Exercise 2. Developing IBM Cloud applications with IBM Cloud Continuous Delivery

Estimated time

01:30

Overview

Development Operations (DevOps) is a software methodology that integrates application development and information technology (IT) operations.

Application development includes writing code, testing the code, building fixes, integrating the fixes, building the application, and deploying the application.

IT operations include managing the environment on which the applications run, providing compute power to the applications, and making the software secure, scale, and run more efficiently.

Because development and operations were traditionally separate groups, “living” in their own isolated worlds several issues arise. DevOps blurs the lines between the development tasks and operational tasks by integrating processes and tools.

DevOps provides real value to the business. For example, it enables continuous delivery, so when new application features are complete, they can be automatically rolled into production. In turn, this action reduces time-to-market, provides competitive advantages, and reduces cost.

DevOps automates the deployment of fixes after they are tested and approved. DevOps enables developers to customize and change applications quickly, improving customer satisfaction.

DevOps enables a more stable environment and better application quality. The combination of a shared code base, continuous integration, test-driven techniques, and automated deployments expose problems in application code, infrastructure, or configuration earlier in software development.

In this exercise, you explore DevOps services in IBM Cloud. The IBM Cloud catalog provides multiple tools for DevOps, but this exercise is focused on IBM Cloud Continuous Delivery. Continuous Delivery enables you to build, test, and deliver applications by using DevOps practices and industry-leading tools.

Objectives

After completing this exercise, you should be able to perform the following tasks:

- Enable your application to use IBM Cloud Continuous Delivery.
- Create a Git repository to manage your source code.
- View and edit code in the Eclipse Orion Web integrated development environment (IDE).

- Build and deploy code to IBM Cloud.
- Test the application in IBM Cloud.

Introduction

IBM Cloud provides an online space to implement your DevOps model. The IBM Cloud Continuous Delivery service provides a set of online development tools. When you enable Continuous Delivery Toolchain for your application, you can choose to have a source code repository that is hosted by IBM and built on GitLab Community Edition.

In the delivery pipeline, you can commit and restore files in your project.

The IBM Cloud Continuous Delivery service also provides a rich IDE for developing, testing, and deploying IBM SDK for Node.js applications by using a web browser. The web IDE is called Eclipse Orion. It provides you with syntax highlighting and API reference features that you expect in a desktop application.

In this exercise, you work with the IBM Cloud Continuous Delivery services to explore, develop, build, and deploy IBM Cloud applications.

Requirements

This exercise requires access to the internet and an IBM Cloud account.

Exercise instructions

In this exercise, you complete the following tasks:

__ 1. Re-create your application.

__ 2. Examine the IBM Cloud application.

__ 3. Enable Continuous Delivery for the starter application.

__ 4. Review your application in Eclipse Orion web IDE.

__ 5. Invite collaborators.

__ 6. Edit the sample application.

__ 7. Commit your changes to the Git repository.

__ 8. Deploy the application from the Git repository to IBM Cloud.

__ 9. Deploy the application directly from your project workspace to IBM Cloud.

__ 10. Automatically push changes to IBM Cloud (optional).

__ 11. Clean up the application and Continuous Delivery service and tools.

Important

If you encounter issues when you follow this exercise, check the Troubleshooting section.

Part 1: Re-creating your application

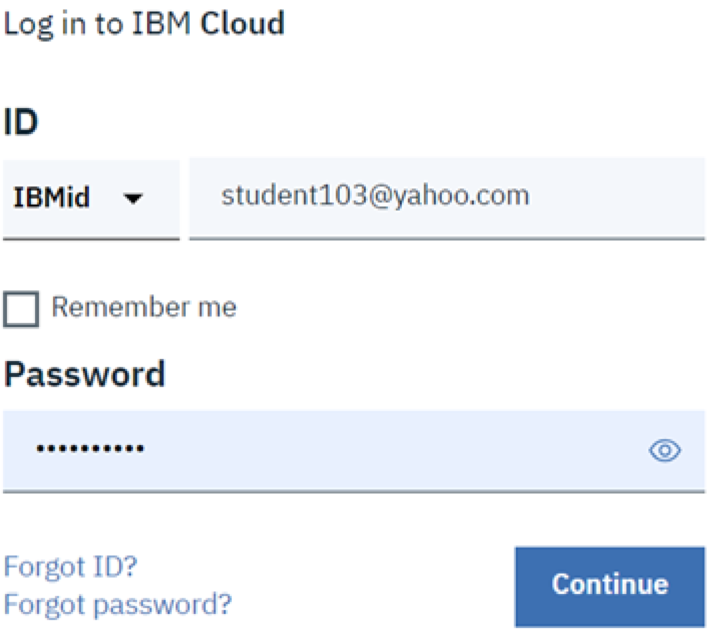
In this part, you re-create the sample Node.js starter application that you deleted at the end of Exercise 1. Complete the following steps:

__ 1. Open the IBM Cloud console in your web browser:

`https://cloud.ibm.com`

__ 2. Log in to your IBM Cloud account:

__ a. Enter your IBM ID and password, as shown in the following figure.



Log in to IBM Cloud

ID

IBMid ▾ student103@yahoo.com

□ Remember me

Password

••••••••••

__ b. Click **Continue**.

__ c. Confirm that your IBM Cloud Dashboard page loads.

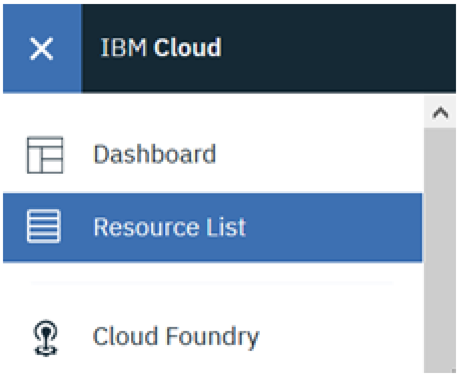
__ 3. Perform the steps in Exercise 1, Part 3, “Creating an application” to create a Node.js application.

Part 2: Examining the IBM Cloud application

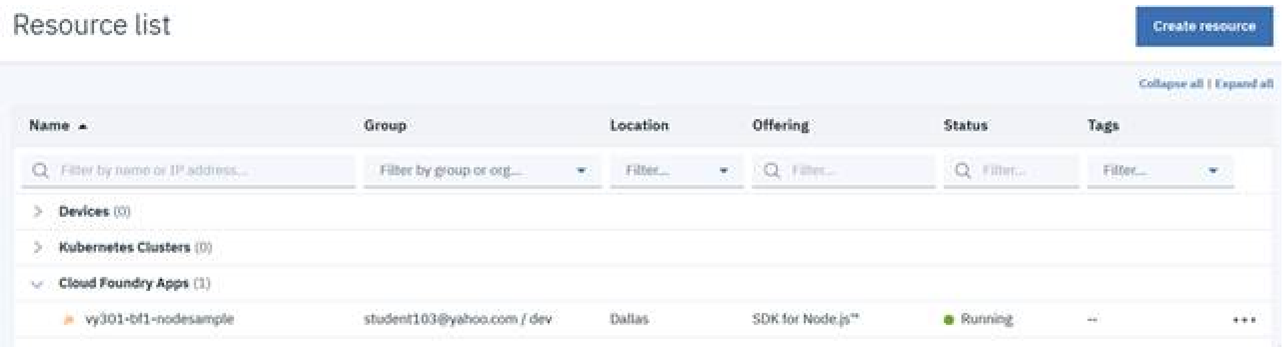
In this part, you explore the application overview page in your IBM Cloud account. The overview page lists the status of your application and the resources that it uses. Complete the following steps:

__ 1. Open the application page for your sample application.

Click the **Navigation Menu** icon at the upper left of the IBM Cloud window and click **Resource List**, as shown in the following picture.

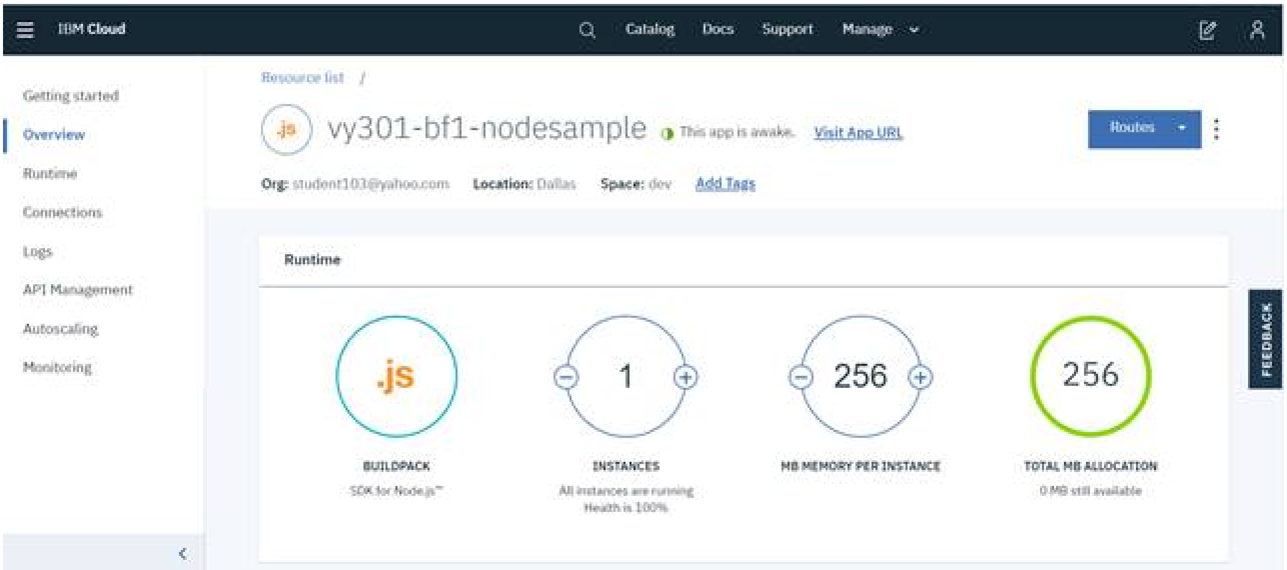


__ 2. Expand the **Cloud Foundry Apps** section and click the application that you created in Part 1 of this exercise, as shown in the following picture.


Resource list
Create resource
Collapse all | Expand all
Filter by name or IP address...
Filter by group or org...
Filter...
Filter...
Filter...
Filter...

Name	Group	Location	Offering	Status	Tags
Devices (0)					
Kubernetes Clusters (0)					
Cloud Foundry Apps (1)					
vy301-bf1-nodesample	student103@yahoo.com / dev	Dallas	SDK for Node.js™	● Running	--

__ a. In the Application Details page, click **Overview** from the left navigation bar, as shown in the following figure.



vy301-bf1-nodesample ◑ This app is awake.

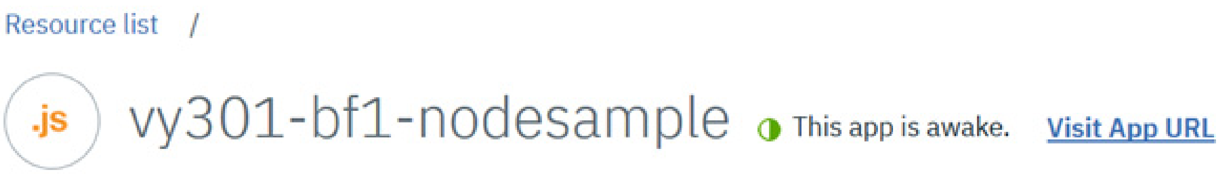
Org: student103@yahoo.com Location: Dallas Space: dev

Note

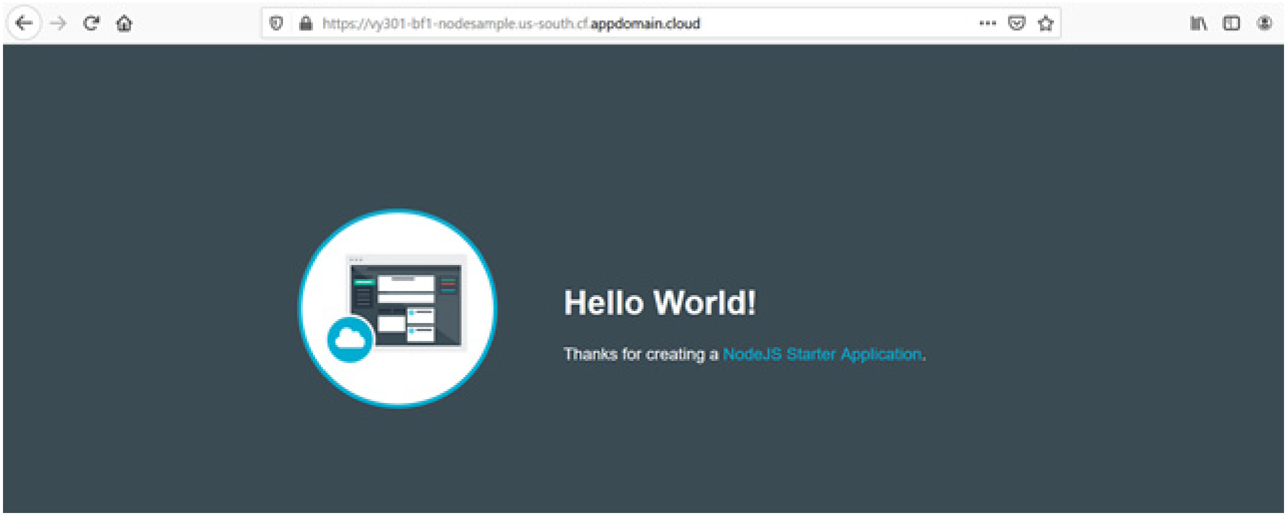
The state of the application might differ according to your account type. It can be in the “This app is awake” state, as shown in the figure, for Lite accounts or “Running” for non-Lite accounts.

__ 3. Test the sample application:

__ a. Click **Visit App URL** for your application, as shown in the following figure.



__ b. A new tab opens in the browser that shows your app. Confirm that the sample application appears, as shown in the following picture.


https://vy301-bf1-nodesample.us-south.cf.appdomain.cloud

Hello World!

Thanks for creating a NodeJS Starter Application.

__ c. Close the Node.js starter application web page.

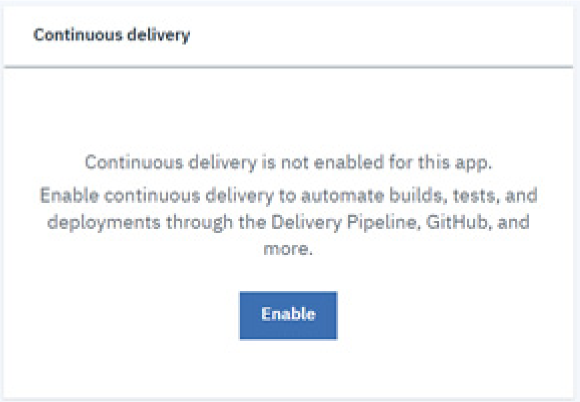
Part 3: *Enabling Continuous Delivery for the starter application*

In this part, you enable continuous delivery for your application. The Continuous Delivery service supports your DevOps workflows:

- You can create integrated DevOps open toolchains to enable tool integrations that support your development, deployment, and operations tasks.
- Deliver continuously by using automated pipelines.
- Edit and push your code from anywhere by using the web-based IDE.
- Collaborate with your team and manage your source code with a Git repository (repos) and issue tracker that is hosted on the cloud.

To enable continuous delivery for your application, perform the following steps

__ 1. In the overview page for the `vy301-xxx-nodesample` application, scroll down to Continuous delivery and click **Enable**, as shown in the following figure.

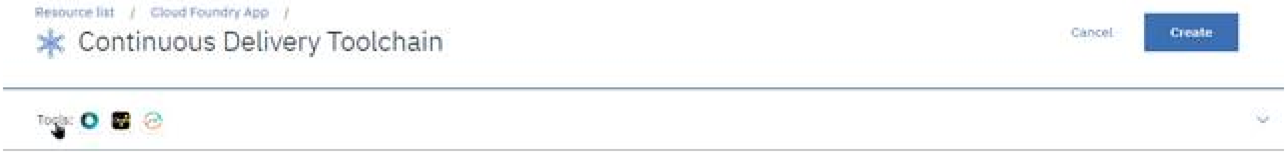


Continuous delivery

Continuous delivery is not enabled for this app.

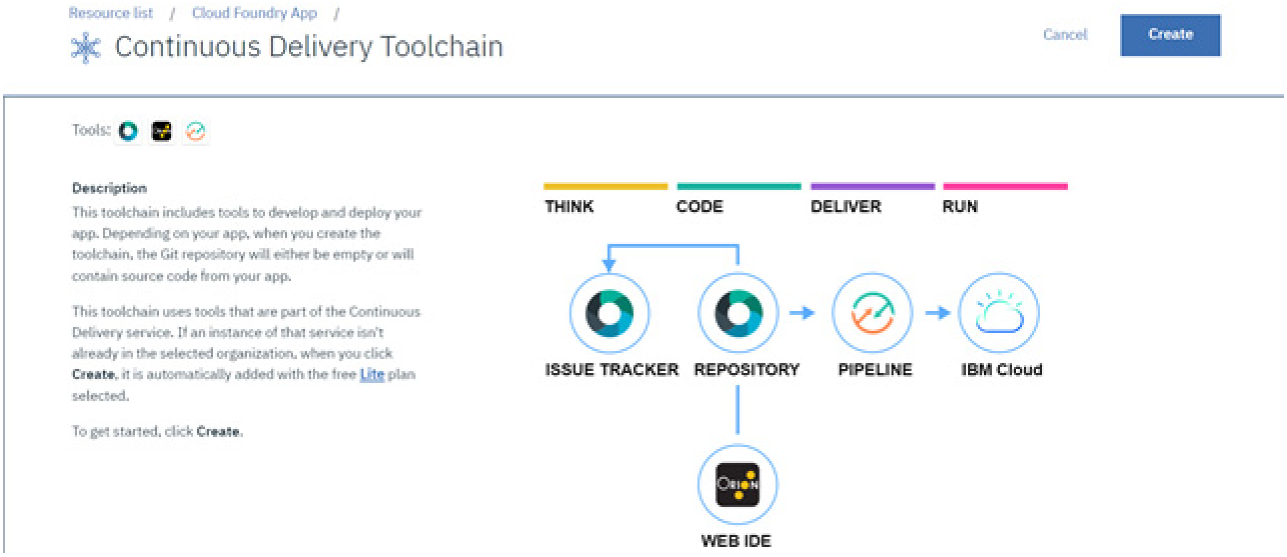
Enable continuous delivery to automate builds, tests, and deployments through the Delivery Pipeline, GitHub, and more.

__ 2. Click **Tools** to expand the tools bar.



Resource list / Cloud Foundry App /
Continuous Delivery Toolchain
Cancel
Create
Tools:

__ 3. The bar expands and displays the tools that are created when you create the toolchain as shown in the following figure.



Description

This toolchain includes tools to develop and deploy your app. Depending on your app, when you create the toolchain, the Git repository will either be empty or will contain source code from your app.

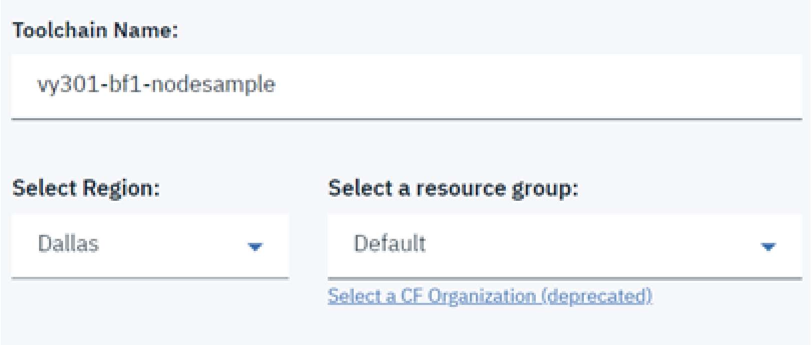
This toolchain uses tools that are part of the Continuous Delivery service. If an instance of that service isn't already in the selected organization, when you click **Create**, it is automatically added with the free Lite plan selected.

To get started, click **Create**.

- Issue Tracker: Use this tool to track problems in your application,
- Repository: Git repo that you can use to control versions of your application and collaborate with your team to manage your source code
- Web IDE (Eclipse Orion): Use this tool to edit and change your code,
- Pipeline: Use this tool to automate builds, unit tests, deployments, and more. The stages in the pipeline can automatically build when you push changes to a linked repository and then deploy to one or more environments on IBM Cloud..

__ 4. Close the bar by clicking **Tools** again.

__ 5. On the same page, check the Toolchain Name, the Select Region, and the Select a resource group fields, which are already populated, as shown in the following figure. Keep the default values and scroll down to the Tool Integrations section.



Toolchain Name:

vy301-bf1-nodesample

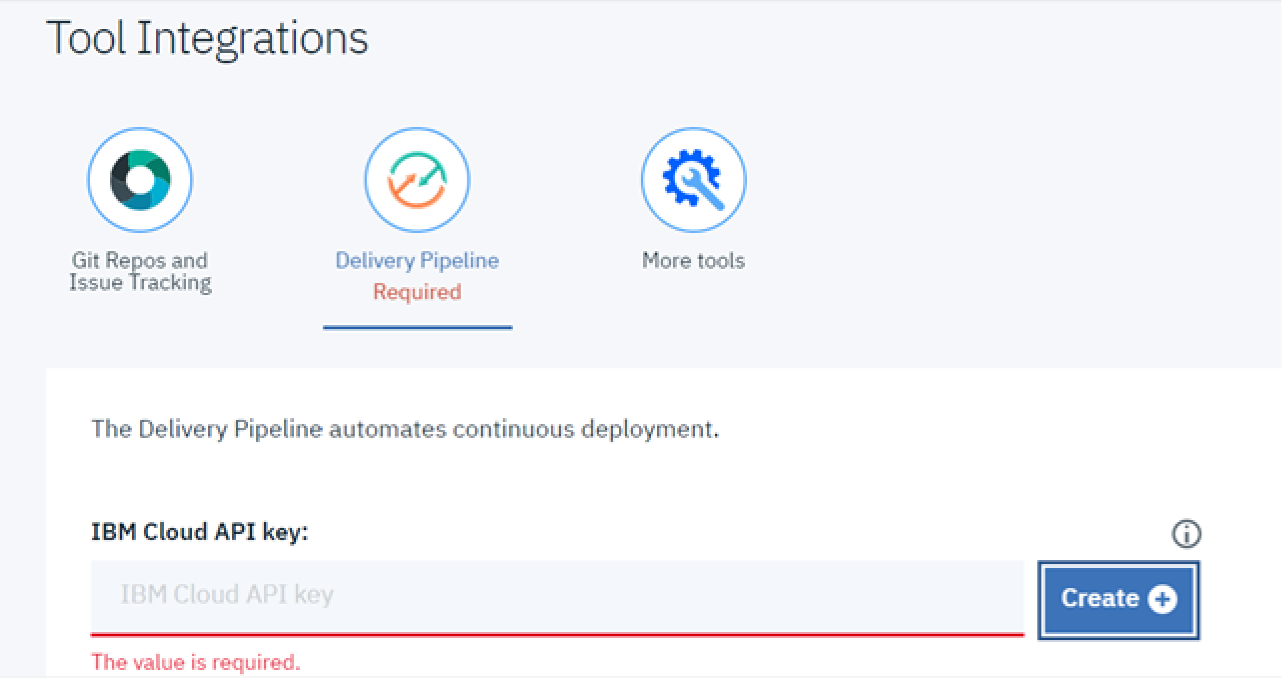
Select Region:

Dallas

Select a resource group:

Default

__ 6. In the Tool Integrations section, click **Delivery Pipeline** to create an API key that enables communication between the different Toolchain components and IBM Cloud. Click **Create+** to create the API key, as shown in the following figure.



Tool Integrations

The Delivery Pipeline automates continuous deployment.

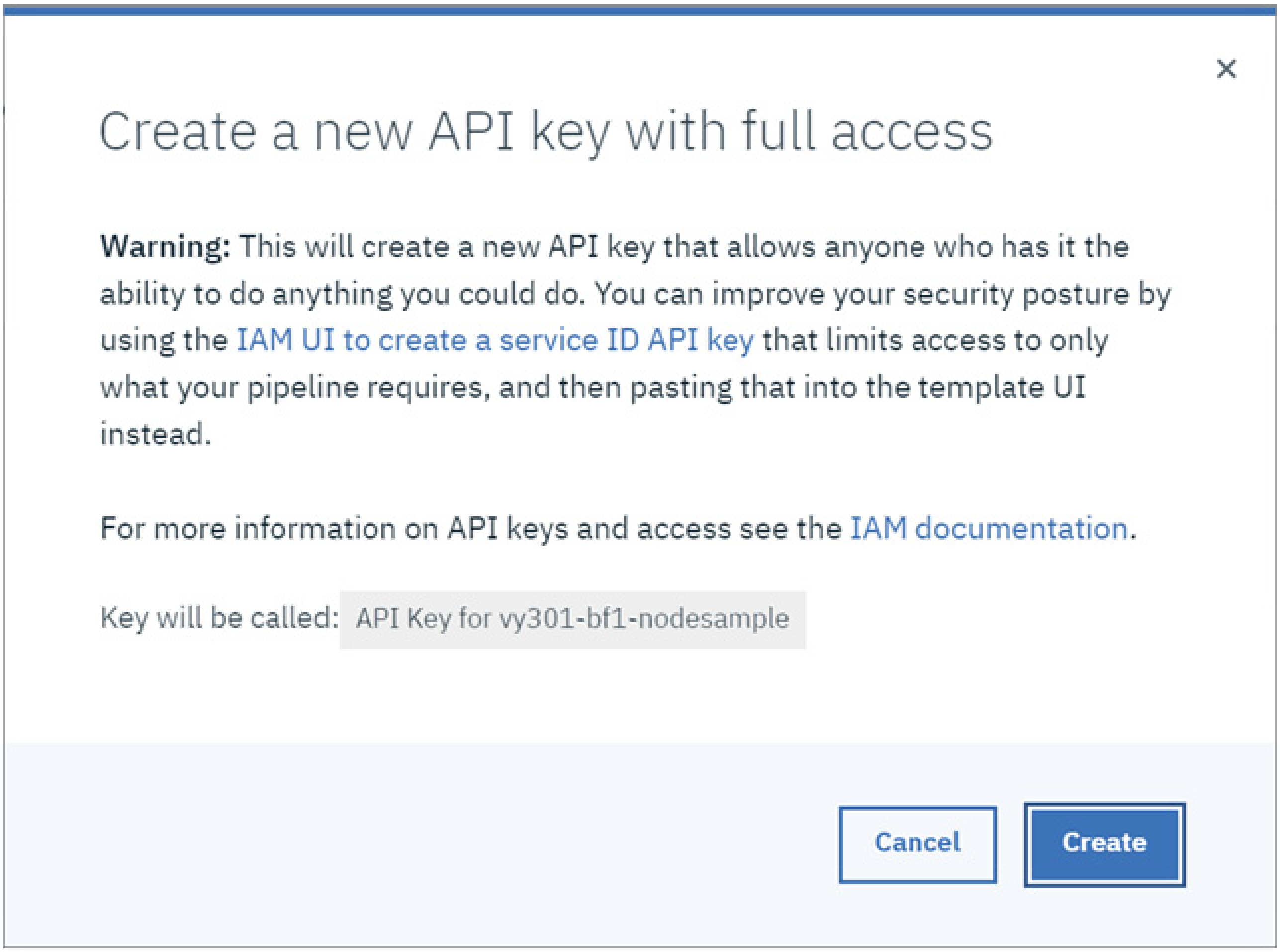
IBM Cloud API key:

The value is required.

API keys: An API key is a unique code that is passed in to an API to identify the calling application or user. API keys are used to track and control how the API is being used, for example to prevent malicious use or abuse of the API. The API key often acts as both a unique identifier and a secret token for authentication, and generally has a set of access that is specific to the identity associated with it.

In this case, you create a new API key for the logged-in user. This allows the Delivery Pipeline to authenticate with this user's credentials to IBM Cloud so that it can deploy the application. If you have an existing IBM Cloud API key, you can enter it in the IBM Cloud API Key field.

__ 7. Confirm the creation of the API key by clicking **Create**, as shown in the following figure.



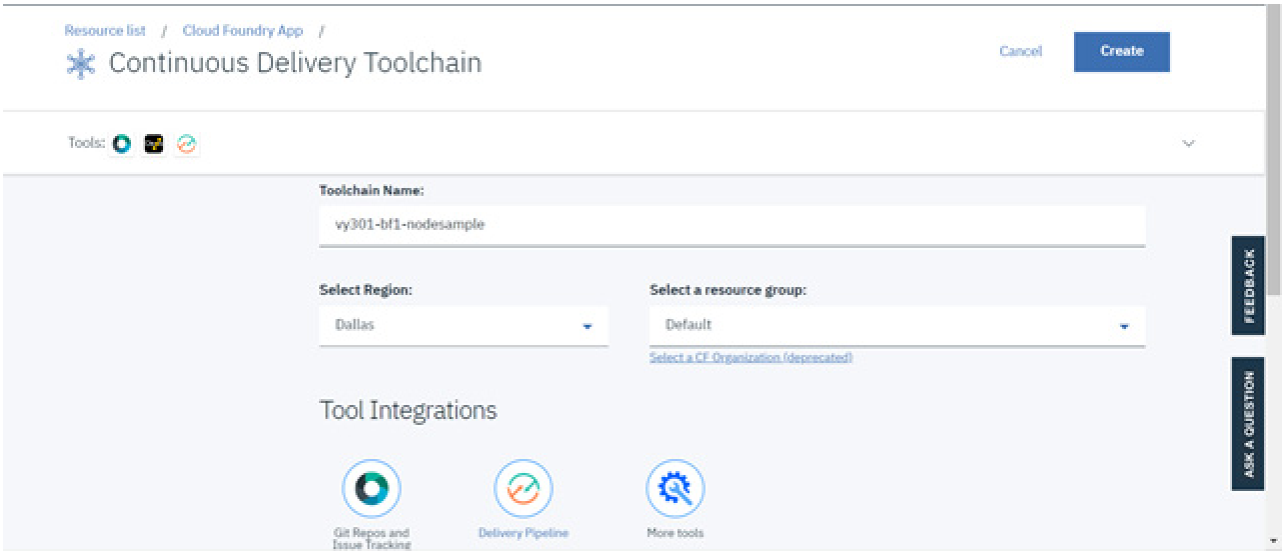
Create a new API key with full access

Warning: This will create a new API key that allows anyone who has it the ability to do anything you could do. You can improve your security posture by using the IAM UI to create a service ID API key that limits access to only what your pipeline requires, and then pasting that into the template UI instead.

For more information on API keys and access see the IAM documentation.

Key will be called: API Key for vy301-bf1-nodesample

__ 8. Finally, click **Create** in the top bar to enable Continuous Delivery Toolchain for the starter application, as shown in the following figure.


Resource list / Cloud Foundry App /
Continuous Delivery Toolchain
Cancel
Create
Tools:
Tool Integrations
Git Repos and Issue Tracking
Delivery Pipeline
More tools
FEEDBACK
ASK A QUESTION

Toolchain Name: vy301-bf1-nodesample

Select Region: Dallas

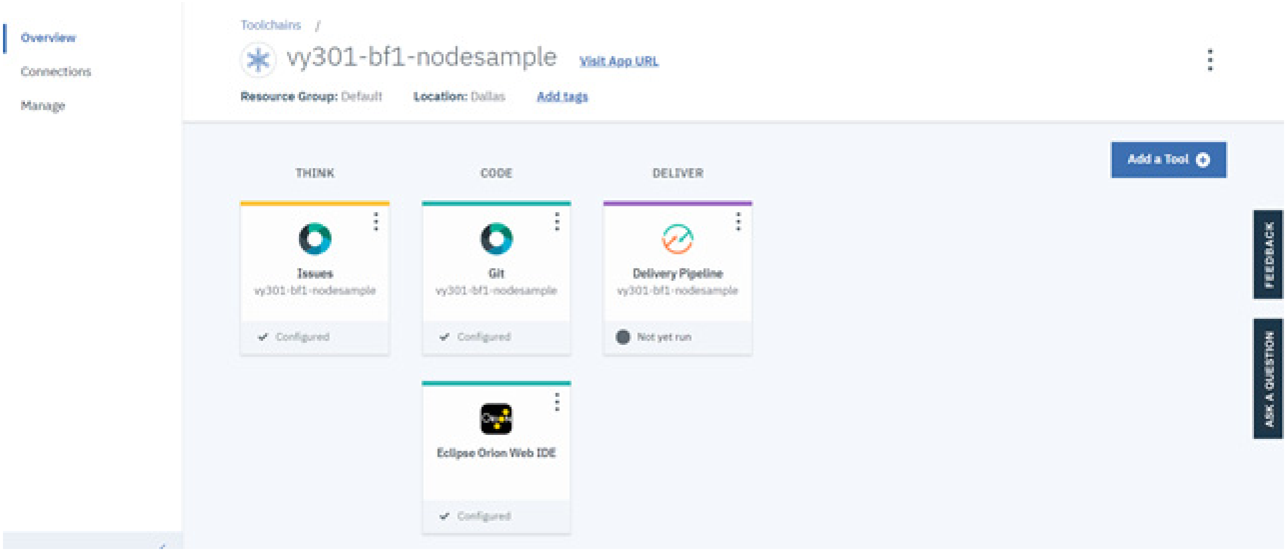
Select a resource group: Default

Select a CF Organization (deprecated)

Stop

Wait until the wizard creates the Git repository for your application.

__ 9. You are redirected to the Toolchains dashboard, as shown in the following figure.



vy301-bf1-nodesample

Resource Group: Default Location: Dallas

Issues vy301-bf1-nodesample ✓ Configured

Git vy301-bf1-nodesample ✓ Configured

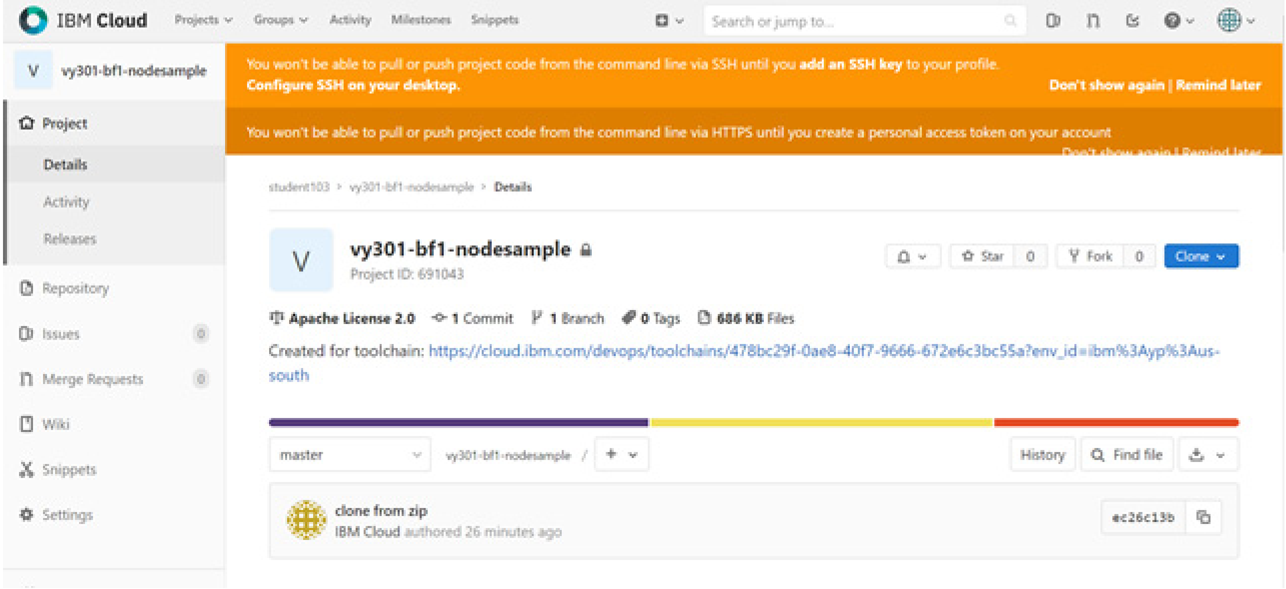
Delivery Pipeline vy301-bf1-nodesample ● Not yet run

Eclipse Orion Web IDE ✓ Configured

Note

By enabling Continuous Delivery Toolchains, you perform a Git clone for the IBM Cloud starter code by default.

__ 10. Confirm that the Git repository was created by right-clicking the **Git** box and then selecting **Open link in new tab**. The GitLab web-based Git repository manager is displayed, as shown in the following figure.



You won't be able to pull or push project code from the command line via SSH until you **add an SSH key** to your profile. **Configure SSH on your desktop.**

You won't be able to pull or push project code from the command line via HTTPS until you create a personal access token on your account

vy301-bf1-nodesample

Project ID: 691043

Apache License 2.0 1 Commit 1 Branch 0 Tags 686 KB Files

Created for toolchain: https://cloud.ibm.com/devops/toolchains/478bc29f-0ae8-40f7-9666-672e6c3bc55a?env_id=ibm%3Ayp%3Aus-south

clone from zip

IBM Cloud authored 26 minutes ago

ec26c13b

__ 11. Close this browser tab.

Information

IBM Cloud Continuous Delivery creates a Git repository as a change management system. You can use any Git client to work with the artifacts that are stored in the repository.

Part 4: Reviewing your application in Eclipse Orion web IDE

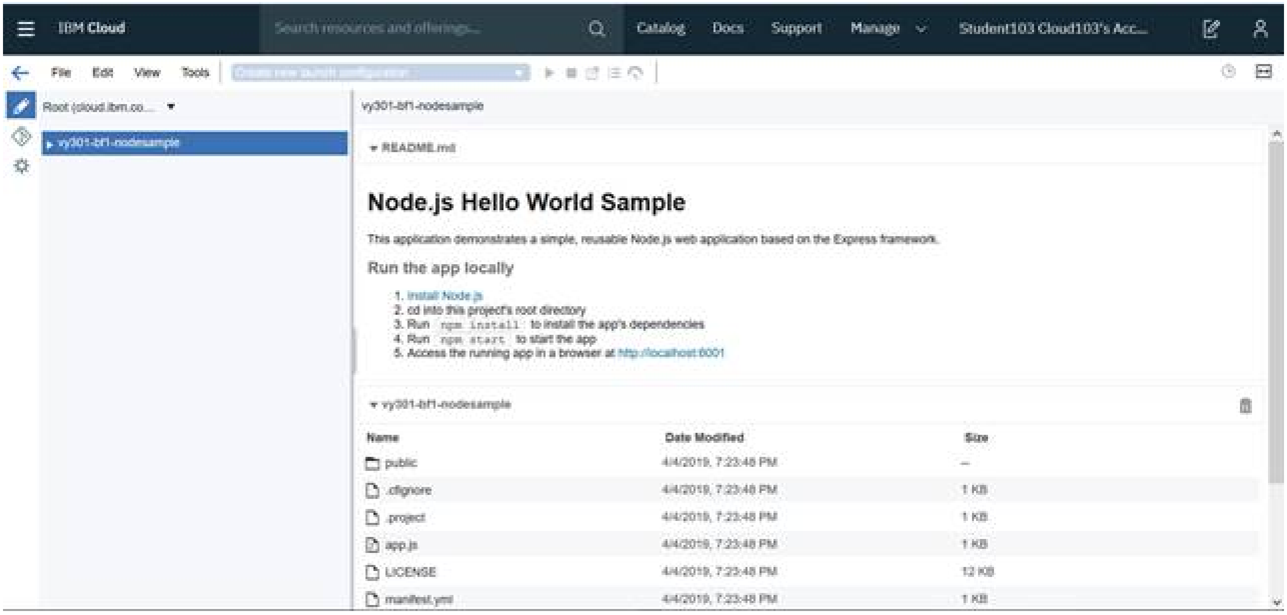
In this part, you review your application project in Eclipse Orion Web IDE, which is a browser-based development environment. You invite other users to your project so the team can collaborate in code development and deployment.

Complete the following steps:

__ 1. Click **Eclipse Orion Web IDE** to open the code editor.

__ 2. Examine the workspace:

__ a. By default, the project's overview page is opened, and you can see the project directory that is collapsed on the left, as shown in the following figure.



Node.js Hello World Sample

This application demonstrates a simple, reusable Node.js web application based on the Express framework.

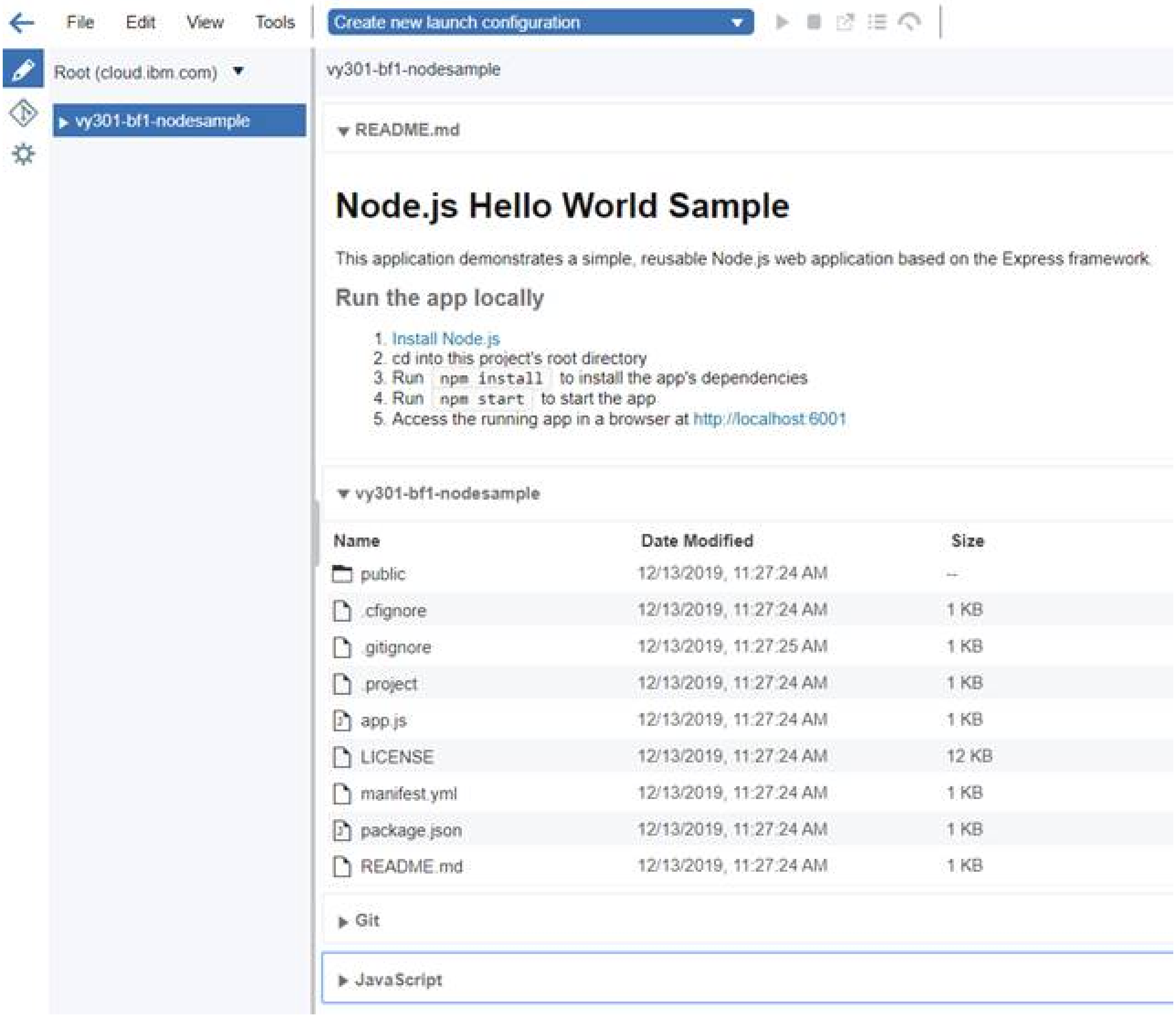
Run the app locally

1. Install Node.js
2. cd into this project's root directory
3. Run `npm install` to install the app's dependencies
4. Run `npm start` to start the app
5. Access the running app in a browser at http://localhost:6001

vy301-bf1-nodesample

Name	Date Modified	Size
public	4/4/2019, 7:23:48 PM	–
.cfignore	4/4/2019, 7:23:48 PM	1 KB
.project	4/4/2019, 7:23:48 PM	1 KB
app.js	4/4/2019, 7:23:48 PM	1 KB
LICENSE	4/4/2019, 7:23:48 PM	12 KB
manifest.yml	4/4/2019, 7:23:48 PM	1 KB

__ b. Review the application components that are shown in the following figure by scrolling down the overview page.



Node.js Hello World Sample

This application demonstrates a simple, reusable Node.js web application based on the Express framework.

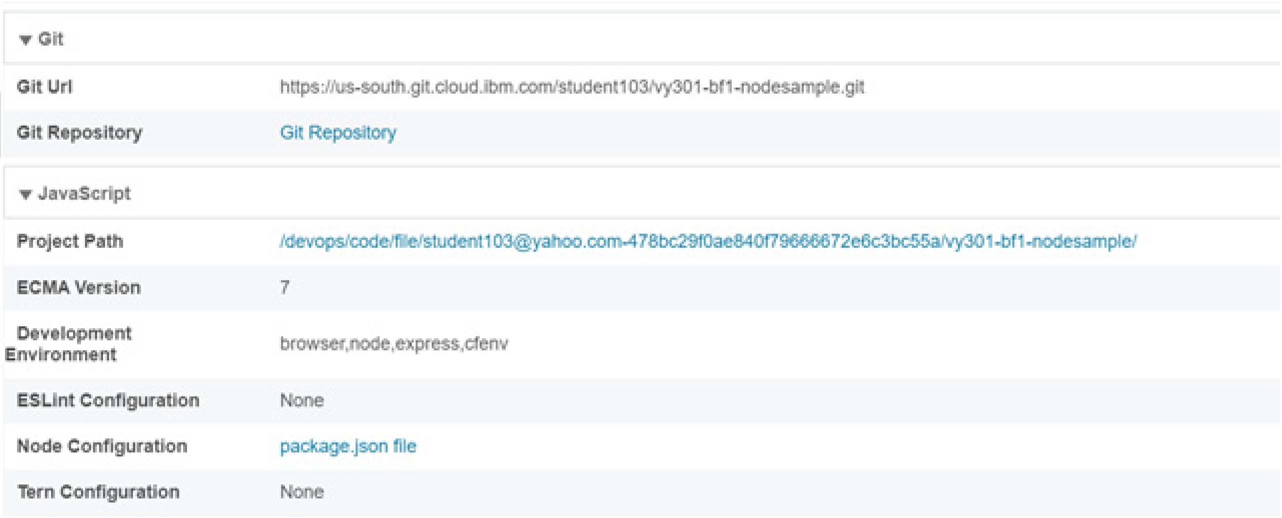
Run the app locally

1. Install Node.js
2. cd into this project's root directory
3. Run `npm install` to install the app's dependencies
4. Run `npm start` to start the app
5. Access the running app in a browser at http://localhost:6001

vy301-bf1-nodesample

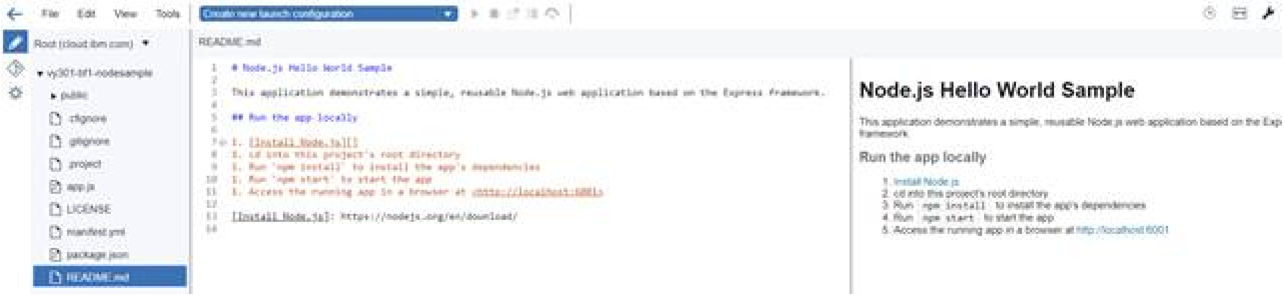
Name	Date Modified	Size
public	12/13/2019, 11:27:24 AM	--
.cfignore	12/13/2019, 11:27:24 AM	1 KB
.gitignore	12/13/2019, 11:27:25 AM	1 KB
.project	12/13/2019, 11:27:24 AM	1 KB
app.js	12/13/2019, 11:27:24 AM	1 KB
LICENSE	12/13/2019, 11:27:24 AM	12 KB
manifest.yml	12/13/2019, 11:27:24 AM	1 KB
package.json	12/13/2019, 11:27:24 AM	1 KB
README.md	12/13/2019, 11:27:24 AM	1 KB

The Node.js application (`vy301-xxx-nodesample`) section includes the `app.js` file that contains the code, a license file, an archive of the deployment in IBM Cloud that is called `manifest.yml`, an archive of libraries that are required for Node.js application that is called `package.json`, and a readme file that is called `README.md`. The last two sections show the Git repository location and some details about JavaScript, including the libraries that are used, as shown in the following figure.



▼ Git	
Git Url	https://us-south.git.cloud.ibm.com/student103/vy301-bf1-nodesample.git
Git Repository	Git Repository
▼ JavaScript	
Project Path	/devops/code/file/student103@yahoo.com-478bc29f0ae840f79666672e6c3bc55a/vy301-bf1-nodesample/
ECMA Version	7
Development Environment	browser,node,express,cfenv
ESLint Configuration	None
Node Configuration	package.json file
Tern Configuration	None

__ 3. Expand the application files in the left pane and open the README.md file by clicking it, as shown in the following figure.



README.md

```
# Node.js Hello World Sample

This application demonstrates a simple, reusable Node.js web application based on the Express framework.

## Run the app locally

1. [Install Node.js][]
1. cd into this project's root directory
1. Run `npm install` to install the app's dependencies
1. Run `npm start` to start the app
1. Access the running app in a browser at <http://localhost:6001>

[Install Node.js]: https://nodejs.org/en/download/
```

Node.js Hello World Sample

This application demonstrates a simple, reusable Node.js web application based on the Exp framework.

Run the app locally

1. Install Node.js
2. cd into this project's root directory
3. Run `npm install` to install the app's dependencies
4. Run `npm start` to start the app
5. Access the running app in a browser at http://localhost:6001

The README.md file provides a quick summary of the repository that manages your application artifacts and source code. It is a best practice to provide an up-to-date description of your project in this document, especially for projects that are publicly shared for all users.

Information

The README.md file is written with the Markdown syntax, which is a lightweight markup language for annotating plain text documents to be displayed as Hypertext Markup Language (HTML) documents. For more information about Markdown, see:

http://daringfireball.net/projects/markdown/

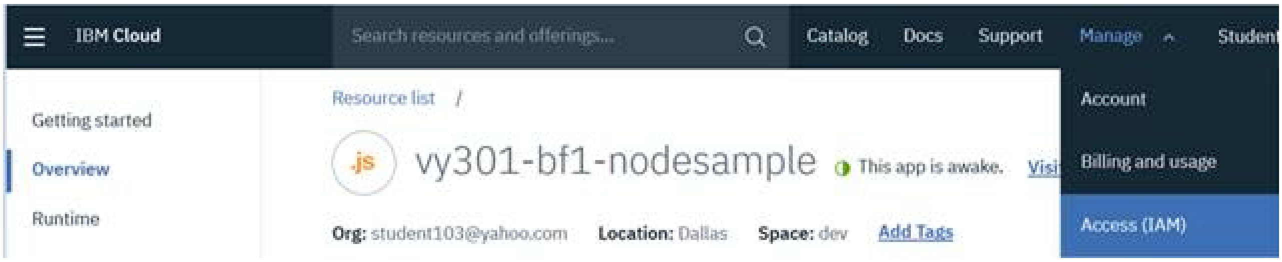
For more information about guidelines for building a good readme file, see:

https://github.com/IBM-Cloud/repo-guidelines/blob/master/README.md#readmemd-structure

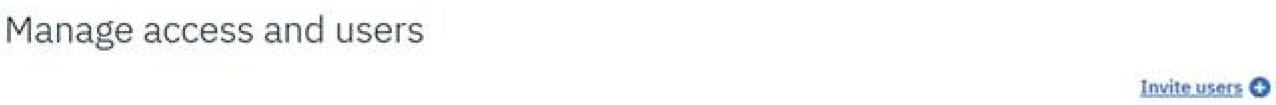
Part 5: Inviting collaborators

Invite project members to the toolchain that you created with the Delivery Pipeline and also to the Cloud Foundry organization that includes your deployed application. The main goal here is to enable other users to change the code and deploy your application in a developing environment with multiple participants.

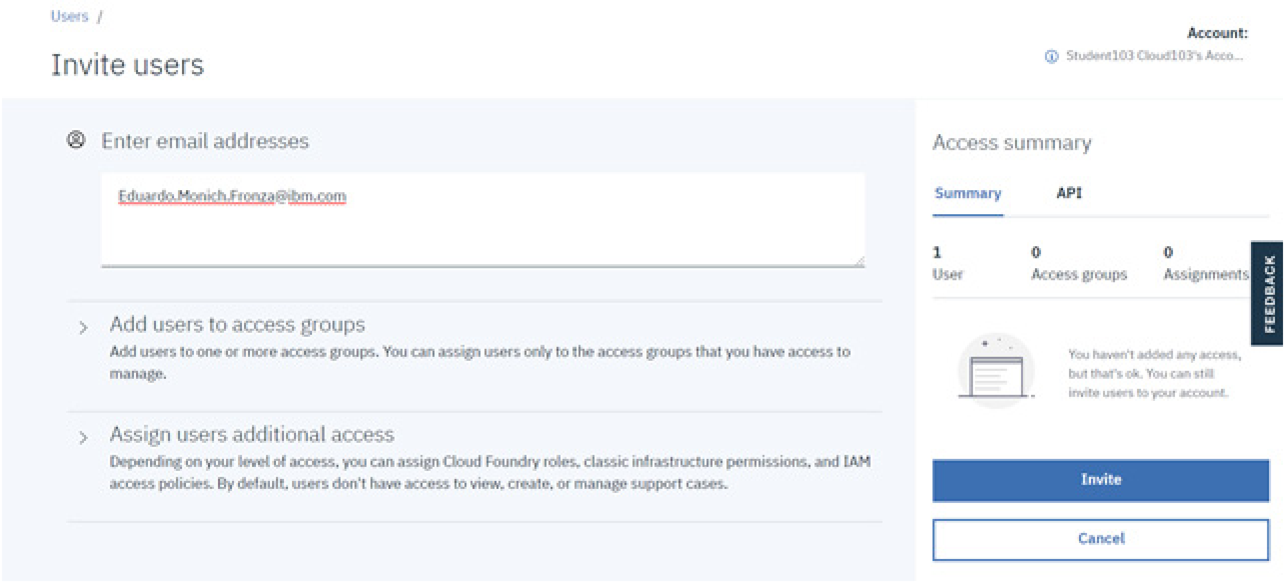
__ 1. Go to the main menu in IBM Cloud and select **Manage** -> **Access (IAM)**, as shown in the following figure.



__ 2. Click **Invite users**, as shown in the following figure.



__ 3. In this exercise, you give users access to a service and more specifically, to a resource, which is the Toolchain resource, and the Cloud Foundry application (Node.js runtime). Enter the **email of the user** to which you want to grant access and scroll down to **Assign users additional access** pane, as shown in the following figure.



Eduardo.Monich.Fronza@ibm.com

Add users to access groups

Add users to one or more access groups. You can assign users only to the access groups that you have access to manage.

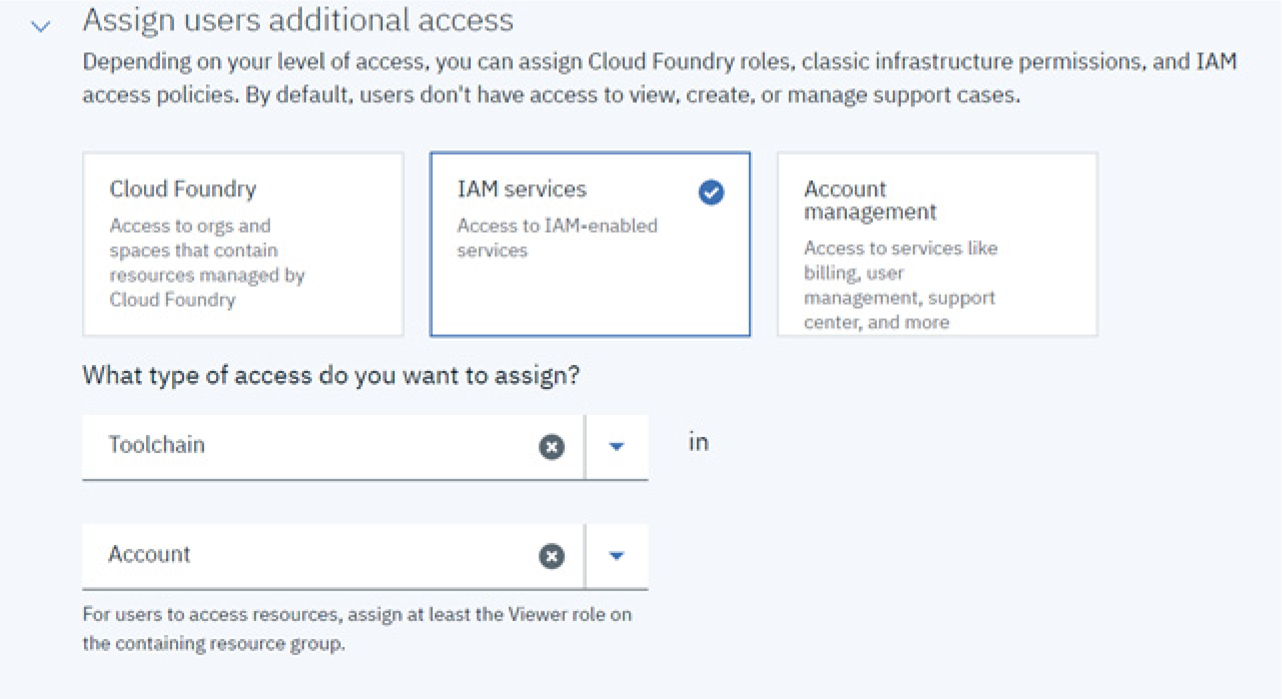
Assign users additional access

Depending on your level of access, you can assign Cloud Foundry roles, classic infrastructure permissions, and IAM access policies. By default, users don't have access to view, create, or manage support cases.

You haven't added any access, but that's ok. You can still invite users to your account.

__ 4. Adding access to the Toolchain instances.

__ a. In the “Assign users additional access” section, select **IAM services**. Under “What type of access do you want to assign?” select **Toolchain** in **Account**, as shown in the following figure.



Assign users additional access

Depending on your level of access, you can assign Cloud Foundry roles, classic infrastructure permissions, and IAM access policies. By default, users don't have access to view, create, or manage support cases.

Cloud Foundry
Access to orgs and spaces that contain resources managed by Cloud Foundry

IAM services ✓
Access to IAM-enabled services

Account management
Access to services like billing, user management, support center, and more

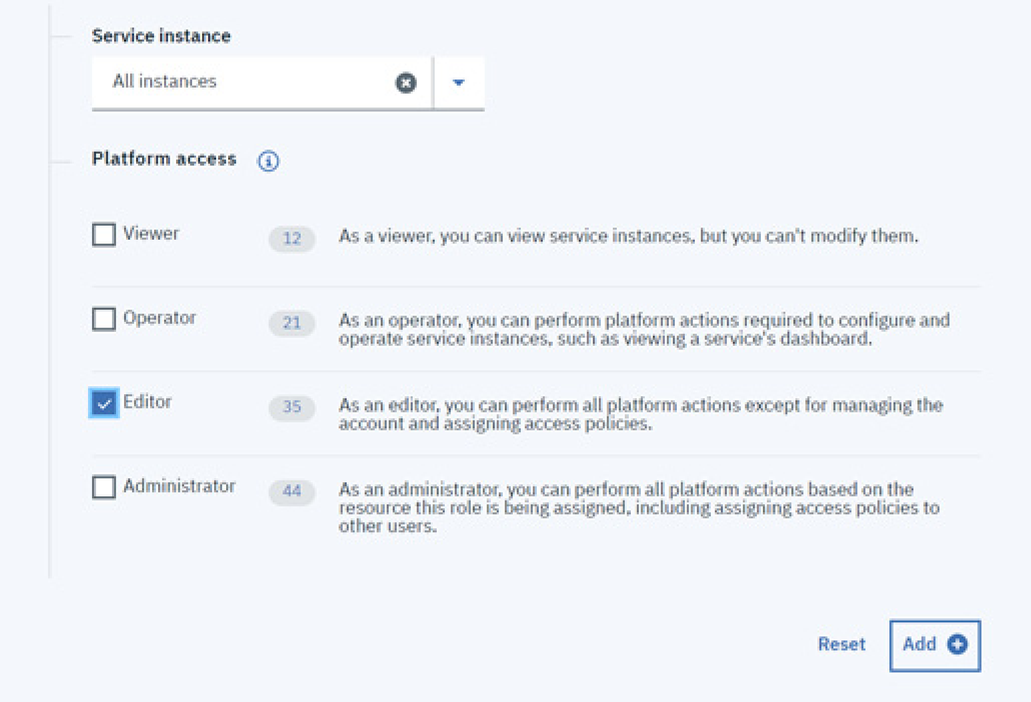
What type of access do you want to assign?

Toolchain in

Account

For users to access resources, assign at least the Viewer role on the containing resource group.

__ b. For **Service instance** select **All instances.** Then, select the level of access (Administrator, Editor, Operator, or Viewer) for the resource under “Platform access” and click **Add** as shown in the figure.



Service instance

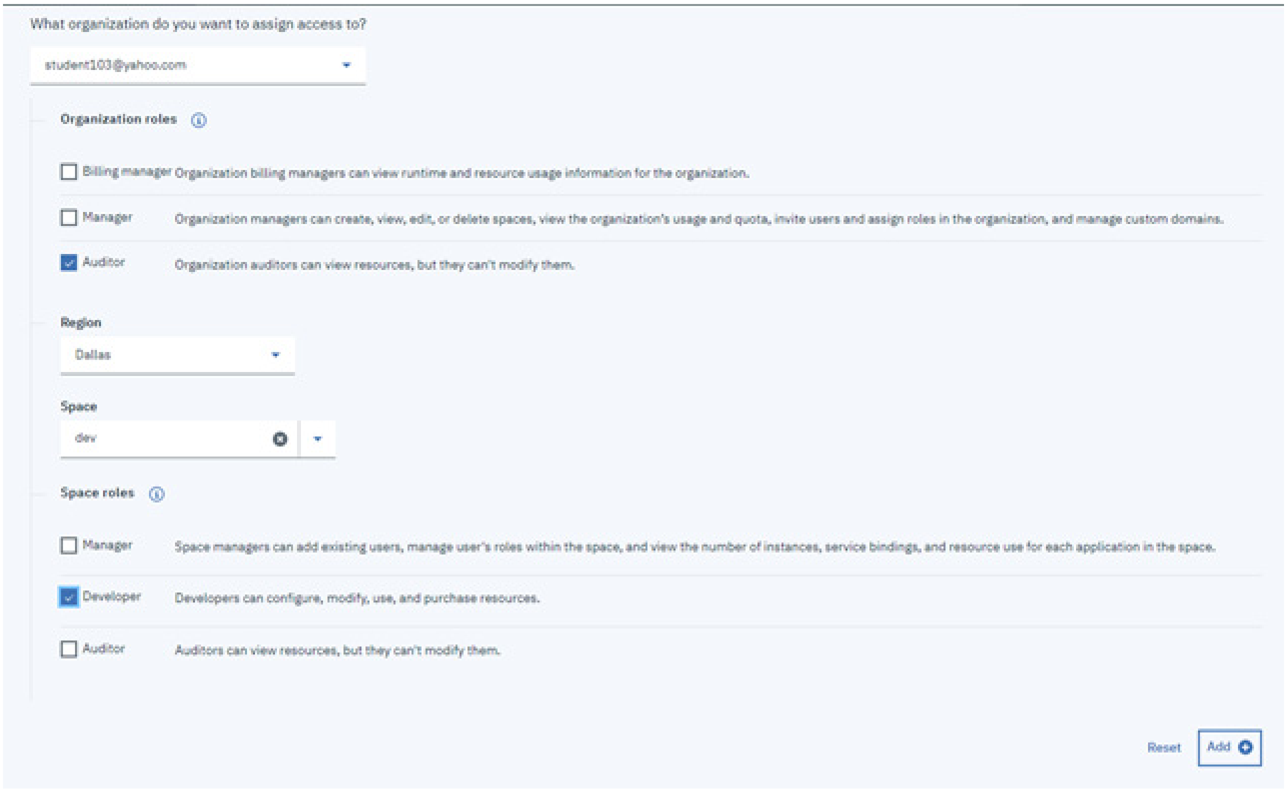
All instances

Platform access

□ Viewer	12	As a viewer, you can view service instances, but you can't modify them.
□ Operator	21	As an operator, you can perform platform actions required to configure and operate service instances, such as viewing a service's dashboard.
☑ Editor	35	As an editor, you can perform all platform actions except for managing the account and assigning access policies.
□ Administrator	44	As an administrator, you can perform all platform actions based on the resource this role is being assigned, including assigning access policies to other users.

__ 5. Adding access to Cloud Foundry application.

__ a. In the “Assign users additional access” pane, select **Cloud Foundry** to give access to the Cloud Foundry organization to which the resource belongs. Complete the Organization (use your IBM ID), Organization roles, Region, Space, and Space roles fields and then click **Add**, as shown in the following figure. In this example, you grant **Developer** access to the user.



What organization do you want to assign access to?

student103@yahoo.com

Organization roles

☐ Billing manager Organization billing managers can view runtime and resource usage information for the organization.

☐ Manager Organization managers can create, view, edit, or delete spaces, view the organization's usage and quota, invite users and assign roles in the organization, and manage custom domains.

☑ Auditor Organization auditors can view resources, but they can't modify them.

Region

Dallas

Space

dev

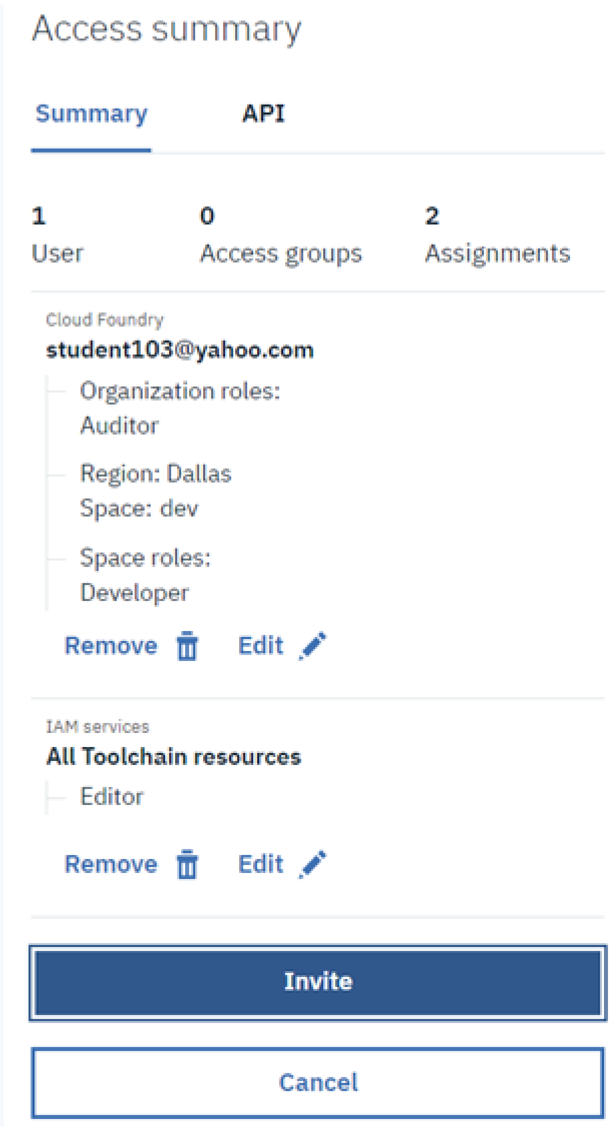
Space roles

☐ Manager Space managers can add existing users, manage user's roles within the space, and view the number of instances, service bindings, and resource use for each application in the space.

☑ Developer Developers can configure, modify, use, and purchase resources.

☐ Auditor Auditors can view resources, but they can't modify them.

__ 6. Click **Invite**.



Access summary

1 User
0 Access groups
2 Assignments

Cloud Foundry
student103@yahoo.com
Organization roles: Auditor
Region: Dallas
Space: dev
Space roles: Developer

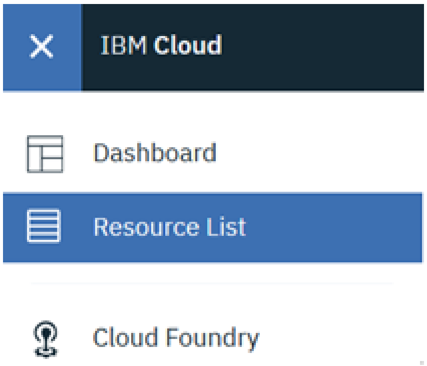
IAM services
All Toolchain resources
Editor

Part 6: Editing the sample application

In this part, edit your application code with Eclipse Orion web IDE by completing the following steps:

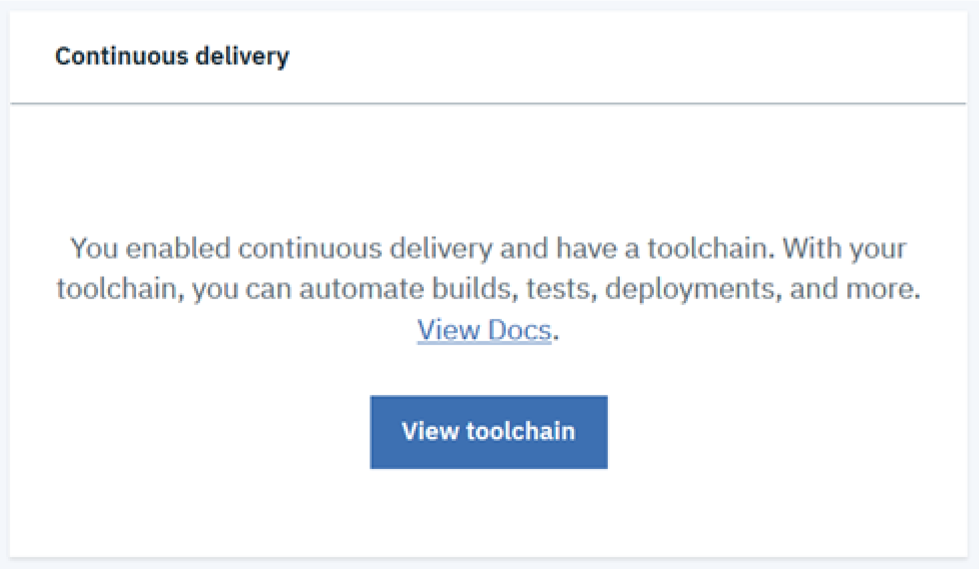
__ 1. Open the sample application in Eclipse Orion:

__ a. Click the Navigation Menu icon ☰ in the upper left and click **Resource List**, as shown in the following figure.



__ b. Expand **Cloud Foundry Apps** and click your application (**vy301-xxx-nodesample**).

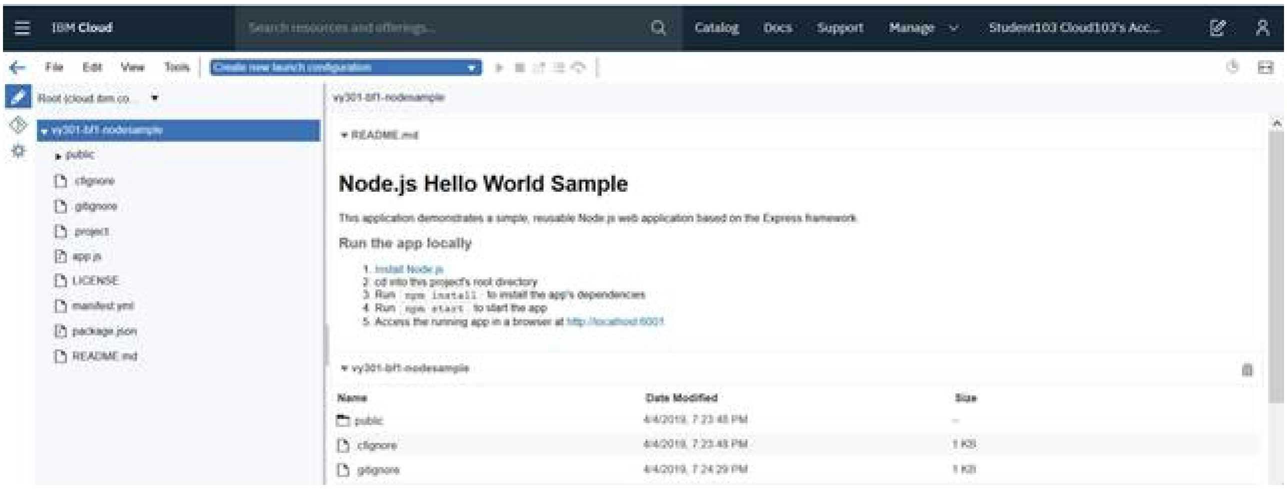
__ c. Scroll down and click **View toolchain**, as shown in the following figure.


Continuous delivery

You enabled continuous delivery and have a toolchain. With your toolchain, you can automate builds, tests, deployments, and more. View Docs.

__ d. Click **Eclipse Orion Web IDE**.

__ e. Expand the project (**vy301-xxx-nodesample**) to confirm that a set of files and directories are in the project, as shown in the following figure.



Node.js Hello World Sample

This application demonstrates a simple, reusable Node.js web application based on the Express framework.

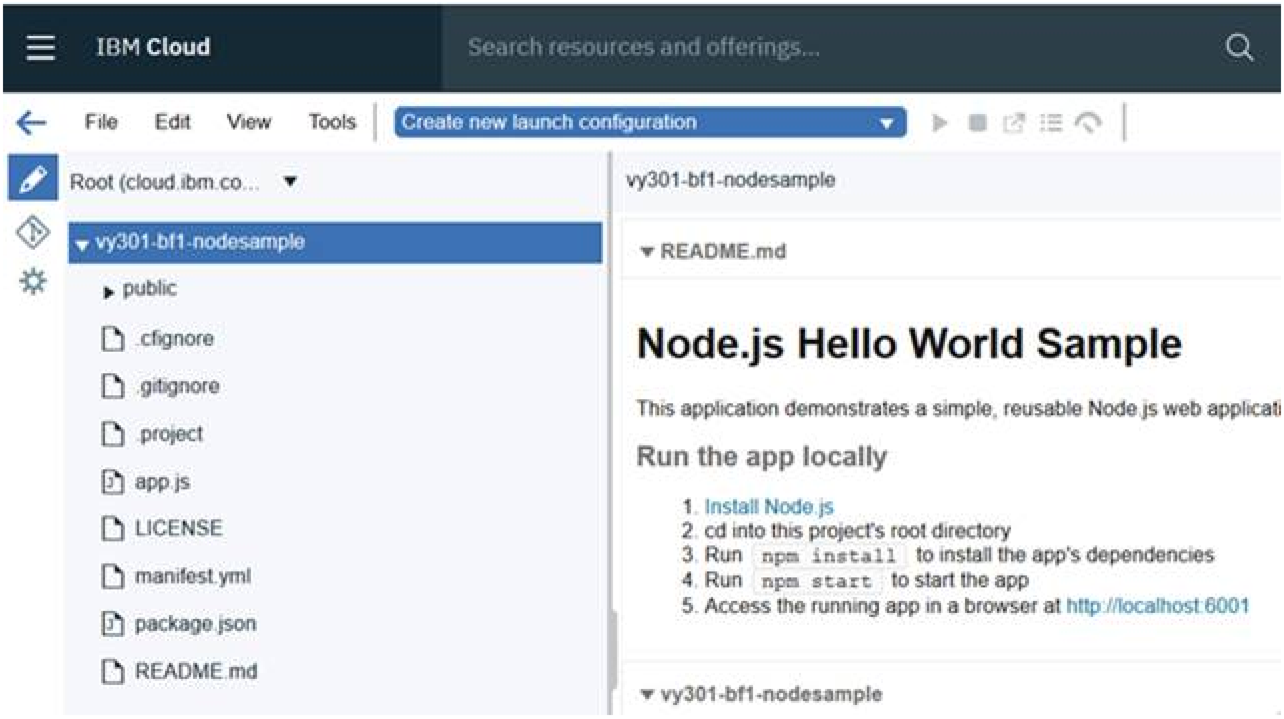
Run the app locally

1. Install Node.js
2. cd into this project's root directory
3. Run `npm install` to install the app's dependencies
4. Run `npm start` to start the app
5. Access the running app in a browser at http://localhost:6001

vy301-bf1-nodesample

Name	Date Modified	Size
public	4/4/2019, 7:23:48 PM	--
.cfignore	4/4/2019, 7:23:48 PM	1 KB
.gitignore	4/4/2019, 7:24:29 PM	1 KB

__ 2. Examine the three options in the left pane, as shown in the following figure. Hover your cursor over the icons to see their descriptions.



Node.js Hello World Sample

This application demonstrates a simple, reusable Node.js web applicati

Run the app locally

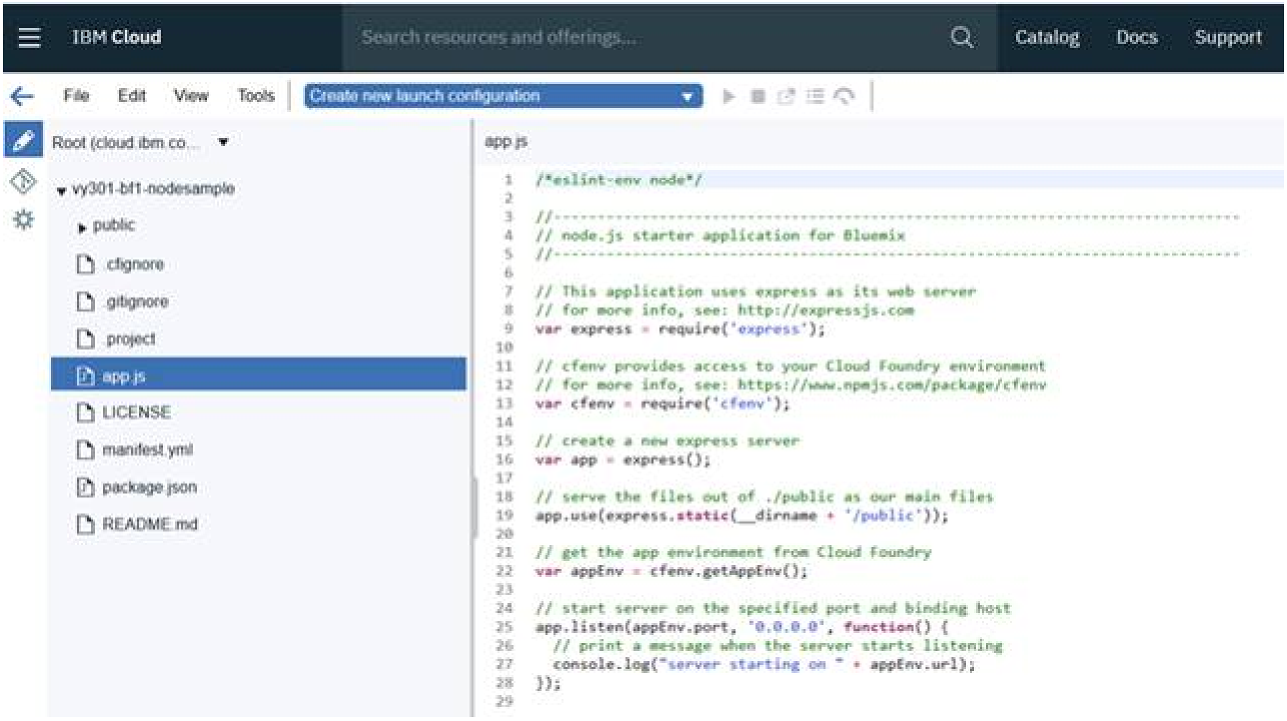
1. Install Node.js
2. cd into this project's root directory
3. Run `npm install` to install the app's dependencies
4. Run `npm start` to start the app
5. Access the running app in a browser at http://localhost:6001

vy301-bf1-nodesample

The following options are available:

- The **Edit** option (first icon) provides a graphical editor to manage and update project files.
- The **Git** option allows you to select and check in files from the project into the Git repository.
- The **Settings** option allows you to view and modify the settings to customize the Web IDE. You can choose the color schemes, technical tools, and settings that meet your development needs.

__ 3. Click **app.js** from the project directory and examine the contents of the `app.js` source code in the editor, as shown in the following figure.



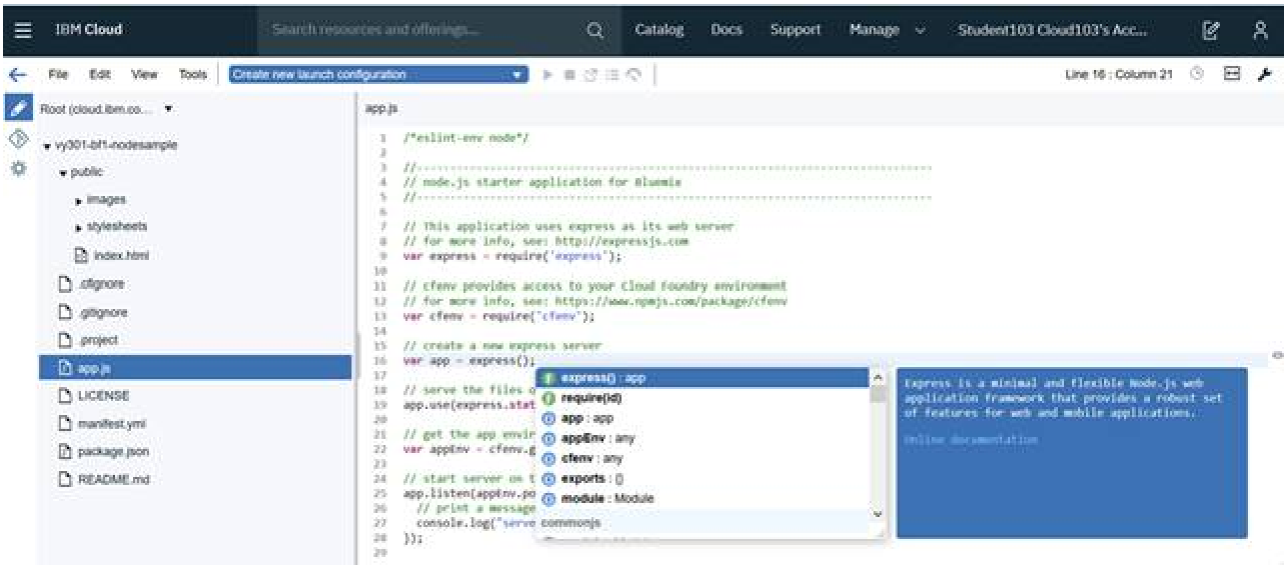
```
1   /*eslint-env node*/
2
3   //------------------------------------------------------------------------------
4   // node.js starter application for Bluemix
5   //------------------------------------------------------------------------------
6
7   // This application uses express as its web server
8   // for more info, see: http://expressjs.com
9   var express = require('express');
10
11  // cfenv provides access to your Cloud Foundry environment
12  // for more info, see: https://www.npmjs.com/package/cfenv
13  var cfenv = require('cfenv');
14
15  // create a new express server
16  var app = express();
17
18  // serve the files out of ./public as our main files
19  app.use(express.static(__dirname + '/public'));
20
21  // get the app environment from Cloud Foundry
22  var appEnv = cfenv.getAppEnv();
23
24  // start server on the specified port and binding host
25  app.listen(appEnv.port, '0.0.0.0', function() {
26    // print a message when the server starts listening
27    console.log("server starting on " + appEnv.url);
28  });
29
```

The editor in the web application provides the same features as the desktop Eclipse application:

- Syntax highlighting.
- Static code analysis of the JavaScript code.
- A preview of the document.

__ 4. Place your cursor in line 16 next to the term **`express()`** in `var app = express()`.

__ 5. Press **Ctrl + Spacebar** on your keyboard. You see the window that is shown in the following figure.



```
/*eslint-env node*/

//------------------------------------------------------------------------------
// node.js starter application for Bluemix
//------------------------------------------------------------------------------

// This application uses express as its web server
// for more info, see: http://expressjs.com
var express = require('express');

// cfenv provides access to your Cloud Foundry environment
// for more info, see: https://www.npmjs.com/package/cfenv
var cfenv = require('cfenv');

// create a new express server
var app = express();

// serve the files o
app.use(express.stat

// get the app envir
var appEnv = cfenv.g

// start server on t
app.listen(appEnv.po
  // print a message
  console.log("serve
});
```

Express is a minimal and flexible Node.js web application framework that provides a robust set of features for web and mobile applications.

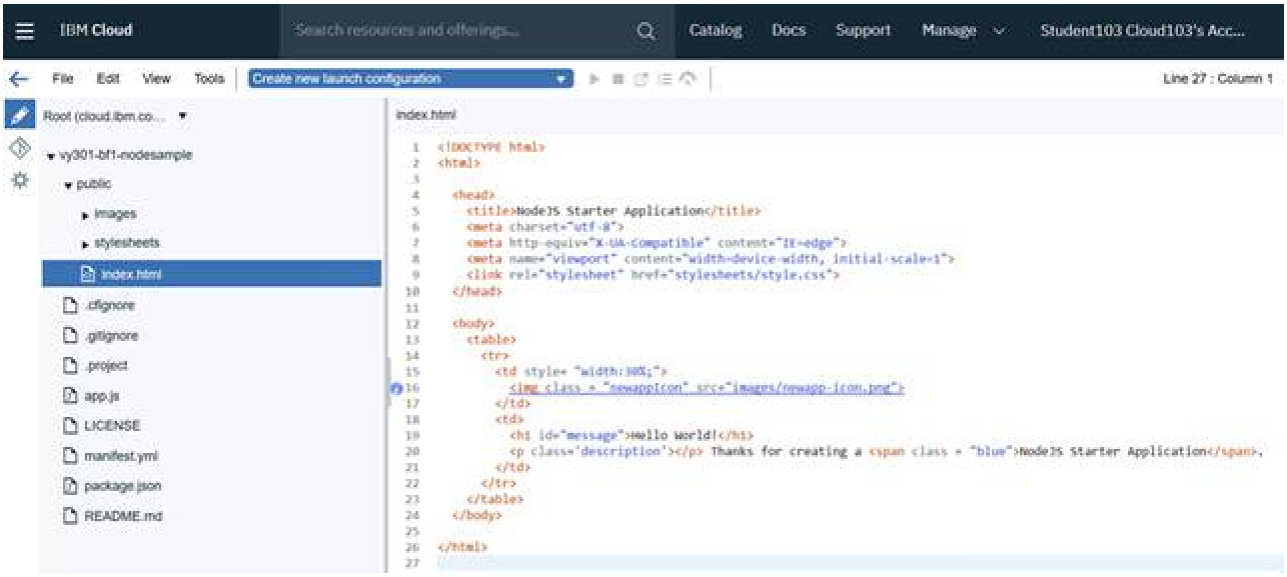
Online documentation

The code completion feature in the JavaScript editor supports Node JavaScript modules and the standard JavaScript functions.

__ 6. Examine the HTML templates in the sample application welcome page:

__ a. In the project directory on the left, expand the **public** folder.

__ b. Click **index.html**. The window that is shown in the following figure opens.

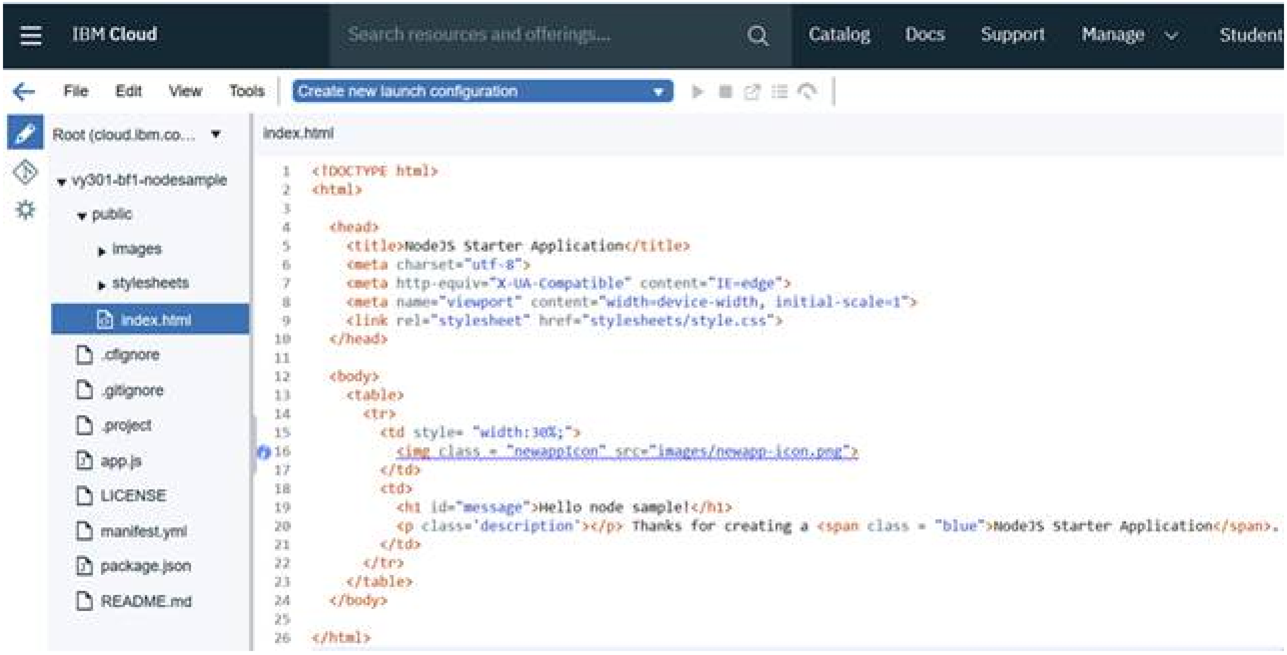


```
1   <!DOCTYPE html>
2   <html>
3
4     <head>
5       <title>NodeJS Starter Application</title>
6       <meta charset="utf-8">
7       <meta http-equiv="X-UA-Compatible" content="IE=edge">
8       <meta name="viewport" content="width=device-width, initial-scale=1">
9       <link rel="stylesheet" href="stylesheets/style.css">
10    </head>
11
12    <body>
13      <table>
14        <tr>
15          <td style= "width:30%;">
16            <img class = "newappIcon" src="images/newapp-icon.png">
17          </td>
18          <td>
19            <h1 id="message">Hello World!</h1>
20            <p class='description'></p> Thanks for creating a <span class = "blue">NodeJS Starter Application</span>.
21          </td>
22        </tr>
23      </table>
24    </body>
25
26  </html>
27
```

The built-in editor also highlights and validates HTML web page markup.

__ c. Select the sentence in line 19.

__ d. Change the phrase inside the <h1> tags to "`Hello node sample!`", as shown in the following figure.



```
<!DOCTYPE html>
<html>

  <head>
    <title>NodeJS Starter Application</title>
    <meta charset="utf-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <link rel="stylesheet" href="stylesheets/style.css">
  </head>

  <body>
    <table>
      <tr>
        <td style= "width:30%;">
          <img class = "newappIcon" src="images/newapp-icon.png">
        </td>
        <td>
          <h1 id="message">Hello node sample!</h1>
          <p class='description'></p> Thanks for creating a <span class = "blue">NodeJS Starter Application</span>.
        </td>
      </tr>
    </table>
  </body>

</html>
```

__ e. All the code changes are automatically saved. To force the save operation, press **Ctrl** + **S** to save your changes.

Part 7: Committing your changes to the Git repository

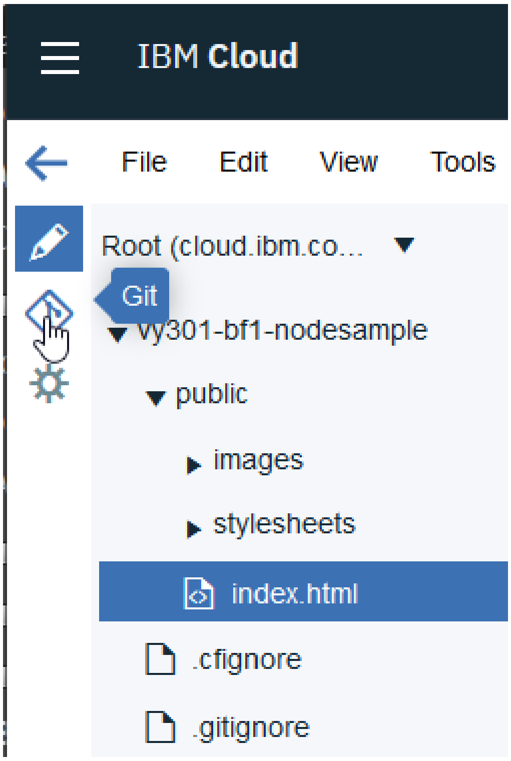
In this part, you see the effects of the source code changes by committing your changes to the Git repository, and then pushing the changes to your IBM Cloud application.

You commit the source code, configuration files, and other artifacts to the Git repository. Also, you deploy the application changes from the Git repository to the application in your IBM Cloud account. In the next Part, you learn that Delivery Pipeline automatically deploys the changes that are pushed to the Git repository as an application in your IBM Cloud account.

Complete the following steps:

__ 1. Switch to Git in the left navigation bar:

__ a. Click the **Git** icon in the left pane, as shown in the following figure.

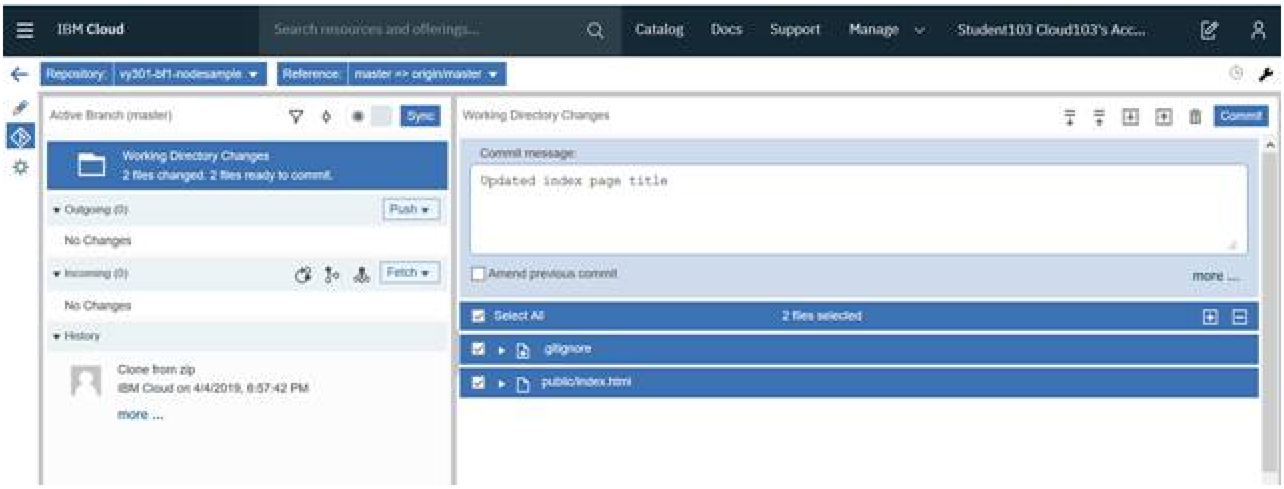


__ 2. Add and commit the changes in the project to the Git repository:

__ a. Examine the Working Directory Changes pane, as shown in the following figure. The Working Directory Changes window lists the files that you added to the workspace or modified.

__ b. Enter **“Updated index page title”** into the comment window.

__ c. Make sure that the **Select All** check box is selected.


IBM Cloud
Search resources and offerings...
Catalog
Docs
Support
Manage
Student103 Cloud103's Acc...
Repository: vy301-bf1-nodesample
Reference: master => origin/master
Active Branch (master)
Sync
Working Directory Changes
2 files changed. 2 files ready to commit.
Outgoing (0)
Push
No Changes
Incoming (0)
Fetch
No Changes
History
Clone from zip
IBM Cloud on 4/4/2019, 6:57:42 PM
more ...
Working Directory Changes
Commit
Amend previous commit
more ...

Commit message:

```
Updated index page title
```

☑ Select All   2 files selected

☑ .gitignore

☑ public/index.html

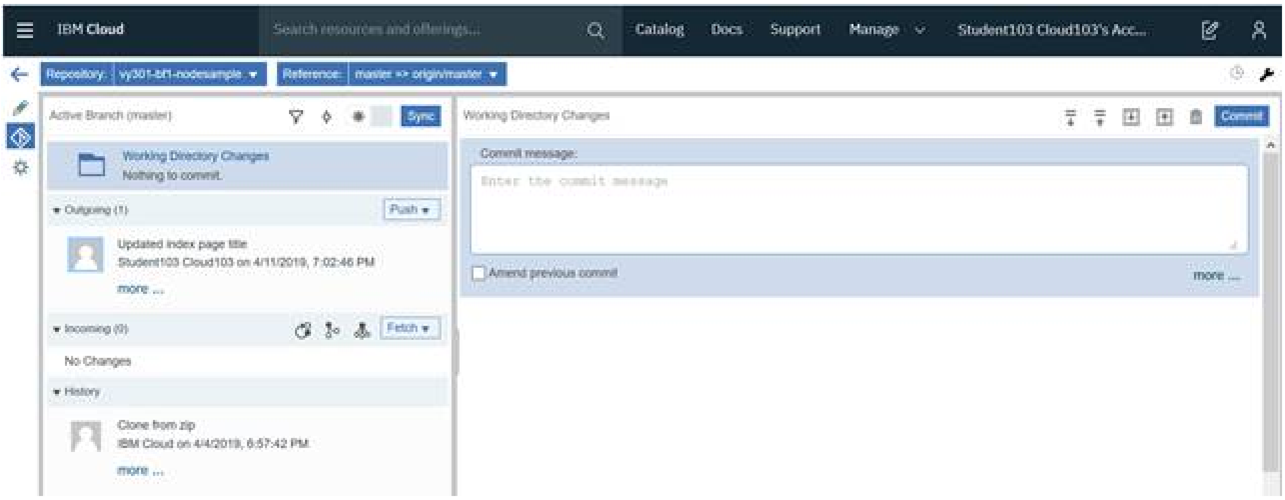
__ d. Click **Commit** to save the files into the Git repository.

Information

If you are prompted to provide the author and committer names as mandatory fields, enter your name and email.

__ e. Examine the Working Directory Changes view.

Confirm that there are no more updated files to commit from the Working Directory Changes section and that the change set is in the Outgoing section, as shown in the following figure.



Repository: vy301-bf1-nodesample

Reference: master => origin/master

Working Directory Changes
Nothing to commit.

Outgoing (1)

Updated index page title
Student103 Cloud103 on 4/11/2019, 7:02:46 PM

Incoming (0)

No Changes

History

Clone from zip
IBM Cloud on 4/4/2019, 6:57:42 PM

Working Directory Changes

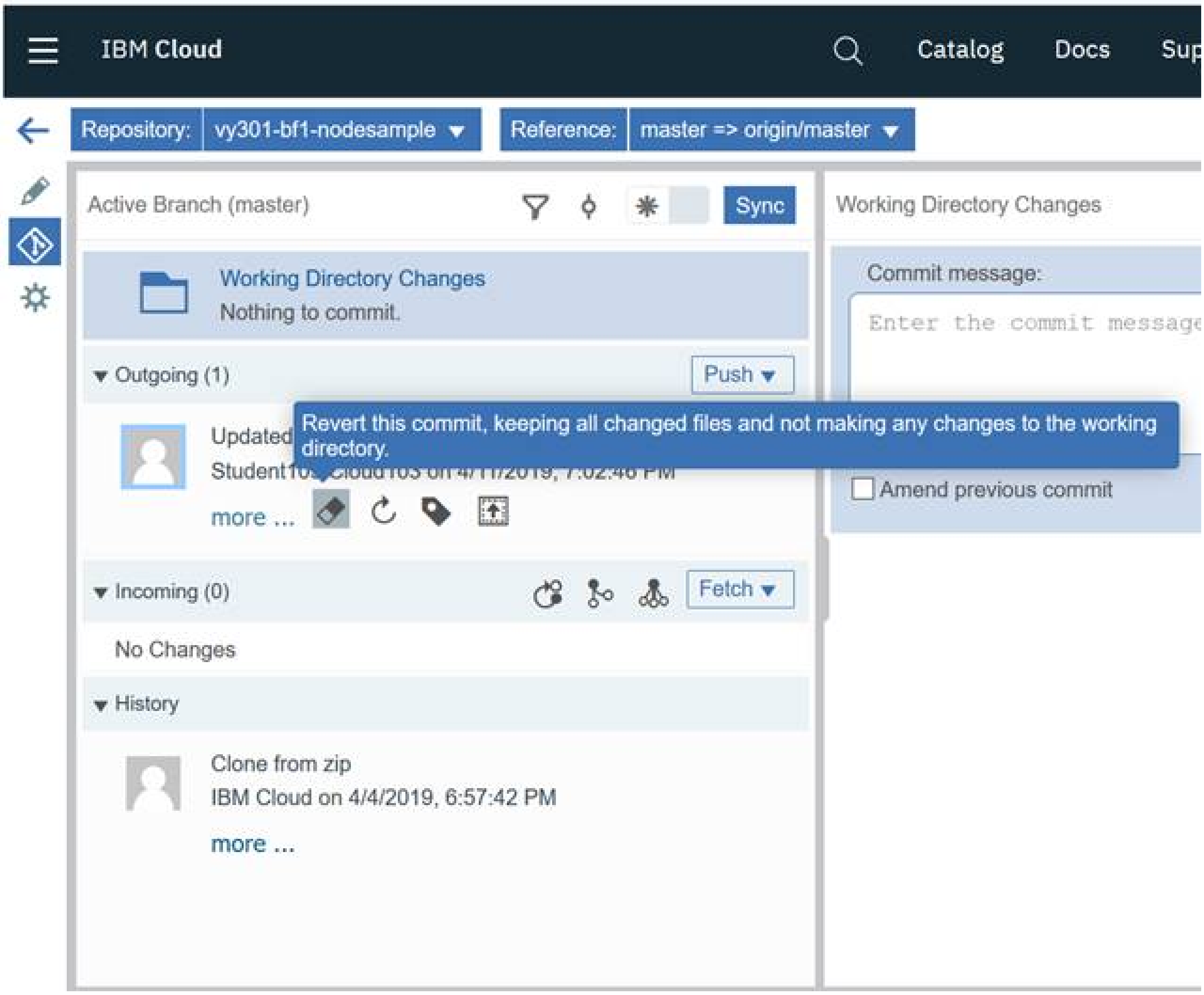
Commit message:

Enter the commit message

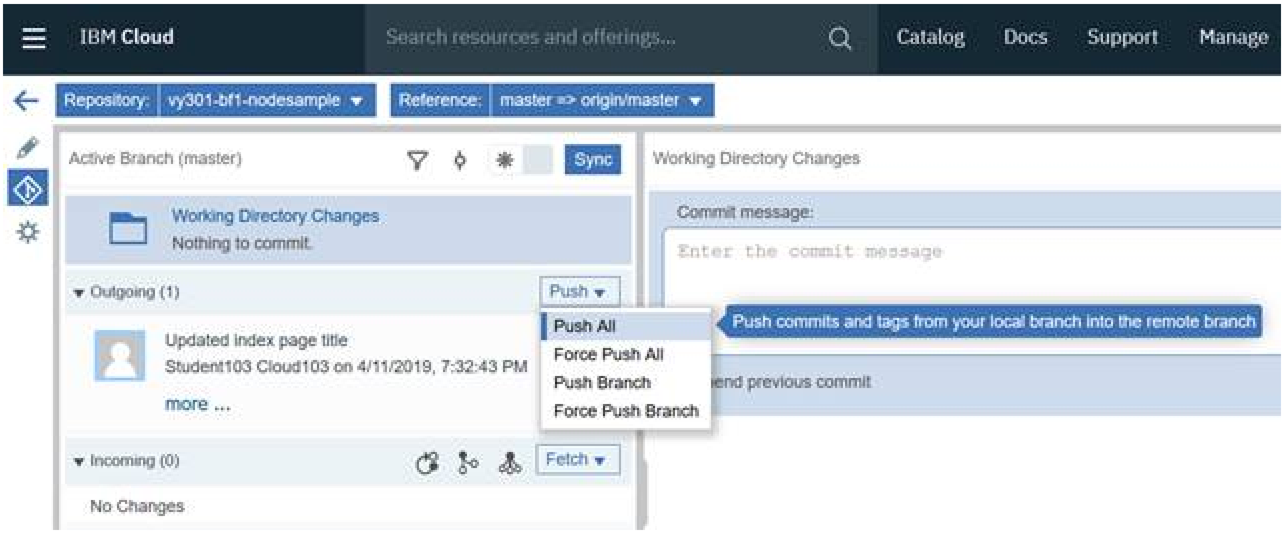
□ Amend previous commit

Information

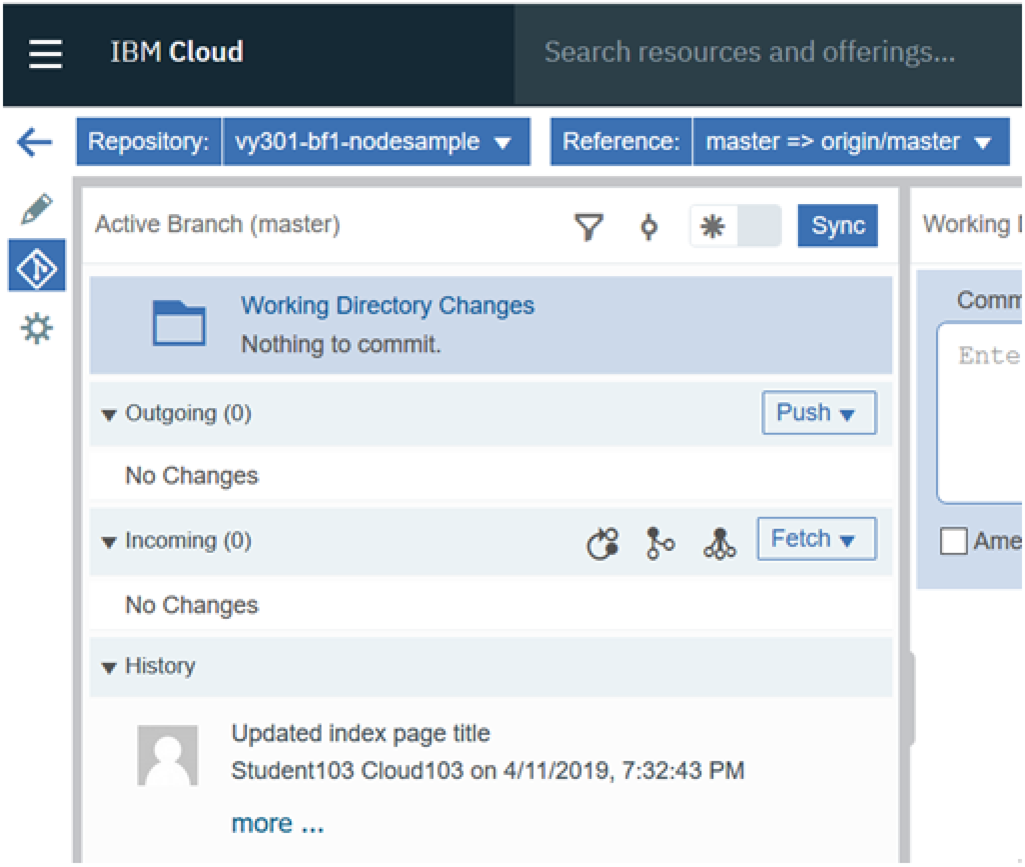
Do **not** perform this step; it is for your information only. You can revert to the last saved version of your source code in the Git repository. To do so, select the change set that you want to revert in the Outgoing section and hover your cursor over the eraser icon, as shown in the following figure. Do **not** click the eraser icon now.



__ f. Push your changes from the local branch to the remote Git branch by clicking the **Push** drop-down menu and selecting **Push All** in the Outgoing pane, as shown in the following figure.

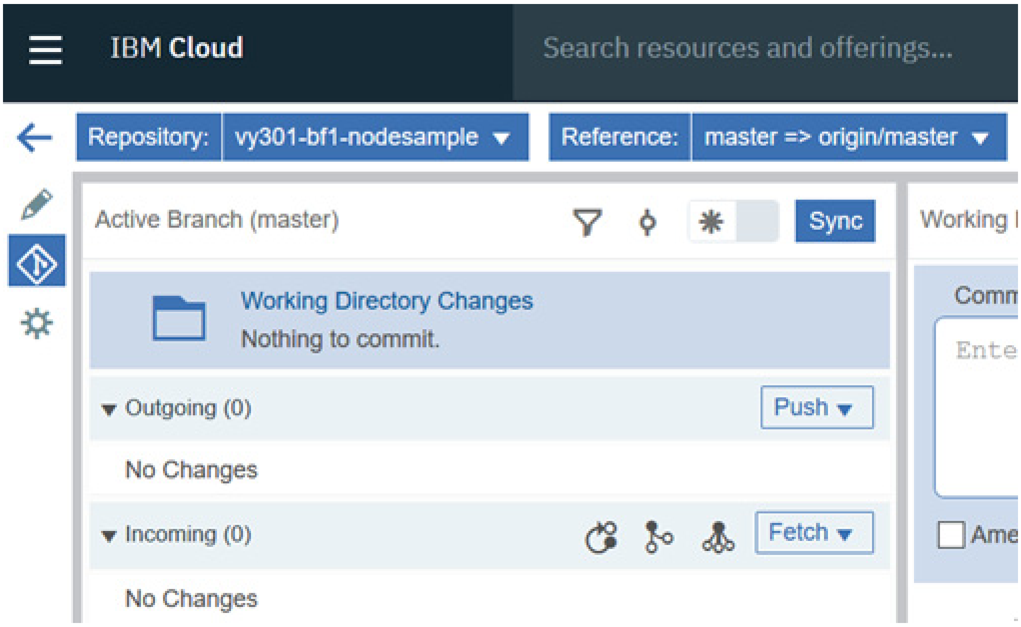

IBM Cloud
Search resources and offerings...
Catalog
Docs
Support
Manage
Repository: vy301-bf1-nodesample
Reference: master => origin/master
Active Branch (master)
Sync
Working Directory Changes
Working Directory Changes
Nothing to commit.
Commit message:
Enter the commit message
Outgoing (1)
Push
Push All
Force Push All
Push Branch
Force Push Branch
Push commits and tags from your local branch into the remote branch
Updated index page title
Student103 Cloud103 on 4/11/2019, 7:32:43 PM
more ...
end previous commit
Incoming (0)
Fetch
No Changes

You can see that the Outgoing pane is clear because all commits were pushed to the remote Git repository, as shown in the following figure.


IBM Cloud
Search resources and offerings...
Repository: vy301-bf1-nodesample
Reference: master => origin/master
Active Branch (master)
Sync
Working I
Working Directory Changes
Nothing to commit.
Comm
Ente
Outgoing (0)
Push
No Changes
Incoming (0)
Fetch
Ame
No Changes
History
Updated index page title
Student103 Cloud103 on 4/11/2019, 7:32:43 PM
more ...

__ 3. Examine the Git repository commits log:

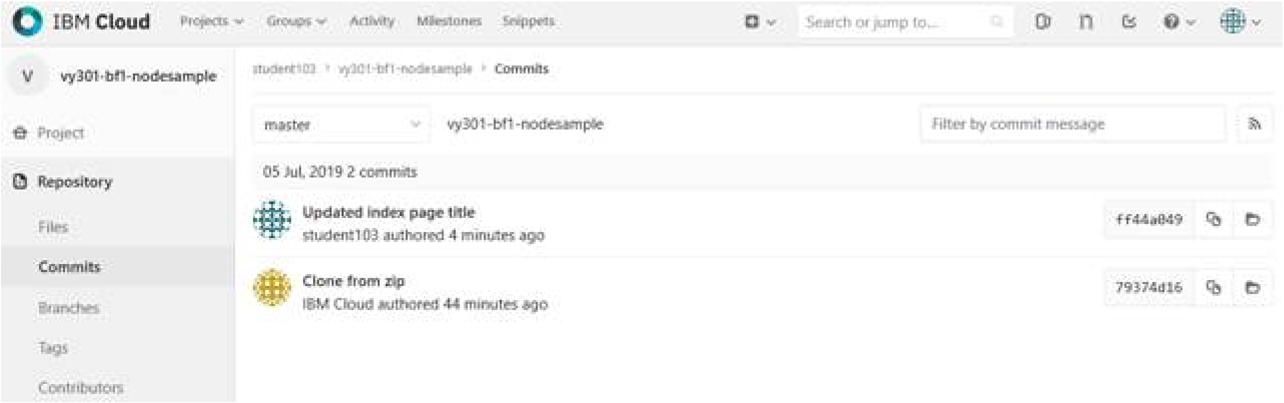
__ a. Click the back-arrow (**Back to Toolchains**) icon in the upper left of the Eclipse Orion Web IDE page, as shown in following figure.



__ b. Click **Git** from the Continuous Delivery Toolchain dashboard.

__ c. Click **Repository** from the left navigation bar.

__ d. Click **Commits**, as shown in the following figure. Examine the Git COMMITS for `vy301-xxx-nodesample`.



05 Jul, 2019 2 commits

Updated index page title
student103 authored 4 minutes ago
ff44a049

Clone from zip
IBM Cloud authored 44 minutes ago
79374d16

You can see the last commit and all commits from the beginning. Because you chose the default value (**Clone**) for the Repository type when you created the Continuous Delivery Toolchain, the figure shows the “Initial commit” action, which indicates that the Node.js sample code was cloned (**Clone from zip**). Each commit action has a unique identifier, user name, and date and time stamp.

Part 8: Deploying the application from the Git repository to IBM Cloud

There are two ways to deploy an application to your IBM Cloud account:

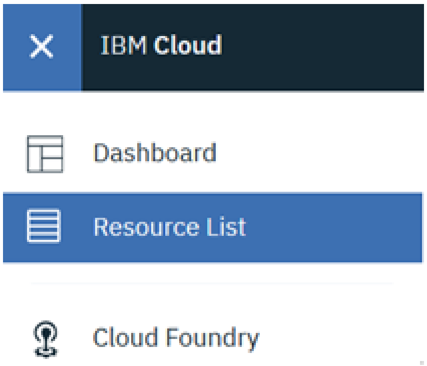
- Deploy from the Git repository. You can use Delivery Pipeline to automate the application deployment when changes are pushed to Git.
- Deploy directly from your project workspace in Eclipse Orion.

Note

If more than one developer is working on a project, consider checking in your changes and pushing the most recent revision from the Git repository.

__ 1. Validate that the app is deployed automatically when changes are pushed to Git by using Delivery Pipeline. Complete the following steps:

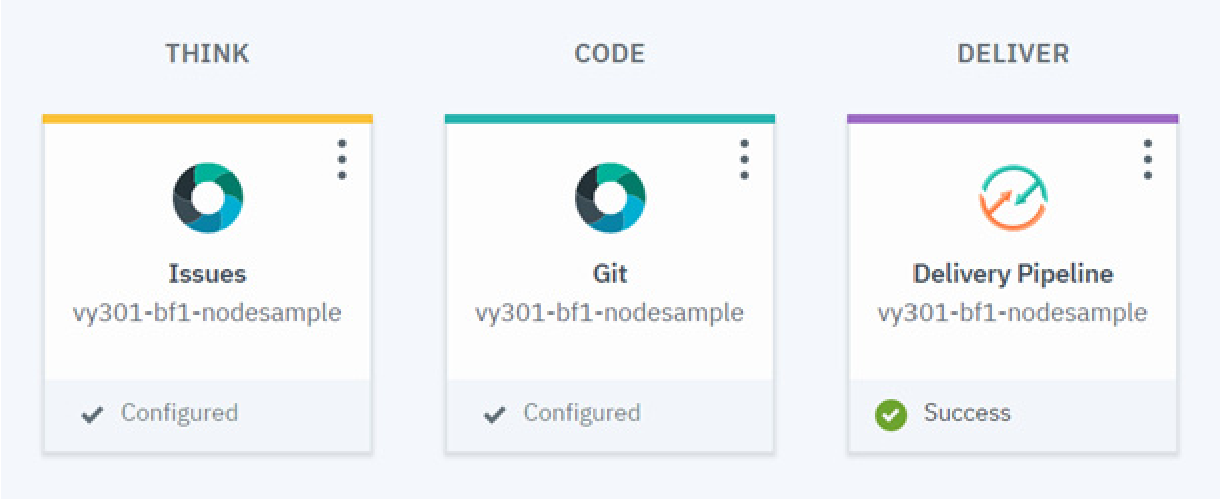
__ a. Click the Navigation Menu icon ☰ in the upper left and select **Resource List**, as shown in the following figure.



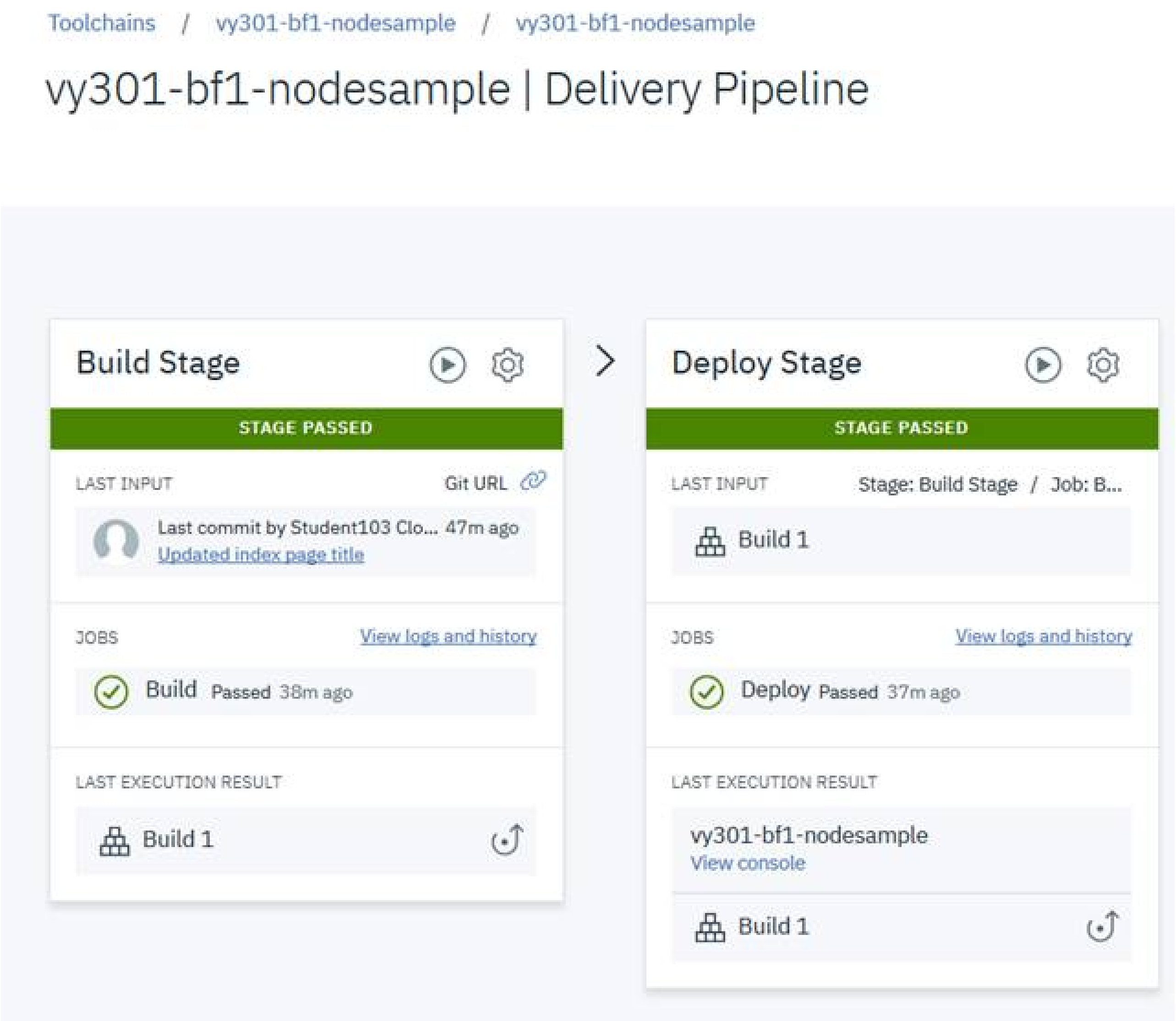
__ b. Select your application from the **Cloud Foundry Apps** menu.

__ c. From the Application Details page, click **View toolchain**.

__ d. A new tab opens and shows the toolchain. Click **Delivery Pipeline**, as shown in the following figure.



__ e. Examine the delivery pipeline. Wait until the build and deploy tasks complete successfully, as shown in the following figure.


Toolchains / vy301-bf1-nodesample / vy301-bf1-nodesample

vy301-bf1-nodesample | Delivery Pipeline

Build Stage

STAGE PASSED

LAST INPUT Git URL

Last commit by Student103 Clo... 47m ago
Updated index page title

JOBS View logs and history

Build Passed 38m ago

LAST EXECUTION RESULT

Build 1

Deploy Stage

STAGE PASSED

LAST INPUT Stage: Build Stage / Job: B...

Build 1

JOBS View logs and history

Deploy Passed 37m ago

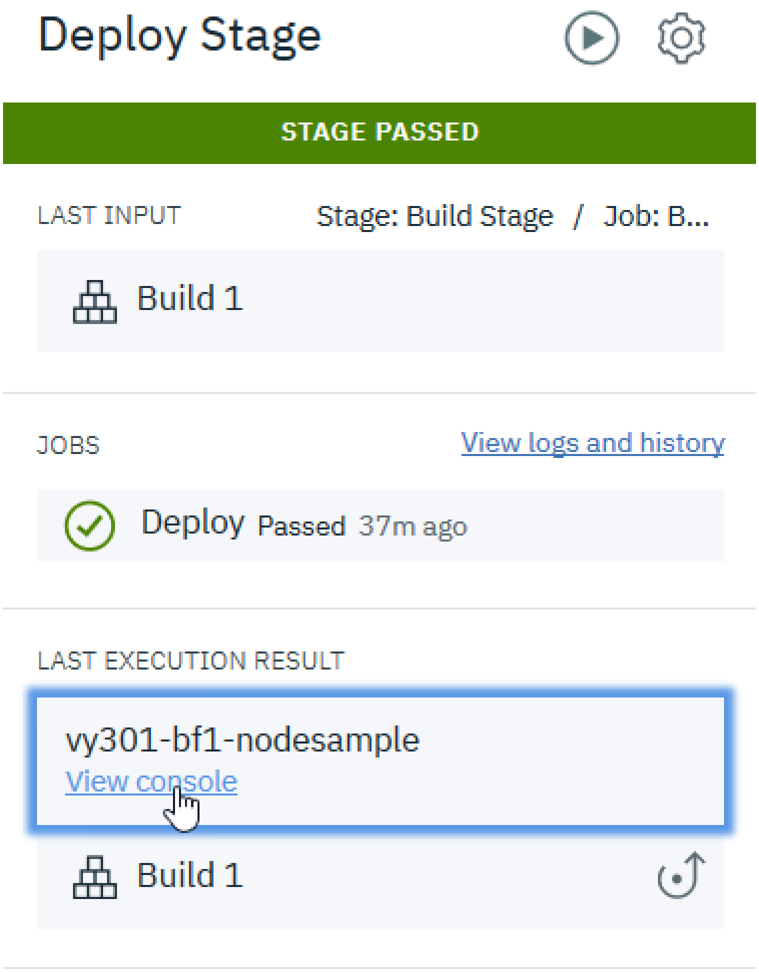
LAST EXECUTION RESULT

vy301-bf1-nodesample
View console

Build 1

__ 2. Examine the updated code:

__ a. In Deploy Stage, click **View console** to view the application overview, as shown in the following figure.



Deploy Stage

STAGE PASSED

LAST INPUT Stage: Build Stage / Job: B...

Build 1

JOBS

Deploy Passed 37m ago

LAST EXECUTION RESULT

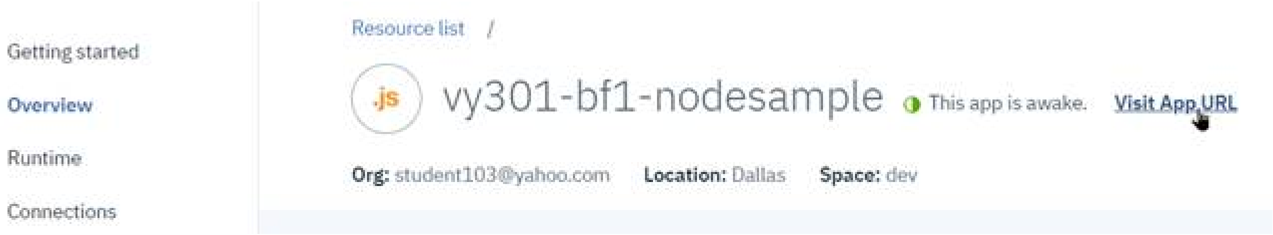
vy301-bf1-nodesample

Build 1

Note

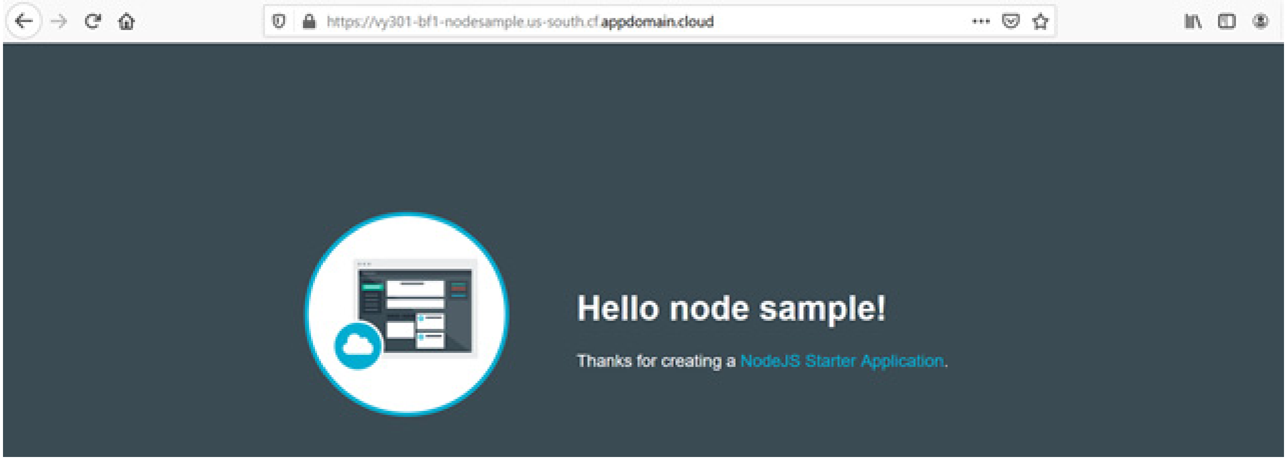
By default, a Git push on the *master* Git branch triggers the application to be built and deployed automatically. Pushing the changes to any other branch does not trigger a deployment unless this branch is merged with the master branch. If required, you can modify the Delivery Pipeline to automatically trigger a build if a change is pushed to a branch other than the master branch.

__ b. In the Application Overview section, click **Visit App URL**, as shown in the following figure.



Org: student103@yahoo.com Location: Dallas Space: dev

__ c. Confirm that the application web page changed to “Hello node sample!” as shown in the following figure.



Hello node sample!

Thanks for creating a NodeJS Starter Application.

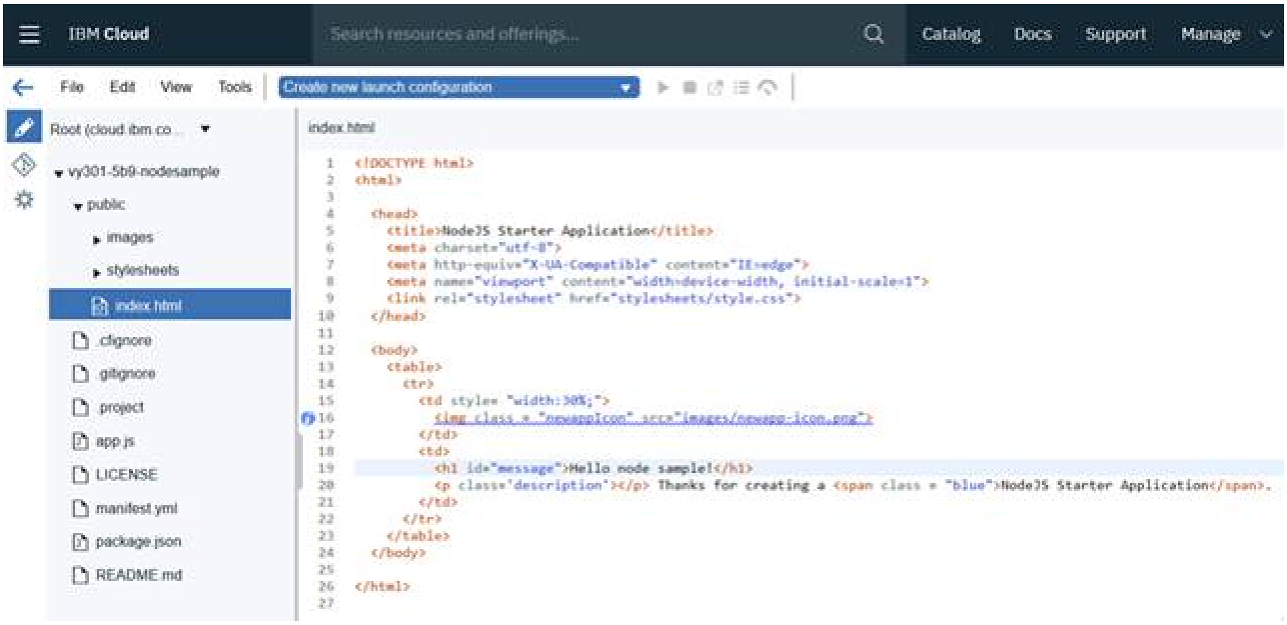
Part 9: Deploying the application directly from your project workspace to IBM Cloud

You can publish changes to your application directly from the Eclipse Orion Web IDE to IBM Cloud. With this technique, you can quickly test changes to your code on an actual IBM Cloud account by completing the following steps:

__ 1. In the Eclipse Orion Web IDE, edit the message in the sample application web page:

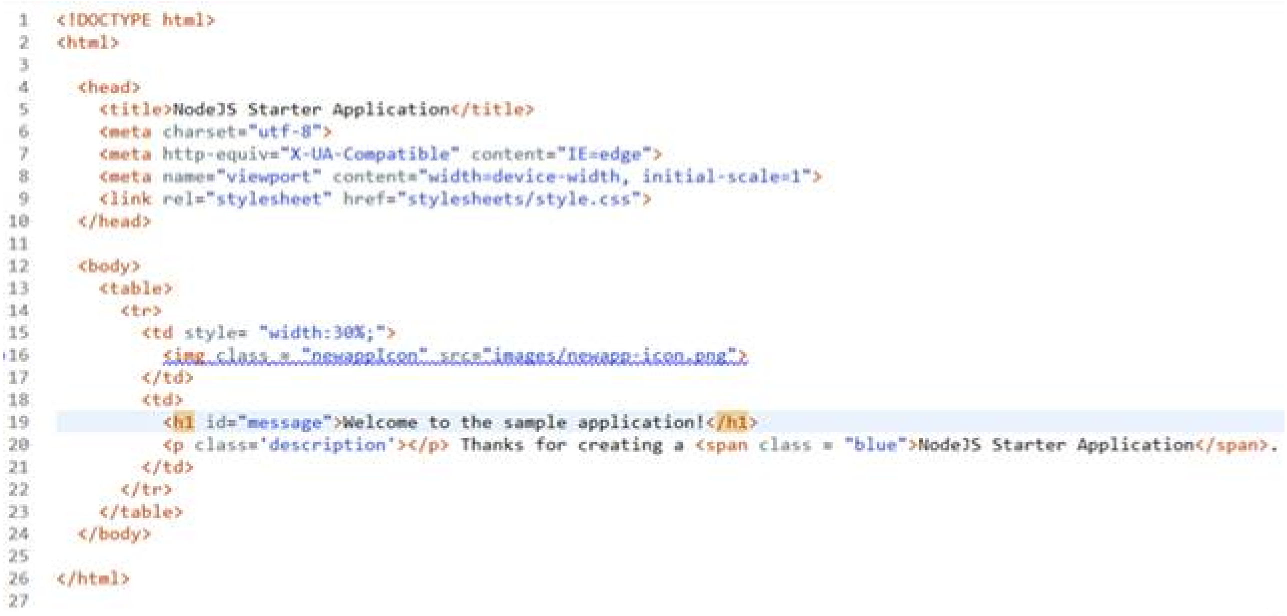
__ a. On the Toolchains page, click **Eclipse Orion Web IDE** and switch to Edit mode from the left navigation bar.

__ b. Open `index.html` in the public folder, as shown in the following figure.



```
1   <!DOCTYPE html>
2   <html>
3
4     <head>
5       <title>NodeJS Starter Application</title>
6       <meta charset="utf-8">
7       <meta http-equiv="X-UA-Compatible" content="IE=edge">
8       <meta name="viewport" content="width=device-width, initial-scale=1">
9       <link rel="stylesheet" href="stylesheets/style.css">
10    </head>
11
12    <body>
13      <table>
14        <tr>
15          <td style= "width:30%;">
16            <img class = "newappIcon" src="images/newapp-icon.png">
17          </td>
18          <td>
19            <h1 id="message">Hello node sample!</h1>
20            <p class='description'></p> Thanks for creating a <span class = "blue">NodeJS Starter Application</span>.
21          </td>
22        </tr>
23      </table>
24    </body>
25
26  </html>
27
```

__ c. In line 19 of `index.html`, change the heading to "`Welcome to the sample application!`", as shown in the following figure.

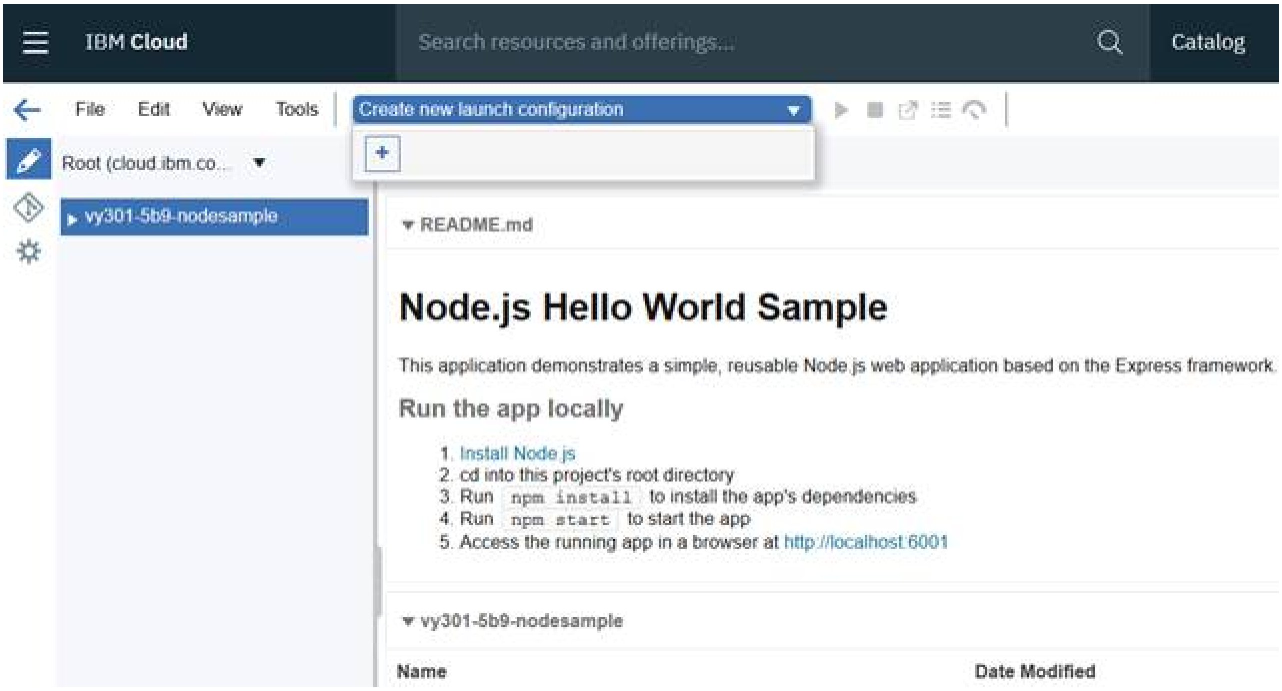


```
1   <!DOCTYPE html>
2   <html>
3
4     <head>
5       <title>NodeJS Starter Application</title>
6       <meta charset="utf-8">
7       <meta http-equiv="X-UA-Compatible" content="IE=edge">
8       <meta name="viewport" content="width=device-width, initial-scale=1">
9       <link rel="stylesheet" href="stylesheets/style.css">
10    </head>
11
12    <body>
13      <table>
14        <tr>
15          <td style= "width:30%;">
16            <img class = "newappIcon" src="images/newapp-icon.png">
17          </td>
18          <td>
19            <h1 id="message">Welcome to the sample application!</h1>
20            <p class='description'></p> Thanks for creating a <span class = "blue">NodeJS Starter Application</span>.
21          </td>
22        </tr>
23      </table>
24    </body>
25
26  </html>
27
```

__ d. All the code changes are automatically saved. To force the save action, press **Ctrl** + **S**.

__ 2. Deploy the application directly from your project workspace:

__ a. Configure the application deployment settings by clicking **Create new launch configuration** and then the plus button (**+**) in the upper center of the Eclipse Orion Web IDE window, as shown in following figure.



▼ README.md

Node.js Hello World Sample

This application demonstrates a simple, reusable Node.js web application based on the Express framework.

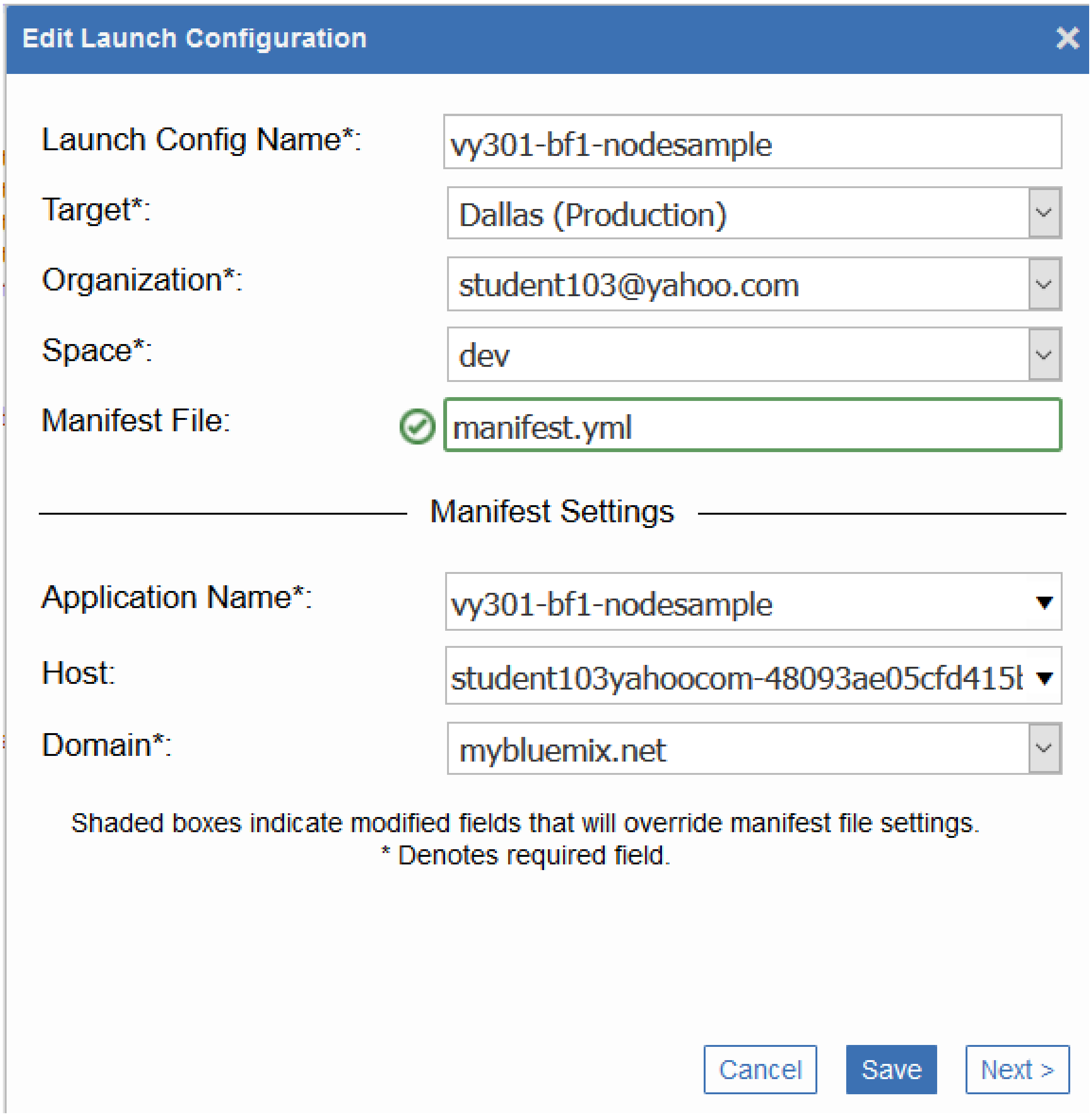
Run the app locally

1. Install Node.js
2. cd into this project's root directory
3. Run `npm install` to install the app's dependencies
4. Run `npm start` to start the app
5. Access the running app in a browser at http://localhost:6001

▼ vy301-5b9-nodesample

Name	Date Modified

__ b. Wait until the deployment settings are loaded in the page. You see the Edit Launch Configuration window with all the fields populated. Click **Save**, as shown in the following figure.



Edit Launch Configuration

Launch Config Name*: vy301-bf1-nodesample

Target*: Dallas (Production)

Organization*: student103@yahoo.com

Space*: dev

Manifest File: manifest.yml

Manifest Settings

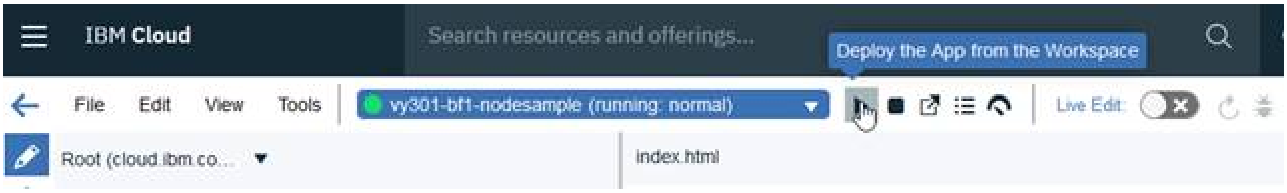
Application Name*: vy301-bf1-nodesample

Host: student103yahoocom-48093ae05cfd415l

Domain*: mybluemix.net

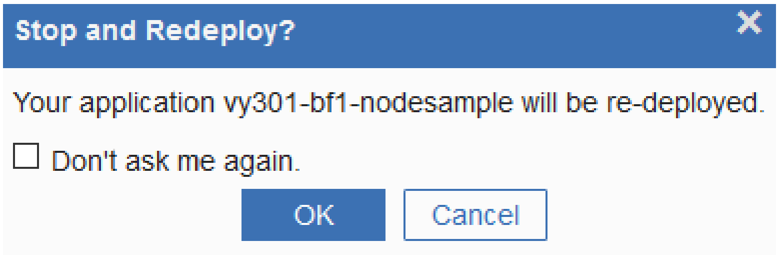
Shaded boxes indicate modified fields that will override manifest file settings.
* Denotes required field.

__ c. In the application status toolbar, click the play icon that has the tooltip **Deploy the App from the Workspace** when you hover your cursor over the icon, as shown in the following figure.



IBM Cloud
Search resources and offerings...
Deploy the App from the Workspace
File
Edit
View
Tools
vy301-bf1-nodesample (running: normal)
Live Edit:
Root (cloud.ibm.co...
index.html

__ d. You are prompted to confirm that you want to stop and redeploy the application `vy301-xxx-nodesample`. Click **OK**, as shown in the following figure.

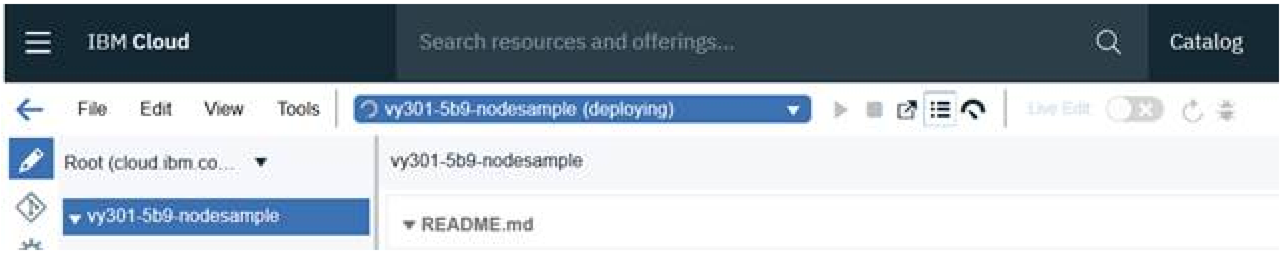


Stop and Redeploy?

Your application vy301-bf1-nodesample will be re-deployed.

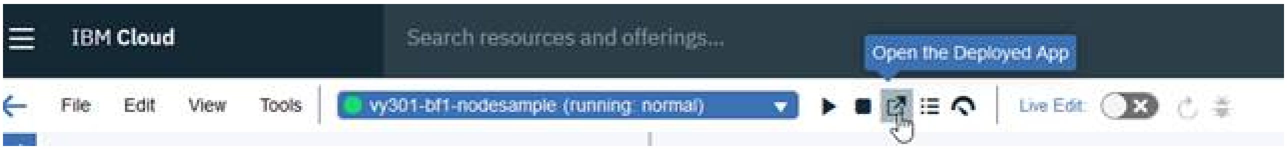
☐ Don't ask me again.

__ e. Wait until the application is stopped, deployed, and started, as shown in the following figure.



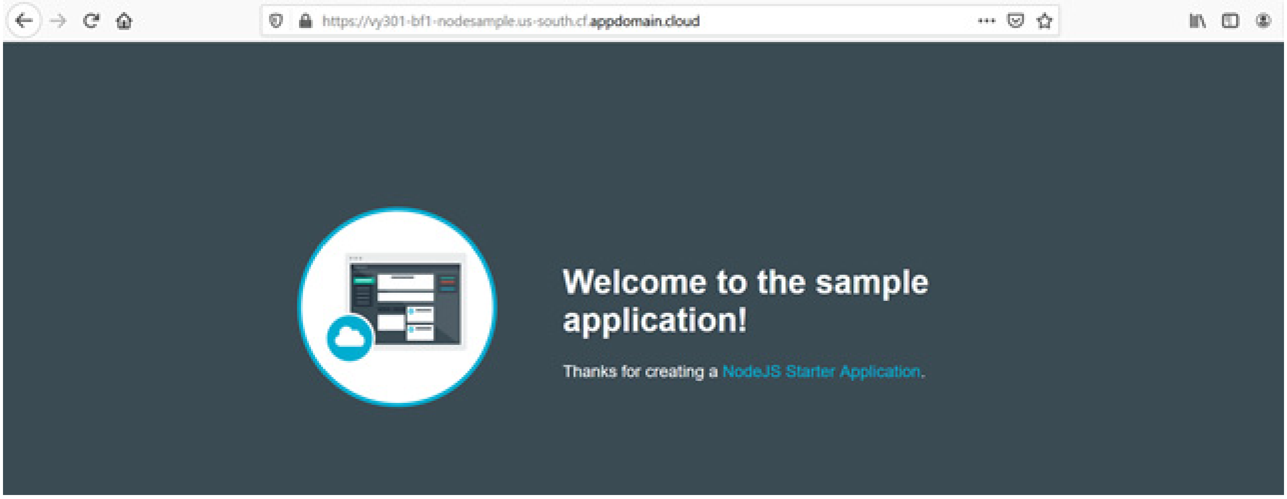
__ 3. Confirm that the changes appear in the application:

__ a. In the server toolbar, click **Open the Deployed App** to view the changes to your application, as shown in the following figure.



IBM Cloud
Search resources and offerings...
Open the Deployed App
File
Edit
View
Tools
vy301-bf1-nodesample (running: normal)
Live Edit

__ b. Verify that the updated heading appears. The “Welcome to the sample application!” message in the sample application web page is displayed as shown in the following figure.



https://vy301-bf1-nodesample.us-south.cf.appdomain.cloud

Welcome to the sample application!

Thanks for creating a NodeJS Starter Application.

Part 10: Automatically push changes to IBM Cloud (optional)

Note

This part cannot be performed with an IBM Cloud Lite account. You need a Pay-As-You-Go account or a Subscription account. If you are a university student, you can apply for a no-charge feature code. Check with your instructor.

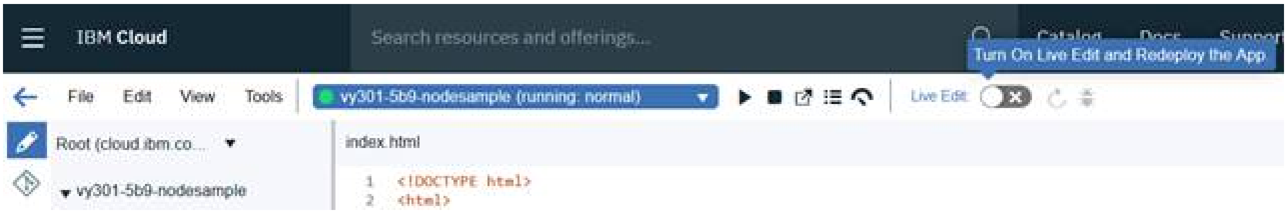
With the stop and redeploy option, you must first manually deploy the changes and trigger an application restart through the server toolbar. Although this option is more convenient than having to commit your changes to the Git repository, it can be disruptive to your software development workflow for minor changes.

The *IBM Cloud Live Sync* feature automatically pushes any saved changes in the project workspace to a linked IBM Cloud application. For IBM SDK for Node.js applications, you can update static files and view the updates on IBM Cloud without restarting your application. Alternatively, for the non-static Node.js code, such as JavaScript files, you need to restart only your application without having to deploy the changes.

To automatically push changes to IBM Cloud, complete the following steps:

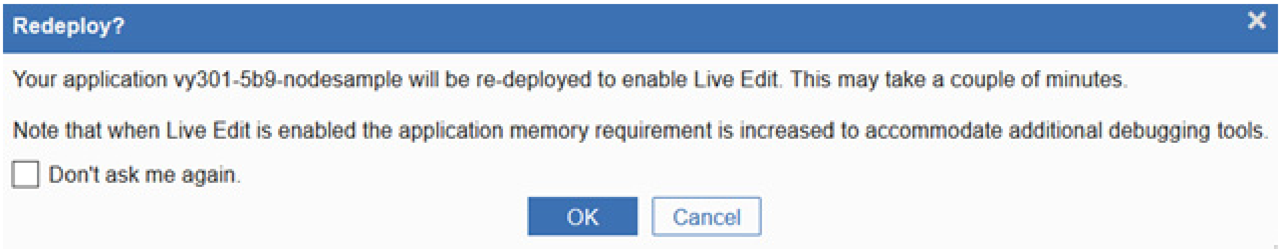
__ 1. Return to the Eclipse Orion Web IDE web page and enable the Live Edit feature:

__ a. Near the server’s toolbar, click the switch icon next to **Live Edit**, as shown in the following figure.


IBM Cloud
Search resources and offerings...
Catalog
Docs
Support
Turn On Live Edit and Redeploy the App
File
Edit
View
Tools
vy301-5b9-nodesample (running: normal)
Live Edit
Root (cloud.ibm.co...
vy301-5b9-nodesample
index.html

```
<!DOCTYPE html>
<html>
```

__ b. Click **OK** to redeploy the application and enable Live Edit mode, as shown in the following figure. Enabling Live Edit mode allocates more memory to the application to enable debugging.



Redeploy?

Your application vy301-5b9-nodesample will be re-deployed to enable Live Edit. This may take a couple of minutes.

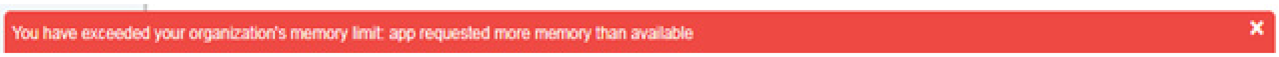
Note that when Live Edit is enabled the application memory requirement is increased to accommodate additional debugging tools.

☐ Don't ask me again.

OK Cancel

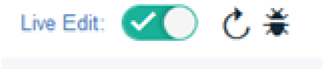
Information

The quota for the IBM Cloud Lite account is 256 MB, and applications that have Live Edit enabled require 800+ MB on average; this is the reason why this part of the exercise cannot be performed with an IBM Cloud Lite account. Here is the error message that you receive if you are using an IBM Cloud Lite account.



You have exceeded your organization's memory limit: app requested more memory than available

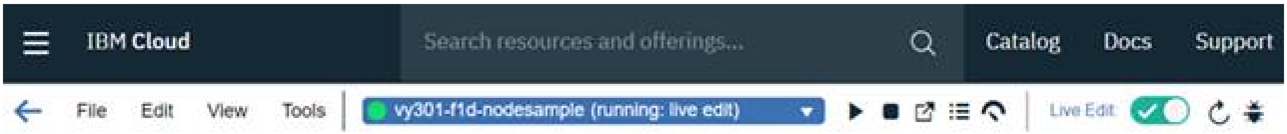
__ c. Confirm that Live Edit mode is enabled, as shown in the following figure.



Note

If you have a synchronization error, restart the application.

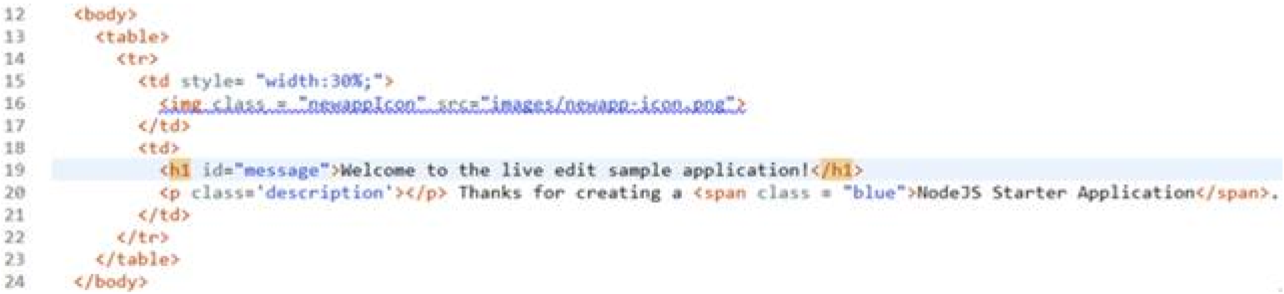
__ d. Wait until the application is running in Live Edit mode, as shown in the following figure. It might take a few minutes.



vy301-f1d-nodesample (running: live edit)

__ 2. Change the sample application web page heading:

__ a. On the `index.html` web page, edit the heading to read `Welcome to the live edit sample application!`, as shown in the following figure.



```
<body>
  <table>
    <tr>
      <td style= "width:30%;">
        <img class = "newappIcon" src="images/newapp-icon.png">
      </td>
      <td>
        <h1 id="message">Welcome to the live edit sample application!</h1>
        <p class='description'></p> Thanks for creating a <span class = "blue">NodeJS Starter Application</span>.
      </td>
    </tr>
  </table>
</body>
```

Note

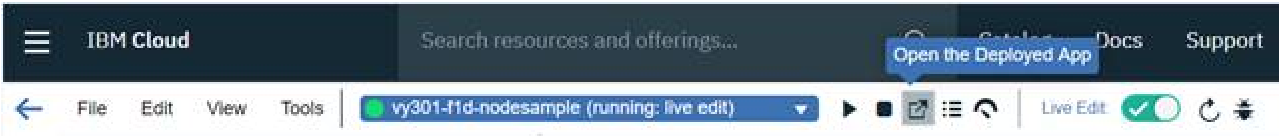
With the Live Edit feature, you do not need to save or deploy your static files. Changes are saved and deployed to your IBM Cloud account automatically.

Examples of static files include HTML web pages and Cascading Style Sheets (CSS).

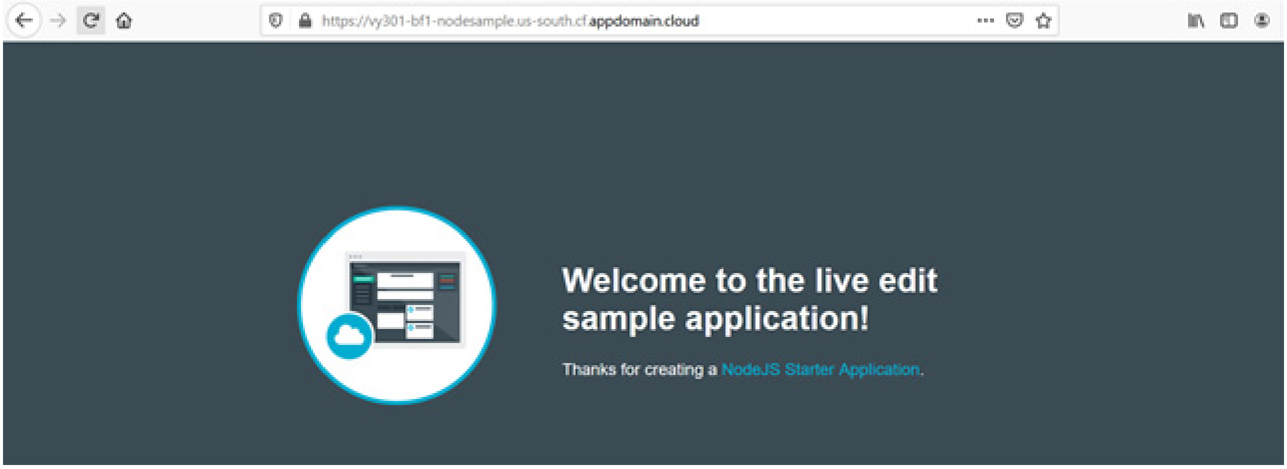
However, you might still need to commit your files in Git if you want other people to see them.

__ 3. Confirm that the changes appear in your IBM Cloud application:

__ a. In the server toolbar, click the **Open the Deployed App** icon to view the changes to your application, as shown in the following figure.


IBM Cloud
Search resources and offerings...
Catalog
Docs
Support
Open the Deployed App
File
Edit
View
Tools
vy301-f1d-nodesample (running: live edit)
Live Edit

Verify that the updated heading appears in the sample application web page, as shown in the following figure.



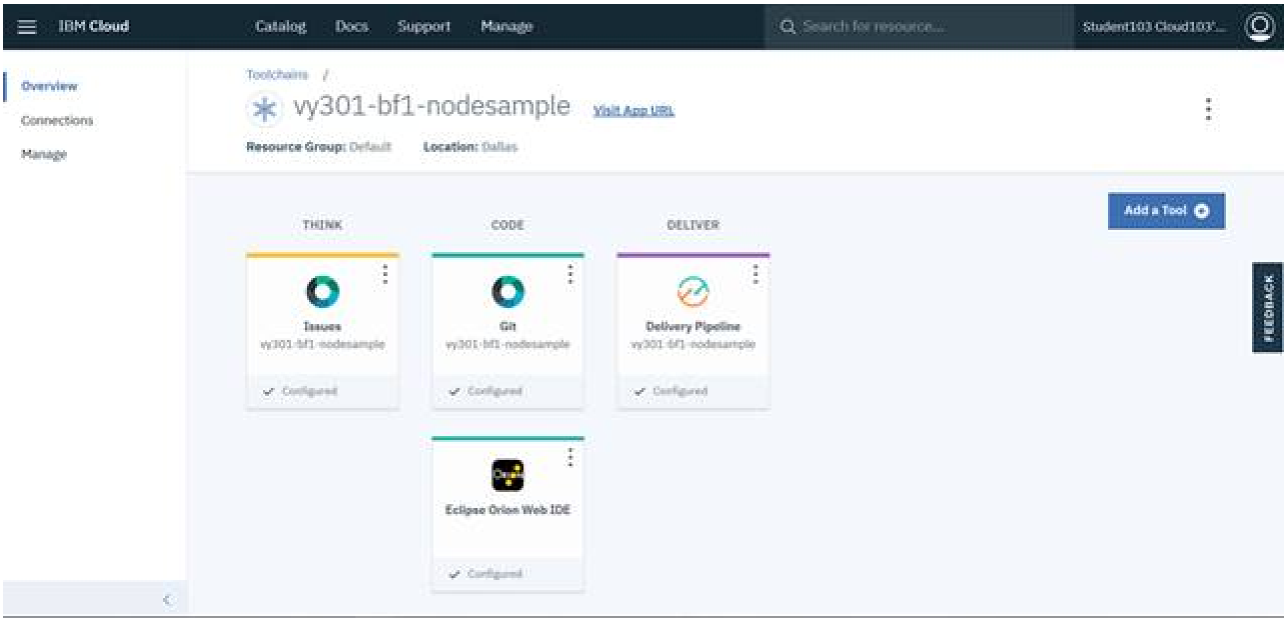
Welcome to the live edit sample application!

Thanks for creating a NodeJS Starter Application.

Part 11: Cleaning up the application and Continuous Delivery service and tools

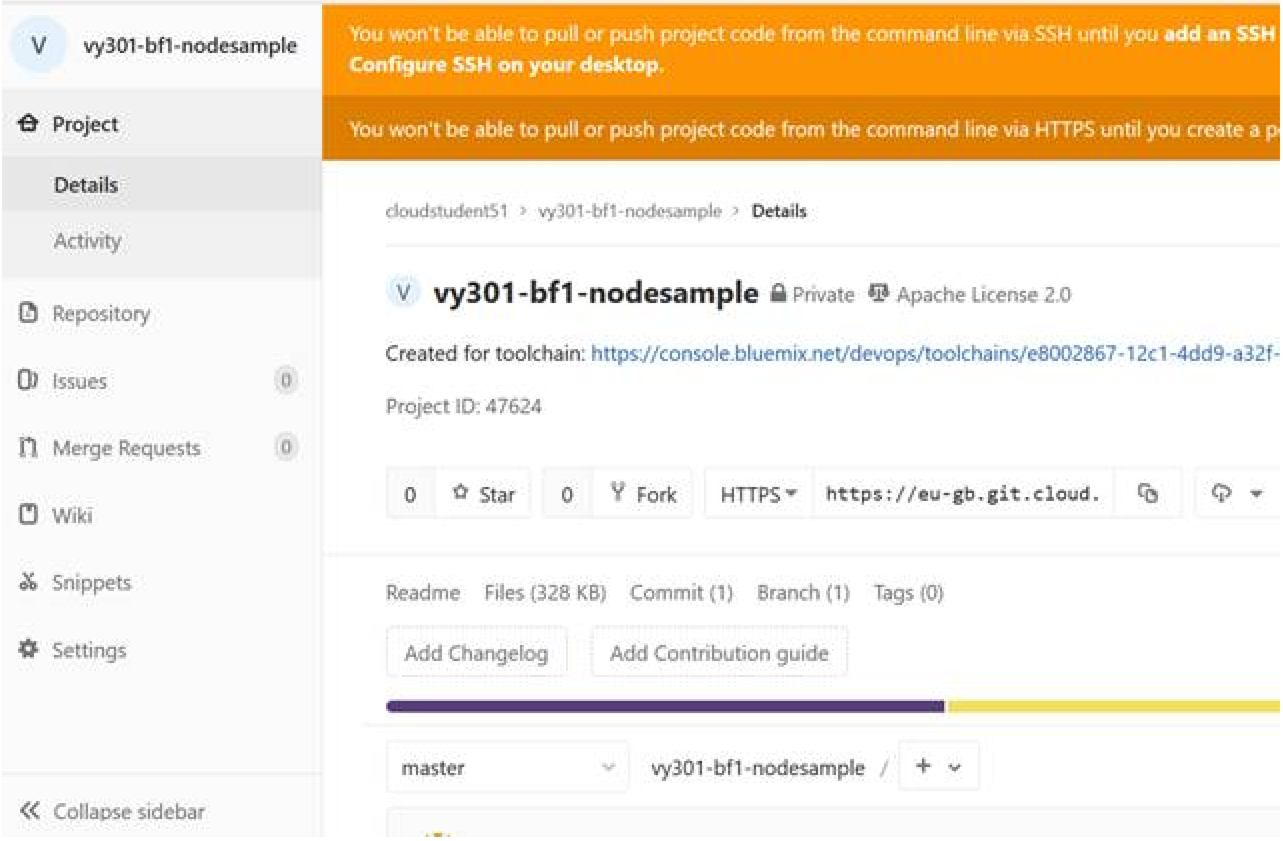
In this part, you clean up the application and the Continuous Delivery Toolchains dashboard by completing the following steps:

__ 1. Delete the Git repository. Go to the Toolchains dashboard and click **Git**.



Resource Group: Default Location: Dallas

__ 2. Click **Settings** on the left tab.



You won't be able to pull or push project code from the command line via SSH until you **add an SSH**
Configure SSH on your desktop.

You won't be able to pull or push project code from the command line via HTTPS until you create a p

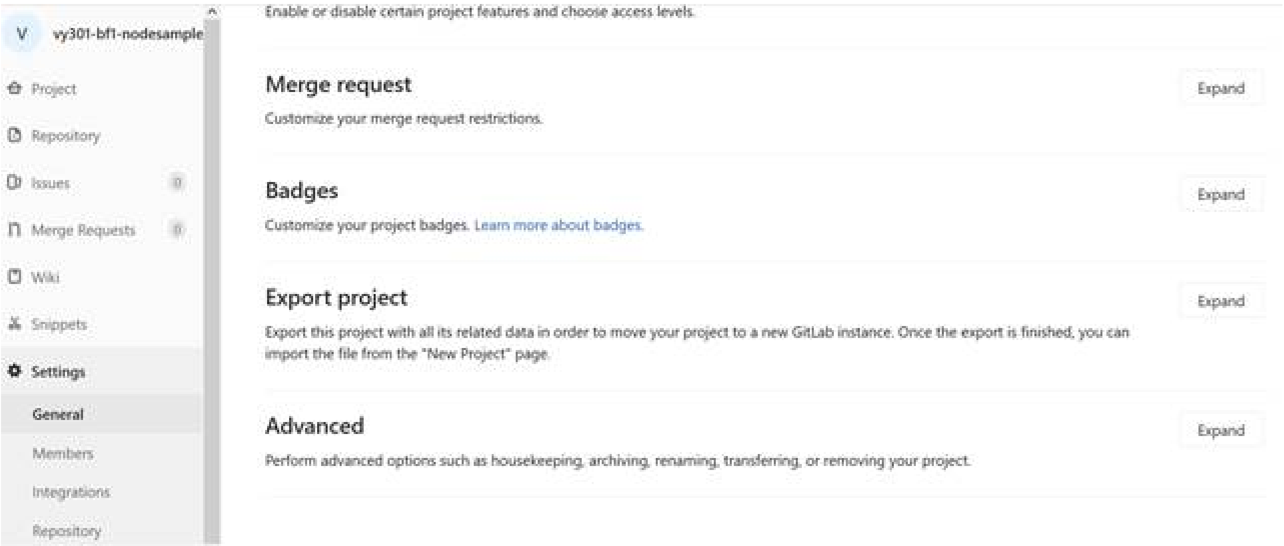
V **vy301-bf1-nodesample** Private Apache License 2.0

Created for toolchain: https://console.bluemix.net/devops/toolchains/e8002867-12c1-4dd9-a32f-

Project ID: 47624

`https://eu-gb.git.cloud.`

__ 3. Scroll-down and click **Expand** in the Advanced section.



Enable or disable certain project features and choose access levels.

Merge request

Customize your merge request restrictions.

Badges

Customize your project badges. Learn more about badges.

Export project

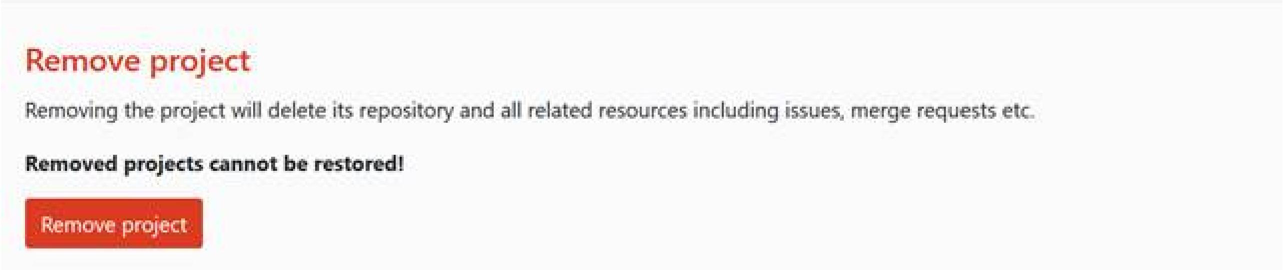
Export this project with all its related data in order to move your project to a new GitLab instance. Once the export is finished, you can import the file from the "New Project" page.

Advanced

Perform advanced options such as housekeeping, archiving, renaming, transferring, or removing your project.

__ 4. Select **Remove project**.

__ 5. Click **Remove project** as shown in the following figure.

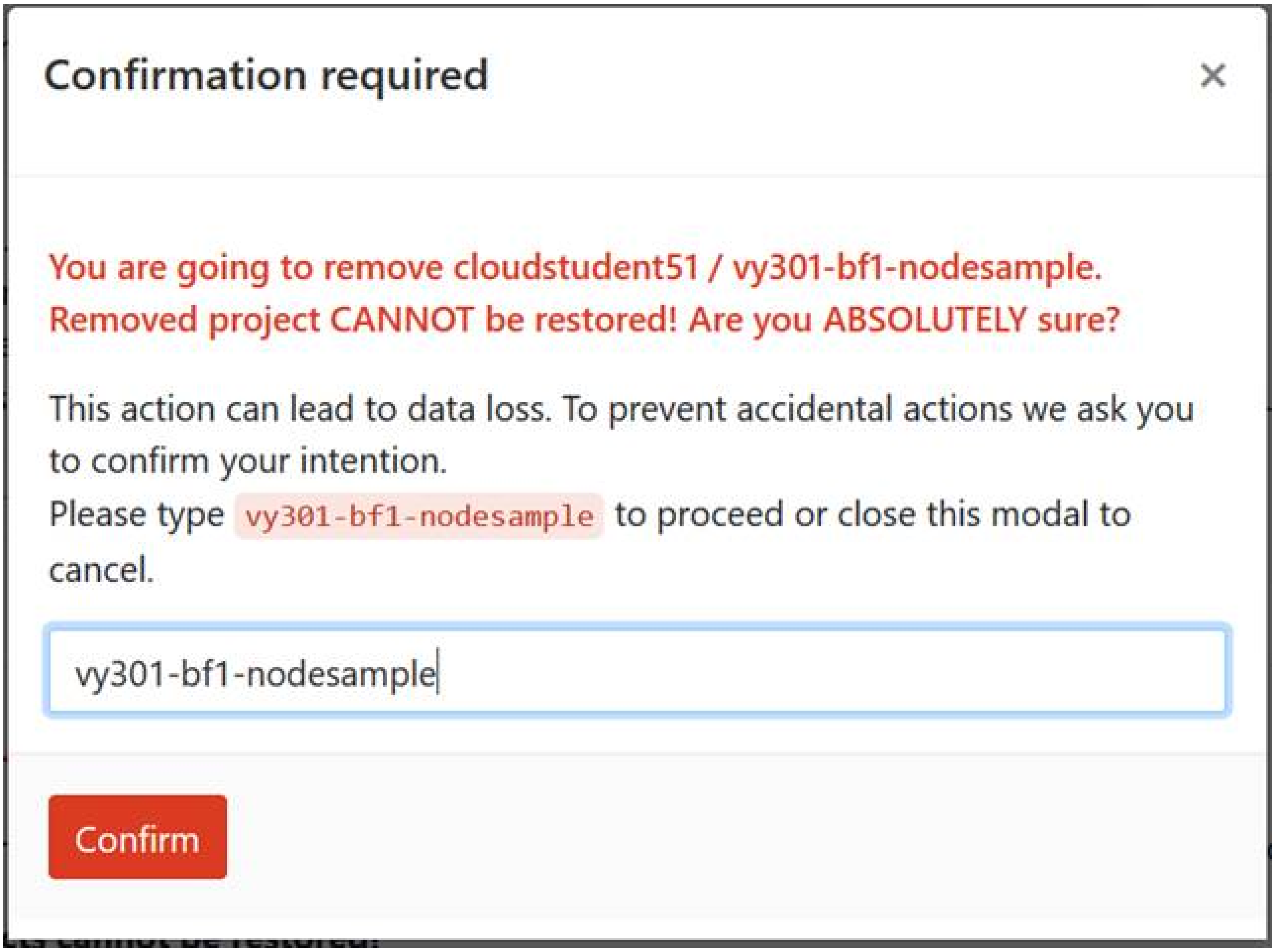


Remove project

Removing the project will delete its repository and all related resources including issues, merge requests etc.

Removed projects cannot be restored!

__ 6. In the confirmation window, enter your application name and click **Confirm** as shown in the following figure.



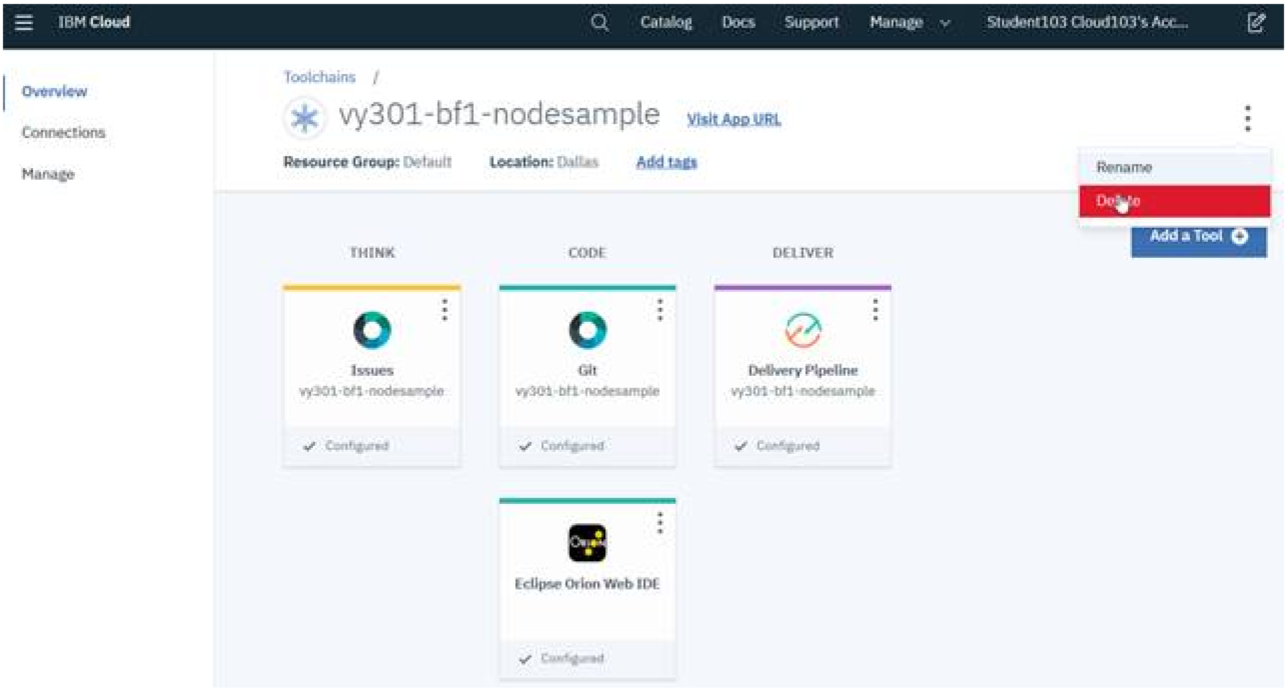
Confirmation required

You are going to remove cloudstudent51 / vy301-bf1-nodesample. Removed project CANNOT be restored! Are you ABSOLUTELY sure?

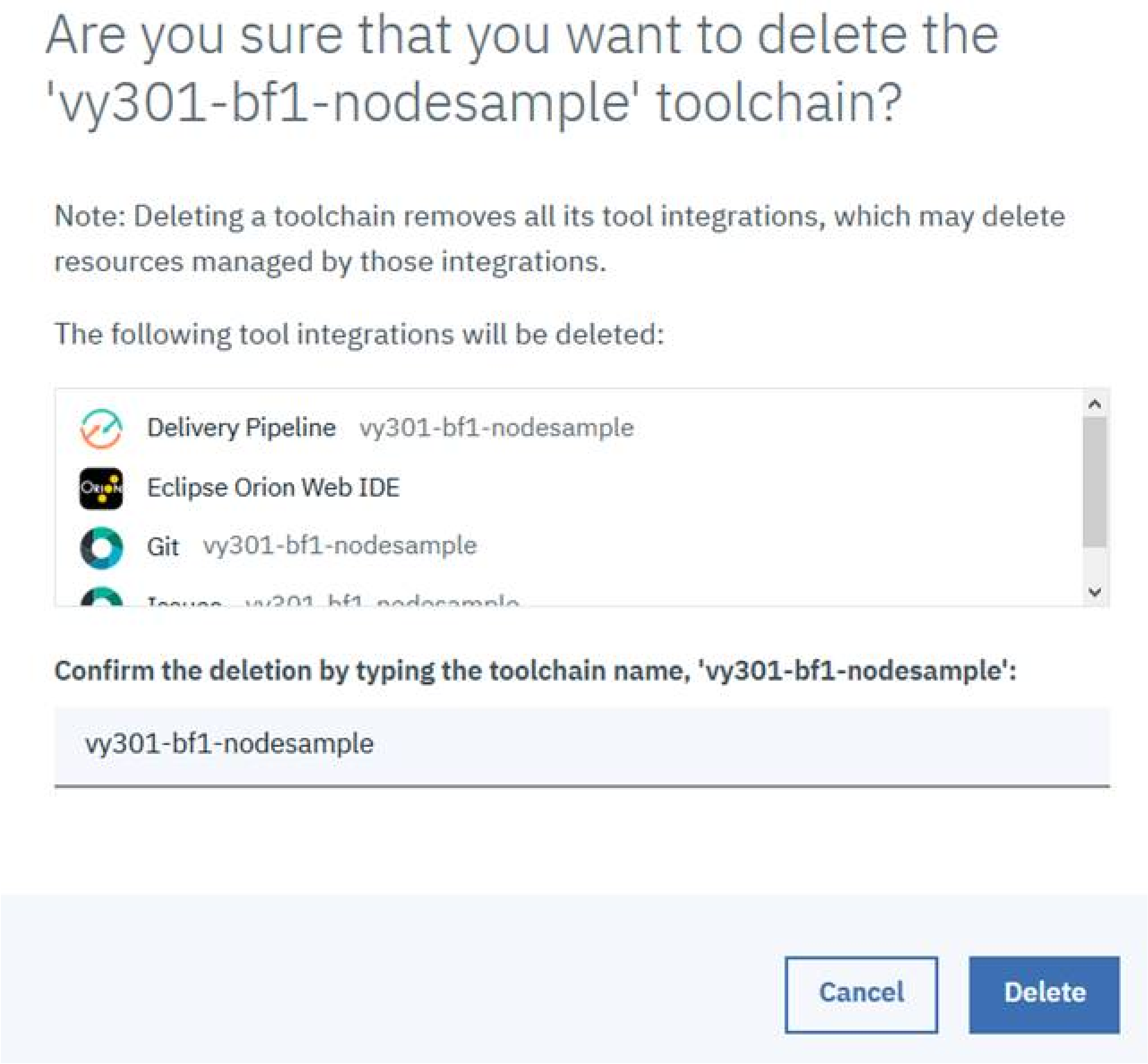
This action can lead to data loss. To prevent accidental actions we ask you to confirm your intention.
Please type `vy301-bf1-nodesample` to proceed or close this modal to cancel.

vy301-bf1-nodesample

__ 7. Delete the toolchain. Open the Continuous Delivery Toolchains dashboard. Click the **menu** icon and click **Delete**, as shown in the following figure.



__ 8. Type your application name (`vy301-xxx-nodesample`) in the confirmation window and click **Delete**, as shown in the following figure.



Are you sure that you want to delete the 'vy301-bf1-nodesample' toolchain?

Note: Deleting a toolchain removes all its tool integrations, which may delete resources managed by those integrations.

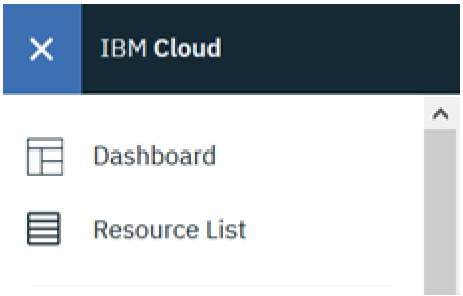
The following tool integrations will be deleted:

Delivery Pipeline vy301-bf1-nodesample
Eclipse Orion Web IDE
Git vy301-bf1-nodesample
Issues vy301-bf1-nodesample

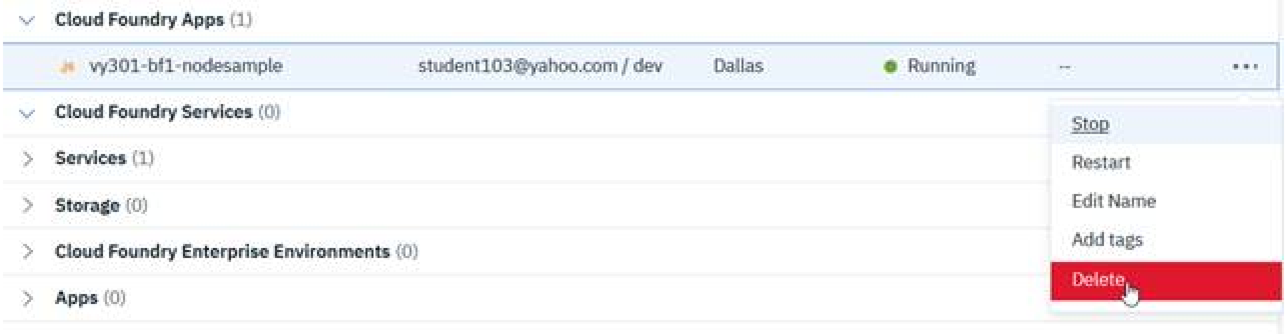
Confirm the deletion by typing the toolchain name, 'vy301-bf1-nodesample':

vy301-bf1-nodesample

__ 9. Click the **Navigation Menu** icon at the upper left and click **Resource List**, as shown in the following figure.



__ 10. In the Resource List window, expand Cloud Foundry Apps, click the **Actions** menu (three dots) for your application, and select **Delete**, as shown in the following figure.



Cloud Foundry Apps (1)				
vy301-bf1-nodesample	student103@yahoo.com / dev	Dallas	● Running	--

Cloud Foundry Services (0)

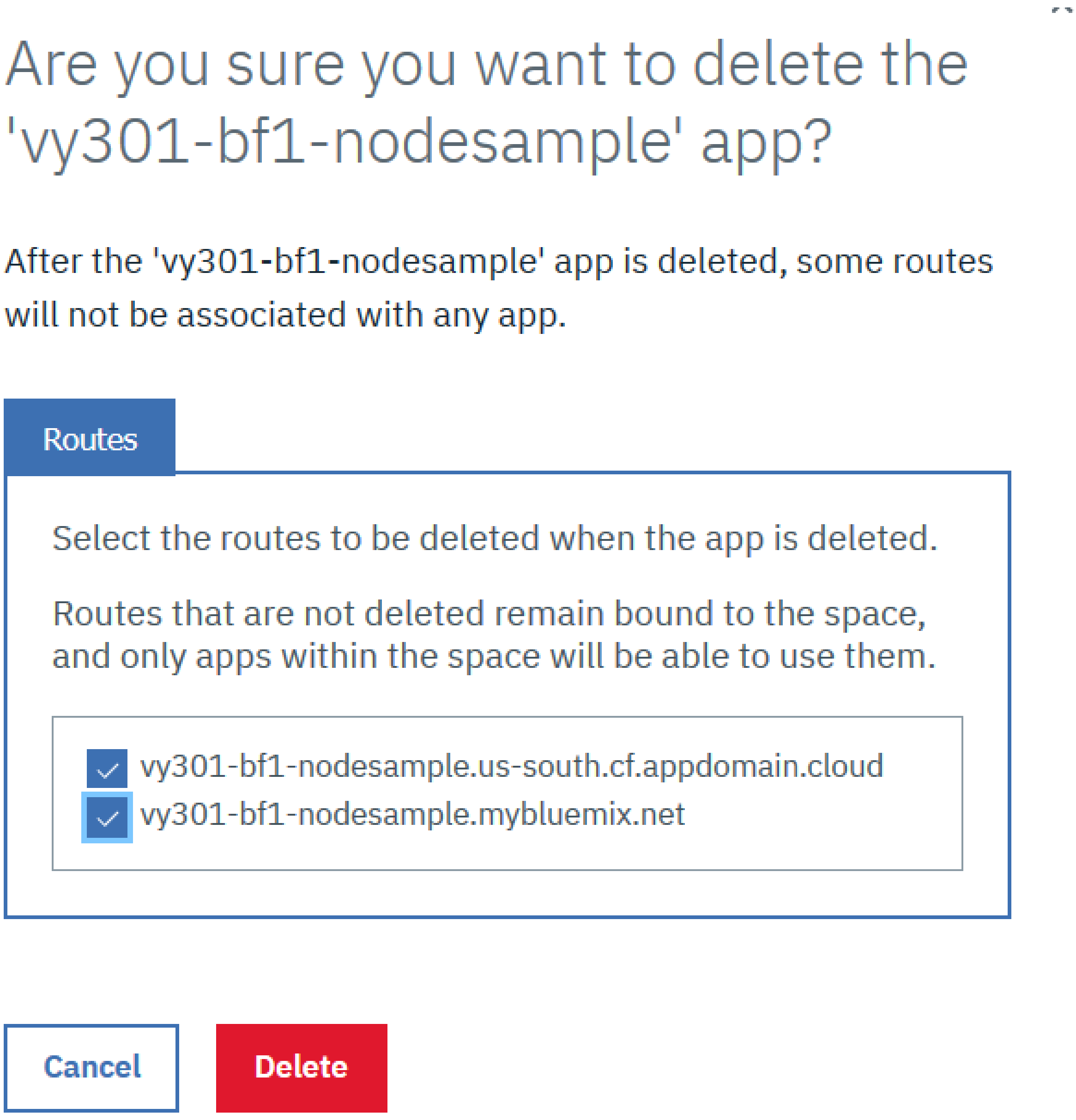
Services (1)

Storage (0)

Cloud Foundry Enterprise Environments (0)

Apps (0)

__ 11. Confirm that you want to delete the routes and click **Delete**, as shown in the following figure.



Are you sure you want to delete the 'vy301-bf1-nodesample' app?

After the 'vy301-bf1-nodesample' app is deleted, some routes will not be associated with any app.

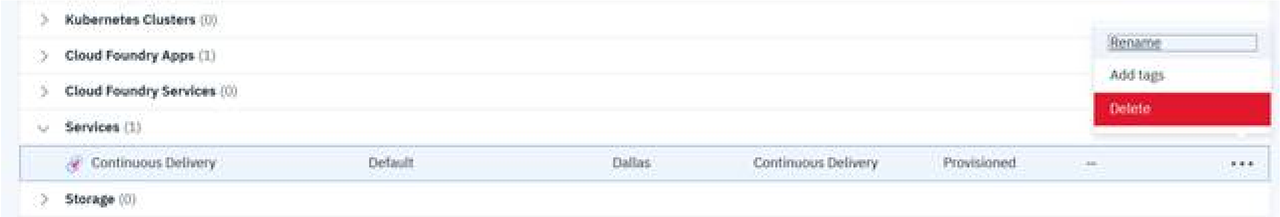
Routes

Select the routes to be deleted when the app is deleted.

Routes that are not deleted remain bound to the space, and only apps within the space will be able to use them.

☑ vy301-bf1-nodesample.us-south.cf.appdomain.cloud
☑ vy301-bf1-nodesample.mybluemix.net

__ 12. In the Resource List window, expand **Services** and delete the Continuous Delivery service. Click the **Actions** menu to the right of the service and select **Delete**, as shown in the following figure.



Kubernetes Clusters (0)

Cloud Foundry Apps (1)

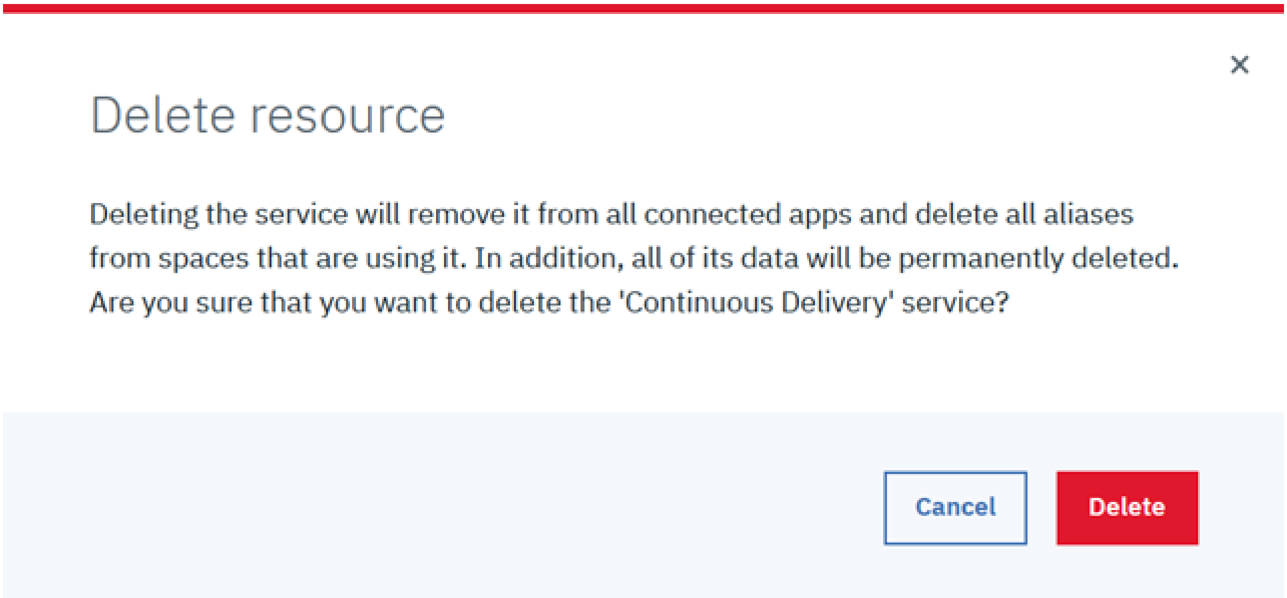
Cloud Foundry Services (0)

Services (1)

Continuous Delivery	Default	Dallas	Continuous Delivery	Provisioned	--

Storage (0)

__ 13. Confirm the deletion of the resource, as shown in the following figure.



Delete resource

Deleting the service will remove it from all connected apps and delete all aliases from spaces that are using it. In addition, all of its data will be permanently deleted. Are you sure that you want to delete the 'Continuous Delivery' service?

__ 14. Log out of IBM Cloud.

__ 15. Close your web browser.

End of exercise

Exercise review and wrap-up

In this exercise, you used IBM Cloud Continuous Delivery to manage your IBM Cloud application that is written for the IBM SDK for Node.js server runtime.

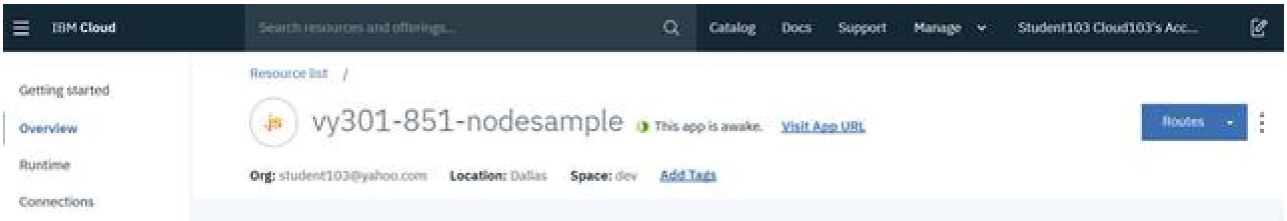
Then, you saved your changes in the Git repository. Through the Delivery Pipeline, you automatically deployed the committed source code changes to your IBM Cloud application.

In the last part of the exercise, you deployed your changes directly from the project workspace. Optionally, you also used the IBM Cloud Live Sync feature to push changes to static files without redeploying the application and without having to restart it.

Troubleshooting

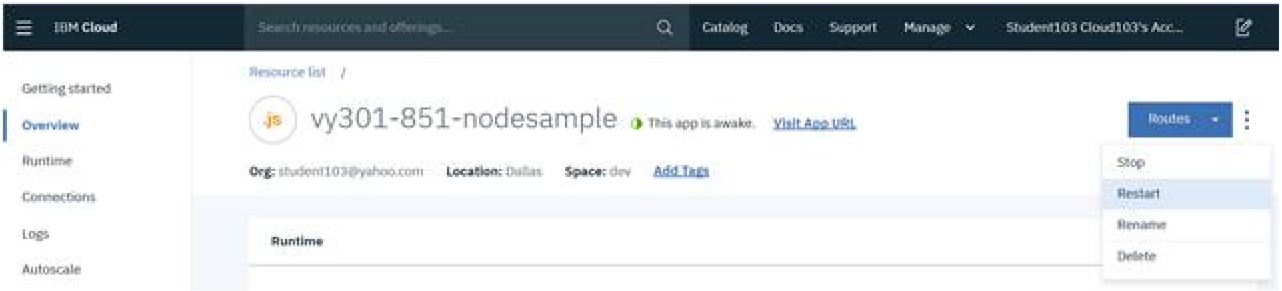
This section lists common problems that students might encounter while you perform this exercise.

- If your app status is “unknown” or your app status is “not running”, click the **Overview** link in the upper left of the window, as shown in the following figure.



Org: student103@yahoo.com Location: Dallas Space: dev Add Tags

Next, click **Restart** at the upper right options, as shown in the following figure to restart your app.



Org: student103@yahoo.com Location: Dallas Space: dev Add Tags

You can also refresh the page to force your browser to fetch the status of your app.