

Rapidly Building Apps on the Cloud

 [udemy.com/building-highly-scalable-apps-on-the-cloud/learn/v4/t/lecture/2963462](https://www.udemy.com/building-highly-scalable-apps-on-the-cloud/learn/v4/t/lecture/2963462)

Exercise 6b – Working with Node using Eclipse

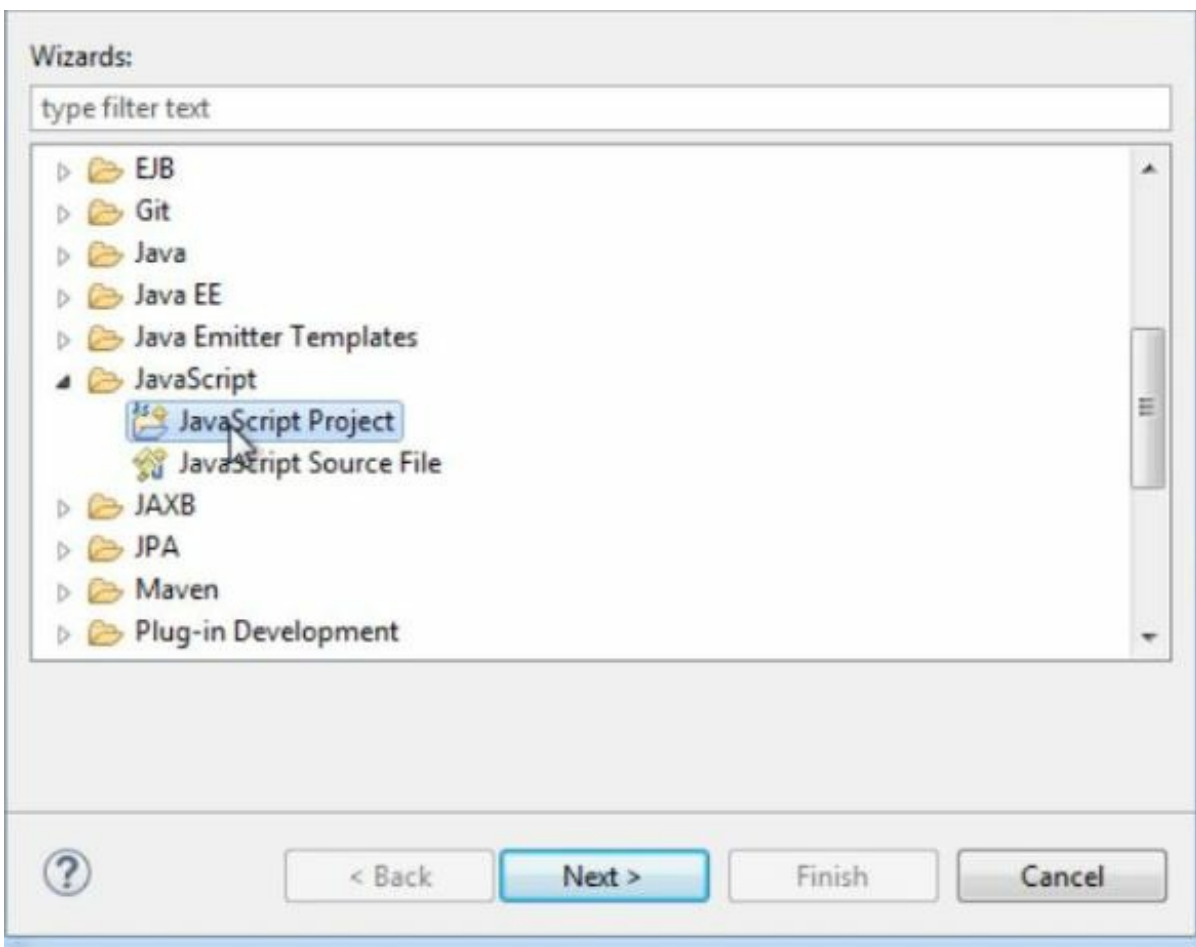
Section 6, Lecture 32

In this exercise you will use Eclipse to create a Node.js application then deploy it to Bluemix.

Launch Eclipse. If using a fresh workspace you need to create the Bluemix server configuration – we did this in section 3.

Create a new javascript project File -> New -> Other

When the Select a wizard dialog is displayed select Javascript -> JavaScript Project.



When the Create a JavaScript project dialog appears. Give the project a name and remove the Web Page Support option.

Create a JavaScript project

Create a JavaScript project in the workspace or in an external location.

Project name:

Contents

☒ Create new project in workspace
☐ Create project from existing source

Directory:

Web Page Support

☐ Include Web Browser Library
☒ Use Window as the default SuperType

Project layout

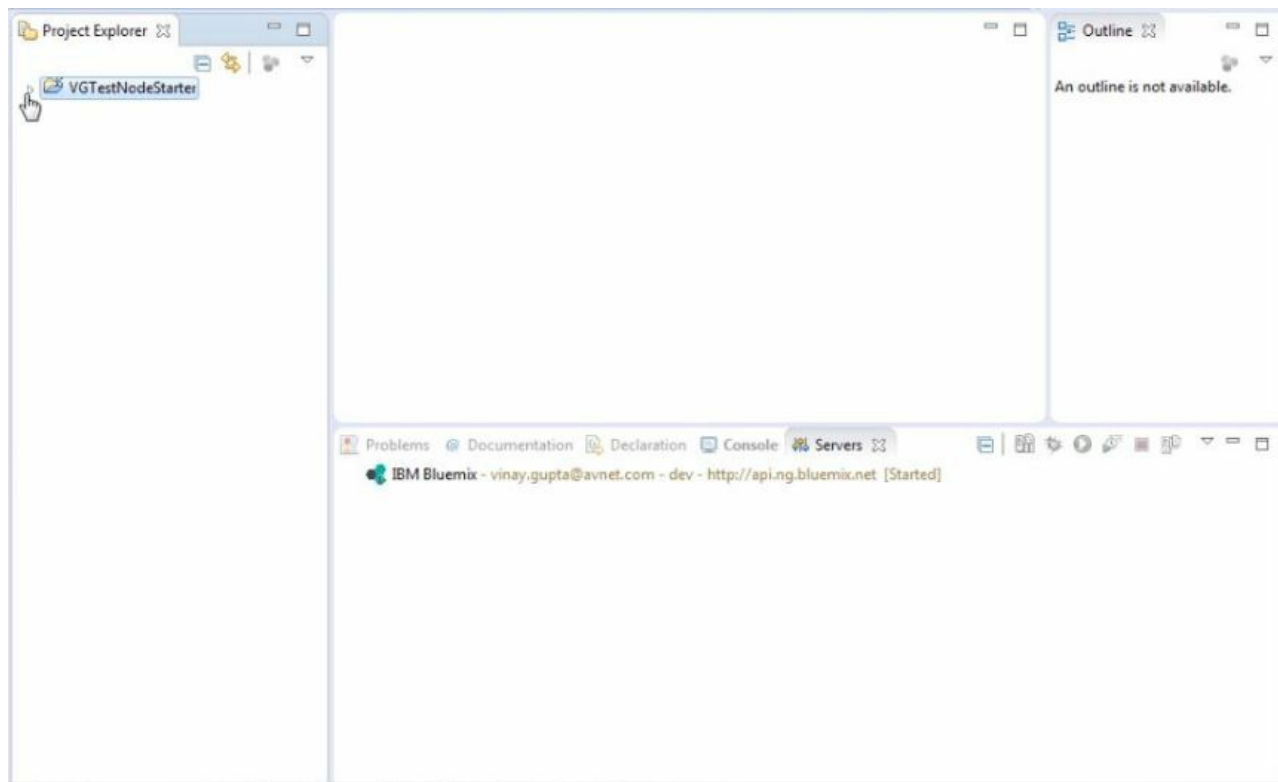
☒ Use project folder as root for the JavaScript global context
☐ Create a separate root folder [Configure default...](#)

Working sets

☐ Add project to working sets

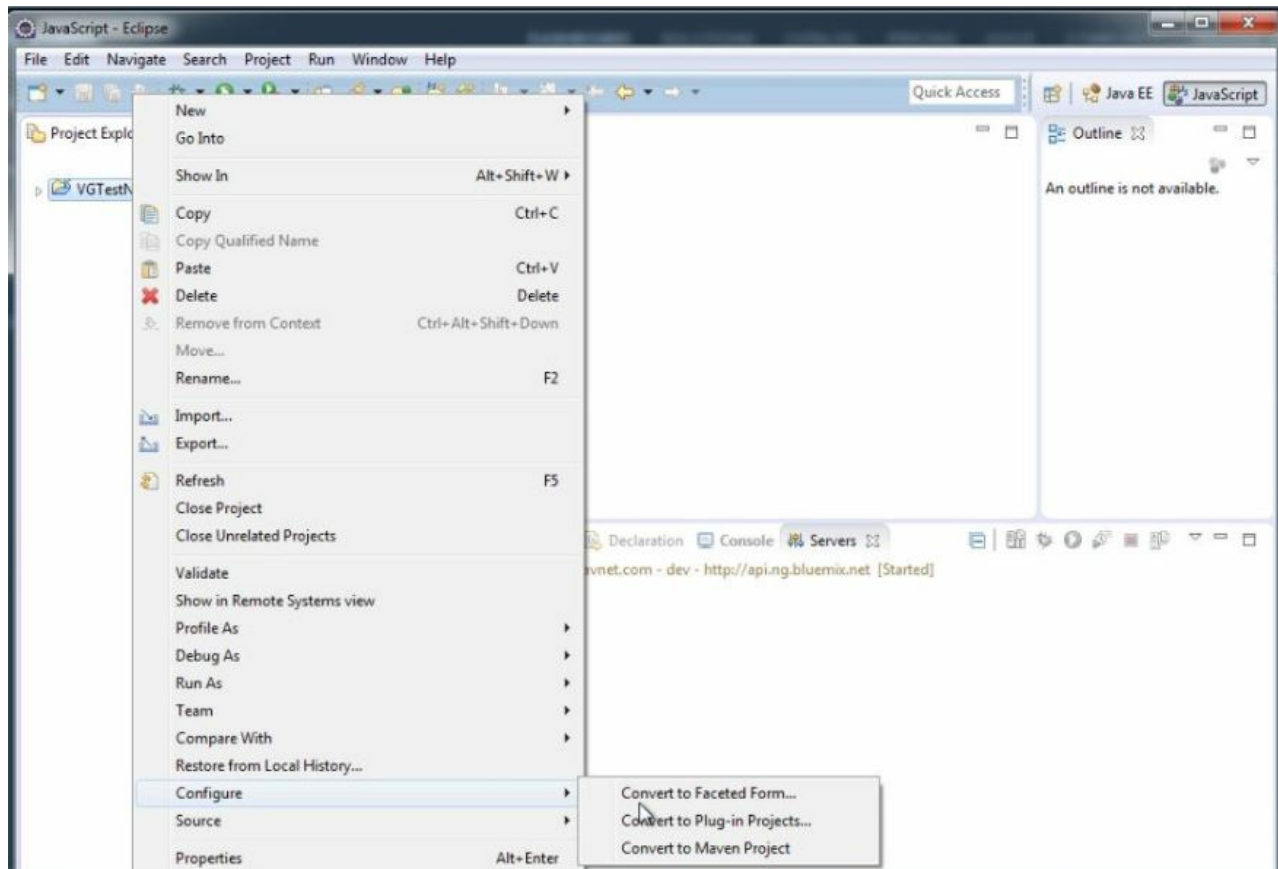
Working sets:

click Finish – will switch to JavaScript perspective – accept this recommendation.



To add Bluemix support to the project you need to add a Facet to the project. Right-click the project name and from 'Configure' select Project Facets – select 'Convert to faceted form...'

Note : The menu items might be different depending on the OS or the version of Eclipse you are using.



then select 'Node.js Application'

Project Facet	Version	
Application Client module	6.0	▼
Axis2 Web Services		
Cloud Foundry Standalone Application	1.0	
CXF 2.x Web Services	1.0	
Dynamic Web Module	3.0	▼
EAR	6.0	▼
EJB Module	3.1	▼
EJBDoclet (XDoclet)	1.2.3	▼
Java	1.7	▼
JavaScript	1.0	
JavaServer Faces	2.2	▼
JAX-RS (REST Web Services)	1.1	▼
JAXB	2.2	▼
JCA Module	1.6	▼
JPA	2.1	▼
Node.js Application	1.0	
Static Web Module		
Utility Module		
Web Fragment Module	3.0	▼
WebDoclet (XDoclet)	1.2.3	▼
WebSphere Application (Co-existence)	8.5	▼
WebSphere Application (Extended)	8.5	▼
WebSphere Application Client (Co-existence)	8.5	▼
WebSphere Application Client (Extended)	8.5	▼
WebSphere EJB (Extended)	8.5	▼
WebSphere Web (Co-existence)	8.5	▼

You will see that app.js and package.json are added to the project and the contents of the file will look similar.

The screenshot shows an IDE with the following components:

- Project Explorer:** Displays a project named 'VGTestNodeStarter' containing a 'JavaScript Resources' folder. Inside this folder are 'app.js', 'http', 'appport', and 'package.json'.
- Editor:** Shows the content of 'app.js', which is a Node.js application script. The code includes comments and logic for setting up a simple HTTP server.
- Outline:** Lists the variables and functions defined in the script, including 'http: any' and 'appport: Number'.

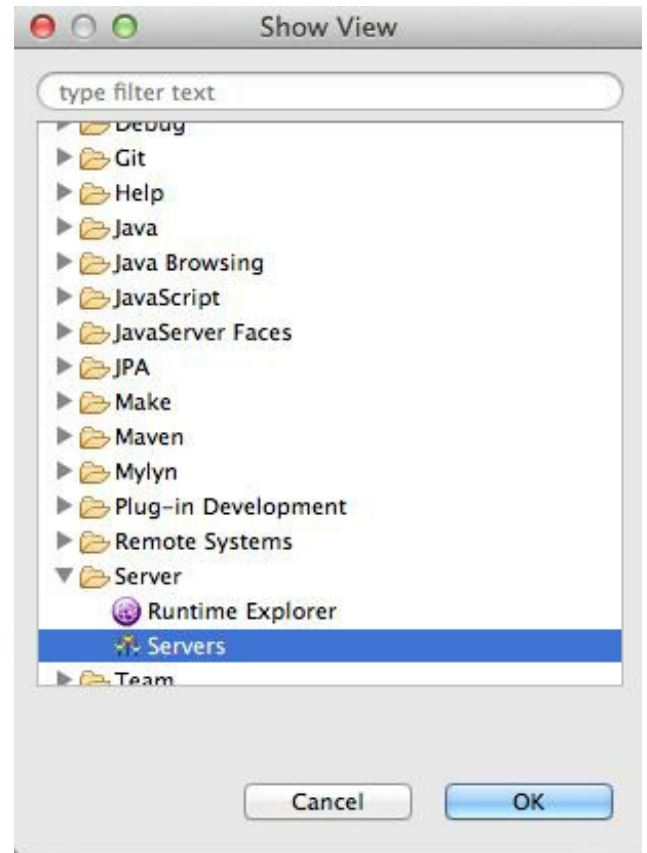
```

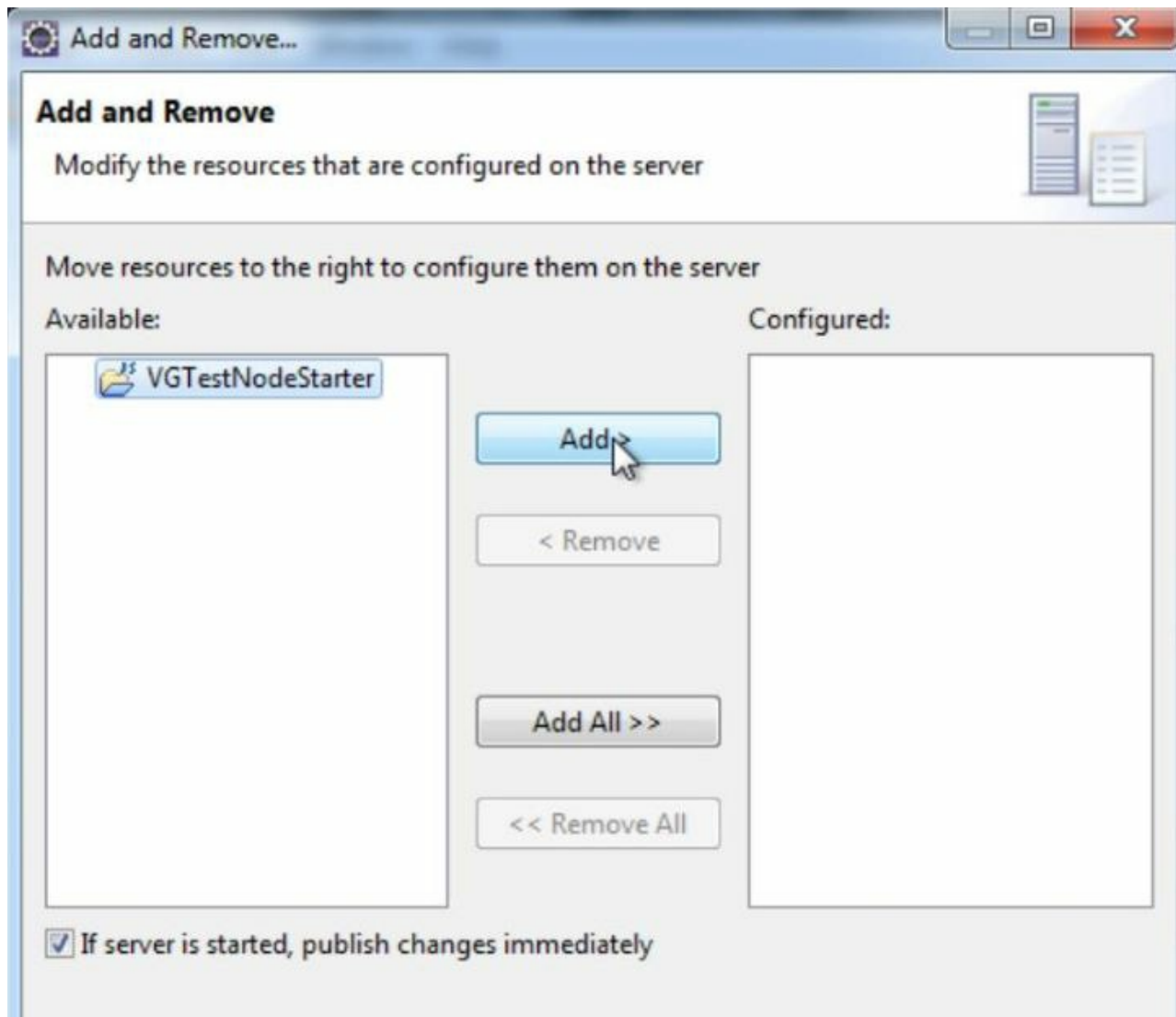
1 |
2 | /**
3 |  * Generated Node.js application that can run on IBM Bluemix
4 |  */
5 |
6 | var http = require("http");
7 |
8 | var appport = process.env.VCAP_APP_PORT || 8888;
9 |
10 | http.createServer(function(request, response) {
11 |
12 |     response.writeHead(200, {"Content-Type": "text/plain"});
13 |     response.write("Generated Node.js application that runs on IBM Bluemix");
14 |     response.end();
15 |
16 | }).listen(appport);
17 |

```

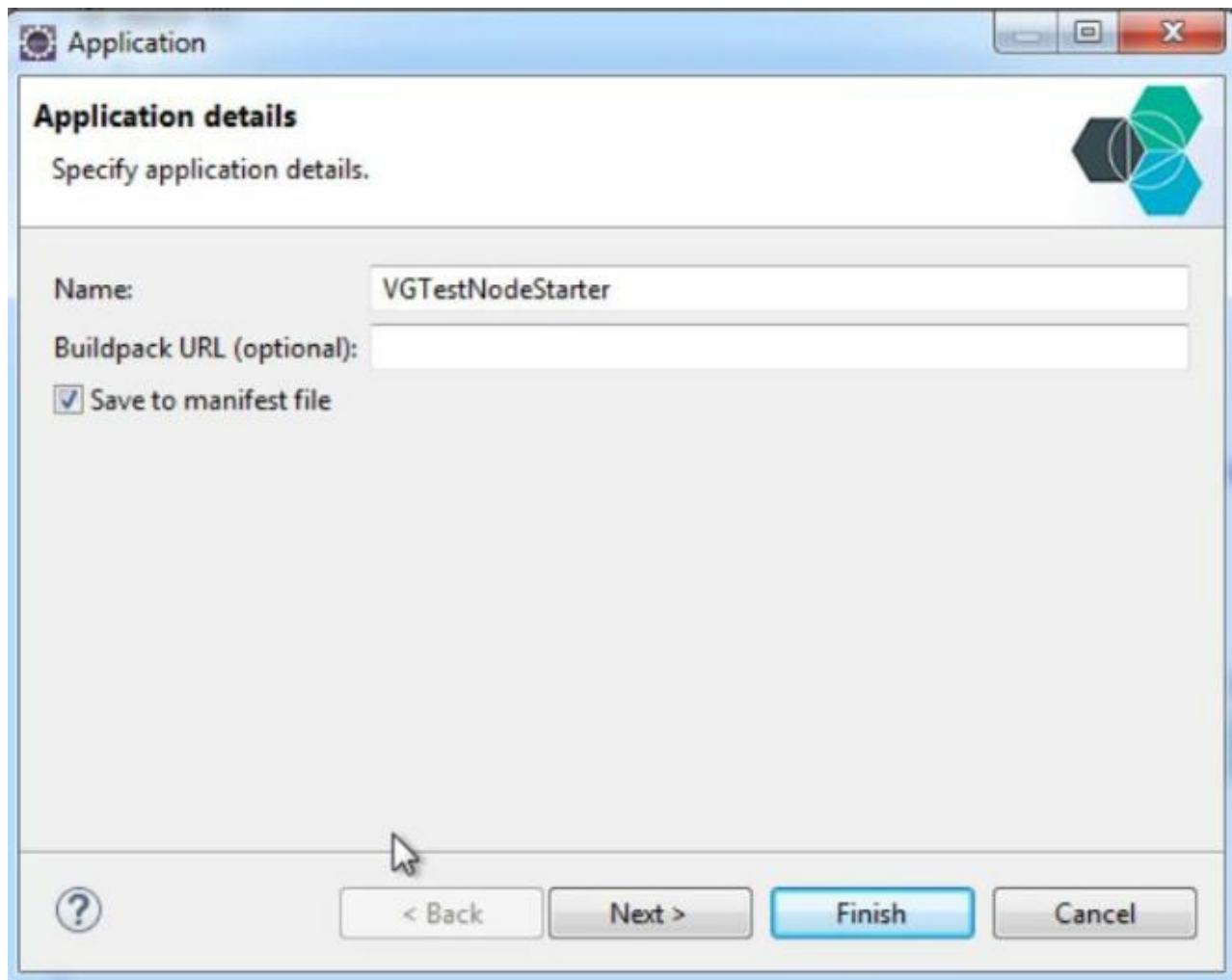
To deploy the application you need to open the Servers view Window -> Show View -> Other. Then in the dialog that opens select Server -> Servers.

Right-click IBM Bluemix (select the server configuration for the space you want to deploy to if more than one server defined) then select 'Add and Remove' and then select your Node application and move it to the Configured column by clicking 'Add'.





The Bluemix deploy wizard will start. Eclipse does not use the manifest file, but you can select to generate a file as a result of the Eclipse deploy:

A screenshot of a software application window titled "Application". The window has a standard Windows-style title bar with minimize, maximize, and close buttons. The main content area is titled "Application details" and contains the instruction "Specify application details.". There is a logo in the top right corner consisting of three overlapping hexagons in dark blue, light blue, and green. Below the instruction, there are two text input fields. The first is labeled "Name:" and contains the text "VGTestNodeStarter". The second is labeled "Buildpack URL (optional):" and is empty. Below these fields is a checkbox labeled "Save to manifest file" which is checked. At the bottom of the window, there is a row of four buttons: a help button with a question mark icon, a "< Back" button, a "Next >" button, and a "Finish" button. The "Finish" button is highlighted with a blue border. A mouse cursor is hovering over the "< Back" button.

Application

Application details

Specify application details.

Name: VGTestNodeStarter

Buildpack URL (optional):

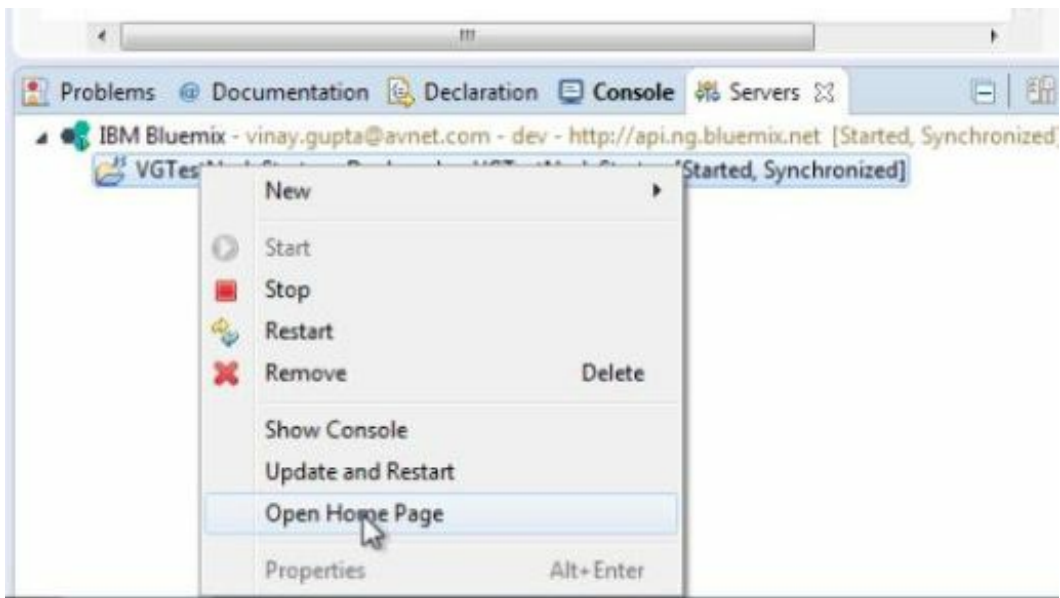
☒ Save to manifest file

? < Back Next > Finish Cancel

Give the application a unique name, modify the memory requirement to 128M. Complete the deployment – there are no services needed for this deployment.

Once the application is running open the generated manifest.yml file and verify the options to selected are in the manifest file.

Once deployed you can run the Node application by right-clicking the application in the Servers view then selecting to open home page.



Once you have tested the application right click the application under the Bluemix server definition and select to remove the application

