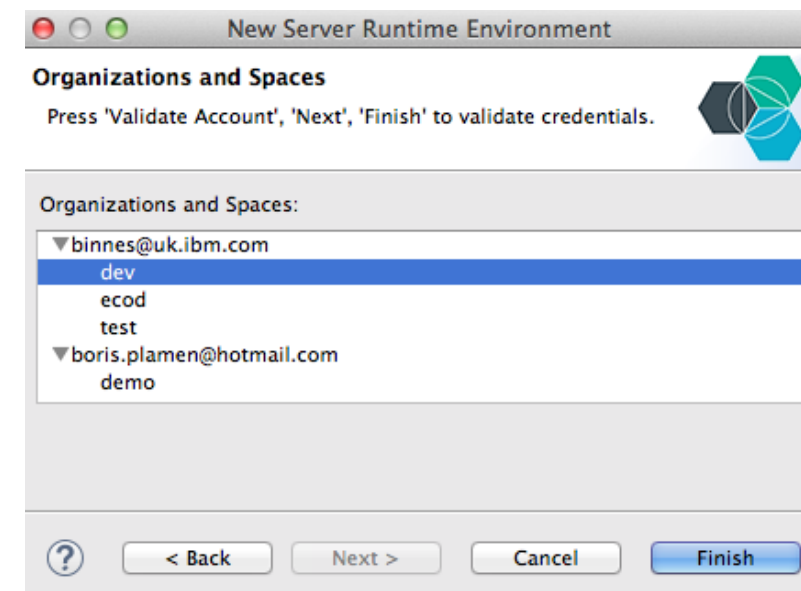
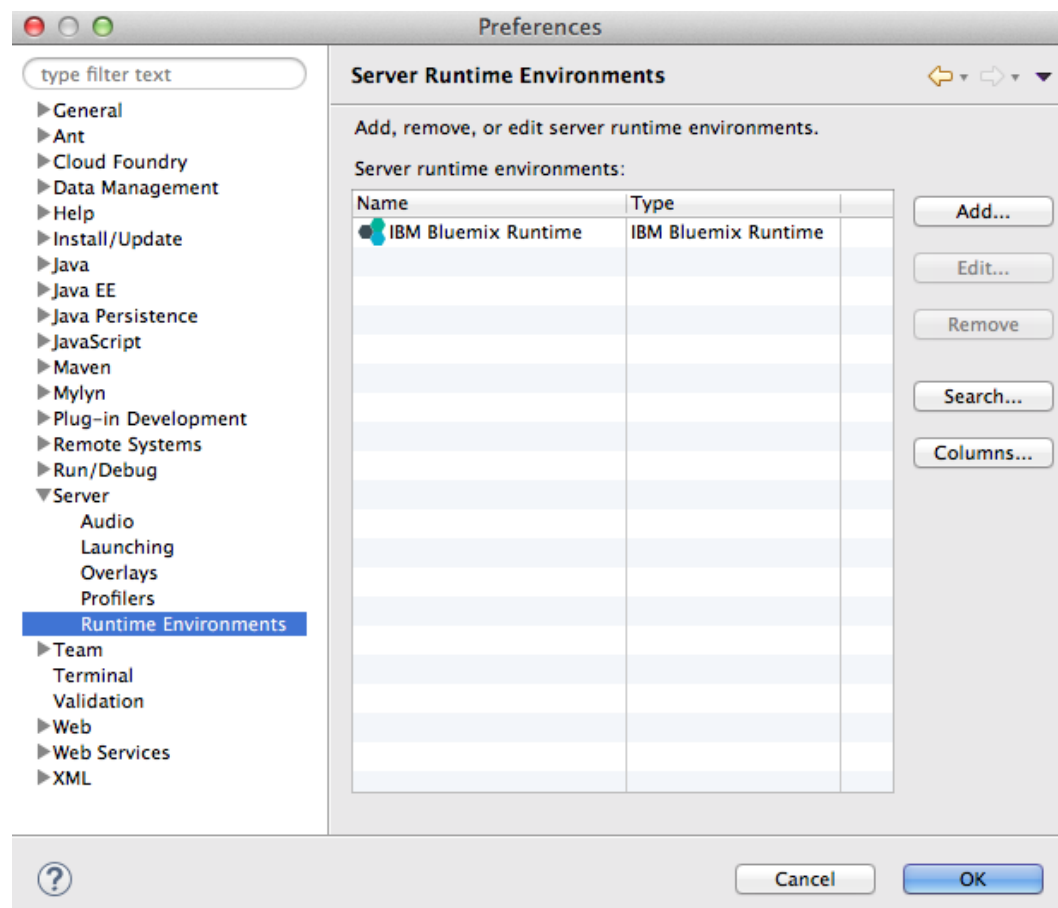
Eclipse plug-in for Bluemix

- Available in the Eclipse marketplace
- Enables developers to develop in Eclipse and then deploy to Bluemix
 - Java and JavaScript are supported



Define your Bluemix server connection in Eclipse

- In Eclipse preferences, select **Server** then **Runtime Environments**.
- Add a server definition for each Bluemix space that you want to deploy to.

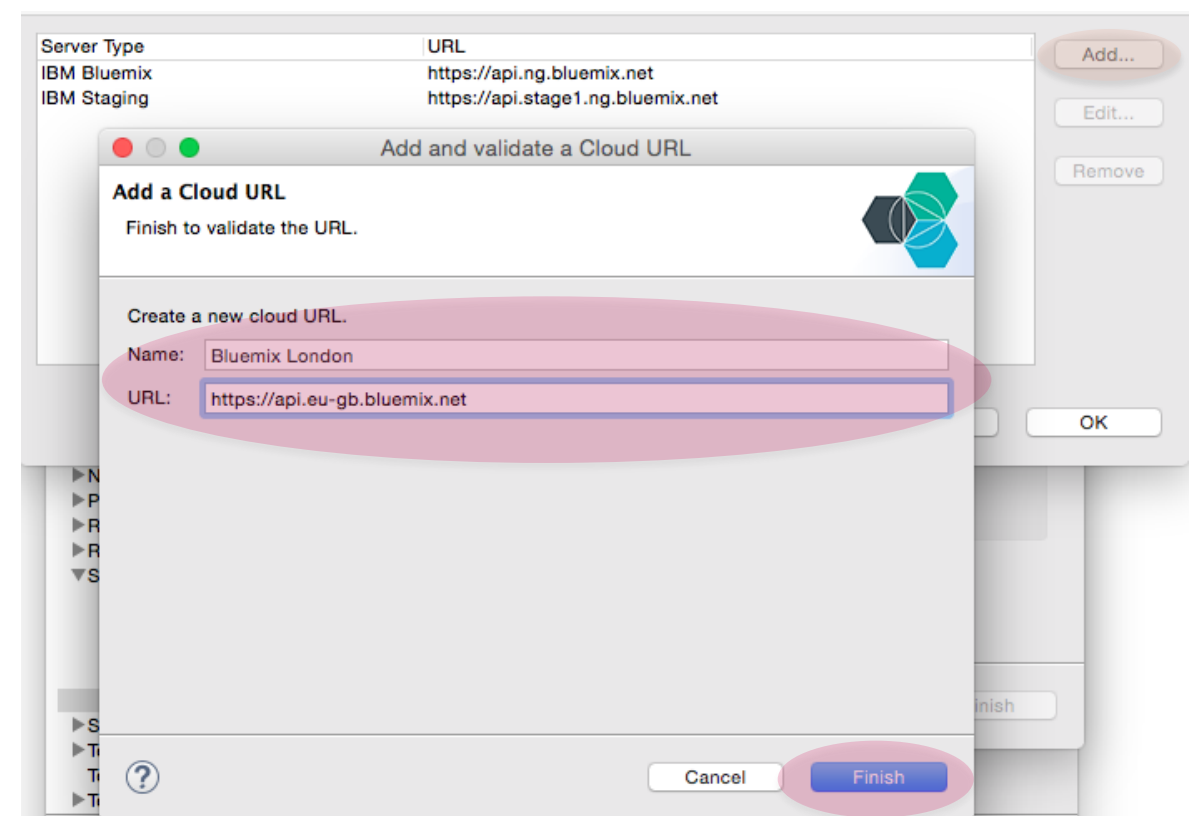
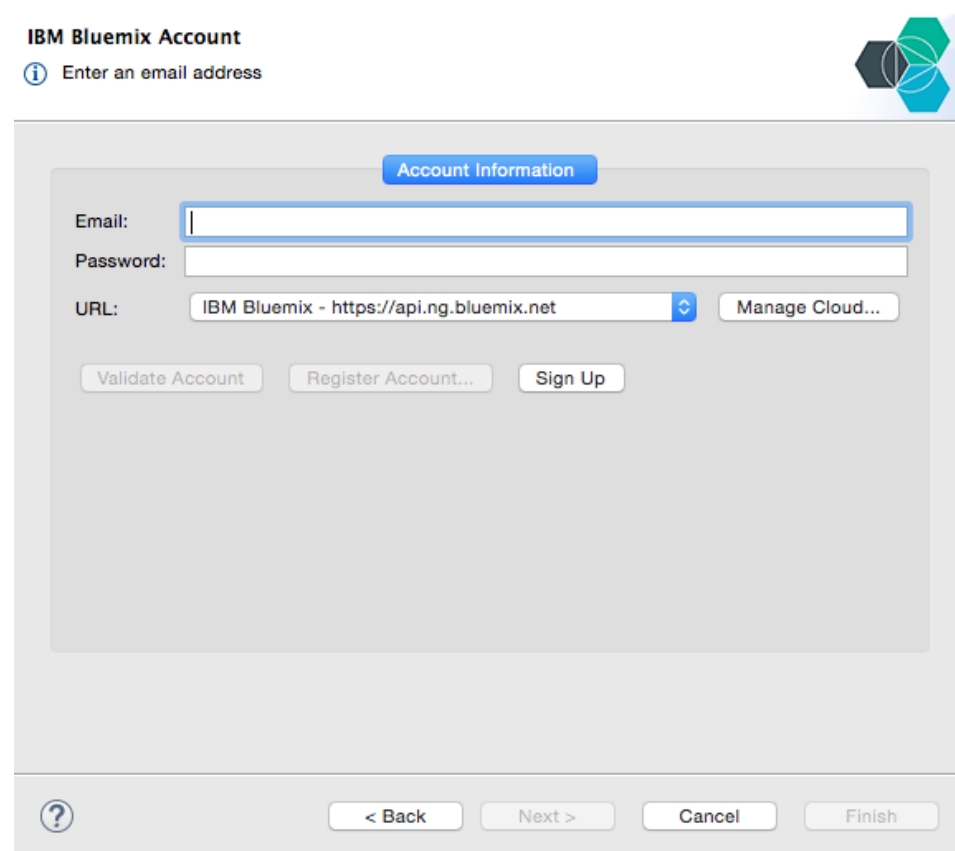


Adding additional Bluemix regions to Eclipse

1. Click **Manage Cloud** from the **Add Bluemix Account** dialog box.
2. Click **Add** to add an additional Bluemix cloud
3. Enter the details of the Bluemix region by providing the name and endpoint URL
4. Click **Finish**.

For more information about region endpoint URLs, see:

https://www.ng.bluemix.net/docs/#overview/overview.html#ov_intro_reg



Summary

- Bluemix provides a number of options for developing applications giving you the ability to integrate Bluemix into the developer tools of your choice
- Bluemix provides starter packs to get you up and running quickly
- Start applications from scratch