

IBM Bluemix demo - NodeRed

In this lab, you will demonstrate several Bluemix capabilities - including the NodeJS runtime, Watson sentiment analysis, Social Media integration via Twitter, the NodeRed graphical editor for building NodeJS programs, and the Cloudant noSQL database.

////////////////////////////////////

Pework

Make sure all these items are completed and working **before** you head into the demo. Also (and this goes without saying) make sure there is a working Internet connection available in the demo location. It's pretty much impossible to demo the Cloud without access to the internet...

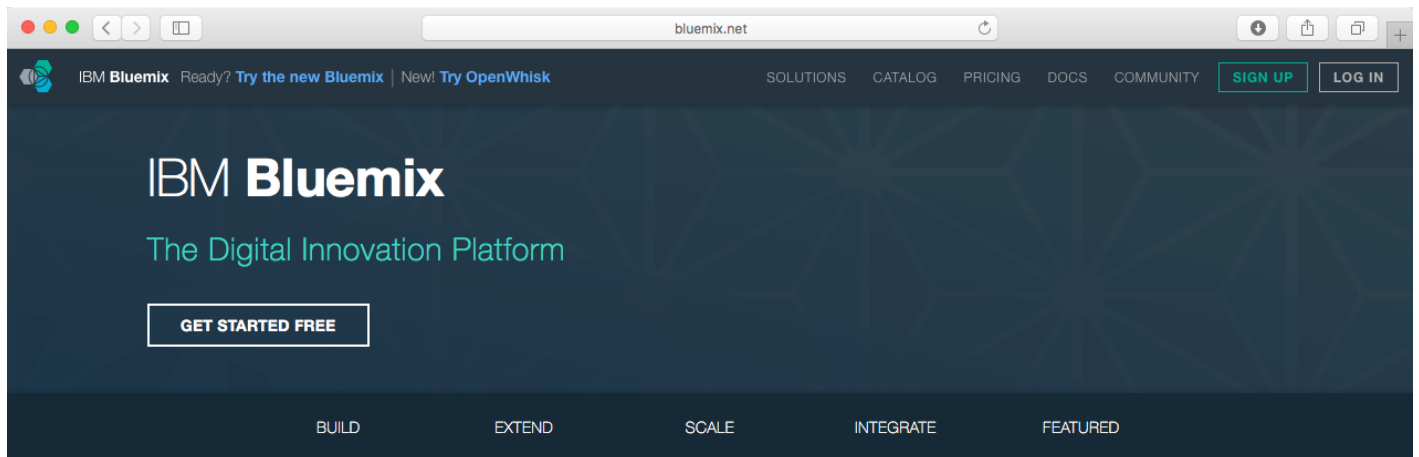
Create a Twitter account

If you already have one, you can skip this step. If you don't, go to www.twitter.com and sign up. Remember your username and password, as you'll need them for the demo.

Create a Bluemix account

////////////////////////////////////

1. Open a browser and visit <http://www.bluemix.net>
2. Press the SIGN UP button



Build your apps, your way.

Use a combination of the most prominent open-source compute technologies to power your apps. Then, let Bluemix handle the rest.

Instant Runtimes

App-centric runtime environments based on Cloud

IBM Containers

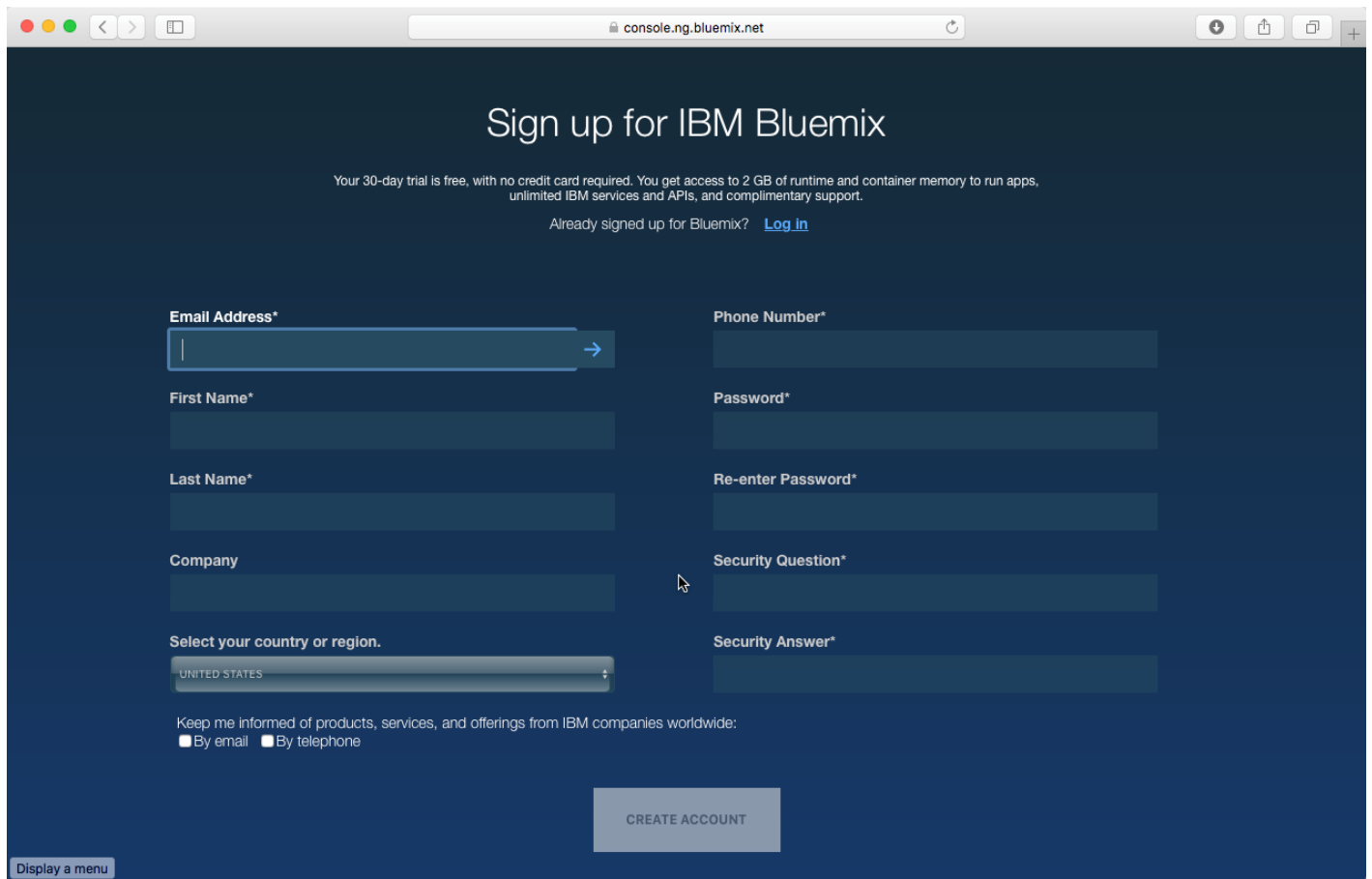
Portable and consistent delivery of your app without

Virtual Servers

Get the most flexibility and control over your environment

Display a menu

3. Complete the form and press the CREATE ACCOUNT button



The image shows a web browser window with the URL `console.ng.bluemix.net`. The page is titled "Sign up for IBM Bluemix". Below the title, a message states: "Your 30-day trial is free, with no credit card required. You get access to 2 GB of runtime and container memory to run apps, unlimited IBM services and APIs, and complimentary support." Below this, a link says "Already signed up for Bluemix? [Log in](#)".

The sign-up form consists of the following fields:

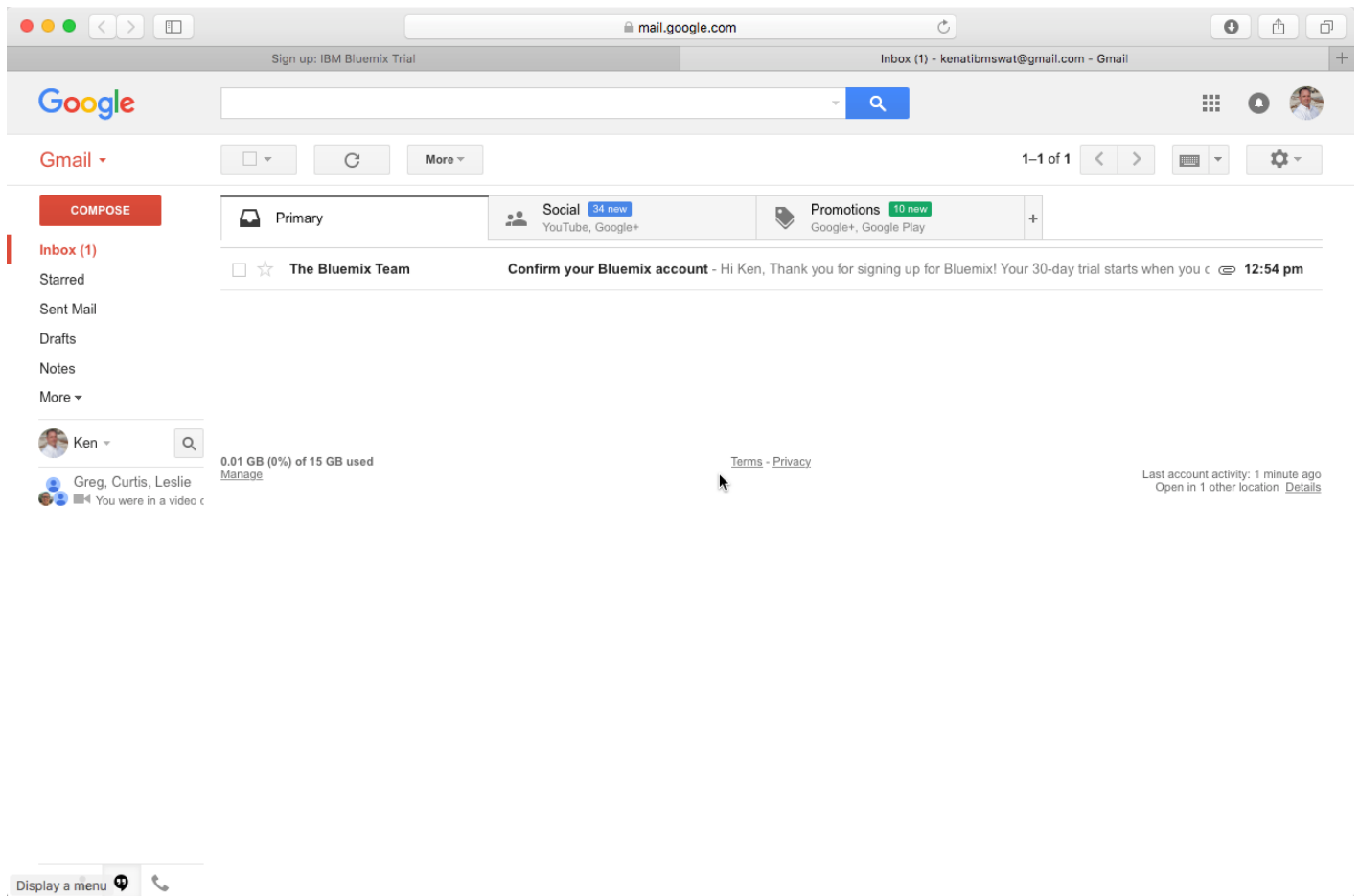
- Email Address***: A text input field with a blue arrow button to its right.
- Phone Number***: A text input field.
- First Name***: A text input field.
- Last Name***: A text input field.
- Company**: A text input field.
- Select your country or region.**: A dropdown menu currently showing "UNITED STATES".
- Password***: A text input field.
- Re-enter Password***: A text input field.
- Security Question***: A text input field.
- Security Answer***: A text input field.

Below the form fields, there is a checkbox section:

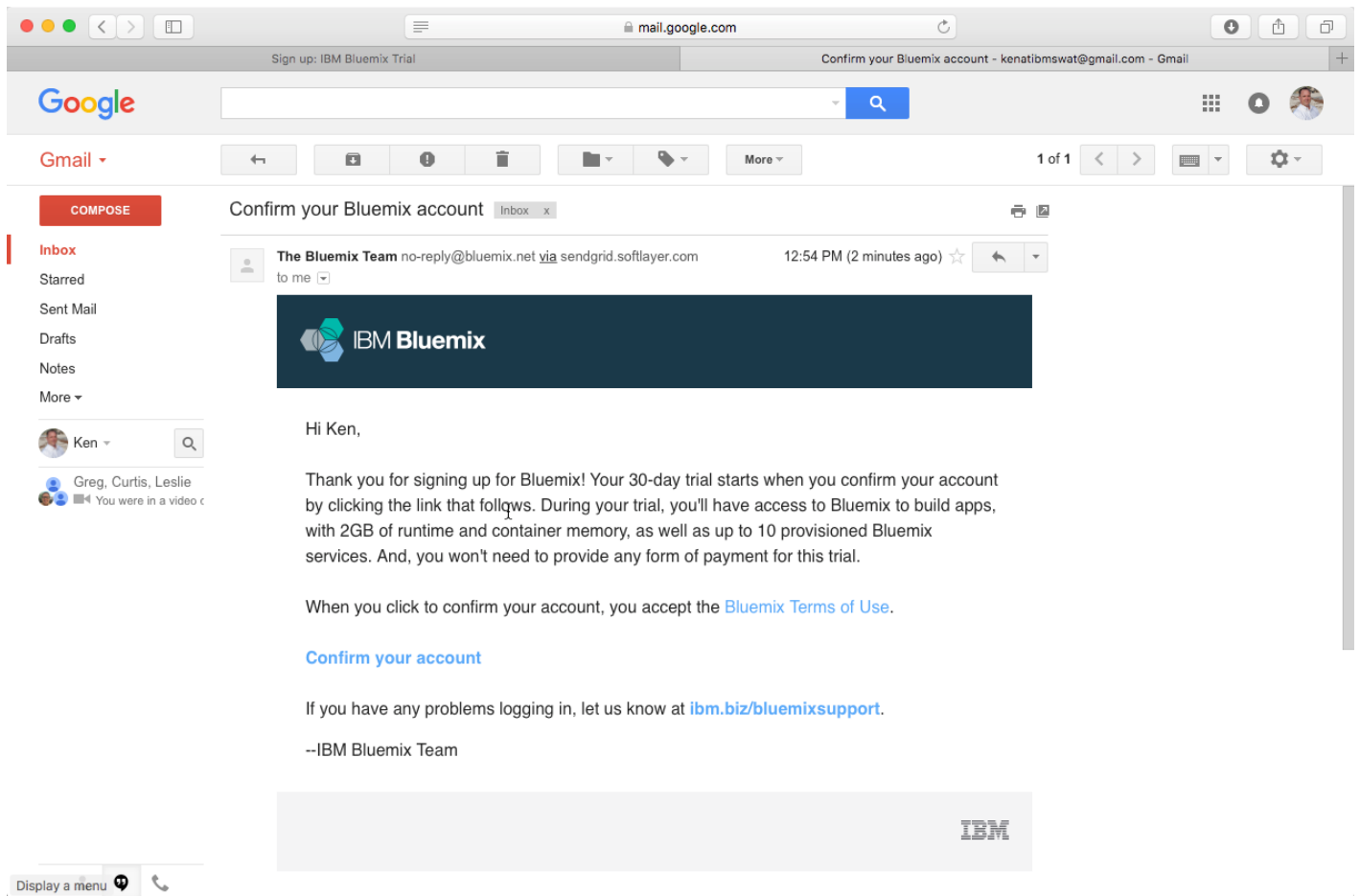
Keep me informed of products, services, and offerings from IBM companies worldwide:
☐ By email ☐ By telephone

At the bottom center is a button labeled "CREATE ACCOUNT". In the bottom left corner, there is a link that says "Display a menu".

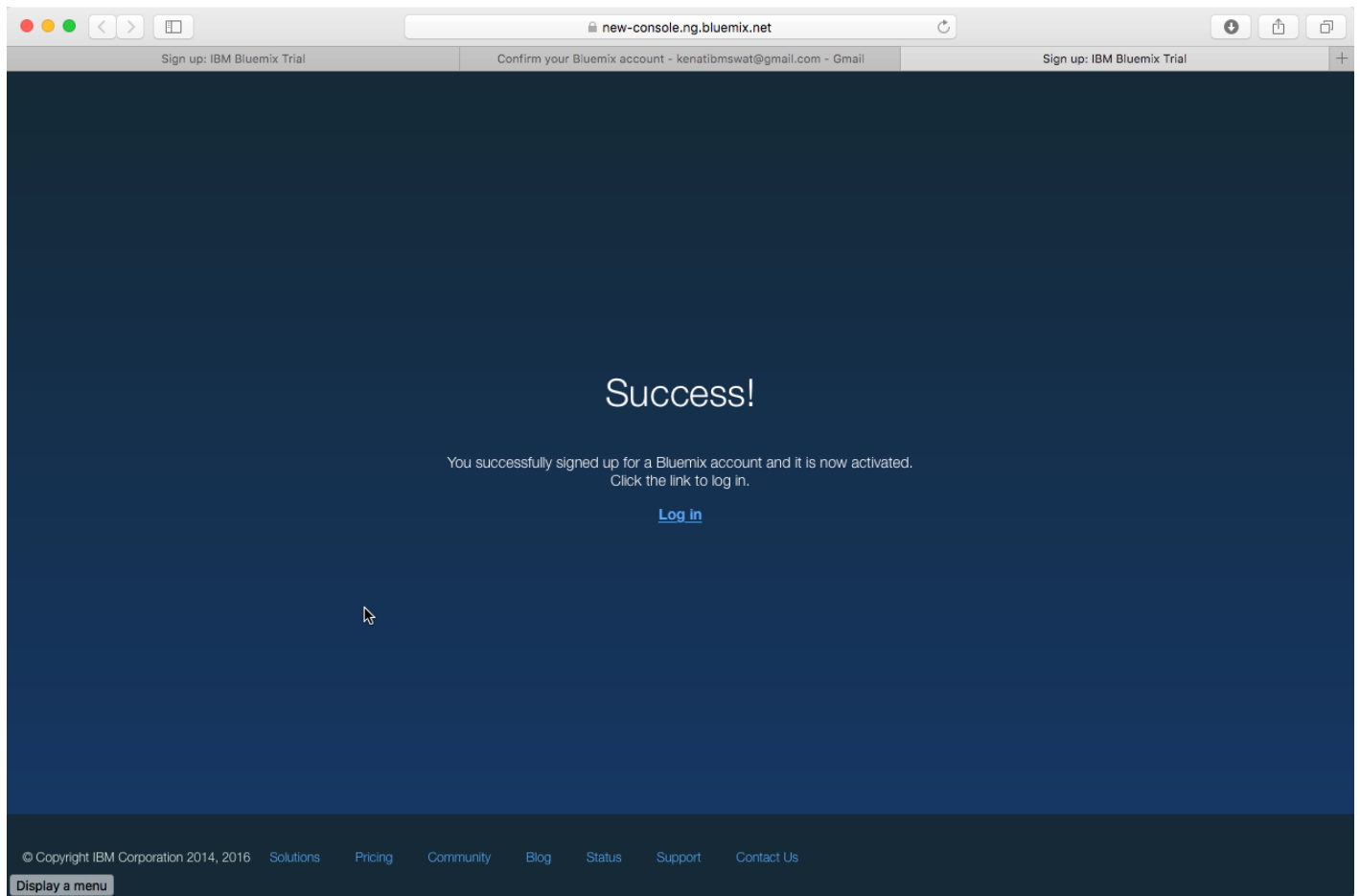
4. Check your email for your next steps.



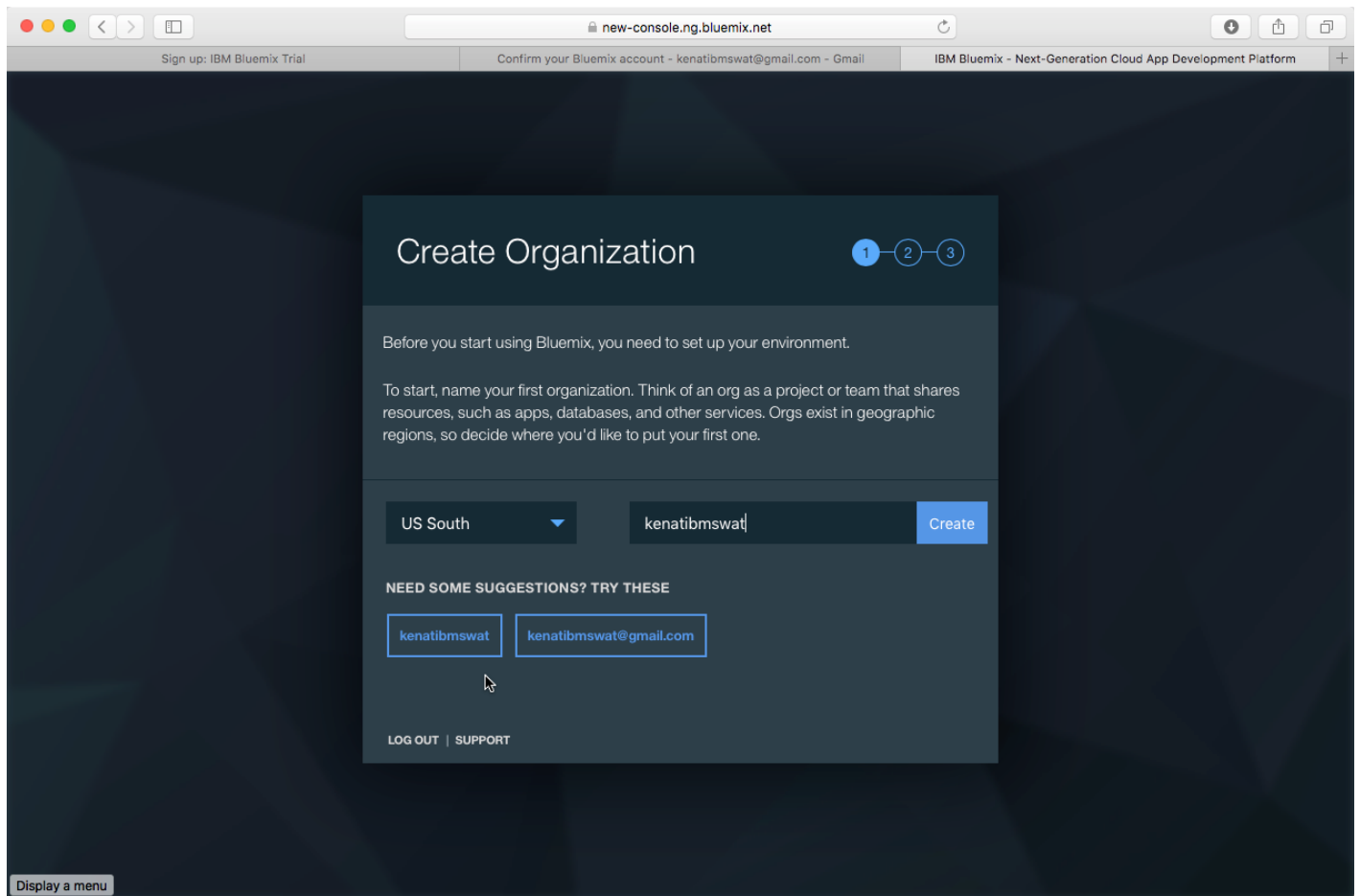
5. Open the email and click the Confirm your account link



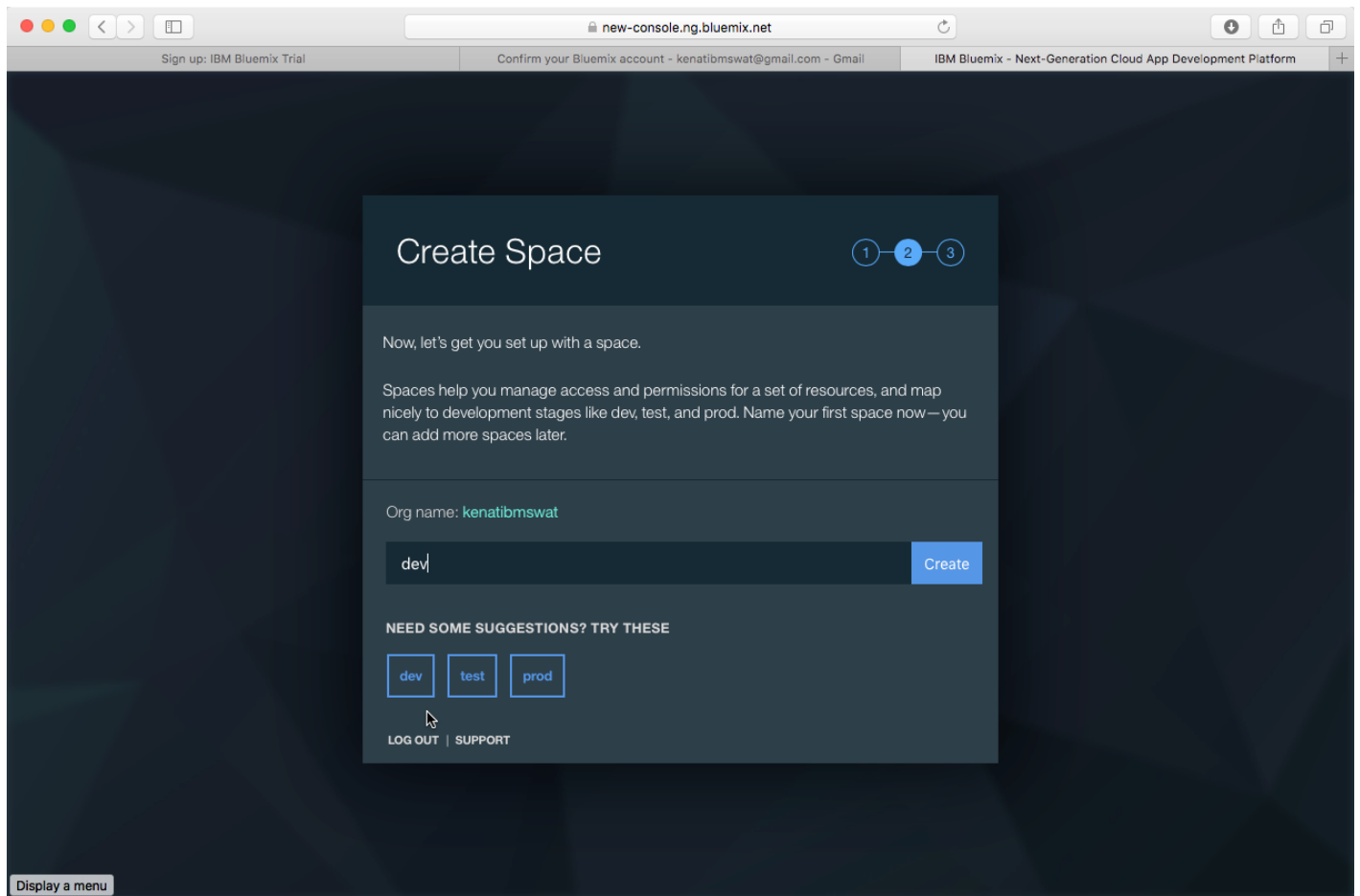
6. Once confirmed, you will be taken to a page that says Success! To login, click the Log In link



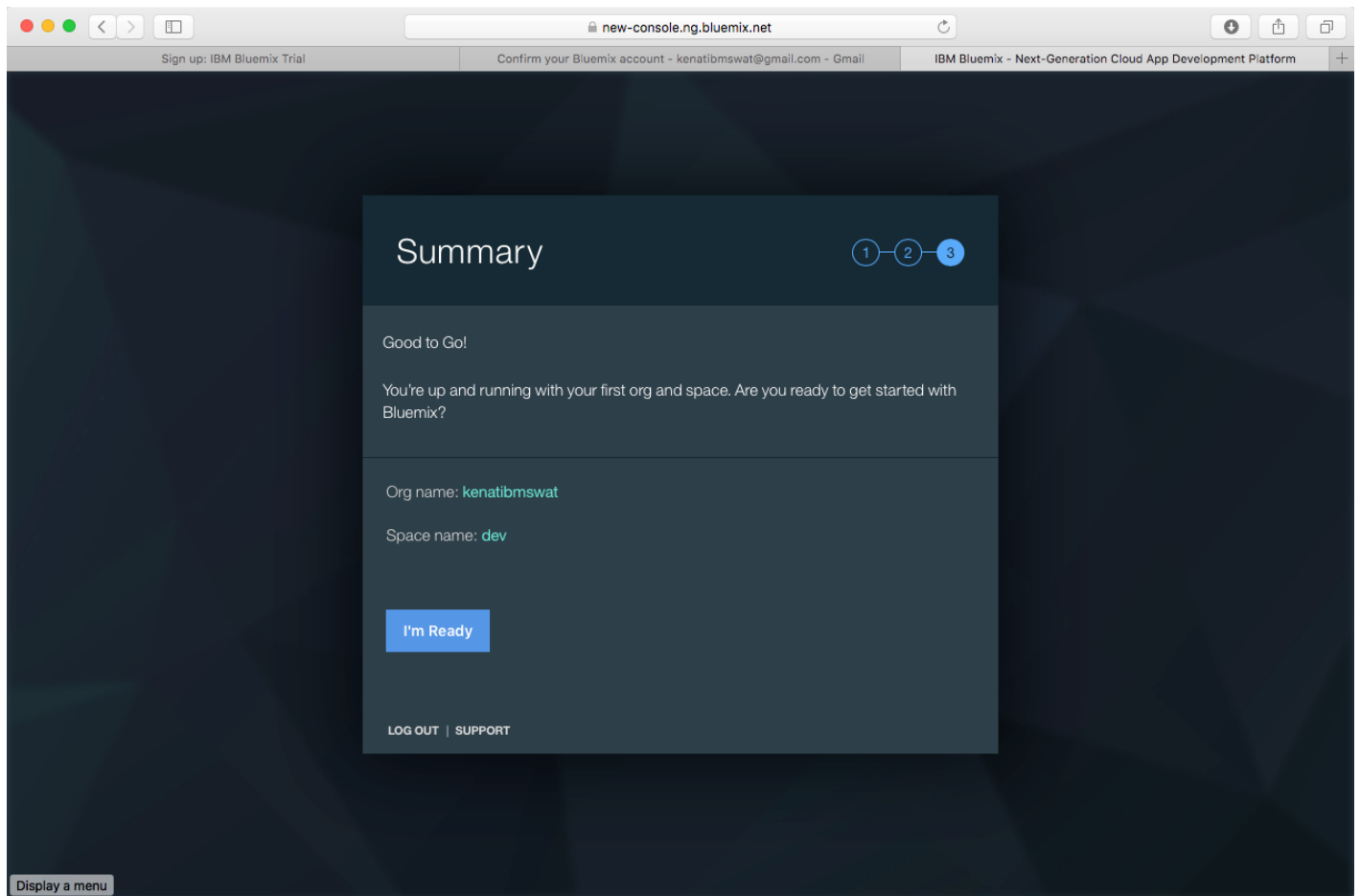
7. Enter your email address and press the CONTINUE button
8. Then enter your password on the next page and press the LOG IN button
9. You will be prompted to create an organization. Enter an organization name (notice that there are suggestions for you). This can be anything, but is typically an email address, since that can be guaranteed to be unique within Bluemix. Make sure **US South** is selected as the region. Press the CREATE button to continue.



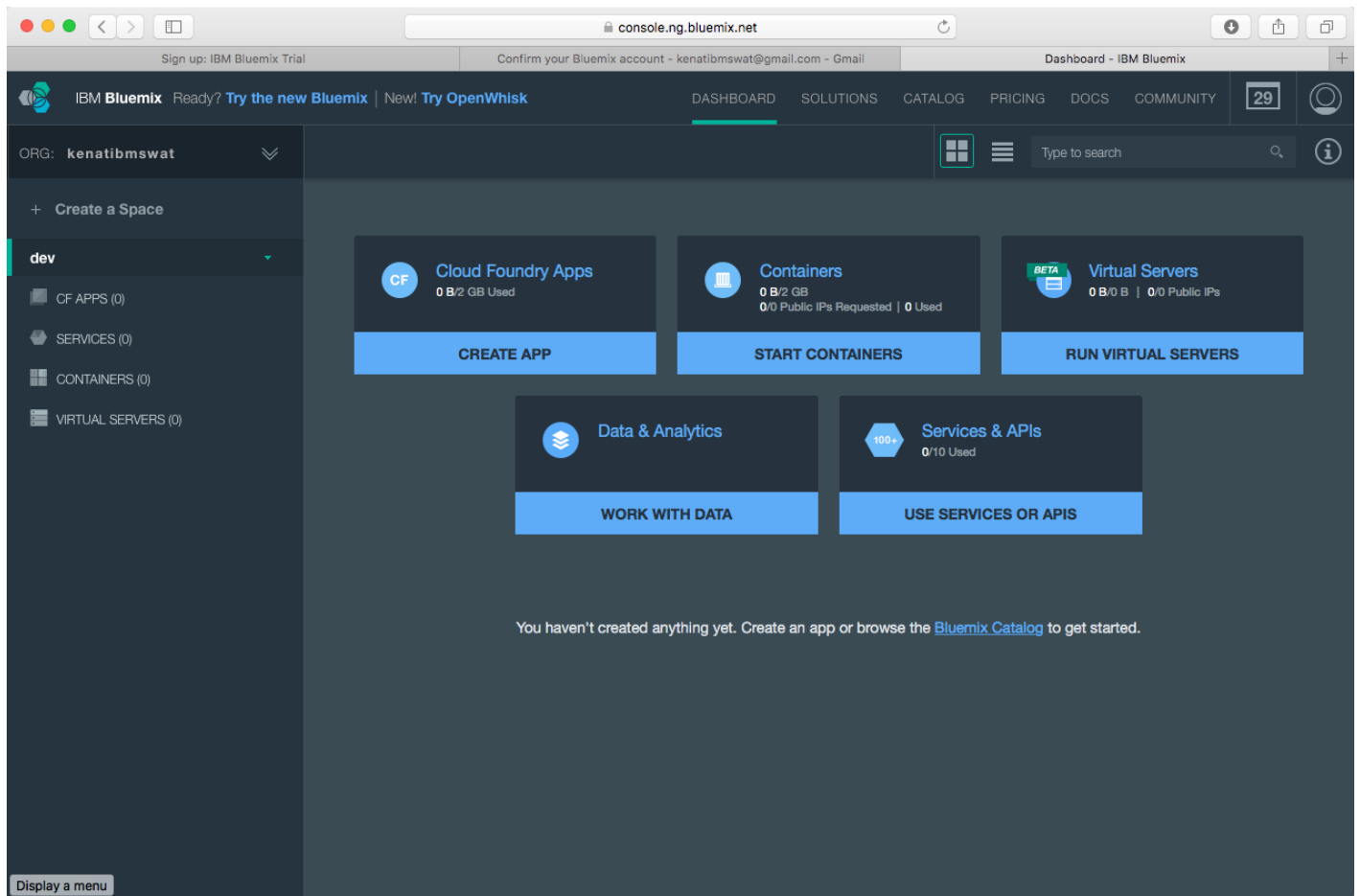
10. Next you will be prompted to create a space such as dev, demo, test, prod, etc. You can name it whatever you would like (again notice the recommendations). Then press the CREATE button.



11. Next you will see the Summary page where you can review your entries. Press the I'm Ready button



12. Once you see the following screen, your account has been created and configured.



Provision the Node-RED Starter boilerplate

1. Logon to your Bluemix account.
2. In the catalog, navigate to the **Boilerplates** section, and click on the Node-RED Starter tile.

IBM Bluemix Console

Catalog Docs 284 Paul Horan p.horan@us.ibm.com | dev

Catalog

All Categories

Boilerplates >

Compute

Network

Storage

Data & Analytics

Watson

Internet of Things

APIs

DevOps

Security


Application Services


Mobile


Integrate


Search


Filter


**ASP.NET Core Cloudant Starter**
Use the Cloudant NoSQL DB Service in an ASP.NET Core application.
IBM


**Internet of Things Platform Starter**
Get started with IBM Watson IoT platform using the Node-RED Node.js sample application. With the...
IBM


**IoT for Electronics Starter**
IoT for Electronics is an integrated end-to-end solution (made of multiple services and apps) th...
IBM


**Java Cloudant Web Starter**
Use the Cloudant NoSQL DB service with the "Liberty for Java™" runtime.
IBM


**Java Workload Scheduler Web Starter**
This application demonstrates how to use the Workload Scheduler service, with the "Liberty f...
IBM


**LoopBack Starter**
This is a sample StrongLoop LoopBack Node.js application, powered by the open source LoopB...
IBM


**MobileFirst Services Starter**
Start building your next mobile app with mobile services on Bluemix.
IBM


**Node.js Cloudant DB Web Starter**
Use the Cloudant NoSQL DB service with the "SDK for Node.js™" runtime.
IBM


**Personality Insights Java Web Starter**
A simple Java app that uses the Personality Insights service to analyze text to derive personality traits.
IBM


**Personality Insights Node.js Web Starter**
A simple Node.js app that uses Personality Insights to analyze text to derive personality traits.
IBM

**StrongLoop Arc**
This application is the StrongLoop Arc graphical UI, which includes tools for building, profiling and...
IBM

**Mendix Rapid Apps**
Model driven rapid app platform that allow users to build, integrate and deploy web and mobile
Community

**Node-RED Starter**
This application demonstrates how to run the Node-RED open-source project within IBM Bluemix.
IBM

**Python Flask**
A simple Python Flask application that will get you up and running quickly.
IBM

**Ruby Sinatra**
Develop a Ruby web application using the Sinatra framework.
IBM

3. Give the Node-RED application a name. The name you provide will be concatenated with the domain name `bluemix.net` to create a URL, so it must be unique across the entire internet. Choose wisely... Click the CREATE button to provision the service.

IBM Bluemix Console | Catalog | Docs | 284 | Paul Horan | pahoran@us.ibm.com | dev

Create a Cloud Foundry Application

Node-RED Starter

This application demonstrates how to run the Node-RED open-source project within IBM Bluemix.

[Community](#) | [View Docs](#)

VERSION: 0.5.0
TYPE: Boilerplate
REGION: US South

App name: PHNodeRedDemo

Host name: PHNodeRedDemo

Domain: mybluemix.net

Selected Plan: Default

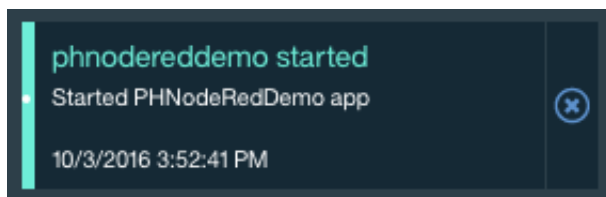
SDK for Node.js™: Default

Cloudant NoSQL DB: Lite

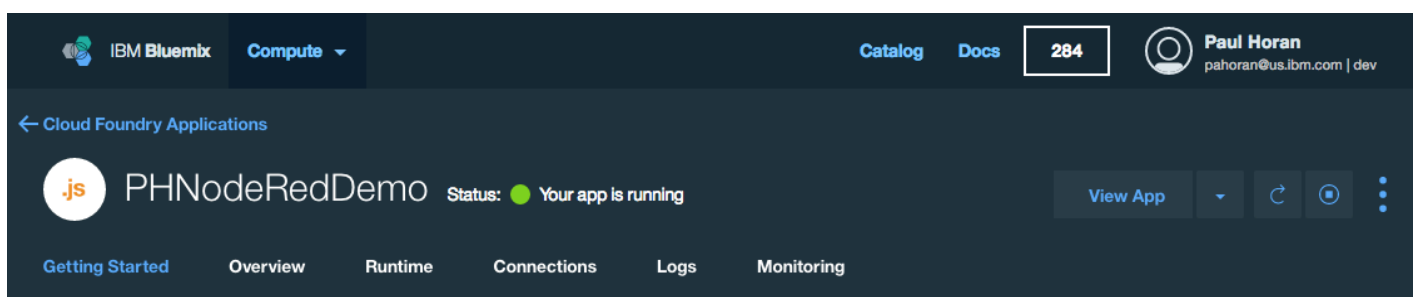
Develop, deploy, and scale server-side JavaScript® apps with ease. The IBM SDK for Node.js™ provides enhanced performance, security, and serviceability.

[Need Help? Contact Bluemix Sales](#) | [Estimate Monthly Cost Cost Calculator](#) | [Create](#)

4. After a few seconds, you will see the notification that your Node-RED app has started.

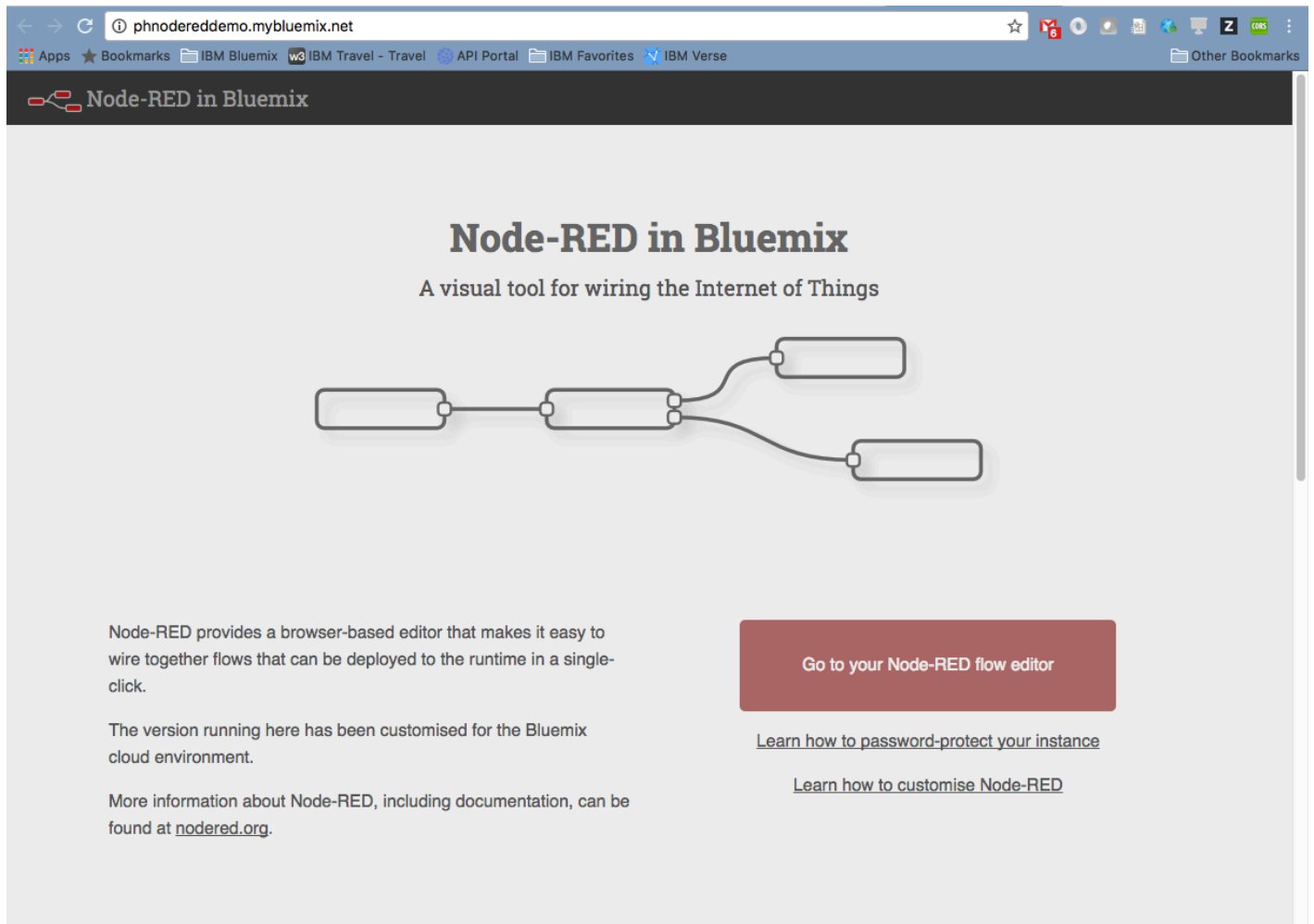


5. To verify that the service is up and that the URL is available, make sure that the Status is showing a green dot, with the text *Your app is running*. Refresh the screen every few seconds until that changes. It should take less than a minute. Once that appears, click the button labeled **View App**.



6. That will open your default browser, and load the URL

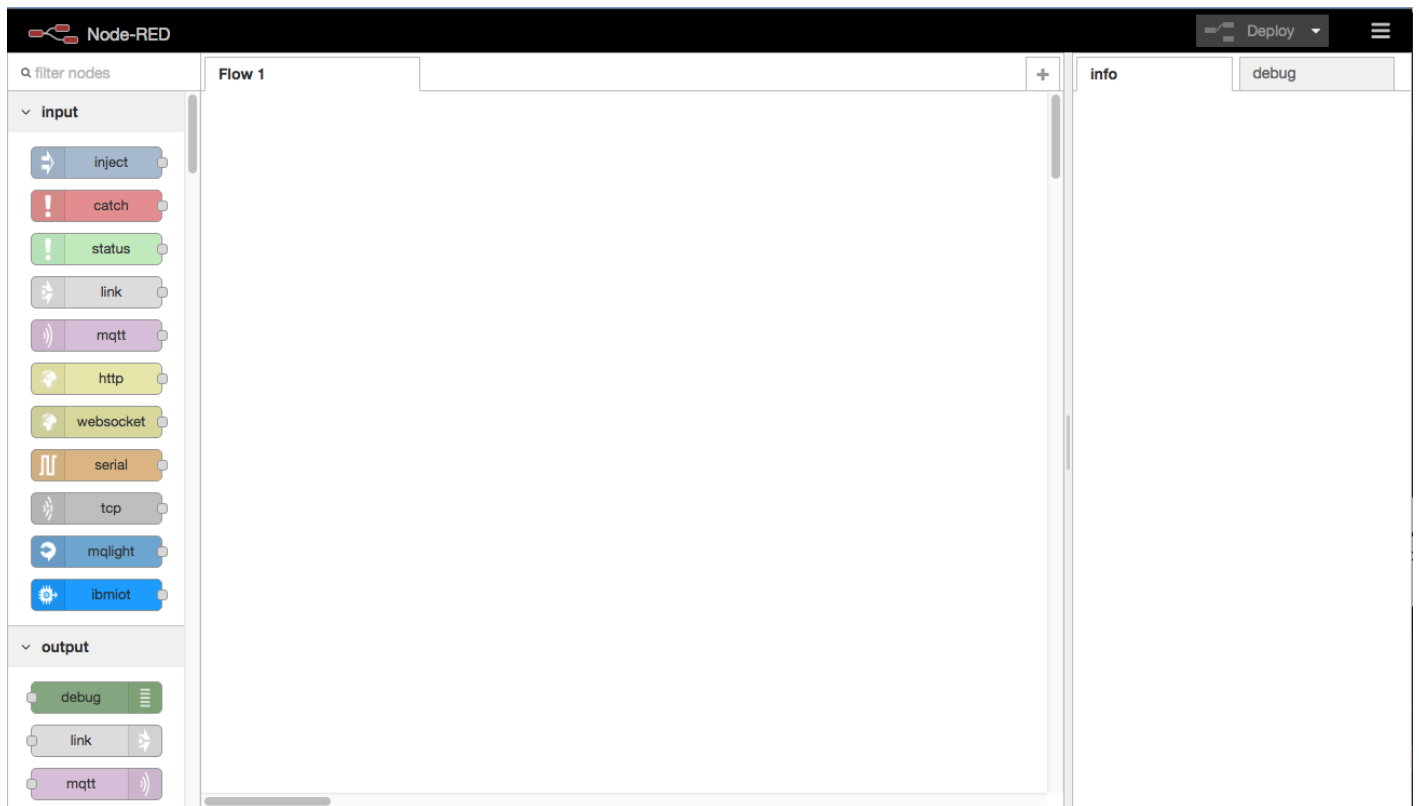
`https://<your app name>/mybluemix.net` . It would be wise at this stage to bookmark this page in the browser, as you'll be returning here during the demo.



(Optional) Preload the Twitter Sentiment analysis app

This is a step that you can certainly do live in front of the customer, but when time is short it's good to have this up and ready to go.

1. Click the red button labeled "Go to your Node-RED flow editor". This launches a browser, and loads a blank Node-RED workflow screen.



2. Open this link: <https://github.com/pahoran/Node-RED/blob/master/Twitter-flow>. It contains the source code for a complex Node-RED flow.
3. Highlight the entire contents of the file, top-to-bottom, and copy it to the clipboard (Ctrl-C, or right-click Copy).

GitHub repository view for **pahoran / Node-RED**. The interface shows the repository name, navigation tabs (Code, Issues, Pull requests, Projects, Wiki, Pulse, Graphs, Settings), and repository statistics (Unwatch, Star, Fork).

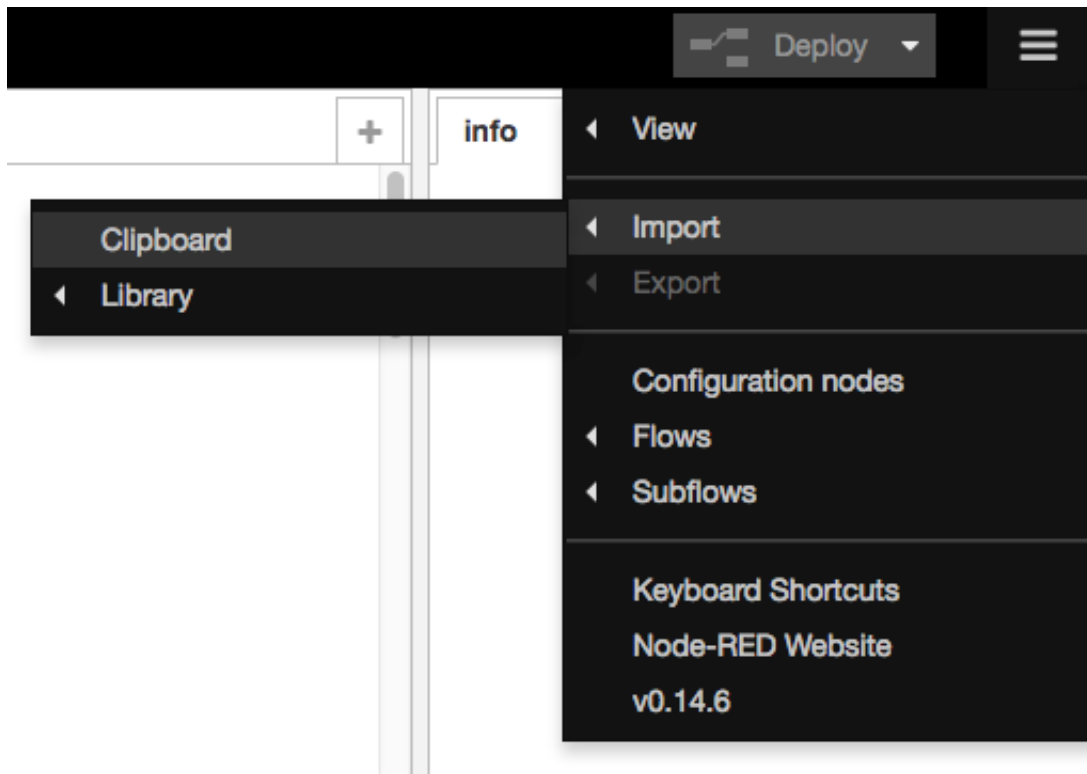
The selected file is **Node-RED / Twitter-flow**, created by **pahoran** 36 minutes ago. It has 1 contributor.

The file content is displayed in a code editor with 19 lines (18 sloc) and 12 KB. The code is a JSON configuration for a Node-RED flow, defining various nodes and their connections.

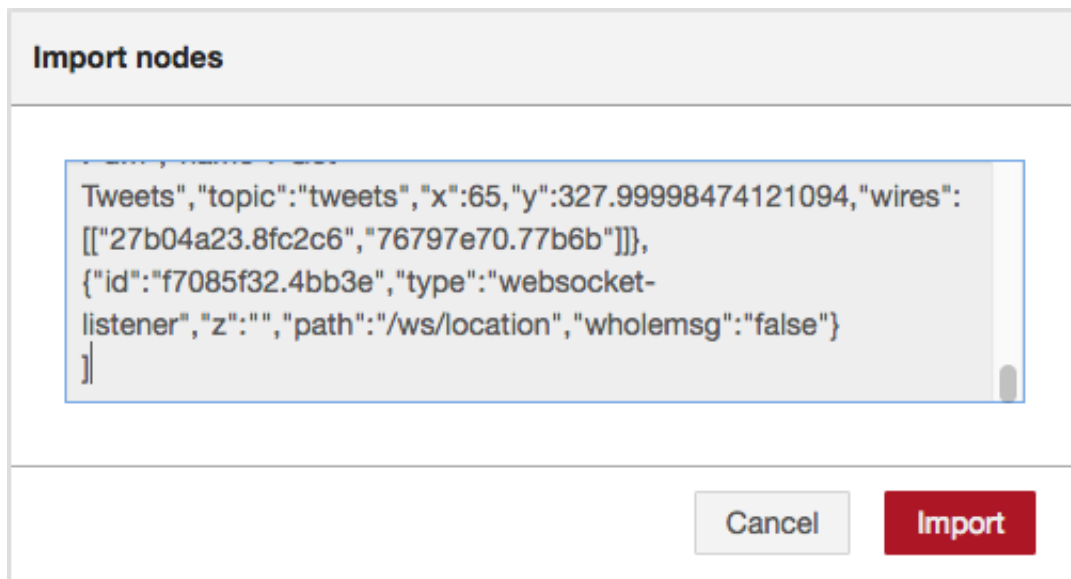
```
1 [{"id":"1081f98c.614586","type":"websocket out","z":"480a492b.dabc08","name":"","server":"f7085f32.4bb3e","client":"","x":865,"y":200}
2 [{"id":"fdf4a291.e972e","type":"websocket in","z":"480a492b.dabc08","name":"","server":"f7085f32.4bb3e","client":"","x":436,"y":200}
3 [{"id":"d150263b.ccb988","type":"function","z":"480a492b.dabc08","name":"Get Global context","func":"// The received message is
4 [{"id":"4db63cfe.ccb94","type":"template","z":"480a492b.dabc08","name":"HTML output","field":"","fieldType":"msg","syntax":"mus
5 [{"id":"6140b33d.043f1c","type":"http response","z":"480a492b.dabc08","name":"","x":829.9999732971191,"y":506.80005073547363,"wi
6 [{"id":"574be875.48f4f8","type":"http in","z":"480a492b.dabc08","name":"","url":"/map","method":"get","swaggerDoc":"","x":424.00
7 [{"id":"74e23af4.868554","type":"function","z":"480a492b.dabc08","name":"Load map json","func":"// Alchemy Method\n/var sentime
8 [{"id":"6b78716f.2b7db","type":"function","z":"480a492b.dabc08","name":"Set Payload=Tweet","func":"msg.payload = msg.tweet.text;
9 [{"id":"9df84343.ab248","type":"function","z":"480a492b.dabc08","name":"set _id","func":"msg._id = msg.tweet.id_str;\nreturn msg
10 [{"id":"a5fc06dc.af8a78","type":"debug","z":"480a492b.dabc08","name":"DEBUG","active":true,"console":"false","complete":"payload
11 [{"id":"89f26f05.422cd","type":"sentiment","z":"480a492b.dabc08","name":"","x":635.5,"y":269,"wires":[["74e23af4.868554","9df843
12 [{"id":"72c6211b.857a1","type":"debug","z":"480a492b.dabc08","name":"DEBUG","active":false,"console":"false","complete":"sentime
13 [{"id":"dab1f44c.1efd98","type":"cloudant out","z":"480a492b.dabc08","name":"Tweet DB","cloudant":"","database":"tweet_map","ser
14 [{"id":"81c71068.ce511","type":"debug","z":"480a492b.dabc08","name":"","active":true,"console":"false","complete":"true","x":843
15 [{"id":"ed79128.75b6af","type":"twitter in","z":"480a492b.dabc08","twitter":"","tags":"trump,clinton","user":"dm","name":"Get Tw
16 [{"id":"f7085f32.4bb3e","type":"websocket-listener","z":"","path":"/ws/location","wholemsg":"false"}
17 ]
18 ]
```

4. Back in the Node-RED editor, click the "hamburger" menu in the upper right-hand corner, and select

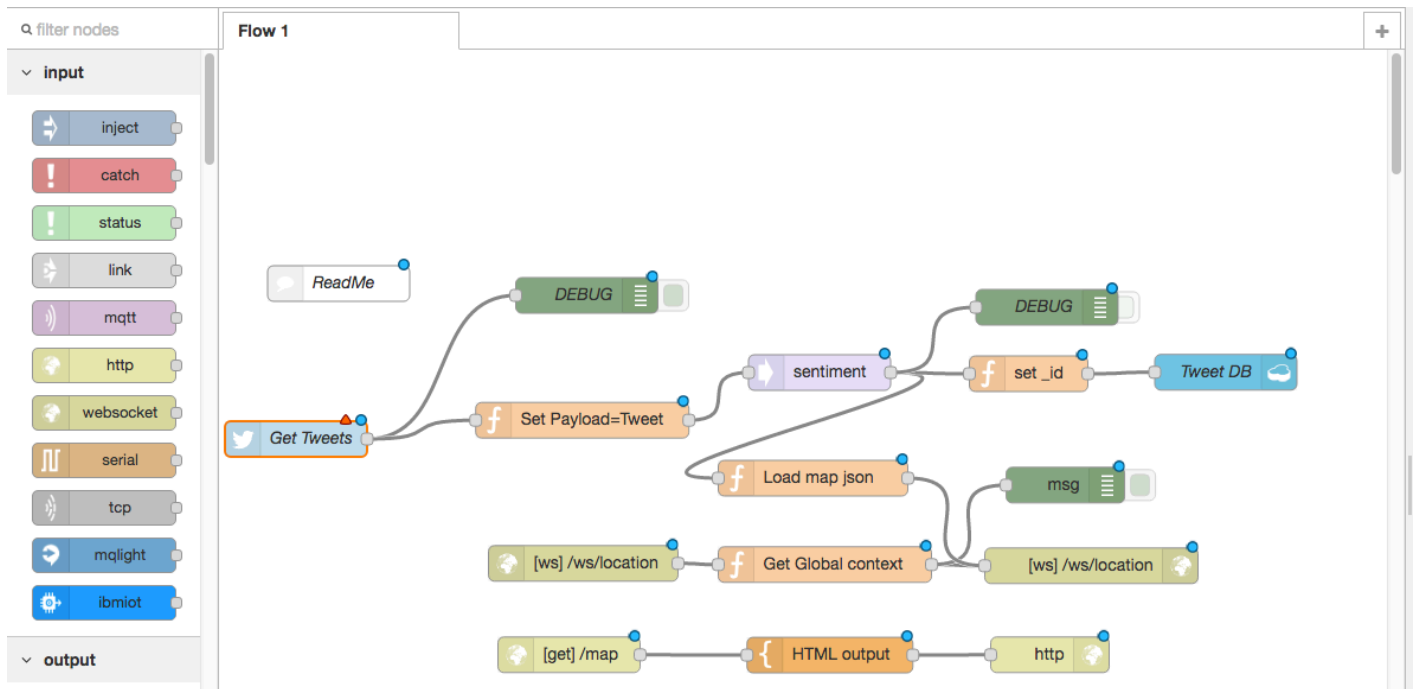
Import > Clipboard



5. That opens the **Import Nodes** dialog. Paste the clipboard contents in the big box in the middle, then click the red **Import** button to create the flow:



You should now see this screen. Click your mouse once to anchor the Flow to the canvas.



6. Double-click the blue `Get Tweets` node to open its Properties dialog.

Edit twitter in node

Cancel
Done

Twitter ID
Add new twitter-credentials...

Search
your direct messages

Name
Get Tweets

Tip: Use commas without spaces between multiple search terms.
Comma = OR, Space = AND.
The Twitter API WILL NOT deliver 100% of all tweets.
Tweets of who you follow will include their retweets and favourites.

Leave for blank to set using msg.payload.

- Click the "pen" icon next to the "Add new twitter-credentials" field. This will take you through a set of screens that will authorize Node-RED to access your Twitter account. Approve this access and return to the `Get Tweets` properties dialog.

- Set the `Search` field to `All Public Tweets`.
- The `for` search criteria field will default to "Trump, Clinton". Change that to anything you want to search for. Separate any distinct search values with commas.
- Click `Done` to return to the flow panel.
- Click the red `Deploy` button in the upper right-hand corner to deploy the revised flow into Bluemix.
- Select the `Debug` tabpage to see the tweets flowing past. This is an indication that the app is working properly. If you don't see Tweets here, your search phrase might not be bring back any data.
- Open a new browser window, and navigate to `http://<your Node-RED app>.mybluemix.net/map`. Bookmark this URL for easy access during the demo. You should see the map with tweet pins appearing dynamically.



The Developer version of the Twitter API is limited to a set number of API calls per minute. If you leave the app running in the background it will quickly reach its threshold, and stop delivering tweets. For a truly effective demo, don't start the Node-RED application in Bluemix until you're ready to show the map. You'll typically have about 15 minutes, depending on the popularity of the search phrase used.

- In the browser, go back to the Bluemix console and stop the Node-RED application. Go to the Console and find the row for the Node-RED application you created. On that row, select the icon that looks like three vertical dots. Choose `Stop App` from that menu to stop the Node-RED app. When it's time to actually deliver the demo, you'll return to this screen, select the same three-dot icon, and choose `Start App`.

The screenshot shows the IBM Bluemix console interface. At the top, there's a navigation bar with 'IBM Bluemix', 'Compute', 'Catalog', 'Docs', and a user profile for 'Paul Horan'. Below this, there's a tabbed interface with 'Cloud Foundry Applications', 'OpenWhisk', 'Containers', and 'Virtual Servers'. The 'Cloud Foundry Applications' tab is active, showing a table of applications. The table has columns: Name, Route, Memory (MB), Instances, Running, State, and Actions. Two applications are listed: 'pahmysql' and 'PHNodeRedDemo'. The 'PHNodeRedDemo' application is highlighted, and a context menu is open over it, showing options: 'Stop App', 'Restart App', 'Rename App', and 'Delete App'. The 'Stop App' option is selected.

Name	Route	Memory (MB)	Instances	Running	State	Actions
pahmysql	apiconnect-7b720c16-8e69-40dd-9fe9-6408...	256	1	0	Stopped	[Icon] [Three dots]
PHNodeRedDemo	PHNodeRedDemo.mybluemix.net	512	1	1	Running	[Stop App] [Restart App] [Rename App] [Delete App]

Delivering the Demo

Before you enter the room

1. Ensure you have an internet connection. If there's no "guest" Wi-Fi, test the connectivity from your phone's hotspot. If you're delivering the demo from inside a secure bunker, the chances of getting a reliable connection are slim, and you should gracefully eject now.
2. Open the Bluemix console, and login with your credentials. Ensure you're getting good response time across your wi-fi connection.

Start the Show

Initial presentation

Slideware, if any, goes here.

Demonstration

1. Show the Bluemix console. Spend as much time as you think is necessary to discuss the various service offerings in the catalog, but eventually land on the **Boilerplates** category.

IBM Bluemix Console

Catalog Docs 284 Paul Horan p.horan@us.ibm.com | dev

Catalog

All Categories

Boilerplates >

Compute

Network

Storage

Data & Analytics

Watson

Internet of Things

APIs

DevOps

Security


Application Services


Mobile


Integrate


Search


Filter


**ASP.NET Core Cloudant Starter**
Use the Cloudant NoSQL DB Service in an ASP.NET Core application.
IBM


**Internet of Things Platform Starter**
Get started with IBM Watson IoT platform using the Node-RED Node.js sample application. With the...
IBM


**IoT for Electronics Starter**
IoT for Electronics is an integrated end-to-end solution (made of multiple services and apps) th...
IBM


**Java Cloudant Web Starter**
Use the Cloudant NoSQL DB service with the "Liberty for Java™" runtime.
IBM


**Java Workload Scheduler Web Starter**
This application demonstrates how to use the Workload Scheduler service, with the "Liberty for Java™" runtime.
IBM


**LoopBack Starter**
This is a sample StrongLoop LoopBack Node.js application, powered by the open source LoopBack...
IBM


**MobileFirst Services Starter**
Start building your next mobile app with mobile services on Bluemix.
IBM


**Node.js Cloudant DB Web Starter**
Use the Cloudant NoSQL DB service with the "SDK for Node.js™" runtime.
IBM


**Personality Insights Java Web Starter**
A simple Java app that uses the Personality Insights service to analyze text to derive personality traits.
IBM


**Personality Insights Node.js Web Starter**
A simple Node.js app that uses Personality Insights to analyze text to derive personality traits.
IBM

**StrongLoop Arc**
This application is the StrongLoop Arc graphical UI, which includes tools for building, profiling and...
IBM

**Mendix Rapid Apps**
Model driven rapid app platform that allow users to build, integrate and deploy web and mobile
Community

**Node-RED Starter**
This application demonstrates how to run the Node-RED open-source project within IBM Bluemix.
IBM

**Python Flask**
A simple Python Flask application that will get you up and running quickly.
IBM

**Ruby Sinatra**
Develop a Ruby web application using the Sinatra framework.
IBM

Script:

Boilerplates are pre-configured groupings of services that give developers a head-start on any project, whether that's a Java web project, an IoT project, or a Mobile project. There's even a boilerplate for Microsoft's stack and ASP.NET development. Let's take a look at one boilerplate in particular, IBM's Node-RED starter.

2. Click on the Node-RED Starter tile, and bring up that screen.

Node-RED Starter

This application demonstrates how to run the Node-RED open-source project within IBM Bluemix.

[Community](#)

[View Docs](#)

VERSION 0.5.0
TYPE [Boilerplate](#)
REGION [US South](#)

App name:


Host name:

Domain:


Selected Plan:

SDK for Node.js™

Cloudant NoSQL DB



SDK for Node.js™



Cloudant NoSQL DB

Develop, deploy, and scale server-side JavaScript® apps with ease. The IBM SDK for Node.js™ provides enhanced performance, security, and serviceability.

Pricing Plans

Monthly prices shown are for country or region: [United States](#)

Need Help?
[Contact Bluemix Sales](#)

Estimate Monthly Cost
[Cost Calculator](#)

[Create](#)

Script:

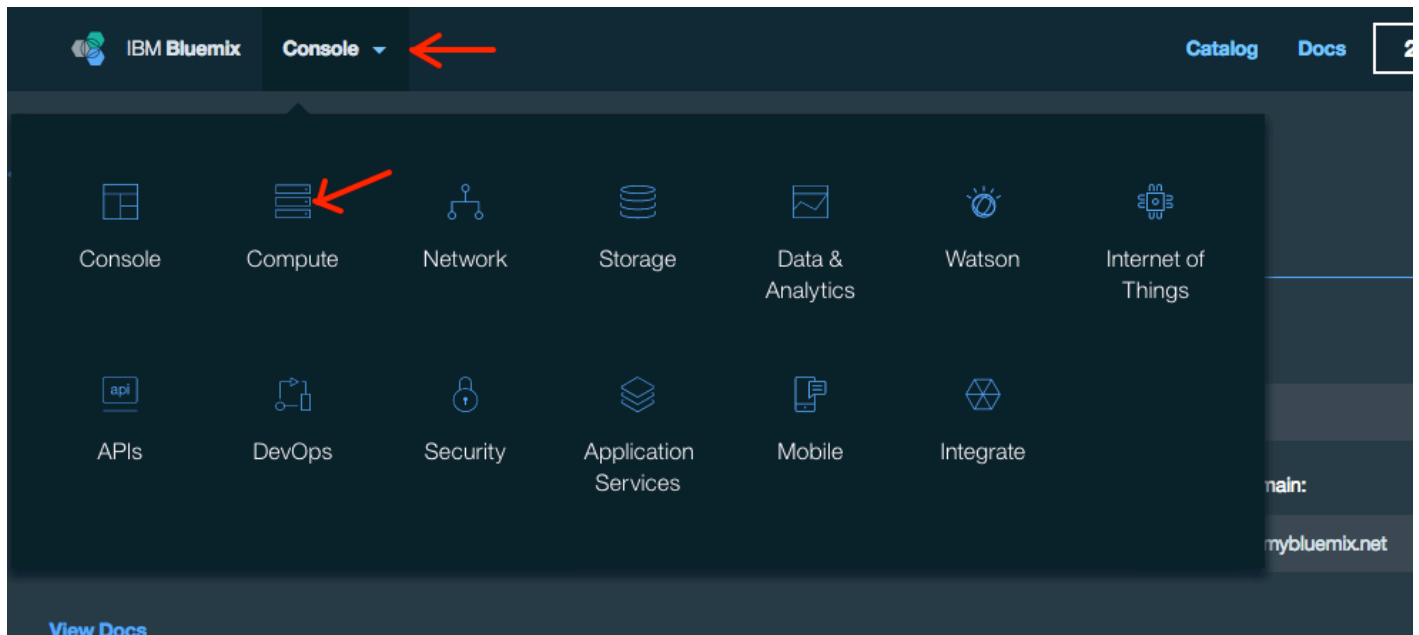
The Node-RED Starter boilerplate actually consists of two separate services that have been pre-integrated so that they work together seamlessly. The first is a nodeJS service, which is a highly scalable open-source runtime platform for Javascript applications. The second is the Cloudant NoSQL database service, which provides persistent storage and retrieval of JSON documents. Node-RED is the open source graphical editor for wiring together APIs and services into a web or IoT application.

IBM is a huge proponent of Open Source, and currently has two Board seats on the NodeJS Foundation. IBM is the only vendor that can offer support plans for open source software such as nodeJS.

All I need to do to create this platform is give the service a unique name, and click the **Create** button. That will spin up a new nodeJS runtime server instance, and add the NodeRED application, and publish the public URLs out to the web. Since Bluemix uses consumption-based pricing, I'll only be charged for the compute cycles I actually use. There's a handy Pricing Calculator link at the bottom of the screen that can be used to estimate monthly charges.

In the interest of time, I've already got that service set up. Let me start that running now.

- Do NOT click the **Create** button now. Instead, click the **Console** button near the top-left of the screen. That drops down a graphical menu, as shown below.



Click the **Compute** button. That takes you to a window that shows the Node-RED service that you created during the pre-work section. At this point, its status should be **Stopped**. Click the icon at the right-hand side that looks like three vertical dots, and select **Start App**. This should only take about 10-20 seconds. Explain that Bluemix is spinning up the NodeJS server instance, and publishing the unique URL.

 The image shows the 'Cloud Foundry Applications' page in the Bluemix console. It has tabs for 'Cloud Foundry Applications', 'OpenWhisk', 'Containers', and 'Virtual Servers'. Below the tabs, it shows '0 B / 8 GB Used'. A table lists the applications:

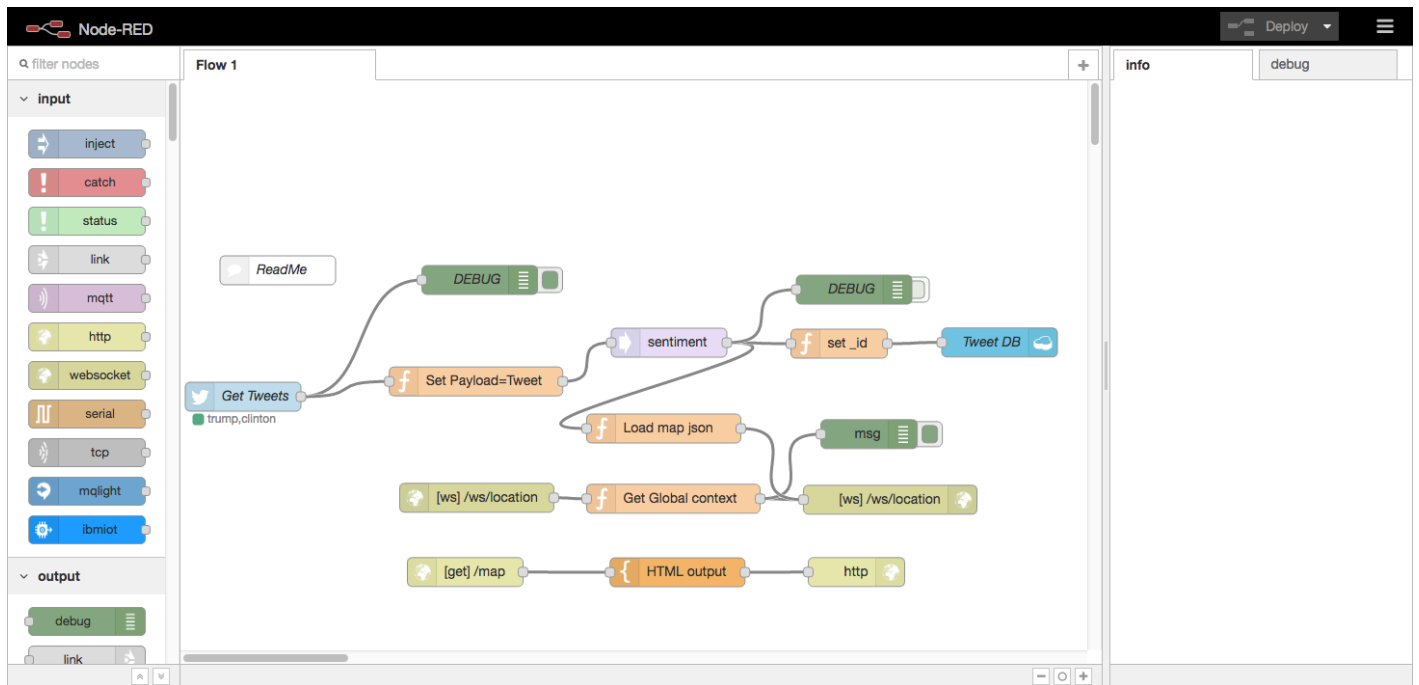
Name	Route	Memory (MB)	Instances	Running	State	Actions
pahmysql	apiconnect-7b720c16-8e69-40dd-9fe9-8...	256	1	0	Stopped	[Icon] [More]
PHNodeRedDemo	PHNodeRedDemo.mybluemix.net	512	1	0	Stopped	[Icon] [More]

4. When the app is finally running, click the blue URL link in the **Route** column.

 This is a partial view of the table from the previous screenshot, showing the 'PHNodeRedDemo' application. In this state, the 'Running' column shows '1' and the 'State' column shows 'Running' with a green dot icon.

PHNodeRedDemo	PHNodeRedDemo.mybluemix.net	512	1	1	Running	[Refresh] [Icon] [More]
---------------	---	-----	---	---	---------	-------------------------

5. This loads the main Node-RED landing page. Just click the big red button labeled **Go to your Node-RED flow editor**. This will load the Node-RED editor into the browser screen. It should also have loaded the complex flow that was imported in the pre-work.



Script:

We're now looking at the Node-RED flow editor, a graphical drag-and-drop programming environment for the creation of web-based apps, internet-of-things apps, and APIs.

Down the left-hand side is the palette of nodes that represent the different inputs, outputs, and processing options available to be wired together. Inputs can be standard HTTP calls, serial port and IOT devices, or messages from an MQ bus. There are a number of possible outputs as well, including HTTP responses, SMS text messages, and even push notifications to smart phones.

6. Start scrolling down through the palette.

Script:

There are different flow of control and processing functions available. Social network integration, including email and Twitter. A number of different databases are represented here in the Storage section. Scrolling down farther, we get into the really interesting functions - the Watson cognitive APIs. These are now available to you through this simple drag-and-drop programming interface. And the open source community is contributing new APIs and nodes on a regular basis.

Let's look at the complex flow we've got here as the sample app.

Point to the light blue node "Get Tweets"

This starts by retrieving tweets from the public Twitter stream, searching for a specific phrase. Right now, we're searching for tweets containing either "Clinton" or "Trump", but we can change that to anything.

At this point, you could click the Get Tweets node, and change the phrase to anything. "Zika", "Climate Change", "World Series"... If you do make a change here, you must click the red **Deploy** button in the upper right-hand corner to push that change up to Bluemix.

Point to the light purple node labeled "sentiment".

It then passes them into the Watson sentiment analysis API, and assigns either a positive, negative, or neutral "sentiment" score.

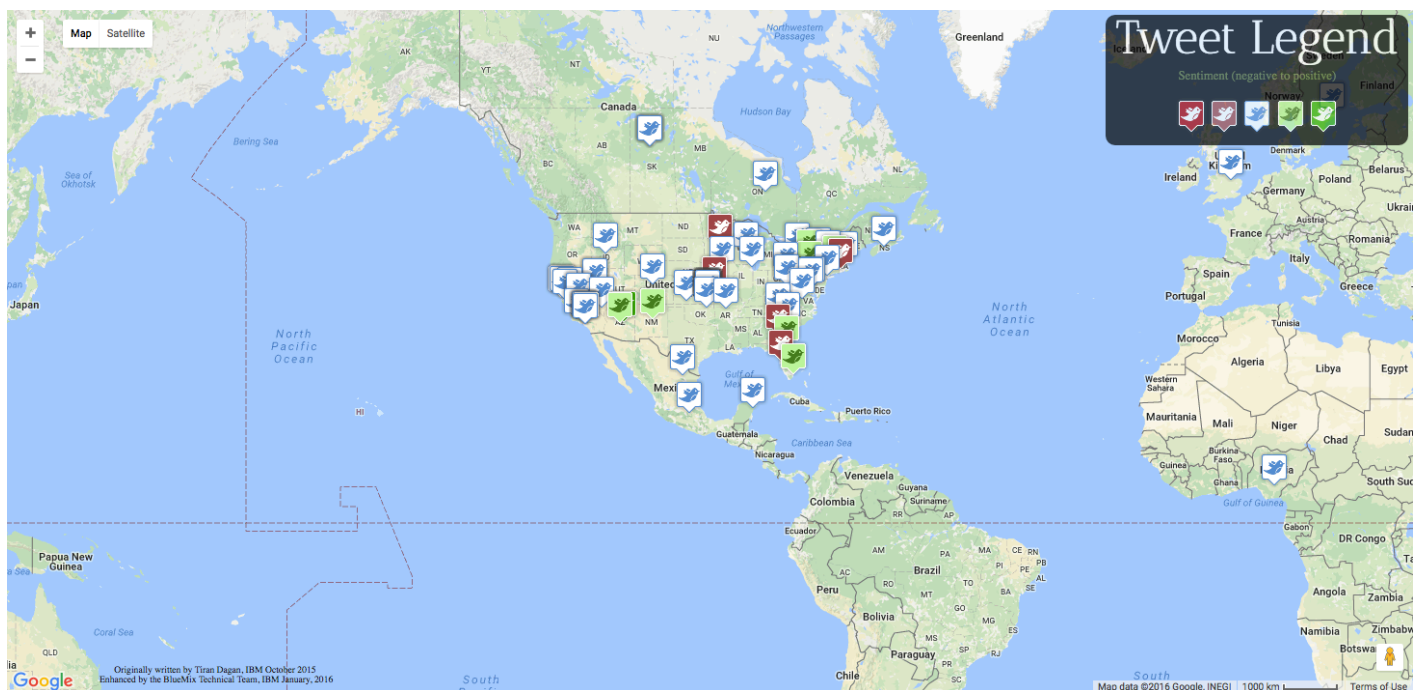
Point to the dark blue node labeled "Tweet DB".

This data is then persisted into the Cloudant database that was provisioned as part of the boilerplate.

Point to the node labeled [get] /map at the bottom

Down here at the bottom we've defined an HTTP request endpoint named "map", and this is where we'll retrieve tweet data from the Cloudant database and plot it on a map. Let's see that in action.

- Now open the link for the map. If you bookmarked it during the pre-work, just open that bookmark. If not, it should be <http://yourNodeRedapp.mybluemix.net/map>.



- Click on various green pins and red pins to pull up the underlying tweet data. This shows that the sentiment analysis is working.



The /map URL is just "http" - it's not "https". If you get that wrong, the map will start to render,

but no tweets will show up.

End of DEMO

Shutting Down

It's important to STOP your Node-RED service, or it will continue running and accumulating charges. If it's not convenient to do this during or after the meeting, make sure you get to it at your next opportunity.

1. Go back into the Bluemix console screen. It should show your Node-RED service as **Running**.
2. Click the three vertical dots on the left, and choose **Stop App**.