# 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.

## *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.

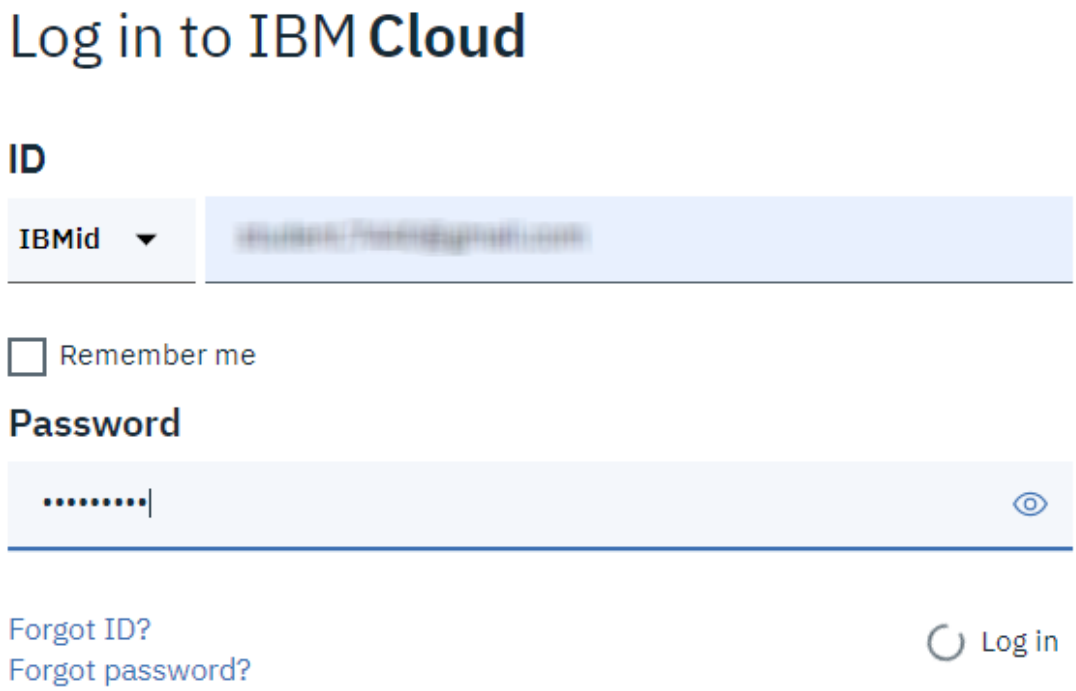
__ 2. In the “Log in to IBM Cloud” right pane, complete the following steps:

__ a. Leave the “IBMid” selected from the drop-down list.

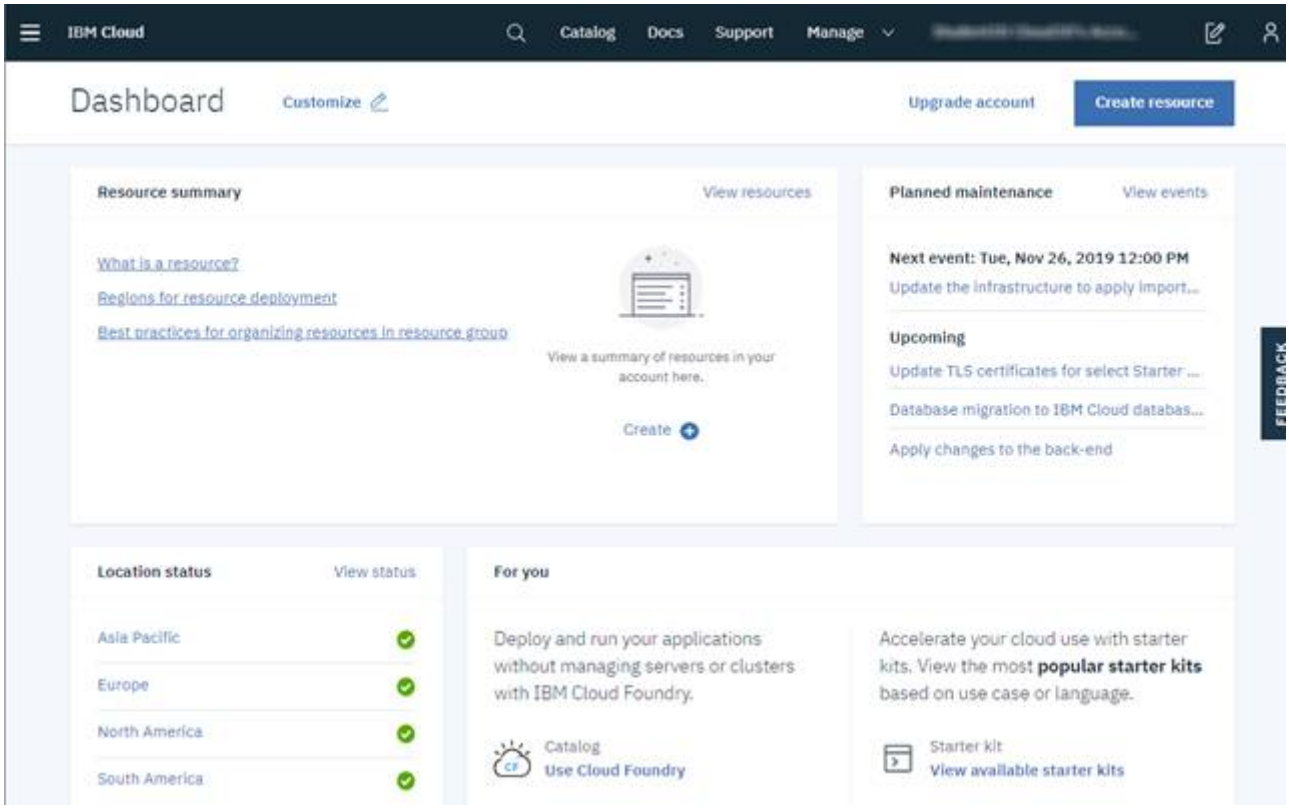
__ b. Enter the email address that you used to register for IBM Cloud,

__ c. Click **Continue.**

__ d. Enter your password and click **Log in**.




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

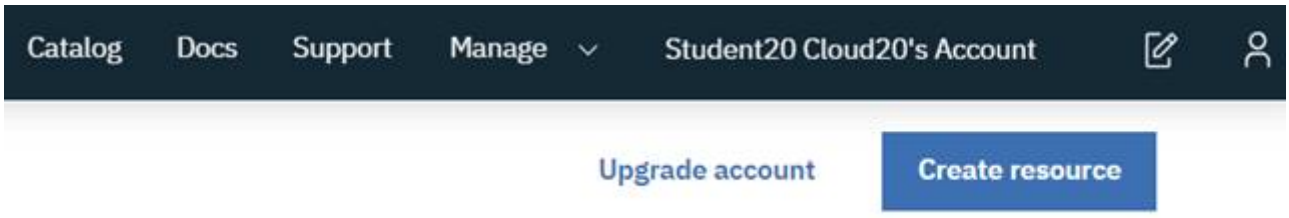



## *Part 3: Creating an application*

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:

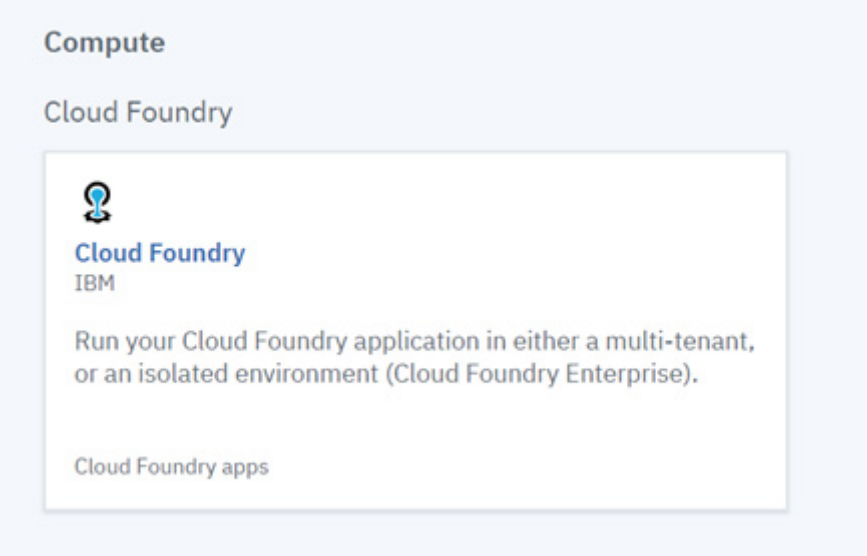
__ 1. Create an application with an instance of the IBM SDK for Node.js runtime environment:

__ a. On the IBM Cloud Dashboard, click **Create resource** on the right, as shown in the following figure.

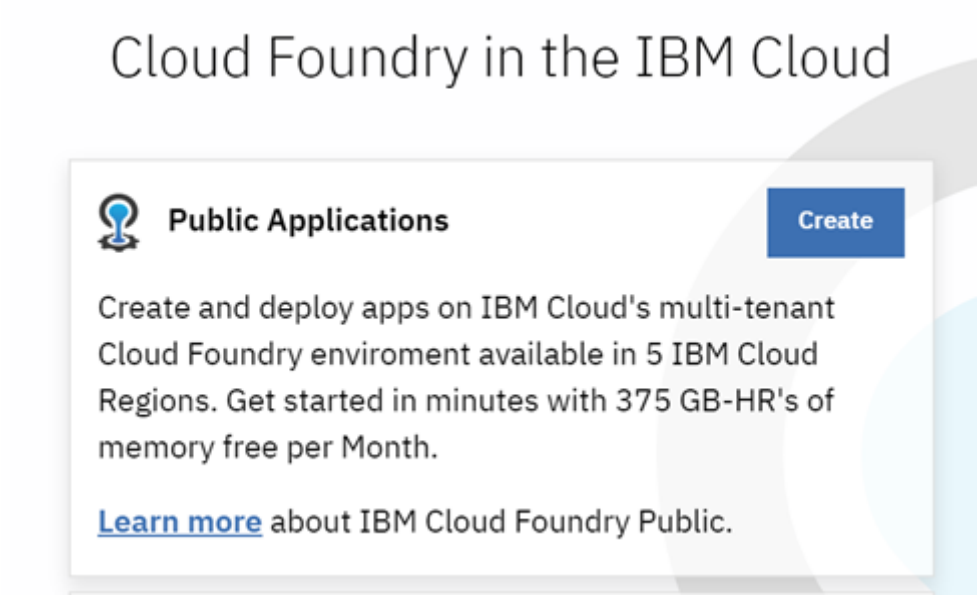



__ b. You can now see the entire catalog. In the Search field, type “Cloud Foundry”.

__ 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.



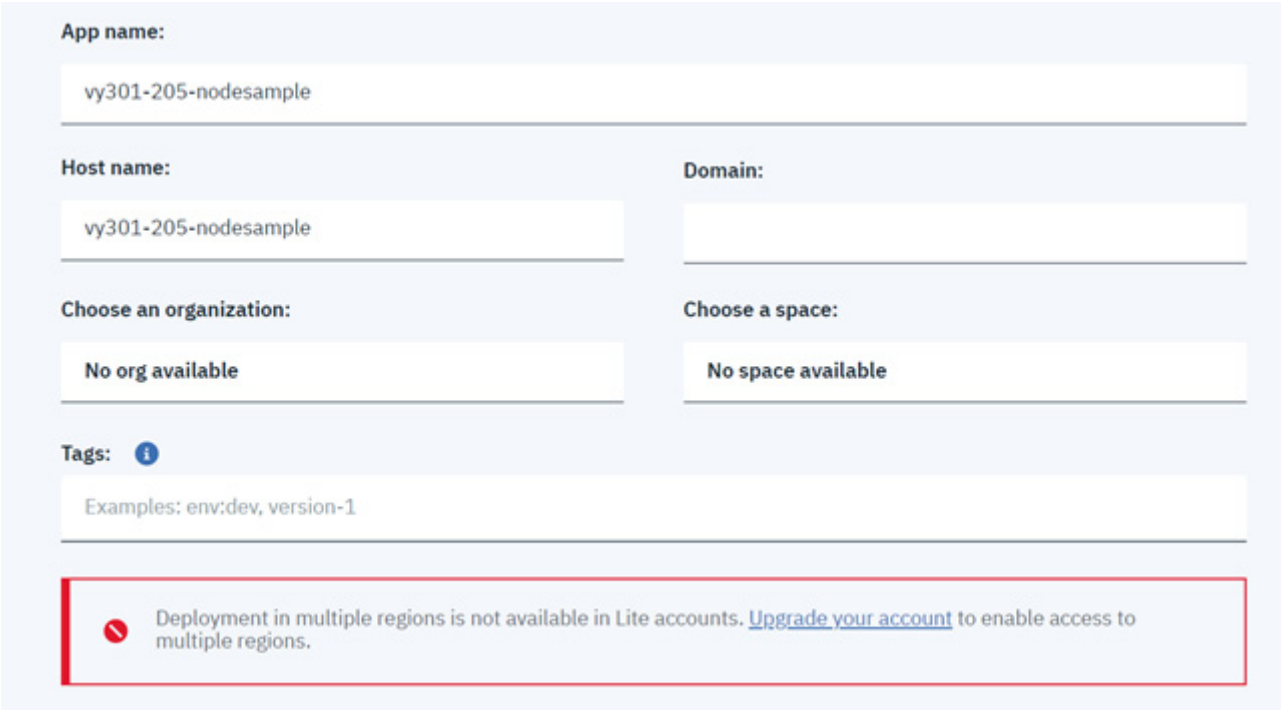

__ 3. Complete the details for creating a Cloud Foundry Sample App:

__ a. Region is selected by default based on your location.

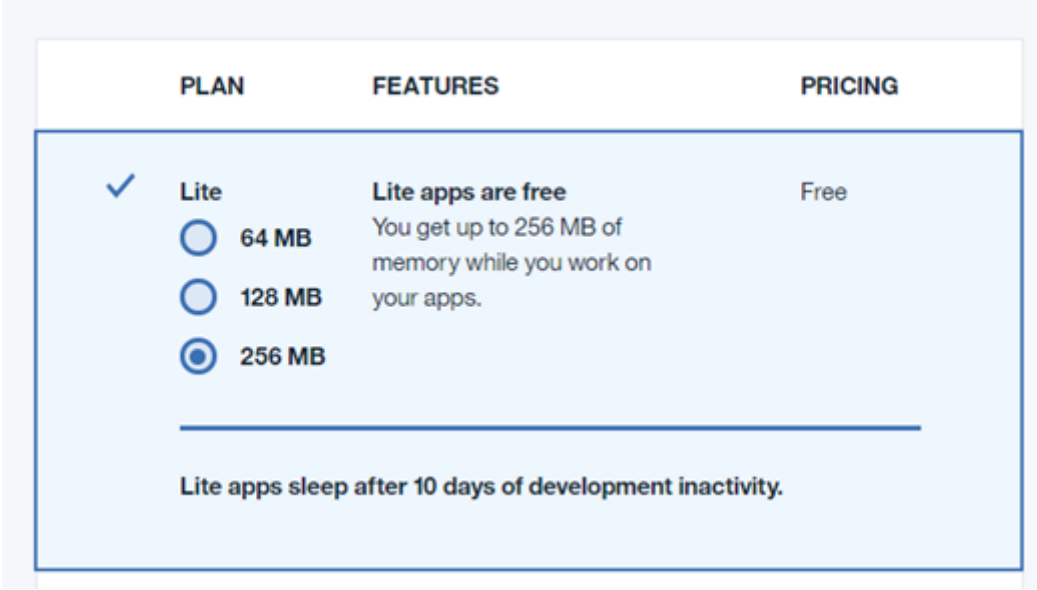
In this example, the region is London.


**Troubleshooting**

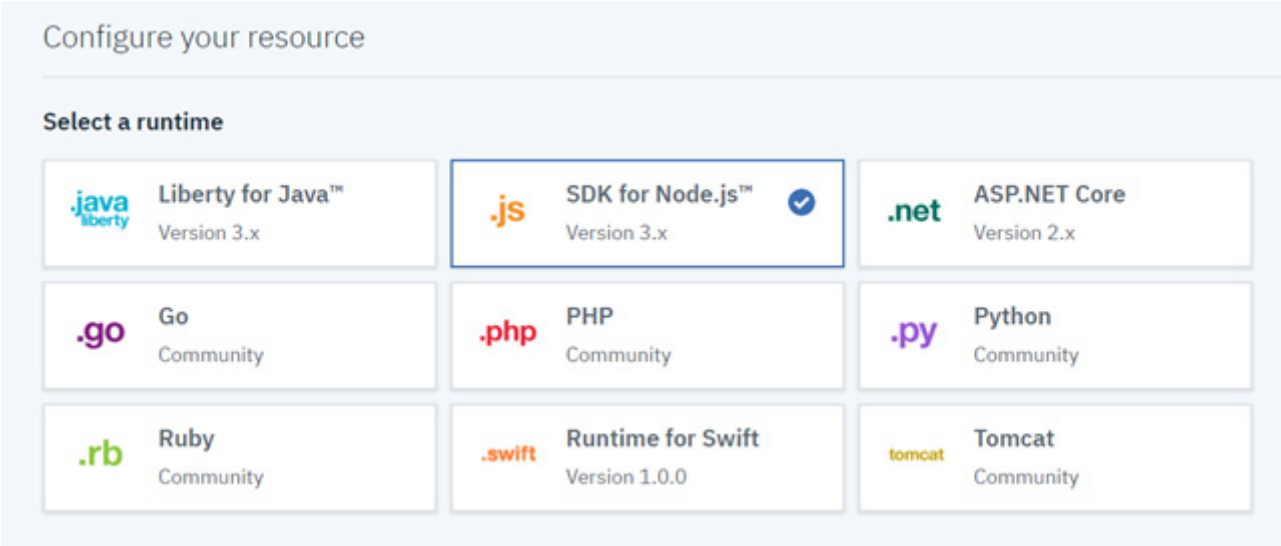
If you see the error *"Deployment in multiple regions is not available in Lite accounts, Upgrade your account to enable access to multiple regions."*, as shown in the following figure, click the drop-down menu under **Select a region** and switch to the region to which you have access. Repeat the step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




__ 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.




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




__ 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**

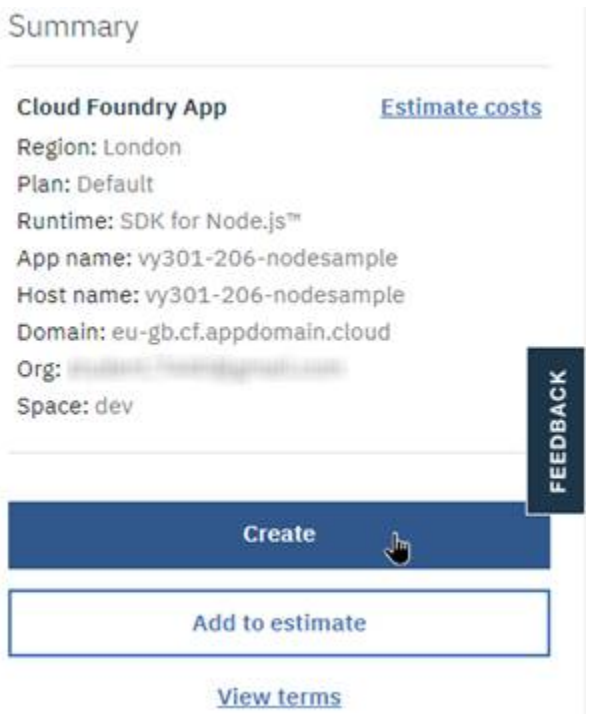
Ensure that the selected domain has the format `{region}.cf.appdomain.cloud`. Make sure **not** to select "*mybluemix.net*" as a domain because it is deprecated.

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.

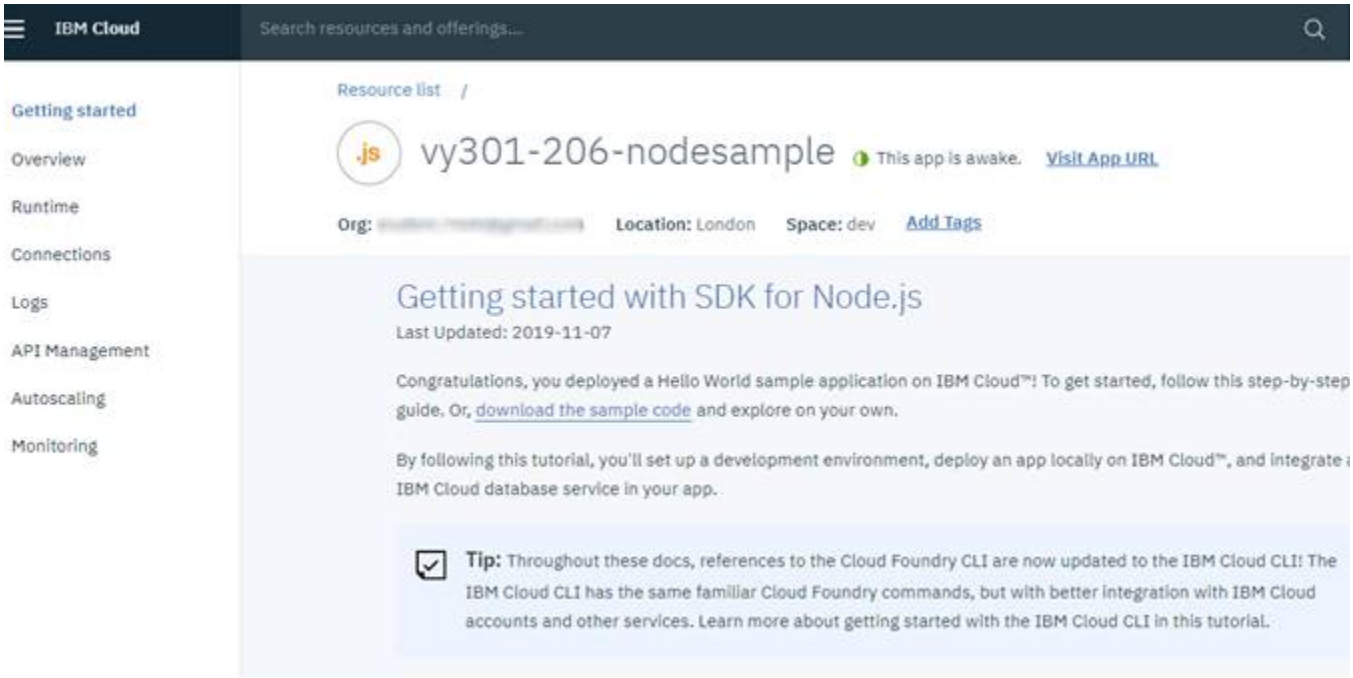

__ i. Click **Create**.



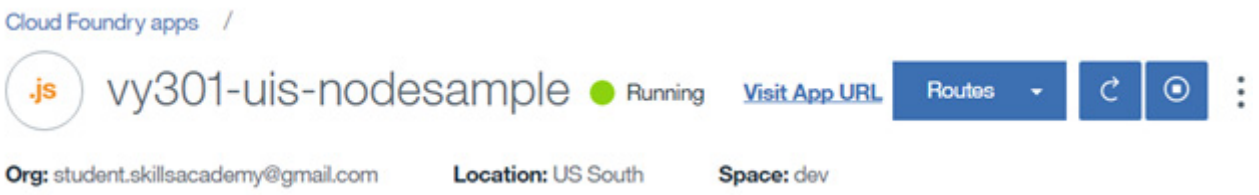
IBM Cloud proceeds to deploy your application. Your application stages and deploys in a few minutes.

**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.

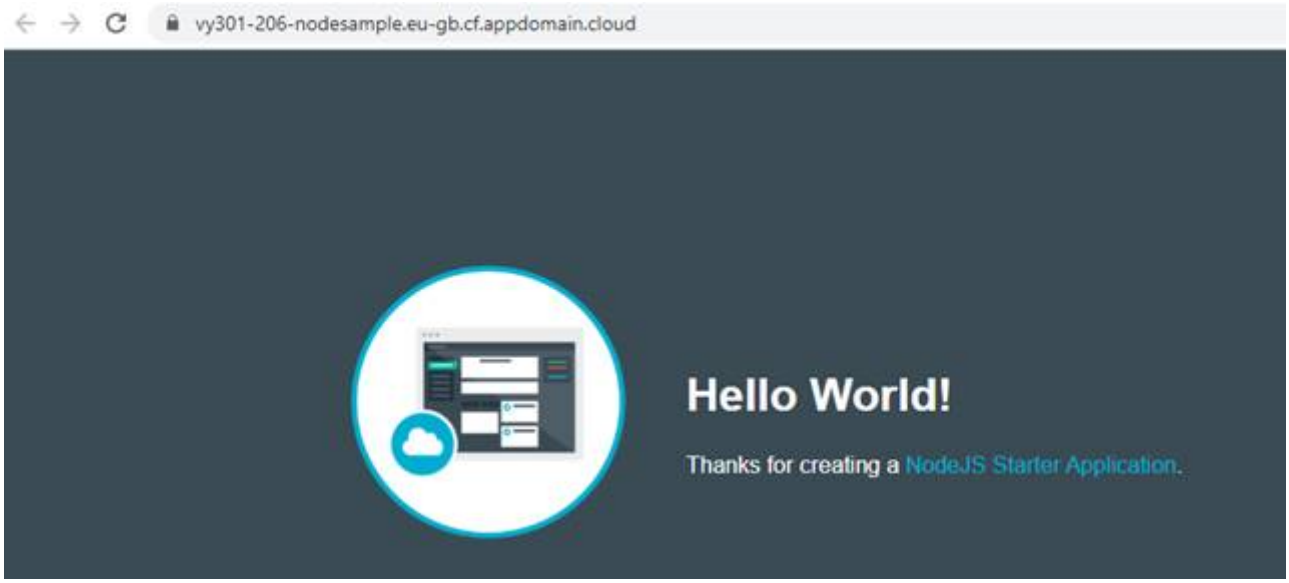



If you do not have a Lite account, the application status should be “Running”, as shown in the following figure.



__ 4. Click **Visit App URL**, which opens a new browser tab with the URL for your app **`https://vy301-xxx-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*

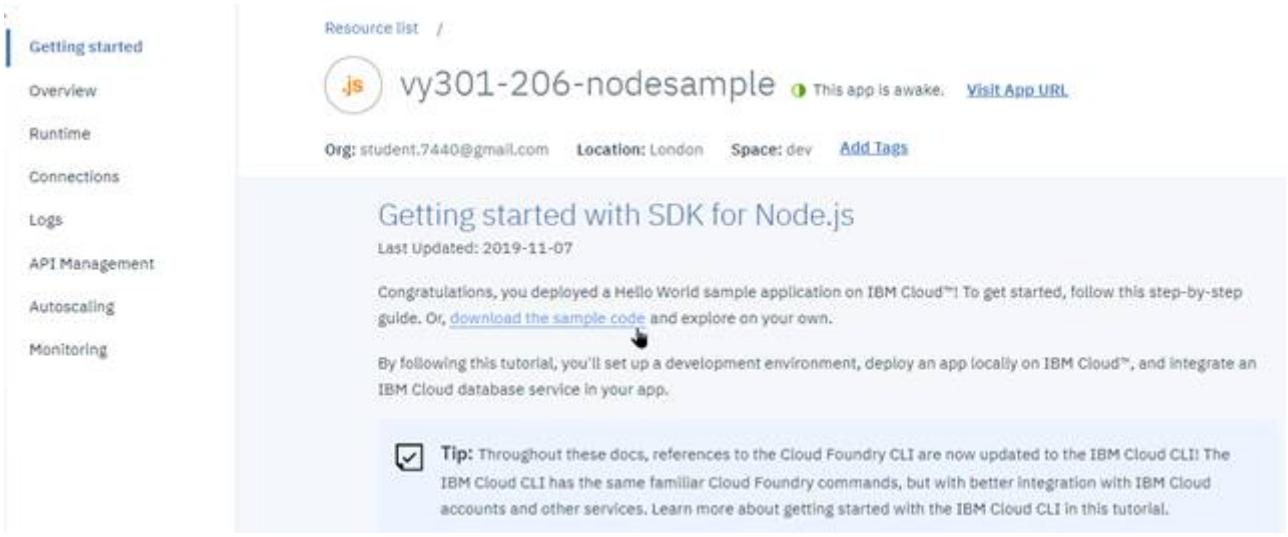
Now that you successfully deployed your application, you modify the code and push the changes out to Cloud Foundry with IBM Cloud CLI by completing the following steps:

__ 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.

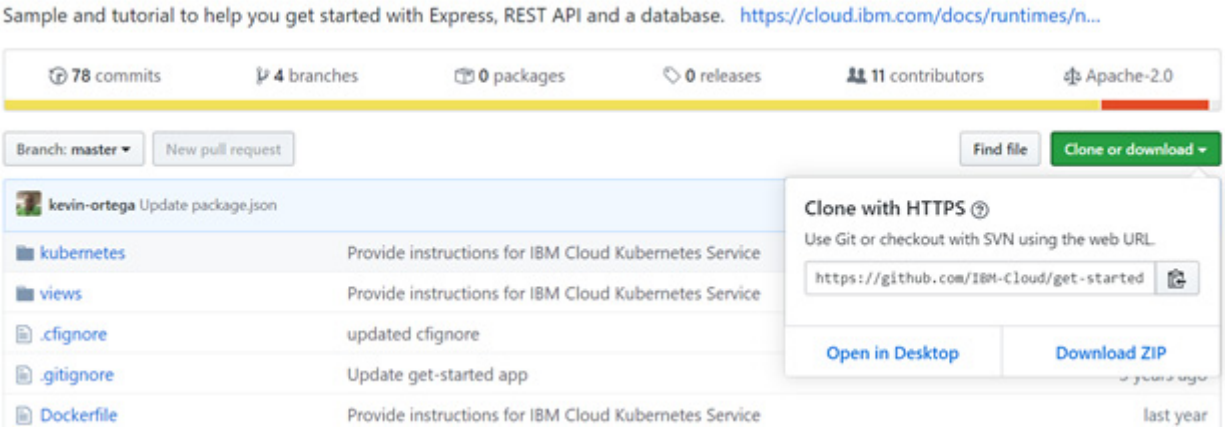
__ c. Click **download the sample code** under **Getting started with SDK for Node.js** as shown in the following figure.




**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.

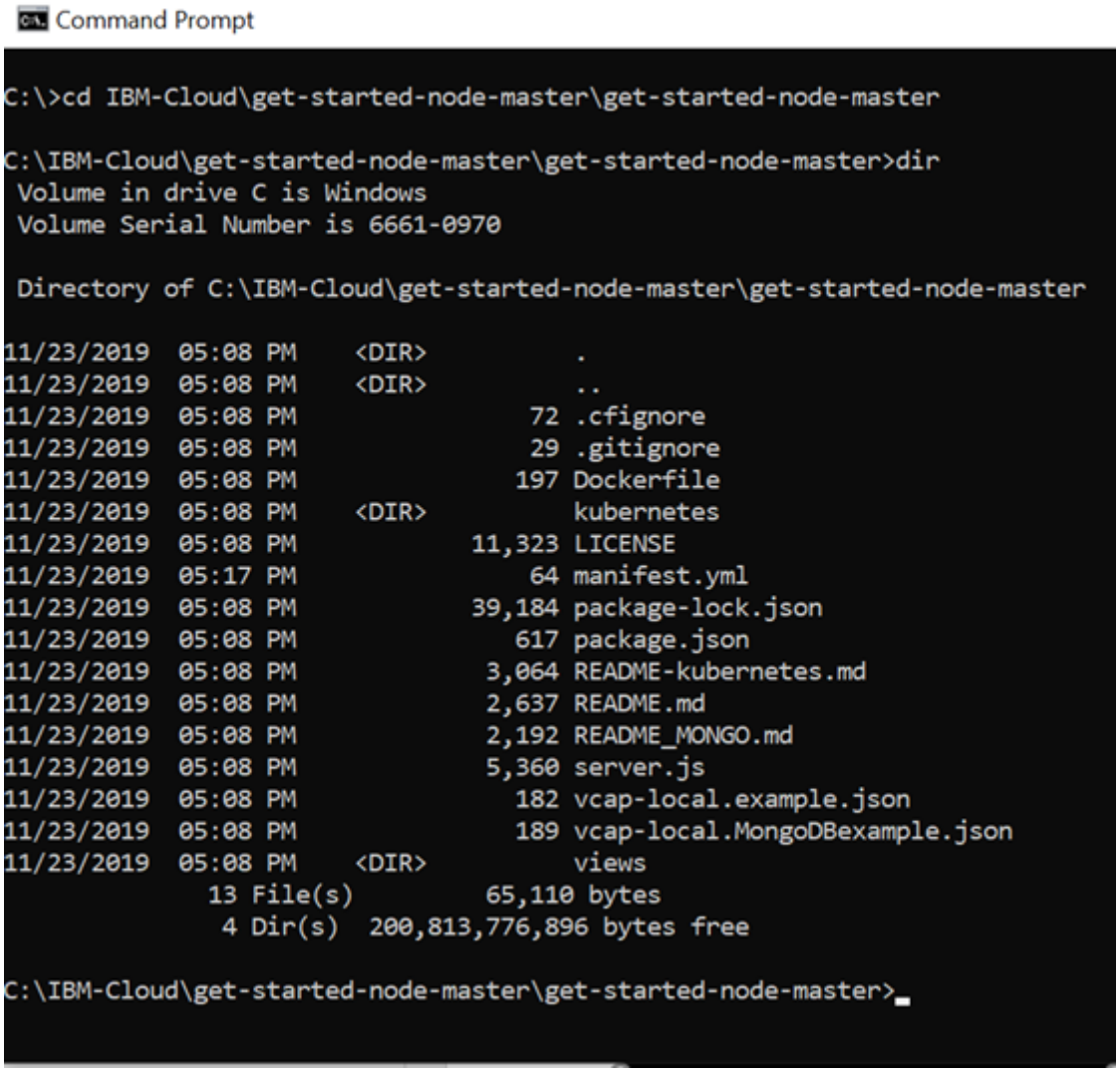
__ d. Download the sample code to your local workstation in the directory `C:\IBM-Cloud`. Click **Clone or download** and select **Download ZIP** as shown in the following figure




__ 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:

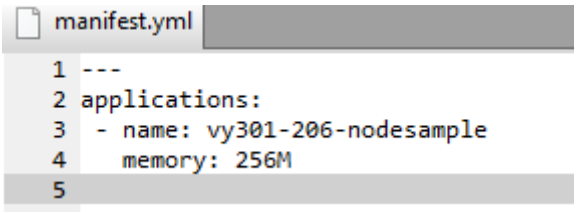
__ a. In a Command Prompt window (Microsoft Windows) or terminal (Mac OS or Linux), change to the directory with the sample code. On almost any operating system, you can do this with the **cd** command.
For example, if you extracted the contents of the application to `\IBM-Cloud\get-started-node-master\get-started-node-master`, 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.




__ 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.




__ 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.


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

- 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* 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:      https://cloud.ibm.com
Region:            eu-gb
User:              <your_email>
Account:           <your_account_name> (f6eeaf39ae6b477fbcead1993509e9c6)
Resource group:    Default
CF API endpoint:   https://api.eu-gb.cf.cloud.ibm.com (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-xxx-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% 1s

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
   Downloading nginx_buildpack...
   Downloaded dotnet-core_v2_4-20190912-1554
   Downloading 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#vendoring
          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.

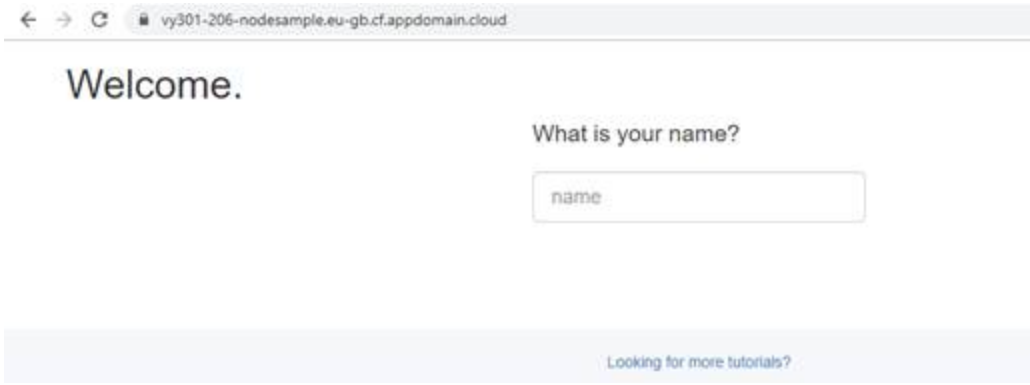
**Note**

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"></sc
ript>
    <!-- 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.messagestore.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.

```
<!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>Greetings</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.messagestore.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": "Greetings.",
                    "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: "
                },
```

__ 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.

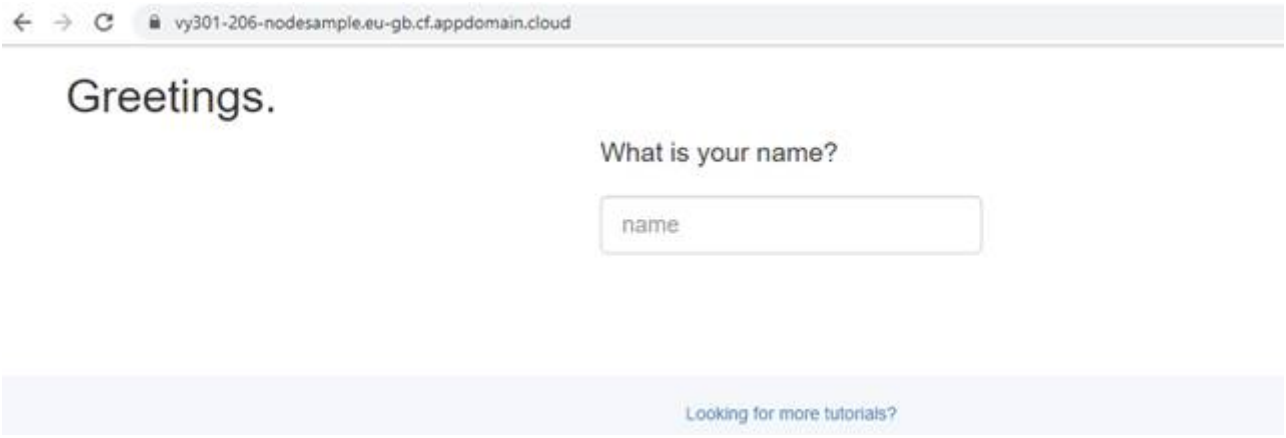
```
ibmcloud cf push vy301-xxx-nodesample
```

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.



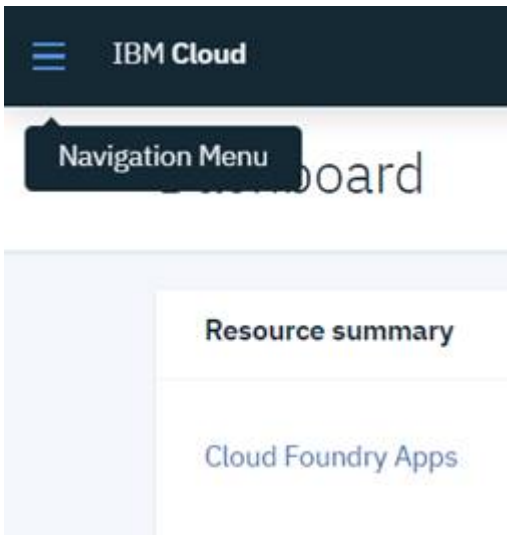

__ 11. Close both the command prompt and the web browser for your application page.

## *Part 5: Checking your organization limits*

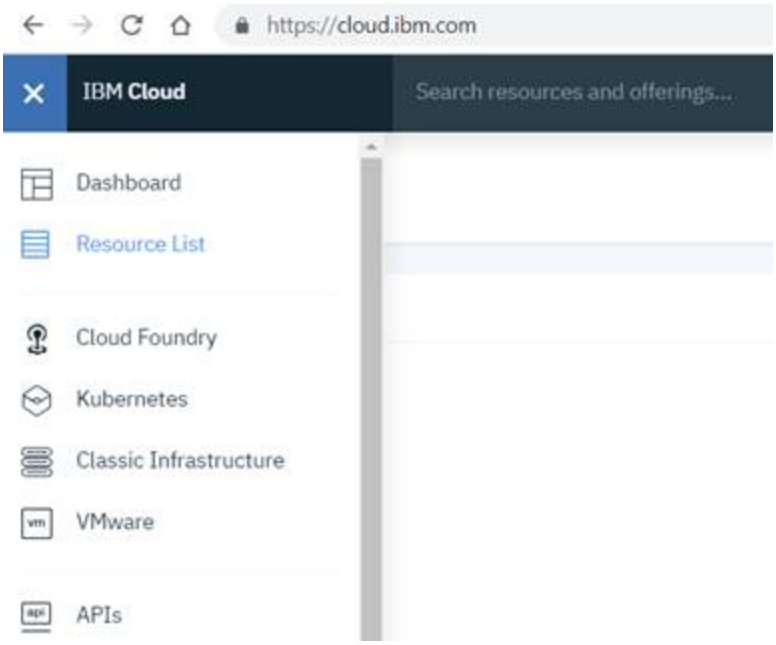
Your IBM Cloud Lite account provides only some services and a maximum amount of random access memory (RAM) that your apps can use. In this section you check the amount of memory allocated to your application.

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.

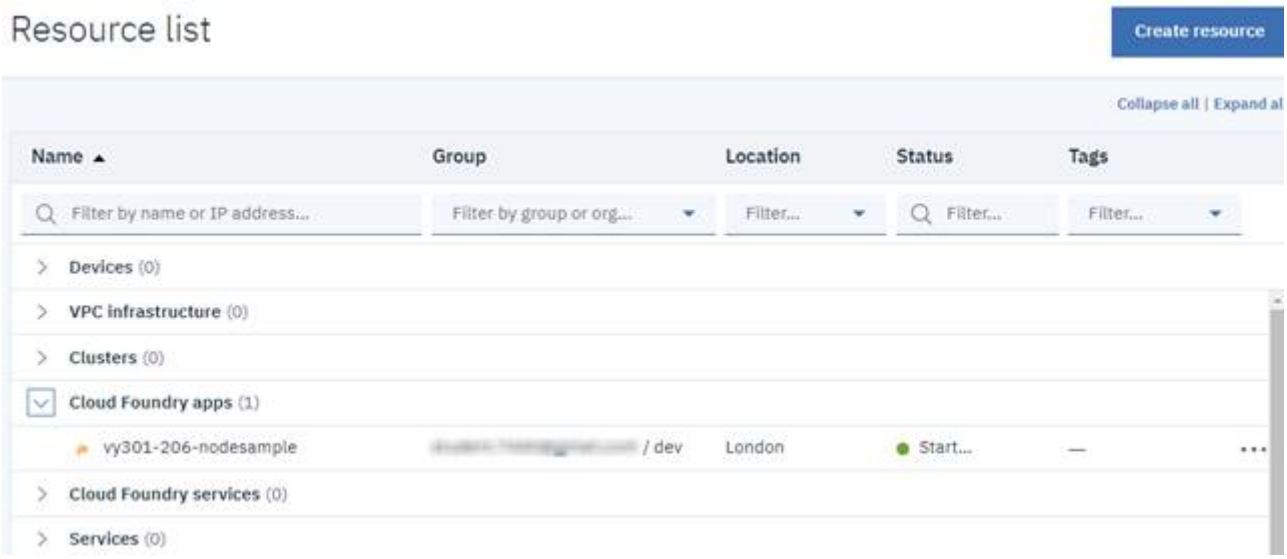



__ 2. Click **Resources List** to open the Resources List to access your application, as shown in the following figure.




__ 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.




__ 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.

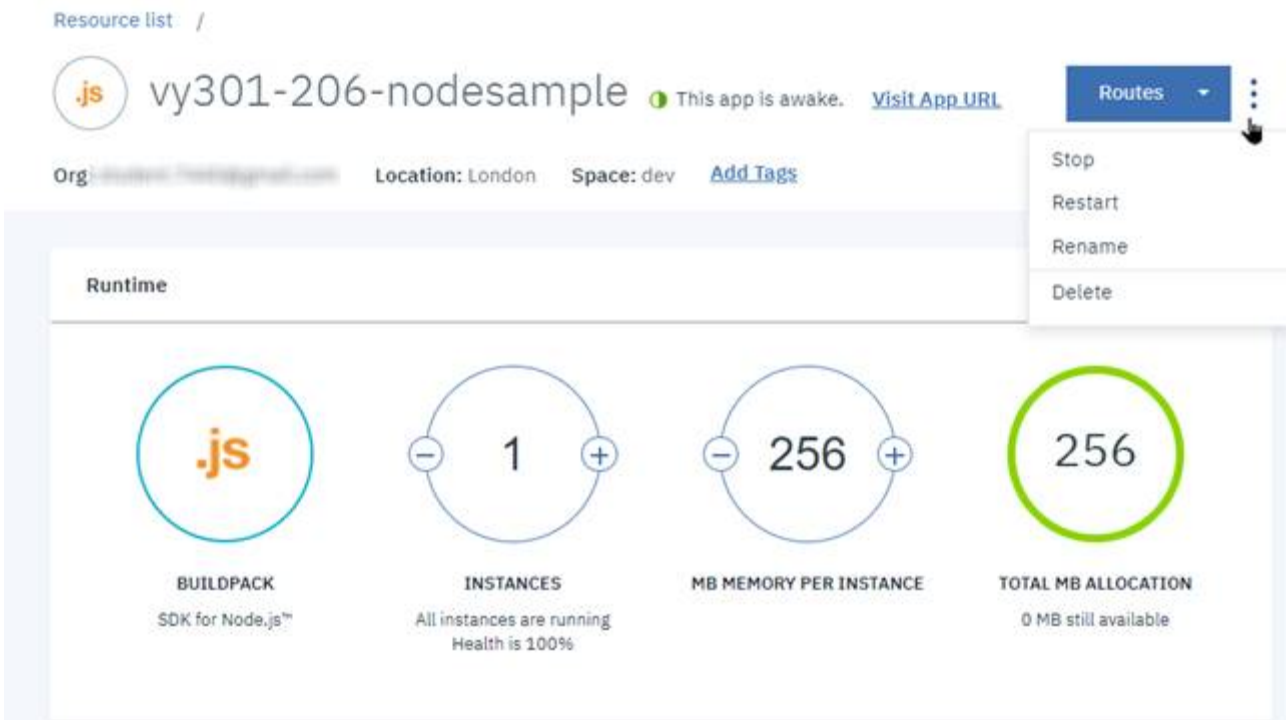

**Note**

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*

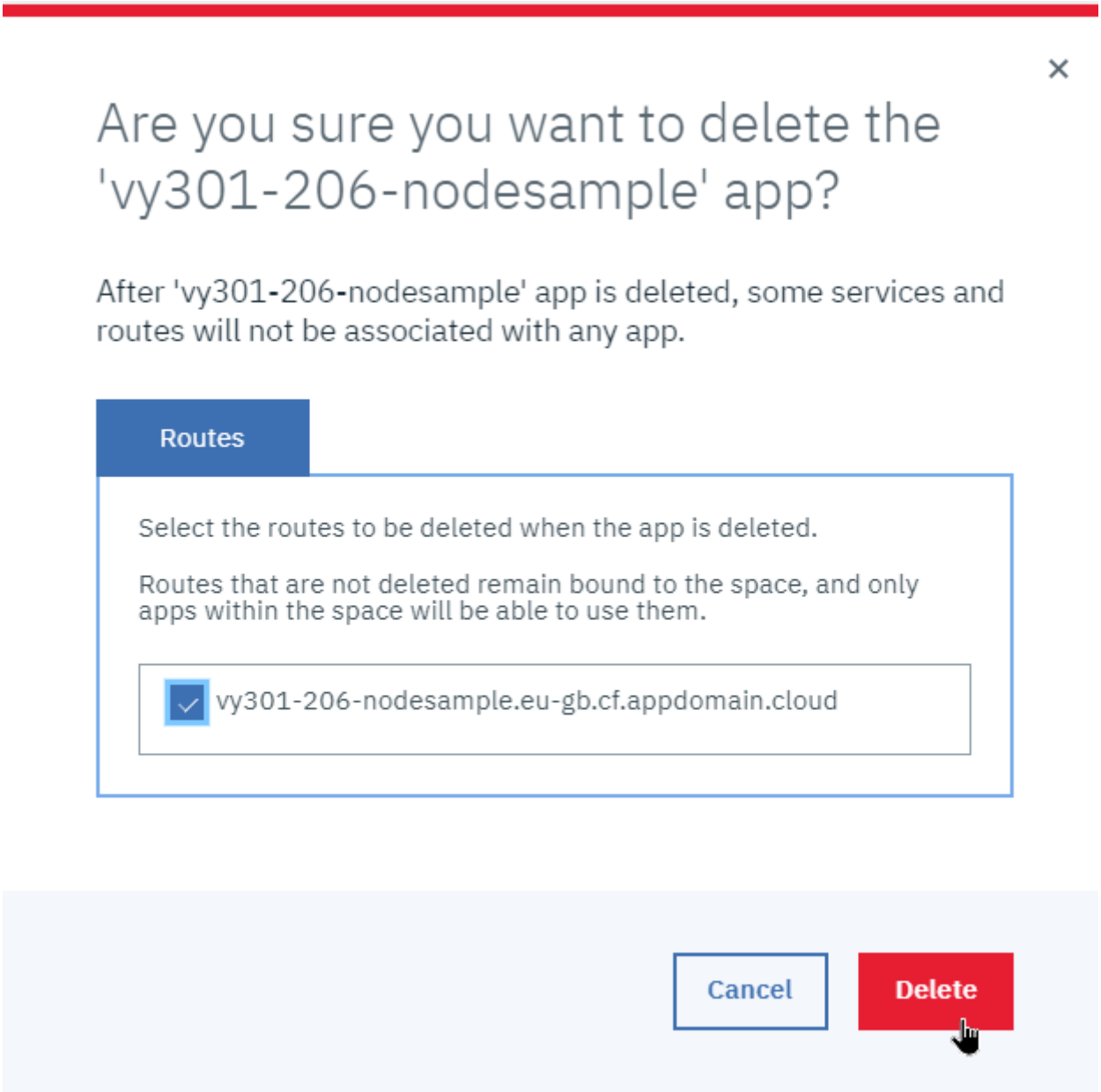
In this section, you delete your app so that your IBM Cloud organization is clean and ready for the next exercise. To delete the sample application complete the following steps:

__ 1. 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.




__ 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.

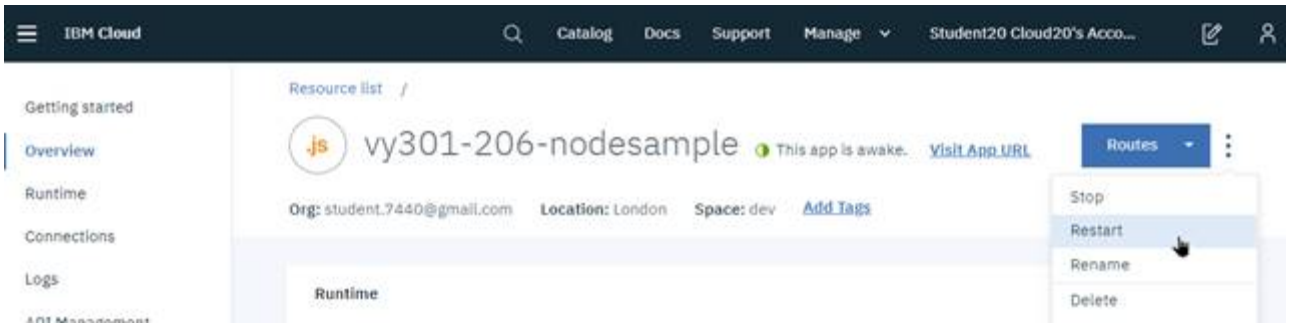
# Troubleshooting

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.