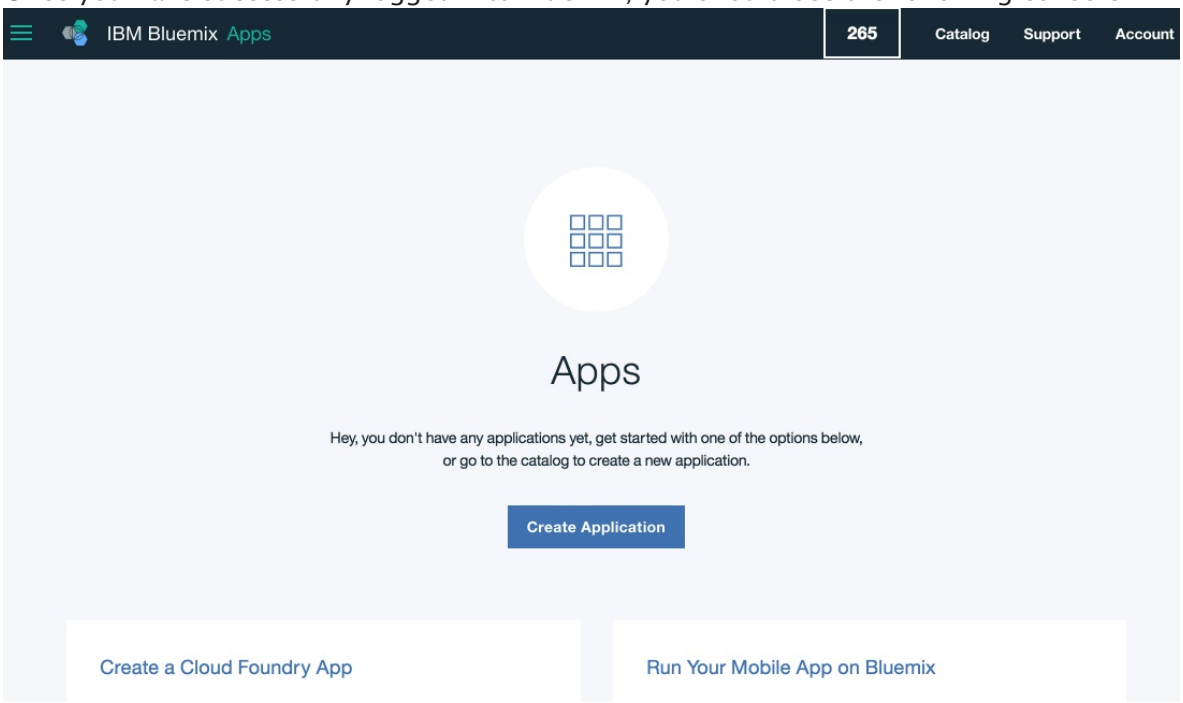
Setup

Follow instructions below to create Node application, and IBM Watson services. Bind the services to the Node application. And finally, download the code. This code will serve as the starting point for the cognitive APIs built later.

Log in to Bluemix

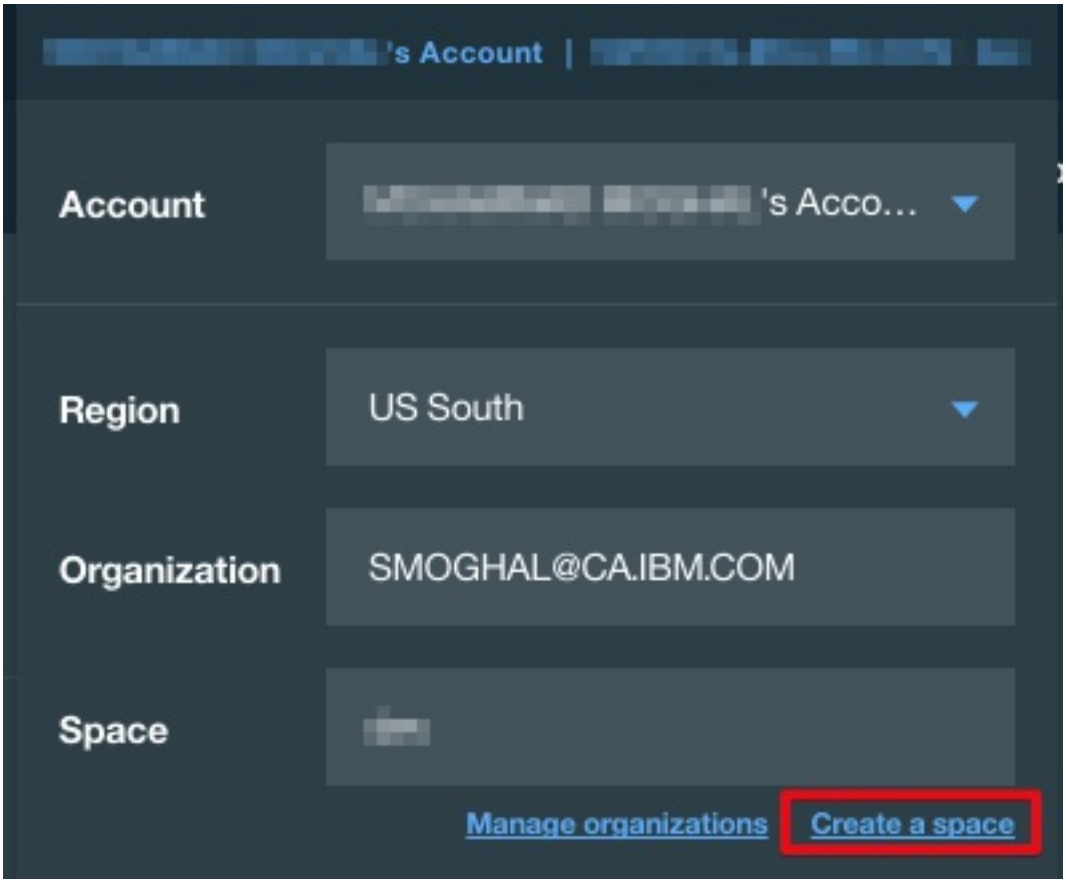
- Launch a new browser window and navigate to IBM Bluemix `http://console.ng.bluemix.net`
- Once you have successfully logged into Bluemix, you should see the following console:



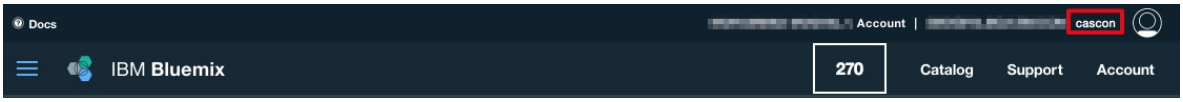
Create a new Space

- We will create a new space called cascon. This is where all you can continue to create all CASCON specific applications. Later on, if you choose to delete the space, all the applications
  and services created in this space will be automatically deleted.

- Click on the Account in the top menu. Then click on `Create a space`

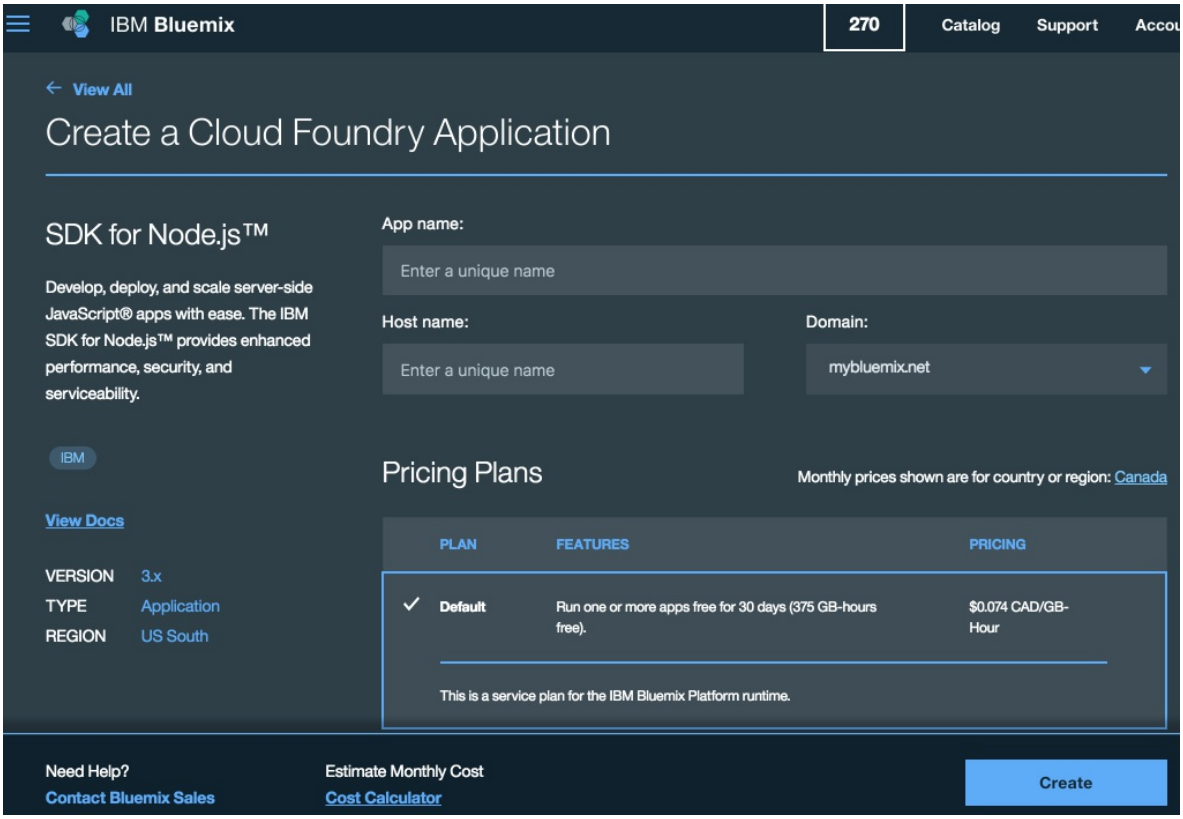


- Type the name of the space as cascon
- Verify that the current space is now set to cascon in the top right corner.



Create Cloud Foundary Application

- In the top menu, click on Catalog (https://new-console.ng.bluemix.net/catalog).
- From the left-hand side menu, click on Cloud Foundary Runtimes (cfruntimes)
- Next, click on SDK for Node.js (nodejsruntime) tile. You should see the following screen:



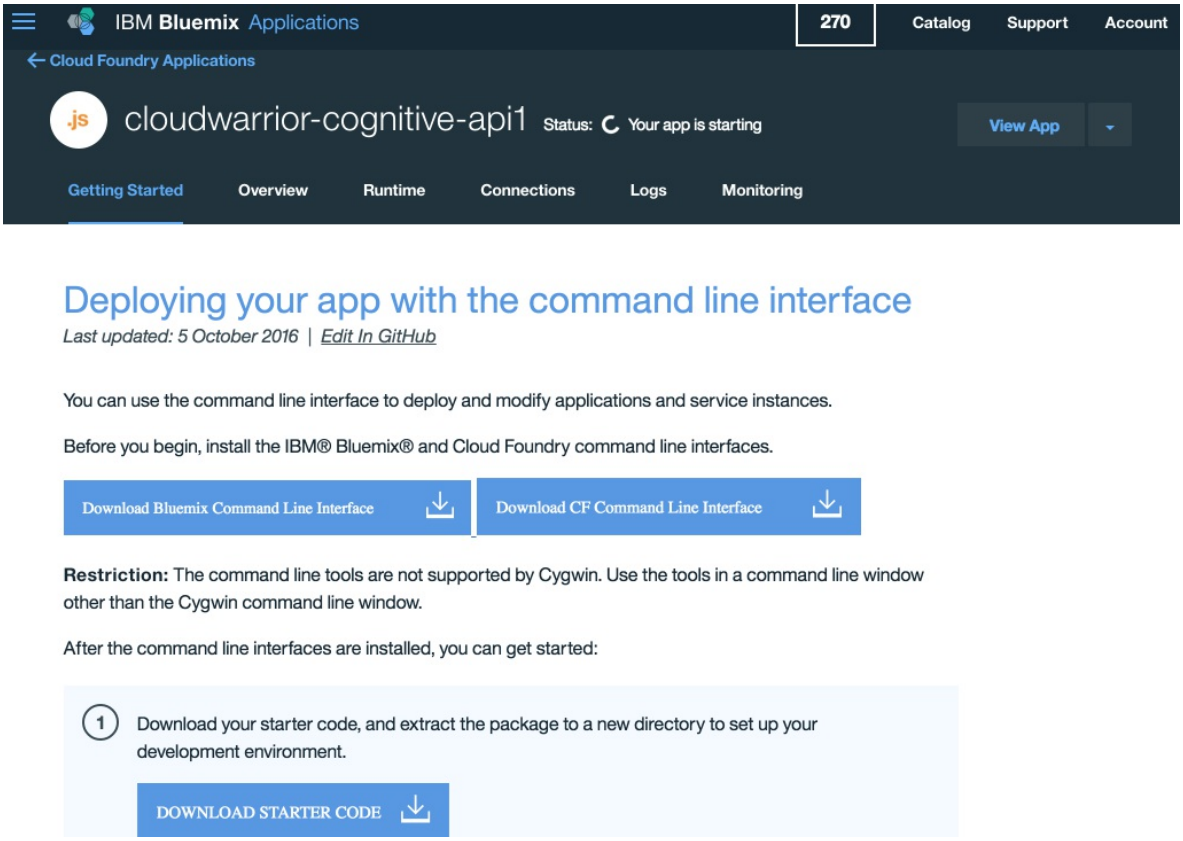
- Under the App Name type a unique application name. We will use <email id>-cognitive-api1.
  For example, if your email id is cloudwarrior@example.com, then the App Name should be cloudwarrior-cognitive-api1.

- Host name is automatically set as App Name is filled.

- Domain should be set to mybluemix.net. This is a publically accessible domain.

- Click Create button at the bottom right corner. Application creation will start. Initially, the status will show Your app is starting. When the application has started, the status will show
  Your app is running.

```
NOTE: If the application state is stopped or not running, you can always
start the application
manually at a later time by clicking the play or restart button next to
"View App" button.
```



- Now, lets create an instance of `Text to Speech` Service. In the top menu, click on Catalog (https://new-console.ng.bluemix.net/catalog).
  From the left menu, click on `Watson` and click on Text to Speech Service (text2speech) tile. This will
  bring up Service instance creation screen.

- Under `Connect to:` drop-down, select your application. And click on `Create` button at the bottom.
  This creates an instance of `Text to Speech` Watson service and binds the service to your Node application.
  This binding will allow the application to interact with the service using the Credentials generated during
  service creation process. We will inspect the credentials in later sections.

  ```
  Remember to click on Restage Application button when prompted
  ```

- Wait for status to change from `Your app is restaging` to `Your app is running`.

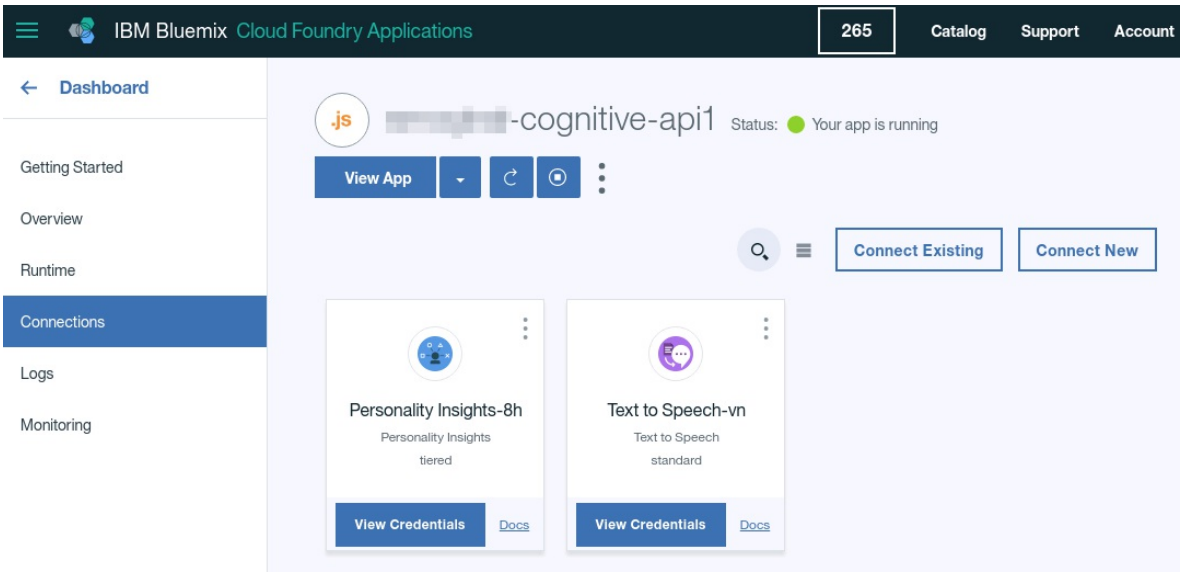
- Next, lets create an instance of `Personality Insights` service. Again, in the top menu, click
  on Catalog (https://new-console.ng.bluemix.net/catalog). From the left menu, click on `Watson` and then click
  Personality Insights Service (personalityinsights) tile. This will bring up Service instance creation screen
  again.

- Under `Connect to:` drop-down, select your application. And click on `Create` button at the bottom.
  This creates an instance of `Personality Insights` Watson service and binds the service

to your Node application.

```
Remember to click on Restage Application button when prompted
```

- Wait for status to change from `Your app is restaging` to `Your app is running`.
- Click on `Overview` menu to view the application.
- Click on your application to bring up the Overview screen. Alternatively, you can navigate to
  the to [Bluemix Dashboard (dashboard)] and click on your application to view the `Overview` screen.
  Verify that the application `Status:` is set to running. The application has two connected services i.e.
  Text to Speech and Personality Insights. The last two characters of service name are always randomly
  generated.




```
IMPORTANT: Note down the name of Text to Speech and Personality Insights
service connections as
they appear in your application Overview screen.  We will need these
service name in later
parts of this tutorial.  In the screen shot above, these service names
are "Personality Insights-8h"
and "Text to Speech-vn", respectively.
```

- Click on `Getting Started` menu, scroll down a bit and locate `DOWNLOAD STARTER CODE` button. Click on the button.
  This will download a zip file and save it in Downloads folder.

```
NOTE: This application has a public URL.  It is  http://<email id>-
cognitive-api1.mybluemix.net.
Try to access the application using a web browser.  Alternatively, click
on "View App" button.
The URL will display default IBM Bluemix Node SDK web page.  An out-of-
box index.html was
created during the application creation.  Since we are building APIs in
this tutorial, we don't
need a web front-end to our application.  We will update index.html and
associated web contents
in the following sections accordingly.
```

Setup Clound Foundary CLI

- Type the following command to point Cloud Foundary (CF) command line to IBM Bluemix API

```
cf api https://api.ng.bluemix.net
```

- Log in to Bluemix using CLI. Use your Bluemix email id and password.

```
cf login
```

- You should see the following output when `cf login` command is successful. Remember to select the
correct organization where you created cascon space earlier.

```
API endpoint: https://api.ng.bluemix.net

Email> your@email.com

Password>
Authenticating...
OK

Select an org (or press enter to skip):
1. YOUR_ORG

Org> 2
Targeted YOUR_ORG

Select a space (or press enter to skip):
1. cascon

Space> 1
Targeted space cascon

API endpoint:   https://api.ng.bluemix.net (API version: 2.54.0)
User:           your@email.com
Org:            YOUR_ORG
Space:          cascon
```

Extract Code

- Open Terminal window.
- Copy the downloaded source code zip file to ~/workshop folder.
  If ~/workshop folder doesn't exist, create it.

```
mkdir ~/workshop
```

- Change directory to ~/workshop folder.

```
cd ~/workshop
```

- Extract the ~Downloads/<email id>-cognitive-api1.zip file into a sub-folder inside ~/workshop
  folder. Keep the name of the folder the same as the app name. Remember to replace <email id>
  with your email id below.

```
unzip ~Downloads/<email id>-cognitive-api1.zip -d <email id>-cognitive-api1
```

Setup Application

- Change directory to the folder where you extracted the source code

```
cd <email id>-cognitive-api1
```

- Install watson-developer-cloud, body-parser and q node modules using the npm command. Make
  sure to use --save parameter; it updates package.json such that when the application is deployed
  in Bluemix environment, appropriate dependencies are installed.

```
npm install watson-developer-cloud body-parser q cors --save
```

- Run npm installation process again to setup the remainder of node modules for local development
  and testing

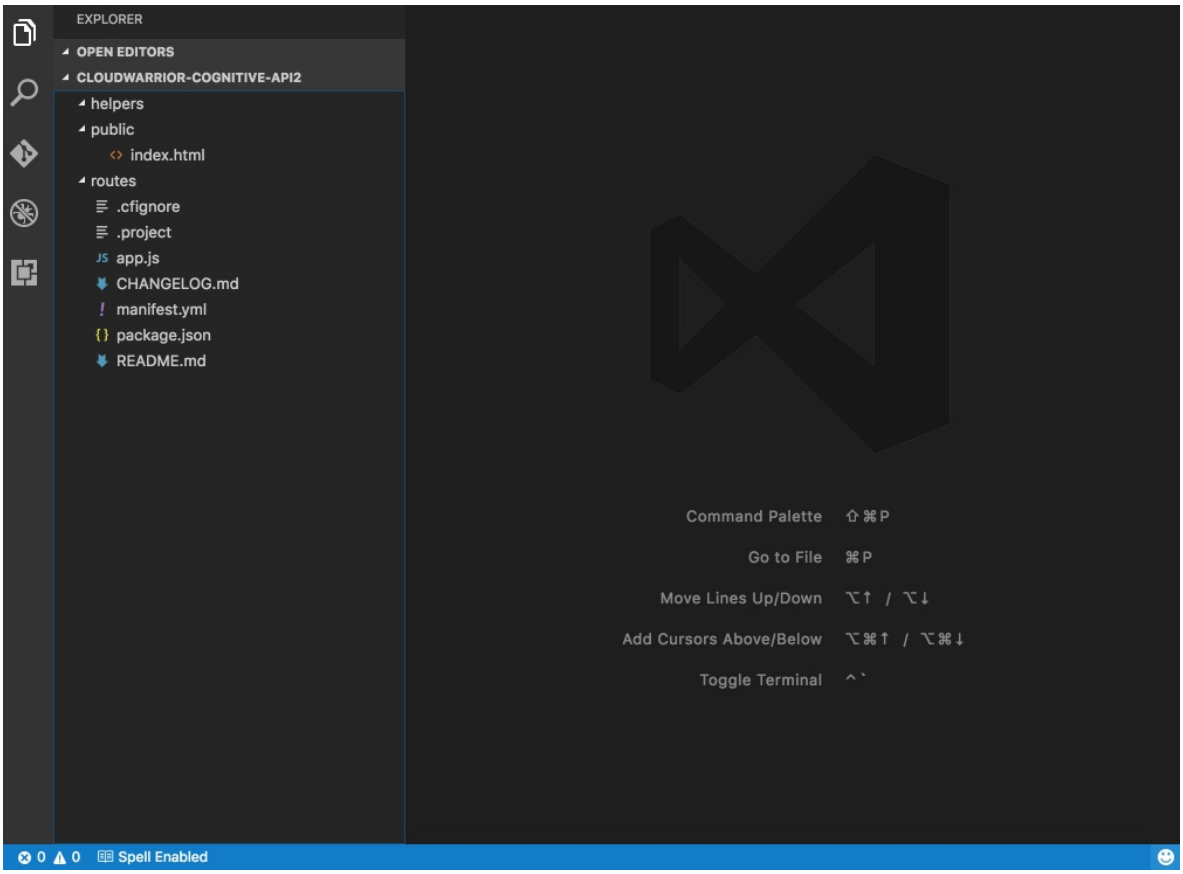
```
npm install
```

- While inside the <email id>-cognitive-api1 directory (the root directory of application), create two
  sub-folders, routes and helpers. These folders will contain our application routes (or APIs) and
  some helper modules.

```
mkdir routes
mkdir helpers
```

Test CLI

- Launch VS Code text editor and open the directory where you extracted the code. VS Code will look like this:



- Notice the empty `routes` and `helpers` folder. We will add code here in later sections.
- Expand `public` folder in VS Code file explorer to the left and delete the `stylesheets` and `images` sub-folders.
- Click on `index.html` to view its contents.
- Change `index.html` as shown in the excerpt below and save the file. We will replace all contents with the
following code. The application that we are building does not have a web-front end. Therefore we don't need
an index.html.

```
<!DOCTYPE html>
<html lang="en">

  <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>CASCON Cognitive API</title>
  </head>

  <body>
    <p>CASCON Cognitive API</p>
  </body>

</html>
```

- Now, lets publish our app to Bluemix and verify that the UI has updated. Type the following
command to publish application updates to Bluemix.

```
cf push
```

- You should see the following output when `cf push` command is successful.

```
Using manifest file YOUR_APP_NAME/manifest.yml

Updating app YOUR_APP_NAME in org YOUR_ORG / space cascon as YOUR_ORG...
OK

Using route YOUR_APP_NAME.mybluemix.net
Uploading YOUR_APP_NAME...
Uploading app files from: YOUR_APP_NAME
Uploading 3.2K, 7 files
Done uploading
OK
Binding service Text to Speech-XX to app YOUR_APP_NAME in org YOUR_ORG /
space cascon as YOUR_ORG...
OK
Binding service Personality Insights-YY to app YOUR_APP_NAME in org
YOUR_ORG / space cascon as YOUR_ORG...
OK

Stopping app YOUR_APP_NAME in org YOUR_ORG / space cascon as YOUR_ORG...
OK

Starting app YOUR_APP_NAME in org YOUR_ORG / space cascon as YOUR_ORG...
-----> Downloaded app package (4.0K)
-----> Downloaded app buildpack cache (488K)

-----> IBM SDK for Node.js Buildpack v3.8-20161006-1211
       Based on Cloud Foundry Node.js Buildpack v1.5.20
       NPM_CONFIG_LOGLEVEL=error
```

```
      NPM_CONFIG_PRODUCTION=true
      NODE_MODULES_CACHE=true
      NODE_ENV=production
.........
-----> Uploading droplet (17M)

0 of 1 instances running, 1 starting
1 of 1 instances running

App started


OK

App YOUR_APP_NAME was started using this command
`./vendor/initial_startup.rb`

Showing health and status for app YOUR_APP_NAME in org YOUR_ORG / space
cascon as YOUR_ORG...
OK

requested state: started
instances: 1/1
usage: 256M x 1 instances
urls: YOUR_APP_NAME.mybluemix.net
last uploaded: Wed Oct 19 05:34:28 UTC 2016
stack: unknown
buildpack: SDK for Node.js(TM) (ibm-node.js-4.6.0, buildpack-v3.8-
20161006-1211)

    state     since                    cpu    memory          disk
details
#0   running   2016-10-19 01:35:19 AM   0.2%   69.7M of 256M   64.7M of
1G
```

- Finally, launch the application in a browser and verify that UI is updated. The web page should now show CASCON Cognitive API. You can either launch by typing application URL http://<email id>-cognitive-api1.mybluemix.net in a browser window
  or by clicking View App button in IBM Bluemix application overview screen.