

## Course Exercises Guide

# Essentials of Cloud Application Development

**IBM MEA Skills Academy**

Course code SACAD ERC 3.0

Ahmed Azraq

Haytham Elkhoja

Surbhi Agarwal

Vikash Gupta

Hala Aziz

Nohran Khaled

Sukumar Beri

Marcelo Mota Manhaes

Deepak Chandrashekhar

Cong Nguyen

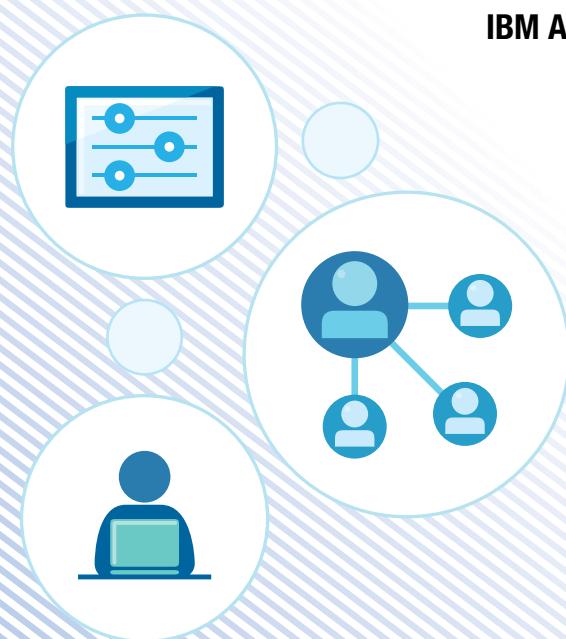
Mihai Criveti

JungHyeon Yoo

Karim El-Bawab

Acknowledgments to members of

**IBM Academy of Technology**



## July 2019 edition

### Notices

This information was developed for products and services offered in the US.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not grant you any license to these patents. You can send license inquiries, in writing, to:

*IBM Director of Licensing  
IBM Corporation  
North Castle Drive, MD-NC119  
Armonk, NY 10504-1785  
United States of America*

INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some jurisdictions do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM websites are provided for convenience only and do not in any manner serve as an endorsement of those websites. The materials at those websites are not part of the materials for this IBM product and use of those websites is at your own risk.

IBM may use or distribute any of the information you provide in any way it believes appropriate without incurring any obligation to you.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to actual people or business enterprises is entirely coincidental.

### Trademarks

IBM, the IBM logo, and ibm.com are trademarks or registered trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the web at "Copyright and trademark information" at [www.ibm.com/legal/copytrade.shtml](http://www.ibm.com/legal/copytrade.shtml).

**© Copyright International Business Machines Corporation 2016, 2019.**

**This document may not be reproduced in whole or in part without the prior written permission of IBM.**

US Government Users Restricted Rights - Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

# Contents

<b>Trademarks .....</b>	<b>v</b>
<b>Exercises description .....</b>	<b>vi</b>
<b>Exercise 1. Getting started with Cloud Foundry apps on IBM Cloud .....</b>	<b>1-1</b>
Part 1: Obtaining your randomly generated key .....	1-3
Part 2: Logging in to IBM Cloud .....	1-4
Part 3: Creating an application .....	1-5
Part 4: Modifying and redeploying your Cloud Foundry app .....	1-11
Part 5: Checking your organization limits .....	1-27
Part 6: Deleting the sample application .....	1-29
<b>Exercise 2. Developing IBM Cloud applications with IBM Cloud Continuous Delivery .....</b>	<b>2-1</b>
Part 1: Re-creating your application .....	2-3
Part 2: Examining the IBM Cloud application .....	2-4
Part 3: Enabling Continuous Delivery for the starter application .....	2-6
Part 4: Reviewing your application in Eclipse Orion web IDE .....	2-11
Part 5: Inviting collaborators .....	2-15
Part 6: Editing the sample application .....	2-19
Part 7: Committing your changes to the Git repository .....	2-24
Part 8: Deploying the application from the Git repository to IBM Cloud .....	2-29
Part 9: Deploying the application directly from your project workspace to IBM Cloud .....	2-33
Part 10: Automatically push changes to IBM Cloud (optional) .....	2-38
Part 11: Cleaning up the application and Continuous Delivery service and tools .....	2-40
<b>Exercise 3. IBM Cloud with IBM Cloudant .....</b>	<b>3-1</b>
Requirements .....	2
Part 1: Logging in to IBM Cloud .....	3-3
Part 2: Creating an instance of the Cloudant service .....	3-4
Part 3: Exploring the Cloudant service on IBM Cloud .....	3-8
Part 4: Creating credentials for a Cloudant service instance .....	3-15
Part 5: Exploring the features of the Cloudant dashboard .....	3-18
Part 6: Using an HTTP API client to access Cloudant .....	3-30
Part 7: Deleting your Cloudant service instance .....	3-48
<b>Exercise 4. Securing a web application with single sign-on (optional) .....</b>	<b>4-1</b>
Part 1: Creating an App ID service instance .....	4-3
Part 2: Exploring the App ID service .....	4-4
Part 3: Retrieving the App ID service credentials .....	4-8
Part 4: Creating a Node.js app and connecting it to the App ID service .....	4-10
Part 5: Configuring the application to integrate with the App ID service .....	4-15
Part 6: Securing the sample application with the App ID service .....	4-20
Part 7: Cleaning up the environment .....	4-24
<b>Exercise 5. Managing IBM Kubernetes Service clusters .....</b>	<b>5-1</b>
Part 1: Creating an IBM Cloud Kubernetes Service cluster .....	5-2
Part 2: Connecting to your cluster on IBM Cloud Kubernetes Service .....	5-7
Part 3: Listing the worker nodes in your cluster .....	5-11
<b>Exercise 6. Deploying an application on Kubernetes .....</b>	<b>6-1</b>

Part 1: Logging in to your IBM Cloud account, connecting to your cluster, and viewing the number of worker nodes .....	6-2
Part 2: Creating a container and building it on IBM Cloud Container Registry .....	6-2
Part 3: Creating a deployment and scaling it .....	6-8
Part 4: Exposing the app over the internet .....	6-13
Part 5: Cleaning up the environment .....	6-16

---

# Trademarks

The reader should recognize that the following terms, which appear in the content of this training document, are official trademarks of IBM or other companies:

IBM, the IBM logo, and ibm.com are trademarks or registered trademarks of International Business Machines Corp., registered in many jurisdictions worldwide.

The following are trademarks of International Business Machines Corporation, registered in many jurisdictions worldwide:

Bluemix®  
SPSS®

Cognitive Era™  
Watson Avatar®

Cognos®  
Worklight®

Windows is a trademark of Microsoft Corporation in the United States, other countries, or both.

Linux is a trademark of Linus Torvalds in the United States, other countries, or both.

Other product and service names might be trademarks of IBM or other companies.

---

# Exercises description

This course includes the following exercises:

- Exercise 1. Getting started with Cloud Foundry apps on IBM Cloud  
This exercise describes how you can deploy a web app without downloading or configuring a runtime environment, a framework, or setting up a server. This exercise also covers how to test and run the app when it is deployed.
- Exercise 2. Developing IBM Cloud applications with IBM Cloud Continuous Delivery  
In this exercise, you will explore DevOps capabilities in IBM Cloud. You work with the IBM Cloud Continuous Delivery services to explore, develop, build, and deploy IBM Cloud applications.
- Exercise 3. IBM Cloud with Cloudant  
This exercise demonstrates how you can create a Cloudant database service on IBM Cloud without installing or configuring the database instance on your workstation. You use an HTTP API client such as Postman to create, read, update, and delete Cloudant documents. You create indexes and query data by using Cloudant API end points.
- Exercise 4. Securing a web application with single sign-on (optional)  
In this exercise, you secure an application by using the App ID service for single sign-on by authenticating your application through trusted server providers.
- Exercise 5. Managing IBM Kubernetes Service clusters  
This exercise demonstrates how to create an IBM Kubernetes Service cluster and manage it by using the kubectl CLI.



## Important

It takes about 30 minutes to provision a cluster. Exercise 5, Part 1 should be started during the lecture or before a break.

- 
- Exercise 6. Deploying an application on Kubernetes  
In this exercise, you build a containerized application and deploy it to IBM Cloud Kubernetes Service.

In the exercise instructions, you can check off the line before each step as you complete it to track your progress.

All exercises include required sections, which should always be completed. It might be necessary to complete these sections before you can start later exercises. If you have sufficient time and want an extra challenge, some exercises might also include optional sections that you can complete.



### Important

You must complete *Exercise 0. Setting up your hands-on environment* which is part of Module I (self-study) before starting the exercises in this course.

---

---



### Information

Students in this course use an IBM Cloud Lite account to perform the exercises. This account will never expire, therefore students can continue working on the IBM Cloud environment, including the optional exercises, after the class.

---

---



### Hint

Before performing the exercises, check the status of the services in the IBM Cloud Status page at <https://cloud.ibm.com/status>

---

---

# Exercise 1. Getting started with Cloud Foundry apps on IBM Cloud

## Estimated time

01:00

## Overview

The focus of the application developers is on the business value of the application that they are creating. Developers should spend most of their time on the features, functions, and usability of their applications.

However, to realize the usefulness of their applications, developers often must deal with server management and installing and configuring the run time and framework that their application requires to run.

This exercise describes how you can deploy a web application (app) without downloading or configuring a runtime environment or framework or setting up a server. This exercise also covers how to test and run the app when it is deployed.

## Objectives

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

- Log in to IBM Cloud from a browser.
- Create an IBM Cloud application by using one of the available run times.
- Install the IBM Cloud command line interface (CLI).
- Sign on to IBM Cloud from the CLI.
- Deploy an application from a local workstation by using the IBM Cloud CLI.
- Test the application with its endpoint after the application is deployed and started.

## Introduction

With IBM Cloud, you focus on rapidly building compelling user experiences rather than worrying about managing the infrastructure that underlies your applications. This exercise shows how to get started with a web app in IBM Cloud. You download and modify a sample app without worrying about the server configuration. Finally, you learn how to deploy the changes to the app and see them in action.

## Requirements

This exercise requires the following prerequisites:

- IBM Cloud Account.
- IBM Cloud CLI that is installed on your workstation.

## Exercise instructions

In this exercise, you complete the following tasks:

- 1. Obtain your randomly generated key.
  - 2. Log in to IBM Cloud.
  - 3. Create an application.
  - 4. Modify and redeploy your Cloud Foundry app.
  - 5. Check your organization limits.
  - 6. Delete the sample application
- 



### Important

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

---

### **Part 1: Obtaining your randomly generated key**

The exercises require that you create several objects, each of which should have unique names. You use an online tool to generate a random key to ensure that the names of your objects are unique:

- 1. Go to <https://www.uuidgenerator.net/>. You see a string of 36 letters, numbers, and hyphens. The online UUID generator shows the Universally Unique Identifier (UUID). Write down the first three characters on a piece of paper. This is your randomly generated key.
- 2. The first three characters in the UUID are used in the naming convention for this exercise. For example, if the UUID that is returned by the UUID generator was 20644124-9e5c-4246-a098-993843099d71, then the key you should use in the object name is 206. Every time that you see xxx as part of the object name in the exercises, replace it with your key.

The screenshot shows the "Online UUID Generator" website. At the top, it displays "Your Version 4 UUID:" followed by the string "20644124-9e5c-4246-a098-993843099d71". To the right of the UUID is a "Copy" button with a clipboard icon. Below the UUID, there is a link "Refresh page to generate another."

## Part 2: Logging in to IBM Cloud

Log in to IBM Cloud by completing the following steps:

- \_\_ 1. Open the IBM Cloud console (<https://cloud.ibm.com>) in a web browser.
- \_\_ 2. In the “Log in to IBM Cloud” right pane, complete the following steps:
  - \_\_ a. Leave the “IBMid” selected from the drop-down list.
  - \_\_ b. Enter the email address that you used to register for IBM Cloud,
  - \_\_ c. Click **Continue**.
  - \_\_ d. Enter your password and click **Log in**.

## Log in to IBM Cloud

The screenshot shows the IBM Cloud login interface. At the top, there is a dropdown menu labeled "IBMid" with a downward arrow. Below it is a text input field containing a blurred email address. Underneath the input field is a checkbox labeled "Remember me". The next section is labeled "Password" with a text input field containing several dots and an "eye" icon to toggle visibility. At the bottom left, there are two blue links: "Forgot ID?" and "Forgot password?". On the right side, there is a circular "Log in" button with a magnifying glass icon and the text "Log in".

You are redirected to the Dashboard, as shown in the following figure.

The screenshot shows the IBM Cloud Dashboard. At the top, there's a navigation bar with links for Catalog, Docs, Support, Manage, and account information. Below the navigation is a "Dashboard" section with a "Customize" button. To the right are buttons for "Upgrade account" and "Create resource".

**Resource summary:** Includes links for "What is a resource?", "Regions for resource deployment", and "Best practices for organizing resources in resource group". It also has a "View resources" link, a "Create" button with a plus sign, and a "View a summary of resources in your account here" note.

**Planned maintenance:** Shows an event: "Next event: Tue, Nov 26, 2019 12:00 PM" with the note "Update the infrastructure to apply Import..." and a "View events" link.

**Location status:** A table showing regions with green checkmarks: Asia Pacific, Europe, North America, and South America. Each row has a "View status" link.

**For you:** A section with a "Catalog" button and a "Use Cloud Foundry" link. It also includes a note about starter kits: "Accelerate your cloud use with starter kits. View the most popular starter kits based on use case or language." and a "Starter kit" button with a "View available starter kits" link.

### Part 3: Creating an application

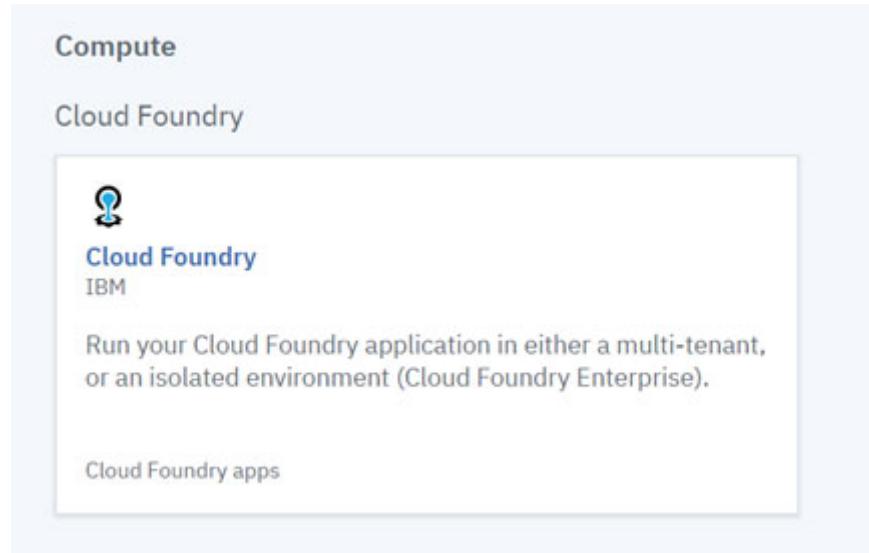
The IBM Cloud catalog lists components and services that help you build your application. In this part, you create an IBM Cloud application with the IBM software development kit (SDK) for the Node.js run time:

- \_\_\_ 1. Create an application with an instance of the IBM SDK for Node.js runtime environment:
  - \_\_\_ a. On the IBM Cloud Dashboard, click **Create resource** on the right, as shown in the following figure.

The screenshot shows the IBM Cloud Catalog page. At the top, there's a navigation bar with links for Catalog, Docs, Support, Manage, and account information. Below the navigation is a "Student20 Cloud20's Account" section with a "Upgrade account" button and a "Create resource" button.

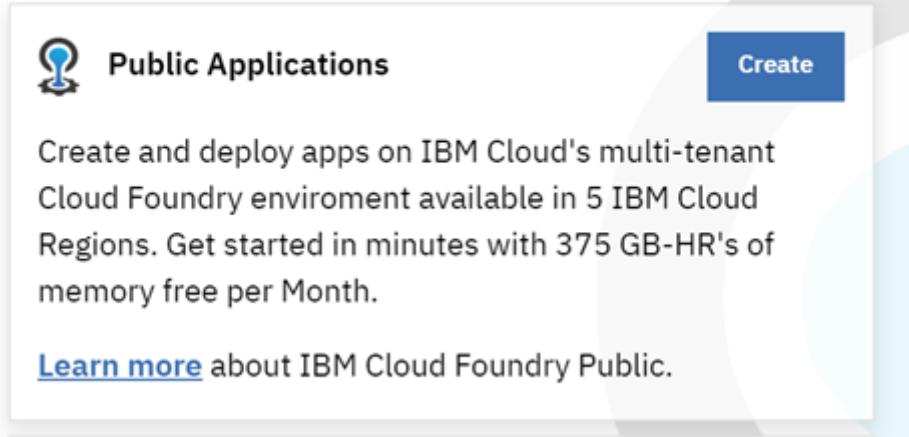
Below the account section is a search bar with the placeholder "Search the catalog". Underneath the search bar, there are category filters: "Compute", "Database", "Networking", "Storage", "Analytics", "Machine Learning", "Cloud Services", and "Marketplace".

- \_\_\_ b. You can now see the entire catalog. In the Search field, type “Cloud Foundry”.
- \_\_\_ c. Select **Cloud Foundry** under Compute, as shown in the following figure.



- \_\_\_ 2. From Cloud Foundry overview page, click **Create** under **Public Applications** as shown in the following figure.

## Cloud Foundry in the IBM Cloud



- \_\_\_ 3. Complete the details for creating a Cloud Foundry Sample App:
  - \_\_\_ a. Region is selected by default based on your location.  
In this example, the region is London.

The screenshot shows a web-based application for creating a Cloud Foundry sample app. At the top, there's a header with the title 'Create a Cloud Foundry Sample App' and tabs for 'Lite', 'IBM', and 'Application'. Below the header, it says 'Version: 3.x'. There are two tabs: 'Create' (which is selected) and 'About'. A dropdown menu labeled 'Select a region' has 'London' listed as the option.



## Troubleshooting

If you see the error [“Deployment in multiple regions is not available in Lite accounts. Upgrade your account to enable access to multiple regions.”](#), as shown in the following figure, click the drop-down menu under **Select a region** and switch to the region to which you have access. Repeat the step to create your app.

The screenshot shows the 'Create a Cloud Foundry Sample App' interface. The 'App name:' field contains 'vy301-205-nodesample'. The 'Host name:' field also contains 'vy301-205-nodesample'. Under 'Choose an organization:', it says 'No org available'. Under 'Choose a space:', it says 'No space available'. In the 'Tags:' section, there is an information icon and the text 'Examples: env:dev, version-1'. A red-bordered callout box at the bottom left contains the message: 'Deployment in multiple regions is not available in Lite accounts. [Upgrade your account](#) to enable access to multiple regions.'

- \_\_\_ b. If you do not have a Lite account, skip this step. For Lite accounts, Pricing Plans show that the memory that is allocated to your app by default is 64 MB. For this exercise, select the maximum allocation of **256 MB**, as shown in the following figure.

PLAN	FEATURES	PRICING
<input checked="" type="checkbox"/> Lite <input type="radio"/> 64 MB <input type="radio"/> 128 MB <input checked="" type="radio"/> 256 MB	<b>Lite apps are free</b> You get up to 256 MB of memory while you work on your apps.	Free

Lite apps sleep after 10 days of development inactivity.

- \_\_\_ c. Select **SDK for Node.js** from the provided runtimes as shown in the following figure.

Configure your resource

Select a runtime

<b>.java</b> Liberty for Java™ Version 3.x	<b>.js</b> SDK for Node.js™ Version 3.x	<b>.net</b> ASP.NET Core Version 2.x
<b>.go</b> Go Community	<b>.php</b> PHP Community	<b>.py</b> Python Community
<b>.rb</b> Ruby Community	<b>.swift</b> Runtime for Swift Version 1.0.0	<b>tomcat</b> Tomcat Community

- \_\_\_ d. Enter the app name. In the App name field, enter “vy301-xxx-nodesample”. Replace xxx with the first three characters of your randomly generated key, as shown in the following figure. For example, if the randomly generated key is 206, the app name is vy301-206-nodesample.
- \_\_\_ e. The host name is set by default to the app name, as shown in the following figure.
- \_\_\_ f. The domain is selected based on your location, which is London in this example.

**Note**

Ensure that the selected domain has the format {region}.cf.appdomain.cloud. Make sure **not** to select “mybluemix.net” as a domain because it is deprecated.

In this example, the location for the IBMid that was used to log in to IBM Cloud is London and the corresponding domain, which is selected automatically is eu-gb.cf.appdomain.cloud.

- \_\_\_ g. The organization is set by default to the email IBMid (email) that you used to log in.
- \_\_\_ h. The space is set by default to dev, as shown in the following figure.

App name:  
vy301-206-nodesample

Host name:  
vy301-206-nodesample

Domain:  
eu-gb.cf.appdomain.cloud

Choose an organization:  
ibmcloud-test@gmail.com

Choose a space:  
dev

Tags: 1  
Examples: env:dev, version:1

- \_\_\_ i. Click **Create**.

Summary

<b>Cloud Foundry App</b>	<a href="#">Estimate costs</a>
Region: London	
Plan: Default	
Runtime: SDK for Node.js™	
App name: vy301-206-nodesample	
Host name: vy301-206-nodesample	
Domain: eu-gb.cf.appdomain.cloud	
Org: ibmcloud-test@gmail.com	
Space: dev	

**Create**

Add to estimate

[View terms](#)

IBM Cloud proceeds to deploy your application. Your application stages and deploys in a few minutes.



### Stop

Wait until the application finishes staging and it is running in IBM Cloud before you proceed to the next step. For Lite accounts, wait for the application status “This app is awake”, as shown in the following figure.

**Getting started with SDK for Node.js**

Last Updated: 2019-11-07

Congratulations, you deployed a Hello World sample application on IBM Cloud™! To get started, follow this step-by-step guide. Or, [download the sample code](#) and explore on your own.

By following this tutorial, you'll set up a development environment, deploy an app locally on IBM Cloud™, and integrate IBM Cloud database service in your app.

Tip: Throughout these docs, references to the Cloud Foundry CLI are now updated to the IBM Cloud CLI! The IBM Cloud CLI has the same familiar Cloud Foundry commands, but with better integration with IBM Cloud accounts and other services. Learn more about getting started with the IBM Cloud CLI in this tutorial.

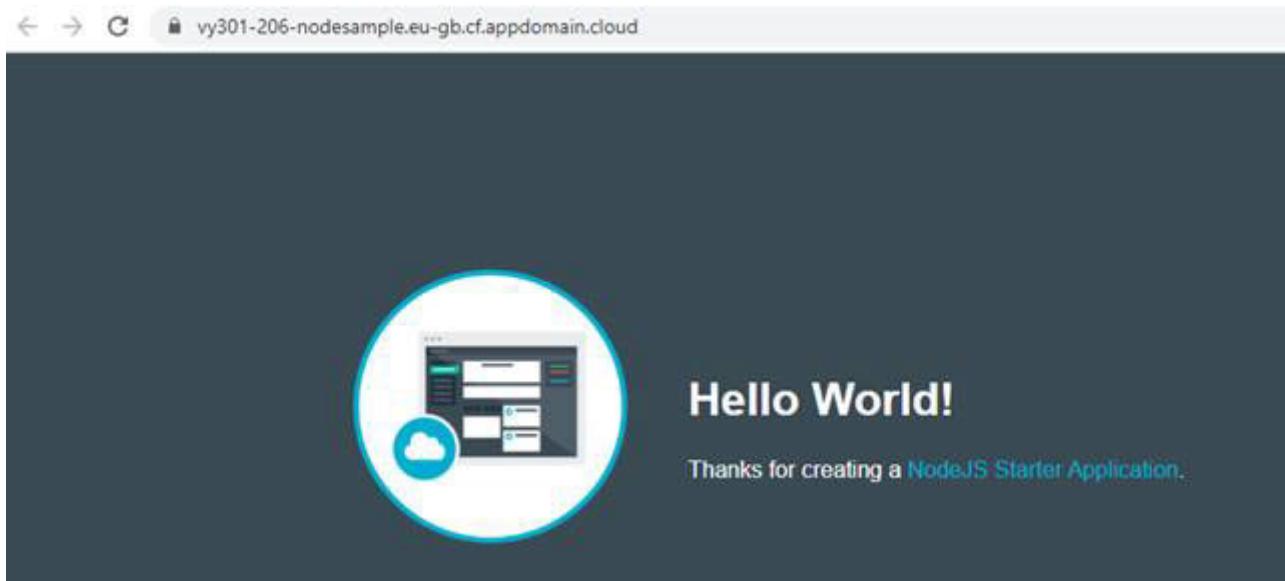
If you do not have a Lite account, the application status should be “Running”, as shown in the following figure.

Cloud Foundry apps /

**vy301-uis-nodesample** ● Running [Visit App URL](#) [Routes](#) ⚙ [Deploy](#) [More](#)

Org: student.skillsacademy@gmail.com Location: US South Space: dev

- \_\_\_ 4. Click **Visit App URL**, which opens a new browser tab with the URL for your app <https://vy301-xxxx-nodesample.{region}.cf.appdomain.cloud/>, where xxx is your randomly generated key. The domain is different according to your region/location; in this example, the domain is eu-gb.cf.appdomain.cloud.
  - \_\_\_ a. Clicking this link opens a new browser tab or page that shows your app. Confirm that the sample application appears, as shown in the following figure.



- \_\_\_ 5. Close the browser page for Hello World.

#### **Part 4: Modifying and redeploying your Cloud Foundry app**

Now that you successfully deployed your application, you modify the code and push the changes out to Cloud Foundry with IBM Cloud CLI by completing the following steps:

- \_\_\_ 1. Download and extract the sample application:
  - \_\_\_ a. Return to the **Application Details - IBM Cloud tab** on your browser.
  - \_\_\_ b. Make sure that **Getting started** is selected, as shown in the following figure.
  - \_\_\_ c. Click **download the sample code** under **Getting started with SDK for Node.js** as shown in the following figure.

The screenshot shows the IBM Cloud dashboard with the following details:

- Getting started** menu is open.
- The main page displays the app **vy301-206-nodesample**.
- Org: student.7440@gmail.com | Location: London | Space: dev | Add Tags
- Getting started with SDK for Node.js**
- Last Updated: 2019-11-07
- Congratulations, you deployed a Hello World sample application on IBM Cloud™! To get started, follow this step-by-step guide. Or, [download the sample code](#) and explore on your own.
- By following this tutorial, you'll set up a development environment, deploy an app locally on IBM Cloud™, and integrate an IBM Cloud database service in your app.
- Tip:** Throughout these docs, references to the Cloud Foundry CLI are now updated to the IBM Cloud CLI! The IBM Cloud CLI has the same familiar Cloud Foundry commands, but with better integration with IBM Cloud accounts and other services. Learn more about getting started with the IBM Cloud CLI in this tutorial.



### Note

If the hyperlink to download the sample code is not available, access the sample code directly from <https://github.com/IBM-Cloud/get-started-node>.

- \_\_\_ d. Download the sample code to your local workstation in the directory C:\IBM-Cloud.  
Click **Clone or download** and select **Download ZIP** as shown in the following figure

Sample and tutorial to help you get started with Express, REST API and a database. <https://cloud.ibm.com/docs/runtimes/n...>

The GitHub repository page for `kevin-ortega/IBM-Cloud/get-started-node` displays the following information:

- 78 commits
- 4 branches
- 0 packages
- 0 releases
- 11 contributors
- Apache-2.0 license
- Branch: master
- New pull request
- Find file
- Clone or download
- Clone with HTTPS
- Use Git or checkout with SVN using the web URL: <https://github.com/IBM-Cloud/get-started-node>
- Open in Desktop
- Download ZIP
- last year

The repository contents listed include:

- kevin-ortega Update package.json
- kubernetes Provide instructions for IBM Cloud Kubernetes Service
- views Provide instructions for IBM Cloud Kubernetes Service
- .cignore updated cignore
- .gitignore Update get-started app
- Dockerfile Provide instructions for IBM Cloud Kubernetes Service

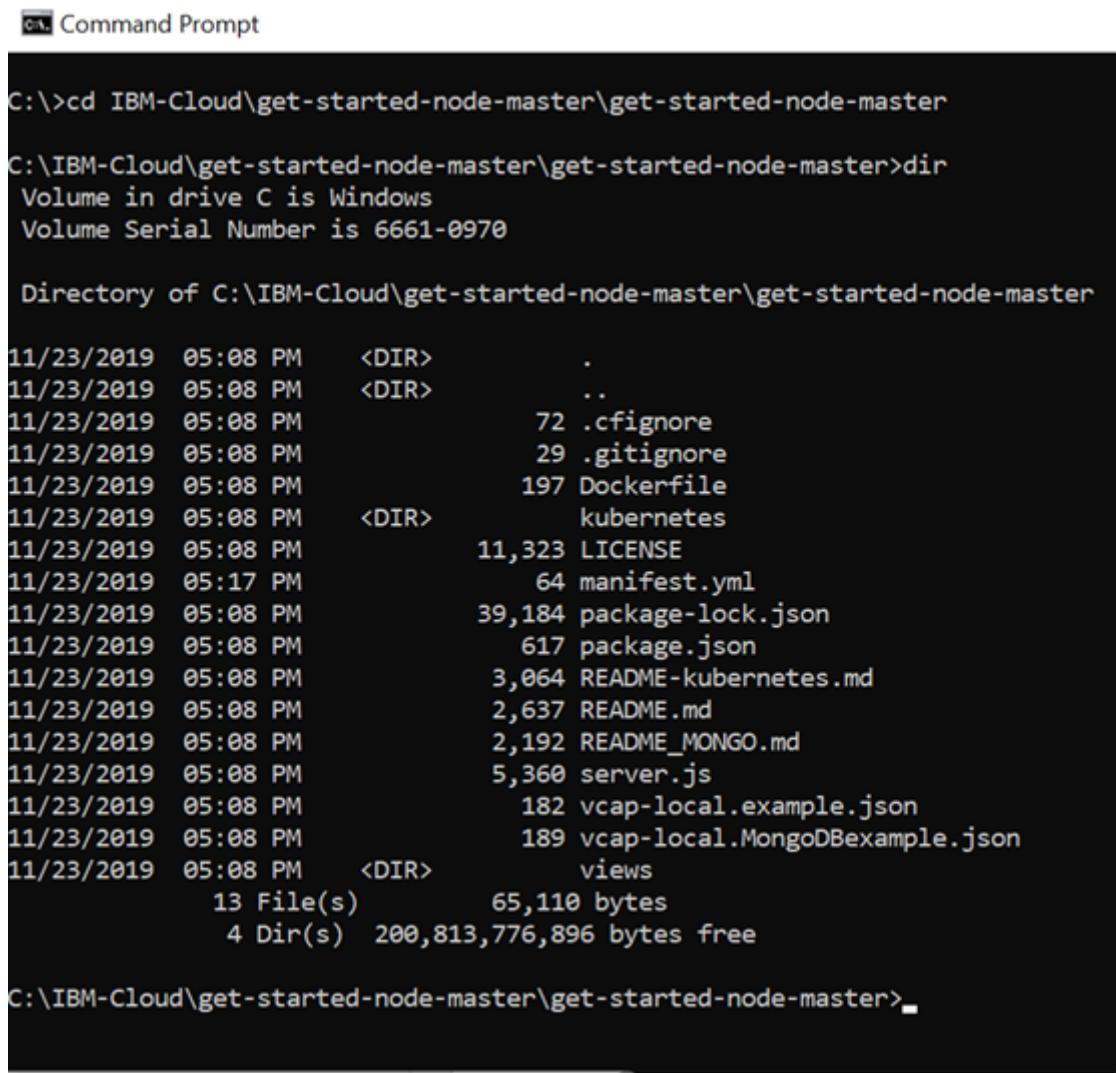
- \_\_\_ e. Extract the contents of the file into a source code directory for your application.

— 2. Connect to IBM Cloud by using the IBM Cloud CLI:

- a. In a Command Prompt window (Microsoft Windows) or terminal (Mac OS or Linux), change to the directory with the sample code. On almost any operating system, you can do this with the `cd` command.

For example, if you extracted the contents of the application to `\IBM-Cloud\get-started-node-master\get-started-node-master`, running `cd C:\IBM-Cloud\get-started-node-master\get-started-node-master` takes you to the correct location, as shown in the following figure.

If you run `dir` (Windows) or `ls` (MAC OS and Linux) and see the `manifest.yml` or `package.json` files, you are in the correct location, as shown in the following figure.



```
C:\>cd IBM-Cloud\get-started-node-master\get-started-node-master
C:\IBM-Cloud\get-started-node-master\get-started-node-master>dir
 Volume in drive C is Windows
 Volume Serial Number is 6661-0970

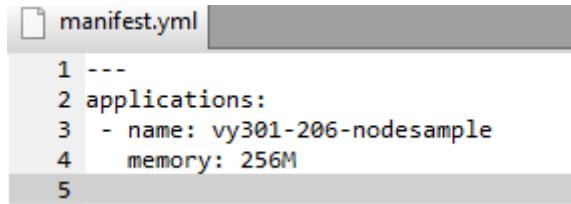
 Directory of C:\IBM-Cloud\get-started-node-master\get-started-node-master

11/23/2019  05:08 PM    <DIR>      .
11/23/2019  05:08 PM    <DIR>      ..
11/23/2019  05:08 PM            72 .cignore
11/23/2019  05:08 PM            29 .gitignore
11/23/2019  05:08 PM            197 Dockerfile
11/23/2019  05:08 PM    <DIR>      kubernetes
11/23/2019  05:08 PM            11,323 LICENSE
11/23/2019  05:17 PM            64 manifest.yml
11/23/2019  05:08 PM            39,184 package-lock.json
11/23/2019  05:08 PM            617 package.json
11/23/2019  05:08 PM            3,064 README-kubernetes.md
11/23/2019  05:08 PM            2,637 README.md
11/23/2019  05:08 PM            2,192 README_MONGO.md
11/23/2019  05:08 PM            5,360 server.js
11/23/2019  05:08 PM            182 vcap-local.example.json
11/23/2019  05:08 PM            189 vcap-local.MongoDBexample.json
11/23/2019  05:08 PM    <DIR>      views
                           13 File(s)       65,110 bytes
                           4 Dir(s)   200,813,776,896 bytes free

C:\IBM-Cloud\get-started-node-master\get-started-node-master>
```

- b. Use your favorite text editor to open the `manifest.yml`. For example, if you extracted the application source code to `C:\IBM-Cloud`, then this file is in `C:\IBM-Cloud\get-started-node-master\get-started-node-master\manifest.yml`
- c. Change the name from `GetStartedNode` to your app name, `vy301-xxx-nodesample`, where `xxx` is the randomly generated key that you use in this exercise.

- \_\_\_ d. random-route: true generates a random route to the application. To use the same application name as the route of the application, remove random-route: true, as shown in the following figure.



```

manifest.yml
1 ---
2 applications:
3   - name: vy301-206-nodesample
4     memory: 256M
5

```

- \_\_\_ e. Make sure that you are in the directory that contains the Node.js app code that you downloaded previously. Log in to the IBM Cloud with the CLI:

```
ibmcloud login -u username -p password -r region
```



### Note

You can use the one-time passcode option to log in with a federated ID. Specify the single sign-on (SSO) parameter `ibmcloud login --sso` to get a one-time passcode that you then enter at login, as shown in the following figure.

```

C:\IBM-Cloud>ibmcloud login --sso
API endpoint: https://cloud.ibm.com

Get One Time Code from https://identity-2.uk-south.iam.cloud.ibm.com/identity/passcode to proceed.
Open the URL in the default browser? [Y/n]> y
One Time Code >
Authenticating...
OK

```

Target the region that corresponds to the location that was set by default when you created the application. See [Regions](#) at [https://cloud.ibm.com/docs/cloud-foundry-public?topic=cloud-foundry-public-endpoints#endpoints\\_regions](https://cloud.ibm.com/docs/cloud-foundry-public?topic=cloud-foundry-public-endpoints#endpoints_regions)

- The region for Sydney is au-syd.
- The region for Frankfurt is eu-de.
- The region for London is eu-gb.
- The region for Dallas is us\_south.
- The region for Washington, D.C is us\_east.

The command response is as follows:

```
c:\IBM-Cloud\get-started-node-master\get-started-node-master>ibmcloud
login -u <your-username> -p <your-password> -r eu-gb
API endpoint: https://cloud.ibm.com
Authenticating...
OK
```

Targeted account Student20 Cloud20's Account  
(f6eeaf39ae6b477fbcead1993509e9c6)

Targeted region eu-gb

```
API endpoint:      https://cloud.ibm.com
Region:           eu-gb
User:              <your-username> (The email that you used to
register to IBM Cloud)
Account:          <your-accountname> (f6eeaf39ae6b477fbcead1)
Resource group:   No resource group targeted, use 'ibmcloud target -g
RESOURCE_GROUP'
CF API endpoint:
Org:
Space:
```

**Tip:** If you are managing Cloud Foundry applications and services

- Use 'ibmcloud target --cf' to target Cloud Foundry org/space interactively, or use 'ibmcloud target --cf-api ENDPOINT -o ORG -s SPACE' to target the org/space.
- Use 'ibmcloud cf' if you want to run the Cloud Foundry CLI with current IBM Cloud CLI

- 3. Now, you are logged in to IBM Cloud. To select the Cloud Foundry API endpoint, organization, and space to which you deploy your application, run the following command. Enter the same organization and space that were used when the application was created.  
`ibmcloud target --cf-api <CF API ENDPOINT> -o <ORG> -s <SPACE>`
- a. Use `--cf-api` to specify the Cloud Foundry API endpoint to which to deploy the application. Select it based on the region where the application was created. For the complete list of API endpoints see [API Endpoints](https://cloud.ibm.com/docs/cloud-foundry-public?topic=cloud-foundry-public-endpoints#api-endpoint-options) at <https://cloud.ibm.com/docs/cloud-foundry-public?topic=cloud-foundry-public-endpoints#api-endpoint-options>
  - <https://api.us-south.cf.cloud.ibm.com> for US SOUTH.
  - <https://api.eu-gb.cf.cloud.ibm.com> for EU-GB.
  - <https://api.us-east.cf.cloud.ibm.com> for US EAST
  - <https://api.eu-de.cf.cloud.ibm.com> for EU-DE.
  - <https://api.au-syd.cf.cloud.ibm.com> for AU-SYD.

In this example, `https://api.eu-gb.cf.cloud.ibm.com` is used, which is the API endpoint for the region EU-GB and location London.

- \_\_\_ b. Use `-o` to specify the organization, which is the email that you used to register to IBM Cloud.
- \_\_\_ c. Use `-s` to specify the space, which is `dev`.

The example shows the `ibmcloud target` command, options, and output on Windows. You receive the following output (API version might be different):

```
C:\IBM-Cloud\vy301-206-nodesample>ibmcloud target --cf-api  
https://api.eu-gb.cf.cloud.ibm.com -o <your_email> -s dev
```

Targeted Cloud Foundry (`https://api.eu-gb.cf.cloud.ibm.com`)

Targeted org <your\_email>

Targeted space dev

API endpoint:	<code>https://cloud.ibm.com</code>
Region:	<code>eu-gb</code>
User:	<your_email>
Account:	<your_account_name> (f6eeaf39ae6b477fbcead1993509e9c6)
Resource group:	Default
CF API endpoint:	<code>https://api.eu-gb.cf.cloud.ibm.com</code> (API version: 2.106.0)
Org:	<your_email>
Space:	dev

- 4. Upload and deploy the sample application to push the contents of the current directory as a Cloud Foundry application on IBM Cloud by running the following command:

```
ibmcloud cf push vy301-xxxx-nodesample
```

The following example shows a sample prompt and results for the `ibmcloud cf push` command on Windows:

```
C:\IBM-Cloud\vy301-206-nodesample>ibmcloud cf push
```

```
Invoking 'cf push'...
```

```
Pushing from manifest to org <your_email> / space dev as <your_email>...
```

```
Using manifest file C:\IBM-Cloud\vy301-206-nodesample\manifest.yml
```

```
Getting app info...
```

```
Updating app with these attributes...
```

```
name: vy301-206-nodesample
```

```
path:
```

```
C:\IBM-Cloud\get-started-node-master\get-started-node-master
```

```
command: npm start
```

```
disk quota: 1G
```

```
health check type: port
```

```
instances: 1
```

```
memory: 256M
```

```
stack: cflinuxfs3
```

```
routes:
```

```
vy301-206-nodesample.eu-gb.cf.appdomain.cloud
```

```
Updating app vy301-206-nodesample...
```

```
Mapping routes...
```

```
Comparing local files to remote cache...
```

```
Packaging files to upload...
```

```
Uploading files...
```

```
43.69 KiB / 43.69 KiB
```

```
[=====] ===== 100.00% ls
```

```
Waiting for API to complete processing files...
```

```
Stopping app...
```

```
Staging app and tracing logs...
```

```
Downloading liberty-for-java_v3_38-20191031-1433...
```

```
Downloading xpages_buildpack_v1_2_1-20160913-103...
```

```
Downloading nodejs_buildpack...
```

```
Downloading liberty-for-java...
```

```
Downloading sdk-for-nodejs...
```

```
Downloaded nodejs_buildpack
```

```
Downloading dotnet-core...
```

```
Downloaded xpages_buildpack_v1_2_1-20160913-103
```

```
Downloading swift_buildpack...
```

```
Downloaded liberty-for-java_v3_38-20191031-1433
```

```
Downloading xpages_buildpack...
Downloaded liberty-for-java
Downloading java_buildpack...
Downloaded sdk-for-nodejs
Downloading staticfile_buildpack...
Downloaded xpages_buildpack
Downloading sdk-for-nodejs_v4_0_1-20190930-1425...
Downloaded dotnet-core
Downloading swift_buildpack_v2_0_18-20190303-1915...
Downloaded swift_buildpack
Downloading swift_buildpack_v2_0_20-20190401-2122...
Downloaded staticfile_buildpack
Downloading swift_buildpack_cflinuxfs3_v2_1_0-20190404-1206...
Downloaded java_buildpack
Downloading dotnet-core_v2_3-20190609-2145...
Downloaded swift_buildpack_v2_0_18-20190303-1915
Downloading dotnet-core_v2_4-20190912-1554...
Downloaded sdk-for-nodejs_v4_0_1-20190930-1425
Downloading sdk-for-nodejs_v3_28-20190722-1336...
Downloaded swift_buildpack_v2_0_20-20190401-2122
Downloading ruby_buildpack...
Downloaded swift_buildpack_cflinuxfs3_v2_1_0-20190404-1206
Downloading liberty-for-java_v3_37-20191002-1726...
Downloaded dotnet-core_v2_3-20190609-2145
Downloaded nginx_buildpack...
Downloaded dotnet-core_v2_4-20190912-1554
Downloaded r_buildpack...
Downloaded sdk-for-nodejs_v3_28-20190722-1336
Downloading liberty-for-java_v3_36-20190905-1704...
Downloaded ruby_buildpack
Downloading python_buildpack...
Downloaded nginx_buildpack
Downloading go_buildpack...
Downloaded liberty-for-java_v3_37-20191002-1726
Downloading binary_buildpack...
Downloaded r_buildpack
Downloading php_buildpack...
Downloaded liberty-for-java_v3_36-20190905-1704
Downloaded python_buildpack
Downloaded binary_buildpack
Downloaded go_buildpack
Downloaded php_buildpack
Cell 476dc2a8-fa9b-4a7f-8572-cf9138ff6cfe creating container for instance
9b232c41-3960-430b-999e-f46d67fb1aae
Cell 476dc2a8-fa9b-4a7f-8572-cf9138ff6cfe successfully created container
for instance 9b232c41-3960-430b-999e-f46d67fb1aae
Downloading app package...
Downloading build artifacts cache...
```

```

Downloaded app package (43.7K)
Downloaded build artifacts cache (717.6K)
cat: /VERSION: No such file or directory
----> IBM SDK for Node.js Buildpack v4.0.1-20190930-1425
      Based on Cloud Foundry Node.js Buildpack 1.6.53

----> Installing binaries
      engines.node (package.json): 10./*
      engines.npm (package.json): unspecified (use default)
      Attempting to install: 10.16.3
----> Installing node 10.16.3
      Copy
[ /tmp/buildpacks/40386ebb61e6725a1463380a55a80e9d/dependencies/9afdb4f3300cc
6a181909f11075912df/node-10.16.3-linux-x64-cflinuxfs3-33294d36.tgz]
      Using default npm version: 6.9.0
----> Installing yarn 1.17.3
      Copy
[ /tmp/buildpacks/40386ebb61e6725a1463380a55a80e9d/dependencies/748132b4ee4ec
af8bbb5bfa5932e6689/yarn-1.17.3-any-stack-e3835194.tar.gz]
      Installed yarn 1.17.3
----> Creating runtime environment
      PRO TIP: It is recommended to vendor the application's Node.js
dependencies
      Visit
http://docs.cloudfoundry.org/buildpacks/node/index.html#vendorizing
      NODE_ENV=production
      NODE_HOME=/tmp/contents866686287/deps/0/node
      NODE_MODULES_CACHE=true
      NODE_VERBOSE=false
      NPM_CONFIG_LOGLEVEL=error
      NPM_CONFIG_PRODUCTION=true
----> Building dependencies
      Installing node modules (package.json + package-lock.json)
added 143 packages from 167 contributors and audited 396 packages in 6.16s
found 4 vulnerabilities (1 moderate, 3 high)
      run `npm audit fix` to fix them, or `npm audit` for details
      **WARNING** Unmet dependencies don't fail npm install but may cause
runtime issues
      See: https://github.com/npm/npm/issues/7494
      Contrast Security no credentials found. Will not write environment
files.
      Exit status 0
      Uploading droplet, build artifacts cache...
      Uploading build artifacts cache...
      Uploading droplet...
      Uploaded build artifacts cache (3.5M)
      Uploaded droplet (24.2M)
      Uploading complete

```

```

Cell 476dc2a8-fa9b-4a7f-8572-cf9138ff6cfe stopping instance
9b232c41-3960-430b-999e-f46d67fb1aae
Cell 476dc2a8-fa9b-4a7f-8572-cf9138ff6cfe destroying container for
instance 9b232c41-3960-430b-999e-f46d67fb1aae

```

Waiting for app to start...

```

Cell 476dc2a8-fa9b-4a7f-8572-cf9138ff6cfe successfully destroyed
container for instance 9b232c41-3960-430b-999e-f46d67fb1aae

```

```
name: vy301-206-nodesample
```

```
requested state: started
```

```
routes: vy301-206-nodesample.eu-gb.cf.appdomain.cloud
```

```
last uploaded: Sun 24 Nov 16:07:50 CST 2019
```

```
stack: cflinuxfs3
```

```
buildpacks: sdk-for-nodejs
```

```
type: web
```

```
instances: 1/1
```

```
memory usage: 256M
```

```
start command: npm start
```

state	since	cpu	memory	disk
-------	-------	-----	--------	------

```
details
```

#0	running	2019-11-24T22:08:04Z	0.1%	41.8M of 256M	80.6M of 1G
----	---------	----------------------	------	---------------	-------------

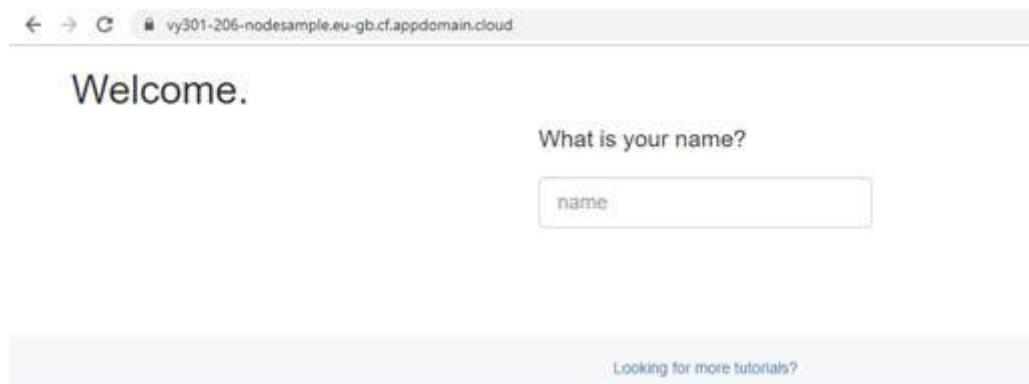
Wait until the CLI exits and you return to the command prompt. This is how you know that your files finished uploading.



### Note

Because you already created the application `vy301-xxx-nodesample` earlier by using the IBM Cloud console, pushing the application from the CLI overwrites the contents of the existing application on IBM Cloud. If you do not have an existing Cloud Foundry application of the same name on IBM Cloud, pushing the application creates an application on IBM Cloud.

- \_\_\_ 5. Confirm that your sample application is running by completing the following steps:
  - \_\_\_ a. Open the application URL in a web browser. The URL is listed in the command prompt output (routes). In this example the URL is `vy301-206-nodesample.eu-gb.cf.appdomain.cloud`.
  - \_\_\_ b. Confirm that the sample application is deployed successfully, as shown in the following figure.



6. Modify the source code to produce a necessary change. Open the file `get-started-node-master\views\index.html` in your favorite text editor. The source code is shown in the following example.

```
<!DOCTYPE html>
<html>

<head>
    <meta charset="utf-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <title>Hello World</title>

    <!-- Bootstrap -->
    <link
        href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.min.css"
        rel="stylesheet">
        <link href="styles.css" rel="stylesheet">
        <!-- jQuery (necessary for Bootstrap's JavaScript plugins) -->
        <script
            src="https://ajax.googleapis.com/ajax/libs/jquery/1.12.4/jquery.min.js"></script>
            <!-- Include all compiled plugins (below), or include individual files as
            needed -->
            <script
                src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/js/bootstrap.min.js"></script>
                <script src="js/lib/jquery.i18n/jquery.i18n.js"></script>
                <script src="js/lib/jquery.i18n/jquery.i18n.messagesstore.js"></script>
                <script src="js/lib/jquery.i18n/jquery.i18n.fallbacks.js"></script>
                <script src="js/lib/jquery.i18n/jquery.i18n.language.js"></script>
                <script src="js/lib/jquery.i18n/jquery.i18n.parser.js"></script>
                <script src="js/lib/jquery.i18n/jquery.i18n.emitter.js"></script>
                <script src="js/lib/jquery.i18n/jquery.i18n.emitter.bidi.js"></script>
                <script src="antixss.js" type="text/javascript"></script>

                <script>
                    $( document ).ready(function() {
                        $.i18n().load( {
                            en: {
                                "welcome": "Welcome.",
                                "name": "name",
                                "what_is_your_name": "What is your name?",
                                "hello": "Hello $1",
                                "added_to_database": "Hello $1, I've added you to the
                                database!",
                                "database_contents": "Database contents: "
                            },
                            ja: {

```

```
"welcome": "&#12424;&#12358;&#12371;&#12381;&#12290;" ,  
"name": "&#21517;&#21069;" ,  
"what_is_your_name":  
"&#12362;&#21517;&#21069;&#12434;&#25945;&#12360;&#12390;&#12367;&#12384;&#1  
2373;&#12356;&#12290;" ,  
"hello": "&#12371;&#12435;&#12395;&#12385;&#12399; $1" ,  
"added_to_database":  
"&#12371;&#12435;&#12395;&#12385;&#12399; $1  
&#12373;&#12435;&#12289;&#12354;&#12394;&#12383;&#12434;&#12487;&#12540;&#12
```

```

479;#12505;#12540;#12473;#12395;#36861;#21152;#12375;#12414;#12375;
&#12383;#12290;" ,
           "database_contents": "
"#!/12487;#12540;#12479;#12505;#12540;#12473;#12398;#20869;#23481;: "
        }
    } );
$( 'body' ).i18n();
$( '#user_name' ).attr("placeholder", $.i18n( 'name' ) );
});
</script>

</head>

<body>
<div class="container" id="container">
    <h1 data-i18n="welcome"></h1> <!-- Welcome -->
    <div id="nameInput" class="input-group-lg center-block helloInput">
        <p class="lead" data-i18n="what_is_your_name"></p>
        <input id="user_name" type="text" class="form-control"
aria-describedby="sizing-addon1" value="" />
    </div>
    <p id="response" class="lead text-center"></p>

    <p id="databaseNames" class="lead text-center"></p>
</div>
<footer class="footer">
    <div class="container">
        <span><a
href="https://console.bluemix.net/docs/tutorials/index.html"
target="_blank">Looking for more tutorials?</a></span>
    </div>
</footer>

</body>

</html>

<script>
    //Submit data when enter key is pressed
    $('#user_name').keydown(function(e) {
        var name = $('#user_name').val();
        if (e.which == 13 && name.length > 0) { //catch Enter key
            //POST request to API to create a new visitor entry in the
            database
            $.ajax({
                method: "POST",

```

```

        url: "./api/visitors",
        contentType: "application/json",
        data: JSON.stringify({name: name })
    })
.done(function(data) {
    if(data && data.name){
        if(data._id)
            $('#response').html($.i18n('added_to_database'),
AntiXSS.sanitizeInput(data.name)));
        else
            $('#response').html($.i18n('hello'),
AntiXSS.sanitizeInput(data.name)));
    }
    else {
        $('#response').html(AntiXSS.sanitizeInput(data));
    }
    $('#nameInput').hide();
    getNames();
});
}

//Retrieve all the visitors from the database
function getNames(){
$.get("./api/visitors")
.done(function(data) {
if(data.length > 0) {
    data.forEach(function(element, index) {
        data[index] = AntiXSS.sanitizeInput(element)
    });
    $('#databaseNames').html($.i18n('database_contents') +
JSON.stringify(data));
}
});
}

//Call getNames on page load.
getNames();

</script>

```

- 7. In line 8, change Hello World to Greetings. and in line 30 change Welcome. to Greetings as shown in the following figure.

```

1  <!DOCTYPE html>
2  <html>
3
4  <head>
5      <meta charset="utf-8">
6      <meta http-equiv="X-UA-Compatible" content="IE=edge">
7      <meta name="viewport" content="width=device-width, initial-scale=1">
8      <title>Greetings</title>
9
10     <!-- Bootstrap -->
11     <link href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.min.css" rel="stylesheet">
12     <link href="styles.css" rel="stylesheet">
13     <!-- jQuery (necessary for Bootstrap's JavaScript plugins) -->
14     <script src="https://ajax.googleapis.com/ajax/libs/jquery/1.12.4/jquery.min.js"></script>
15     <!-- Include all compiled plugins (below), or include individual files as needed -->
16     <script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/js/bootstrap.min.js"></script>
17     <script src="js/lib/jquery.i18n.i18n.js"></script>
18     <script src="js/lib/jquery.i18n/jquery.i18n.messagestore.js"></script>
19     <script src="js/lib/jquery.i18n/jquery.i18n.fallbacks.js"></script>
20     <script src="js/lib/jquery.i18n/jquery.i18n.language.js"></script>
21     <script src="js/lib/jquery.i18n/jquery.i18n.parser.js"></script>
22     <script src="js/lib/jquery.i18n/jquery.i18n.emitter.js"></script>
23     <script src="js/lib/jquery.i18n/jquery.i18n.emitter.bidi.js"></script>
24     <script src="antixss.js" type="text/javascript"></script>
25
26     <script>
27         $( document ).ready(function() {
28             $.i18n().load( {
29                 on: {
30                     "welcome": "Greetings.",
31                     "name": "name",
32                     "what_is_your_name": "What is your name?",
33                     "hello": "Hello $1",
34                     "added_to_database": "Hello $1, I've added you to the database!",
35                     "database_contents": "Database contents: "
36                 },
37             });
38         });
39     </script>

```

- 8. Save the file and exit your text editor.
- 9. Return to the Command Prompt and run the push command to upload and deploy the application again with the new changes.

`ibmcloud cf push vy301-xxx-nodesample`

The following figure shows an example of the `cf push` command.

```
c:\IBM-Cloud\get-started-node-master\get-started-node-master>ibmcloud cf push vy301-206-nodesample
```

Wait until the CLI exits and you return to a command prompt.

- 10. To confirm that your changes are deployed successfully, refresh the application page in the browser and check that “Welcome.” is replaced by “Greetings.” as shown in the following figure.



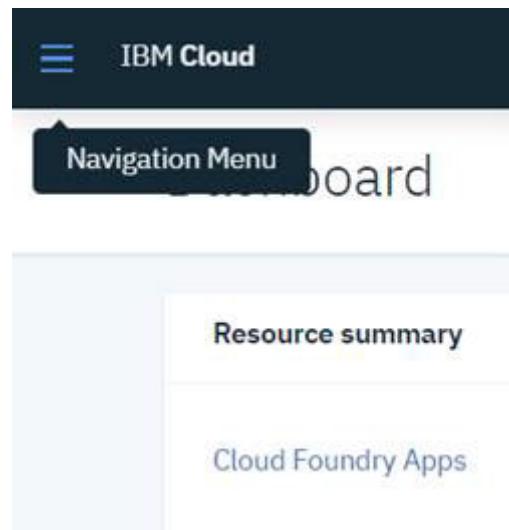
- \_\_\_ 11. Close both the command prompt and the web browser for your application page.

### **Part 5: Checking your organization limits**

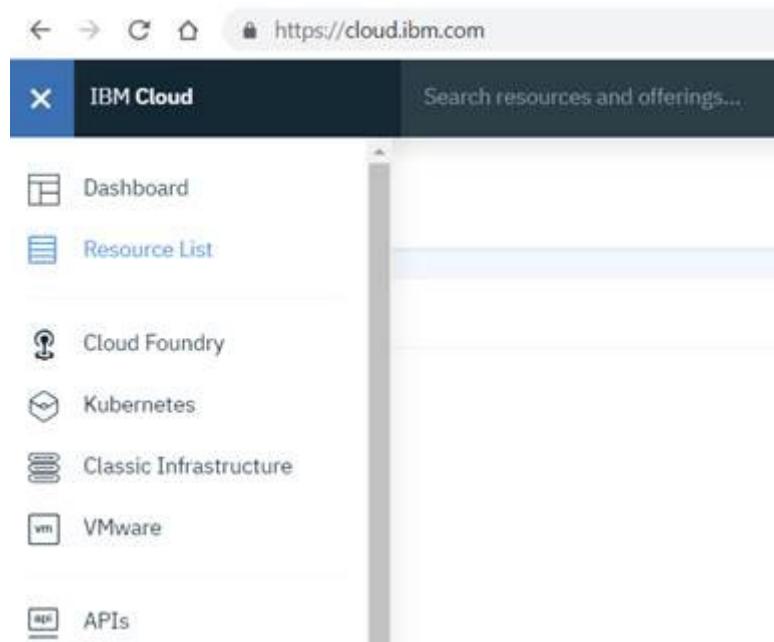
Your IBM Cloud Lite account provides only some services and a maximum amount of random access memory (RAM) that your apps can use. In this section you check the amount of memory allocated to your application.

Complete the following steps:

- \_\_\_ 1. Open IBM Cloud console (<https://cloud.ibm.com/>) in your browser. Click the **Navigation Menu** (the three dashes at the upper left) as shown in the following figure.



- \_\_\_ 2. Click **Resources List** to open the Resources List to access your application, as shown in the following figure.



- \_\_\_ 3. View the amount of consumed resources in your dashboard:
- \_\_\_ a. Click your application name that is listed under Cloud Foundry Apps, as shown in the following figure.

The screenshot shows the 'Resource list' page. At the top, there is a 'Create resource' button. Below it, there are filter options for 'Name', 'Group', 'Location', 'Status', and 'Tags'. The main area displays a list of resources categorized by type. Under 'Cloud Foundry apps', there is one entry: 'vy301-206-nodesample' (indicated by a yellow warning icon). This entry includes columns for Name ('vy301-206-nodesample'), Group ('/ dev'), Location ('London'), Status ('Start...'), and Actions ('...'). Other collapsed sections include 'Devices (0)', 'VPC infrastructure (0)', 'Clusters (0)', 'Cloud Foundry services (0)', and 'Services (0)'.

- \_\_\_ b. Note the **Runtime** tile. The following figure shows that your application consumes 256 MB of the allotted memory in this IBM Cloud organization. In the next part, you delete the application to clean up your organization for the next exercise.



### Note

Depending on your account type, you might be allotted a different amount of RAM or other resources in IBM Cloud. The amount of RAM for you might be different from the example.

## Part 6: Deleting the sample application

In this section, you delete your app so that your IBM Cloud organization is clean and ready for the next exercise. To delete the sample application complete the following steps:

- Find and click the gear menu icon to the right of your app name. You are presented with a list of options, as shown in the following figure.

The screenshot shows the IBM Cloud Resource list for the application 'vy301-206-nodesample'. The application details are:

- Org:** [redacted]
- Location:** London
- Space:** dev
- Add Tags**

The 'Routes' dropdown menu is open, showing the following options:

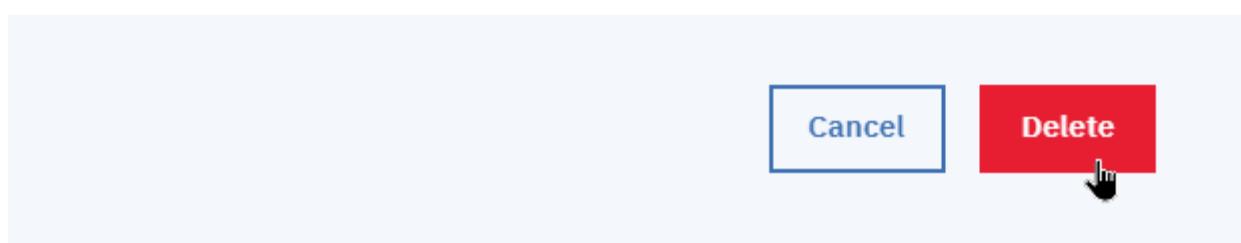
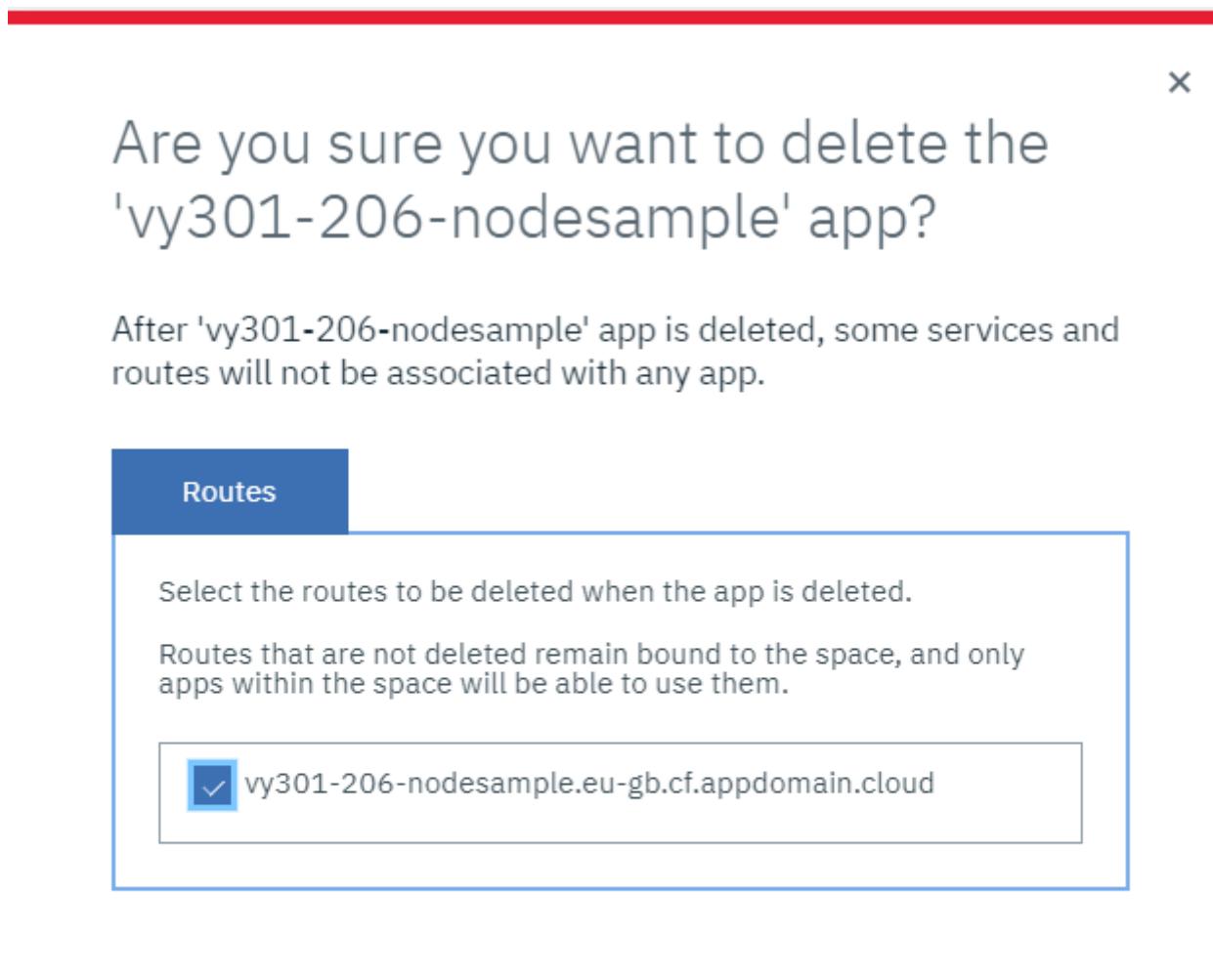
- Stop
- Restart
- Rename
- Delete** (highlighted with a green border)

The application configuration is identical to the one shown in the first screenshot:

- BUILDPACK:** SDK for Node.js™
- INSTANCES:** 1 (All instances are running, Health is 100%)
- MB MEMORY PER INSTANCE:** 256
- TOTAL MB ALLOCATION:** 256 (0 MB still available)

2. Select **Delete** to delete your application.

3. You are presented with a pane, as shown in the following figure. Confirm that you want to delete the application, and delete the route (ensure that this box is selected). Click **Delete**.



### Important

An IBM Cloud Lite account provides up to 256 MB of application memory for Cloud Foundry Apps and 100 Cloud Foundry services. To free the resources that are assigned to your application, either stop your application or delete it. If you used up your quota for services, you must delete the existing services to make room for new ones.

## End of exercise

## Exercise review and wrap-up

In this exercise, you registered for an IBM Cloud account and explored the IBM Cloud dashboard and catalog by signing in with your IBMid. Then, you created an IBM Cloud application with the IBM SDK for Node.js runtime environment. You used the IBM Cloud CLI to modify the source code of the sample application on your local workstation and redeployed the updated application from your local workstation to your IBM Cloud account by using the IBM Cloud CLI.

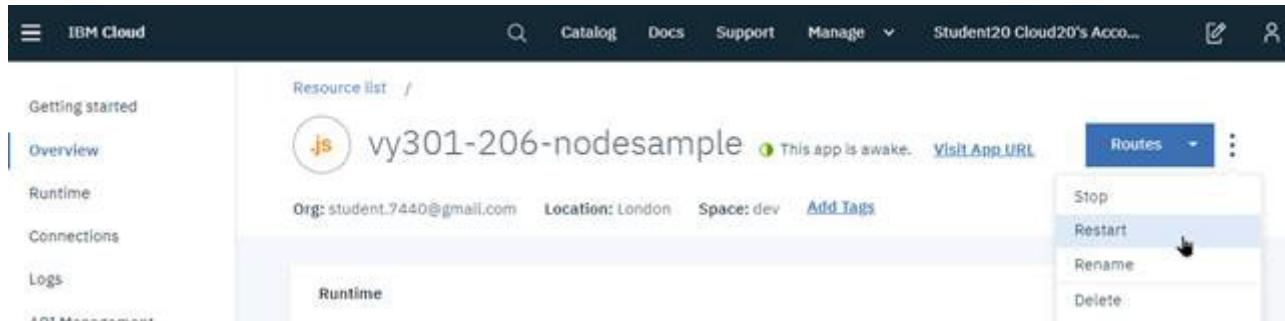
## Troubleshooting

This section lists common problems that students might encounter while performing this exercise:

- If you receive an error message while running the `ibmcloud cf app` command, try running `ibmcloud cf push` again from the same directory.

Confirm that you are in the correct directory. You must run `ibmcloud cf push` from a directory that contains the `package.json` and `manifest.yml` files.

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



Then, click **Restart** to try starting your app again.

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

- The following error is listed when you deploy the node.js application with the `ibmcloud cf push` command::

```
-----> Installing binaries
  engines.node (package.json): 6.x
  engines.npm (package.json): unspecified (use default)
  **ERROR** Unable to install node: no match found for 6.x in [8.16.0
8.16.1 10.16.0 10.16.3 12.7.0 12.8.1]
Failed to compile droplet: Failed to run all supply scripts: exit status 14
Exit status 223
Cell 66ff3ac8-67d1-4939-bdc5-d8f78a7ff3ee stopping instance
f85af53d-9081-45a7-97ef-bf4ba5c1a7a6
Cell 66ff3ac8-67d1-4939-bdc5-d8f78a7ff3ee destroying container for instance
f85af53d-9081-45a7-97ef-bf4ba5c1a7a6
Cell 66ff3ac8-67d1-4939-bdc5-d8f78a7ff3ee successfully destroyed container for
instance f85af53d-9081-45a7-97ef-bf4ba5c1a7a6
Error staging application: App staging failed in the buildpack compile phase
FAILED
```

To fix this error, change the node version in `package.json` to match one of the supported node.js versions that are listed in the error as shown in the following figure.

```
{  
  "name": "get-started-node",  
  "main": "server.js",  
  "description": "An introduction to developing Node.js apps on the IBM Cloud platform",  
  "version": "0.1.1",  
  "private": false,  
  "engines": {  
    "node": "10.*"  
  },  
  "scripts": {  
    "start": "node server.js"  
  },  
  "repository": {  
    "type": "git",  
    "url": "https://github.com/IBM-Cloud/get-started-node"  
  },  
  "dependencies": {  
    "@cloudant/cloudant": "^3.0.2",  
    "body-parser": "^1.17.x",  
    "cfenv": "^1.0.x",  
    "dotenv": "^4.0.0",  
    "express": "^4.15.x",  
    "mongodb": "^3.0.10"  
  },  
  "author": "IBM Corp",  
  "license": "Apache-2.0"  
}  
}
```

- Each organization has a memory quota and number of services quota. Before creating an application, check that you have enough memory. If you do not have enough memory available, stop the running applications.

---

# 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

.....



[Forgot ID?](#)

[Forgot password?](#)

**Continue**

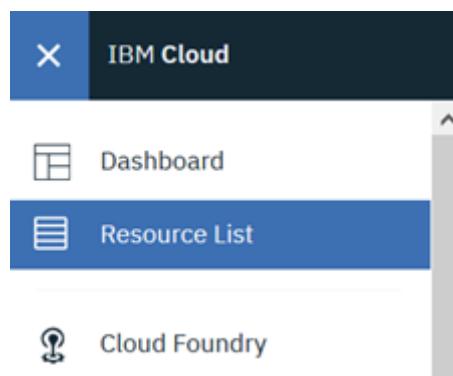
- \_\_\_ b. Click **Continue**.
- \_\_\_ c. Confirm that your IBM Cloud Dashboard page loads.
- \_\_\_ 3. Perform the steps in Exercise 1, Part 3, “Creating an application” to create a Node.js application.

## 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

Name	Group	Location	Offering	Status	Tags
<input type="text"/> Filter by name or IP address...	<input type="text"/> Filter by group or org...	<input type="button"/> Filter...	<input type="text"/> Filter...	<input type="text"/> Filter...	<input type="text"/> Filter...
> Devices (0)					
> Kubernetes Clusters (0)					
Cloud Foundry Apps (1)					
vy301-bf1-nodesample	student103@yahoo.com / dev	Dallas	SDK for Node.js™	<span style="color: green;">Running</span>	<input type="button"/>

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

The screenshot shows the IBM Cloud Application Details page for an application named "vy301-bf1-nodesample". The left sidebar has "Overview" selected. The main area displays the application's runtime configuration:

- Runtime:**
  - BUILDPACK:** SDK for Node.js™
  - INSTANCES:** 1 (All instances are running, Health is 100%)
  - MB MEMORY PER INSTANCE:** 256
  - TOTAL MB ALLOCATION:** 256 (0 MB still available)
- Routes:** A button with a dropdown arrow.
- Feedback:** A button.

At the top, it says "This app is awake." and provides a "Visit App URL" link.



### 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.

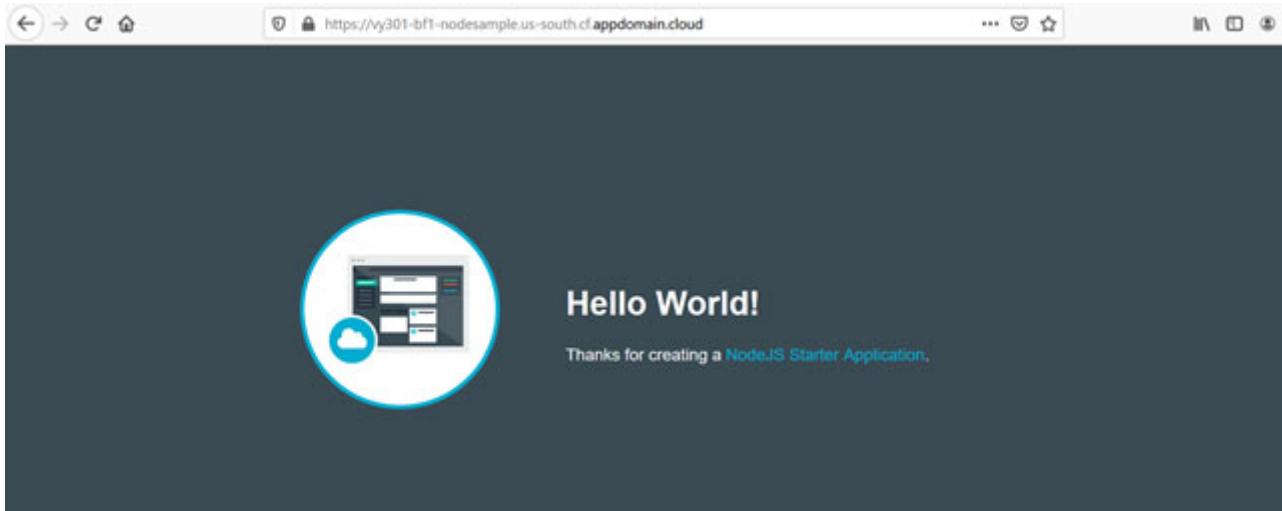
- 3. Test the sample application:

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

Resource list /



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



- \_\_\_ c. Close the Node.js starter application web page.

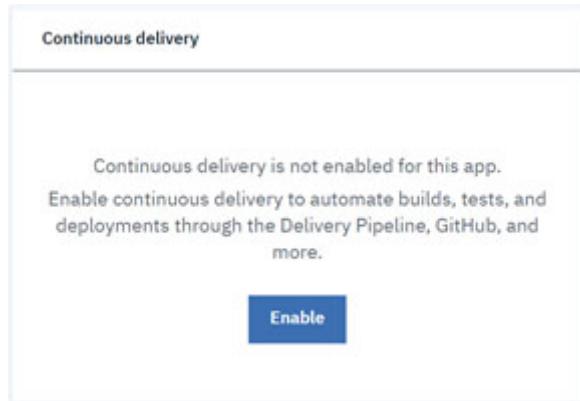
### **Part 3: Enabling Continuous Delivery for the starter application**

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.



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

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

- 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  
Select a CF Organization (deprecated)

- \_\_\_ 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

Git Repos and Issue Tracking      Delivery Pipeline Required      More tools

The Delivery Pipeline automates continuous deployment.

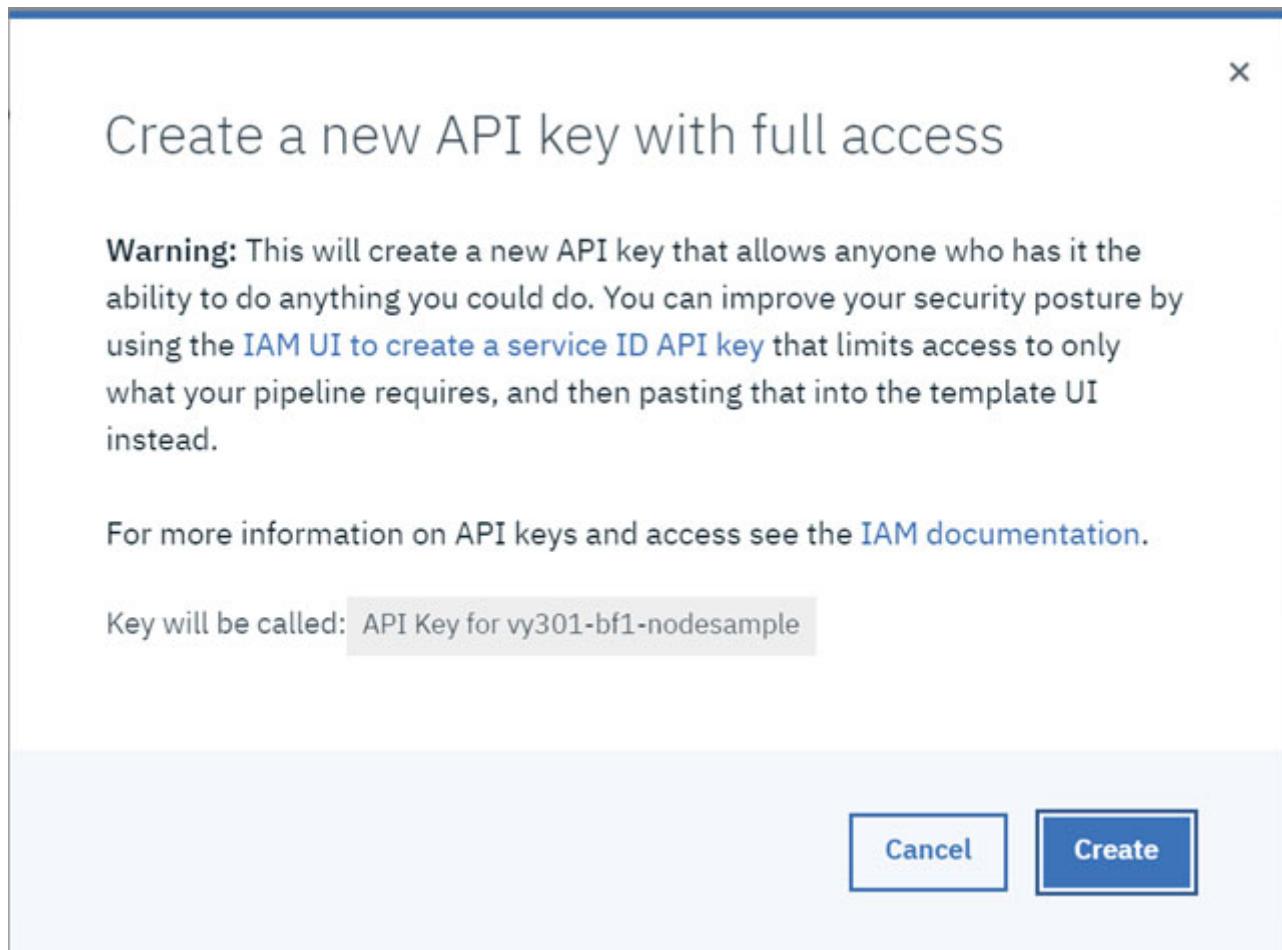
IBM Cloud API key:

IBM Cloud API key The value is required. Create + (i)

**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.



- 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:

Toolchain Name:  
vy301-bf1-nodesample

Select Region: Dallas

Select a resource group: Default

Select a CF Organization (deprecated)

Tool Integrations

- Git Repos and Issue Tracking
- Delivery Pipeline
- More tools



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.

Toolchains / vy301-bf1-nodesample [Visit App URL](#)

Resource Group: Default Location: Dallas Add tags

THINK	CODE	DELIVER
Issues vy301-bf1-nodesample <span style="color: green;">✓ Configured</span>	Git vy301-bf1-nodesample <span style="color: green;">✓ Configured</span>	Delivery Pipeline vy301-bf1-nodesample <span style="color: gray;">● Not yet run</span>
Eclipse Orion Web IDE <span style="color: green;">✓ Configured</span>		

Add a Tool Feedback Ask a Question



### 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.

- 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.

The screenshot shows the IBM Cloud interface with the following details:

- Header:** IBM Cloud, Search resources and offerings, Catalog, Docs, Support, Manage, Student103 Cloud103's Acc...
- Left sidebar:** Root (cloud.ibm.com), vy301-bf1-nodesample (selected), Help.
- Central pane:**
  - Project Name:** vy301-bf1-nodesample
  - 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:8001>
  - File listing:** vy301-bf1-nodesample
    - Name: public, Date Modified: 4/4/2019, 7:23:48 PM, Size: -
    - Name: .ignore, Date Modified: 4/4/2019, 7:23:48 PM, Size: 1 KB
    - Name: .project, Date Modified: 4/4/2019, 7:23:48 PM, Size: 1 KB
    - Name: app.js, Date Modified: 4/4/2019, 7:23:48 PM, Size: 1 KB
    - Name: LICENSE, Date Modified: 4/4/2019, 7:23:48 PM, Size: 12 KB
    - Name: manifest.yml, Date Modified: 4/4/2019, 7:23:48 PM, Size: 1 KB

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

The screenshot shows the IBM Cloud application dashboard interface. At the top, there's a navigation bar with File, Edit, View, Tools, and a 'Create new launch configuration' button. Below the navigation is a sidebar with icons for Root (cloud.ibm.com), a specific application named 'vy301-bf1-nodesample' (which is selected and highlighted in blue), and other options like Git and JavaScript.

The main content area displays the 'vy301-bf1-nodesample' application details. It includes a section titled 'Node.js Hello World Sample' with a brief description: 'This application demonstrates a simple, reusable Node.js web application based on the Express framework.' Below this is a 'Run the app locally' section with a numbered list of steps:

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>

Below this is a file listing for the application's root directory:

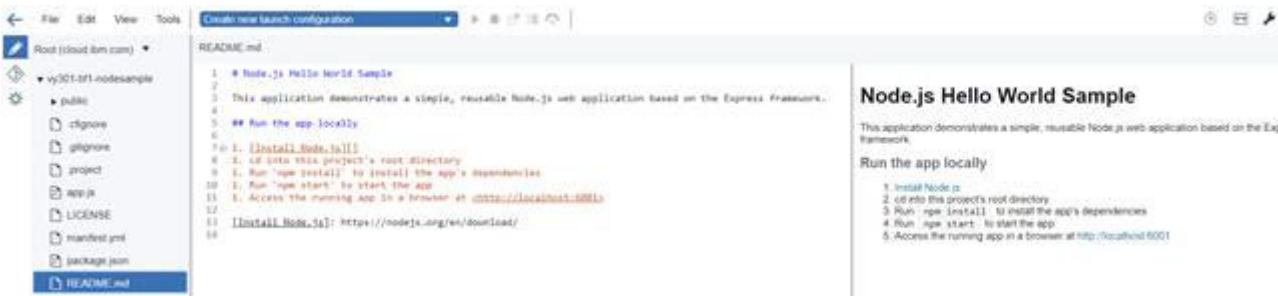
Name	Date Modified	Size
public	12/13/2019, 11:27:24 AM	—
.cfignore	12/13/2019, 11:27:24 AM	1 KB
.gitignore	12/13/2019, 11:27:25 AM	1 KB
project	12/13/2019, 11:27:24 AM	1 KB
app.js	12/13/2019, 11:27:24 AM	1 KB
LICENSE	12/13/2019, 11:27:24 AM	12 KB
manifest.yml	12/13/2019, 11:27:24 AM	1 KB
package.json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

At the bottom of the main content area, there are two collapsed sections: 'Git' and 'JavaScript'.

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.

<b>Git</b>	
Git Url	<a href="https://us-south.git.cloud.ibm.com/student103/vy301-bf1-nodesample.git">https://us-south.git.cloud.ibm.com/student103/vy301-bf1-nodesample.git</a>
Git Repository	<a href="#">Git Repository</a>
<b>JavaScript</b>	
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	<a href="#">package.json file</a>
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.



```

# 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 `node start` to start the app
5. Access the running app in a browser at http://localhost:8081
6. Uninstall Node.js: https://nodejs.org/en/download/

```

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.

The screenshot shows the IBM Cloud dashboard. The top navigation bar includes links for Catalog, Docs, Support, Manage, and Student. The 'Manage' link is currently selected, highlighted in blue. Below the navigation bar, there's a search bar and a 'Resource list /' section. A specific application named 'vy301-bf1-nodesample' is listed, showing its status as 'awake'. Below the application details, it shows the organization ('Org: student103@yahoo.com'), location ('Location: Dallas'), and space ('Space: dev'). At the bottom of the screen, the 'Access (IAM)' tab is also highlighted in blue, indicating it is the active section.

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

### Manage access and users

[Invite users](#) +

- 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.

The screenshot shows the 'Invite users' page. At the top, there's a header with 'Users /' and 'Account: Student103 Cloud103's Acco...'. Below the header, there's a section titled 'Enter email addresses' with a text input field containing 'Eduardo.Monich.Fronza@ibm.com'. To the right of this, there's an 'Access summary' section with tabs for 'Summary' (selected) and 'API'. The 'Summary' tab shows 1 User, 0 Access groups, and 0 Assignments. Below this, there's a note: 'You haven't added any access, but that's ok. You can still invite users to your account.' At the bottom of the page are two buttons: 'Invite' and 'Cancel'.

— 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.

The screenshot shows the 'Platform access' section of the IBM Cloud interface. It lists four roles: Viewer, Operator, Editor, and Administrator. The 'Editor' role is selected, indicated by a checked checkbox. The other three roles have unchecked checkboxes. Each role has a corresponding description and a small numbered callout bubble (e.g., 12, 21, 35, 44) next to it. At the bottom right of the list are two buttons: 'Reset' and 'Add'.

— 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.

Reset Add ⌂

6. Click **Invite**.

## Access summary

### Summary

### API

<b>1</b>	<b>0</b>	<b>2</b>
User	Access groups	Assignments

Cloud Foundry

**student103@yahoo.com**

Organization roles:

Auditor

Region: Dallas

Space: dev

Space roles:

Developer

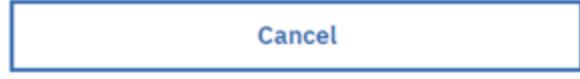
[Remove](#)  [Edit](#) 

IAM services

**All Toolchain resources**

Editor

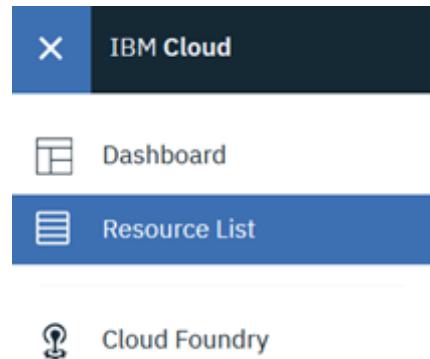
[Remove](#)  [Edit](#) 



## 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.

A screenshot of a "Continuous delivery" section. It contains a message: "You enabled continuous delivery and have a toolchain. With your toolchain, you can automate builds, tests, deployments, and more." Below the message is a blue button labeled "View toolchain".

- \_\_\_ 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.

The screenshot shows the IBM Cloud Web IDE interface. On the left, there's a file tree with a blue selected icon next to 'vy301-bf1-nodesample'. The tree contains 'public', '.cignore', '.gitignore', '.project', 'app.js', 'LICENSE', 'manifest.yml', 'package.json', and 'README.md'. The right pane displays the contents of 'README.md' under the heading 'Node.js Hello World Sample'. It includes instructions for running the app locally and a table showing the contents of 'vy301-bf1-nodesample'.

Name	Date Modified	Size
public	4/4/2019, 7:23:48 PM	-
.cignore	4/4/2018, 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.

This screenshot is similar to the one above, showing the IBM Cloud Web IDE. The 'vy301-bf1-nodesample' project is selected. Three icons are highlighted with blue boxes: a pencil icon for 'Edit', a circular arrow icon for 'Git', and a gear icon for 'Settings'. The right pane shows the 'Node.js Hello World Sample' details and the local run instructions.

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.

The screenshot shows the IBM Cloud application editor interface. On the left, there is a file tree for a project named 'vy301-bf1-nodesample'. The 'app.js' file is selected and highlighted with a blue background. The right pane contains the code for 'app.js', which is a Node.js application using Express.js and cfenv. The code includes comments explaining its purpose and how it interacts with the Cloud Foundry environment.

```

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.

```

1  /*eslint-env node*/
2
3  // mode.js starter application for Bluemix
4
5  // This application uses express as its web server
6  // for more info, see: http://expressjs.com
7  var express = require('express')
8
9  // cfenv provides access to your Cloud Foundry environment
10 // for more info, see: https://www.npmjs.com/package/cfenv
11 var cfenv = require('cfenv')
12
13 // create a new express server
14 var app = express()
15
16 // serve the files from the app's directory
17 app.use(express.static(require('./')))
18
19 // get the app environment variable
20 var appEnv = cfenv.getAppEnv()
21
22 // start server on the port specified in the environment variable
23 app.listen(appEnv.port, exports.module.exports)
24
25 // print a message to the logs
26 console.log("serve commons")
27
28 });
29

```

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
11 </head>
12
13 <body>
14   <table>
15     <tr>
16       <td style="width: 30%;>
17         
18       </td>
19       <td>
20         <h1 id="message">Hello World!</h1>
21         <p class="description"><span class="blue">Thanks for creating a <span class="orange">NodeJS Starter Application</span>,</span></p>
22       </td>
23     </tr>
24   </table>
25
26 </body>
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.

The screenshot shows the IBM Cloud developer tools interface. The top navigation bar includes 'IBM Cloud', a search bar, and links for 'Catalog', 'Docs', 'Support', 'Manage', and 'Student'. Below the header is a toolbar with icons for file operations like 'File', 'Edit', 'View', 'Tools', and a 'Create new launch configuration' button. The left sidebar displays a file tree for a project named 'vy301-bf1-nodesample'. The 'public' directory contains 'images' and 'stylesheets', and the root directory contains '.ignore', '.gitignore', '.project', 'app.js', 'LICENSE', 'manifest.yml', 'package.json', and 'README.md'. The 'index.html' file is selected and shown in the main code editor area. The code in 'index.html' is as follows:

```

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                     
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>

```

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

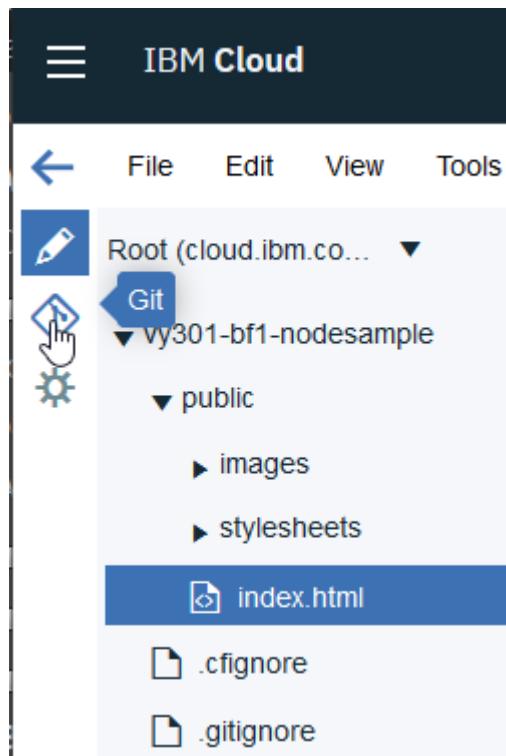
## **Part 7: Committing your changes to the Git repository**

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.

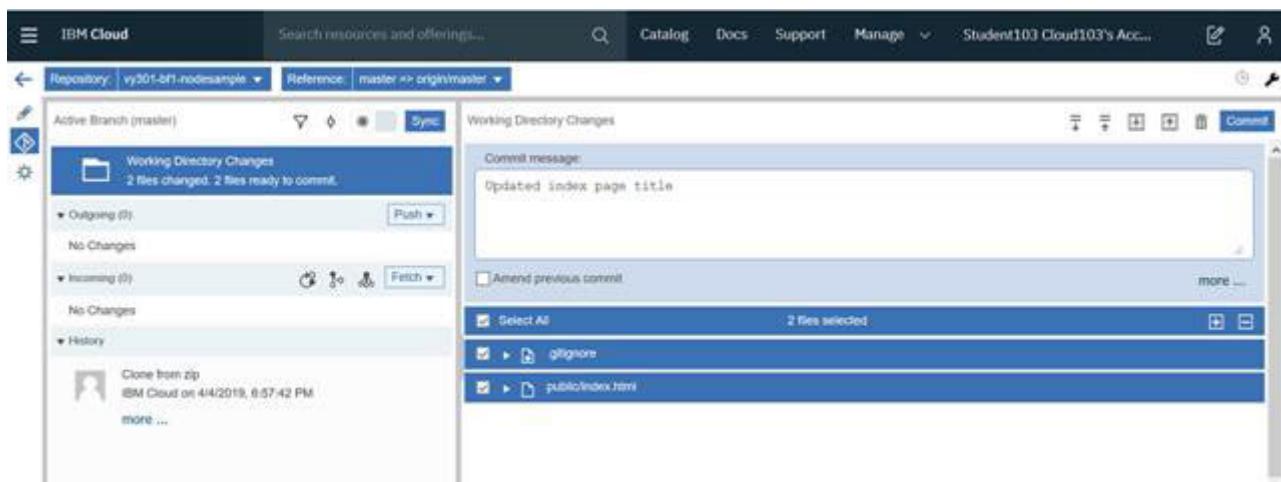
You commit the source code, configuration files, and other artifacts to the Git repository. Also, you deploy the application changes from the Git repository to the application in your IBM Cloud account. In the next Part, you learn that Delivery Pipeline automatically deploys the changes that are pushed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



- \_\_\_ 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.

The screenshot shows the IBM Cloud interface for a repository named 'vy301-br1-nodesample'. The 'Working Directory Changes' section is open, showing the 'Outgoing' tab which contains one item: 'Updated index page title' by 'Student103' on '4/11/2019, 7:02:46 PM'. The 'Commit message' field is empty, and there is a checkbox for 'Amend previous commit'. The 'Incoming' and 'History' tabs are also visible but show 'No Changes'.

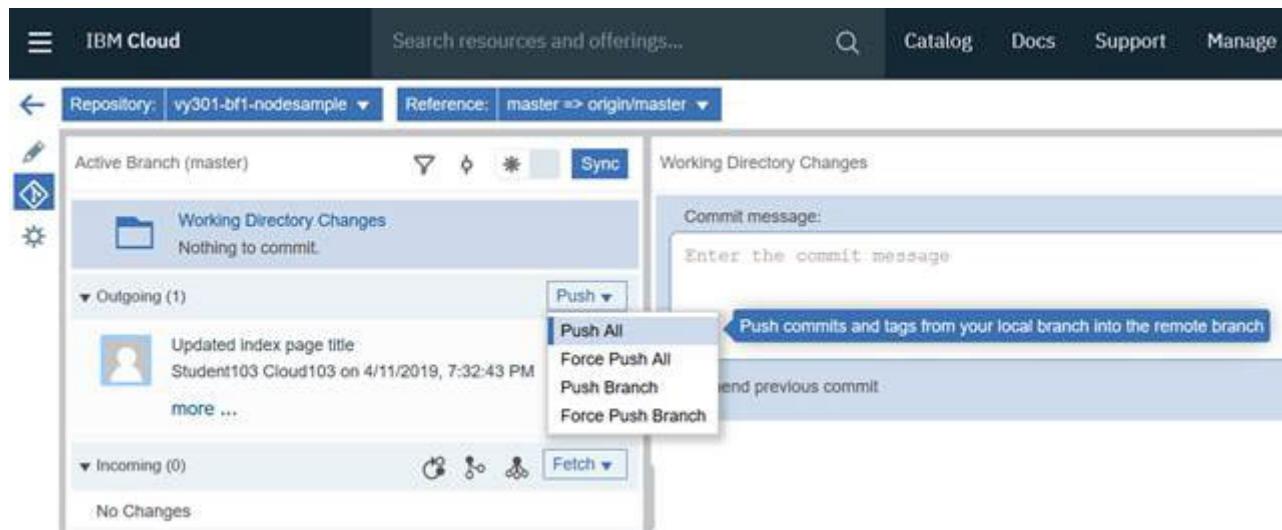


## Information

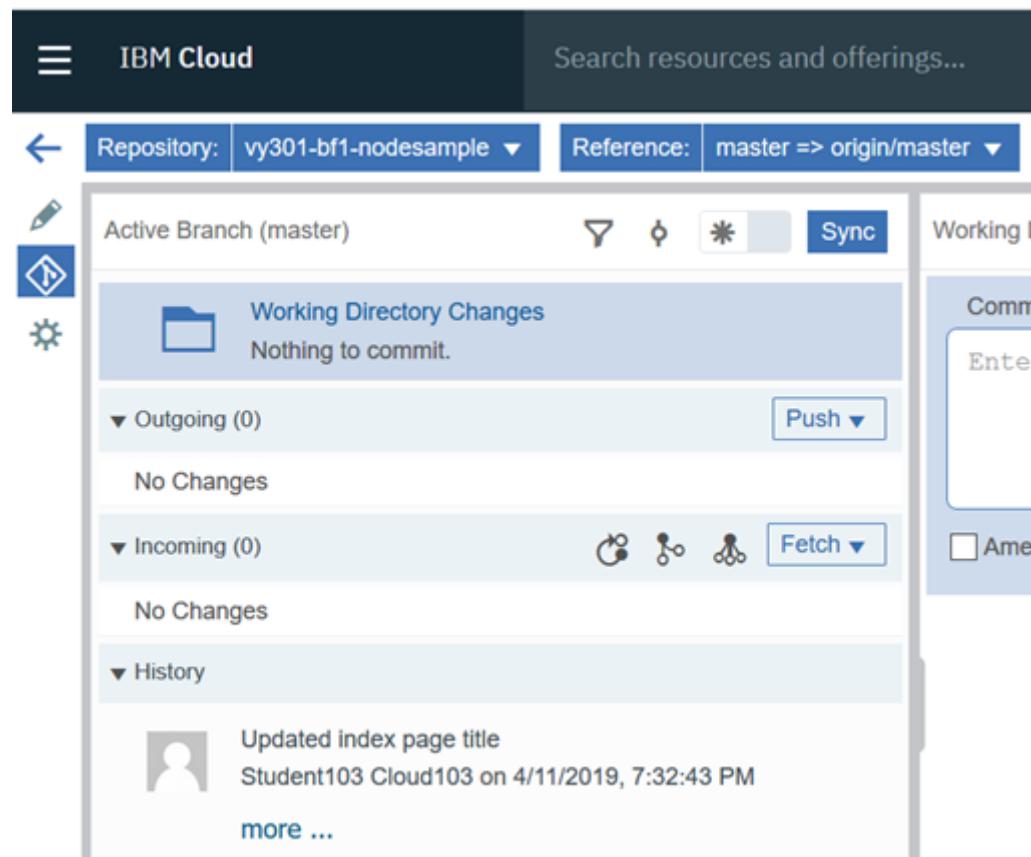
Do **not** perform this step; it is for your information only. You can revert to the last saved version of your source code in the Git repository. To do so, select the change set that you want to revert in the Outgoing section and hover your cursor over the eraser icon, as shown in the following figure.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

The screenshot shows the IBM Cloud Git interface. At the top, there's a navigation bar with 'IBM Cloud', a search icon, 'Catalog', 'Docs', and 'Support'. Below that is a header with 'Repository: vy301-bf1-nodesample' and 'Reference: master => origin/master'. On the left, there's a sidebar with icons for file operations like edit, clone, and push/pull. The main area has tabs for 'Active Branch (master)' and 'Working Directory Changes' (which shows 'Nothing to commit'). The 'Outgoing' tab is selected, showing one item: 'Updated [User] Cloud on 4/11/2019, 7:02:46 PM'. A tooltip for the eraser icon next to this item says 'Revert this commit, keeping all changed files and not making any changes to the working directory.' To the right of the outgoing list is a 'Commit message:' field with placeholder text 'Enter the commit message' and a checkbox for 'Amend previous commit'. The 'Incoming' and 'History' tabs are also visible but empty.

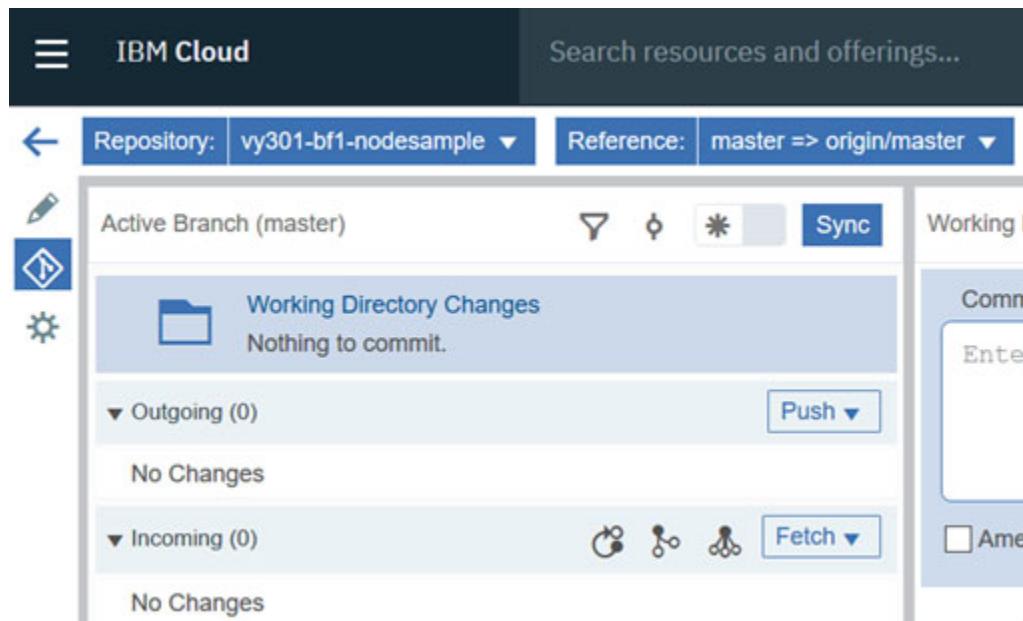
- 
- f. Push your changes from the local branch to the remote Git branch by clicking the **Push** drop-down menu and selecting **Push All** in the Outgoing pane, as shown in the following figure.



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.



- \_\_\_ 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.

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:

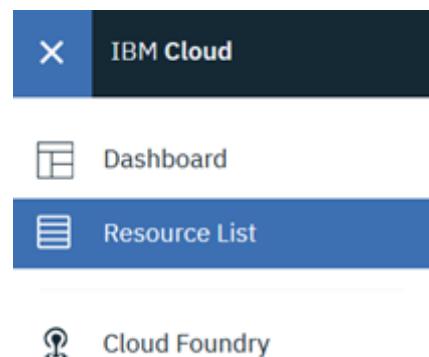
- Deploy from the Git repository. You can use Delivery Pipeline to automate the application deployment when changes are pushed to Git.
- Deploy directly from your project workspace in Eclipse Orion.

**Note**

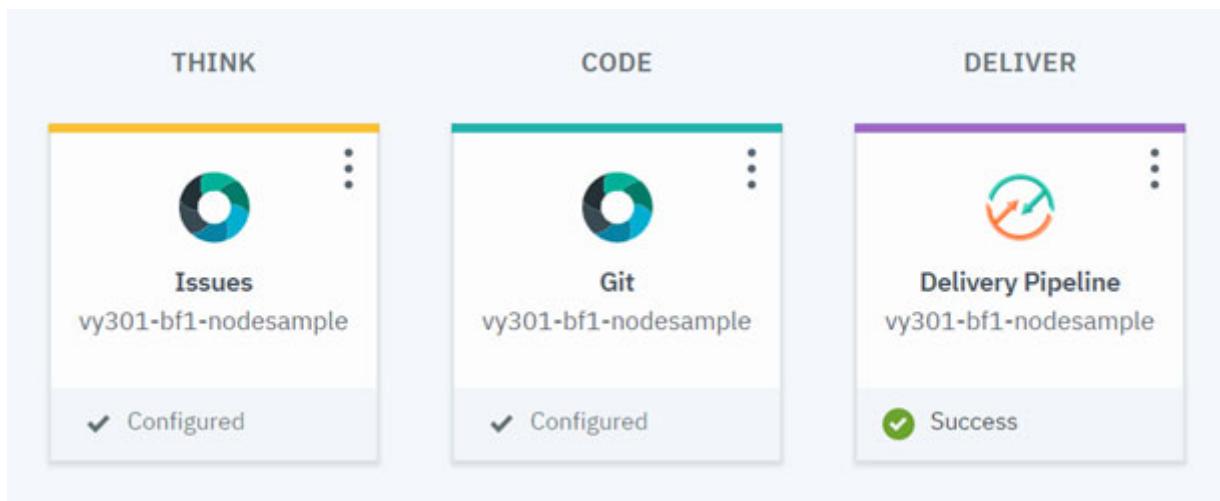
If more than one developer is working on a project, consider checking in your changes and pushing the most recent revision from the Git repository.

- 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

The screenshot shows the delivery pipeline interface for the 'vy301-bf1-nodesample' application. It consists of two main sections: 'Build Stage' on the left and 'Deploy Stage' on the right, separated by a large right-pointing arrow.

**Build Stage:**

- STAGE PASSED:** A green header bar.
- LAST INPUT:** Shows a GitHub icon and the text "Last commit by Student103 Clo... 47m ago". Below it is a link "Updated index page title".
- JOB:** Shows a green checkmark icon next to "Build Passed 38m ago".
- LAST EXECUTION RESULT:** Shows a build icon next to "Build 1" and a refresh/circular arrow icon.

**Deploy Stage:**

- STAGE PASSED:** A green header bar.
- LAST INPUT:** Shows the text "Stage: Build Stage / Job: B...".
- JOB:** Shows a green checkmark icon next to "Deploy Passed 37m ago".
- LAST EXECUTION RESULT:** Shows the text "vy301-bf1-nodesample" and a link "View console". Below it is another build icon next to "Build 1" and a refresh/circular arrow icon.

- 2. Examine the updated code:
- a. In Deploy Stage, click **View console** to view the application overview, as shown in the following figure.

The screenshot shows the 'Deploy Stage' section of the IBM Cloud Continuous Delivery interface. At the top, there are two circular icons: a play button and a gear. Below them is a green bar with the text 'STAGE PASSED'. Underneath, it says 'LAST INPUT' and 'Stage: Build Stage / Job: B...'. A card labeled 'Build 1' with a stack icon is shown. Below this, under 'JOBS', is a card for 'Deploy' which has passed 37 minutes ago. A link 'View logs and history' is available. In the 'LAST EXECUTION RESULT' section, a box highlights 'vy301-bf1-nodesample' and a 'View console' link with a hand cursor icon pointing at it. A 'Build 1' card with a stack icon is also present.



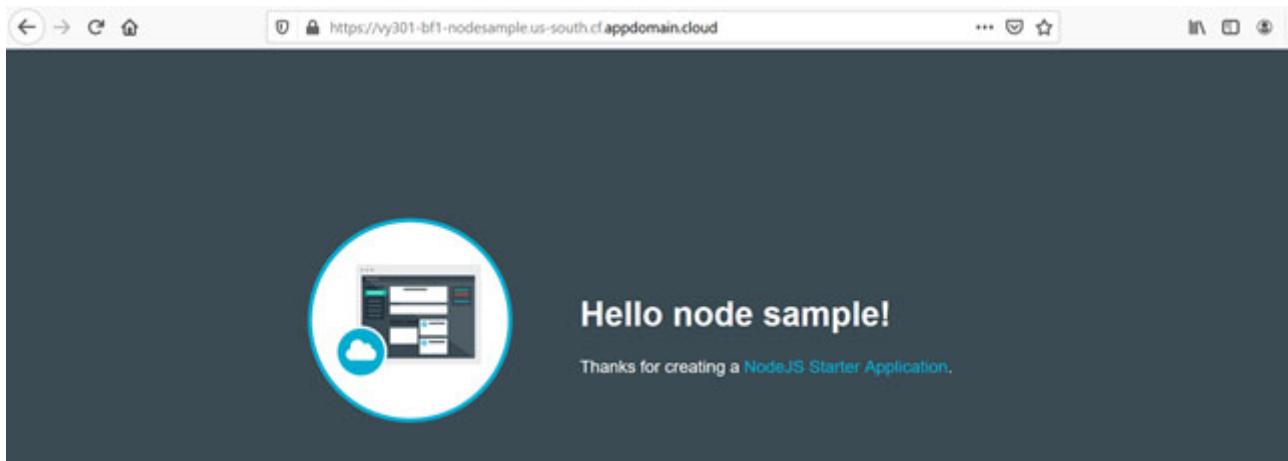
### Note

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.

The screenshot shows the IBM Cloud Resource list interface. On the left, there's a sidebar with links: 'Getting started', 'Overview', 'Runtime', and 'Connections'. The main area displays the application 'vy301-bf1-nodesample'. It features a circular icon with 'js' and a green status indicator. The application name is 'vy301-bf1-nodesample'. Below it, it says 'This app is awake.' and has a 'Visit App URL' link. Underneath, it shows 'Org: student103@yahoo.com', 'Location: Dallas', and 'Space: dev'. A horizontal bar at the bottom of the sidebar has 'Dashboard', 'Create', 'Analytics', and 'Logs' tabs.

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



## ***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          
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          
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.

The screenshot shows the IBM Cloud interface. At the top, there's a navigation bar with 'IBM Cloud', a search bar, and a 'Catalog' button. Below the navigation bar, the left sidebar shows a file structure: 'Root (cloud.ibm.co...)' and 'vy301-5b9-nodesample'. The 'vy301-5b9-nodesample' folder is expanded, revealing a 'README.md' file. The main content area displays the following text:

## 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>

Below this, there's a table header for a file list:

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:	<input checked="" type="checkbox"/> manifest.yml

---

**Manifest Settings**

Application Name*	vy301-bf1-nodesample
Host:	student103yahoomcom-48093ae05cf4151
Domain*	mybluemix.net

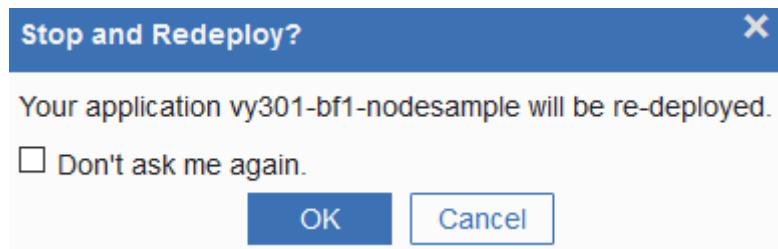
Shaded boxes indicate modified fields that will override manifest file settings.  
\* Denotes required field.

**Buttons:** Cancel, Save, Next >

- \_\_\_ 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.



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



- \_\_\_ 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.

- \_\_\_ 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.

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



### Note

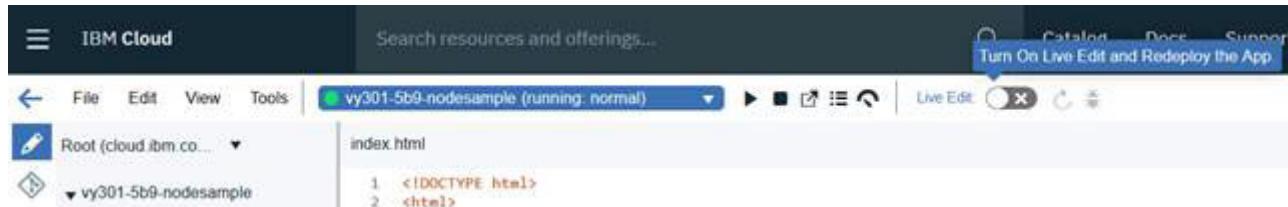
This part cannot be performed with an IBM Cloud Lite account. You need a Pay-As-You-Go account or a Subscription account. If you are a university student, you can apply for a no-charge feature code. Check with your instructor.

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.

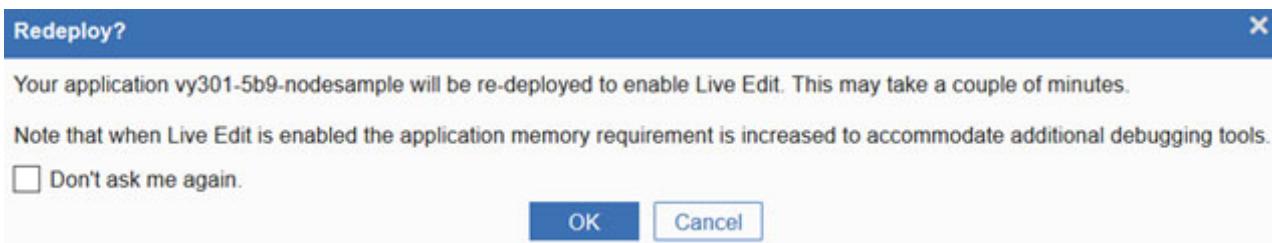
The [IBM Cloud Live Sync](#) feature automatically pushes any saved changes in the project workspace to a linked IBM Cloud application. For IBM SDK for Node.js applications, you can update static files and view the updates on IBM Cloud without restarting your application. Alternatively, for the non-static 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.



- 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.





## 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 X

- \_\_\_ 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.



- \_\_\_ 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.

```

12 <body>
13   <table>
14     <tr>
15       <td style= "width:30%;">
16         
17       </td>
18       <td>
19         <h1 id="message">Welcome to the live edit 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>
```

**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.

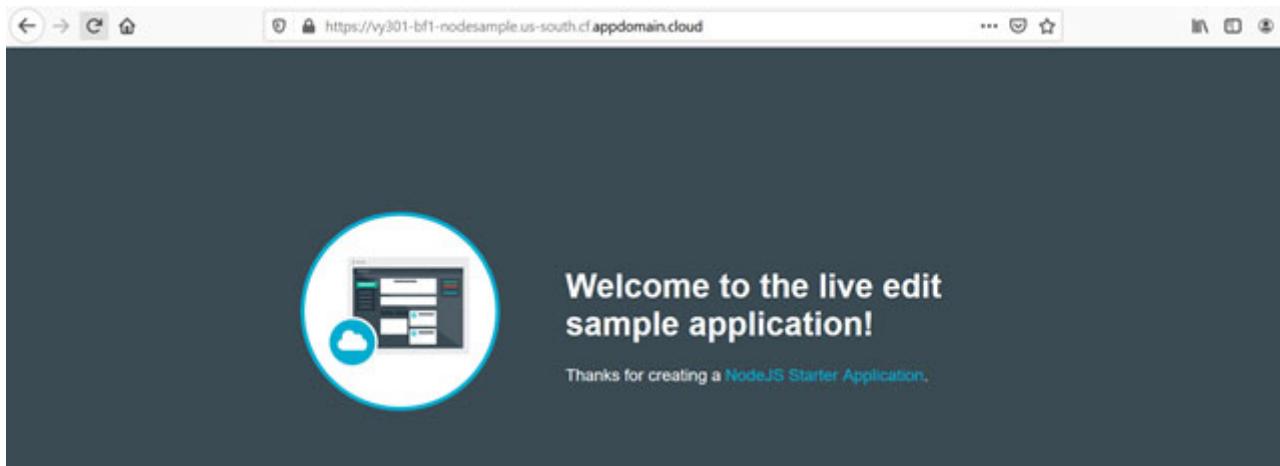
Examples of static files include HTML web pages and Cascading Style Sheets (CSS).

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.



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



## ***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**.

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

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

vy301-bf1-nodesample

Project

Repository

Issues

Merge Requests

Wiki

Snippets

**Settings**

General

Members

Integrations

Repository

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.

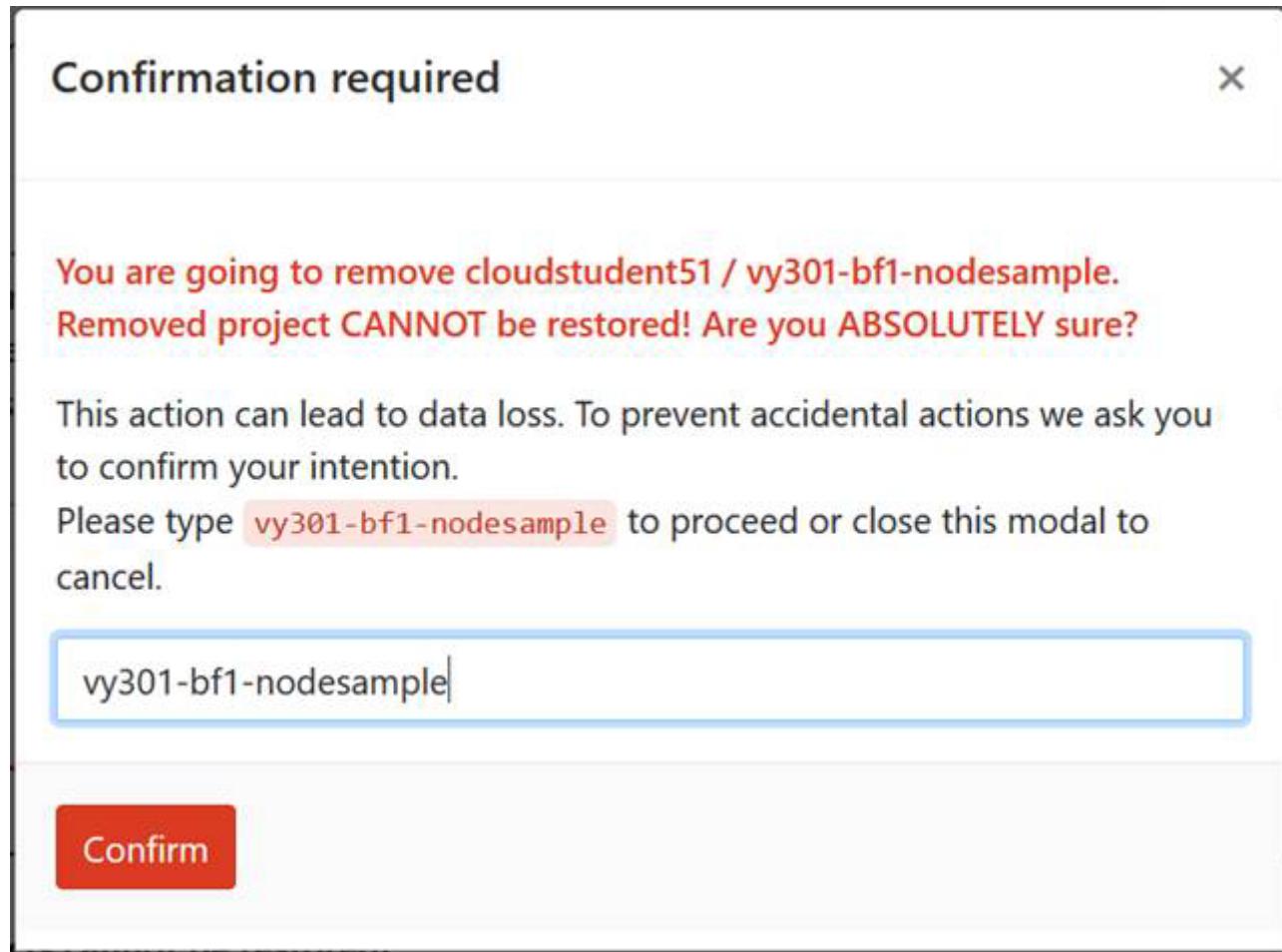
**Remove project**

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

**Removed projects cannot be restored!**

**Remove project**

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



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

The screenshot shows the IBM Cloud Toolchains interface for the application 'vy301-bf1-nodesample'. The application details are: Resource Group: Default, Location: Dallas, and Add tags. A context menu is open over the application name, with 'Delete' highlighted in red. Below the application details, there are three tool categories: THINK (Issues), CODE (Git), and DELIVER (Delivery Pipeline). Under the DELIVER category, there is an Eclipse Orion Web IDE tool. All tools are marked as 'Configured'.

- 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:

A screenshot showing a list of tool integrations that will be deleted. The list includes: Delivery Pipeline vy301-bf1-nodesample, Eclipse Orion Web IDE, Git vy301-bf1-nodesample, and Docker vy301-bf1-nodesample. Each item has a small icon to its left.

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.

A screenshot of the IBM Cloud navigation menu. The menu bar at the top says "IBM Cloud". Below it, there are two items: "Dashboard" and "Resource List".

- 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.

The screenshot shows the IBM Cloud dashboard interface. In the top navigation bar, there's a dropdown menu with the text 'IBM Cloud' and a gear icon. Below the navigation, there's a search bar with the placeholder 'Search services and resources'. Under the search bar, there's a 'Cloud Foundry Apps' section with a sub-section 'Cloud Foundry Services'. The 'Cloud Foundry Apps' section lists one app: 'vy301-bf1-nodesample' by 'student103@yahoo.com / dev' in the 'Dallas' space, which is currently 'Running'. To the right of the app name is a context menu with several options: 'Stop', 'Restart', 'Edit Name', 'Add tags', and 'Delete'. The 'Delete' button is highlighted with a red background.

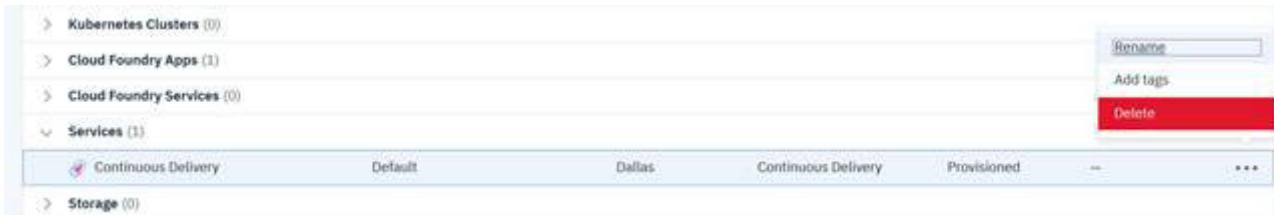
- 11. Confirm that you want to delete the routes and click **Delete**, as shown in the following figure.

## 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.

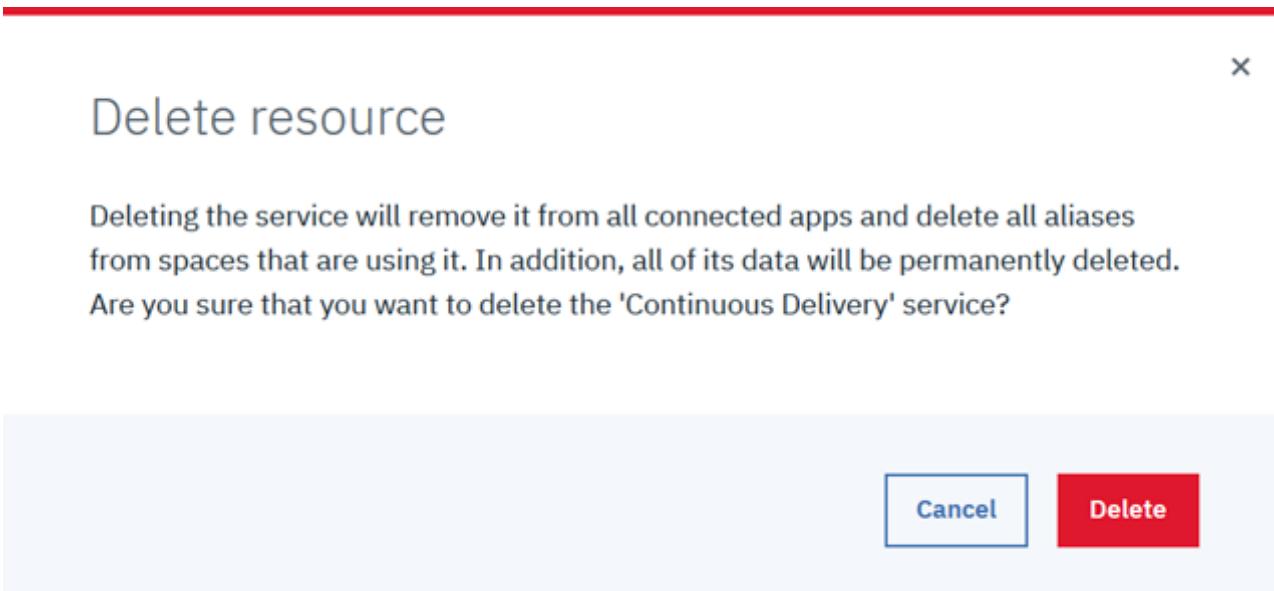
The screenshot shows a confirmation dialog box. At the top left, it says 'Routes'. The main text reads: 'Select the routes to be deleted when the app is deleted. Routes that are not deleted remain bound to the space, and only apps within the space will be able to use them.' Below this, there's a list of routes with checkboxes next to them, both of which are checked: 'vy301-bf1-nodesample.us-south.cf.appdomain.cloud' and 'vy301-bf1-nodesample.mybluemix.net'. At the bottom of the dialog are two buttons: 'Cancel' (in a blue box) and 'Delete' (in a red box).

- 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.



The screenshot shows the IBM Cloud Resource List interface. On the left, there's a sidebar with categories: Kubernetes Clusters (0), Cloud Foundry Apps (1), Cloud Foundry Services (0), Services (1), Storage (0). The 'Services' item is expanded, showing a single entry: 'Continuous Delivery'. To the right of the list, there's a toolbar with 'Rename', 'Add tags', and a red 'Delete' button. Below the list, there are tabs for Default, Dallas, Continuous Delivery, Provisioned, and three more tabs represented by ellipses (...).

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



- 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.

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

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

---

# Exercise 3. IBM Cloud with IBM Cloudant

## Estimated time

00:45

## Overview

NoSQL databases are built from the ground up to scale globally, run non-stop, and handle a wide variety of data types, such as JavaScript Object Notation (JSON), full-text, and geospatial. Cloudant is a NoSQL database that is optimized for handling heavy workloads of concurrent reads and writes in the cloud. These workloads are typical of large, fast-growing web, and mobile applications (apps).

This exercise shows you how to create an instance of a Cloudant service on IBM Cloud. You use your Cloudant service instance to discover various features of the Cloudant service and understand the different methods that are available to use the HTTP APIs by using IBM Cloud Identity and Access Management. You also learn how to use Cloudant HTTP APIs to apply create, read, update, and delete operations by creating indexes and by using Cloudant Query on Cloudant documents.

## Objectives

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

- Create an instance of the Cloudant service on IBM Cloud.
- Create service credentials by using IBM Cloud Identity and Access Management (IAM).
- Access the Cloudant documentation.
- Explore the features of the Cloudant dashboard.
- Create, read, update, and delete Cloudant documents by using HTTP APIs.
- Verify the data that is stored in the database by using the Cloudant dashboard.
- Create indexes and query Cloudant documents by using HTTP APIs.

## Introduction

Cloudant is built on Apache CouchDB and contributes to the open source project.

IBM Cloudant is a NoSQL database as a service (DBaaS) that frees developers from worrying about managing the database and enables them to focus on the application.

Cloudant is designed to ensure that the flow of data between an application and its database remains uninterrupted and performs to users' satisfaction. Cloudant can run across many servers in

a database cluster, resulting in high availability (HA) and fault tolerance. The data replication technology also enables developers to put data closer to where their applications need it most.

Your app's data persistence layer can be durable and highly available with IBM Cloudant. Your data is securely hosted and globally managed by big data experts 24x7.

This exercise demonstrates how you can create a Cloudant database service on IBM Cloud without installing or configuring the database instance on your workstation. You use an HTTP API client such as Postman to create, read, update, and delete Cloudant documents. You learn how to create indexes and query data by using Cloudant API endpoints.

# Requirements

This exercise requires:

- Access to the internet and the IBM Cloud console from your workstation by using one of the modern web browsers with the current version (such as Chrome, Firefox, or Safari).
- An IBM Cloud account (Lite, Pay-As-You-Go, or Subscription).
- Postman.
- IBM Cloud CLI that is installed in your workstation (optional).
- Any text editor, such as Notepad, Notepad++, VS code, or Sublime (optional).

## Exercise instructions

In this exercise you will complete the following tasks:

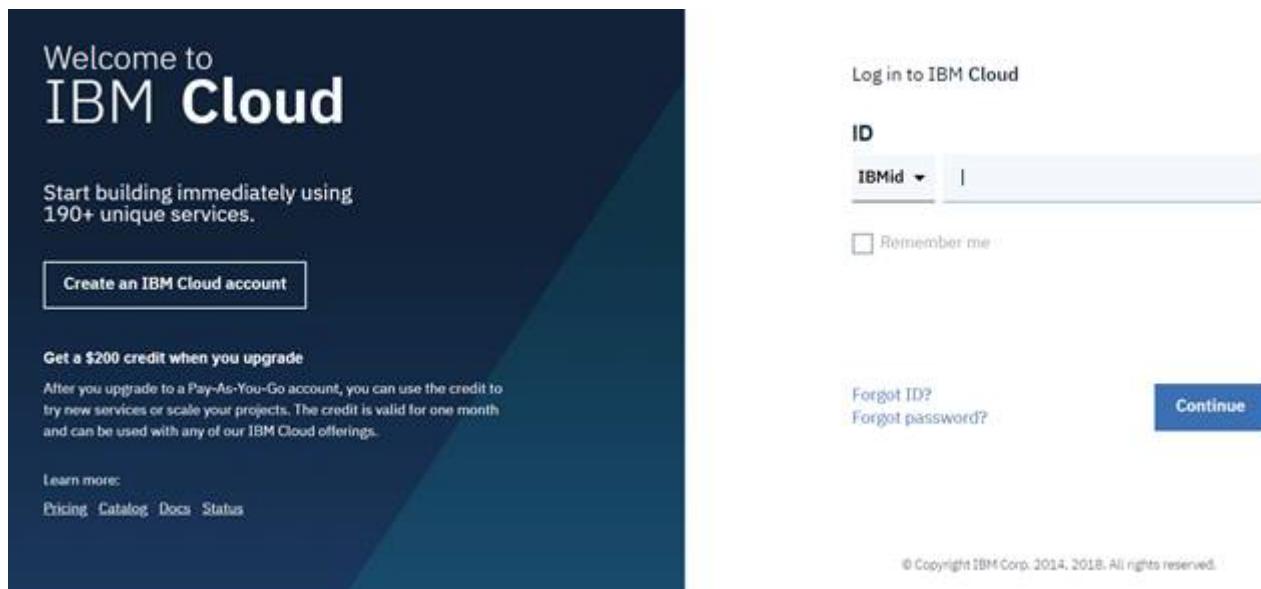
- \_\_\_ 1. Log in to IBM Cloud.
- \_\_\_ 2. Create an instance of the Cloudant service.
- \_\_\_ 3. Explore the Cloudant service on IBM Cloud.
- \_\_\_ 4. Create credentials for a Cloudant service instance.
- \_\_\_ 5. Explore the features of the Cloudant dashboard.
- \_\_\_ 6. Use an HTTP API client to access Cloudant.
- \_\_\_ 7. Delete your Cloudant service instance.

### Part 1: Logging in to IBM Cloud

In this section, you log in to your IBM Cloud account by providing your own credentials.

Complete the following steps:

- \_\_\_ 1. Log in to your IBM Cloud account:
  - \_\_\_ a. From your web browser, go to <https://cloud.ibm.com/login>.
  - \_\_\_ b. Enter your IBMid, which is the email that you used to register to IBM Cloud, and click **Continue**.
  - \_\_\_ c. Enter your password and then click **Log in**, as shown in the following figure.



**Note**

If your company uses the single sign-on service, you provide only your IBMid, and then you are redirected to the single sign-on page to log in by using your company credentials.

- 2. You are redirected to the IBM Cloud dashboard page, as shown in the following figure.

The screenshot shows the IBM Cloud dashboard. At the top, there's a navigation bar with 'IBM Cloud', a search bar, and links for Catalog, Docs, Support, and Manage. Below the navigation is a 'Dashboard' header. The main area has several sections: 'Resource summary' (with links to 'What is a resource?', 'Regions for resource deployment', and 'Best practices for organizing resources in resource group'), 'Planned maintenance' (listing 'Next event: Tue, Dec 11 2018 New worker node creation'), 'Location status' (showing 'Asia Pacific' with a green checkmark), 'Apps' (with a laptop icon), and 'Support cases' (with a support ticket icon). A large blue 'Create resource' button is located at the top right of the dashboard area.

## **Part 2: Creating an instance of the Cloudant service**

Cloudant is a document-oriented DBaaS. Cloudant is available as a service with different plans over IBM Cloud: Lite, Standard, and Dedicated Hardware. This section shows you how to create an instance of the Cloudant service on IBM Cloud. You use this database service in the next part to create, read, update, and delete data.

Complete the following steps:

- 1. On the IBM Cloud dashboard, click **Create resource** at the upper right, as shown in the following figure.

The screenshot shows the IBM Cloud dashboard with the 'Create resource' button highlighted in blue. Below it, there are three tabs: 'View resources', 'Planned maintenance', and 'View events'. A banner at the bottom indicates 'Next event: Tue, Aug 08 2018'.

- 2. You can now see the entire catalog. In the search field, type **Cloudant**.
- 3. Select the **Cloudant** service, as shown in the following figure.

## Catalog

The screenshot shows the IBM Cloud Catalog interface. At the top, there is a search bar with the text "Cloudant" and a "Filter" button. Below the search bar, a sidebar on the left lists "All Categories (1)" and various service categories: Compute, Containers, Networking, Storage, AI, Analytics, Databases (1), and Developer Tools. The main area is titled "Databases" and displays a single service entry for "Cloudant". The entry includes a blue circular icon, the name "Cloudant", and the text "Lite • IBM • IAM-enabled". A brief description follows: "A scalable JSON document database for web, mobile, IoT, and serverless applications." On the far right, there is a vertical "FEEDBACK" button.

- \_\_\_ 4. To complete the creation of the Cloudant service instance:
  - \_\_\_ a. Accept the default value for region and keep the **Lite** price plan as the selected plan as shown in the following figure.

The screenshot shows the "Create" page for the Cloudant service. At the top, it says "Cloudant" with tabs for "Lite", "IBM", "Service", and "IAM-enabled". It also shows "Author: IBM" and "Date of last update: 10/29/2019". On the right, there are links for "Need Help?", "Contact Support", and "View docs". Below that, there are "Create" and "About" buttons, with "Create" being underlined. The main section is titled "Select a region" and shows a dropdown menu with "London" selected. Below that, it says "Select a pricing plan" and notes "Displayed prices do not include tax. Monthly prices shown are for country or region: United States". A table compares different plans:

PLAN	FEATURES	PRICING
<input checked="" type="checkbox"/> Lite	Limited throughput & storage Capped at 1GB of data storage Provisioned throughput capacity fixed at: 20 reads/sec, 10 writes/sec, 5 global queries/sec Max JSON document size of 1MB	Free

A note below the table states: "The Lite plan provides access to the full functionality of Cloudant for development and evaluation. The plan has a set amount of provisioned throughput capacity as shown and includes a max of 1GB of encrypted data storage. Users exceeding data storage limit will have instances disabled for writes until total storage is below 1GB or instance is upgraded to Standard plan. Users are limited to one Lite plan instance per service." At the bottom, it says "Lite plan services are deleted after 30 days of inactivity."

- \_\_\_ b. Accept the default values for service name and resource group.
- \_\_\_ c. In **Available authentication methods**, select **Use only IAM**, as shown in the following figure.

Configure your resource

Service name:	Select a resource group: <a href="#">1</a>
Cloudant-fs	Default
Tags: <a href="#">1</a>	
Examples: env:dev, version:1	
<b>Available authentication methods:</b>	
Legacy credentials enable login to Cloudant using HTTP Basic authentication	
<a href="#">Use only IAM</a> <a href="#">Use both legacy credentials and IAM</a> <b><a href="#">Use only IAM</a></b>	



### Note

The Cloudant service has two authentication methods that are available: **Use both legacy credentials and IAM**, which means that both IAM and Legacy credentials can be used to access the account, and **Use only IAM**, which means that only IAM credentials are provided through service binding and credential generation.

- Legacy credentials enable login to Cloudant by using HTTP Basic authentication by providing a user name and password that is used for authentication.
- Identity and Access Management (IAM) provides a unified approach to managing user identities, services, and access control. IAM authentication requires that an IAM API key is exchanged for a time-limited access token before you can make a request to Cloudant. When the access token expires, the client must obtain a new one from the IAM token service.

In this exercise, you use the IAM method as the authentication method for the Cloudant service.

- 
- \_\_\_ d. Click **Create**, as shown in the following figure.

## Summary

**Cloudant** *Free*

Region: London  
Plan: Lite  
Service name: Cloudant-fs  
Resource group: Default

Create FEEDBACK

Add to estimate

[View terms](#)

- 5. After the service is created, you are redirected to the Resource list window, as shown in the following figure. Open the Cloudant service instance overview by clicking the newly created service in the Resource list window.

## Resource list

						Create resource
						Collapse all   Expand all
Name	Group	Location	Status	Tags		
<input type="text"/> Filter by name or IP address...	<input type="text"/> Filter by group or org...	<input type="text"/> Filter...	<input type="text"/> Filter...	<input type="text"/> Filter...		
<span style="font-size: 1.2em;">▼</span> Devices (0)						
<span style="font-size: 1.2em;">▼</span> VPC Infrastructure (0)						
<span style="font-size: 1.2em;">▼</span> Kubernetes Clusters (0)						
<span style="font-size: 1.2em;">▼</span> Cloud Foundry Apps (0)						
<span style="font-size: 1.2em;">▼</span> Cloud Foundry Services (0)						
<span style="font-size: 1.2em;">▼</span> Services (1)						
Cloudant-fs	Default	London	Provisioned	--		***
<span style="font-size: 1.2em;">▼</span> Storage (0)						
<span style="font-size: 1.2em;">▼</span> Cloud Foundry Enterprise Environments (0)						

**Note**

After you create the service, it takes a few minutes to be provisioned, as shown in the following figure. The status is “Provision in progress” for a while. If it takes too long for the status to change to “Provisioned”, refresh the browser.

Resource list

[Create resource](#)

Collapse all | Expand all

Name ▲	Group	Location	Status	Tags
<input type="text"/> Filter by name or IP address...	<input type="text"/> Filter by group or org...	<input type="text"/> Filter...	<input type="text"/> Filter...	<input type="text"/> Filter...
<span>▼ Devices (0)</span> <span>▼ Kubernetes Clusters (0)</span> <span>▼ Cloud Foundry Apps (0)</span> <span>▼ Cloud Foundry Services (0)</span> <span>▼ Services (3)</span> <ul style="list-style-type: none"> <li><span>Cloudant-Is</span> Default London Provision in progress ...</li> </ul> <span>▼ Storage (0)</span> <span>▼ Cloud Foundry Enterprise Environments (0)</span>				

### **Part 3: Exploring the Cloudant service on IBM Cloud**

The Cloudant service instance has an overview page on IBM Cloud where you can perform the following actions:

- Access the Cloudant dashboard.
- Create credentials to connect manually to an external consumer.
- Connect automatically with an internal IBM Cloud application.
- Upgrade the plan.
- Manage a Cloudant instance and perform many other actions that are related to the Cloudant service on IBM Cloud.
- In this section, you discover these capabilities.
- Complete the following steps:

- 1. In the **Resource list** under **Services**, click the newly created Cloudant service instance.

Resource list

[Create resource](#)

[Collapse all](#) | [Expand all](#)

Name	Group	Location	Status	Tags
<input type="text"/> Filter by name or IP address...	<input type="text"/> Filter by group or org...	<input type="button"/> Filter...	<input type="text"/> Filter...	<input type="button"/> Filter...
<ul style="list-style-type: none"> <li><a href="#">Devices (0)</a></li> <li><a href="#">VPC Infrastructure (0)</a></li> <li><a href="#">Kubernetes Clusters (0)</a></li> <li><a href="#">Cloud Foundry Apps (0)</a></li> <li><a href="#">Cloud Foundry Services (0)</a></li> <li><a href="#">Services (1)</a> <ul style="list-style-type: none"> <li><a href="#"> Cloudant-fs</a> Default London Provisioned -- ***</li> </ul> </li> <li><a href="#">Storage (0)</a></li> <li><a href="#">Cloud Foundry Enterprise Environments (0)</a></li> </ul>				

- 2. After you open the Cloudant service instance, you are redirected to the overview page of the service. In the left pane are the **Manage**, **Service credentials**, **Plan**, and **Connections** menu options, as shown in the following figure.

Resource list /

**Cloudant-fs**

Resource group: Default Location: London [Add Tags](#)

[Overview](#) [Dashboard](#) [Capacity](#) [Docs](#) [Launch Cloudant Dashboard](#)

**Deployment details**

CRN	crn:v1:bluemix:public:cloudantnosqldb:eu-gb:a/253673fed66746c2a4e47e3e6bbbeddf:ecf4f9ec-46cf-449f-8d59-4a63a72e3274::
Location	London
External Endpoint	<a href="https://e6331570-ede2-4e07-87e6-0fbeab4ec5d3-bluemix.cloudant.com">https://e6331570-ede2-4e07-87e6-0fbeab4ec5d3-bluemix.cloudant.com</a>

- 3. The Manage section has four tabs: Overview, Dashboard, Capacity, and Docs, which are in the right pane. The Manage section also includes the **Launch Cloudant Dashboard** button, which opens the Cloudant dashboard.
- a. The Overview tab (shown in the following figure) shows data about the current service, such as the deployment details, which include the location, endpoint, and authentication method that is used. It also shows capacity details, which include the current plan and the available storage. It also includes a section for important announcements that are related to Cloudant, such as showing new features and updates.

Resource list /

## Cloudant-fs

Resource group: Default Location: London Add Tags

Overview Dashboard Capacity Docs Launch Cloudant Dashboard

**Deployment details**

CRN	crn:v1:bluemix:public:cloudantnosqldb:eu-gb:a/253673fed66746c2a4e47e3e6bbbedd:39667b9b-17ec-451a-8ad9-e02713658603::
Location	London
External Endpoint	<a href="https://33915899-7504-454f-8690-19a1403c0adf-bluemix.cloudant.com">https://33915899-7504-454f-8690-19a1403c0adf-bluemix.cloudant.com</a>
External Endpoint (preferred)	<a href="https://33915899-7504-454f-8690-19a1403c0adf-bluemix.cloudantnosqldb.appdomain.cloud">https://33915899-7504-454f-8690-19a1403c0adf-bluemix.cloudantnosqldb.appdomain.cloud</a>
Authentication methods	IBM Cloud IAM
Activity Tracker event types	Management
Disk encryption	Yes

**Capacity details**

You are currently subscribed to the Lite plan.

Upgrade

Reads	20 per second
Writes	10 per second
Global queries	5 per second
Storage	0.00GB of 1GB

**Announcements**

**Partition Queries is now GA**  
IBM Cloudant released a new capability for partitioned databases which will require both a partition key and doc key for every document stored. Users can leverage partition queries against a partitioned database which are up to 20X cheaper than global queries with reduced latency. See the [documentation](#) for more details. The partition query feature has been rolled out to all Cloudant environments and is now officially GA.

**TLS 1.2+ required starting on June 1, 2019**  
Starting on June 1, 2019, the IBM Cloudant API will require Transport Layer Security (TLS) 1.2 and above. The IBM Cloudant API requires HTTPS and currently supports TLS 1.0 and above. The IBM Cloud [will retire support for TLS 1.0 and 1.1 for many services in March 2018](#) and IBM Cloudant is following suit. If your applications are still using TLS 1.0 or 1.1, please update them immediately to TLS 1.2 or above to avoid any impact. If you are unsure what TLS version your application is using to connect to Cloudant or have any questions, open a support ticket. See IBM Cloudant security docs for more detail.

**IBM Identity & Access Management (IAM)**  
IBM Cloudant instances in the IBM Cloud now support IBM IAM for authentication and authorization. IBM IAM allows for centralized control for granting user and programmatic access to all the Cloudant instances in your IBM Cloud account. See the [IAM documentation](#) for a comparison between using IAM and legacy Cloudant authentication for your applications.

In addition, IBM Cloudant instances support Resource Groups and we encourage all customers to migrate their Cloudant instances that remain in a Cloud Foundry space to Resource Groups to take advantage of the security access benefits. See the [docs](#) on using Cloudant with Resource Groups for more details.

**IBM Cloudant Dedicated Hardware environment security enhancements**  
The IBM Cloudant Dedicated Hardware plan in the IBM Cloud catalog is for customers who require additional security and compliance benefits. IBM Cloudant Dedicated Hardware instances are isolated environments for the sole use of a customer to run one or more IBM Cloudant Standard plan instances. We have added the ability to connect via IBM's internal network, meaning customers can avoid public network and outbound bandwidth costs when connecting to Cloudant from their applications. With that, Dedicated Hardware environment now offer the following benefits:

- Isolation at the database compute and storage layers for a single customer
- Choice of any IBM Cloud location
- Optional HIPAA readiness in US locations
- Both external and internal service endpoints
- IP whitelisting
- Bring your own encryption keys (BYOK) with Key Protect

- \_\_\_\_\_ b. The Dashboard tab (shown in the following figure) includes quick access to specific sections in the Cloudant dashboard:

Databases, where you can create, update, and delete databases.

Replication, where you can create or edit existing replication jobs.

Active tasks, where you can see indexing, replication, and compaction tasks that are actively running on your Cloudant instance.

Monitoring, where you can see the current consumption of provisioned throughput capacity and data storage that is being used by your applications.

Resource list /

**Cloudant-fs**

Resource group: Default Location: London [Add Tags](#)

[Launch Cloudant Dashboard](#)

[Overview](#) [Dashboard](#) [Capacity](#) [Docs](#)

**Databases**  
Create, update, and delete your IBM Cloudant databases in your instance. Click into each database to view JSON documents, set permissions, and to create, edit, and query IBM Cloudant indexes. [Launch](#)

**Replication**  
Create new or edit existing replication jobs. See a status of deployed replication jobs to see if they completed or review replication docs for troubleshooting purposes. [Launch](#)

**Active tasks**  
See indexing, replication, and compaction tasks that are actively running on your IBM Cloudant databases. [Launch](#)

**Monitoring**  
View the current consumption of provisioned throughput capacity and data storage being used by your application(s) against the IBM Cloudant instance. Provisioned throughput capacity is broken up between reads/sec, writes/sec, and global queries/sec. [Launch](#)

- \_\_\_ c. The Capacity tab shows the current capacity that is available for your application and the current plan that is used, as shown in the following figure.

The screenshot shows the 'Capacity' tab of the Cloudant resource details page. At the top, it says 'You are currently subscribed to the Lite plan.' Below this, there are two tables: 'Current Capacity' and 'Target Capacity'. Both tables have three rows: 'Reads per second' (20), 'Writes per second' (10), and 'Global queries per second' (5). A central column between the tables contains a 'Upgrade to Change Capacity' button with a checkmark icon. Below the tables, it says 'Free Cost /month'. At the bottom, there's a 'Select capacity' dropdown set to '1', an 'Upgrade to Standard Plan' button, and two links: 'Need additional capacity?' and 'Increase capacity through API'.

Current Capacity		Upgrade to Change Capacity	Target Capacity	
20	Reads per second		20	Reads per second
10	Writes per second		10	Writes per second
5	Global queries per second	5	Global queries per second	
<b>Free</b> <b>Cost /month</b>		<b>Free</b> <b>Cost /month</b>		

Select capacity  Upgrade to Standard Plan

- › Need additional capacity?
- › Increase capacity through API

- \_\_\_ d. The Docs tab includes a direct link for Cloudant documentation, some tutorials about Cloudant, and open source libraries for connecting with Cloudant for all major development languages, as shown in the following figure.

The screenshot shows the IBM Cloudant documentation page. At the top, there are tabs for Overview, Dashboard, Capacity, and Docs (which is selected). Below the tabs, there are several sections: 'Cloudant Documentation' (describing Cloudant's API and querying), 'Supported Client Libraries' (listing libraries for various languages), 'Learning Center' (with a link to videos and tutorials), 'Plans & Provisioning' (listing IBM Cloudant plans, pricing, and provisioning), 'Connecting/Authenticating' (listing IBM Identity and Access Management (IAM) and Cloudant Legacy Authentication), 'Replication' (listing Overview, API, and Advanced Topics), and 'Indexing and Querying' (listing Cloudant Query, MapReduce Views, Search, and Geospatial).

- \_\_\_ e. In the Docs tab, select **Learning Center** or alternatively, from another browser tab, go to the following address:

<https://developer.ibm.com/clouddataservices/docs/cloudant/>

- \_\_\_ f. In the IBM Cloudant Learning Center window, click **Load Data**, as shown in the following figure.

The screenshot shows the IBM Watson and Cloud Learning Center. On the left, there is a sidebar with categories: IBM Watson OpenScale, IBM Analytics Engine, IBM Streaming Analytics, IBM Cloud SQL Query, IBM Lift, and IBM Cloudant (which is expanded, showing sub-links: Get started, Load Data, HTTP API, Replication, Indexes and Queries, Cloudant Query, Views, and Search). On the right, there is a main content area with a heading 'place? Simple: data replication & synchronization.' and a paragraph explaining Cloudant's CouchDB-style replication and sync capabilities. Below this is a video player for a video titled 'IBM Cloudant overview, NoSQL database-as-a-service...' with options to 'Watch later' and 'Share'. The video thumbnail features the IBM Cloudant logo.

You can also get to this page by going to the following address:

<https://developer.ibm.com/clouddataservices/docs/cloudant/load-data/>

- \_\_\_ g. Read the entire document, including the tutorials, and watch all the videos in this page. You do not need to do the tutorials. When you finish reading this document, close the browser tab for IBM Cloudant dashboard.
- \_\_\_ 4. In the left pane of the Cloudant service overview page in IBM Cloud, explore the other sections:  
**Service credentials**, where you can create a set of credentials so that an external consumer can use the Cloudant Rest APIs, as shown in the following figure.

**Plan**, where you can upgrade your current plan, as shown in the following figure.

PLAN	FEATURES	PRICING
<input checked="" type="checkbox"/> Lite	Limited throughput & storage Capped at 1GB of data storage Provisioned throughput capacity fixed at 20 reads/sec, 10 writes/sec, 5 global queries/sec Max JSON document size of 1MB	Free
<input type="checkbox"/> Standard	Serverless scaling of throughput & storage Includes 20 GB of free data storage; additional storage metered Users can adjust provisioned throughput capacity in blocks of 100 reads/sec, 50 writes/sec, 5 global queries/sec Max JSON document size of 1MB	\$1.00 USD/Gigabyte-Months \$0.25 USD/Events per Second \$0.50 USD/Events per Second \$5.00 USD/Events per Second
<input type="checkbox"/> Dedicated Hardware	Add-on premium for a single tenant environment Available in any IBM Cloud location GDPR readiness option for US locations IP whitelisting Customer-managed encryption keys with Key Protect Internal endpoints (Provisioned in 2019 or later)	\$6,000.00 USD/Instance

**Connections**, where you can connect the service with any existing Cloud Foundry applications on the same organization.

The screenshot shows the Cloudant service Overview page. On the left sidebar, the 'Connections' tab is selected. The main area displays a message: 'No connected Cloud Foundry applications' with a 'Create connection' button. The top navigation bar includes 'Resource list / Cloudant-fs', 'Resource group: Default', 'Location: London', and 'Add Tags'. A large question mark icon is centered on the page.

## Part 4: Creating credentials for a Cloudant service instance

To connect your Cloudant service with external consumers such as external applications or REST API clients, you must create credentials. In this section, you create a set of credentials for the Cloudant service that will be used in the next sections to submit HTTP API calls from an HTTP API client.

Complete the following steps:

- \_\_\_ 1. From the Cloudant service Overview page, select **Service credentials** from the left pane, as shown in the following figure.

The screenshot shows the Cloudant service Overview page. The 'Service credentials' tab is selected on the left sidebar. The main area displays deployment details: CRN (cnv1:bluemixpublic:cloudantnosqldb:gb:a/253673fed66786c2a4e47e3e6bbbedf:39667b9b-17ec-451a-8a29-e32713658603::), Location (London), and External Endpoint (<https://33915899-7504-4541-8690-19a1d03cf0aff-bluemix.cloudant.com>). A 'Launch Cloudant Dashboard' button is also present.

- \_\_\_ 2. In the Service credentials page, click **New credential +** as shown in the following figure.

The screenshot shows the IBM Cloudant service credentials management interface. On the left sidebar, there are links for Manage, Service credentials (which is selected), Plan, and Connections. The main area displays a resource named "Cloudant-fs" under the "Resource list". It shows the "Resource group: Default" and "Location: London" with a "Add Tags" button. A "Service credentials" section contains a note about JSON format and API keys, with a "Learn more" link. Below this is another "Service credentials" section with a "New credential" button and a note to click it to create a set of credentials. The overall interface is clean with a light blue and white color scheme.

- 3. Keep the default values that are shown in the window and click **Add**, as shown in the following figure.

The screenshot shows the "Add new credential" dialog box. It has fields for "Name" (set to "Service credentials-1"), "Role" (set to "Manager"), and "Select Service ID (Optional)". The "Select Service ID..." dropdown is open. At the bottom, there are "Cancel" and "Add" buttons. The dialog box has a standard Windows-style border with a close button in the top right corner.



### Note

The Add new credential window includes the following fields:

- Role defines the actions that are permitted when the service is accessed. This field currently allows the Manager role only.
- Service ID identifies a service or application similar to how a user ID identifies a user. There are two options for creating service ID: Auto generating one or by creating a customized one.
- Add Inline Configuration Parameters (Optional) provides service-specific configuration parameters in a valid JSON object.

- 
- 4. After the set of credentials is created successfully, open the new service credentials by clicking **View credentials**, as shown in the following figure.

KEY NAME	DATE CREATED	ACTIONS
Service credentials-1	APR 12, 2019 - 11:41:36 PM	<a href="#">View credentials</a>

- 5. Click the **Copy to clipboard** icon on the right to copy the credentials data that is kept in JSON format, as shown in the following figure. Then, create a file on your favorite text editor, paste the credentials into the newly created file, save it in your local workstation, and name it `cloudant_credentials.json`.

The screenshot shows the Cloudant dashboard for a service named 'Cloudant-fs'. At the top, it displays 'Resource group: Default' and 'Location: London' with a 'Add Tags' button. Below this is a table titled 'Service credentials' with one item listed:

KEY NAME	DATE CREATED	ACTIONS
Service credentials-1	APR 12, 2019 - 11:41:36 PM	<a href="#">View credentials</a> <a href="#">Edit</a>

The 'View credentials' link is expanded, showing the following JSON data:

```
{
  "apikey": "H0eJpmgzwCvuo89TCdFCaRPZ-fYAhViVn6a4tJUAdaVd",
  "host": "7471c743-b853-429f-8367-bcfbe87fef0e-bluemix.cloudantnosqlb.appdomain.cloud",
  "iam_apikey_description": "Auto generated apikey during resource-key operation for Instance - crn:v1:bluemix:public:cloudantnosqlb:eu-gb:a/253673fed66746c2a4e47e3e6bbbeddf:4a4a725c-5e9d-4e02-ad5f-bca4845393e8::",
  "iam_apikey_name": "auto-generated-apikey-db884cf2-5b0c-49cc-9942-ca6eb94b02a0",
  "iam_role_crn": "crn:v1:bluemix:public:iam::::serviceRole:Manager",
  "iam_serviceid_crn": "crn:v1:bluemix:public:iam-identity::a/253673fed66746c2a4e47e3e6bbbeddf::serviceId:ServiceId-cd42a73b-6f32-4aa5-9459-3dfad5698c37",
  "url": "https://7471c743-b853-429f-8367-bcfbe87fef0e-bluemix.cloudantnosqlb.appdomain.cloud",
  "username": "7471c743-b853-429f-8367-bcfbe87fef0e-bluemix"
}
```

## Part 5: Exploring the features of the Cloudant dashboard

The Cloudant dashboard has a simple structure that you can use to discover and use Cloudant features. These features include the following ones:

- Database management, including database creation and deletions.
- Documents management: create, read, update, and delete (CRUD) operations, indexes, and Cloudant queries.
- Monitoring, including the operations activity and storage.
- Replication of both remote and local databases.

In this section, you explore and apply some of these features. You create a database, and then you apply CRUD operations to Cloudant documents. You also use Cloudant query to view Cloudant documents based on some specified conditions.

Complete the following steps:

1. From the left pane of the Cloudant instance service, select **Manage**, as shown in the following figure.

Resource list / Cloudant-fs

Resource group: Default Location: London Add Tags

Service credentials

Credentials are provided in JSON format. The JSON snippet lists credentials, such as the API key and secret, as well as connection information for the service.

KEY NAME	DATE CREATED	ACTIONS
Service credentials-1	NOV 30, 2019 - 02:18:13 PM	<a href="#">View credentials</a>

2. Click **Launch Cloudant Dashboard**, as shown in the following figure.

Resource list / Cloudant-fs

Resource group: Default Location: London Add Tags

Overview    Dashboard    Capacity    Docs    Launch Cloudant Dashboard

3. The Cloudant dashboard opens in a new browser tab, as shown in the following figure.

Databases

Your Databases

Name	Size	# of Docs	Partitioned	Actions
------	------	-----------	-------------	---------

- \_\_\_ 4. Create a database and name it **novels** by completing the following steps:
- \_\_\_ a. Click **Create Database** on the top bar, as shown in the following figure.



- \_\_\_ b. Name the database **novels**, select **Non-partitioned**, and then click **Create**, as shown in the following figure.

Name	Size	# of Docs	Partitioned

Showing 1–0 of 0 databases



### Note

While you are creating a database, there is the option to make the database partitioned. A partitioned database is a newer type of IBM Cloudant database. This option is an advanced feature, so it is out of the scope of this exercise. For more information about this topic, see the following link:

<https://cloud.ibm.com/docs/services/Cloudant/guides?topic=cloudant-database-partitioning#database-partitioning>

- \_\_\_ c. After you create the database, you are redirected to the new database overview, as shown in the following figure.

The screenshot shows the IBM Cloudant interface. At the top left, a green notification bar displays the message "Database created successfully". On the left side, there is a vertical sidebar with icons for different database operations: All Documents, Query, Permissions, Changes, Design Documents, and a user icon. The main area features a large blue button labeled "Create Document". In the center, there is a large cloud icon, and below it, the text "No Documents Found".

- \_\_\_ 5. Create a document for the “Oliver Twist” novel by completing the following steps:
- \_\_\_ a. Click **Create Document**, as shown in the following figure.

**Create Document**

- \_\_\_ b. You are redirected to the document editor page, which contains by default a JSON object with an auto-generated `_id`, as shown in the following figure.

The screenshot shows the "New Document" editor. At the top, it says "novels > New Document". There are buttons for "Create Document" (with a checked checkbox) and "Cancel". Below the buttons is a JSON editor. The JSON code is as follows:

```

1. {
2.   "_id": "dd4b9cf9a9e7aa6ac818ff4adb2c63f7"
3. }

```



The `_id` field is automatically generated by Cloudant because it is considered the unique key identifier for the document. You can override this key by providing a `_id` with a unique value across the other documents.

- \_\_\_ c. Replace the JSON object in the document editor with the following JSON object:

```
{
  "_id": "novel_001",
  "name": "Oliver Twist",
  "author": "Charles Dickens",
  "year": 1839
}
```

```

1. 
2. "_id": "novel_001",
3. "name": "Oliver Twist",
4. "author": "Charles Dickens",
5. "year": 1839
6. }
```

- \_\_\_ d. Click **Create Document**.

- \_\_\_ e. After you create the document, you are redirected back to the database overview page, which contains the newly created document, as shown in the following figure.

id	key	value
novel_001	novel_001	{ "rev": "1-a9b60df..."}

- \_\_\_ f. To show the full data of the newly created document in JSON format, select **{ } JSON** from the top tabs, as shown in the following figure.

The screenshot shows the IBM Cloudant interface for a database named 'novels'. On the left, there's a sidebar with options like 'All Documents', 'Query', 'Permissions', 'Changes', and 'Design Documents'. The main area displays a document with the ID 'novel\_001'. The document's JSON structure is shown below:

```

id "novel_001"
{
  "id": "novel_001",
  "key": "novel_001",
  "value": {
    "rev": "1-a9b60df0448b15a0cd6ded38cbb91154"
  },
  "doc": {
    "_id": "novel_001",
    "_rev": "1-a9b60df0448b15a0cd6ded38cbb91154",
    "name": "Oliver Twist",
    "author": "Charles Dickens",
    "year": 1839
  }
}

```



## Information

After you create the document, you find a field that is called `_rev` that is automatically generated by Cloudant. The `_rev` field is used internally by the Cloudant database as a revision number. A revision number is added to your documents by the server when you insert or modify documents. You must specify the latest `_rev` when you update a document or your request fails and returns a 409 error. This field helps you avoid conflicting data states. The revision number is also used to confirm that a client is trying to modify the current version of a document.

**Note**

You can also create a document by clicking the three dots vertical menu icon and selecting **+ New Doc**, as shown in the following figure.

The screenshot shows the Apache CouchDB Futon interface for managing a database named 'novels'. On the left, there's a sidebar with options: All Documents (selected), Query, Permissions, Changes, and Design Documents. On the right, there's a main area showing a list of documents. A context menu is open over one of the documents, with the 'Add New' option expanded. The 'New Doc' option is highlighted with a mouse cursor. The document details visible in the background include '\_id': 'novel\_001', '\_rev': '1-a9b6e', '\_source': 'novel\_001', '\_type': 'novel', '\_id': 'novel\_001', '\_rev': '1-a9b6e', '\_source': 'novel\_001', '\_type': 'novel', and 'author': 'Olive'.

Another option to create a document is to click the **+** icon next to **All Documents** in the left pane then select **New Doc +** as shown in the following figure.

This screenshot shows the same Apache CouchDB Futon interface as the previous one, but the '+' icon to the right of the 'All Documents' tab is now highlighted with a mouse cursor, indicating it's the active selection point for creating a new document.

- 6. Repeat the steps to create a new document three times by using the following JSON objects to add documents for “King Solomon's Mines”, “Treasure Island”, and “The Merry Adventures of Robin Hood” novels:

```
{
  "_id": "novel_002",
  "name": "King Solomon's Mines",
  "author": "H. Rider Haggard",
  "year": 1885
}
{
  "_id": "novel_003",
  "name": "Treasure Island",
  "author": "Robert Louis Stevenson",
  "year": 1883
}
{
  "_id": "novel_004",
  "name": "The Merry Adventures of Robin Hood",
  "author": "Howard Pyle",
  "year": 1883
}
```

- 7. Update the document of “Oliver Twist” novel by completing the following steps:

- a. Return to the metadata view by selecting the **Metadata** tab from the top tabs, as shown in the following figure.

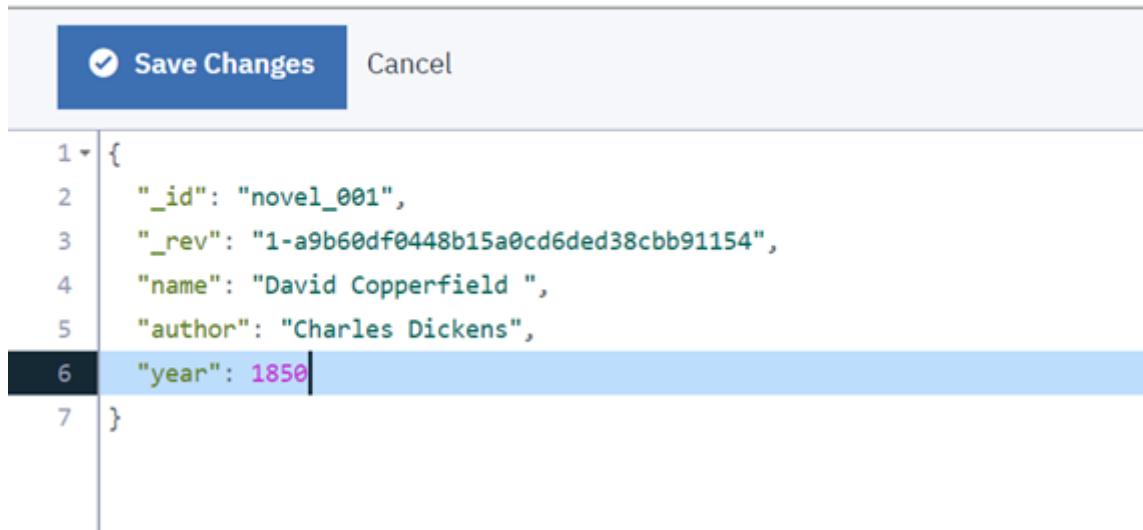
	id	key	value
<input type="checkbox"/>	novel_001	novel_001	{ "rev": "1-a9b60df0448b15a0cd..." }
<input type="checkbox"/>	novel_002	novel_002	{ "rev": "1-0928562182e3d6736..." }

- b. Select the document where `_id` corresponds to `novel_001` from the list of documents, as shown in the following figure.

	id	key	value
<input checked="" type="checkbox"/>	novel_001	novel_001	{ "rev": "1-a9b60df0448b15a0cd..." }
<input type="checkbox"/>	novel_002	novel_002	{ "rev": "1-0928562182e3d6736..." }
<input type="checkbox"/>	novel_003	novel_003	{ "rev": "1-56ff7fa2010f51..." }
<input type="checkbox"/>	novel_004	novel_004	{ "rev": "1-fd3f8e0831970..." }

- \_\_\_ c. The document editor opens. Change the name field to David Copperfield and the year to 1850, as shown in the following figure.

novels ➤ novel\_001



```

1  {
2    "_id": "novel_001",
3    "_rev": "1-a9b60df0448b15a0cd6ded38cbb91154",
4    "name": "David Copperfield",
5    "author": "Charles Dickens",
6    "year": 1850
7  }

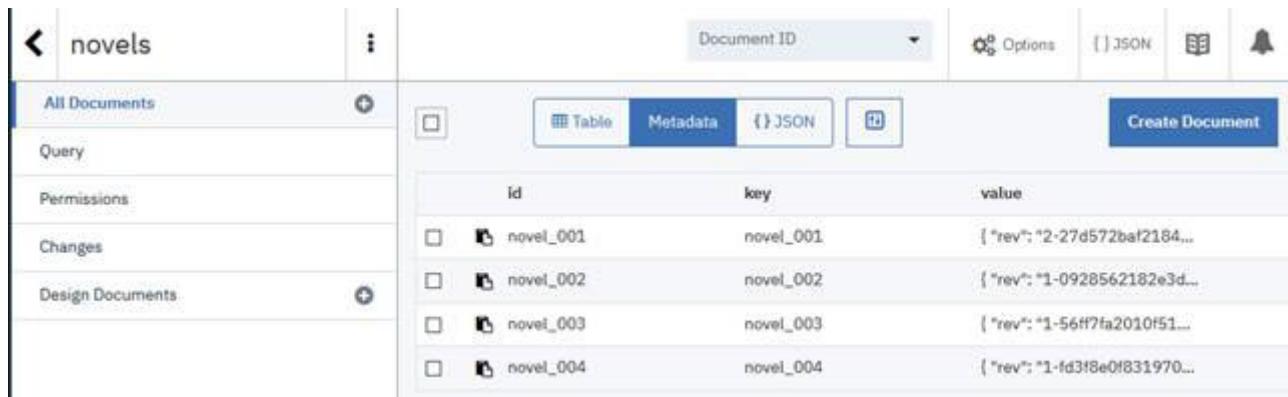
```

- \_\_\_ d. Click **Save Changes**.



### Important

When you create a document, the `_rev` field that is generated by Cloudant always starts with 1-, which indicates that this is the first version of this document. This number increments by 1 with each document update to highlight the document version number. The first digit in the revision number increments after the document update, as shown in the following figure.



	Id	key	value
<input type="checkbox"/>	novel_001	novel_001	{ "rev": "1-27d572baef2184..." }
<input type="checkbox"/>	novel_002	novel_002	{ "rev": "1-0928562182e3d..." }
<input type="checkbox"/>	novel_003	novel_003	{ "rev": "1-56ff7fa2010f51..." }
<input type="checkbox"/>	novel_004	novel_004	{ "rev": "1-fd3f8e0f831970..." }

- \_\_\_ 8. Delete the document for the “The Merry Adventures of Robin Hood” novel by completing the following steps:
- \_\_\_ a. From the **novels** database overview page, select the document where `_id` corresponds to `novel_004` from the list of documents, as shown in the following figure.

The screenshot shows the Apache CouchDB Futon interface. On the left, there's a sidebar with options like 'All Documents', 'Query', 'Permissions', 'Changes', and 'Design Documents'. The main area displays a table with columns 'id', 'key', and 'value'. There are five rows of data, with the fifth row ('novel\_004') highlighted by a blue selection bar at the bottom.

	id	key	value
<input type="checkbox"/>	novel_001	novel_001	{ "rev": "2-27d572baf2184...
<input type="checkbox"/>	novel_002	novel_002	{ "rev": "1-0928562182e3d...
<input type="checkbox"/>	novel_003	novel_003	{ "rev": "1-56ff7fa2010f51...
<input checked="" type="checkbox"/>	novel_004	novel_004	{ "rev": "1-fd3f8e0f831970ce76d6a0cc9910e222

\_\_\_ b. Click **Delete** from the top bar, as shown in the following figure.

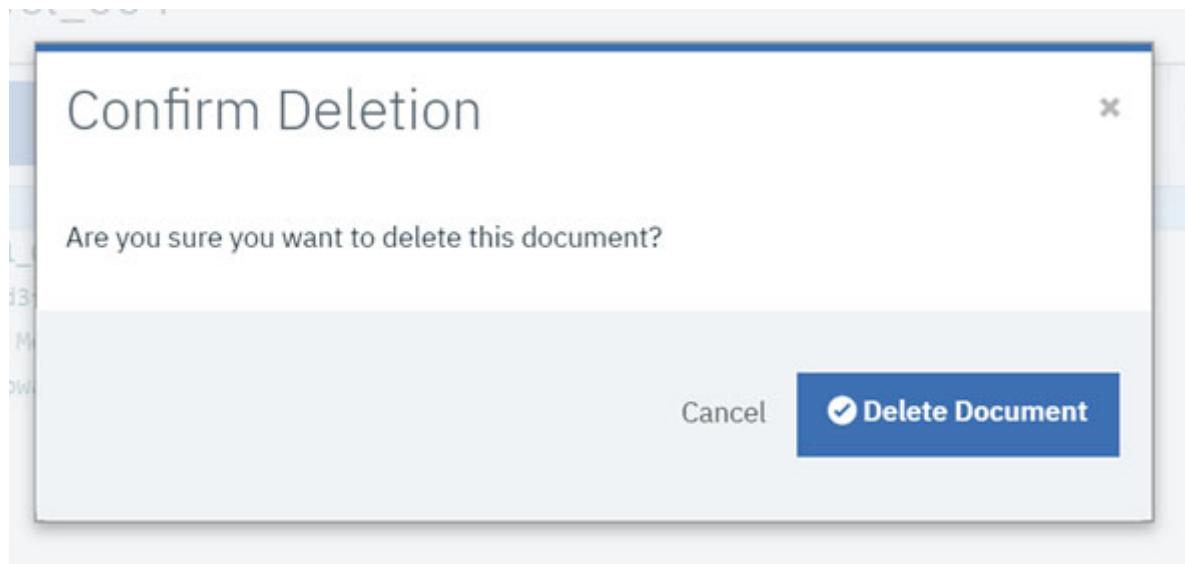
This screenshot shows the document details for 'novel\_004'. At the top, there are buttons for 'Save Changes', 'Cancel', 'Upload Attachment', 'Clone Document', and 'Delete'. The 'Delete' button is highlighted with a cursor. Below it, the document's JSON structure is displayed:

```

1. {
2.   "_id": "novel_004",
3.   "_rev": "1-fd3f8e0f831970ce76d6a0cc9910e222",
4.   "name": "The Merry Adventures of Robin Hood",
5.   "author": "Howard Pyle",
6.   "year": 1883
7. }

```

\_\_\_ c. Click **Delete Document**, as shown in the following figure.



- 9. Create a query to get the documents of the novels that are published after 1870 by completing the following steps:

- a. Select **Query** from left pane, as shown in the following figure.

id	key	value
novel_001	novel_001	{ "rev":
novel_002	novel_002	{ "rev":
novel_003	novel_003	{ "rev":

- b. You are redirected to the query editor page, as shown in the following figure.

A large cloud icon is in the center. Below it, the text 'No Documents Found' is displayed. At the bottom, there are buttons for 'Run Query' and 'Explain', and a link to 'manage indexes'. The status bar at the bottom says 'Showing 0 documents. Documents per page: 20'."/&gt;

- \_\_ c. Replace the JSON object in the editor with the following code:

```
{
  "selector": {
    "year": {
      "$gt": 1870
    }
  },
  "fields": [
    "_id",
    "_rev",
    "name",
    "author",
    "year"
  ]
}
```

- \_\_ d. Click **Run Query**, as shown in the following figure.

The screenshot shows the Cloudant Query interface. At the top, there's a dropdown menu labeled 'Query history'. Below it is a title bar 'Cloudant Query' with a help icon and a refresh/crop icon. The main area is a code editor with syntax highlighting for JSON. The code is as follows:

```

1  {
2   "selector": {
3     "year": {
4       "$gt": 1870
5     }
6   },
7   "fields": [
8     "_id",
9     "_rev",
10    "name",
11    "author",
12    "year"
13  ]
14 }
```

At the bottom of the code editor, there are two buttons: 'Run Query' (in blue) and 'Explain' (in white). To the right of these buttons is a link 'manage indexes'. Below the buttons, a message says 'Executed in 1 ms' with a small info icon.

- \_\_ e. The results are populated in the right pane, as shown in the following figure.

The screenshot shows the IBM Cloudant interface. At the top, there are three buttons: 'Table' (selected), '{} JSON', and a square icon. To the right is a 'Create Document' button. Below this is a table with four columns: '\_id', 'author', 'name', and 'year'. There are two rows of data:

_id	author	name	year
novel_002	H. Rider Haggard	King Solomon's Mines	1885
novel_003	Robert Louis Steven...	Treasure Island	1883

## **Part 6: Using an HTTP API client to access Cloudant**

Cloudant uses HTTP for all external communication to provide simple, web-based access to data in the Cloudant data store. The HTTP API is a programmatic way of accessing the data from any HTTP client consumer. It provides a set of HTTP access methods for create, read, update, and delete operations on Cloudant documents, and for using Cloudant query and many other features that are related to Cloudant.

In this part, you use an HTTP API client (Postman) to interact with Cloudant. You can use another HTTP API client, such as cURL or SOAP-UI, if you prefer.

Complete the following steps:

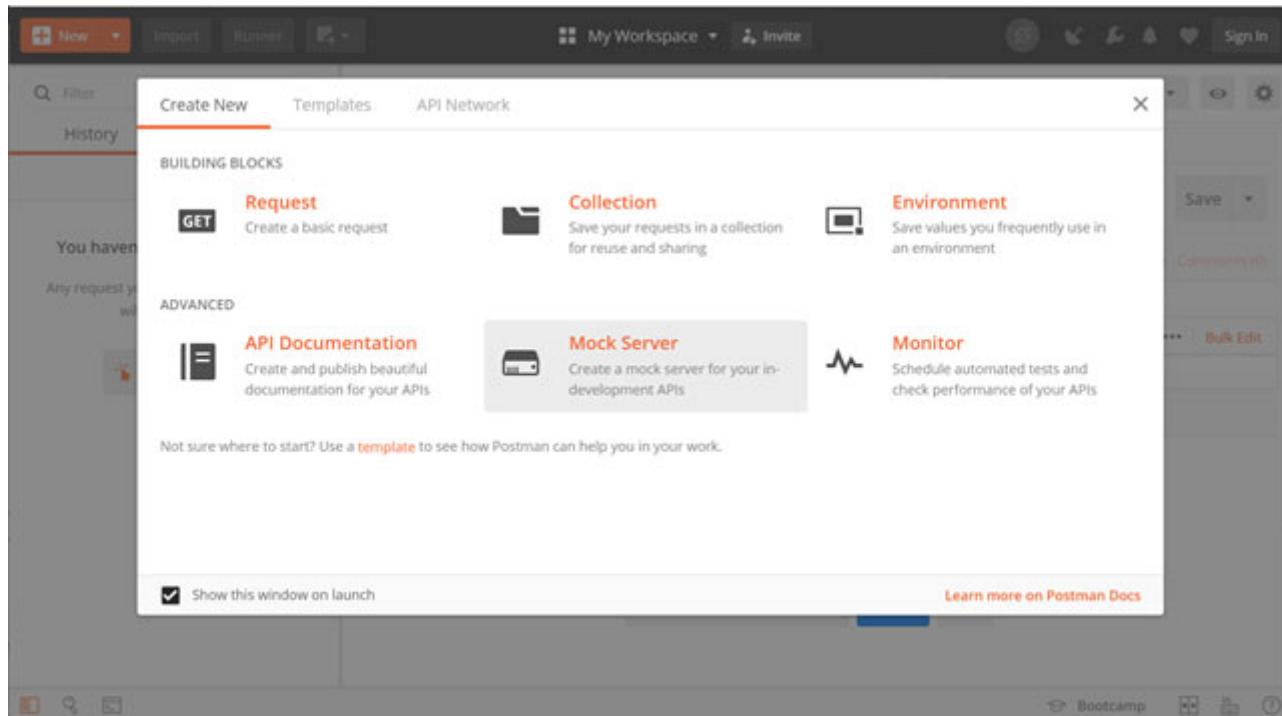
- 1. From the Start menu, find and open Postman. The window that is shown in the following figure opens.

The screenshot shows the Postman application interface. The top navigation bar includes 'New', 'Import', 'Runner', 'My Workspace', 'Invite', and 'Sign In'. The main area is titled 'Untitled Request' and shows a 'GET' request. The 'Params' tab is selected, showing a table with one row: 'Key' and 'Value'. The 'Headers' tab is also visible. Below the request area is a 'Response' section with the placeholder text 'Hit the Send button to get a response.' A 'Send' button is located at the bottom of the request area.

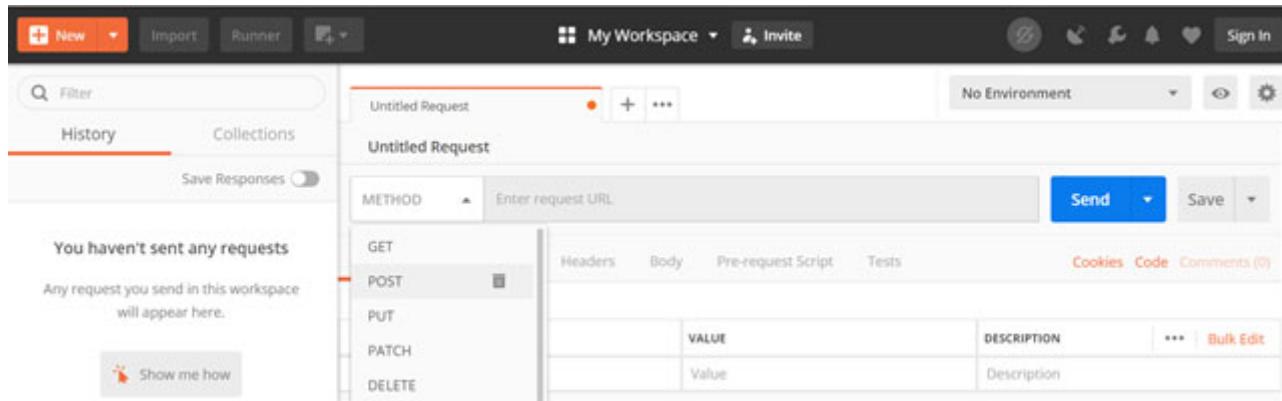


### Note

If a pop-up window opens when you open Postman, as shown in the following figure, close it and proceed with the steps normally.



- 2. You use the Cloudant API key to get a time-limited access token that is used in all the upcoming requests to authenticate and authorize your access to Cloudant. After this token expires, you cannot use it anymore. To retrieve this access token, complete the following steps.
  - a. Select the **POST** method from the **METHOD** menu, as shown in the following figure.



- b. Enter the following request URL into the POST field, as shown in the following figure:
- <https://iam.cloud.ibm.com/identity/token>

The screenshot shows the Postman interface. At the top, there's a header with 'POST https://iam.cloud.ibm.com/identity/token'. Below it, the main area has 'POST https://iam.cloud.ibm.com/identity/token' again. To the right are 'Send', 'Save', and other icons. A dropdown menu says 'No Environment'. Below the URL input, tabs are labeled 'Params', 'Authorization', 'Headers', 'Body', 'Pre-request Script', and 'Tests'. The 'Body' tab is highlighted with a red underline. To the right of these tabs are 'Cookies', 'Code', and 'Comments (0)'. Under the 'Body' tab, there's a section titled 'Query Params' with a table. The table has columns: KEY, VALUE, and DESCRIPTION. There are two rows: one with 'Key' and 'Value' and another with 'Description'. At the bottom of the table are 'Bulk Edit' and 'Description' buttons.

- \_\_\_ c. Select the **Body** tab from the bar below the Post menu URL field, as shown in the following figure.

This screenshot shows the same Postman interface as the previous one, but with different settings. The 'Body' tab is selected in the navigation bar. Below it, a dropdown menu shows options: 'none' (selected), 'form-data', 'x-www-form-urlencoded', 'raw', and 'binary'. The text 'This request does not have a body' is displayed. The 'Response' section is visible at the bottom.

- \_\_\_ d. Select `x-www-form-urlencoded` and then add the following keys and values, as shown in the following figure:
- Key: `grant_type` and value: `urn:ibm:params:oauth:grant-type:apikey`
  - Key: `response_type` and value: `cloud_iam`
  - Key: `apikey`, and the value for this field must include your Cloudant instance API key, which you can retrieve from the `cloudant_credentials.json` file. This file includes the Cloudant credentials, and you already saved it in your workstation in a previous step.

The screenshot shows the Postman interface with a POST request to <https://iam.cloud.ibm.com/identity/token>. The 'Body' tab is selected, showing the following parameters:

KEY	VALUE	DESCRIPTION
grant_type	urn:ibm:params:oauth:grant-type:apikey	
response_type	cloud_iam	
apikey	H0ejpwgzwCvuo89TCdFCaRPZ-fYAhviVn...	
Key	Value	Description

- \_\_ e. Click **Send**, and you receive a JSON object response that includes the access token, which is valid for 60 minutes, as shown in the following figure.

```

1 - {
2   "access_token": "eyJraWQioiIyMDE3MTYEyOSiImFsZyI6IlJTMju2In0
    .eyJpYW1faWQioiIjpyW0tU2VydmljZU1kLWNkNDJhnzNlTzWmItNGFhNS05NDU5LTnkZmFkNTY5OGMzNyIsImlkIjoiaWFtLVNlc
    ZpY2VJZC1jZDQyYTcziYi02ZjMyLTrhYTUt0TQ10Se0ZGZhZDU20ThjMzcilCjyZWfsBwlkIjoiaWFtIiwiawRlbnRpZmllciI6IlNlc
    nZpY2VJZC1jZDQyYTcziYi02ZjMyLTrhYTUt0TQ10Se0ZGZhZDU20ThjMzcilCjzdwiioiJTZXj2awNlSwQtY2Q0MmE3M2ItNmYzMi00
    YWE1Tk0Ntktm2RmpYQ1nkj4yzM3iwiic3ViX3R5cGuioiJtZXj2awNlSwQilCjy2Nvdw50ijp7InzhbGlkjlp0cnVlCjic3MioiI
    yNTM2NzNmZw2Njc0NmMyYTRlNddlM2U2YmjzWRKzJ9LCj0pYXQiojE1NTU10DY3NTAsImV4ccI6MTU1NTU5MDM1MCwiaXNzIjoiAH
    R0chM6Ly9pYw0uY2xdwQuaWjtLmIvb5pZGVudGloeiSisImdyYW50X3R5cGuioiJ1cm46aWj0nbHcmFtczpvYXv0aDpncmfudC10e
    XB10mFwaWtlesisInNjb3BLijoiaWtjIg9wZW5pZCisImNsakVudF9pZC16ImrlZmF1bHQiLCjhy3IiOjEsImftciI6WyJwd2qixX0
    .nE1AonH00JFJ4A9WAdwJwkvmbZ2beAw0ODwJHASEBvhjCH0sErMWD0EQZG2DyHW-brZz3vT5RrIYvTFdu99RXVIFGLVRPK
    -a3t0Sjb5PNmj2rFCU0aCzvUz2qFGAp6IXh7rT8jX97vjFvCT_18AHw0AgXFlncaPTHk7gnYJMsmeQ_Ts73UR0fn-Qe3AEbhwljn
    -_k2lsika_sUe0SneARrrRH1DfC1tkFr0VXmcSDfivB7WKmV8im4sfz1Er3Pn4wjwHeFxxt5ZRHALQo69xCfho_Z1AoMq0DCux4k
    bIxCu5I0oPqKsD8T5Qp0QcbTOZtgQdU7nvrxRmnQ",
3   "refresh_token": "ReUiBux-nqncuNyRm3eyfEfFdq5cyf0WAJpdLeC1GRoVFIE4KiukuKnJufUPX
    -MtJlc3Nrwf_Y40hbcXG5NNIgwtZz9hzZuMTGB4_xun7iUyT0cPtevW9JVUVfpqc9GxHy4D1_ptPU2EwnQkCeizclWa3v7LzUqDNFcV
    rpxX-1urConFa1Ga9an5HgpsPchAt6fkcta2GsNsK2hWoLWTktqss3b3kb-NSMaDftp1jwMYiW8IsYlrINOH20de8C71gDwiouK
    -ugfa4K5bnnq02C0pu1lw0KwiovxoLts2R6V6pcijuvUdv2vUpQKofu2Mkbc2m5ilmu2Y172R1FGqzo_ufJInoAUHkdbgQFKCS8jr64M
    -shO2EdhD2Gqx0Dyff7XTcz0-V2QA-xm06gzn0fhiR8tAZXndl1v_jrYsonZux
    -eguIKGyslmRNPyXunSmdqZ3SsZcr78Rpm0mWGEmOn1fDKemtwVtb2MndkvZByeFOP27CPB5f_u_Nis
    -NVwg2Xe5oCzioVB7rCZV_GYssej504Velqi9YQgvAaIYU3In6oGZnZ5RVAXgnac2cnJv3w9UMsTKTj0xEwPGFAv4Y_pCD9Kwku9
    -bzvR7e-5jsMr7Qk03C8zKaR8Z1I8hs0jh70
    -Ag_RfisCnqJH7cg339hqH9da9jfk43mahYGwdhv2K0ddGtqVY_0QprdXi7lyQb87nCk04eQcAjspCpds5q_RbBW8SFTzssxdCozvNE
    EoNth01Gyd0ltWyn_ocAvGpfmfZYqycMEM9MyQkdVTxijt8_z0R880KWRhpKkyagphhm1URdgsM3CQFlAFrea4NSfZZ1-Tat_iqHC
    -wAG-3Pf6ooobZFFFaKafwCLNvpSTqw3UYd7sygK-La1Fsg14ESe3p8_Q4zuQ12YqT3b9NMvGUmm8aT"

```

- \_\_ f. Copy the value of the `access_token` because it is used in all the upcoming requests.



## Important

If the `access_token` expires while you make any of the upcoming requests, repeat step 2 to retrieve a new access token and use it instead of the old one.



## Troubleshooting

If you receive the error in the response that is shown in the following figure, verify that you provided the correct `apikey`.

```

1  {
2   "context": {
3     "requestId": "e0b6ffb554204b279c16d2789c717a8e",
4     "requestType": "incoming.Identity_Token",
5     "userAgent": "PostmanRuntime/7.6.1",
6     "clientIp": "156.204.159.136",
7     "url": "https://iam.cloud.ibm.com",
8     "instanceId": "iamid-3.8-5353-9ad23d9-64d874b89d-7fqqq",
9     "threadId": "3b87fa",
10    "host": "iamid-3.8-5353-9ad23d9-64d874b89d-7fqqq",
11    "startTime": "18.04.2019 13:50:53:423 UTC",
12    "endTime": "18.04.2019 13:50:53:481 UTC",
13    "elapsedTime": "58",
14    "locale": "en_US",
15    "clusterName": "iam-id-prelon02-xiji"
16  },
17  "errorCode": "BXNIM0415E",
18  "errorMessage": "Provided API key could not be found"
19 }
```

You can check the credentials of the service again from the service overview page on IBM Cloud, as shown in the following figure.

The screenshot shows the 'Service credentials' section for a Cloudant instance named 'Cloudant-fs'. The JSON object displayed is:

```
{
  "apikey": "H0eJpwgzwCvuo89TCdFCaRPZ-fYqhiVn6a4tJUAdaVd",
  "host": "7471c743-b853-429f-8367-bcfbe87fef0e-bluemix.cloudantnosqldb.appdomain.cloud",
  "iam_apikey_description": "Auto generated apikey during resource-key operation for Instance - crn:v1:bluemix:public:cloudantnosqldb:eu-gb:a/253673fed66746c2a4e47e3e6bbbeddf:4a4a725c-5e9d-4e02-ad5f-bca4845393e8::",
  "iam_apikey_name": "auto-generated-apikey-db884cf2-5b0c-49cc-9942-ca6eb94b02a0",
  "iam_role_crn": "crn:v1:bluemix:public:iam::::serviceRole:Manager",
  "iam_serviceid_crn": "crn:v1:bluemix:public:iam:identity::a/253673fed66746c2a4e47e3e6bbbeddf::serviceid:ServiceId-cd42a73b-6f32-4aa5-9459-3dfad5698c37",
  "url": "https://7471c743-b853-429f-8367-bcfbe87fef0e-bluemix.cloudantnosqldb.appdomain.cloud",
  "username": "7471c743-b853-429f-8367-bcfbe87fef0e-bluemix"
}
```



## Information

There is another way to retrieve an access token by using the IBM Cloud CLI as follows:

1. Open the Command Prompt and log in to the IBM Cloud with the CLI where **username** is the IBM ID (email) that you use to log in to IBM Cloud), **password** is your password and **region** is the region that corresponds to your location.

```
ibmcloud login -u username -p password -r region
```

The following list shows the list of regions and locations

- o Sydney: au-syd
- o Frankfurt: eu-de
- o London: eu-gb
- o Dallas: us-south
- o Washington DC: us-east

- To verify your Cloudant service region, you can check the location from the overview page on the IBM Cloud page, as shown in the following figure.

The screenshot shows the 'Overview' tab for the 'Cloudant-fs' service. The location is listed as 'London'.

- \_\_ 2. Run the following command to get the access token that you can use in the Cloudant HTTP requests:

```
ibmcloud iam oauth-tokens
```

```
C:\>ibmcloud iam oauth-tokens
IAM token: Bearer eyJraWQiOiIyMDE5MDcyNCIsImFsZyI6IlJTMyU2In0.eyJpYwifalwQioiJJQk1pZC01NTAwMDJUMUFICIiwiaWQiOiJJQk1pZC01NTAwMDJUMUFICIiwicmVhbG1pZCI6Ik1lCTWlkIiwiawRlbwRpZmlciI6IjU1MDAwMlQxQUIiLCJnaXZlb19uYW1lIjoiU3R1ZGVudDEwMyIsImZhbwIseV9uYw1lIjoi2xvdwQxhDMiLCJuW1lIjoiU3R1ZGVudDEwMyBDbg91ZDEwMyIsImvYwlsIjoiC3R1ZGVudDEwM0B5Ywhvby5jb20iLCJzdwiIoIjzdhMvkZw50MTA9QHlaG9vLmNvbSIsmFjY291bnQiOnsidmFsawQiOnRhmuIsImJzcyI6IjcyNmEwNWJlNzKxzjQzTRiZjBhMDg4NTRkyTdmZDg0In0sImlhC16MTU3NTgZNID2MCwiZXhwIjoxNTc1ODM3ODYwLCJpc3MiOiJodHRwczovL2lhbs5jbG91ZC5pYm0uY29tL21kZW50aXR5IwiZ3JhbnRfdh1wZSI6InVbjppYm06cGFyYw1zOm9hdXRoOmdyYw50LXR5cGU6cGfcz2NvZGUiLCJzY29wZSI6ImlibSBvcGVuaWQilCJjbG1lbnRfaWQioiJieCisImFjciI6MSwiYw1yIjpBIn83ZCJdfQ.lciW8UqtrG6J0eY-JIv6atweYfba2SmXfbXAFGssZ9MnZfwf1WMD9t047HQUNe_I-2XzjDFXkB8V6ad5JgIHjB8Dcc-KYJG8-FGPymFt5MsCsN_-Ce0vmm55j2T_N8qEse57zbYbOG155j30_FjyatcJ368B8GpIizaZLRCVZ_UYPEHnRvWcVJSFpC1DgmCnqsyFuRyAR3MAPUsxD1ywL9t6fe7HdEn6XCq9C1q2Pme5iiUrPHv-jSgZAhNT179CaqeK7B11gYjLK0Hjb-6aS4UTz4i2GyvEc4a6EbTrXGECbp8461nP10P5d5F64_15H29uAaYQrxu8GqOfErIlw
```

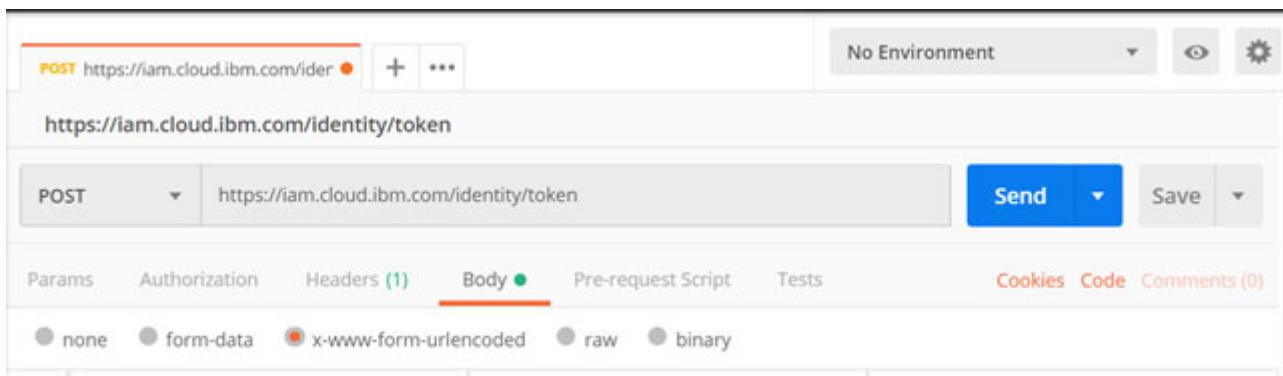
- \_\_ 3. Copy the access token; you use it in the HTTP requests.

- \_\_ 4. To view all the documents in a database, issue a **GET** request to the following URL:

`$URL/$DATABASE/_all_docs?include_docs=true`

To show all the documents in the novels database, complete the following steps:

- \_\_ a. Open a new tab in Postman by clicking **+** in the top bar, as shown in the following figure.



- \_\_ b. Set the request method to the **GET** method.

- \_\_ c. Set the request URL to `$URL/$DATABASE/_all_docs?include_docs=true`, as shown in the following figure:

Replace `$URL` with the value of the URL key in the `cloudant_credentials.json` file.

Replace `$DATABASE` with the database name **novels**.

POST https://iam.cloud.ibm.com/identity/api/v1/token  
GET https://7471c743-b853-429f-8367-bcfbe87fef0e-bluemix.cloudantnosqldb.appdomain.cloud/novels/\_all\_docs?include\_docs=true

**Params** ● Authorization Headers Body Pre-request Script Tests Cookies Code Comments (0)

KEY	VALUE	DESCRIPTION	***	Bulk Edit
<input checked="" type="checkbox"/> include_docs	true			
Key	Value	Description		

Response

- \_\_\_ d. Select the **Headers** tab and add the following key and value, as shown in the following figure:

Key: Authorization.

Value: Bearer \$ACCESS\_TOKEN. Replace \$ACCESS\_TOKEN with the access token that you obtained in the previous step Step 2).

GET https://7471c743-b853-429f-8367-bcfbe87fef0e-bluemix.cloudantnosqldb.appdomain.cloud/novels/\_all\_docs?include\_docs=true

**Headers (1)** ● Authorization Body Pre-request Script Tests Cookies Code Comments (0)

KEY	VALUE	DESCRIPTION	***	Bulk Edit	Presets
<input checked="" type="checkbox"/> Authorization	Bearer \$ACCESS_TOKEN				
Key		Description			

Response

- \_\_\_ e. Click **Send**. All three documents are displayed, as shown in the following figure.

```

1 [
2   "total_rows": 3,
3   "offset": 0,
4   "rows": [
5     {
6       "id": "novel_001",
7       "key": "novel_001",
8       "value": {
9         "rev": "2-71b3630e17d937689cc099acab41e8b2"
10      },
11      "doc": {
12        "_id": "novel_001",
13        "_rev": "2-71b3630e17d937689cc099acab41e8b2",
14        "name": "David Copperfield",
15        "author": "Charles Dickens",
16        "year": 1850
17      }
18    },
19    {
20      "id": "novel_002",
21      "key": "novel_002",
22      "value": {
23        "rev": "1-0928562182e3d6736e0169e4eef1717c"
24      },
25    }
26  ]

```

5. To create a document, send a **POST** request to **\$URL/\$DATABASE** with the document's JSON content in the request body. To create a new document for “The Merry Adventures of Robin Hood” novel, complete the following steps, as shown in the following figure:
- Update the request URL to **\$URL/\$DATABASE**.
  - Update the HTTP method to **POST**.

KEY	VALUE	DESCRIPTION ***	Bulk Edit	Presets
Authorization	Bearer eyJraWQiOiIyMDE3MTEyOSIsImFs...			
Key	Value	Description		

- To specify the content of the document that you want to create, click the **Body** tab from the tabs bar below the request URL, select the **raw** radio button, and then select the type as **JSON (application/json)**, as shown in the following figure.

- \_\_\_ d. Enter the following code as the content of the document:

```
{
  "_id": "novel_004",
  "name": "The Merry Adventures of Robin Hood",
  "author": "Howard Pyle",
  "year": 1883
}
```



### Information

The `_id` should be a unique identifier for the document. If the `_id` is not provided, Cloudant generates an ID.

- \_\_\_ e. Click **Send**.

- \_\_\_ f. Check the response that shows the `_id` and `_rev` fields to verify the creation of the document, as shown in the following figure.

```

1 {
2   "ok": true,
3   "id": "novel_004",
4   "rev": "3-096bc2ebfdac5bc066f34fd7da816f06"
5 }
```

- \_\_ 6. Create another document for the “Oliver Twist” novel by completing the following steps:
- \_\_ a. Update the content of the body as follows. The results are shown in the following figure.

```
{
  "_id": "novel_005",
  "name": "Oliver Twist",
  "author": "Charles Dickens",
  "year": 2005
}
```

The screenshot shows the Postman interface. The top bar has 'POST' selected and a URL pointing to a Cloudant database. Below the URL are tabs for 'Params', 'Authorization', 'Headers (2)', 'Body', 'Pre-request Script', 'Tests', 'Cookies', 'Code', and 'Comments (0)'. The 'Body' tab is active, set to 'raw' and 'JSON (application/json)'. The JSON payload is:

```

1 - {
2   "_id": "novel_005",
3   "name": "Oliver Twist",
4   "author": "Charles Dickens",
5   "year": 2005
6 }
7

```

- \_\_ b. Click **Send**.
- \_\_ c. Check the response that shows the `_id` and `_rev` fields to verify the creation of the new document, as shown in the following figure.

The screenshot shows the Postman response view. It includes tabs for 'Pretty', 'Raw', 'Preview', 'JSON', and a copy icon. The JSON response is:

```

1 - {
2   "ok": true,
3   "id": "novel_005",
4   "rev": "1-6ef601ccf0bdb9bf2a6e9a88e928b88a"
5 }

```

- \_\_ 7. Create another document for the “Ivanhoe” novel by completing the following steps:
- \_\_ a. Update the content of the body as follows. The results are shown in the following figure.

```
{
  "_id": "novel_006",
  "name": "Ivanhoe",
  "author": "Walter Scott",
  "year": 1820
}
```

The screenshot shows the Postman interface with a POST request to the URL `https://7471c743-b853-429f-8367-bcfbe87fef0e-bluemix.cloudantnosqldb.appdomain.cloud/novels`. The Body tab is selected, displaying the following JSON payload:

```

1+ {
2   "_id": "novel_006",
3   "name": "Ivanhoe",
4   "author": "Walter Scott",
5   "year": 1820
6 }
7

```

- \_\_\_ b. Click **Send**.
- \_\_\_ c. Check the response that shows the `_id` and `_rev` fields to verify the creation of the new document, as shown in the following figure.

The screenshot shows the Postman response preview in JSON format. The response indicates success (`ok: true`) and provides the newly created document's `_id` and `_rev` values.

```

1+ {
2   "ok": true,
3   "id": "novel_006",
4   "rev": "1-d89012bd607f4bdf341ab6bd0cdaf6d8"
5 }

```

- \_\_\_ 8. To read the data of a specific document, send a **GET** request to `$URL/$DATABASE/$DOCUMENT_ID`. To get the data of the document of the “Oliver Twist” novel, complete the following steps. The results are show in the following figure.

  - \_\_\_ a. Update the request URL to `$URL/$DATABASE/$DOCUMENT_ID`.
  - \_\_\_ b. Update `$DOCUMENT_ID` with the ID of the document of the “Oliver Twist” novel, which is `novel_005`.
  - \_\_\_ c. Update the HTTP method to **GET**.

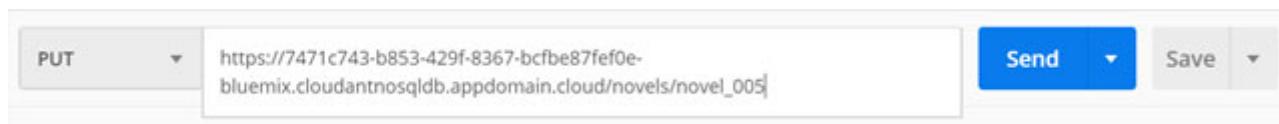
The screenshot shows the Postman interface with a GET request to the URL `https://7471c743-b853-429f-8367-bcfbe87fef0e-bluemix.cloudantnosqldb.appdomain.cloud/novels/novel_005`.

- \_\_\_ d. Click **Send**.
- \_\_\_ e. Check the response that is shown in the following figure and copy the `_rev` value because it is required for updating the document in the next step.

```

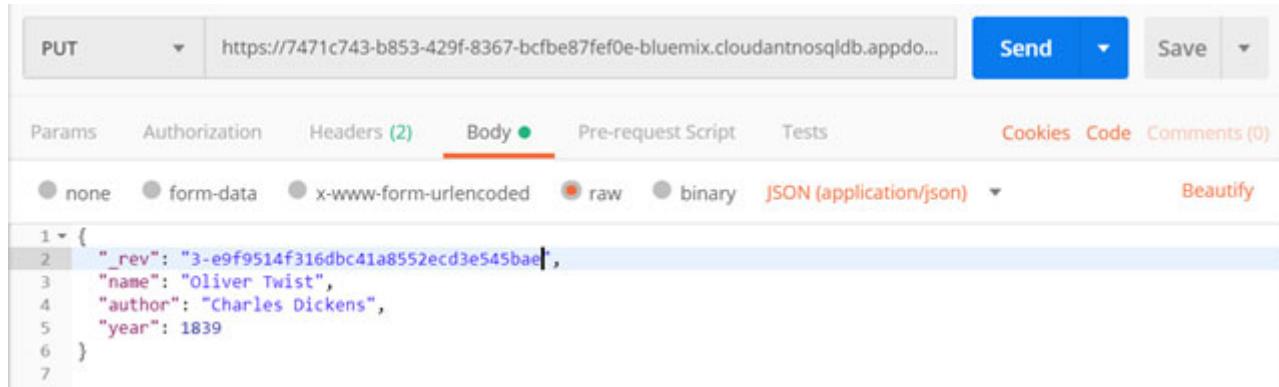
1 {
2   "_id": "novel_005",
3   "_rev": "3-e9f9514f316dbc41a8552ecd3e545bae",
4   "name": "Oliver Twist",
5   "author": "Charles Dickens",
6   "year": 2005
7 }
  
```

- \_\_\_ f. Copy the value of the `_rev` field.
- \_\_\_ 9. To update a document, send a `PUT` request to `$URL/$DATABASE/$DOCUMENT_ID` with the updated document JSON content and the latest `_rev` field. To update the year in the document of the “Oliver Twist” novel, complete the following steps, as shown in the following figure:
- \_\_\_ a. Keep the request URL as `$URL/$DATABASE/novel_005`.
  - \_\_\_ b. Update the HTTP method to `PUT`.



- \_\_\_ c. Update the request body with the following code, as shown in the following figure. Replace `$REV` with the `_rev` value that you copied in the previous step.

```
{
  "_rev": "$REV",
  "name": "Oliver Twist",
  "author": "Charles Dickens",
  "year": 1839
}
```



- \_\_\_ d. Click **Send**. In the response, notice that the value of `_rev` is updated (Send response), as shown in the following figure.

A screenshot of a REST client interface. The top bar shows 'PUT' selected, a URL 'https://7471c743-b853-429f-8367-bcfbe87fef0e-bluemix.cloudantnosqldb.appdomain.cloud/novels/novel\_006', and buttons for 'Send' and 'Save'. Below the URL input is a JSON editor with tabs 'Pretty', 'Raw', 'Preview', 'JSON', and a diff icon. The JSON content is:

```

1 {
2   "ok": true,
3   "id": "novel_006",
4   "rev": "4-0aea19048ed14d532ce3db06751d3c56"
5 }

```

- \_\_\_ 10. To read the data of a specific document, send a **GET** request to `$URL/$DATABASE/$DOCUMENT_ID`. So, to get the data of the document of the “Ivanhoe” novel, complete the following steps, as shown in the following figure:
  - \_\_\_ a. Keep the request URL as `$URL/$DATABASE/$DOCUMENT_ID`.
  - \_\_\_ b. Update `$DOCUMENT_ID` with the ID of the document of the “Ivanhoe” novel, which is `novel_006`.
  - \_\_\_ c. Update the HTTP method to **GET**.
  - \_\_\_ d. Click **Send**.
  - \_\_\_ e. Check the response as shown in the following figure and copy the `_rev` value because it is required for deleting the document in the next step.

A screenshot of a REST client interface. The top bar shows 'GET' selected, a URL 'https://7471c743-b853-429f-8367-bcfbe87fef0e-bluemix.cloudantnosqldb.appdomain.cloud/novels/novel\_006', and buttons for 'Send' and 'Save'. Below the URL input is a JSON editor with tabs 'Pretty', 'Raw', 'Preview', 'JSON', and a diff icon. The JSON content is:

```

1 {
2   "_id": "novel_006",
3   "_rev": "1-d89012bd607f4bdf341ab6bd0cdaf6d8",
4   "name": "Ivanhoe",
5   "author": "Walter Scott",
6   "year": 1820
7 }

```

- \_\_\_ 11. To delete a document, issue a **DELETE** request for `$URL/$DATABASE/$DOCUMENT_ID?rev=$REV`. To delete the document of the “Ivanhoe” novel, complete the following steps, as shown in the following figure:
  - \_\_\_ a. Update the request URL to `$URL/$DATABASE/$DOCUMENT_ID?rev=$REV`.
  - \_\_\_ b. Update `$DOCUMENT_ID` with the ID of the document of the “Ivanhoe” novel, which is `novel_006`.
  - \_\_\_ c. Update `$REV` with the `_rev` value that you copied in the previous step

- \_\_\_ d. Update the HTTP method to **DELETE**.

- \_\_\_ e. Click **Send**.
- \_\_\_ f. Check the response that is shown in the following figure to verify the deletion of the document.

```

1 {
2   "ok": true,
3   "id": "novel_006",
4   "rev": "2-3e13de27a669e6f3dca4e738dad254be"
5 }
```



## Troubleshooting

If you receive a conflict error in the response, as shown in the following figure, consider these reasons:

- Document creation: the error indicates that the `_id` that is used to create this document is not unique and belongs to another document in the database
- Document update or deletion: either the `_id` or the `_rev` that are used in the request are wrong.

```

1 {
2   "error": "conflict",
3   "reason": "Document update conflict."
4 }
```

- \_\_\_ 12. In the next steps, you query the documents of the novels that are published after the year 1880 and sort them by year in ascending order. To achieve this query, create an index for the year field that you apply to the query in the next step.
- \_\_\_ 13. To create an index, send a **POST** request to `$URL/$DATABASE/_index` with a body that includes the fields to be indexed. To create an index for the `year` field, complete the following steps, as shown in the following figure:
- Update the request URL to `$URL/$DATABASE/_index`.
  - Update the HTTP method to **POST**.

The screenshot shows a browser window with a POST request to the URL `https://7471c743-b853-429f-8367-bcfbe87fef0e-bluemix.cloudantnosqldb.appdomain.cloud/novels/_index`. The method dropdown shows "POST".

\_\_ c. Update the request body with the following code, as shown in the following figure:

```
{
  "index": {
    "fields": [
      "year"
    ]
  },
  "name": "year-json-index",
  "type": "json"
}
```

The screenshot shows a POST request in Postman. The URL is `https://7471c743-b853-429f-8367-bcfbe87fef0e-bluemix.cloudantnosqldb.appdomain.cloud/novels/_index`. The "Body" tab is selected, and the JSON content is:

```

1  {
2    "index": {
3      "fields": [
4        "year"
5      ]
6    },
7    "name": "year-json-index",
8    "type": "json"
9  }
10

```

\_\_ d. Click **Send**.

\_\_ e. Check the response that is shown in the following figure and verify that the index is created successfully.

The screenshot shows a successful POST response in Postman. The status is 201 Created. The response body is:

```

1  {
2    "result": "created",
3    "id": "_design/4a53d1a437fa6d983869064423f5575a9de6047a",
4    "name": "year-json-index"
5  }

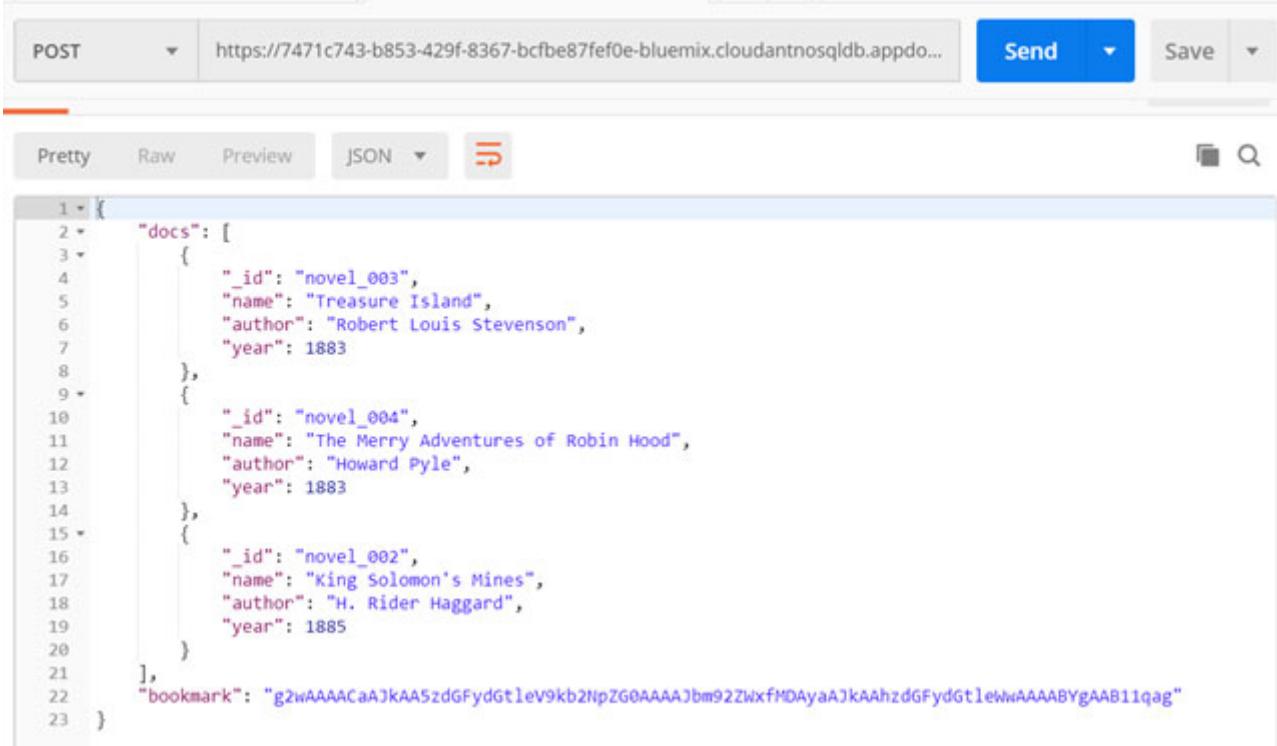
```

- \_\_ 14. To query a document, issue a **POST** request to `$URL/$DATABASE/_find` with a selector in the body. A selector is a JSON object that describes the criteria that is used to select documents. So, to query the documents of the novels that are published after the year 1880 and sort them by year in ascending order, complete the following steps:
- \_\_ a. Update the Request URL to `$URL/$DATABASE/_find`.

- \_\_\_ b. Keep the HTTP Method as **POST**.
- \_\_\_ c. Update the Request body with the Cloudant Query content:
  - “selector” specifies querying all documents with a `year` greater than 1880.
  - “fields” specifies that `_id`, `name`, `author`, and `year` should be returned in the query results.
  - “sort” specifies to sort by year. To sort by any other field, an index should be created for the other field.

```
{
  "selector": {
    "year": {
      "$gt": 1880
    }
  },
  "fields": [
    "_id",
    "name",
    "author",
    "year"
  ],
  "sort": [
    {
      "year": "asc"
    }
  ]
}
```

- \_\_\_ d. Click **Send**. The query results are returned in the response that is shown in the following figure. It is a good practice to create an index for each field that you are searching for in the selector to optimize query performance.



```

POST https://7471c743-b853-429f-8367-bcfbe87fef0e-bluemix.cloudantnosqldb.appdomain.com/_utils/database.html
Send Save
Pretty Raw Preview JSON
1 [
2   "docs": [
3     {
4       "_id": "novel_003",
5       "name": "Treasure Island",
6       "author": "Robert Louis Stevenson",
7       "year": 1883
8     },
9     {
10       "_id": "novel_004",
11       "name": "The Merry Adventures of Robin Hood",
12       "author": "Howard Pyle",
13       "year": 1883
14     },
15     {
16       "_id": "novel_002",
17       "name": "King Solomon's Mines",
18       "author": "H. Rider Haggard",
19       "year": 1885
20     }
21   ],
22   "bookmark": "g2wAAAACaAjkAA5zdGFydgtleV9kb2NpZG0AAAAJbm92ZWxfMDAyaAjkAAhzdGFydgtleWwAAAABYgAAB1lqag"
23 ]

```



## Information

By creating an index for the `year` field, you create and store a type of document that is called a *design document*, which you store it in the Cloudant database. Design documents are special documents that serve specific functions other than storing the data. Design documents are used to build indexes, validate updates, and format query results.



## Troubleshooting

If in the response to a Cloudant HTTP API request you receive the error HTTP status code 401 Unauthorized, as shown in the following figure, the access token is expired and you must create a new access token.

KEY	VALUE	DESCRIPTION ***
Authorization	Bearer eyJraWQiOiIyMDE3MTEyOSIsImF...	
Key	Value	Description

```

1 {
2   "error": "unauthorized",
3   "reason": "exp not in future"
4 }
  
```

## Part 7: Deleting your Cloudant service instance

In this part, you delete your Cloudant service instance from your IBM Cloud account by completing the following steps:

- 1. Open the IBM Cloud dashboard. In the Resource summary pane, click **Services**, as shown in the following figure.

Resource summary		View resources
Services	1	

- 2. You are redirected to the Resource list page. Under Services, you find your Cloudant service instance. Click the **More Actions** icon (the three horizontal dots), and then click **Delete**, as shown in the following figure.

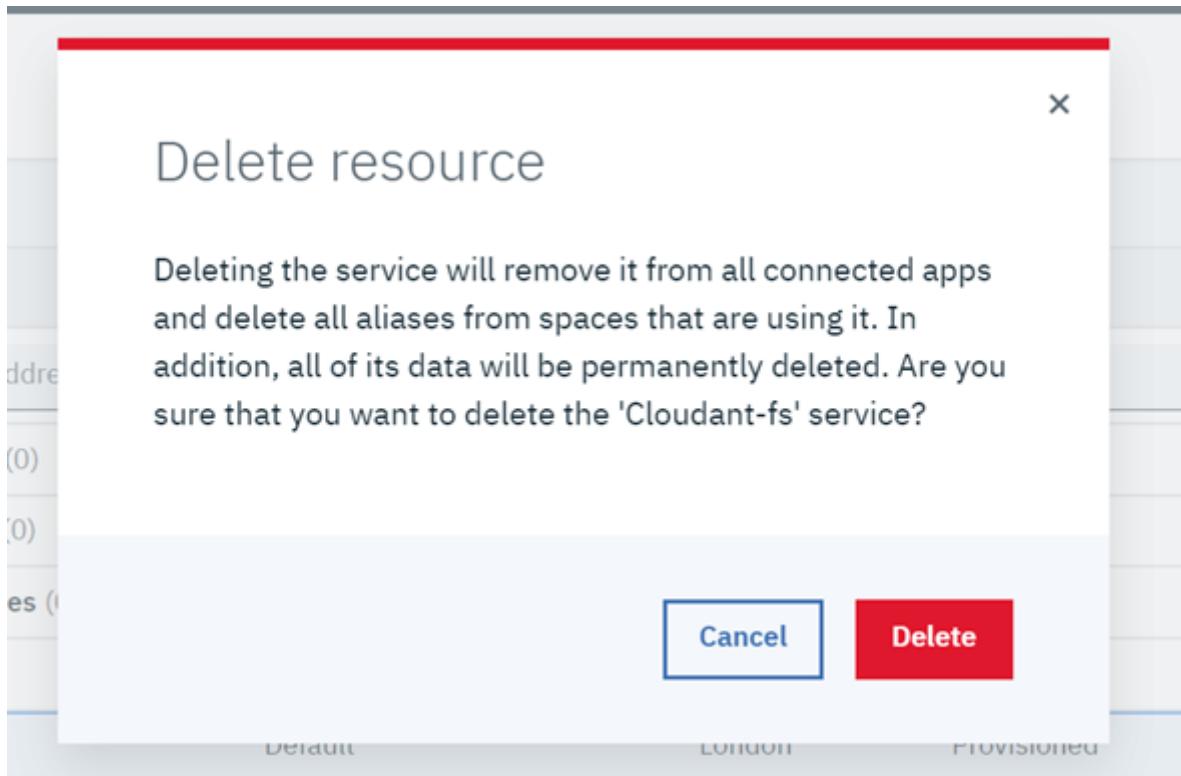
Resource list

Create resource

Collapse all | Expand all

Name ▾	Group	Location	Status	Tags
<input type="text"/> Filter by name or IP address...	<input type="text"/> Filter by group or org...	<input type="button"/> Filter...	<input type="text"/> Filter...	<input type="button"/> Filter...
> Kubernetes Clusters (0)				
> Cloud Foundry Apps (0)				
> Cloud Foundry Services (0)				
▼ Services (1)				
Cloudant-fs	Default	London	Provisioned	<input type="button"/> Delete
Cloudant-fs				

3. Confirm that you want to delete the service by clicking **Delete**, as shown in the following figure.



4. Wait until the service is deleted.

## Exercise review and wrap-up

In this exercise, you created an instance of Cloudant on IBM Cloud. You created credentials for the service to make requests from an HTTP API client by using these credentials. You explored the Cloudant dashboard and the features that are available in the Cloudant dashboard. You also created, read, updated, deleted, and queried Cloudant documents by using an HTTP API client.

---

# Exercise 4. Securing a web application with single sign-on (optional)

## Estimated time

00:45

## Overview

In this exercise, you secure an application by using the App ID service for single sign-on by authenticating your application through trusted server providers.

## Objectives

After completing this exercise, you should be able to:

- Create an App ID service.
- Bind the App ID service to an application to add single sign-on capability to the application.
- Describe different configurations in the App ID service.

## Introduction

In all web and mobile applications, security is important to protect all your sensitive data. You can use App ID to add authentication to your mobile and web apps and protect your APIs and back ends running on IBM Cloud. You also can use it to add an email and password-based sign-up and sign-in process, multi-factor authentication (MFA) with the App ID scalable user registry Cloud Directory, or a social media login, with Google or Facebook.

For employee apps, App ID uses SAML 2.0 federation so that users can sign in by using their enterprise credentials.

For all app users, you can enrich their profiles with additional information so that you can build engaging experiences. For example, if you know the age of the user, you can filter content in your application, or if you know the gender of the user, you can customize the application UI according to their preferences.

In this exercise, you create an App ID service, learn about different configurations, and bind the service to a Node.js application to explore the service capabilities.

## Requirements

This exercise requires:

- Access to the internet and the latest version of a modern web browser, such as Chrome, Firefox, and Safari.
- IBM Cloud account.
- IBM Cloud CLI.
- A valid Google account.

# Exercise instructions

In this exercise, you will complete the following tasks:

- 1. Create an App ID service instance.
- 2. Explore the App ID service.
- 3. Retrieve the App ID service credentials.
- 4. Create a Node.js app and connect it to the App ID service.
- 5. Configure the application to integrate with the App ID service.
- 6. Secure the sample application with the App ID service.
- 7. Clean up the environment.

## **Part 1: Creating an App ID service instance**

In this part, you create an App ID service to learn about its security capabilities and how to increase user interaction capabilities. Integrating an App ID service into your application can secure resources and add authentication even if you do not have a security background.

Complete the following steps:

- 1. Log in to IBM Cloud at <https://cloud.ibm.com/login>.
- 2. Click **Catalog**.
- 3. In the search field, enter “App ID”.
- 4. The App ID service is listed, as shown in the following figure.

The screenshot shows the IBM Cloud Catalog interface. At the top, there is a navigation bar with links for Catalog, Docs, Support, Manage, and Cloud Student's Account. Below the navigation bar, the word "Catalog" is displayed. A search bar contains the text "App ID". On the left, there is a sidebar with categories: All Categories (2), Compute, Containers, Networking, Storage, AI, Analytics, and Databases. The "All Categories (2)" link is highlighted. In the main content area, there is a section titled "Security and Identity" which contains a card for the "App ID" service. The card includes a lock icon, the text "App ID", "Lite • IBM • IAM-enabled", and a description: "User Authentication and User Profiles for your apps.". There is also a "Feedback" button on the right side of the card. The overall interface is clean and modern, typical of cloud provider catalog pages.

- 5. Click the **App ID** service under Security and Identity.
- 6. Create an App ID service instance by accepting the default values and then click **Create**, as shown in the following figure.

The screenshot shows the IBM Cloud interface for creating an App ID. At the top, there's a navigation bar with 'IBM Cloud', search, catalog, docs, support, manage, and account information. Below it, the 'App ID' service card is displayed. It shows the service is 'Lite', 'IBM Service', and 'IAM-enabled'. The author is 'IBM' and the date of last update is '11/11/2019'. There are 'Create' and 'About' buttons. A 'Need Help?' link leads to 'Contact Support' and 'View docs API docs'. On the right, a 'Summary' section shows the 'App ID' is 'London', 'Region' is 'London', 'Plan' is 'Lite', 'Service name' is 'App ID-7g', and 'Resource group' is 'Default'. A 'Create' button is prominent, along with 'Add to estimate' and 'View terms' links. A 'Feedback' button is also visible.



### Note

Save the service name and region for later use.

7. After you click **Create**, you are redirected to the App Service Overview window, as shown in the following figure.

The screenshot shows the 'Resource list / App ID-7g' page. The service is 0.3% used with 997 authentication events available. It's located in the 'Default' resource group and 'London' location. There are 'Details', 'Add Tags', and 'Edit' buttons. The main content area is titled 'Welcome to the App ID service' and includes a message: 'Take your first steps with App ID authentication and begin building secure and engaging digital applications. With this step-by-step walkthrough, you'll be up and running in no time.' A 'Download Sample' button is present, along with a note: 'You can also add APP ID to an existing app, learn more'. On the left, a sidebar lists navigation options: Overview, Manage Authentication, Cloud Directory, Identity Providers, User Profiles, Login Customization, Applications, Connections, Service credentials, and Plan.

## Part 2: Exploring the App ID service

In this part, you discover more about App ID configuration terms and concepts, such as identity providers and user profiles. You must configure them when you use the service to secure your application.

Complete the following steps:

- 1. In the side bar menu, click **Manage Authentication** to configure identity providers. Two tabs, which are called Identity Providers and Authentication Settings, are displayed, as shown in the following figure.

Identity Provider	Status	Action
Cloud Directory	Enabled	Edit
SAML 2.0 Federation	Disabled	Edit
Facebook	Enabled	Edit
Google	Enabled	Edit
Custom Identity	Disabled	Edit

An identity provider creates and manages information about an entity, such as a user, a functional ID, or an application. The provider verifies the identity of the entity by using credentials, such as a password. Then, the identity provider sends the identity information to another service provider. Because the identity provider authenticates the entity, the App ID can authorize it and grant access to your apps.



### Note

Make sure that you have a Google account because you are going to use it in this exercise. If you do not have a Google account, use Facebook instead.

- 2. Click the **Authentication Settings** tab. You can add your application URLs in the **Add web redirect URLs** field to “whitelist” your application to redirect to the App ID service. See the following figure. You add your application URL later in this exercise.



## Information

A [redirect URL](#) is a technique that is used to send a user from one URL to another. For example, if the user enters the URL `website.com/page-a`, a redirect can cause the browser to open the web page `website.com/page-b`.

This function is useful when, for example, you move your website and want to shut down the old one. To prevent users from receiving a `404 Not Found` error, you redirect the old URLs to the new ones.

- 
- 3. App ID provides a login window that the user uses to log in to one of the configured identity providers. Click **Login Customization** to change the UI of the redirected login window, as shown in the following figure.

- 4. Expand **Cloud Directory** and click **Users**. This tab is used to add Cloud Directory users that can log in by using a user name and password.

**Note**

A cloud directory is a user registry that is maintained in the cloud. When users sign up for your app, they are added to your directory of users. The directory acts as another identity provider if you are not depending on Facebook, Google, or SAML for your logins.

- 5. Click **Applications**. In this tab, you see all the Cloud Foundry applications that are connected to your service, as shown in the following figure.

The screenshot shows the App ID dashboard interface. On the left, a sidebar menu lists various options: Overview, Manage Authentication, Cloud Directory, Identity Providers (with SAML 2.0 Federation, Facebook, Google, and Custom Identity), User Profiles, Login Customization, Applications (which is selected and highlighted in blue), Connections, Service credentials, and Plan. The main content area is titled "Resource list / App ID-7g". It displays basic information: Resource group: Default, Location: London, and Add Tags. Below this, a large section is titled "Applications". It contains a brief description: "Keep track of your server-side apps, also known as OAuth clients, that use App ID for authentication. Each app is associated with a client ID and a secret, which you need in order to connect with App ID." A note below it says: "Note: In the Connection tab, you can see a list of Cloud Foundry web apps that can be bound to App ID to use it for authentication. For more information see our [documentation](#)." At the bottom right of this section is a blue button labeled "Add application". At the very bottom of the main content area is a dropdown menu labeled "Applications".

### **Part 3: Retrieving the App ID service credentials**

In this part, you retrieve the App ID service credentials. You use this information to customize the sample app for integration with the App ID service.

Complete these steps:

- 1. Return to the App ID dashboard and click **Service credentials**.
- 2. If there are no service credentials, click **New credentials +**. If the service credentials exist, skip to View credentials.

Resource list /

**App ID-7g** 0.1% Used | 999 Authorized user available [details](#)

Resource group: Default Location: London [Add Tags](#)

Service credentials

Credentials are provided in JSON format. The JSON snippet lists credentials, such as the API key and secret, as well as connection information for the service. [Learn more](#)

New credential +

Click **New credentials** to create a set of credentials for this instance

3. At the Add new credentials window, keep the default values, and click **Add**.

### Add new credential

Name:

Role:

Select Service ID (Optional)

Add Inline Configuration Parameters (Optional):

**Add**

- 4. Expand **View credentials** under ACTIONS. The App ID service credentials are displayed in JSON format, as shown in the following figure.

KEY NAME	DATE CREATED	ACTIONS
Service credentials-1	JUL 8, 2019 - 09:11:48 AM	<a href="#">View credentials</a>

```

{
  "apikey": "Ckso4prMkE2tYfouEajlxE9FC-tVq-DvuXBe-MmtJ",
  "appidServiceEndpoint": "https://eu-gb.appid.cloud.ibm.com",
  "clientId": "c36e18d4-9a7d-4ad4-aff3-cc6b0ab709df",
  "discoverEndpoint": "https://eu-gb.appid.cloud.ibm.com/oauth/v4/9dc942b4-5f63-4da8-8433-7f9f43dfe648/we
  ll-known/openId-configuration",
  "iam_apikey_description": "Auto-generated for key c36e18d4-9a7d-4ad4-aff3-cc6b0ab709df",
  "iam_apikey_name": "Service credentials-1",
  "iam_role_crn": "cnvibluemixpubliciam::serviceRole:Reader",
  "iam_serviceid_crn": "cnvibluemixpublic:iam-identity::a/bd271530789f4046b990411373423ab5::serviceId:5
  serviceId-93e2acfd-9a92-4924-b035-8ce347614c96",
  "managementUrl": "https://eu-gb.appid.cloud.ibm.com/management/v4/9dc942b4-5f63-4da8-8433-7f9f43dfe648",
  "oauthServerUrl": "https://eu-gb.appid.cloud.ibm.com/oauth/v4/9dc942b4-5f63-4da8-8433-7f9f43dfe648",
  "profileUrl": "https://eu-gb.appid.cloud.ibm.com"
}

```

- 5. Click the Copy to clipboard icon and paste the credentials in a text editor. You use this information to configure your application to integrate with the App ID service.

## Part 4: Creating a Node.js app and connecting it to the App ID service

In this part, you create a Node.js app on IBM Cloud and connect it to the App ID service that you created in Part 1.

### Creating a Node.js application on IBM Cloud

Create a Node.js application by performing the following steps:

- 1. From the IBM Cloud catalog search for **Cloud Foundry**.
- 2. Select the **Cloud Foundry** service.
- 3. In the Cloud Foundry overview page, select **.js** under **Application Runtimes** to create a Node.js application as shown in the following figure.

The screenshot shows the Cloud Foundry Application Runtimes page. On the left, there's a sidebar with links for Overview, Public, Enterprise, Getting Started, Environments, Applications, and Services. The main area is titled "Application Runtimes" and features a sub-section titled "A range of popular language runtimes that make working with your chosen language easy and automatic." Below this, there are icons for Java, Node.js, Swift, Python, Ruby, .NET, PHP, Go, and Tomcat.

- 4. Enter a unique app name, for example, **appId-application-xxxxx** where “xxxxx” is a unique identifier for your application. Keep the default values for the other fields as shown in the following figure.

The screenshot shows the "Create" dialog box for a new Cloud Foundry application. In the "Select a runtime" section, "SDK for Node.js™ Version 3.x" is selected. The "Summary" section on the right shows the following details:

Cloud Foundry App	Free
Region:	London
Plan:	Lite
Runtime:	SDK for Node.js™
App name:	appId-application-1234
Host name:	appId-application-1234
Domain:	eu-gb.cf.appdomain.cloud
Choose an organization:	cloud-foundry@ibm.com
Choose a space:	dev
Tags:	env:dev, version:1

At the bottom, there are buttons for "Create", "Add to estimate", and "View terms". A "FEEDBACK" button is also visible.

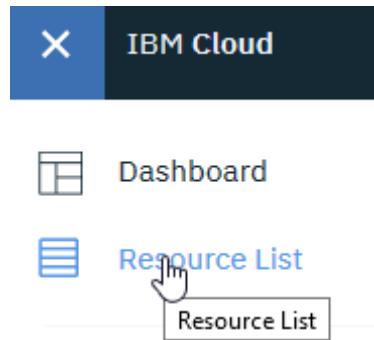
- 5. Click **Create**.

## Connecting the application to the App ID service

In this part, you connect the Node.js Cloud Foundry app that you created on IBM Cloud to the App ID service that you created in Part 1.

Perform the following steps:

- 1. Access the App ID service. From the Navigation Menu (upper left) select Resource List.



- \_\_\_ 2. Expand **Services** and click your App ID service. The Welcome to the App ID service page is displayed.
- \_\_\_ 3. From the options on the left, click **Connections** as shown in the following figure.

A screenshot of the 'Welcome to the App ID service' page. On the left, a sidebar menu lists several options: Overview, Manage Authentication, Cloud Directory, Identity Providers, User Profiles, Login Customization, Applications, Connections (which is highlighted with a blue background), Service credentials, and Plan. The main content area shows the title 'Welcome to the App ID service' and a sub-section 'Take your first steps with App ID authentication and begin building secure step-by-step walkthrough, you'll be up and running in no time.' Below this, there's a 'Download Sample' button and a note 'You can also add APP ID to an existing app, [learn more](#)'. At the top right, it shows 'Resource list / App ID-7g 0.3% Used | 997 Authentication event available' and 'Resource group: Default Location: London Add Tags'.

- \_\_\_ 4. Click **Create connection**.

The screenshot shows the IBM Cloud Connections interface. On the left, there's a sidebar with options like Overview, Manage Authentication, Cloud Directory, Identity Providers, User Profiles, Login Customization, Applications, Connections (which is selected), Service credentials, and Plan. The main area is titled 'Resource list / App ID-7g'. It shows '0.3% Used | 997 Authentication event available'. Below that, it says 'Resource group: Default Location: London Add Tags'. A large blue button labeled 'Create connection +' is visible. In the center, there's a placeholder icon with a question mark and the text 'No connected Cloud Foundry applications'. A sub-instruction below says 'Click Create connection to connect App ID-7g to one of your existing Cloud Foundry applications.'

- 5. Make sure that the correct region and organization (your IBM Cloud ID) are selected and choose your Node.js application, as shown in the following figure.

The screenshot shows the 'Connect Existing Cloud Foundry App' dialog box. The left sidebar is identical to the previous one. The main dialog has fields for REGION (London), CLOUD FOUNDRY ORG (selected), and CLOUD FOUNDRY SPACE (dev). Below these are sections for CONNECTION LOCATION (dev // eu-gb) and CLOUD FOUNDRY APPS. A table lists one app: 'appID-application-1234' with a status of '0/1 Not running'. A blue 'CONNECT' button is at the bottom right.

- 6. Click **CONNECT**.  
— 7. In the next window, click **Connect & restage app**, as shown in the following figure.

## Connect Cloud Foundry Application

To connect, you can customize the ServiceID and access role used for this binding. Restaging your app is required to connect this service and may result in application downtime.

### Access Role for Connection i

Reader

### Service ID for Connection (Optional) i

Select Service ID...

### Add Inline Configuration Parameters (Optional): i

Cancel

Connect & restage app

8. Click **Restage**.

### Restage app

Your 'appID-application-1234' app needs to be restaged. Do you want to restage it now?

Cancel

Restage

## Part 5: Configuring the application to integrate with the App ID service

In this part, you will:

- Download a sample Node.js application code from a Git repo.
- Modify application files to configure the sample app to integrate it with the App ID service.
- Deploy the app by pushing the changes out to Cloud Foundry on IBM Cloud with IBM Cloud CLI.

### Downloading the sample Node.js app code

Complete these steps:

- 1. Download the sample app. Access the sample app at this link <https://github.com/IBM-SkillsAcademy/Cloud-Application-Developer/blob/master/CloudAppDev/Ex4/SampleApp-Node.zip> and click **Download**.



- 2. Open the file and extract its content into your preferred folder. The sample app files are shown in the following figure.

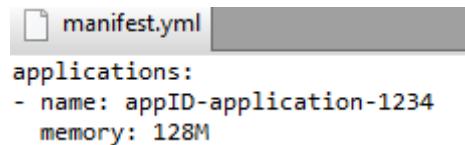
public	7/7/2019 8:27 PM	File folder
protected	7/7/2019 8:27 PM	File folder
.git	7/7/2019 8:27 PM	File folder
manifest.yml	7/7/2019 8:27 PM	YML File
kube_deployment.yml	7/7/2019 8:27 PM	YML File
.gitignore	7/7/2019 8:27 PM	Text Document
README.md	7/7/2019 8:27 PM	MD File
package.json	7/7/2019 8:27 PM	JSON File
localdev-config.json	7/7/2019 8:27 PM	JSON File
app.js	7/7/2019 8:27 PM	JavaScript File
LICENSE	7/7/2019 8:27 PM	File
Dockerfile	7/7/2019 8:27 PM	File
.dockerignore	7/7/2019 8:27 PM	DOCKERIGNORE File
.cignore	7/7/2019 8:27 PM	CIGNORE File

## Modifying the sample app files

In this section, you modify the `manifest.yaml` file to customize your app name and you modify the `localdev-config.json` file to integrate the sample app with the App ID service that you created in Part 1.

Complete these steps:

- 1. Edit the **manifest.yaml** file and change the application name to match the unique name of the application that you created in Part 4. For example, **appId-application-xxxx** where “xxxx” is your unique identifier. The following figure shows an example.



```
manifest.yaml
applications:
- name: appId-application-1234
  memory: 128M
```

- 2. Edit the **localdev-config.json** file with your preferred editor.
- 3. Replace the values for `clientId`, `oauthServerUrl`, `profilesUrl`, `secret`, and `tenantId` that are shown in the following figure, with the corresponding values from the App ID service credentials that you saved in Part 3.

```
clientId: "4e01274f-4d63-4261-aa48-08597a385fe3",
"oauthServerUrl": "https://eu-gb.appid.cloud.ibm.com/oauth/v4/d521f007-ead0-4894-9f3d-88f49c79ffd7",
"profilesUrl": "https://eu-gb.appid.cloud.ibm.com",
"secret": "NTI3Zjg5NjAtMDg4NC00YjAwLTk5NDEtNGEzYjUxYjY2ZWQy",
"tenantId": "d521f007-ead0-4894-9f3d-88f49c79ffd7",
"redirectUri": "http://localhost:3000/ibm/cloud/appid/callback"
```

## Deploying the sample app to IBM Cloud

To deploy the sample application to IBM Cloud, complete the following steps:

- 1. Open the Command Prompt and set the current directory to the folder where the sample app code is.
- 2. Login to IBM Cloud by running the following command:

```
ibmcloud login
```

```
C:\IBM-Cloud\SampleApp-Node>ibmcloud login
API endpoint: https://cloud.ibm.com
Region: eu-gb

Email: cloudstudent51@gmail.com

Password:
Authenticating...

Targeted account [Cloud Student's Account (bd271530789f4846b990411373423abf)]

API endpoint: https://cloud.ibm.com
Region: eu-gb
User: cloudstudent51@gmail.com
Account: Cloud Student's Account (bd271530789f4846b990411373423abf)
Resource group: No resource group targeted, use 'ibmcloud target -g RESOURCE GROUP'
CF API endpoint:
Org:
Space:

Tip: If you are managing Cloud Foundry applications and services
- Use 'ibmcloud target --cf' to target Cloud Foundry org/space interactively, or use 'ibmcloud target --cf-api ENDPOINT'
- -o ORG -s SPACE' to target the org/space.
- Use 'ibmcloud cf' if you want to run the Cloud Foundry CLI with current IBM Cloud CLI context.

C:\IBM-Cloud\SampleApp-Node>
```



### Note

If the incorrect region is selected by default, you can change it by running the command `ibmcloud target -r <region>`, for example `ibmcloud target -r eu-gb`. For the complete list of available regions, see [Regions](#) at [https://cloud.ibm.com/docs/cloud-foundry-public?topic=cloud-foundry-public-endpoints#endpoints\\_regions](https://cloud.ibm.com/docs/cloud-foundry-public?topic=cloud-foundry-public-endpoints#endpoints_regions)

- 3. Use `--cf-api` to specify the Cloud Foundry API endpoint to which to deploy the application. Select it based on the region where the application was created. Run the following command:

```
ibmcloud target --cf-api <CF API ENDPOINT> -o <ORG> -s <SPACE>
```

In this example, the command is:

```
ibmcloud target --cf-api https://api.eu-gb.cf.cloud.ibm.com -o <your-email>
-s dev
```

**Note**

The organization is set by default to your IBMid, which is the email that you use to log in to IBM Cloud, and the space is set by default to dev. For the complete list of API endpoints see API Endpoints at

<https://cloud.ibm.com/docs/cloud-foundry-public?topic=cloud-foundry-public-endpoints#api-endpoint-options>

The following figure shows an example of the command output.

```
C:\IBM-Cloud\SampleApp-Node>ibmcloud target --cf-api https://api.eu-gb.cf.cloud.ibm.com -o cloudstudent51@gmail.com -s dev
Targeted Cloud Foundry (https://api.eu-gb.cf.cloud.ibm.com)
Targeted org [REDACTED]
Targeted space dev

API endpoint: https://cloud.ibm.com
Region: eu-gb
User: [REDACTED]
Account: Cloud Student's Account (bd271530789f4046b990411373423abf)
Resource group: No resource group targeted, use 'ibmcloud target -g RESOURCE GROUP'
CF API endpoint: https://api.eu-gb.cf.cloud.ibm.com (API version: 2.142.0)
Org: [REDACTED]
Space: dev

C:\IBM-Cloud\SampleApp-Node>
```

- 4. Deploy the application on IBM Cloud by running the following command:

ibmcloud cf push

The following figure shows the output of the command.

```
Waiting for app to start...

name: appID-application-1234
requested state: started
routes: appID-application-1234.eu-gb.cf.appdomain.cloud
last uploaded: Thu 28 Nov 18:26:15 CST 2019
stack: cflinuxfs3
buildpacks: sdk-for-nodejs

type: web
instances: 1/1
memory usage: 128M
start command: npm start
      state  since          cpu    memory      disk      details
#0   running  2019-11-29T00:26:43Z  0.1%  41.6M of 128M  80.6M of 1G

C:\IBM-Cloud\SampleApp-Node>
```

## Copying the sample app URL

In this section, you copy the sample app URL to use it later in the configuration.

Complete these steps:

- 1. Open the application in the browser. Display IBM Cloud **Resource List**, and expand **Cloud Foundry Apps**, as shown in the following figure.

Resource list

Create resource

Collapse all | Expand all

Name	Group	Location	Status	Tags
<input type="text"/> Filter by name or IP address...	<input type="text"/> Filter by group or org...	<input type="text"/> Filter...	<input type="text"/> Filter...	<input type="text"/> Filter...
> Devices (0)				
> VPC infrastructure (0)				
> Clusters (0)				
▼ Cloud Foundry apps (1)				
appID-application-1234	/dev	London	Started	...
> Cloud Foundry services (1)				
Services (1)				
> Storage (0)				
> Network (0)				
> Cloud Foundry enterprise environments (0)				

- 2. Click your application name under Cloud Foundry Apps.
- 3. Click **Visit App URL**, as shown in the following figure.

Resource list /

Org: cloudstudent51@gmail.com Location: London Space: dev Add Tags

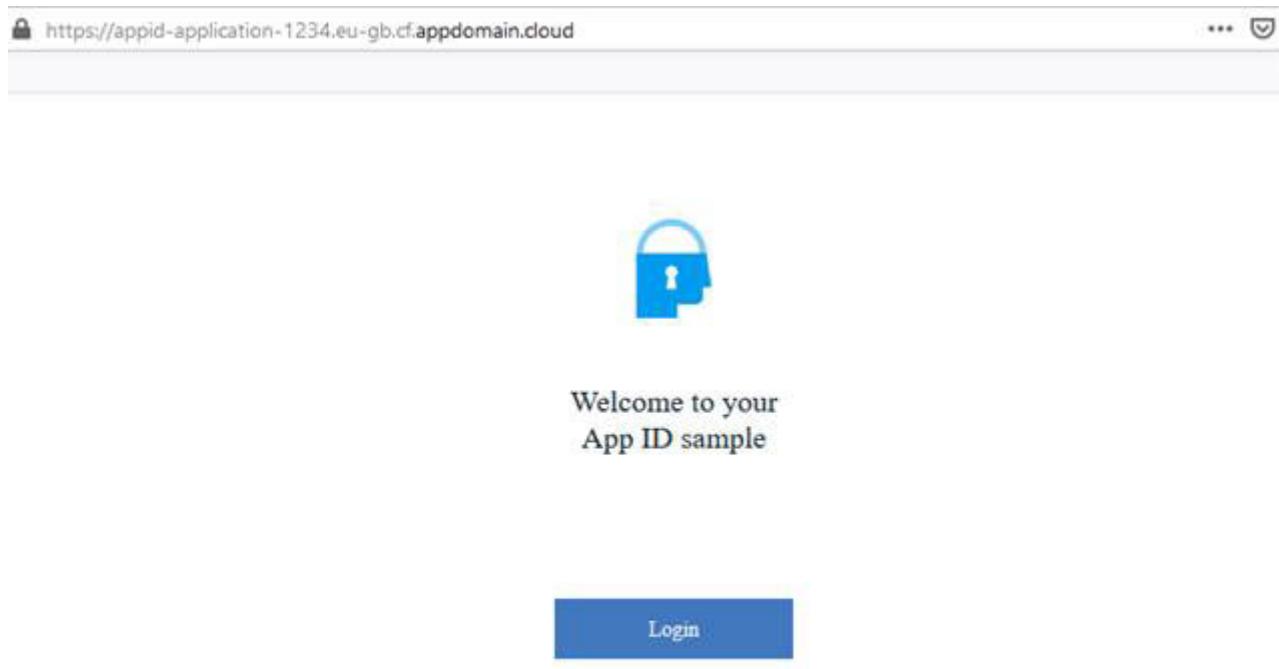
Routes : ▾

appID-application-1234 ● This app is awake. Visit App URL

Runtime

<b>js</b>	<b>1</b>	<b>128</b>	<b>128</b>
BUILDPACK SDK for Node.js™	INSTANCES All instances are running Health is 100%	MB MEMORY PER INSTANCE	TOTAL MB ALLOCATION 128 MB still available

- The application opens in another tab, as shown in the following figure.



- \_\_\_ 4. Copy and save the URL for later use.

## **Part 6: Securing the sample application with the App ID service**

In this part, you learn how to use the App ID service and bind it to your application to secure your application and authenticate through different identity providers.

### **Configuring the application URL**

Now, you configure the URL of the deployed application in the service instance. You explored this process in Part 2 when you looked at Managing Identity Providers.

Complete the following steps:

- \_\_\_ 1. Open a new browser tab and access your App ID service dashboard.
- \_\_\_ 2. Click **Manage Authentication** and then click **Authentication Settings**, as shown in the following figure.

Resource list /

App ID-7g 0.8% Used | 992 Authentication event available

Resource group: Default Location: London Add Tags

## Manage Identity Providers

**Identity Providers** **Authentication Settings**

Add web redirect URLs i

Enter a Redirect URL

http://localhost:3000/\*

## Sign-in Expiration

- 3. Add the application URL that you saved previously to **Add web redirect URLs** so that the application can redirect the user to the App ID for authentication. Add /\* after the URL and then click + as shown in the following figure.

**Add web redirect URLs**  
Registering your redirect\_uri with App ID ensures that only authorized clients are allowed to participate in the authorization workflow. Make sure you register the redirect\_uri of applications that you trust. For more information about redirects [see the docs](#).

https://appid-application-1234.eu-gb.cf.appdomain.cloud/\*

We recommend that you do not use wildcards as part of your URL in production.

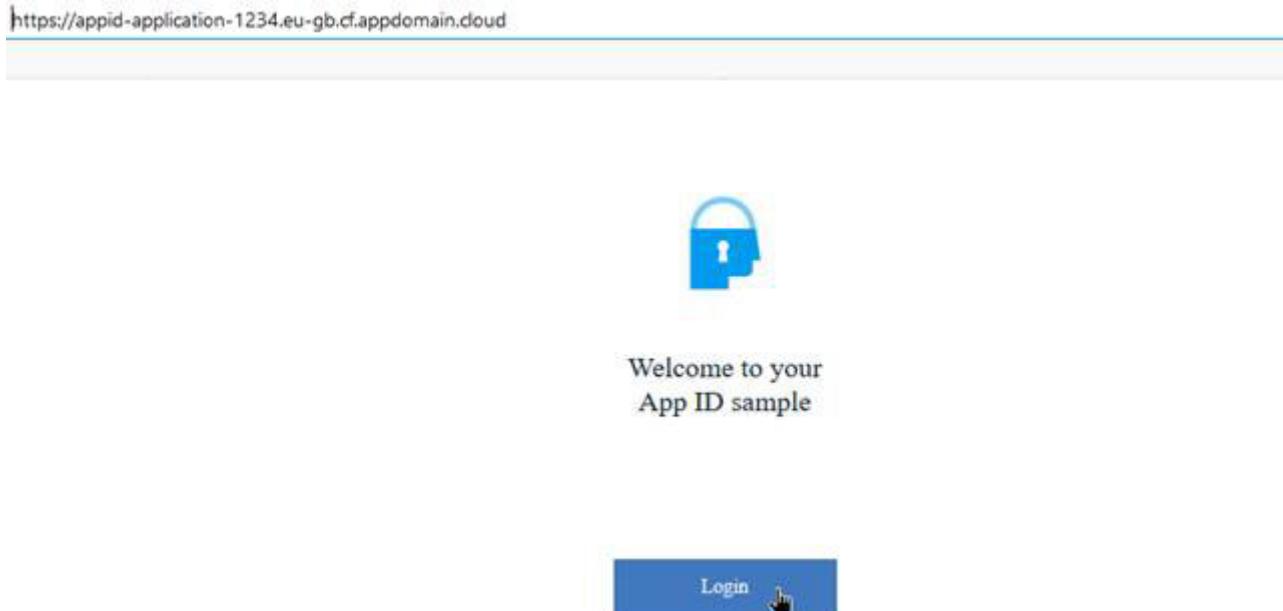
 

Now, the App ID accepts requests from only this URL.

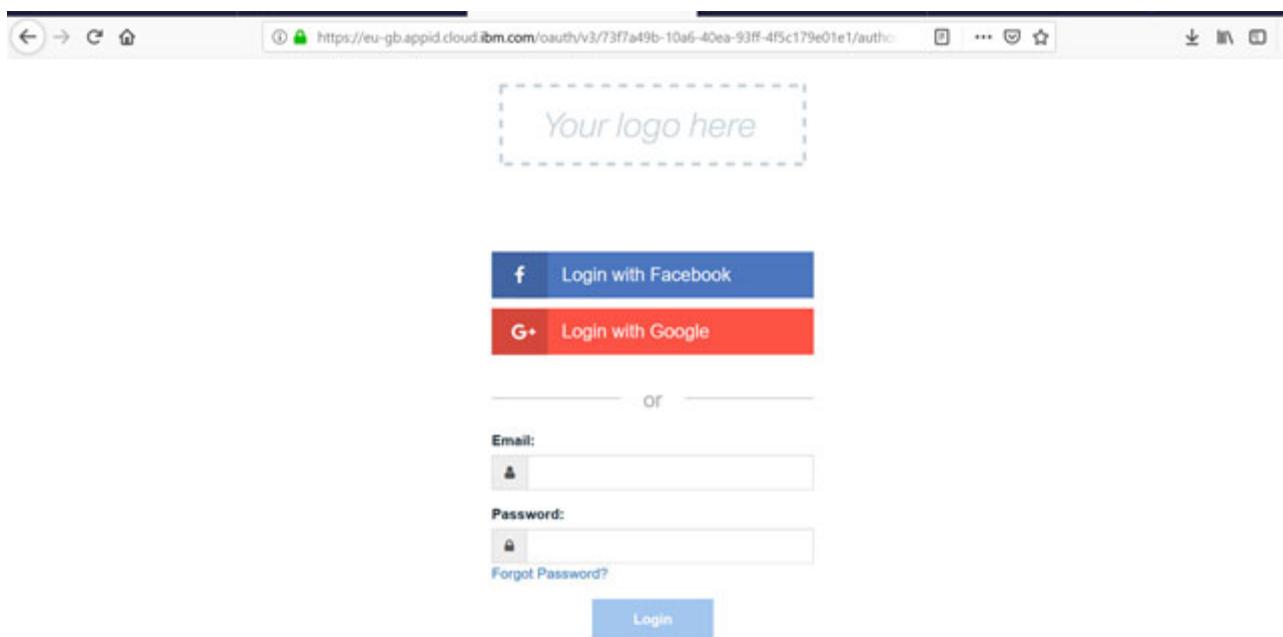
## Signing in to the secured application

To verify that authentication is now required to access your application, complete the following steps:

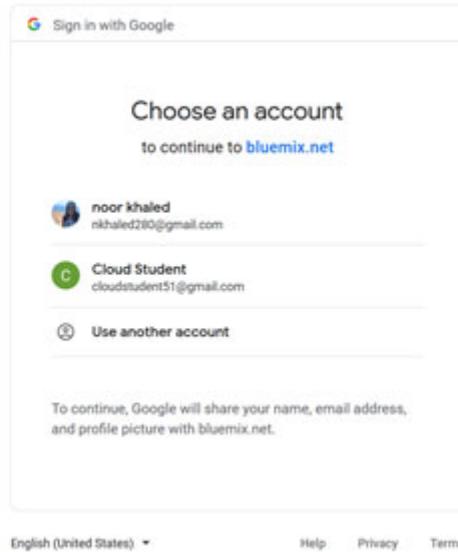
- 1. On the opened application tab, take note of the URL of the application and click **Login**, as shown in the following figure.



2. You are redirected to the App ID login to verify your identity, as shown in the following figure.



The identity providers that you saw in Part 2 are listed. If you have a Gmail account, login with your Google credentials. If you want to log in by using your Facebook account, click Login with Facebook. In this exercise, we chose to log in by using Google, as shown in the following figure.



- \_\_\_ 3. After you enter your Google account credentials, you are logged in to your application, as shown in the following figure. The application received the user's name and image from the identity provider (Google in this case).

Hi noor khaled, Congratulations!  
You've made your first authentication.

What you can do next:

Add App ID to your own app

ID Token

```

▼ {
  iss: https://eu-gb.appid.cloud.ibm.com/oauth/v4/d521f007-ead8-4894-9f3d-88f49c79ffd7,
  ▼ aud: [
    "fa8833f8-3d4e-46cf-a5e0-3590a309a134"
  ],
  exp: 1562538968,
}

```

- \_\_\_ 4. Browse through the ID Token to display the user's verified token and the basic user's information as shown in the following figure.

ID Token

```

{
  iss: "https://eu-gb.appid.cloud.ibm.com/oauth/v4/d521f007-ead8-4894-9f3d-88f49c79ffd7",
  aud: [
    "fa8833f8-3d4e-46cf-a5e0-3590a309a134"
  ],
  exp: 1562538968,
  tenant: "d521f007-ead8-4894-9f3d-88f49c79ffd7",
  iat: 1562535368,
  email: "nkhaled280@gmail.com",
  name: "noor khaled",
  locale: "en",
  picture: "https://lh3.googleusercontent.com/-jRZn4KGNJNs/AAAAAAAQc/8X6GPhZAQZ0/photo.jpg",
  sub: "b41bbd89-6563-4285-9c20-9ccd6485e898",
  identities: [
    {
      provider: "google",
      id: "108621067275761862267"
    }
  ],
  amr: [
    "google"
  ],
  ver: 4
}

```

## **Part 7: Cleaning up the environment**

In this part, you delete the instance of the application and App ID service that you created.

Complete the following steps:

- \_\_\_ 1. Open the IBM Cloud dashboard.
- \_\_\_ 2. Under Cloud Foundry Apps, click **Actions** (the three dots) for your application, as shown in the following figure.

## Resource list

[Create resource](#)[Collapse all](#) | [Expand all](#)

Name	Group	Location	Status	Tags
<input type="text"/> Filter by name or IP address...	<input type="text"/> Filter by group or org...	<input type="text"/> Filter...	<input type="text"/> Filter...	<input type="text"/> Filter...
> Devices (0)				
> VPC infrastructure (0)				
> Clusters (0)				
Cloud Foundry apps (1)				
appID-application-1234	/dev	London	Started	...
> Cloud Foundry services (1)				<a href="#">Stop</a>
> Services (1)				<a href="#">Restart</a>
> Storage (0)				<a href="#">Edit name</a>
> Network (0)				<a href="#">Add tags</a>
> Cloud Foundry enterprise environments (0)				<a href="#">Delete</a>

- 3. Click **Delete**.
- 4. Select all the associated services and routes and click **Delete**, as shown in the following figure.

X

Are you sure you want to delete the 'appID-application-1234' app?

After 'appID-application-1234' app is deleted, some services and routes will not be associated with any app.

The screenshot shows a modal dialog box with two tabs: 'Services' (selected) and 'Routes'. The main area contains the following text:

Select the services to be deleted when the app is deleted.  
Services that are not deleted can still be managed from the resource list.

A checkbox is checked next to 'App ID-7g'. At the bottom of the dialog are two buttons: 'Cancel' and a red 'Delete' button, which has a hand cursor icon pointing to it.

- \_\_\_ 5. Expand **Services**.
- \_\_\_ 6. Click the **Actions** menu (three dots) for your **App ID** service and select **Delete**.

The screenshot shows the 'Services' section of the Cloud Foundry Services page. It lists one service named 'App ID-7g' with details: Default, London, Provisioned. To the right of the service name is a context menu with options: 'Rename', 'Add tags', and 'Delete'. A hand cursor icon is pointing to the 'Delete' button.

- \_\_\_ 7. At the Delete resource prompt, click **Delete** to confirm.

## 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 'App ID-7g' service?



**End of exercise**

## Exercise review and wrap-up

In this exercise, you learned how to create an App ID service and explore the various tabs of the service. You also learned some terms, such as identity providers and cloud directory users.

You downloaded a sample Node.js application, bound it to the App ID service, deployed it on IBM Cloud, and tested the authentication by using an identity provider (in this example, Google).

You saw how Google verified the user and sent all the details about this user to the application to provide a better user experience.

You saw how App ID supports user profiles when you logged in with Google and clicked the user information. You found all information about the user, which you can use to enhance the user experience in your application.

You deleted the application instance to perform other exercises.

# Exercise 5. Managing IBM Kubernetes Service clusters

## Estimated time

00:30



### Important

It takes about 30 minutes to provision a cluster. Part 1 should be started during the lecture or before a break.

## Overview

This exercise demonstrates how to create an IBM Kubernetes Service cluster and manage it by using the `kubectl` CLI.

## Objectives

After completing this exercise, you should be able to:

- Create an IBM Kubernetes Service cluster.
- Connect to a cluster on IBM Cloud Kubernetes Service.
- List the worker nodes in a cluster.

## Introduction

IBM Cloud Kubernetes Service creates a cluster of compute hosts and deploys highly available containers. It provides intelligent scheduling, self-healing, horizontal scaling, service discovery and load balancing, automated rollouts and rollbacks, and secret and configuration management.

Developers can easily roll out and roll back application versions, whether they are collaborating in development and test environments or deploying to production. Therefore, developers can spend more time coding and less time working with the infrastructure.

## Requirements

- You must have a Pay-As-You-Go or Subscription IBM Cloud account so that you can create a cluster. If you are a university student, you can apply for a no-charge feature code.
- The IBM Cloud CLI must be installed.
- The Kubernetes CLI, the IBM Cloud Kubernetes Service plug-in, and the IBM Cloud Container Registry plug-in must be installed.

## Exercise instructions

In this exercise, you complete the following tasks:

- \_\_\_ 1. Create an IBM Cloud Kubernetes Service cluster.
- \_\_\_ 2. Connect to your cluster on IBM Cloud Kubernetes Service.
- \_\_\_ 3. List the worker nodes in your cluster.

### **Part 1: Creating an IBM Cloud Kubernetes Service cluster**

Before you dive into Kubernetes, you must provision a cluster for your containerized app. A cluster is a set of resources, worker nodes, networks, and storage devices that keep apps highly available. After you have your cluster, you can deploy your apps into the containers.



#### Important

It takes about 30 minutes to provision a cluster. This part should be started during the lecture or before a break.

Complete the following steps:

- \_\_\_ 1. Log in to <https://cloud.ibm.com> and click **Catalog**, as shown in the following figure.

The screenshot shows the IBM Cloud dashboard. At the top, there is a navigation bar with links for 'IBM Cloud', 'Catalog' (which is highlighted with a cursor), 'Docs', 'Support', 'Manage', and a user account 'Uni Student02's Account'. Below the navigation bar, the word 'Dashboard' is displayed. On the right side of the dashboard, there are two buttons: 'Upgrade account' and 'Create resource'. The main area of the dashboard is currently empty, indicating no active resources.

- \_\_\_ 2. In the search field, enter **Kubernetes Service** and click the service, as shown in the following figure.

The screenshot shows the 'Catalog' page. At the top, there is a search bar containing the text 'kubernetes service'. Below the search bar, there are two tabs: 'Services (3)' (which is selected) and 'Software (2)'. On the left, there is a sidebar with a 'All Categories (3)' dropdown menu and a list of categories: VPC Infrastructure, Compute (2), Containers (2), Networking, Storage (1), AI, Analytics, Databases (1), Developer Tools (1), Integration, and Internet of Things (1). The main content area is titled 'Services' and contains a brief description: 'Explore our broad portfolio of managed services for infrastructure, developer tools, and more to build your apps on the public cloud.' Below this, there is a section titled 'All Categories' with three service cards: 'Container Registry' (IBM), 'Kubernetes Service' (IBM), and 'Portworx Enterprise' (Third party). The 'Kubernetes Service' card is highlighted with a cursor, indicating it has been selected.

- \_\_\_ 3. In the upper right, click **Create**, as shown in the following figure.

The screenshot shows the IBM Cloud Kubernetes Service page. At the top right is a blue 'Create' button. Below it, there's a section titled 'Helpful links' with several links: 'Creating a cluster and setting up your environment', 'Deploying a Watson app to your cluster', 'Managed add-ons for Istio and Knative', 'Integration with IBM Cloud and 3rd party services', 'IBM Developer Kubernetes community', 'View docs', and 'Terms and conditions'. To the right of these links is the 'Overview' section, which includes a brief description of what the service does and a 'Features' section with five bullet points: 'Native Kubernetes Experience', 'Leverage Watson', 'Secure Clusters', 'Self-healing Containers', and 'Logging and Monitoring'. At the bottom left are links for 'Need help?', 'Contact IBM Cloud Sales', 'Estimate Monthly Cost', and 'Cost Calculator'. A large blue 'Create' button is located at the bottom right.

4. For the plan, select **Free** and keep the defaults. Click **Create cluster**, as shown in the following figure.

A free cluster creates only one worker node, which is fine for this exercise, but you should consider a standard cluster for a production environment.

Tiered pricing. Additional charges for bandwidth might apply. [Learn more](#)'. Buttons for 'Create cluster' and 'Add to estimate' are visible."/>

Create a new cluster

Select a plan

**Free**

New to Kubernetes? Create a cluster with 1 worker node to explore the capabilities.

Free

**Standard**

Ready for production? Create a fully customizable cluster with your choice of hardware isolation.

Starting from \$0.11 hourly

Learn more about the differences between Free and Standard clusters in our [docs](#).

Cluster type and version i

Kubernetes

1.14.9 (Latent, Stable, Default)

Cluster name

mycluster

Resource group

Default

Order summary

Free - 2 vCPUs 4GB RAM	
1 worker node	Free
Total*	Free

\*Actual monthly total will vary with [Tiered pricing](#). Additional charges for bandwidth might apply. [Learn more](#).

Create cluster

Add to estimate

Need help? [Contact IBM Cloud](#)

5. Explore the page that opens. You will perform these steps in this exercise.

Clusters / mycluster

 mycluster  Preparing master, workers... Expires in 30 days

Access Overview Worker Nodes Worker Pools Add-ons DevOps **New**

Before your cluster provisions, set up your CLI tools  
Run this command with your PowerShell to download and install a few CLI tools and plugins.

```
Set-ExecutionPolicy Unrestricted; iex(New-Object Net.WebClient).DownloadStr 
```

< >

After your cluster provisions, gain access

1. Log in to your IBM Cloud account. Include the --sso option if using a federated ID.

```
ibmcloud login -a cloud.ibm.com -r us-south -g Default 
```

2. Download the kubeconfig files for your cluster.

```
ibmcloud ks cluster config --cluster bniji8hd0ml53nh47g8g 
```

3. Set the KUBECONFIG environment variable. Copy the output from the previous command and paste it in your terminal. The command output looks similar to the following example:

```
export KUBECONFIG=%HOMEPATH%\bluemix\plugins\container-service\clusters\bniji8hd0ml53nh47g8g\kube-config-hou02-mycluster.yml
```

Alternatively, you can directly [download](#) your kubeconfig files to manually configure the cluster context.

---

6. Open the **Worker Nodes** tab, as shown in the following figure.

You can see the progress in the Worker Nodes tab.

Clusters / mycluster

## mycluster

Preparing master, workers... Expires in 30 days

Web terminal Kubernetes dashboard Connect via CLI

Access Overview Worker Nodes Worker Pools Add-ons DevOps New

**Worker Nodes**

Search Add worker pool +

<input type="checkbox"/>	Name	Status	Worker Pool	Zone	Private IP	Public IP	Version
> <input type="checkbox"/>	0000000a	Provision pending	default	hou02	--	--	1.14.9_1541

Items per page: 10 | 1-1 of 1 items 1 of 1 pages < 1 >

7. Wait until the status becomes Normal, as shown in the following figure. When the status reaches Normal, you can start working with your cluster.



### Note

It might take several minutes (approximately 30 minutes) to deploy the cluster.

Clusters / mycluster

## mycluster

Normal Expires in 30 days

Web terminal Kubernetes dashboard Connect via CLI

Access Overview Worker Nodes Worker Pools Add-ons DevOps New

**Worker Nodes**

Search Add worker pool +

<input type="checkbox"/>	Name	Status	Worker Pool	Zone	Private IP	Public IP	Version
> <input type="checkbox"/>	0000000a	Normal	default	hou02	10.76.152.49	173.193.102.51	1.14.9_1541

Items per page: 10 | 1-1 of 1 items 1 of 1 pages < 1 >

## Part 2: Connecting to your cluster on IBM Cloud Kubernetes Service

You can access your cluster with the IBM Cloud CLI by completing the following steps:

- 1. Log in to IBM Cloud by using the IBM Cloud CLI. Open a command prompt and run the following command:

```
ibmcloud login -a https://cloud.ibm.com -r <your region> -u <your IBM ID  
email> -p <your password> -g <your resource group>
```

Example:

```
ibmcloud login -a https://cloud.ibm.com -r us-south -u myuser@example.com -p  
mypassword -g Default
```



### Note

A resource group is a way for you to organize your account resources in customizable groupings. You have a default resource group in your account that is called **Default** which you can use in this exercise.



### Optional

If you have a federated ID, use `ibmcloud login --sso` to log in to get started.

```
~/Box Sync $ ibmcloud login --sso
API endpoint: https://cloud.ibm.com

Get One Time Code from https://identity-1.ap-north-1.iam.cloud.ibm.com/identity/passcode to proceed.
Open the URL in the default browser? [Y/n]> Y
```

After you press **Y**, log in to IBM Cloud by using your federated ID (IBM ID account in this example). Copy the passcode, and paste it into the command prompt, as shown in the following figure.



The following table shows the Kubernetes Service region and the corresponding IBM Cloud location.

<b>IBM Cloud Kubernetes Service region</b>	<b>Corresponding IBM Cloud location</b>
ap-north (standard clusters only)	Tokyo
ap-south	Sydney
eu-central	Frankfurt
uk-south	London
us-east (standard clusters only)	Washington DC
us-south	Dallas

The following figure shows the output of the `ibmcloud login` command.

```
C:\Users>ibmcloud login -a https://cloud.ibm.com -r us-south -u uni_student02@yahoo.com -p R1iverplate0 -g Default
API endpoint: https://cloud.ibm.com
Authenticating...
OK

Targeted account Uni Student02's Account (982419359bb3ef0787bbeb395b906e1c)
Targeted resource group Default
Targeted region us-south

API endpoint: https://cloud.ibm.com
Region: us-south
User: uni_student02@yahoo.com
Account: Uni Student02's Account (982419359bb3ef0787bbeb395b906e1c)
Resource group: Default
CF API endpoint:
Org:
Space:
```

- 2. Get the command to set the environment variable and download the Kubernetes configuration files and certificates to connect to your cluster by using `kubectl` commands.

```
ibmcloud ks cluster-config --cluster <cluster_name_or_ID>
```

Specify the cluster name that you created in Part 1. For example:

```
ibmcloud ks cluster-config mycluster
```

This command downloads the configuration files that are needed to access your Kubernetes cluster locally from your machine. It outputs an environment variable that is called KUBECONFIG, as shown in the following figure, which you copy and paste into your terminal or command prompt so that the `kubectl` tool can point to your cluster.

```
C:\Users>ibmcloud ks cluster-config mycluster
Kubernetes version 1.16 has removed deprecated APIs. For more information, see <http://ibm.biz/k8s-1-16-apis>
OK
The configuration for mycluster was downloaded successfully.

Export environment variables to start using Kubernetes.

PowerShell
$env:KUBECONFIG = "C:\Users\MarcelaAdan\.bluemix\plugins\container-service\clusters\mycluster\kube-config-hou02-mycluster.yml"

Command Prompt
SET KUBECONFIG=C:\Users\MarcelaAdan\.bluemix\plugins\container-service\clusters\mycluster\kube-config-hou02-mycluster.yml
```

- 
- 3. Set the KUBECONFIG environment variable. Copy the output from the previous command and paste it into your terminal or command prompt. The command output looks similar to the following examples.

## Example

MacOS (Enter the command in a single line)

```
export
KUBECONFIG=/Users/<user_name>/.bluemix/plugins/container-service/clusters/pr_firm_
cluster/kube-config-prod-par02-pr_firm_cluster.yml
```

```
~/Box Sync $ ibmcloud ks cluster-config mycluster
OK
The configuration for mycluster was downloaded successfully.

Export environment variables to start using Kubernetes.

export KUBECONFIG=/Users/junghyeonyoo/.bluemix/plugins/container-service/clusters/mycluster/kube-config-hou02-mycluster.y
ml
```

---

**1+1=2 Example**

Windows (enter the command in a single line).

---

**1+1=2 Example**

```
SET
KUBECONFIG=C:\Users\<user_name>\.bluemix\plugins\container-service\clusters\mycluster\kube-config-mel01-mycluster.yml
```

```
C:\Users\ITSOUSER>ibmcloud ks cluster-config --cluster mycluster
OK
The configuration for mycluster was downloaded successfully.

Export environment variables to start using Kubernetes.

SET KUBECONFIG=C:\Users\ITSOUSER\.bluemix\plugins\container-service\clusters\mycluster\kube-config-hou02-mycluster.yml
```

---

**Note**

For Windows PowerShell users, instead of copying and pasting the `SET` command from the output of `ibmcloud ks cluster-config`, you must set the `KUBECONFIG` environment variable by running, for example, the following command:

```
$env:KUBECONFIG =
"C:\Users\<user_name>\.bluemix\plugins\container-service\clusters\mycluster\kube-config-prod-dal10-mycluster.yml"
```

```
PS C:\Users> ibmcloud ks cluster-config mycluster
OK
The configuration for mycluster was downloaded successfully. Export environment variables to start using Kubernetes.

SET KUBECONFIG=C:\Users\lab21\.bluemix\plugins\container-service\clusters\mycluster\kube-config-mel01-mycluster.yml
PS C:\Users> $env:KUBECONFIG= C:\Users\lab21\.bluemix\plugins\container-service\clusters\mycluster\kube-config-mel01-mycluster.yml
```

---

- 4. Verify that the `kubectl` commands run properly with your cluster by checking the Kubernetes CLI server version:

```
kubectl version --short
```

The `kubectl version --short` command output is shown in the following figure.

```
C:\Users>kubectl version --short
Client Version: v1.7.0
Server Version: v1.14.9+IKS
```

### Part 3: Listing the worker nodes in your cluster

By using `kubectl` commands, you can manage your apps, cluster, and cluster resources. Verify that you can connect to your cluster by listing your worker nodes. Run the following command:

```
kubectl get nodes
```

The `kubectl get nodes` command output is shown in the following figure.

```
C:\Users>kubectl get nodes
NAME           STATUS    AGE     VERSION
10.76.152.49  Ready     1h      v1.14.9+IKS
```

**End of exercise**

## Exercise review and wrap-up

Now that you completed this exercise, you understand what Kubernetes is, how it works, and how you can access the cluster.

You also know how to retrieve cluster information.

---

# Exercise 6. Deploying an application on Kubernetes

## Estimated time

01:30

## Overview

In this exercise, you build a containerized application and deploy it to IBM Cloud Kubernetes Service.

## Objectives

After completing this exercise, you should be able to:

- Create a containerized Node.js application and build it on IBM Cloud Container Registry.
- Explain how the container security analysis capability of Vulnerability Advisor can identify the security vulnerabilities by scanning an image.
- Create a deployment and scale it.
- Expose your application on the internet.

## Introduction

In this exercise, you create configuration files to deploy an application to the IBM Cloud Kubernetes Service.

To make your app more resilient, you can determine the number of instances of the app in your deployment and let Kubernetes automatically create a replica set for you. If one of the pods becomes unresponsive, the pod is re-created automatically.

Also, you expose your app to a port that can be accessed through all your worker nodes public IPs.

## Requirements

- The Kubernetes CLI must be installed.
- The IBM Cloud Dev plug-in must be installed.
- The IBM Cloud Kubernetes Service plug-in must be installed.
- The IBM Cloud Container Registry plug-in must be installed.
- Successfully complete Exercise 5.

## Exercise instructions

In this exercise, you complete the following tasks:

- \_\_\_ 1. Log in to your IBM Cloud account, connect to your cluster, and view the number of worker nodes.
- \_\_\_ 2. Create a container and build it on IBM Cloud Container Registry.
- \_\_\_ 3. Create a deployment and scale it.
- \_\_\_ 4. Expose the app to the internet
- \_\_\_ 5. Clean up the environment

### **Part 1: Logging in to your IBM Cloud account, connecting to your cluster, and viewing the number of worker nodes**

You completed these steps in Exercise 5. If you did not completed Exercise 5 yet, you must complete it before you continue with this exercise.

### **Part 2: Creating a container and building it on IBM Cloud Container Registry**

IBM Cloud offers many starter kits to help you get started with coding quickly. In this tutorial, you complete the steps to create a Node.js application and then build it into IBM Cloud Container Registry.

Also, you can review the Vulnerability Advisor report for details about any vulnerable packages, insecure containers, or app settings.

- \_\_\_ 1. Target a region by running the following command

```
ibmcloud target -r <region>
```



#### Note

A region is a specific geographical location where you can deploy apps, services, and other IBM Cloud resources. IBM Cloud regions differ from IBM Cloud Kubernetes Service regions.

You must choose the same region in which your organization and space are.

You can find where your space is by running the following command:

```
ibmcloud account orgs
```

The following screen shows the output of the command.

```
~/Box Sync $ ibmcloud account orgs
Getting orgs in all regions as darkmong0111@gmail.com...
Retrieving current account...
OK

Name          Region    Account owner      Account ID           Status
darkmong0111@gmail.com  us-south  darkmong0111@gmail.com  12061a956fe34fa8ab89939b905e7f40  active
```

---

**1+1=2 Example**

```
ibmcloud target -r us-south
```

---

- \_\_\_ 2. To target an organization and a space, run the following command:

```
ibmcloud target --cf
```

---

**Note**

Targeting an organization and a space is required for Cloud Foundry services and also to use Cloud App Service starter kits.

You can interactively choose the org and space by running the `ibmcloud target --cf` command. If you have only one space, the org and space are automatically selected by the command.

---

- \_\_\_ 3. To create a starter app from scratch, create a Node.js simple web application by using the Cloud App Service starter kit. The starter application prints a simple text on the browser.

**Note**

IBM Cloud App Service starter kits are pre-configured and integrated.

A starter kit sets up a starter application so that you can get started quickly with creating a cloud native application. Each starter app is generated with configurations to connect to the added services, files for deploying to Cloud Foundry, Kubernetes or a DevOps pipeline. It includes capabilities to monitor the health of your application. They are also generated to reflect the architecture pattern that you choose.

---

Complete the following steps to create your app:

- \_\_\_ a. At the Command Prompt, change to the directory that you want to use for your app. Your app will be saved into this current directory.
- \_\_\_ b. Run the `ibmcloud dev create` command.
- \_\_\_ c. Select 1. Blank App, which creates an application with no predefined architecture pattern. It is just a simple application.
- \_\_\_ d. Select 4. Node to create a Node.js application.

Enter a name for your app. For example, `hello-student`.

- \_\_\_ e. Type `n` to the question “Do you want to select a service to add to this application?”

You can bind IBM Cloud services to the application in this step, but for this exercise, you do not need to do so.

- \_\_\_ f. Select 4. No DevOps, with manual deployment. If you select another option, it creates a DevOps Pipeline to deploy to Cloud Foundry or IBM Cloud Kubernetes Service. In this exercise, you deploy the application manually.

```
Select from the following DevOps toolchain and target runtime environment
options:
```

1. IBM DevOps, deploy to Knative-based Kubernetes containers
2. IBM DevOps, deploy to Helm-based Kubernetes containers
3. IBM DevOps, deploy to Cloud Foundry buildpacks
4. No DevOps, with manual deployment

```
=====
```

? Enter selection number:> 4

- 4. Go to the created application by running `cd hello-student`.
- 5. Edit the `index.js` file in the sample application by opening `/public/index.js`. Go to line 11, and replace `Congratulations!` By `Welcome!` and then save the file.

```
1  <!DOCTYPE html>
2  <html>
3  <head>
4      <title>IBM Cloud Web Starter</title>
5      <style>flex-header,body,main ul{-webkit-box-direction: normal}body{background-color:#fff;font-family:IBMPlexSans;display:-webkit-
6      <meta name="viewport" content="width=device-width, initial-scale=1">
7  </head>
8  <body>
9  <header id="flex-header">
10     <div class="cloud-header"></div>
11     <h1>Welcome!</h1>
12     <h2>You are currently running a Node.js app built for the IBM Cloud.</h2>
13  </header>
14  <main>
15      <ul>
16          <li>
17              <div class="right-arrow"></div>
18              <div>
19                  <a target="_blank" rel="noopener" href="https://cloud.ibm.com/developer/appservice/dashboard?env_id=ibm3Ayp3Aus-"
20              </div>
21          </li>
22          <li>
23              <div class="right-arrow"></div>
24              <div>
25                  <a target="_blank" rel="noopener" href="https://slack-invite-ibm-cloud-tech.mybluemix.net/">Ask questions on Slack
26              </div>
27          </li>
28          <li>
29              <div class="right-arrow"></div>
30              <div>
31                  <a target="_blank" rel="noopener" href="https://www.ibm.com/cloud/cli">Install IBM Cloud Developer Tools</a>
32              </div>
33          </li>
34      </ul>
35  </main>
36  </body>
37  </html>
```

- 6. Create a namespace in the IBM Cloud Container Registry by running the following command:

```
ibmcloud cr namespace-add <namespace name>
```

### Example

```
ibmcloud cr namespace-add cr-studentxx
```

The name space in the registry must be unique. Replace “xx” in the example by your initials or a unique identifier.

7. Build an image by using the Dockerfile in the IBM Cloud Container Registry and give it a tag.  
A Dockerfile is a text document that contains all the commands that a user might call from the command line to assemble an image. Run the following command:

```
ibmcloud cr build -t <region domain>/<namespace>/<application name>:<tag>  
<directory>
```

**1+1=2 Example**

```
ibmcloud cr build -t us.icr.io/cr-studentxx/hello-student:v1 .
```

Do not miss “.” at the end. It indicates that the current directory is the location of your build context, which contains your Dockerfile and prerequisite files.

The following table shows the supported IBM Cloud Container Registry local regions.

IBM Cloud Container Registry region	Domain name
ap-north	jp.icr.io
ap-south	au.icr.io
eu-central	de.icr.io
uk-south	uk.icr.io
us-south	us.icr.io



## Information

The Dockerfile must be defined to build an image. When you create the sample application by running the `ibmcloud dev create` command, a Dockerfile is created automatically. Look at the specifics of your Dockerfile, which is shown in the following figure.

```

1  FROM node:8-stretch
2
3  # Change working directory
4  WORKDIR "/app"
5
6  # Update packages and install dependency packages for services
7  RUN apt-get update \
8      && apt-get dist-upgrade -y \
9      && apt-get clean \
10     && echo 'Finished installing dependencies'
11
12 # Install npm production packages
13 COPY package.json /app/
14 RUN cd /app; npm install --production
15
16 COPY . /app
17
18 ENV NODE_ENV production
19 ENV PORT 3000
20
21 EXPOSE 3000
22
23 CMD ["npm", "start"]

```

- Line 1 `FROM node:8-stretch`: This line determines the base image on which you base your image. In this example, you are creating an image from the Node.js 8-stretch version of the image.
- Line 3 `#Change working directory`: This line is a comment. You can add comments to the Dockerfile by adding `#` at the beginning of the line.
- Line 4 `WORKDIR "/app"`: This line specifies where to place the application code inside the image, which becomes the working directory for your application.
- Lines 7 – 10: The `RUN` command installs your application and its required packages. In this example, it updates the OS image and prints the `'Finished installing dependencies'` message.
- Line 13: The `COPY` command copies the files or directories from the host to the Docker image.
- Line 14: This `RUN` command installs an NPM package of the application inside the image.

- Line 16: The `COPY` command copies the files from the local source to the image. In this example, the files in the current directory are copied to the location `/app` within the image.
- Lines 18 – 19: The `ENV` instruction sets the environment variable and the value is in the environment of all “descendant” Dockerfile commands.
- Line 21: The `EXPOSE` instruction is used to specify the network port on which the Docker container listens at run time.

The `EXPOSE` instruction does not publish the port. It functions as a type of documentation between the person who builds the image and the person who runs the container about which ports are intended to be published. To publish the port when running the container, use the `-p` flag on Docker to publish and map one or more ports, or the `-P` flag to publish all available ports and map them to high-order ports.

- Line 23: You can use `CMD` to specify the command to run when the Docker container starts. In this case, “`npm start`” runs when the container starts.

8. Verify that the image is created. The Vulnerability Advisor is automatically run and indicates the issues with the image as shown in the SECURITY STATUS column.

Run the following command:

```
ibmcloud cr images
```

REPOSITORY	TAG	DIGEST	NAMESPACE	CREATED	SIZE	SECURITY STATUS
us.icr.io/cr-student/hello-student	v1	5330d20f987e	cr-student	26 minutes ago	431 MB	3 Issues
<b>OK</b>						



## Troubleshooting

If you see issues in the SECURITY STATUS column, to learn more details about these issues, run the following command:

```
ibmcloud cr va <region domain>/<namespace>/<application name>:<tag>
```

Example: `ibmcloud cr va us.icr.io/cr-studentxx/hello-student:v1`

```

~/Box Sync/data_2019/RESIDENCY/Ex6/hello-student $ ibmcloud cr vd us.icr.io/cr-student/hello-student:v1
Checking security issues for 'us.icr.io/cr-student/hello-student:v1'...
Image 'us.icr.io/cr-student/hello-student:v1' was last scanned on Sat May 4 05:57:47 UTC 2019
The scan results show that 3 ISSUES were found for the image.

Configuration Issues Found

Configuration Issue ID          Policy Status   Security Practice                                         How to Resolve
application_configuration:mysql.ssl-ca Active          A setting in /etc/mysql/my.cnf that specifies the ssl-ca is not specified in /etc/mysql/my.cnf
                                               
application_configuration:mysql.ssl-cert-nf Active          Certificate Authority (CA) certificate. ssl-cert is not specified in /etc/mysql/my.c
                                               
application_configuration:mysql.ssl-key-f Active          A setting in /etc/mysql/my.cnf that identifies the server public key certificate. This certificate file.
                                               
                                               
                                               
server private key. ssl-key is not specified in /etc/mysql/my.cnf

OK

```

In this case, all issues are related to SSL. You can ignore these issues in this exercise.

Vulnerability Advisor checks the security status of container images.

When you add an image to a namespace, the image is automatically scanned by Vulnerability Advisor to detect security issues and potential vulnerabilities. If any security issues are found, instructions are provided to help fix the reported vulnerability.

### **Part 3: Creating a deployment and scaling it**

Deployments are used to manage pods, which include containerized instances of an app. You can run an application by creating a Kubernetes deployment, and you can describe a deployment in a YAML file.

When you create a deployment, a Kubernetes pod is created for each container that you defined in the deployment.

Complete the following steps:

- 1. Create a folder that is named **yaml** to store the configuration files by running the following commands:

```

mkdir yaml
cd yaml

```

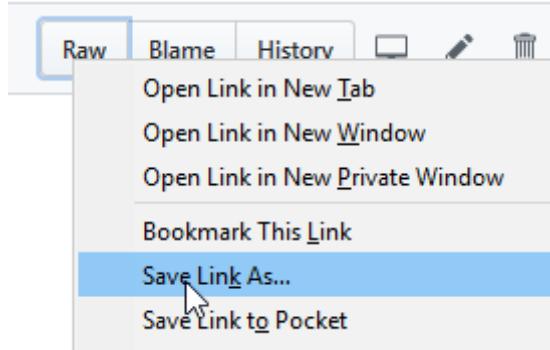
- 2. Create a deployment file (**01hello-student-deployment.yaml**). Download the file from the following link and replace the image tag by the one that you created in Part 2:

<https://github.com/IBM-SkillsAcademy/Cloud-Application-Developer/blob/master/CloudAppDev/Ex6/01hello-student-deployment.yaml>



### Hint

To download the file from the Git repo, right-click **Raw** and select **Save Link As** as shown in the following figure.



Save the file in the **yaml** directory that you previously created.

The following code snippet shows the `01hello-student-deployment.yaml` file in this example. A pod with the following specifications is deployed:

- Your deployment name is `hello-student-deployment`.
- When you bind the deployment to the service, you use a selector. Selectors are key value pairs. In this exercise, the selector is `app : hello-student`.
- The pod has one replica with `100Mi` memory and `125m` CPU.
- The application is exposed on port 3000.

- The target image is “us.icr.io/cr-student/hello-student:v1”.

```

apiVersion: apps/v1
kind: Deployment
metadata:
  name: hello-student-deployment
spec:
  replicas: 1
  selector:
    matchLabels:
      app: hello-student
  template:
    metadata:
      labels:
        app: hello-student
    spec:
      containers:
        - name: hello-student
          image: us.icr.io/cr-student/hello-student:v1
          resources:
            limits:
              memory: "100Mi"
              cpu: "125m"
            requests:
              memory: "100Mi"
              cpu: "125m"
          ports:
            - containerPort: 3000

```

   3. Modify some of the properties of 01hello-student-deployment.yaml:

- a. Increase the upper limits of the memory and CPU to allow vertical scalability of the containers inside the pods. Under limits, change the memory to 200Mi and the CPU to 250m.
- b. Change image to the image tag that you created in Part 2.
- c. To set up high availability for your deployments, add two more replicas, which enable horizontal scalability. Change replicas: 1 to replicas: 3.

Your deployment file now looks like the following example.

```

apiVersion: apps/v1
kind: Deployment
metadata:
  name: hello-student-deployment
spec:
  replicas: 3
  selector:
    matchLabels:
      app: hello-student
  template:
    metadata:
      labels:
        app: hello-student
    spec:
      containers:
        - name: hello-student
          image: us.icr.io/cr-studentxx/hello-student:v1
      resources:
        limits:
          memory: "200Mi"
          cpu: "250m"
        requests:
          memory: "100Mi"
          cpu: "125m"
      ports:
        - containerPort: 3000

```

- \_\_\_ 4. Apply the deployment to your Kubernetes cluster by running the following command:

```
kubectl apply -f 01hello-student-deployment.yaml
```

- \_\_\_ 5. Retrieve all the deployments in your cluster and observe the number of replicas by running the following command:

```
kubectl get deployment
```

The following figure shows the output.

~/Box Sync/data_2019/RESIDENCY/Ex6/hello-student/yaml \$ kubectl get deployment						
NAME	DESIRED	CURRENT	UP-TO-DATE	AVAILABLE	AGE	
hello-student-deployment	3	3	3	3	5m	

- \_\_\_ 6. Retrieve all the pods in your cluster by running the following command. Your app is deployed three times in each pod.

```
kubectl get pods
```

The following figure shows the output.

NAME	READY	STATUS	RESTARTS	AGE
hello-student-deployment-7d799d48c5-7l6xl	1/1	Running	0	7m
hello-student-deployment-7d799d48c5-l8sqc	1/1	Running	0	7m
hello-student-deployment-7d799d48c5-xsf9v	1/1	Running	0	7m

- 7. Try Kubernetes self-healing by deleting all the pods. After the pods are deleted, Kubernetes self-heals by always ensuring that three replicas are deployed.
- a. Run the `kubectl delete pods --all` command to delete all the pods. The following figure shows the output.

```
~/Box Sync/data_2019/RESIDENCY/Ex6/hello-student/yaml $ kubectl delete pods --all
pod "hello-student-deployment-7d799d48c5-7l6xl" deleted
pod "hello-student-deployment-7d799d48c5-l8sqc" deleted
pod "hello-student-deployment-7d799d48c5-xsf9v" deleted
```

- b. To retrieve all pods in your cluster again, run the following command:

```
kubectl get pods
```

As shown in the following figure, you can see that the statuses of some containers are Terminating and some are ContainerCreating.

NAME	READY	STATUS	RESTARTS	AGE
hello-student-deployment-7d799d48c5-24h9s	1/1	Terminating	0	47s
hello-student-deployment-7d799d48c5-4zn2r	1/1	Terminating	0	46s
hello-student-deployment-7d799d48c5-9d9x9	1/1	Terminating	0	46s
hello-student-deployment-7d799d48c5-chrdr	0/1	ContainerCreating	0	4s
hello-student-deployment-7d799d48c5-hpcf9	0/1	ContainerCreating	0	4s
hello-student-deployment-7d799d48c5-wzppr	0/1	Pending	0	3s



## Troubleshooting

If you cannot see the ContainerCreating status, it is because containers are healing before you can run the `kubectl get pods` command. Delete again and check the pods' statuses faster.

After a few seconds, run the `kubectl get pods` command again. The following figure shows that three pods are running as deployed.

NAME	READY	STATUS	RESTARTS	AGE
hello-student-deployment-7d799d48c5-chrdr	1/1	Running	0	4m
hello-student-deployment-7d799d48c5-hpcf9	1/1	Running	0	4m
hello-student-deployment-7d799d48c5-wzppr	1/1	Running	0	4m

**Note**

After the pods are terminated for any reason, Kubernetes automatically self-heals by creating pods.

- 8. Scale in and out the deployment by changing the number of replicas:
  - a. Find the deployment name by running the `kubectl get deployment` command. The following figure shows the output.

```
~/Box Sync/data_2019/RESIDENCY/Ex6/hello-student/yaml $ kubectl get deployment
NAME          DESIRED   CURRENT   UP-TO-DATE   AVAILABLE   AGE
hello-student-deployment   3         3         3            3           5m
```

- b. Run the following command:

```
kubectl scale deployment <deployment name> --replicas=4
```

**Example**

```
kubectl scale deployment hello-student-deployment --replicas=4
```

- c. Check the status of pods by running the following command. The following figure shows that there are four pods.

```
kubectl get pods
```

```
~/Box Sync/data_2019/RESIDENCY/Ex6/hello-student/yaml $ kubectl get pods
NAME          READY   STATUS    RESTARTS   AGE
hello-student-deployment-7d799d48c5-chrdr  1/1     Running   0          25m
hello-student-deployment-7d799d48c5-ghpzw  1/1     Running   0          4s
hello-student-deployment-7d799d48c5-hpcf9  1/1     Running   0          25m
hello-student-deployment-7d799d48c5-wzppr  1/1     Running   0          25m
```

- 9. Run a command to create a Horizontal Pod Autoscaler that maintains between 4 and 10 replicas of the Pods to achieve an average CPU utilization across all Pods of 80%.

```
kubectl autoscale deployment <deployment name> --cpu-percent=80 --min=4 --max=10
```

**Example**

```
kubectl autoscale deployment hello-student-deployment --cpu-percent=80 --min=4 --max=10
```

## **Part 4: Exposing the app over the internet**

Kubernetes supports four basic types of network services: ClusterIP, NodePort, LoadBalancer, and Ingress. ClusterIP services make your apps accessible internally to allow communication among

pods in your cluster only. The NodePort, LoadBalancer, and Ingress services make your apps externally accessible from the public internet or a private network.

In this example, you expose the service by using NodePort.



### Note

When you expose apps with the NodePort service, a NodePort of 30000 - 32767 and an internal cluster IP address is assigned to the service. To access the service from outside the cluster, you use the public or private IP address of any worker node and the NodePort in the format <IP\_address>:<nodeport>.

Complete the following steps:

1. Download the `02hello-student-service.yaml` file from the Git repo at the following link and save it in the **yaml** directory:

<https://github.com/IBM-SkillsAcademy/Cloud-Application-Developer/blob/master/CloudAppDev/Ex6/02hello-student-service.yaml>

A service with the following specifications is deployed, as shown in the following example:

- The service name is `hello-student-service`.
- The service is bound to the pods that have the label `app: x`.
- The application is exposed on port 3000.

```
apiVersion: v1
kind: Service
metadata:
  name: hello-student-service
spec:
  selector:
    app: x
  ports:
  - name: http
    protocol: TCP
    port: 3000
    type: NodePort
```

- 2. Update `app: x` to `app: hello-student` to match the selector that is defined in the deployment. The final deployment file looks like the following example.

```
apiVersion: v1
kind: Service
metadata:
  name: hello-student-service
spec:
  selector:
    app: hello-student
  ports:
  - name: http
    protocol: TCP
    port: 3000
    type: NodePort
```

- 3. Apply the deployment file by running the following command:

```
kubectl apply -f 02hello-student-service.yaml
```

- 4. Retrieve the details about your services by running the following command:

```
kubectl get services
```

The following figure shows the output.

~/Box Sync/data_2019/RESIDENCY/Ex6/hello-student \$ kubectl get services						
NAME	TYPE	CLUSTER-IP	EXTERNAL-IP	PORT(S)	AGE	
hello-student-service	NodePort	172.21.200.155	<none>	3000:30456/TCP	1m	
kubernetes	ClusterIP	172.21.0.1	<none>	443/TCP	18d	

Your service is exposed on a specific port number, which you can find in the `PORT(S)` column. In the previous screen, it is 30456, which means any requests to your worker nodes on this port are directed to your application.

- 5. Retrieve your worker node public IP by running the following command:

```
ibmcloud ks workers mycluster
```

The following figure shows the output.

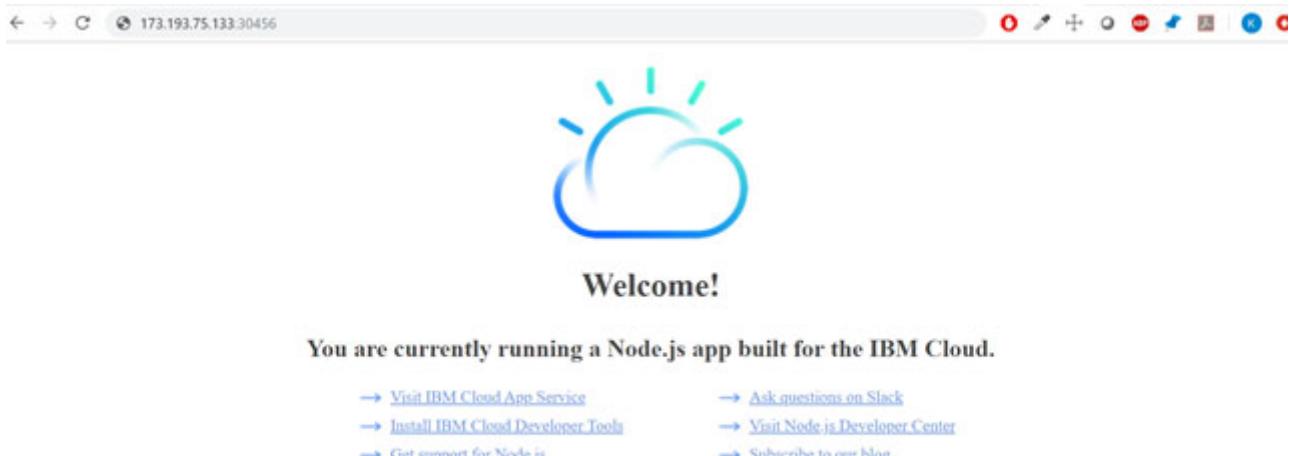
~/Box Sync/data_2019/RESIDENCY/Ex6/hello-student.yaml \$ ibmcloud ks workers mycluster								
OK	ID	Public IP	Private IP	Machine Type	State	Status	Zone	Version
OK	kube-hou02-paf7cb9bfddde7c455680e17077ce0231a3-w1	173.193.75.133	10.44.12.8	free	normal	Ready	hou02	1.12.7_1548*

- 6. Access your application on the browser through the public IP of your worker node that you retrieved in Step 5 and the port that you retrieved in Step 4:

```
<Public IP>:<Port>
```

### 1+1=2 Example

173.193.75.133:30456



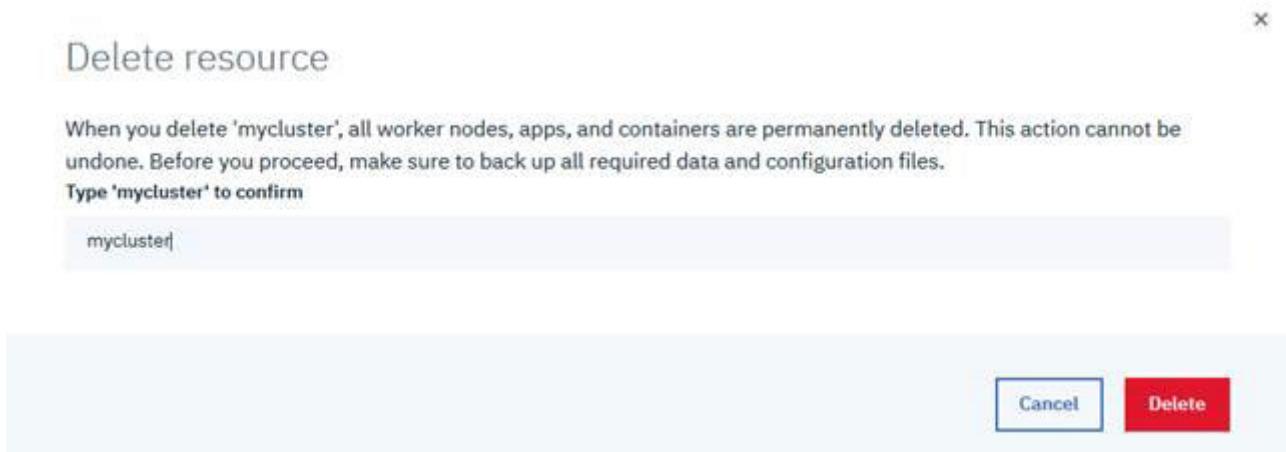
## Part 5: Cleaning up the environment

In this part, you delete the Kubernetes cluster that you created on IBM Cloud.

- \_\_\_ 1. Open IBM Cloud dashboard.
- \_\_\_ 2. Expand **Kubernetes Clusters** and click the three dots next to the cluster name as shown in the figure

Name	Group	Location	Status	Tags
<input type="text"/> Filter by name or IP address...	<input type="text"/> Filter by group or org...	<input type="text"/> Filter...	<input type="text"/> Filter...	<input type="text"/> Filter...
> Devices (0)				
▽ Kubernetes Clusters (1)				
mycluster	Default	Dallas	Normal	...
> Cloud Foundry Apps (0)				
> Cloud Foundry Services (0)				
> Services (0)				
> Storage (0)				
> Cloud Foundry Enterprise Environments (0)				
> Apps (7)				

- \_\_\_ 3. Click **Delete**.
- \_\_\_ 4. At the Delete resource window, enter the name of your cluster to confirm that you want to delete it, and click **Delete** as shown in the figure.



- \_\_\_ 5. Delete all namespaces from IBM Cloud registry by accessing this URL  
<https://cloud.ibm.com/kubernetes/registry/main/start>
- \_\_\_ 6. Click **Namespaces**. Select your namespace, click the three dots to the right and select **Delete namespace**, as shown in the following figure.

The screenshot shows the IBM Cloud Registry interface. On the left, a sidebar has "Kubernetes" selected. The main area has "Registry" selected under "LOCATION Dallas". Below this are sections for "Quick Start", "Contents", and "Settings". The "Namespaces" section shows 1 namespace (cr-studentbox). The "Repositories" section shows 1 repository. The "Images" section shows 1 image. A "Settings" section has a gear icon. Below these, a "Namespaces" table lists "cr-studentbox" with 1 repository and 1 image. A "Delete namespace" dialog is open over the table, showing "1 item selected" and a "Delete namespace" button at the bottom right.

Name	Repository Count	Image Count
cr-studentbox	1	1

**End of exercise**

## Exercise review and wrap-up

In this exercise, you created a Node.js application by using IBM Cloud App Service starter kits and deployed the application to the IBM Cloud Kubernetes Service with three replicas to ensure high availability.

Also, you learned how self-healing works, and you exposed your application to the public by using the NodePort service.

## Troubleshooting

This section provides some basic troubleshooting techniques to fix errors that you may encounter while you perform this exercise.

### Storage quota exceeded

If you encounter the error that is shown in the following figure, delete one or more images.

```
[18495ddff4: Preparing
[182b00c71e: Preparing
[185b1a8b49: Preparing
[183ea5415f: Preparing
[1890f129c3: Preparing
[1877893bad: Preparing
[186aadc86d: Preparing
[182fe51680: Preparing
[18c4b192e8: Preparing
[18c551a0da: Preparing
[1831157b81: Preparing
[18d801397d: Preparing
[1868cde90b: Preparing
[18FAILED9a: Waiting g
Status: denied: You have exceeded your storage quota. Delete one or more images, or review your storage quota and pricing plan. For more information, see https://ibm.biz/BdjFwL, Code: 1
```

- 1. List the available images by running this command:

```
ibmcloud cr image-list
```

```
C:\Users\NorhanKhaled\Downloads\hello-student-772019>ibmcloud cr image-list
Listing images...

REPOSITORY                      TAG    DIGEST      NAMESPACE     CREATED      SIZE    SECURITY STATUS
us.icr.io/cr-student/hello-student   v1    b658d76c2f50  cr-student   46 minutes ago  422 MB  3 Issues
us.icr.io/cr-student12/hello-student  v1    7dc6570b1c6c  cr-student12  1 month ago   416 MB  4 Issues

OK
```

- 2. Remove the image by running this command:

```
ibmcloud cr image-rm <REPOSITORY>:<TAG>
```

```
C:\Users\NorhanKhaled\Downloads\hello-student-772019>ibmcloud cr image-rm us.icr.io/cr-student/hello-student:v1
Deleting image 'us.icr.io/cr-student/hello-student:v1' ...

Successfully deleted image 'sha256:b658d76c2f50f895d600f6e73dabc4b023039328808fdd5449912657332e0f30'

OK
```

### Requested namespace already exists

If you encounter the error that is shown in the following figure, remove the existing namespace or create a new namespace with a unique name.

```
C:\Users\NorhanKhaled\Downloads\hello-student-772019\yaml>ibmcloud cr namespace-add cr-student
Adding namespace 'cr-student'...
The requested namespace is already owned by your account.

OK
```

To list all the namespaces own by your account, run the command

```
ibmcloud cr namespace-list
```

```
C:\Users\NorhanKhaled\Downloads\hello-student-772019\yaml>ibmcloud cr namespace-list
Listing namespaces for account 'Student91's Account' in registry 'us.icr.io'...

Namespace
cr-student
cr-student12
cr-studentaa

OK
```

To delete a namespace, run the command:

```
ibmcloud cr namespace-rm <namespace>
```

Enter **y** for *Deleting the namespace will also delete all images and trust information in the namespace. [y/N]>*, as shown in the following figure.

```
C:\Users\NorhanKhaled\Downloads\hello-student-772019\yaml> ibmcloud cr namespace-rm cr-student
Are you sure you want to delete namespace 'cr-student'? Deleting the namespace will also delete all images and trust information in the namespace. [y/N]> y
Deleting namespace 'cr-student'...

Successfully deleted namespace 'cr-student'

OK
```

## Error validating data with kubectl apply

If you receive the error that is shown in the following figure when you run the `kubectl apply` command, update the `kubectl` version in your workstation to the latest stable version by following the steps in Exercise 0, section **Installing the Kubernetes CLI**.

```
error: error validating "02hello-student-service.yaml": error validating data: the server could not find the requested resource; if you choose to ignore these errors, turn validation off with --validate=false
```

## Accessing the Kubernetes dashboard

You can view, create, and delete namespaces, clusters, repositories and images through the Kubernetes dashboard at <https://cloud.ibm.com/kubernetes/overview> as shown in the following figure.

The screenshot shows the IBM Cloud interface with the 'Kubernetes' service selected. The main content area features a large blue button labeled 'Create a cluster'. Above the button, the text reads: 'Deploy, scale, and manage your containerized application workloads' and 'Deploy Docker images using the native Kubernetes experience.' A blue cube icon is positioned next to the text. The bottom right corner of the dashboard has a blue circular icon with a white speech bubble symbol.

For more information see [Troubleshooting clusters](#) at  
[https://cloud.ibm.com/docs/containers?topic=containers-cs\\_troubleshoot](https://cloud.ibm.com/docs/containers?topic=containers-cs_troubleshoot)



IBM Training



© Copyright International Business Machines Corporation 2016, 2019.