Preparing a Linux machine or virtual machine for the Labs

This document provides the steps to install the software necessary for the lab exercises on a Linux machine. It can also be used to set up the software on a VMware Linux virtual machine.

If you are working with a native Linux machine and not a virtual machine, you can start with Step 4. You can also skip to Step 4 if you want to work with an existing Ubuntu virtual machine image.

1. Install VMware
2. Download the Ubuntu Operating System
3. Create a virtual machine
4. Install Java JDK 1.8
5. Create JAVA_HOME environment variable
6. Install GIT
7. Install cURL
8. Install node.js
9. Install API Connect Developer Toolkit
10. Install docker
11. Install cloudfoundry CLI
12. Install the IBM Container plugin for cloudfoundry

If you are going to create a Linux virtual machine on a Windows host, verify the following:

1. make sure you have at least 22GB of available disk space (20GB for virtual disk and 2GB for the Ubuntu iso image).
2. Make sure IntelVT-X is enabled . For example, on a Lenovo Thinkpad W530, in the BIOS select **Security** > **Virtualization** > **Enable Intel VT-X**.

Step 1: Install VMware

1. In a browser, open http://www.vmware.com/.
2. From the left navigation bar, Click the **DOWNLOADS** link.

3. In the new window, click the link for **Workstation Pro**.
4. You see two download options, one for Windows 64-bit and one for Linux 64-bit. From the appropriate option, click **Go to Downloads**.
5. Download the installation file.
6. Install the VMware product with the default options.

Step 2: Download the Ubuntu Operating System

1. In a browser, open https://releases.ubuntu.com/16.04.
2. Click the **64-bit PC (AMD64) desktop image** link.
3. Save the .iso file anywhere that is convenient. The file size is approximately 1.5 GB.

Step 3: Create a virtual machine

You use the Ubuntu iso file as the OS in the VMware image. You create a new virtual machine, and point it to the iso file you just downloaded.

1. If VMware Workstation is not open, open it now.
2. On the Home page, click the icon to Create a New Virtual Machine.
3. In the dialog that opens, accept the default selection (**Typical**) and click **Next**.
4. Click **Browse** and search for the iso file.
5. Select the iso file and click **Open**. If there is no problem with your downloaded file, you see the message **Ubuntu 64-bit 16.04.4 detected**.
   NOTE: If there is a problem, you see the message **Cannot read this file**, and the **Next** button is not active. Verify your download.
6. Click **Next**.
7. Enter the following information:

- Full name: **IBMCloudCourse**
- User name: **localuser**
- Password: **passw0rd** (and confirm)

8. Click **Next**.
9. For default name type **microservices**.
10. You can change the default location now if you want. Alternatively (as

the comment in the dialog says) you can modify this later, when the virtual machine has been created.

11. Click **Next**.
12. If you are sure that this virtual image will not be moved to another computer, you can choose to store it as a single file (thus improving performance).
13. Click **Next**.
14. Verify the information shown in the summary, then click **Finsh**.
15. Wait while the virtual machine is installed (files copied, language packs, and so on). There will be several progress bars.
16. When prompted, enter the password.
17. Remove some icons from the desktop that are not required for the exercises (there may be others; for clarity, remove what is not needed). Right-click and select **Unlock from Launcher** :

    - LibreOffice Writer
    - LibreOffice Calc
    - LibreOffice Impress
    - Ubuntu Software
    - Amazon
    - FloppyDisk

18. Click the **Search your computer** icon and type '**t**' in the search field.
19. Click the **Terminal** icon and verify that the Terminal opens.
20. In the navigation bar to the left, right-click the **Terminal** icon and select **Lock to Launcher**.
21. Type **exit** in the Terminal window. When it closes, verify that the icon remains on the navigation bar.

Step 4: Install Java JDK 1.8

The rest of the instructions apply to the image you have just created if you did that, or your native Linux operating system. Thus, 'In a browser...' means a browser on the image or on your native Linux system.

1. In a browser, open https://www.oracle.com
2. In the list of menu options, click **Downloads and Trials**.

3. Click **Java for Developers**.
4. In the list, look for the newest version that starts with **Java SE 8**. At the time of writing, the newest version was Java SE 8u171. If your version is newer, make the appropriate adjustment in the commands that follow. In that section, click the **DOWNLOAD** button under **JDK** (Note: your minor version may be different)
5. In the section that lists installation files for the various operating systems, click the **Accept License Agreement** radio button at the top.
6. Select the **tgz** file for Linux 64-bit (jdk-8u171-linux-x64.tar.gz).
7. **Save** the file.

You now have the compressed file on your image. The next step is to move it to the correct location and unzip it.

1. Create a directory for the file. In the Terminal, type the following:
   `sudo mkdir /usr/local/java`
2. Change to the download directory:
   `cd ~/Downloads`
3. You copy recursively (-r) the file in this directory to the location you want:
   `sudo cp -r jdk-8u171-linux-x64.tar.gz /usr/local/java`
4. Change to the new java directory:
   `cd /usr/local/java`
5. Verify that the tar.gz file was copied:
   `ls`
6. Unzip the file:
   `sudo tar xvf jdk-8u171-linux-x64.tar.gz`
7. Verify that the extraction was successful:
   `cd jdk1.8.0_171`
   `ls`
   You should see several directories, files, and zip files.

Step 5: Create JAVA_HOME environment variable

You add JAVA_HOME to the PATH by editing the profile file.

1. Open the /**etc**/**profile** file in an editor:
   `sudo gedit /etc/profile`

2. **Add** these lines to the bottom of the file:
   ```
   JAVA_HOME=/usr/local/java/jdk1.8.0_111
   PATH=$JAVA_HOME/bin:$PATH
   export JAVA_HOME
   export PATH
   ```
3. **Save** and **close** the profile file.

The final step is to provide the information about the new PATH to the system. You do this with three update-alternative commands for Java, javac, and javaws:

1. Update the java information (the final argument is the priority):
   ```
   sudo update-alternatives --install "/usr/bin/java" "java"
   "/usr/local/java/jdk1.8.0_171/jre/bin/java" 1
   ```
2. Update the compiler information:
   ```
   sudo update-alternatives --install "/usr/bin/javac" "javac"
   "/usr/local/java/jdk1.8.0_171/bin/javac" 1
   ```
3. Update the javaws information:
   ```
   sudo update-alternatives --install "/usr/bin/javaws" "javaws"
   "/usr/local/java/jdk1.8.0_171/bin/javaws" 1
   ```
4. Verify that your installation of Java is recognized. Type:
   ```
   java -version
   ```
5. You should see information about the java version, Java runtime, and Java HotSpot.

Step 6: Install GIT

1. Run the following command (At the message **Do you want to continue**, type **Y**):
   ```
   sudo apt-get install git
   ```
2. When the installation is complete, verify that it was successful by typing
   ```
   git -version
   ```

Step 7: Install curl

1. Run the following command (At the message **Do you want to continue**, type **Y**):
   ```
   sudo apt-get install curl
   ```
2. When the installation is complete, verify that it was successful by typing
   ```
   curl https://www.google.com
   ```

3. Verify that the response is HTML for the Google home page.
   NOTE: If the response is **The document has moved**, then curl was successfully installed, but the url is not correct. According to your geography, you need to change the extension.

Step 8: Install node.js

Ubuntu includes Node.js in its default repositories. The version that was used for this installation was **4.2.6**. Your version might be different.

1. Install it by typing the **apt install** command (At the message **Do you want to continue**, type **Y**):
   `sudo apt install nodejs-legacy`
2. Verify the installation:
   `node -v`
3. The response is the version number (for example, v4.2.6).
4. Install the Node Package Manager (npm) (at the message **Do you want to continue**, type **Y**):
   `sudo apt install npm`
5. Verify the installation:
   `npm version`
6. You should see a JSON object similar to this:

```
{ npm: '3.5.2',
  ares: '1.10.1-DEV',
  http_parser: '2.5.0',
  icu: '55.1',
  modules: '46',
  node: '4.2.6',
  openssl: '1.0.2g-fips',
  uv: '1.8.0',
  v8: '4.5.103.35',
  zlib: '1.2.8' }
```

Step 9: Install Docker

1. Type
   `docker`
   Since it is not installed yet, the response suggests that you install it

using apt install docker.io.

2. Type the following (at the message **Do you want to continue**, type **Y**):
   `sudo apt install docker.io`
3. Add the user to the docker group. The group may have been created automatically, in which case the first command will generate a warning message. You can ignore this and continue with the second command:
   `sudo groupadd docker`
   `gpasswd docker -a localuser`
4. When the installation has completed type
   `sudo docker run hello-world`
   You should see the response starting `Hello from Docker!`.
   NOTE: You may need to run the command a second time to see the correct response.

Step 10: Install the IBM Cloud (Bluemix) CLI

1. In a browser, go to https://console.bluemix.net/docs/cli/reference/bluemix_cli/get_started.html
2. Click the installer for Ubuntu 64-bit and follow the instructions to install the IBM Cloud CLI.
3. Type `bx -v` to return the version number.

Step 11: Install the IBM Cloud Container Service plugins

There are two plugins associated with the IBM Cloud Container Service that you will need for the labs. They are the plugin for the container service itself (`bx cs` commands) and the plugin for the container registry (`bx cr` commands).

1. Type
   `bx plugin install container-service -r Bluemix`
   `bx plugin install container-registry -r Bluemix`
2. Check that the plugins are installed by typing `bx plugin list`

Step 12: Install the Kubernetes kubectl CLI

1. An easy way to install the kubectl CLI on Ubuntu is using the snap package manager. Type `sudo snap install kubectl --classic` More

information can be found at https://kubernetes.io/docs/tasks/tools/install-kubectl.

Step 13: Install the MYSQL client

You can install MYSQL version 5.7 client using apt-get.

1. Type `sudo apt install mysql-client-5.7`

This completes the setup tasks for Linux.