

Lab 1 - Introduction to IBM API Connect

In this lab, you'll start from scratch to install Node.js and the components of the API Connect Developer toolkit. Once you have the toolkit installed, you'll get a chance to use the APIC command line interface for creating LoopBack applications, the intuitive Web-based user interface, and explore the various aspects associated with solution's configuration of RESTful based services as well as their operation.

Lab 1 - Objective

In the following lab, you will learn:

- How to install Node.js on to your machine
 - How to enable API Connect on your Bluemix account
 - How to install the APIC Developer Toolkit
 - How to create a simple LoopBack application
 - How to create a Representational State Transfer (REST) API definition
 - How to test a REST API
 - How to publish an API to the Developer Portal
-

Lab 1 - Case Study used in this tutorial

In this tutorial, you will be starting from scratch to set up your development environment on your local machine, and then you will test the components of the toolkit by creating the default Hello World LoopBack application. We will look at the created artifacts to get a better understanding of what is created for you. The hello-world application is a simple RESTful service that holds a set of "notes" in memory. In this lab we may be creating a new RESTful service and API, but in future labs you'll be creating APIs with existing services.

Lab 1 - Before you begin

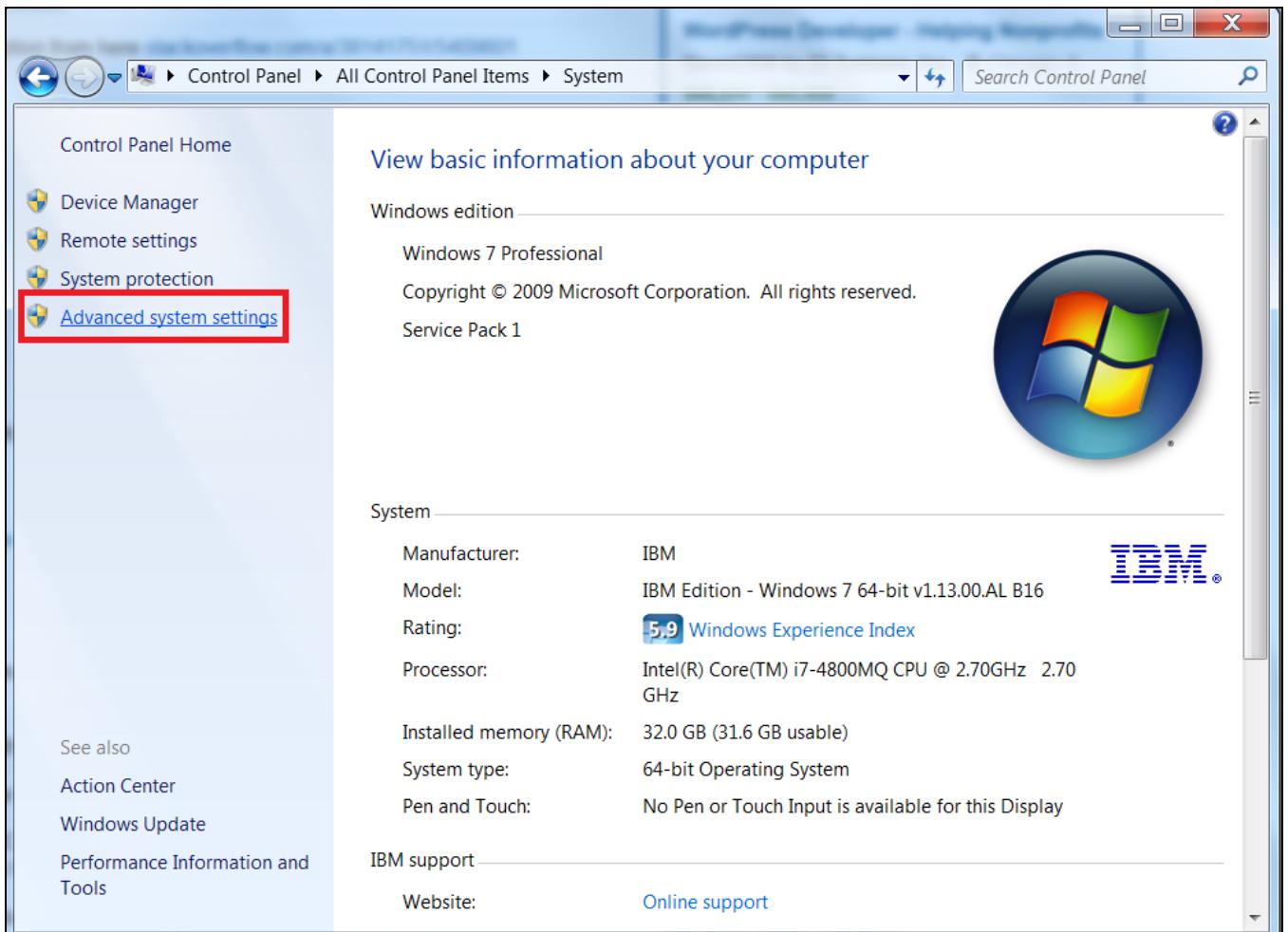
For this lab, you will be starting with your local image and installing node.js and the developer toolkit. After that, if you do not already have a Bluemix account, you will be creating one for you and enabling the API Connect Essentials service on your account. The instructions you will follow will vary by the host operating system you

are using. The three host OS's supported by this are [Windows](#), [Linux - Intel](#) and [Linux - Mac](#). Skip to the section for your appropriate operating system.

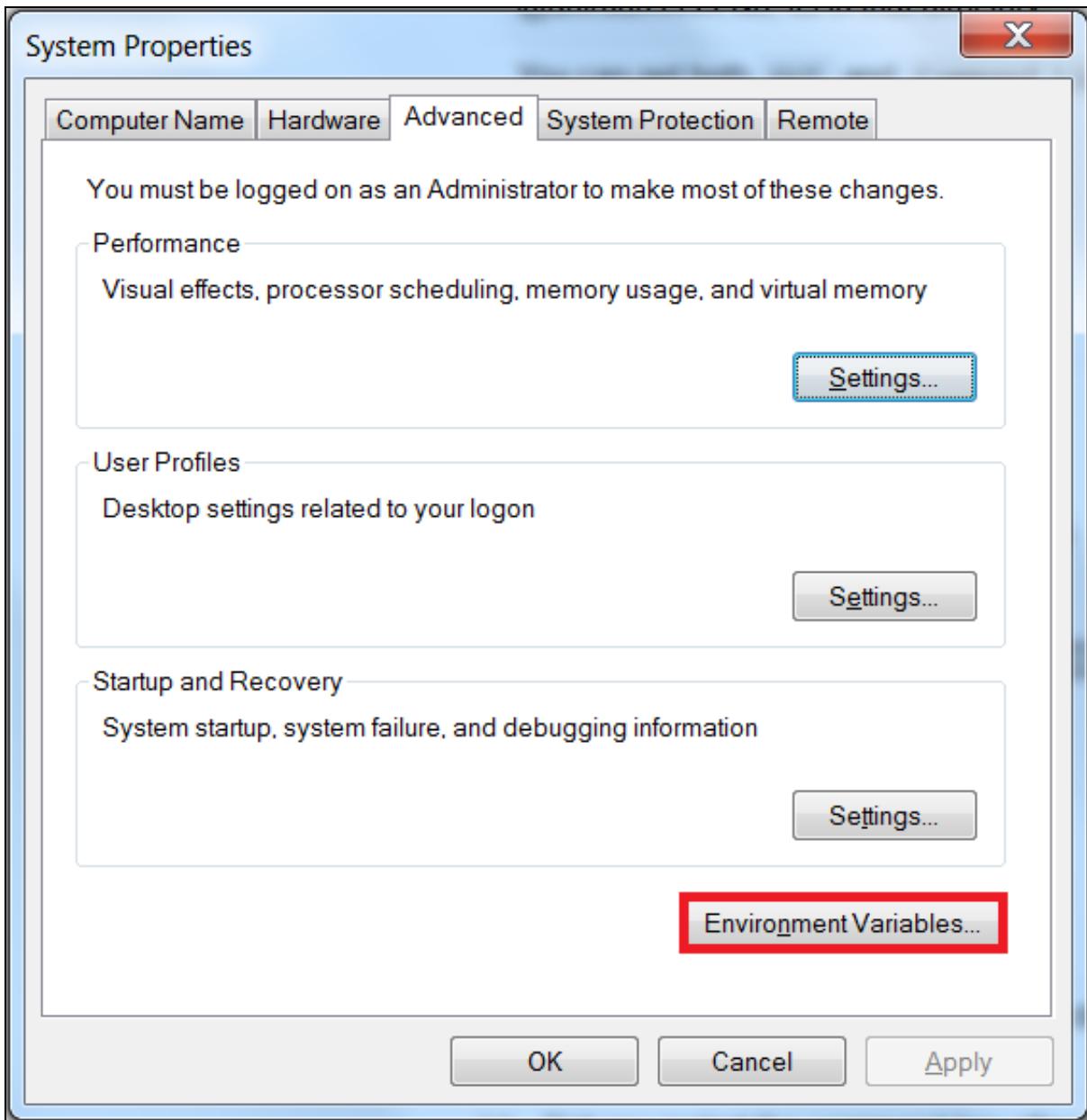
Lab 1 - Step by Step Lab Instructions

1.0a - Install Node.js on Windows

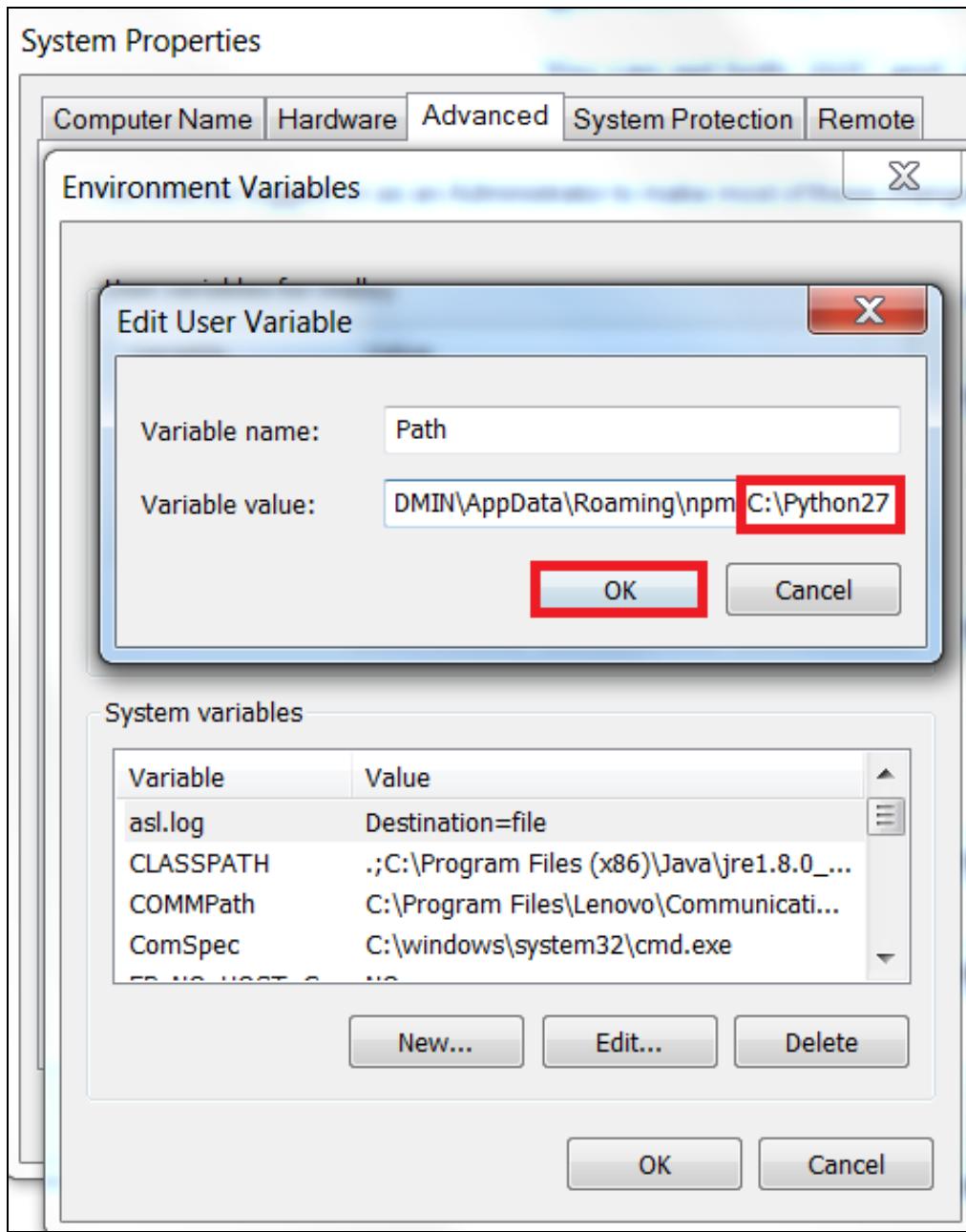
1. If you don't have node already installed on your machine, then please proceed. Otherwise, move on to section 1.1
2. Installing Node on Windows based machines (Tested on Windows 7):
3. The following are pre-requisites that need to be installed before installing node:
 1. Install Python
 1. version 2.7 is required. You can download it here:
<https://www.python.org/download/releases/2.7/>
 2. Note the location where it is installed. You will need to add this location to your path later on.
 2. Install .Net Frameworks SDK 2.0
(<https://www.microsoft.com/en-us/download/confirmation.aspx?id=19988>)
 3. Install Visual Studio Express
(<https://www.microsoft.com/en-us/download/confirmation.aspx?id=44914>)
 4. Add Python to the System Path in Windows 7 to the System Path.
 1. Go to [Control Panel](#) -> [System](#) -> [Advanced System Settings](#)



1. Click **Environment Variables**.



1. Click **Edit** and append `;C:\python27` (or wherever you installed python) to the Path variable.



4. Install Node.js

1. Go to this link to install Node <https://nodejs.org/en/download/>
2. Select the "LTS version - Recommended for most users". Version at the top should depict v4.4.7 - includes npm 2.15.8.

The screenshot shows the Node.js website at <https://nodejs.org/en/download/>. The page has a dark header with the Node.js logo and navigation links for HOME, ABOUT, DOWNLOADS, DOCS, FOUNDATION, GET INVOLVED, SECURITY, and NEWS. Below the header is a section titled "Downloads" with a sub-section for the "Current version: v4.4.7 (includes npm 2.15.8)". A callout box highlights the "LTS" section, which is described as "Recommended For Most Users" and "Current Latest Features". This section includes links for "Windows Installer (.msi)" and "Windows Binary (.exe)". To the right, there are sections for "Windows Installer" (with link to "node-v4.4.7-x64.msi"), "Macintosh Installer" (with link to "node-v4.4.7.pkg"), and "Source Code" (with link to "node-v4.4.7.tar.gz"). Below these are tables for 32-bit and 64-bit versions of the Windows installer.

	32-bit	64-bit
Windows Installer (.msi)	Download	Download
Windows Binary (.exe)	Download	Download

1. Select the Windows Installer .msi binary and then follow the prompts to install on your machine
5. Install API Connect on to your machine by starting up a command line window as Administrator and issue this command - `npm install -g apiconnect` .
6. Once complete start up a new terminal window - enter in `apic -v` . It should show you output that looks similar to below (the version numbers might be different).

```
C:\windows\system32\cmd.exe
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.

C:\Users\IBM_ADMIN>apic -v
apiconnect : v2.0.18
apiconnect-cli-apis 2.0.1
apiconnect-cli-apps 2.0.0
apiconnect-cli-auth 2.0.5
apiconnect-cli-catalogs 2.0.0
apiconnect-cli-config 2.0.1
apiconnect-cli-create 2.0.0
apiconnect-cli-drafts 2.0.0
apiconnect-cli-edit 2.0.0
apiconnect-cli-loopback 2.0.0
apiconnect-cli-orgs 2.0.0
apiconnect-cli-pm 2.0.0
    microgateway 1.1.3
        flow-engine 1.0.3
apiconnect-cli-policies 1.0.2
apiconnect-cli-products 2.0.0
apiconnect-cli-validate 2.0.1
plugins 2.0.0

C:\Users\IBM_ADMIN>
```

7. You are now ready to move on to section 1.1 below.

1.0b - Install Node.js on Linux

- ## 1. Install the essentials to run node.js

```
sudo apt-get install build-essential libssl-dev curl git-core
```

2. Install Node Version Manager aka "nvm" by issuing this command here:

```
curl https://raw.githubusercontent.com/creationix/nvm/v0.31.2/install.sh | bash
```



When you install `nvm` it will also install `npm` which is the node package manager. npm is used to install node.js based software packages including the API Connect Developer Toolkit

3. close and restart your terminal as indicated in the terminal window or run this command

```
source ~/.profile
```

4. Run this command to confirm your version `nvm --version`
5. This command will provide documentation on nvm if you wish to learn more about it `nvm help`. NVM is exceptionally useful as it will allow you to support multiple installations of node.js on your machine, and select the version you want to use. For our lab today, we only need the one version as indicated in the steps below.
6. To check and see which versions of node are available, issue this command `nvm ls-remote`. It will provide a very long list. For API Connect, we will use a specific version of node. That is version 4.4.7
7. To install node 4.4.7 issue this command `nvm install 4.4.7`

```
student@ubuntu:~$ nvm install 4.4.7
Downloading https://nodejs.org/dist/v4.4.7/node-v4.4.7-linux-x64.tar.xz...
Now using node v4.4.7 (npm v2.15.8)
```

8. Install API Connect by issuing this command `npm install -g apiconnect` it will take several minutes to complete.
9. Run this command to ensure all is set up properly. `apic -v` You might be asked to accept the license. Select the default `yes` and hit enter. You are now ready to move on to section 1.1 below.

1.0c - Install Node.js on Mac

1. Check to see if node is already installed.

```
node -v
```

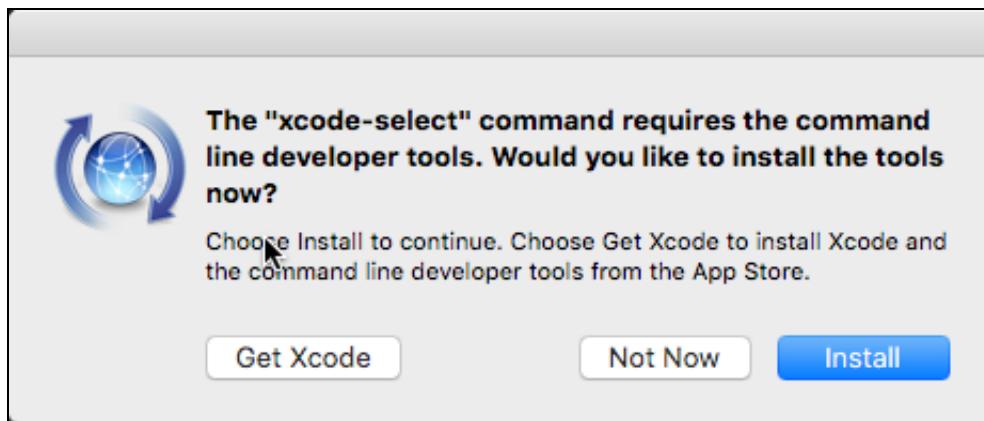
2. Check the version, if it is version 4.4.7 then you can move on to section 1.1 - Create your Bluemix account. However if you would prefer to uninstall and reinstall node using Node Version Manager or if node was not found then please continue.
3. Remove any existing versions of node. You will only need to complete these steps if node current exists. To remove node completely from your Mac OSX environment run each of the following commands individually via terminal. Open a terminal session and type each of the commands below individually.

```
sudo rm /usr/local/bin/npm  
sudo rm /usr/local/share/man/man1/node.1  
sudo rm /usr/local/lib/dtrace/node.d  
sudo rm -rf ~/.npm  
sudo rm -rf ~/.node-gyp  
sudo rm /opt/local/bin/node  
sudo rm /opt/local/include/node  
sudo rm -rf /opt/local/lib/node_modules  
sudo rm -rf /usr/local/include/node/
```

4. Install Xcode Command Line Tools. Open a terminal and type

```
xcode-select --install
```

5. This will open a dialog informing you that the command line developer tools are required for install. Press the Install button.



6. Agree to the license

License Agreement

Command Line Tools License Agreement

APPLE INC. MAC SDK AND XCODE AGREEMENT

PLEASE READ THIS MAC SDK AND XCODE AGREEMENT ("LICENSE") CAREFULLY BEFORE USING THE DEVELOPER SOFTWARE (DEFINED BELOW). BY USING THE DEVELOPER SOFTWARE, YOU ARE AGREEING TO BE BOUND BY THE TERMS OF THIS LICENSE. IF YOU ARE ACCESSING THE DEVELOPER SOFTWARE ELECTRONICALLY, SIGNIFY YOUR AGREEMENT TO BE BOUND BY THE TERMS OF THIS LICENSE BY CLICKING THE "AGREE" BUTTON. IF YOU DO NOT AGREE TO THE TERMS OF THIS LICENSE, DO NOT USE THE DEVELOPER SOFTWARE AND CLICK "DISAGREE".

IMPORTANT NOTE: To the extent that this software may be used to reproduce materials, it is licensed to you only for reproduction of non-copyrighted materials, materials in which you own the copyright, or materials you are authorized or legally permitted to reproduce. If you are uncertain about your right to copy any material, you should contact your legal advisor.

1. General.

A. The Apple software, tools, utilities, sample or example code, documentation, interfaces, content, data, and other materials accompanying this License, whether on disk, print or electronic documentation, in read only memory, or any other media or in any other form, (collectively, the "Developer Software") are licensed, not sold, to you by Apple Inc. ("Apple") for use only under the terms of this License. Apple and/or Apple's licensors retain ownership of the Developer Software itself and reserve all rights not expressly granted to you. The terms of this License will govern any software upgrades provided by Apple that replace and/or supplement the original Developer Software, unless such upgrade is accompanied by a separate license in which case the terms of that license will govern.

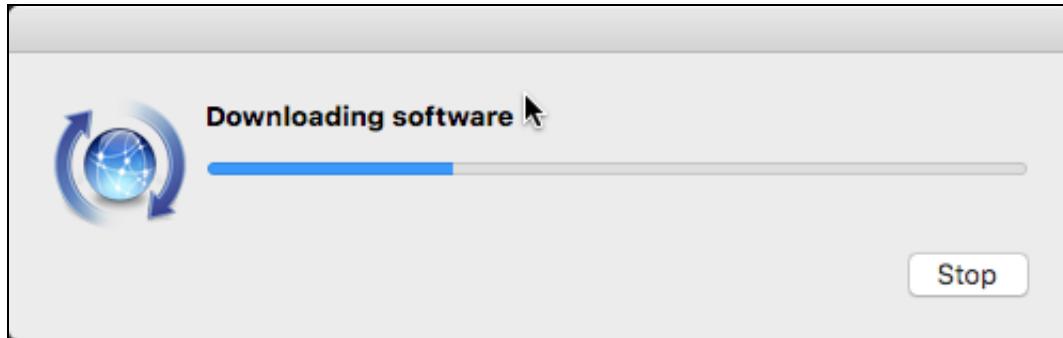
B. Title and intellectual property rights in and to any content displayed by or accessed through the Developer Software belongs to the respective content owner. Such content may be protected by copyright or other intellectual property laws and treaties, and may be subject to terms of use of the third party providing such content. This License does not grant you any rights to use such content nor does it guarantee that such content will continue to be available to you.

2. Permitted License Uses and Restrictions.

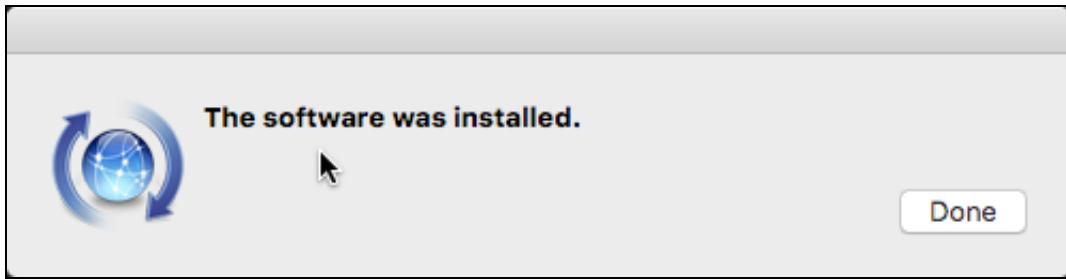
A. License. Subject to the terms and conditions of this License, you are granted a limited, non-exclusive license to use the Developer Software on Apple-branded computers to develop and test application and other software. You may make only as many

Disagree Agree

7. The software install will begin



8. The install is complete when you see the "The software was installed" message.



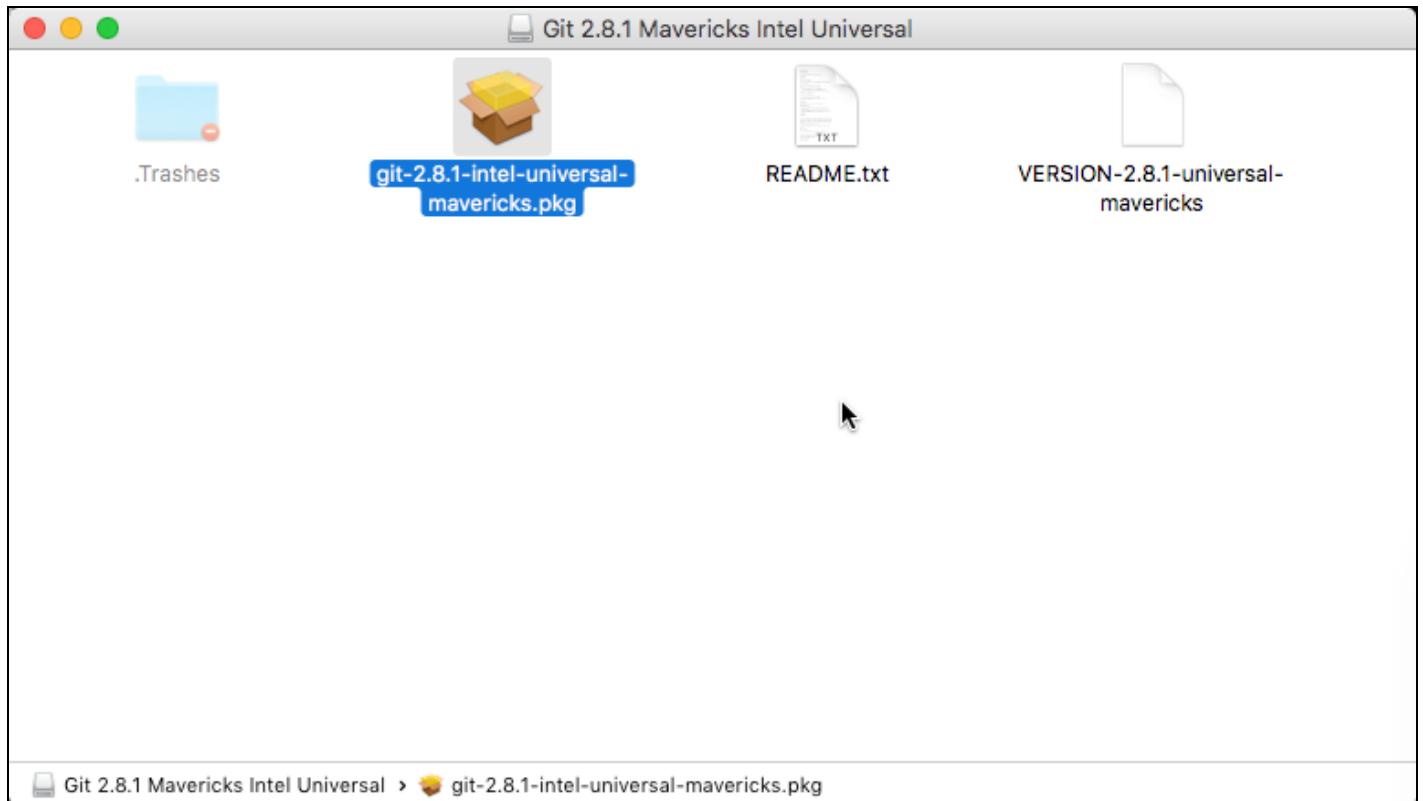
9. Next install git. You may already have a version, but if you want to have the latest version you will need to download the install from <https://git-scm.com/download/mac>
10. Once on the install page the download should start immediately. However if it does not start immediately, click the "click here to download manually" link.

The entire [Pro Git book](#) written by Scott Chacon and Ben Straub is available to [read online for free](#). Dead tree versions are available on [Amazon.com](#).

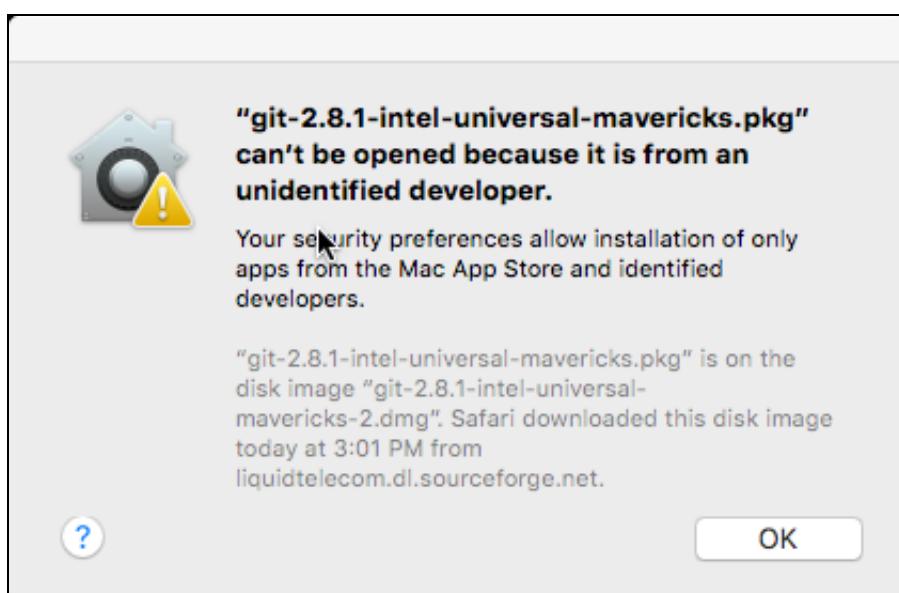
<https://git-scm.com>

11. Once the download has completed, you should have a file named similar to git-2.8.1-intel-universal-mavericks.dmg. Double-click to unpack the disk image.
12. Once the disk image is unpacked and open, double-click the git-2.8.1-intel-universal-mavericks.pkg file.

Note that if you downloaded a different version the file name will be reflected accordingly.



13. You may get a dialog box with a message stating that the file can't be opened because it is from an unidentified developer. Click the OK button. Open your System Preferences and select the Security & Privacy component. In the section called "Allow apps downloaded from:" you will notice a message that states "git-2.8.1-..ericks.pkg" was blocked from opening because it is not from an identified developer.', press the Open Anyway button. This will open the installer for git.



Security & Privacy

General FileVault Firewall Privacy

A login password has been set for this user [Change Password...](#)

Require password after sleep or screen saver begins

Show a message when the screen is locked [Set Lock Message...](#)

Allow apps downloaded from:

Mac App Store

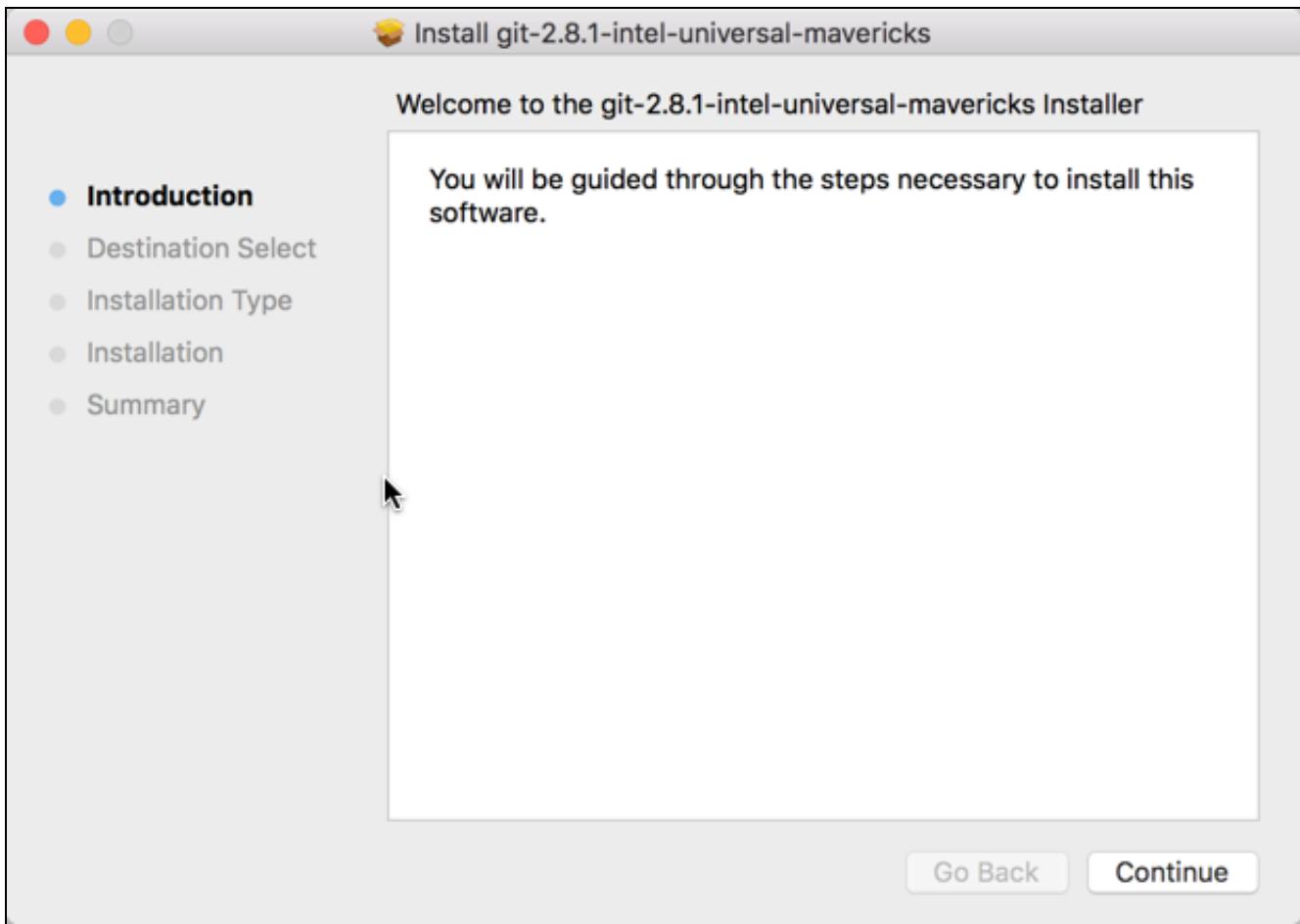
Mac App Store and identified developers

Anywhere

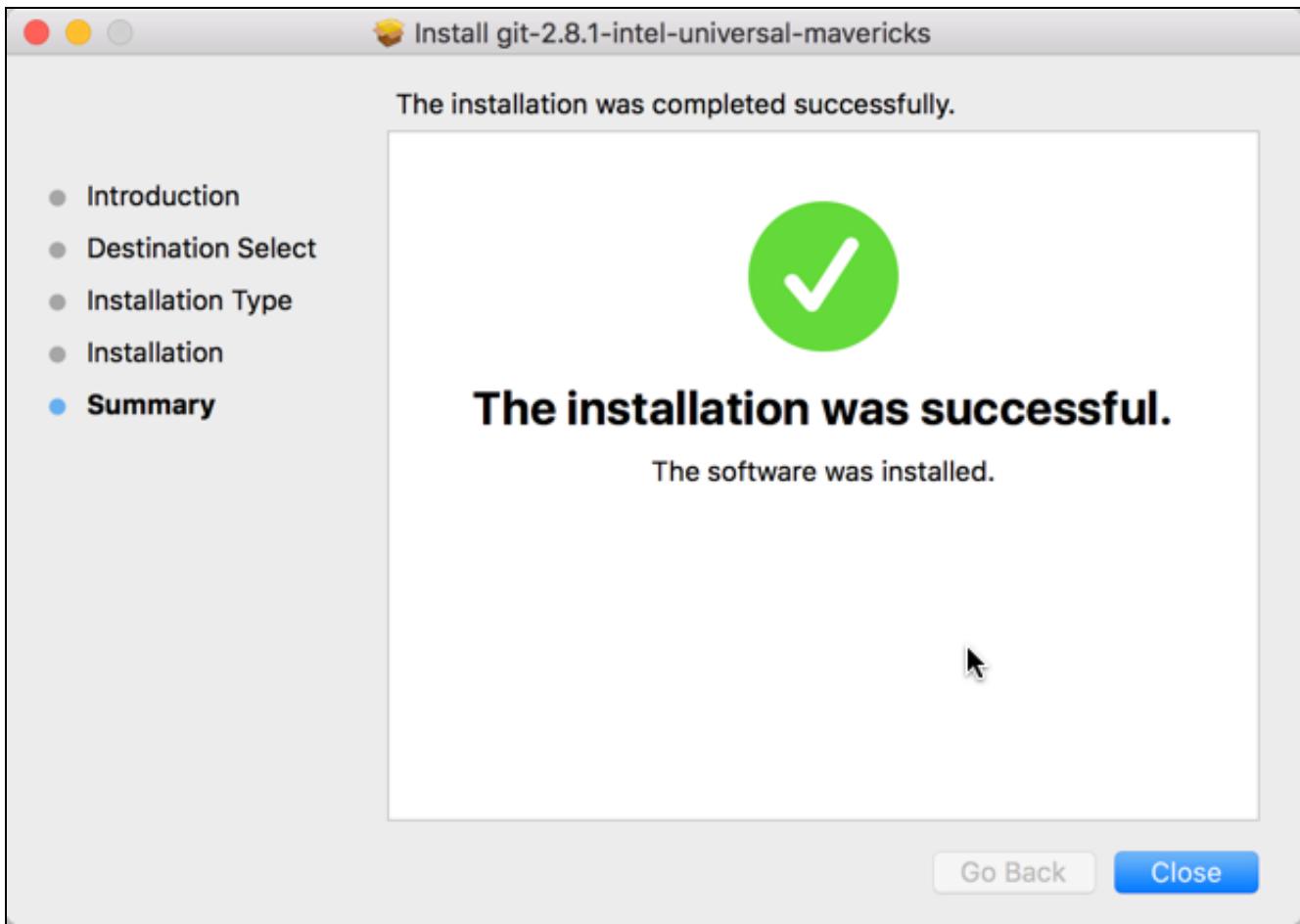
"git-2.8.1-...ericks.pkg" was blocked from opening because it is not from an identified developer. [Open Anyway](#)

 Click the lock to make changes.

[Advanced...](#) 



14. Follow the wizard to install. The installation is complete when you see a message stating that "The installation was successful."



15. Install Node Version Manager aka "nvm" by opening a terminal session and typing the following commands:

```
git clone git://github.com/creationix/nvm.git ~/.nvm
printf "\n\n# NVM\nif [ -s ~/.nvm/nvm.sh ]; then\n\tNVM_DIR=~/.nvm\n\tsource ~/.nvm
NVM_DIR=~/.nvm
source ~/.nvm/nvm.sh
nvm install 4.4.7
nvm alias default node
nvm use node
```

Note: You may want to update your .bash file so you do not have to source the shell script everytime. To do that, add the following to your .bash file located in your user home directory:

```
export NVM_PATH=~/.nvm
source ~/.nvm/nvm.sh
```

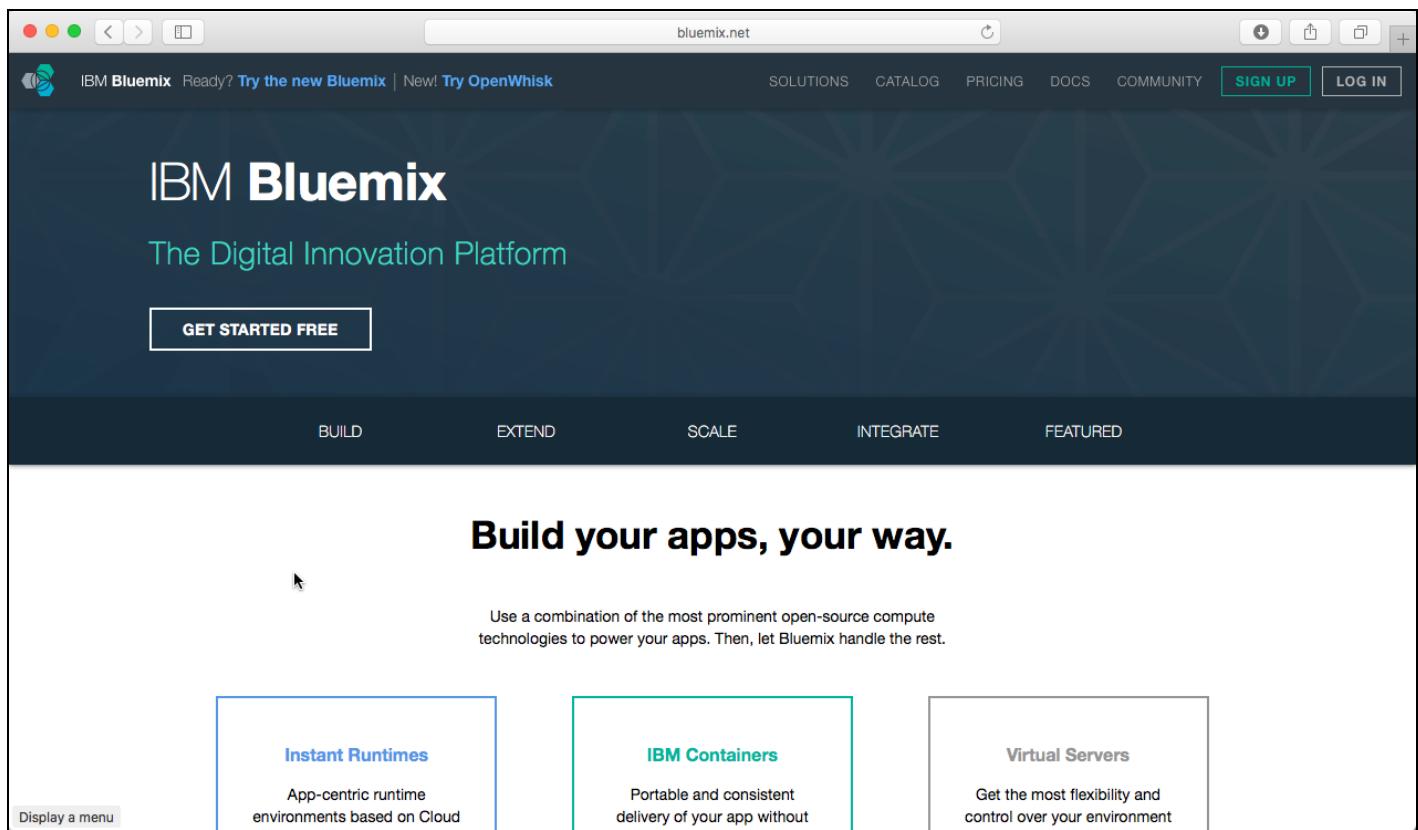
16. Install API Connect by issuing this command `npm install -g apiconnect` it will take several

minutes to complete

17. Run this command to ensure all is set up properly. `apic -v` You might be asked to accept the license. Select the default `yes` and hit enter.

1.1 - Create your Bluemix account

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



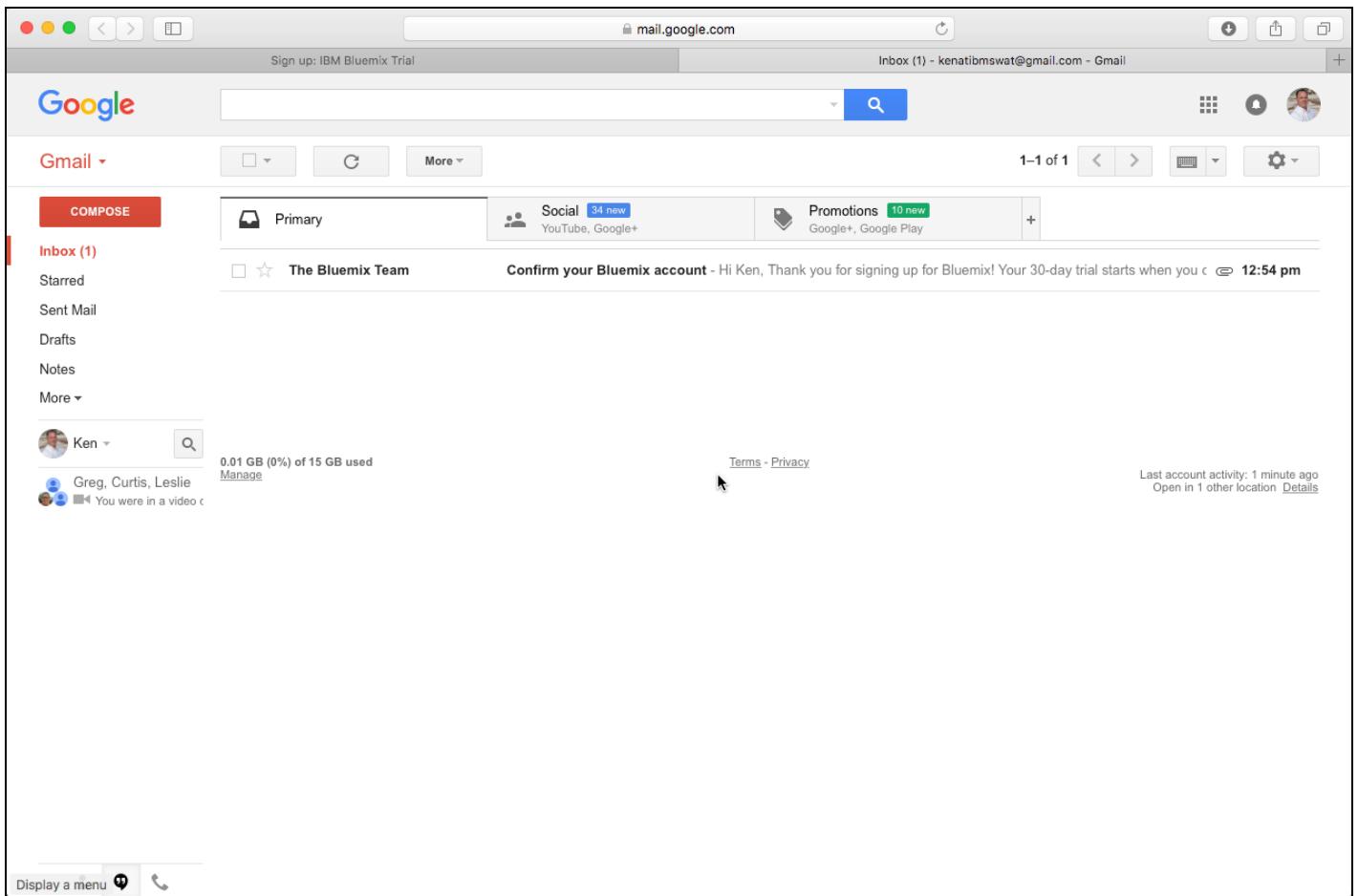
3. Complete the form and press the CREATE ACCOUNT button

The screenshot shows the 'Sign up for IBM Bluemix' page. At the top, it says '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 that, there's a link 'Already signed up for Bluemix? [Log in](#)'. The form fields are as follows:

- Email Address*: A text input field with a placeholder and a right-pointing arrow icon.
- 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 showing 'UNITED STATES'.
- Phone Number*: A text input field.
- 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 dropdown menu, there's a note: 'Keep me informed of products, services, and offerings from IBM companies worldwide:' followed by two checkboxes: 'By email' and 'By telephone'. At the bottom right is a 'CREATE ACCOUNT' button.

4. Check your email for your next steps.



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

SIGN UP: IBM Bluemix Trial

Confirm your Bluemix account - kenatibmswat@gmail.com - Gmail

Google

Gmail ▾

Compose

Inbox

Starred

Sent Mail

Drafts

Notes

More ▾

Ken

Greg, Curtis, Leslie

You were in a video c

Sign up: IBM Bluemix Trial

Confirm your Bluemix account

The Bluemix Team no-reply@bluemix.net via sendgrid.softlayer.com to me 12:54 PM (2 minutes ago)

IBM Bluemix

Hi Ken,

Thank you for signing up for Bluemix! Your 30-day trial starts when you confirm your account by clicking the link that follows. During your trial, you'll have access to Bluemix to build apps, with 2GB of runtime and container memory, as well as up to 10 provisioned Bluemix services. And, you won't need to provide any form of payment for this trial.

When you click to confirm your account, you accept the [Bluemix Terms of Use](#).

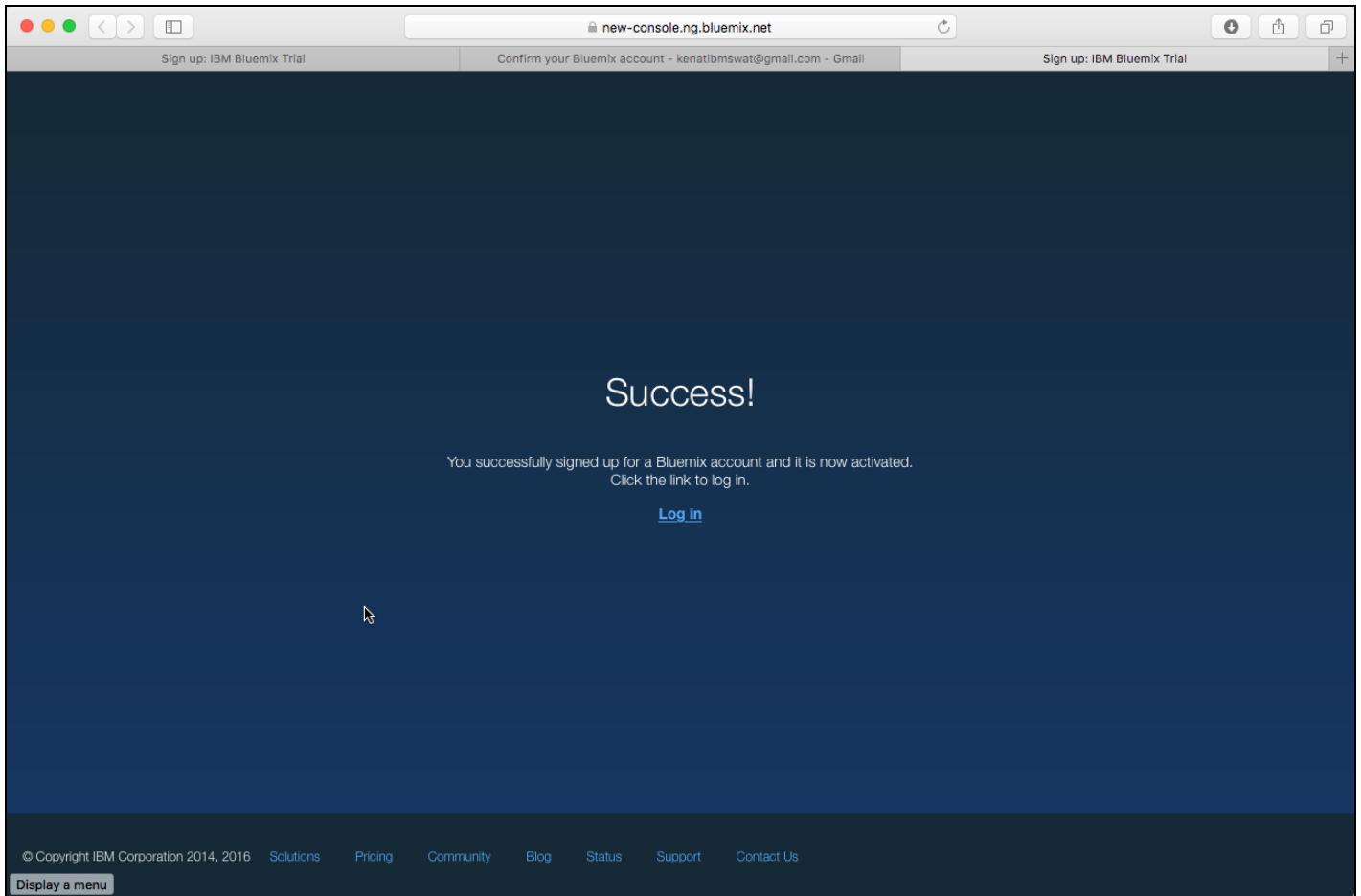
[Confirm your account](#)

If you have any problems logging in, let us know at [ibm.biz/bluemixsupport](#).

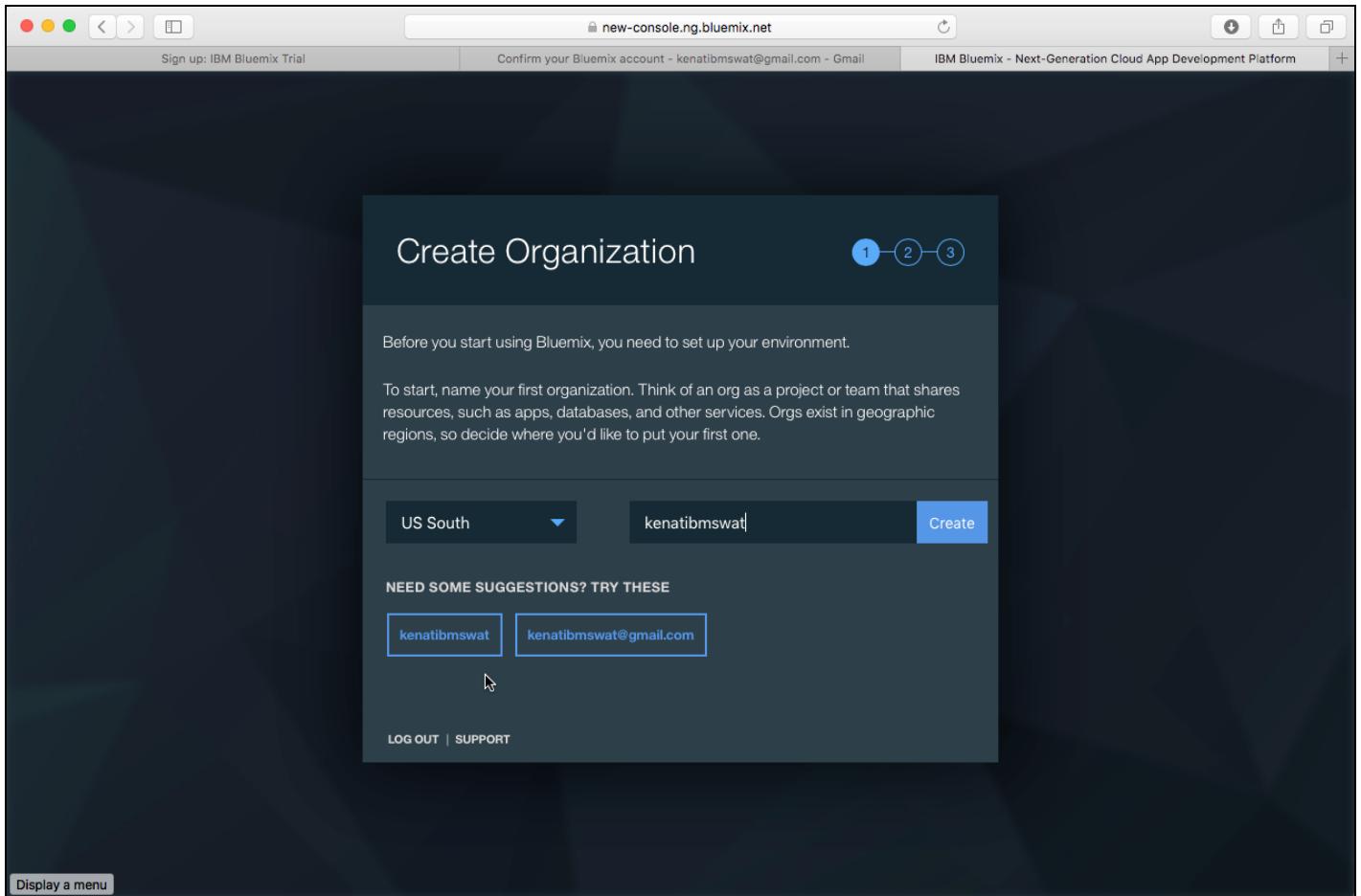
-IBM Bluemix Team

IBM

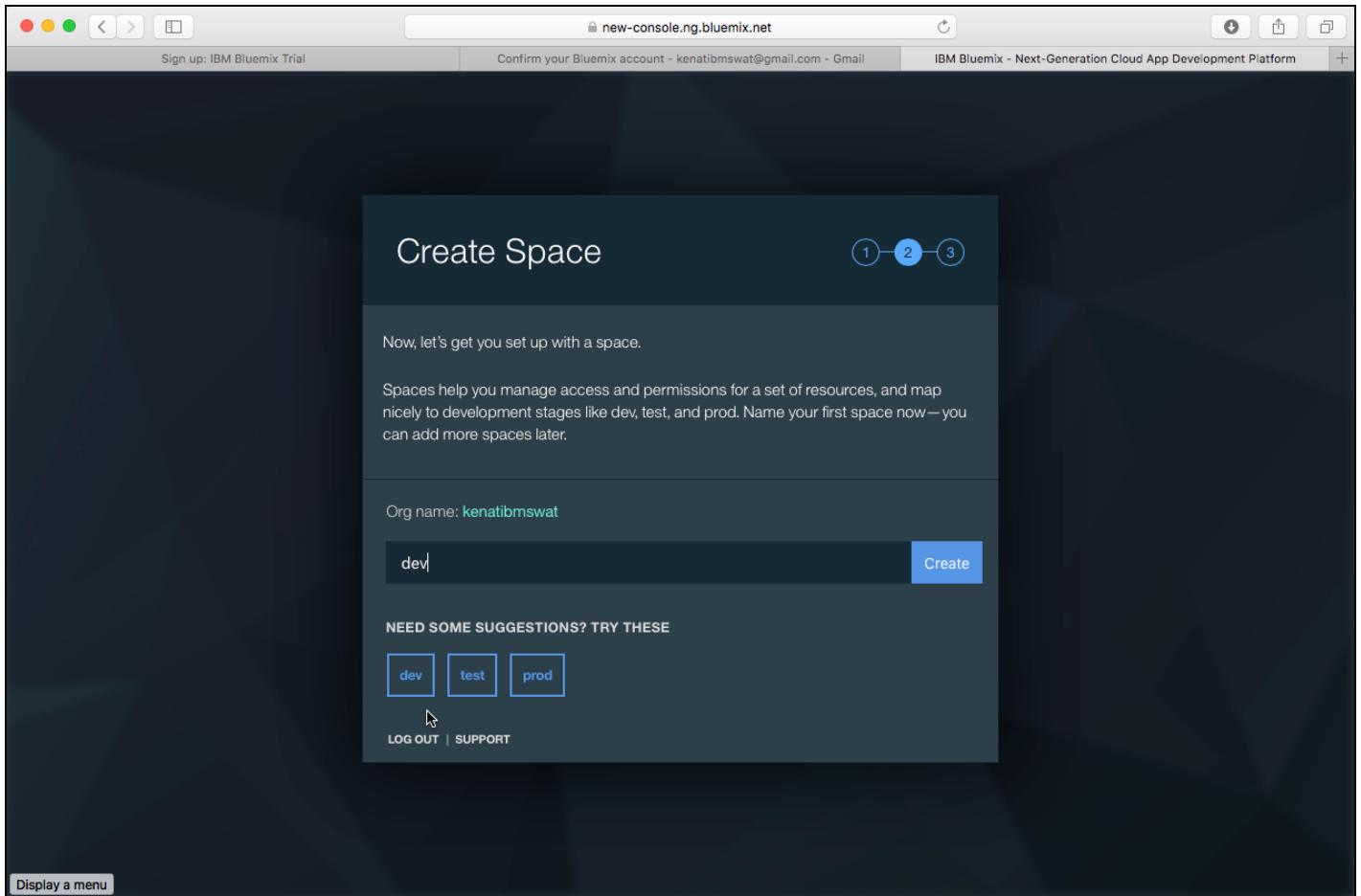
- Once confirmed, you will be taken to a page that says Success! To login, click the Log In link



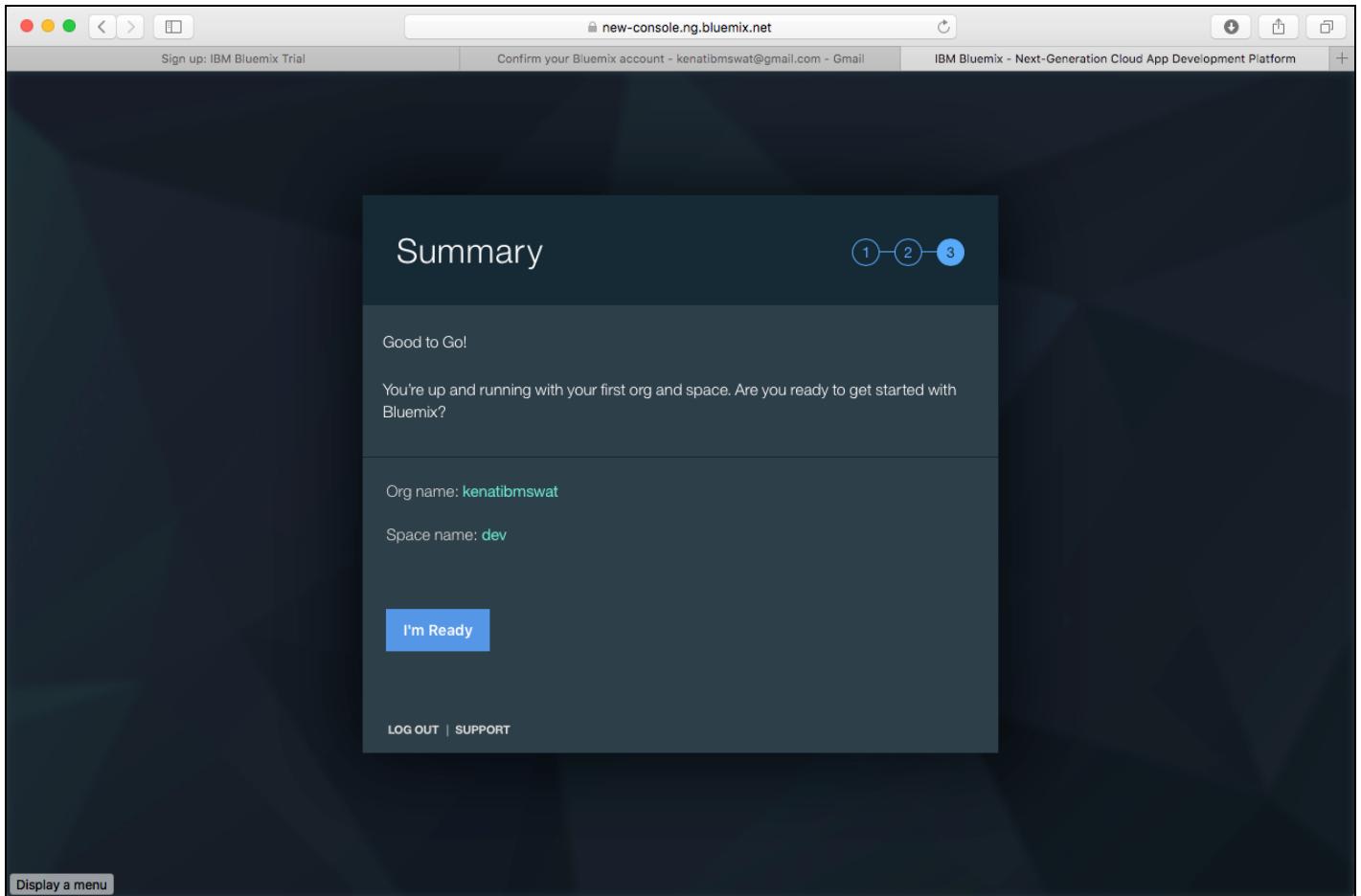
7. Enter your email address and press the CONTINUE button
8. Then enter your password on the next page and press the LOG IN button
9. You will be prompted to create an organization. Enter an organization name (notice that there are suggestions for you). Also select an appropriate region. Then press the CREATE button.



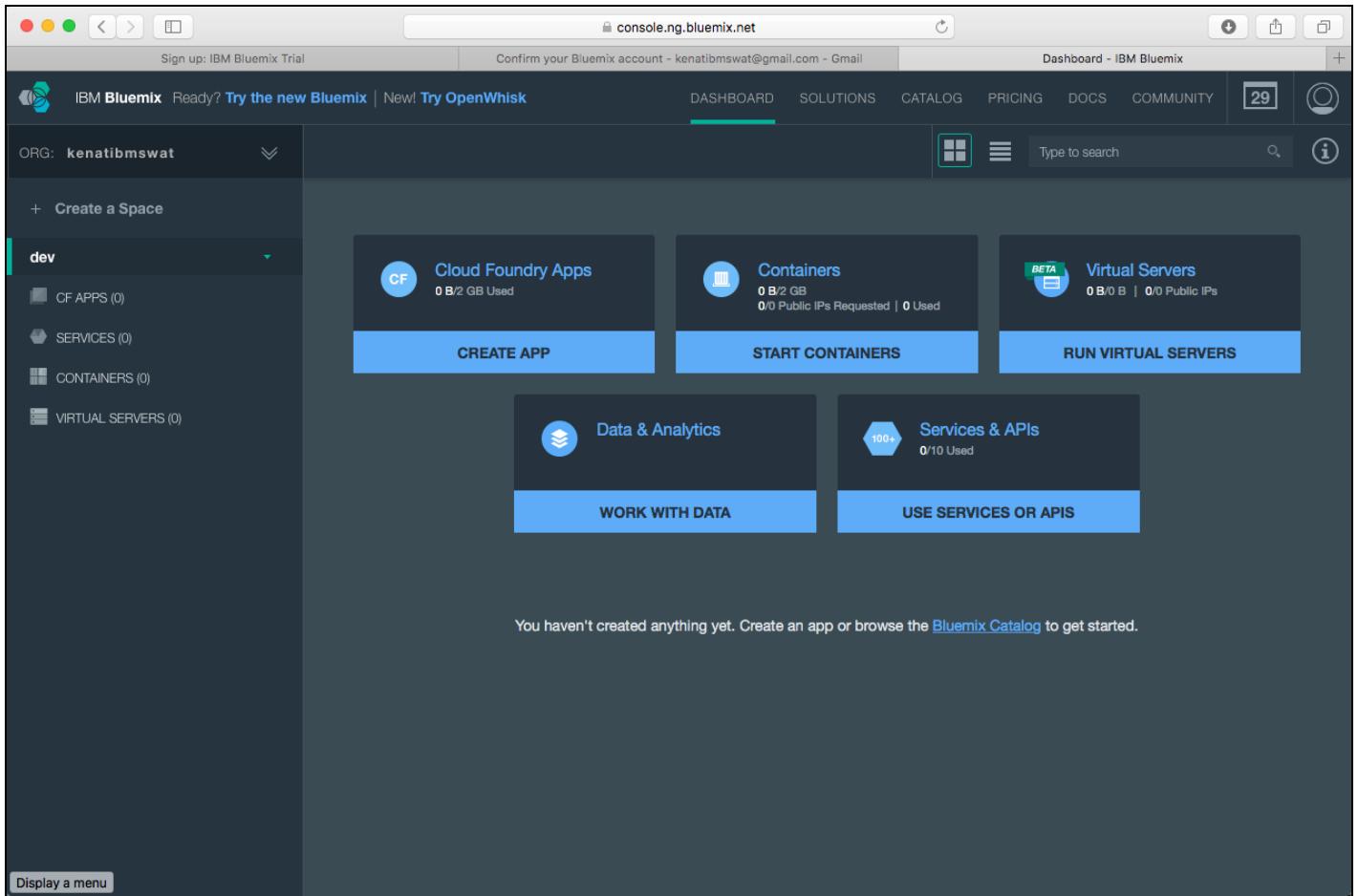
10. Next you will be prompted to create a space such as dev, test, prod, etc. You can name it whatever you would like (again notice the recommendations). Then press the CREATE button.



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

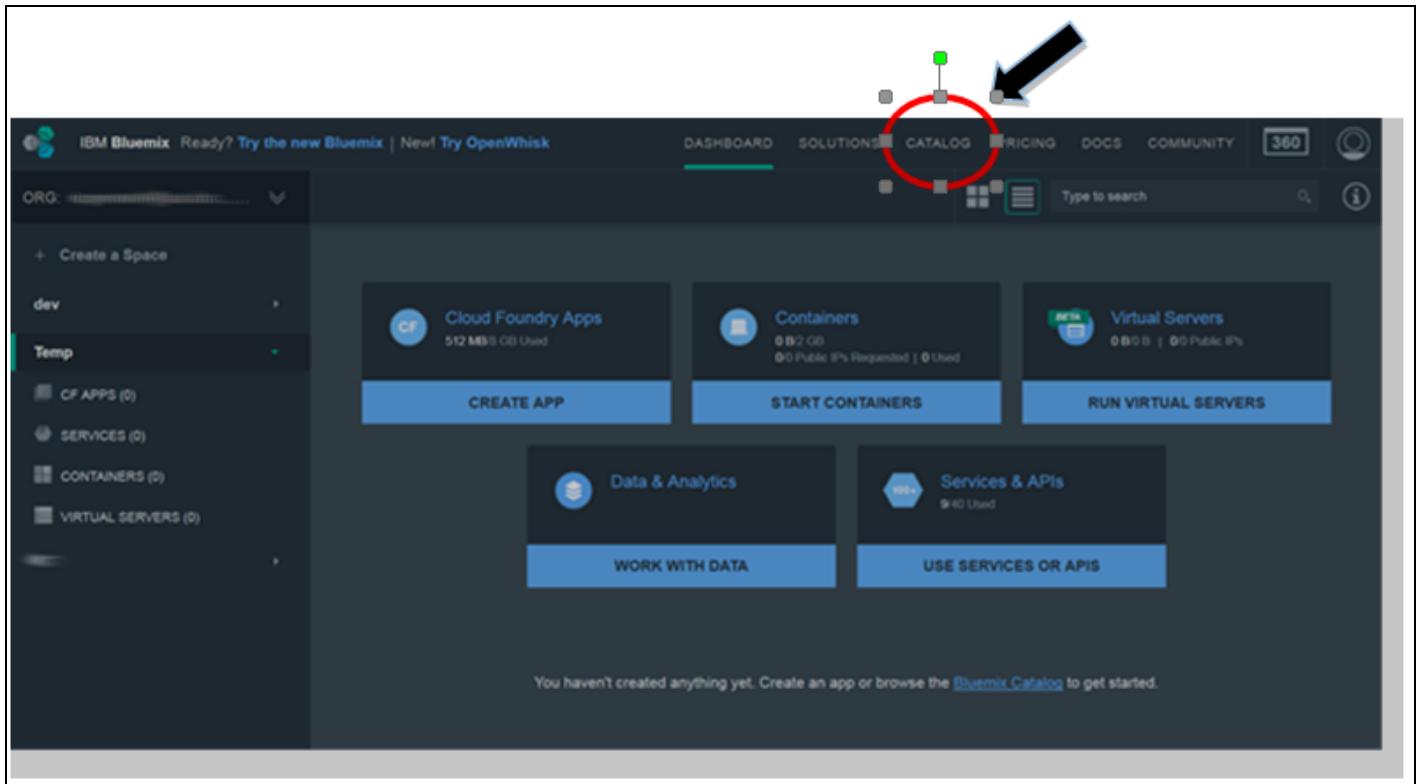


12. Your account has been created and configured once you see the following screen.

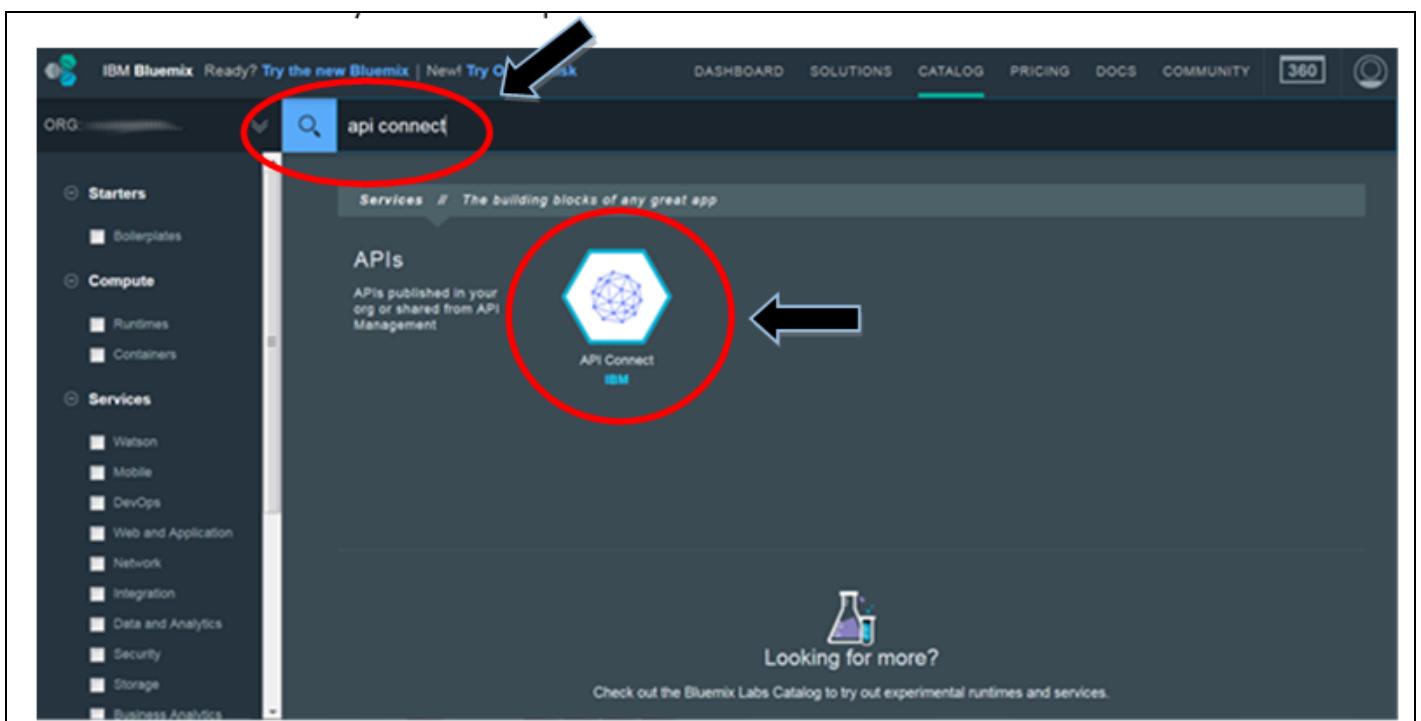


1.1a - Enable the API Connect Service on your Bluemix Account

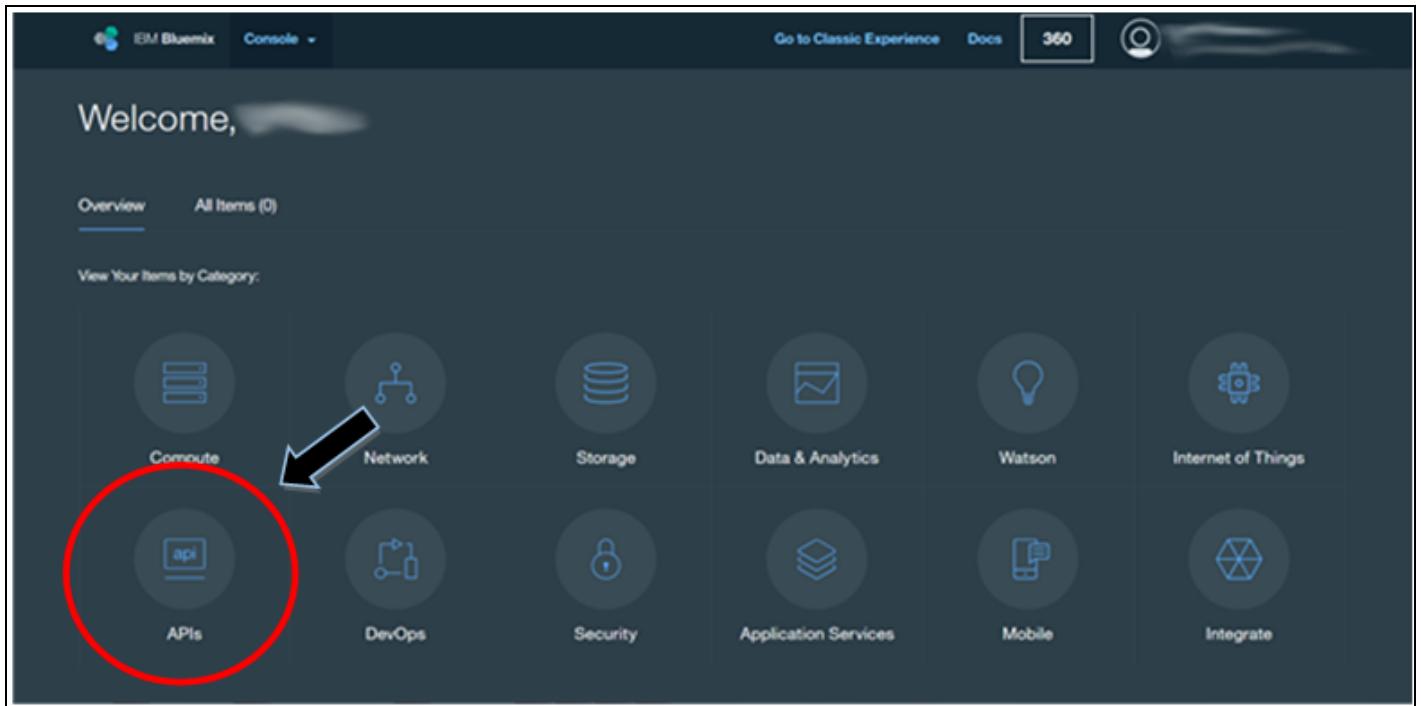
1. The next step is to enable the API Connect service on your Bluemix account. To start the process, click on the **CATALOG** tab in the upper right hand side of the screen



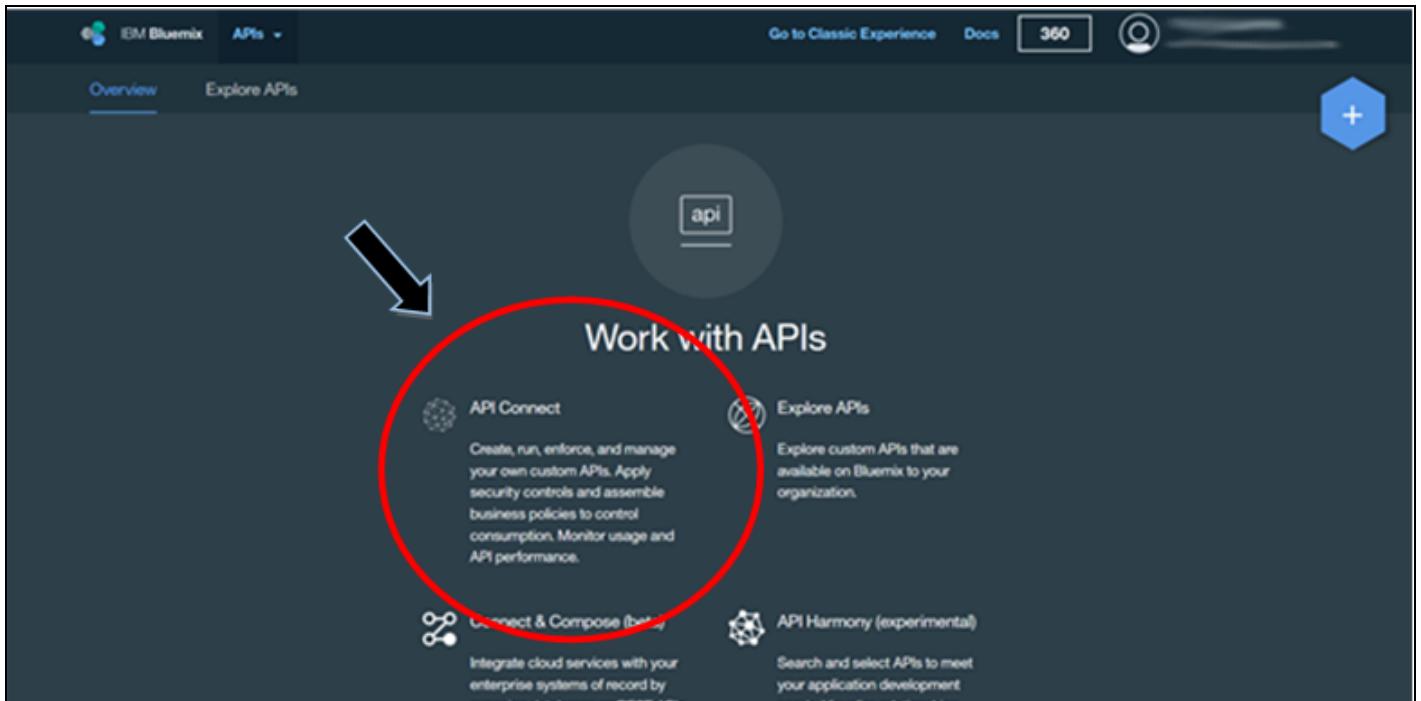
2. Type **Api connect** in the search box next to the magnifying glass icon
3. Click on the **API Connect** Icon to install a new instance of API Connect into your Bluemix space.



4. Click on the **APIs** tile.



5. Click on the **API Connect** tab.



6. Click on the **Create** button.

IBM Bluemix APIs > Back to Overview

Get started with API Connect

Looks like you don't have an API Connect account in this Bluemix space yet.
Click create to select a plan and provision your account.

CREATE

7. In the **Add Service** frame on the right side of the page, fill in the information based on your needs.
 - SPACE – The name of your Bluemix Space to deploy the API Connect Service
 - APP – Keep the default “Leave unbound”
 - Service Name – The name you want to give to your API Connect implementation if required
 - Selected Plan – Keep the default
8. Click on the **Create** button as shown below.

IBM Bluemix Ready? Try the new Bluemix | New! Try OpenWhisk DASHBOARD SOLUTIONS CATALOG PRICING DOCS COMMUNITY 360

Back to All Categories

API Connect IBM

PUBLISH DATE 07/11/2016

TYPE Service

VIEW DOCS

IBM API Connect is a comprehensive end-to-end API lifecycle solution that enables the automated creation of APIs, simple discovery of systems of records, self-service access for internal and third party developers and built-in security and governance. Using automated, model-driven tools, create new APIs and microservices based on Node.js and Java runtimes—all managed from a single unified console. Ensure secure & controlled access to the APIs using a rich set of enforced policies. Drive innovation and engage with the developer community through the self-service developer portal. IBM API Connect provides streamlined control across the API lifecycle and also enables businesses to gain deep insights around API consumption from its built-in analytics.

Securely unlock existing IT assets

Rapidly generate Swagger compliant APIs from backend datasources. Iteratively design and test APIs thereby shortening development cycles.

Community & Subscription management

Create developer communities to publish and share APIs and engage with them through a self-service portal. Also manage the subscriptions of the

Graphical API assembly

Graphically assemble the API invocation flow and apply policies that need to be enforced for secure controlled access.

Gain insights about API consumption

Built-in analytics and customized dashboards empower businesses to make informed decisions.

Add Service

Space: Temp

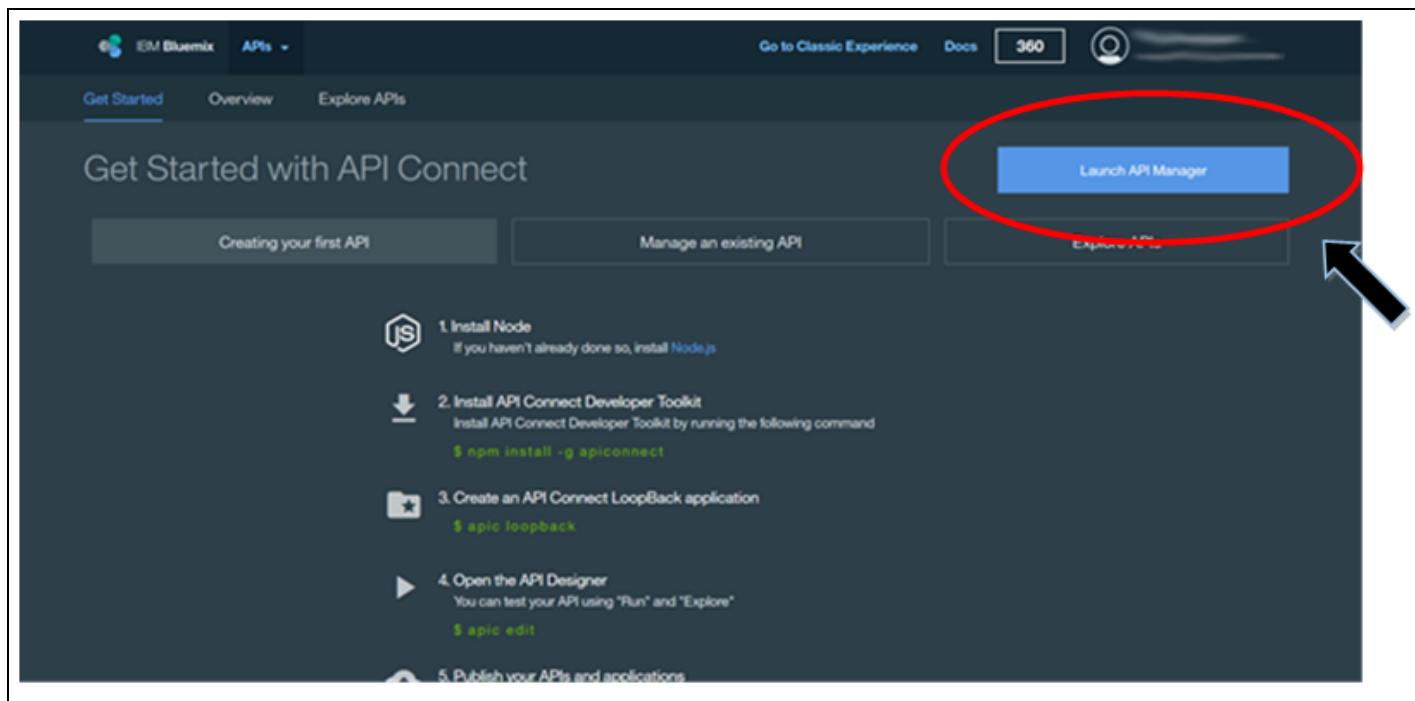
App: Leave unbound

Service name: API Connect-60

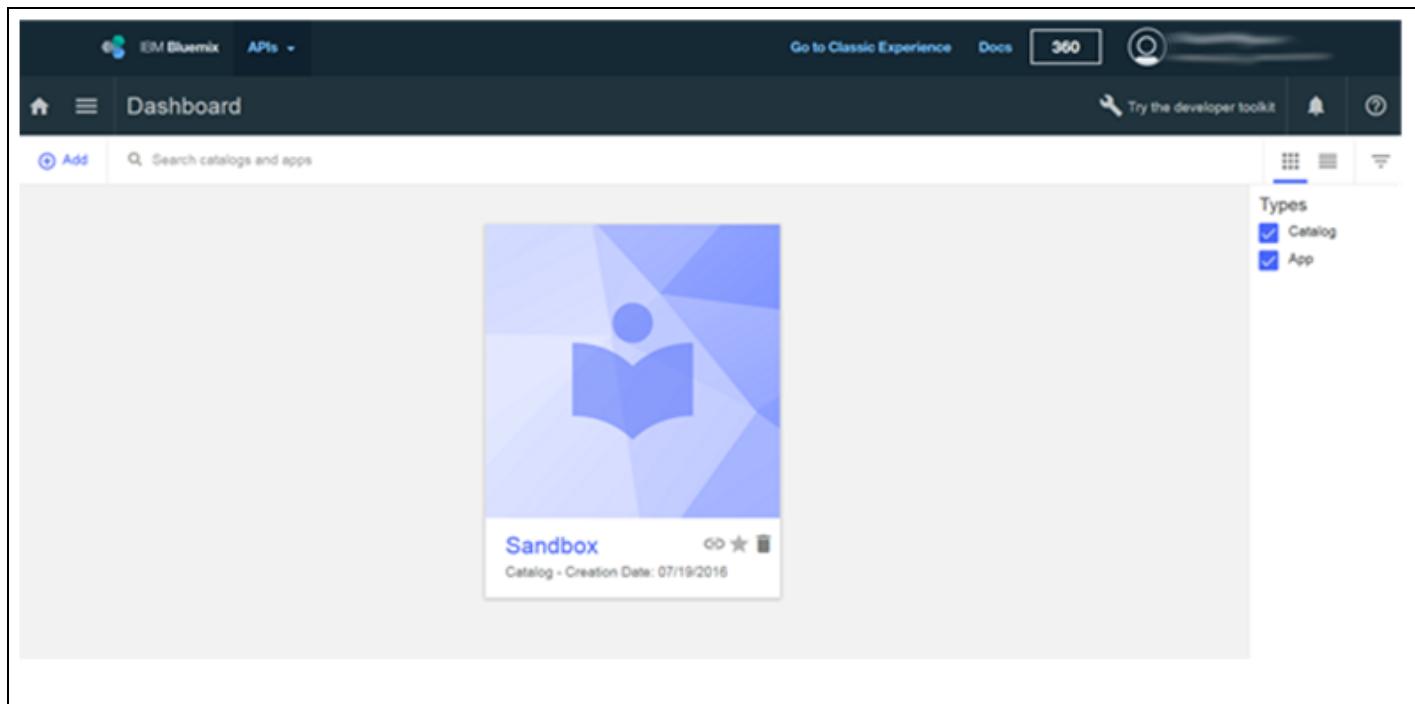
Selected Plan: Essentials

CREATE

9. The API Connect service is now deployed. To confirm the instance is properly created, click on **Launch API Manager**



10. If you are able to see your **Sandbox** catalog as shown below, this means the process of creating and setting up your Bluemix account to use API Connect is now complete. Close your browser window and proceed to section 1.2 below



1.2 - Creating a `hello-world` Application

1. We will use the API Connect Developer Toolkit command line interface to create the initial application and explore the created artifacts.
2. From the terminal command line type:

```
apic loopback
```

This command starts the application generator, Yeoman, to help scaffold the new project. Just press enter for each of the three questions.

```
-----  
|       | .-----.  
|--(o)--| | Let's create a LoopBack |  
`-----' |       application! |  
( _`U`_ ) '-----'  
/___A___\ | ~ |  
|   | .---.|  
`-|°` Y `
```

? What's the name of your application? hello-world
? Enter name of the directory to contain the project: hello-world
create hello-world/
info change the working directory to hello-world

? What kind of application do you have in mind? hello-world (A project containing a sample, including a memory database)

This creates an application named "hello-world" in a directory of the same name. The application is a basic Hello World application. You will see a lot of messages printed to the command line window. It is creating a few resources for you and installing the various node modules. Once the node modules are loaded you'll notice that the process creates swagger and product definitions for you. Finally, the process displays some hints about what to do next. Since we've been given such lovely suggestions about what to do next, we may as well follow the first one at least.

3. Change directories to the project directory:

```
cd hello-world
```

4. List the directory:

```
$ ls
client  common  definitions  node_modules  package.json  server
```

If you're familiar with LoopBack or Node.js applications, some of the directories may be familiar to you. If not, here is a description of some of what is created.

Directory or file	Description
client	If the application has a front end, this is where the HTML, CSS, JavaScript, etc would be located. For our sample application there is only a stubbed out README.md.
common	Files common to both the server and the client application.
common/models	This sub-directory contains all the model JSON and JavaScript files. By model we're referring to a data model.
common/models/note.js	Custom script for the note data model. This file contains the implementation of the methods defined by the model definition file.
common/models/note.json	The note model definition file. This file contains the definition of the properties and methods for this model.
definitions	This directory contains the definitions for the APIs and the product contained in this application.
definitions/hello-world-product.yaml	YAML file for the the hello-world product. Which includes default plans for testing locally.
definitions/hello-world.yaml	Swagger definition file for the hello-world API. Includes information about the REST paths and operations, schemas for data models, security requirements, etc.
node_modules	Directory containing all the required node modules for the default application.
package.json	Standard Node.js package specification. Most importantly it contains the application package dependencies.

server	This directory contains the Node.js application and configuration files. We'll not look at them all in this tutorial, but here are a few.
server/server.js	The main application script for this application. Note: this is defined in the package.json file.
server/config.json	This file contains the global application settings, such as REST API root, hostname and port.
server/model-config.json	This file binds models to data sources and specifies whether a model is exposed over REST, among other things.
server/datasources.json	This file is the data source configuration file.



* .md files, such as that found in the client directory, are markdown files used for internal documentation.

1.3 - Launching the `hello-world` Application

- Now that we've explored what is created by the application generator, let's move on to the API Designer. From the command line:

```
apic edit
```

- Next, click on the `start` button located on the bottom panel of the API Designer to launch the `hello-world` application. On a Windows environment, you might see 2 node windows pop up on your screen. Minimize, but do not close these windows.



Stopped

- Once start completes, you should see a screen similar to this:

```
■   ▶ | Running | Micro Gateway: https://127.0.0.1:4002/ | Application: http://127.0.0.1:4001/
```

- Notice that once the application is up and running, stop and restart buttons will appear on the bottom side of the screen:

At this point we're ready to Explore and test our services.

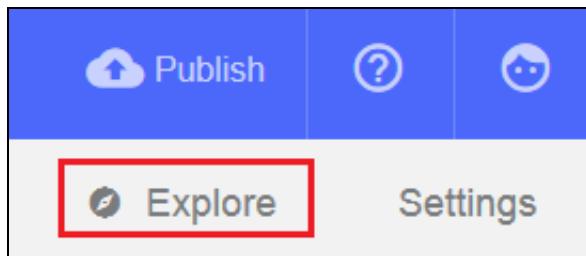


We used the web-based editor to launch the application. There's also a command provided with the API Connect Developer Toolkit that can be utilized from the terminal to launch the application:

```
apic start
```

1.4 - Testing the hello-world Application

1. Click the **Explore** button to switch to the API Explorer view.



You will see all the exposed service paths displayed.

The screenshot shows the IBM API Connect interface for the 'hello-world' service. The left sidebar lists various service paths. The main panel displays the 'hello-world 1.0.0' service with a 'Security' section and a selected 'POST /notes' operation. The right panel includes a code editor with a curl command and tabs for different programming languages.

```
curl --request POST \
--url https://localhost:4004/api/notes \
--header 'accept: application/json' \
--header 'content-type: application/json' \
--header 'x-ibm-client-id: default' \
--header 'x-ibm-client-secret: SECRET' \
--data '{"title":"et eligendi ut assumenda","content":"soluta","id":49233121}'
```

2. Now we're going to test the services using the GUI presented on the explore screen. You'll notice that on

the left several REST services are defined for us. In particular, take a look `POST /notes` and `GET /notes`.

If you're not familiar with GET and POST, they are HTTP methods (sometimes called verbs). The POST method is used for creation calls to the service. The GET method is used to retrieve information from a service. In this case, we see that both methods are used against the `/notes` path. So, POST will create a `note` and GET will retrieve all the `notes` that have been created.

3. Start by creating a couple of notes. Click the `POST /notes` link in the left column. The other columns will scroll so that you're looking at information and controls pertaining to the `POST /notes` service.
4. To test creation of a note, scroll down in the right hand column until the `Call operation` button is visible at the bottom. Just above the `Call operation` button you'll see a `Generate` link. This link will generate dummy data for you to create a test call to this service.
5. Press the `Generate` link to generate some sample data.

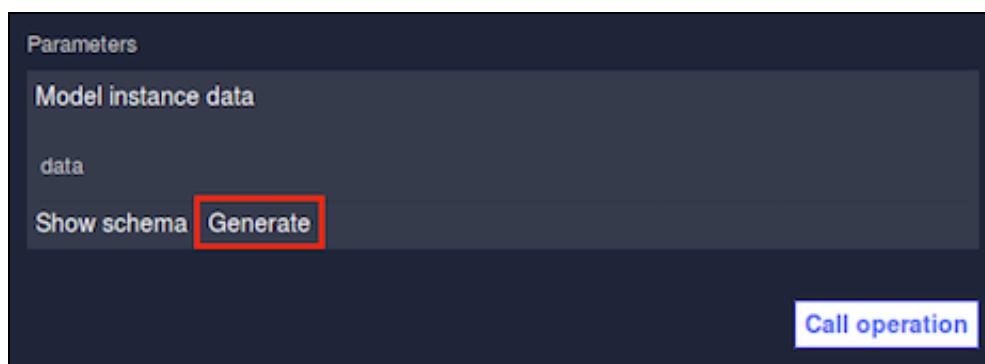
Parameters

Model instance data

```
data
```

Show schema | **Generate**

Call operation

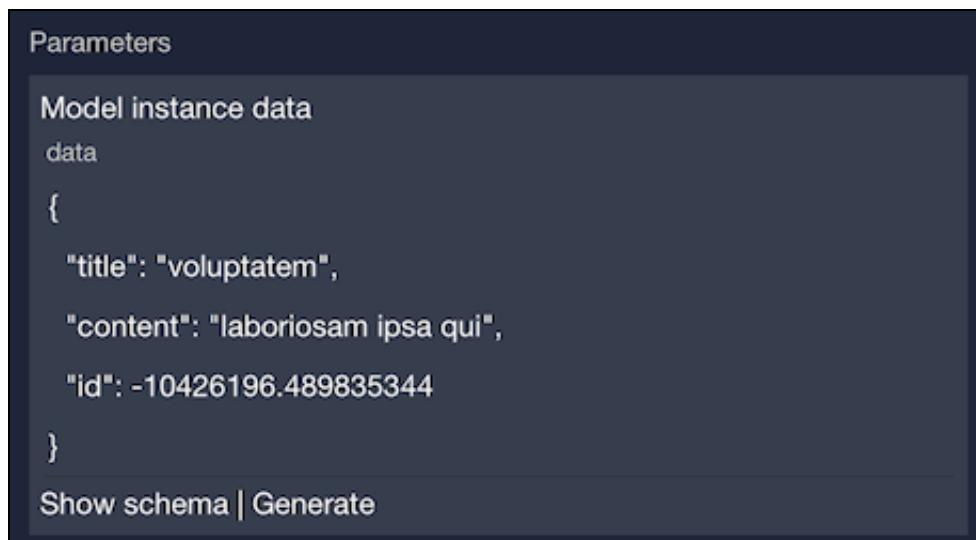


Parameters

Model instance data

```
data
{
  "title": "voluptatem",
  "content": "laboriosam ipsa qui",
  "id": -10426196.489835344
}
```

Show schema | **Generate**



Your data will look different, but you're ready to test the service.

6. Go ahead and press the `Call operation` button and scroll down to the `Response` section to see

the results.

In the results, you should see a `Code: 200 OK` which indicates that a new `note` was created. If you received a different response, see the troubleshooting steps below.

Response

```
Code: 200 OK
Headers:
content-type: application/json; charset=utf-8
x-ratelimit-reset: 3561141
x-ratelimit-limit: 100
x-ratelimit-remaining: 99

{
  "title": "nostrum in cum necessitatibus sapiente",
  "id": 1
}
```



You may see an error displayed that mentions a CORS issue. This has to do with certificates in your browser. Go ahead and click the given link to rectify this, accept any certificate, close the opened tab, and press the `Call operation` button again. Additionally, be sure not to skip step 5, as doing a `POST` operation without generating payload will cause an error.

Response

```
Code: -1

No response received. Causes include a lack of CORS support on the target server, the server being unavailable, or an untrusted certificate being encountered.

Clicking the link below will open the server in a new tab. If the browser displays a certificate issue, you may choose to accept it and return here to test again.
https://localhost:4002/api/notes
```



If you see a 500 error like the one below, make sure you press the `Generate` button before you press the `Call operation` button again. Otherwise, you're trying to create a duplicate note.

Response

Code: 500 Internal Server Error

Headers:

```
content-type: application/json; charset=utf-8
x-ratelimit-reset: 3594231
x-ratelimit-limit: 100
x-ratelimit-remaining: 98
```

```
{
  "error": {
    "name": "Error",
    "status": 500,
    "message": "Duplicate entry for note.id",
    "stack": "Error: Duplicate entry for note.id\nat Memory._createSync (/Users/tcat/Documents/data/gigs
```

- Once you have created one note, go ahead and create another one or two.



It should be noted that you don't need to use the `Generate` link. You can type data directly into the `Parameters`. You can also use `Generate` to create a template for you to use and then change the generated parameters. You may also notice that not all the parameters are always generated. This is because only the `title` parameter is required. Try pressing `Generate` several times to get a feel for how it works.

- Finally, let's test the `GET /notes` service. We should have two, three, or more notes created at this point. In the left hand column click the `GET /notes` link.

hello-world 1.0.0

POST /notes

PUT /notes

GET /notes

GET /notes/{id}/exists

HEAD /notes/{id}

3. Scroll down to the `Call operation` button and press it. Then scroll down to the results.

Response

Code: 200 OK

Headers:

content-type: application/json; charset=utf-8

x-ratelimit-reset: 3571999

x-ratelimit-limit: 100

x-ratelimit-remaining: 95

```
[  
  {  
    "title": "test test test",  
    "id": 1  
  },  
  {  
    "title": "quia itaque est minima",  
    "content": "molestiae iure nostrum magnam",  
    "id": 2  
  }]
```

You should see all the notes that you generated in the result set.



If you see an empty array, `[]`, as your result, then you've not successfully created any notes. This is also true if you stop the application and restart it. With the hello-world example, we're using an in-memory database which means that nothing is persisted to disk. So, it is lost when the server is stopped and restarted. Lab 2 will walk through how to connect to your application to a persistent data source.

4. At this point, we are done testing the app locally. Click on the `Run` button again to return to the

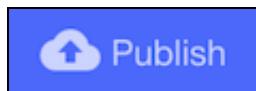
application launch screen.

5. Click on the `Stop` button to stop the application.

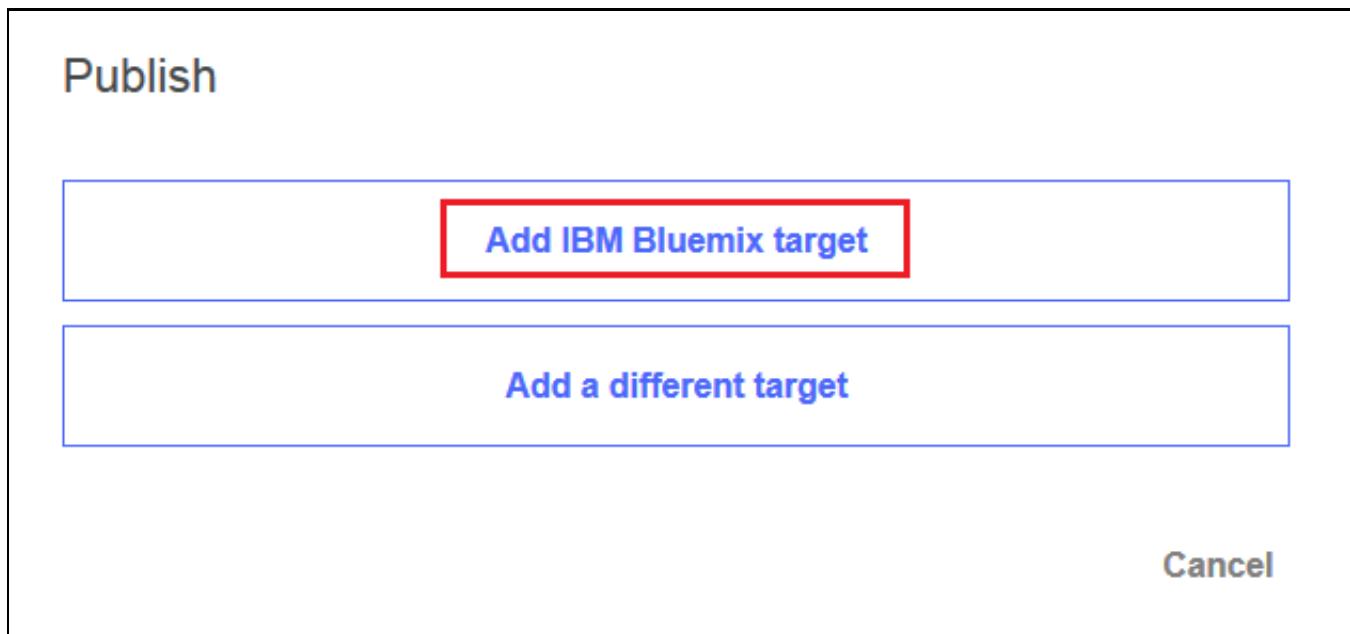


1.5 - Publishing the API to the Developer Portal

1. Click the `Publish` icon.



2. Select `Add and Manage Targets` from the menu.
3. Select `Add IBM Bluemix target`.



4. Provide connection information to sign into the IBM API Connect Bluemix service then click the `Sign in` button: **Note you should be logged in already to Bluemix, so you should not need to reauthenticate. It will automatically populate your information and have the `Sandbox` catalog already selected, so at this point you only need to click `Next`.

Select an organization and catalog

Region

US South

Organization

your org name here (dev)

 Search

None

Sandbox

[Back](#)

[Cancel](#)

[Next](#)

5. On the "Select an App" screen, choose **None** application and click the **Save** button.

Select a Bluemix application

Search

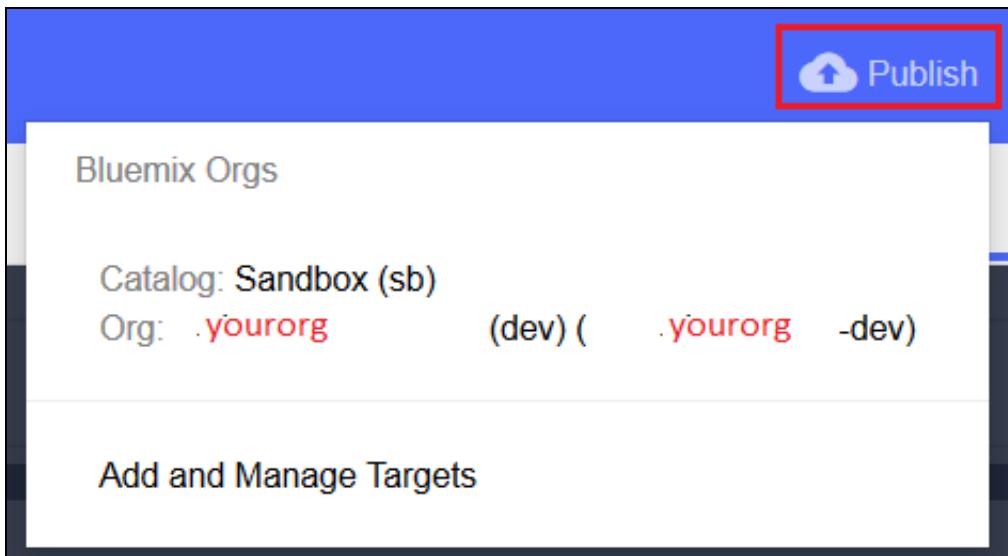
None

Type new application name

[Back](#) [Cancel](#) [Save](#)

Our offline toolkit environment is now configured to speak to API Connect on Bluemix

6. Click **Publish** button once more and select our target:



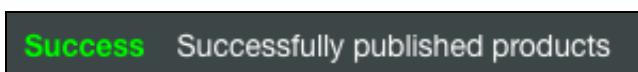
7. Here we have the opportunity to select what gets published. If we were working on multiple API products as part of this project, we could choose specific ones to publish.

Also, the option exists to only Stage the product. A Stage-only action implies that we'll push the configuration to the Management server, but not actually make it available for consumption yet. The reason for doing this may be because your user permissions only allow staging, or that a different group is in charge of publishing Products.

Click the **Publish** button:

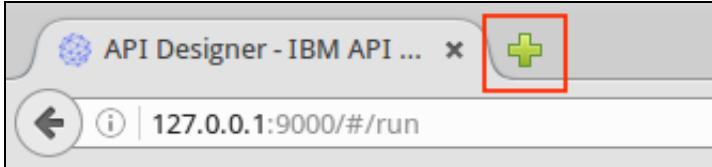


8. Click the **Publish** button to make the API available in the Developer Portal.
9. Wait a moment while the Product is published, a Success message will appear letting you know the step is complete:

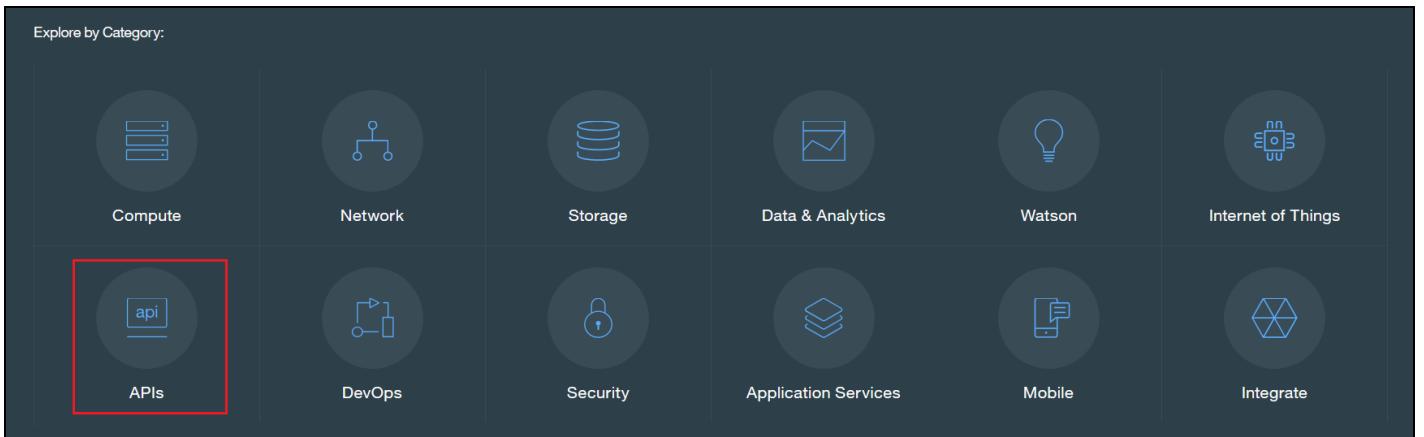


1.6 - Browsing the API in the Developer Portal

1. Open a new tab in your web browser by clicking on the new tab  button.



2. Log into your Bluemix Account using this url: <https://new-console.ng.bluemix.net/>
3. Via the dashboard, click on the  Icon.



4. Click on your  instance

Work with APIs



API Connect

Create, run, enforce, and manage your own custom APIs. Apply security controls and assemble business policies to control consumption. Monitor usage and API performance.



Explore APIs

Explore custom APIs that are available on Bluemix to your organization.



Connect & Compose (beta)

Integrate cloud services with your enterprise systems of record by exposing databases as REST APIs to be used by applications.



API Harmony (experimental)

Search and select APIs to meet your application development needs. Visualize relationships between selected APIs and narrow down results based on search-history.

Service Instances API Bindings

The screenshot shows a dark-themed user interface for managing service instances. A single service instance is listed: 'API Connect-5z' (with the sub-label 'APIConnect'). Below the instance name is the 'Plan: Essentials'. A red rectangular box highlights the entire service instance card.

1. Click on **Launch API Manager** to launch your API Manager

The screenshot shows the 'Get Started with API Connect' page. At the top, there are three buttons: 'Creating your first API', 'Manage an existing API', and 'Explore APIs'. To the right, a large blue button is highlighted with a red border and the text 'Launch API Manager'. Below these buttons, there is a numbered list of steps:

1. Install Node
If you haven't already done so, install [Node.js](#)
2. Install API Connect Developer Toolkit
Install API Connect Developer Toolkit by running the following command
`$ npm install -g apiconnect`
3. Create an API Connect LoopBack application
`$ apic loopback`
4. Open the API Designer
You can test your API using "Run" and "Explore"
`$ apic edit`

2. Click on your **Sandbox** Catalog

The screenshot shows the 'Sandbox' catalog interface. At the top left is the 'Sandbox' logo. On the right, the date and time are shown as '03/28/2016 9:26 AM'. Next to the date are icons for 'Catalog', a star, and a trash bin. The main area is currently empty, showing a light gray background.

3. Click on **Settings**

The screenshot shows the IBM Bluemix API Management interface. At the top, there's a navigation bar with 'IBM Bluemix' and 'APIs'. Below it is a header bar with 'Sandbox' and a menu icon. The main content area has a navigation bar with 'Dashboard', 'Products' (which is underlined in blue), 'Approvals', 'Developers', 'Subscriptions', 'Analytics', and 'Settings'. A red box highlights the 'Settings' tab.

4. Here you will see that you have no developer portal set up.

This screenshot shows the 'Portal' configuration screen. At the top, there are tabs for 'Configuration', 'Portal' (which is highlighted with a red box), 'Permissions', and 'Extensions'. The main section is titled 'Portal Configuration' and contains a radio button group. The 'None' option is selected (indicated by a blue circle). There are also other options: 'IBM Developer Portal' and 'Other'. Below this is a 'URL' section with a 'URL' input field.

5. Select the **IBM Developer Portal** Radio Button to create your own custom portal that is tied to your catalog and then click **Save**. It will automatically generate the portal URL and the portal as well.

This screenshot shows the 'Portal' configuration screen after selecting the 'IBM Developer Portal' option. The 'Portal' tab is highlighted with a blue underline. The 'IBM Developer Portal' radio button is selected (blue circle). Below it, a message box displays the generated 'Portal URL: https://new-catalog-1-malleyusibmcom-dev.developer.us.apiconnect.ibmcloud.com'. At the bottom, there's a note: 'Thank you for enabling the IBM Developer Portal. Please note that due to the time taken for the DNS definition to propagate around the Internet it may take up to 2 hours for your Portal to become accessible.'

6. A pop up screen will let you know that the process to create your portal has started.

7. It might take some time for your developer portal to get created, but usually the process is pretty quick. If this piece doesn't work for you right away, move on to the next lab and circle back to this lab in a bit. Once the portal is done creating, you will receive an email. If it still doesn't work by the time you get to the later labs, inform your instructor.

8. Once the process is complete, click on the link of the portal URL as seen in the screenshot above in step #5.
9. This will bring you to your portal login page. In order to view the API we have, we will need to register ourselves as a developer on the portal.
10. Click on **Login** at the top right

11. Enter in the name of a Development Organization. API Products are Published to Development Organizations that will be consuming these APIs

12. Once you are authenticated in, you can then Browse the **API Products** and see your **hello-world** product that is now published to your environment.

13. At this point we will stop, as we will be building additional apis and services that will be published to the portal.

It should be noted that the hello-world application has more to it than we've shown in this lab. You're encouraged to dig in and discover the custom method that's implemented. In future labs, we'll be doing more work in the Developer Portal. For instance we'll be customizing the portal theme, registering an application, subscribing to APIs and testing them from a separate consumer application.

14. Close the `Firefox` web browser. If a warning is presented about closing multiple tabs, deselect the option to notify you in the future and click the `Close Tabs` button.
15. In the terminal, use the `control+c` keyboard command to quit the API Designer program.

Lab 1 - Conclusion

Congratulations! You have installed node.js, the API Connect developer toolkit and have developed and published your first API!

In this lab you learned:

- How to install node.js and its prerequisites and the API Connect Developer Toolkit on your native OS.
- Set up your Bluemix Account.
- How to create a simple LoopBack application
- How to create a Representational State Transfer (REST) API definition
- How to test a REST API
- How to publish an API to the Developer Portal

Proceed to [Lab 2 - Create a LoopBack Application](#)