IBM Watson Workshop - Workstation Setup

Cognitive Solutions Application Development

IBM Global Business Partners
Duration:15 minutes
Updated: May 6, 2020
Klaus-Peter Schlotter
kps@de.ibm.com

Version 2

Overview

The IBM Watson Developer Cloud (WDC) offers a variety of services for developing cognitive applications. Each Watson service provides a Representational State Transfer (REST) Application Programming Interface (API) for interacting with the service. Some services, such as the Speech to Text service, provide additional interfaces.

IBM Cloud (aka IBM Bluemix) is the cloud platform in which you deploy applications that you develop with Watson Developer Cloud services. The Watson Developer Cloud documentation provides information for developing applications with Watson services in IBM Cloud. The complete list of Watson Developer Cloud services is available from the Watson services catalog.

The catalog provides a link to the landing page for each service, which provides an overview of the service and links to the documentation and a demo application for the service. The documentation for every Watson service provides links to the source code for the sample applications. APIs and SDKs are available for Node.js, Java, and other programming languages.

Objectives

- Learn how to get started with IBM Cloud
- Learn how to install the Cloud Foundry command-line interface
- Learn how to configure your workstation for IBM Watson Node.js development

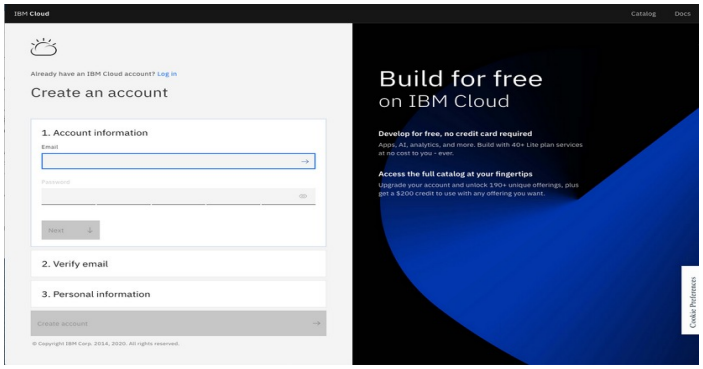
The steps described in the documents are:

- Create an IBM Cloud account
- Install the IBM Cloud command line interface (or cli)
- Install and configure Node.js and Express
- Install an Integrated Development Environment (IDE) or Text Editor

Create an IBM Cloud Account

Step 1 In a web browser navigate to the following URL:

https://cloud.ibm.com/registration/trial/



You can also create a ***Lite*** account with no time restrictions, that can be upgraded at a later time. The lite account is sufficient for the labs in this workshop but may have more restrictions (default Region assigned) than the *30 days trial*.

https://cloud.ibm.com/registration/

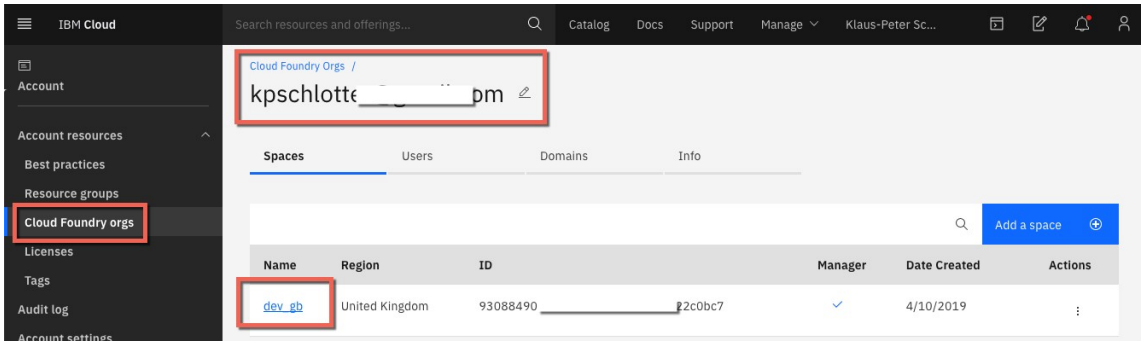
Step 2 Fill in the required information and press the button **Create Account**

Step 3 After you submit your registration, you will receive an e-mail from the IBM Cloud team with details about your account. In this e-mail, you will need to click the link provided to confirm your registration

Step 4 Now you should login to your new IBM Cloud account and see what region and space was automatically created.

a) **Select** *Manage → Account → Cloud Foundry orgs*

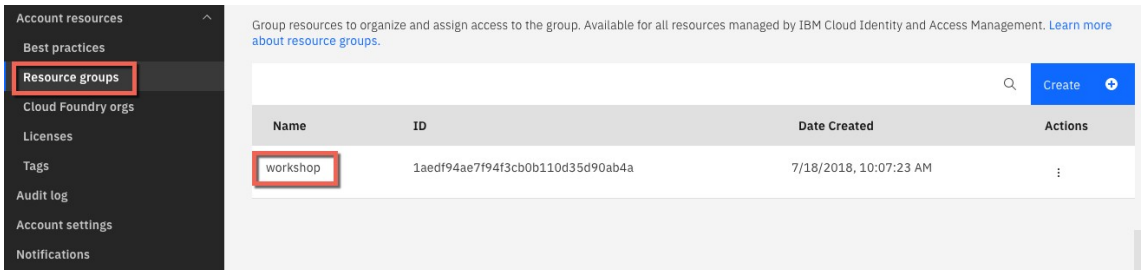
b) **Click** on the *organization* created by default (should be your Cloud Id).



The *Organization,* the *Space* and the *Region* are needed to work with Cloud Foundry based cloud objects via the Command Line Interface (CLI).

Region examples:
US South → us-south
US East → us-east
UK South → eu-gb
EU Central → eu-de

c) Now **click** on *Resource groups* and check the name there. In a Trial account you can have one resource group. Should be names *Default*. In the example below the name is *workshop*.



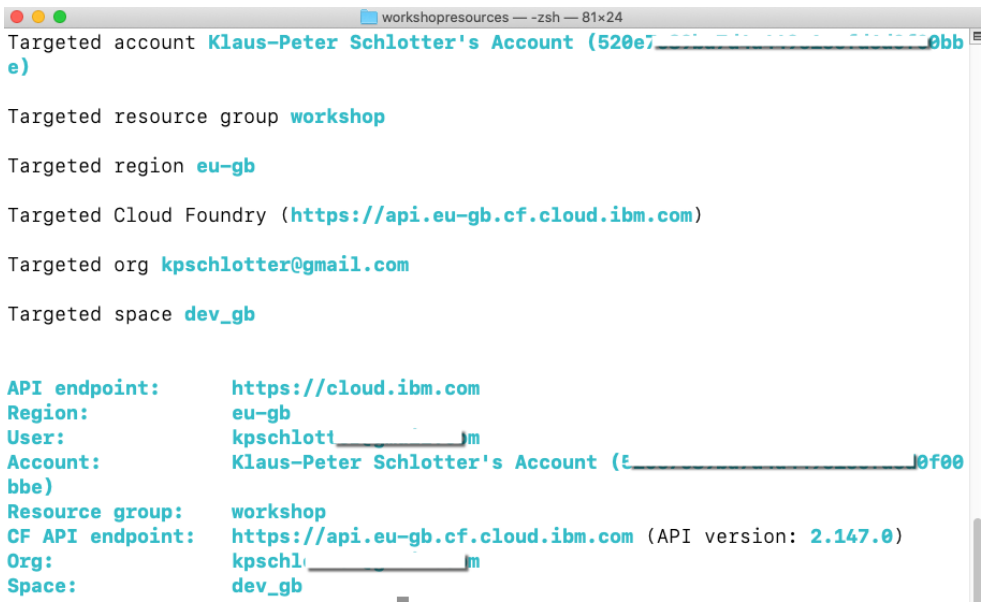
Step 5 Now install the IBM Cloud Command Line Interface (CLI) with these instructions here.

Step 6 Open a Terminal/Command Window and set your api endpoint.

```
ibmcloud api https://cloud.ibm.com
```

Step 7 Login to your IBM Cloud account (data from Step 4)

```
ibmcloud login -u <your ibmcloud user account> -o <yourOrg> -s
<yourSpace>  -r <yourRegion> -g <yourResourceGroup>
```



Install Google Chrome or Mozilla Firefox on your platform

Although any of the major Web browser should work for the workshop labs, it is recommended to install Google Chrome or Mozilla Firefox. Google Chrome provides a plugin for the REST client Postman that we will use in some of the labs.

Install and configure Node.js and Express

This section is intended for developers interested in the IBM Watson Node.js APIs.

Node.js is an open-source, cross-platform runtime environment for developing server side web applications using JavaScript. Node.js has an event-driven architecture capable of asynchronous I/O utilizing callbacks.

Express is a minimal and flexible Node.js web application framework that provides a robust set of features to develop web and mobile applications. Its ability to facilitate rapid development of Node based web applications makes it the de facto framework for Node.js.

Step 8 In a web browser, navigate to the Node.js homepage.

https://nodejs.org

Step 9 Download and install the latest Node.js runtime on your local workstation. The procedure may vary per your operating system.

Step 10 As suggested on the last window of the installer, update your PATH variable (if necessary).

Step 11 Validate that the installation is successful in a Terminal/Command window with the command:

```
node –version
```
or
```
node -v
```

Step 12 In a terminal/command window, enter the following commands to install the Express framework so that it is globally available to all your Node applications.

```
npm install -g express
npm install -g express-generator
```

Note: In case you receive a message like “permission denied access”, prefix your commands with the prefix sudo which give you superuser rights:

```
sudo npm install -g express
sudo npm install -g express-generator
```

Install a Development Environment or Text Editor

To work with the labs a simple editor will do it. But for convenience specific code editors will be a better choice because they can open a complete project folder where you then can navigate between the various files in the project very easily and they also provide syntax highlighting depending on the file content type.

The screenshots in the labs are taken with Microsoft Visual Code.

Code Editor Examples:

- **Sublime Text** – https://www.sublimetext.com
- **Atom** – https://atom.io

Integrated Development Environments

- **Microsoft Visual Studio Code** – https://code.visualstudio.com
  No additional software needed.
- **Eclipse** – https://eclipse.org
  Additional Plugins from the eclipse marketplace have to be installed, such as **IBM Cloud Tools** and **IBM Node.js Tools**.

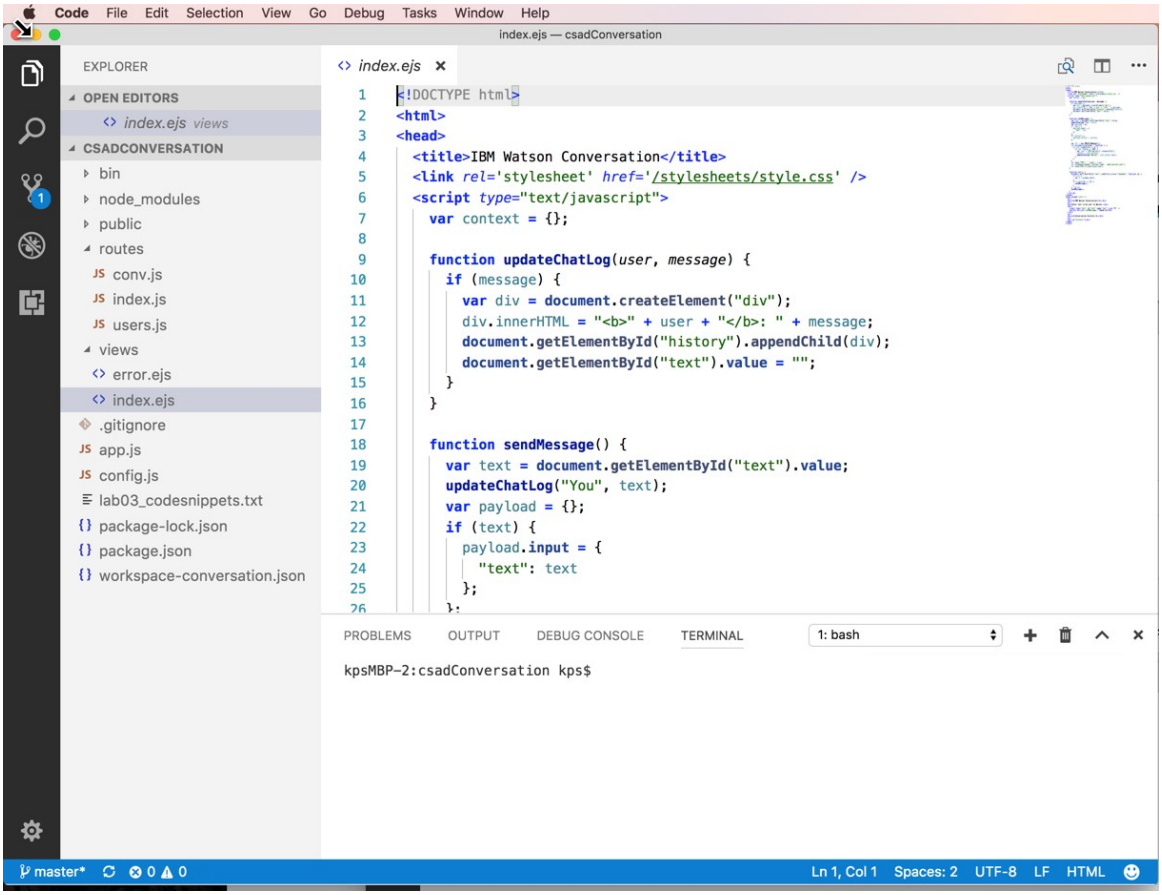


Image 3: Visual Studio Code

Hardware for the Labs

All the documentation for the labs is distributed via a publicly available internet site. The labs can be done on a Windows or Linux PC or on a Mac. An additional Tablet to display the PDF with the lab instructions may help. All Cut 'n' Paste information can be accessed from a browser window on the PC/ Mac. With iDisplay the tablet can even extend the Windows or Mac desktop.

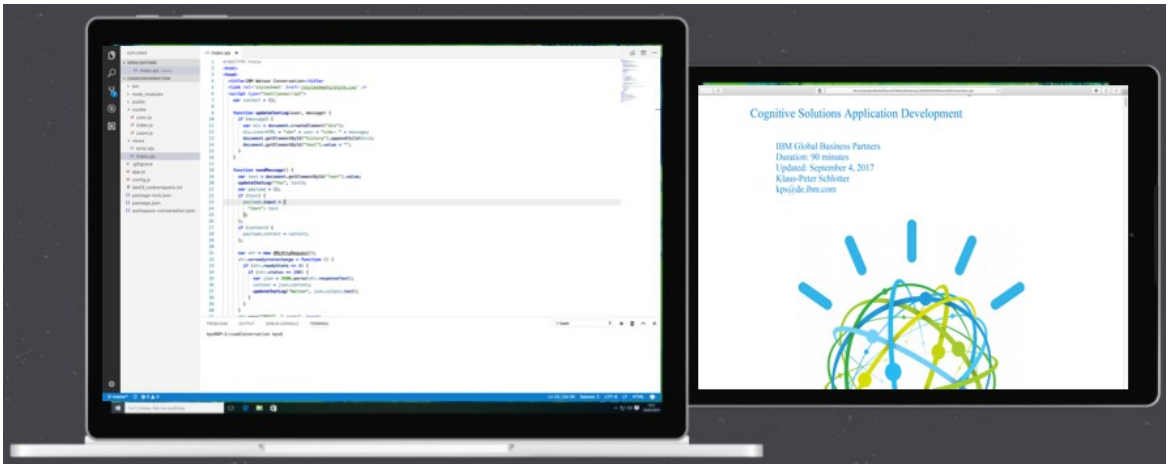


Image 4: Workstation and Tablet together