

PIVOTAL CLOUD FOUNDRY

LAB ASSIGNMENTS

Prerequisites:

1. Use Pivotal Web Services (PWS). Go to <https://console.run.pivotal.io/register> and go through the registration process for a trial account. If you already have a PWS account you can skip this step.
2. Install Cloud Foundry Command Line Interface (CLI) Go to <https://github.com/cloudfoundry/cli/releases> or <https://console.run.pivotal.io/tools> and download the installer for your platform. Run the installer.

Lab 1:

Deploy an application using the CLI (Command Line Interface)

1. Push a simple Java/Spring application to Cloud Foundry
2. Observe and scale a running application.
3. You need to know the three URLs for either your Cloud Foundry installation or for PWS. Refer back to the presentation for details on the System Domain, Apps Domain and Apps Manager URLs.

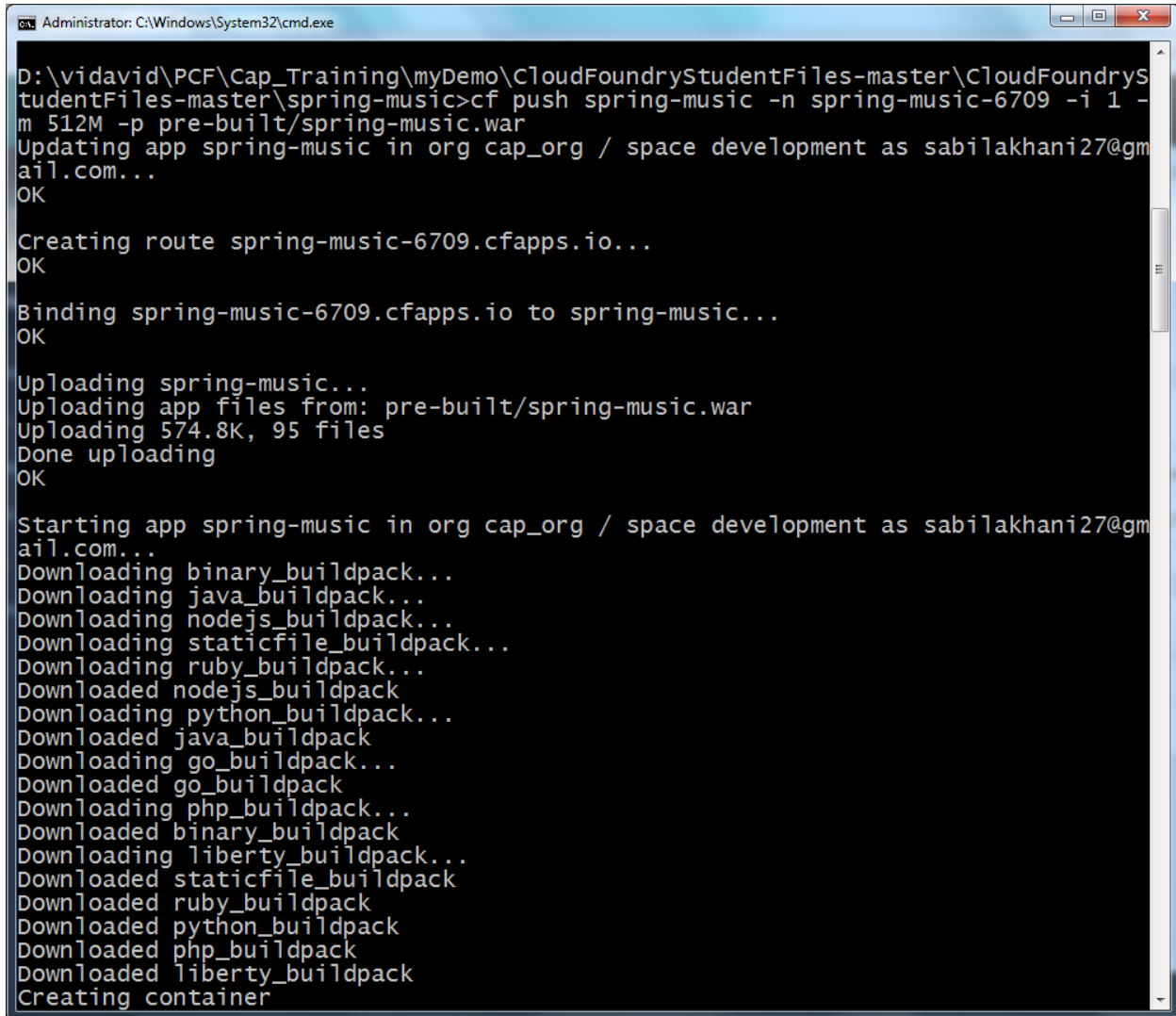
Steps:

1. Download the Spring-Music file.
2. In command prompt, locate the unzip folder (spring-music application) location.
3. Enter below command to push your application into Pivotal.

```
cf push spring-music -n spring-music-6709 -i 1 -m 512M -p pre-built/spring-music.war
```

4. Once you enter the above command in the command prompt you will be getting the following information, finally you could see the deployed application in Pivotal.

Note : If you get any error related to host number, please change into any other number like 6708, 6701 , etc.



```
Administrator: C:\Windows\System32\cmd.exe
D:\vidavid\PCF\Cap_Training\myDemo\CloudFoundryStudentFiles-master\CloudFoundryStudentFiles-master\spring-music>cf push spring-music -n spring-music-6709 -i 1 -m 512M -p pre-built/spring-music.war
Updating app spring-music in org cap_org / space development as sabilakhani27@gmail.com...
OK

Creating route spring-music-6709.cfapps.io...
OK

Binding spring-music-6709.cfapps.io to spring-music...
OK

Uploading spring-music...
Uploading app files from: pre-built/spring-music.war
Uploading 574.8K, 95 files
Done uploading
OK

Starting app spring-music in org cap_org / space development as sabilakhani27@gmail.com...
Downloading binary_buildpack...
Downloading java_buildpack...
Downloading nodejs_buildpack...
Downloading staticfile_buildpack...
Downloading ruby_buildpack...
Downloaded nodejs_buildpack
Downloading python_buildpack...
Downloaded java_buildpack
Downloading go_buildpack...
Downloaded go_buildpack
Downloading php_buildpack...
Downloaded binary_buildpack
Downloading liberty_buildpack...
Downloaded staticfile_buildpack
Downloaded ruby_buildpack
Downloaded python_buildpack
Downloaded php_buildpack
Downloaded liberty_buildpack
Creating container
```

```
Administrator: C:\Windows\System32\cmd.exe
Downloaded liberty_buildpack
Creating container
Successfully created container
Downloading app package...
Downloaded app package (21M)
Staging...
-----> Java Buildpack Version: v3.8.1 (offline) | https://github.com/cloudfoundry/java-buildpack.git#29c79f2
-----> Downloading Open Jdk JRE 1.8.0_91-unlimited-crypto from https://java-buildpack.cloudfoundry.org/openjdk/trusty/x86_64/openjdk-1.8.0_91-unlimited-crypto.tar.gz (found in cache)
    Expanding Open Jdk JRE to .java-buildpack/open_jdk_jre (1.0s)
-----> Downloading Open JDK Like Memory Calculator 2.0.2_RELEASE from https://java-buildpack.cloudfoundry.org/memory-calculator/trusty/x86_64/memory-calculator-2.0.2_RELEASE.tar.gz (found in cache)
    Memory Settings: -Xss228K -Xmx317161K -XX:MaxMetaspaceSize=64M -Xms317161K -XX:MetaspaceSize=64M
-----> Downloading Spring Auto Reconfiguration 1.10.0_RELEASE from https://java-buildpack.cloudfoundry.org/auto-reconfiguration/auto-reconfiguration-1.10.0_RELEASE.jar (found in cache)
-----> Downloading Tomcat Instance 8.0.36 from https://java-buildpack.cloudfoundry.org/tomcat/tomcat-8.0.36.tar.gz (found in cache)
    Expanding Tomcat Instance to .java-buildpack/tomcat (0.1s)
-----> Downloading Tomcat Lifecycle Support 2.5.0_RELEASE from https://java-buildpack.cloudfoundry.org/tomcat-lifecycle-support/tomcat-lifecycle-support-2.5.0_RELEASE.jar (found in cache)
-----> Downloading Tomcat Logging Support 2.5.0_RELEASE from https://java-buildpack.cloudfoundry.org/tomcat-logging-support/tomcat-logging-support-2.5.0_RELEASE.jar (found in cache)
-----> Downloading Tomcat Access Logging Support 2.5.0_RELEASE from https://java-buildpack.cloudfoundry.org/tomcat-access-logging-support/tomcat-access-logging-support-2.5.0_RELEASE.jar (found in cache)
Exit status 0
Staging complete
Uploading droplet, build artifacts cache...
Uploading build artifacts cache...
Uploading droplet...
Uploaded build artifacts cache (109B)
Uploaded droplet (74M)
Uploading complete
Destroying container
```

```

Administrator: C:\Windows\System32\cmd.exe
Uploading complete
Destroying container
Successfully destroyed container

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

App started

OK

App spring-music was started using this command `CALCULATED_MEMORY=$(($PWD/.java-buildpack/open_jdk_jre/bin/java-buildpack-memory-calculator-2.0.2_RELEASE -memorySizes=metaspace:64m..,stack:228k.. -memoryWeights=heap:65,metaspace:10,native:15,stack:10 -memoryInitials=heap:100%,metaspace:100% -stackThreads=300 -totMemory=$MEMORY_LIMIT) && JAVA_HOME=$PWD/.java-buildpack/open_jdk_jre JAVA_OPTS="-Djava.io.tmpdir=$TMPDIR -XX:OnOutOfMemoryError=$PWD/.java-buildpack/open_jdk_jre/bin/killjava.sh $CALCULATED_MEMORY -Daccess.logging.enabled=false -Dhttp.port=$PORT" exec $PWD/.java-buildpack/tomcat/bin/catalina.sh run`

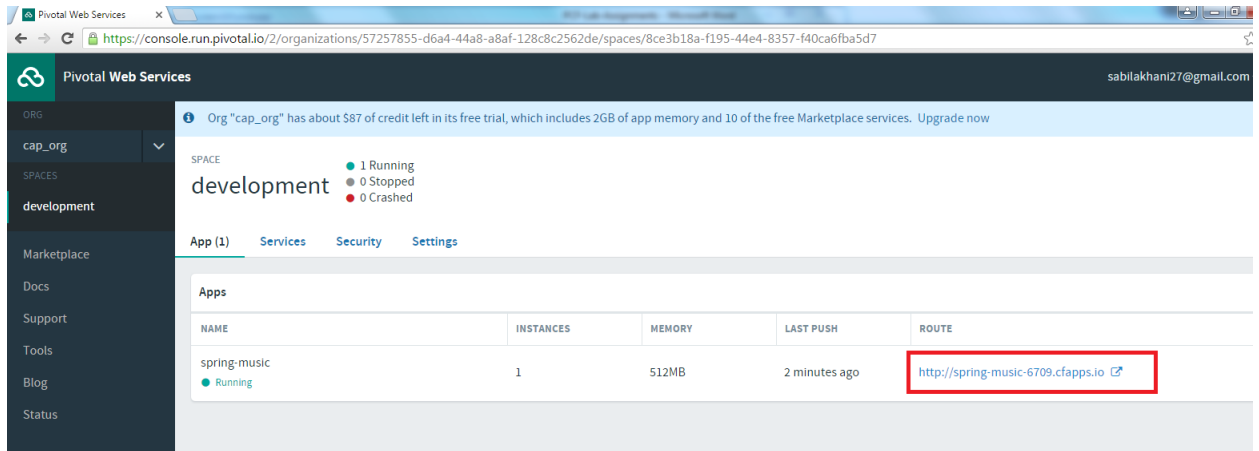
Showing health and status for app spring-music in org cap_org / space development as sabilakhani27@gmail.com...
OK

requested state: started
instances: 1/1
usage: 512M x 1 instances
urls: spring-music-6709.cfapps.io
last uploaded: Sun Aug 7 05:22:00 UTC 2016
stack: cflinuxfs2
buildpack: java-buildpack=v3.8.1-offline-https://github.com/cloudfoundry/java-buildpack.git#29c79f2 open-jdk-like-jre=1.8.0_91-unlimited-crypto open-jdk-like-memory-calculator=2.0.2_RELEASE spring-auto-reconfiguration=1.10.0_RELEASE tomcat-access-logging-support=...

state      since                cpu      memory      disk
details
#0  running  2016-08-07 10:52:56 AM  174.2%   347.9M of 512M  155.2M of 1G

```

- You can see the deployed application in the Pivotal Web services Browser also. As mentioned below screen shot. Your application host URL is highlighted in the below; if you click the URL it will open the Spring-music application which has been deployed in Pivotal Cloud. ☺



Push

Push the Spring Music application.

1. Change folders to the location where you downloaded the **Spring-Music** and go into the spring-music folder.
2. Run the `cf push` command to push this application to Cloud Foundry
 - a. Use any value you like for the application name. We suggest something brief and easy to type.
 - b. Use the `-n` option to specify the host. This value must be unique within the domain used by Cloud Foundry.
 - c. Use the `-p` option to specify the package to upload. This is a Java application, and a WAR file has already been built for you. It is located in “pre-built/spring-music.war”.
3. Observe the push log output. Can you identify the different stages of the process?

Observe

1. When the process is complete, identify the URL of the application. Open a browser and connect to this URL to make sure that the application comes up.
2. Run the `cf logs` command for your app.
3. Return to the browser, click the links, make changes to the data, then return to the command line and observe the generated log activity.

Conclusion:

1. Open the Apps Manager associated with your Pivotal CF instance. For Pivotal Web Services, this is at <http://run.pivotal.io>. For a private CF installation you will need to find out what the System Domain URL is - this is installation dependent. Once you know the System Domain, then the Apps Manager URI is `console.<system-domain>`.
2. Login and navigate to the organization and space that your application was pushed to.
3. Open the details page associated with your application. Change the number of instances to 4 and save your changes. From the browser, refresh the application's web page repeatedly and notice that the application does not go down while it is being scaled. Return to the command prompt and observe the activity that occurs as Cloud Foundry scales your application horizontally.
4. Back on the Apps Manager, scroll down to the "instances" section to observe the running instances.
5. Scale back to 1 instance.
6. Stop application, once you observed all the changes.

Congratulations, you have completed this exercise!

Prerequisites:

1. Use **Pivotal Web Services (PWS)**. Go to <https://console.run.pivotal.io/register> and go through the registration process for a trial account. If you already have a PWS account you can skip this step.
2. **Install an IDE (Eclipse or Spring Tool Suite)**
Some of the exercises work best when run from an IDE, such as Eclipse or Spring Tool Suite (STS), but again if you are not familiar with these tools, you may prefer to simply use the CLI and your favorite editor (such as Wordpad, Notepad++, JEdit, TextWrangler, XCode, gEdit or even `vi` or `emacs`). If so you are done, please ignore the rest of these instructions.
 - a. For Eclipse, Go to <https://www.eclipse.org/downloads/> and download Eclipse. Select the latest version, download it, and install it. If you already have Eclipse 4.3 or higher, you can skip this step.
 - b. For STS Go to <http://spring.io/tools/sts> and download STS. Select the latest version. If you already have STS version 3.6.0 or higher, you can skip this step.

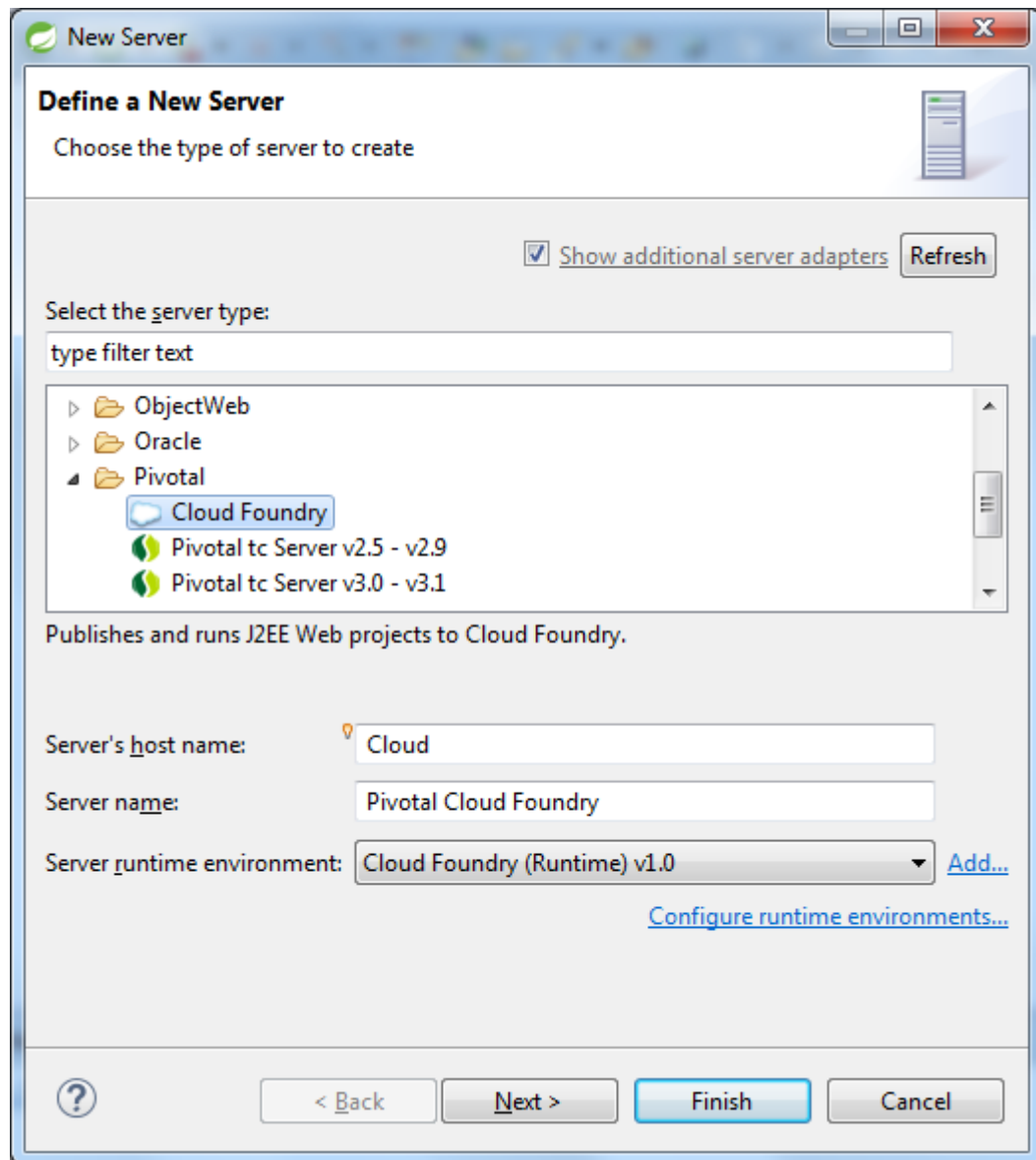
Lab 2:

Deploy an application using the IDE(STS / Eclipse)

Steps:

3. **Install IDE Support for Cloud Foundry**
 - a. Eclipse / STS: Install Cloud Foundry Integration
 - i. If using Eclipse or STS, select Eclipse Marketplace from the Help menu.
 - ii. In the marketplace dialog, enter "Cloud Foundry" in the Find field, and click Go.
 - iii. Find the entry for "Cloud Foundry Integration for Eclipse" in the search results. The most recent versions of STS have Cloud Foundry installed by default. There are three possibilities:
 - A. The Cloud Foundry plugin is not installed, click the Install button now.
 - B.If Cloud Foundry is already installed there will be two buttons Update and Uninstall. If the Update button is disabled there is nothing more to do. Go to the next section "Add a "Cloud Foundry" Server" below.
 - C.If the Update button is enabled, you may wish to update now. Click Update to continue. If you do not wish to update, go to the next section "Add a "Cloud Foundry" Server" below.
 - iv. In the next dialog just click Confirm
 - v. Finally, accept the license agreement and click Finish. The install/update process will take a bit of time, and you will be prompted to restart Eclipse to make the changes take effect.

- vi. Full details: Install to Eclipse
- 4. Add a “Cloud Foundry” Server
 - a. Eclipse or STS
 - i. Within Eclipse or STS, Select the “Servers” view; this is typically located in the lower left corner of the workspace. (If it is not present, go to Window / Show View / Other / Server / Servers.)
 - ii. Within the Servers view, right click, select New / Server and browse to Pivotal / Cloud Foundry. Enter the email and password for your account.
 - iii. By default the URL specifies PWS (run.pivotal.io). If using a different installation, click on “Manage Cloud”, then “Add” and then specify a name for your CF instance (call it whatever you want) and its API URL. If you are not sure of the API URL, you will need to obtain it from an administrator familiar with your CF setup.
 - iv. If prompted, it is fine to take the default values for Organization / Space or make a selection from the choices offered. If unsure, accept the defaults. We will explain "organization" and "space" more fully later in the course.
 - v. Click Finish. We now have a ‘server’ in Eclipse/STS that points to a Cloud Foundry environment.
 - vi. Look at the below screen shots for your guidance:



New Server

Cloud Foundry Account

Press 'Validate Account', 'Next', 'Finish' to validate credentials.

Account Information

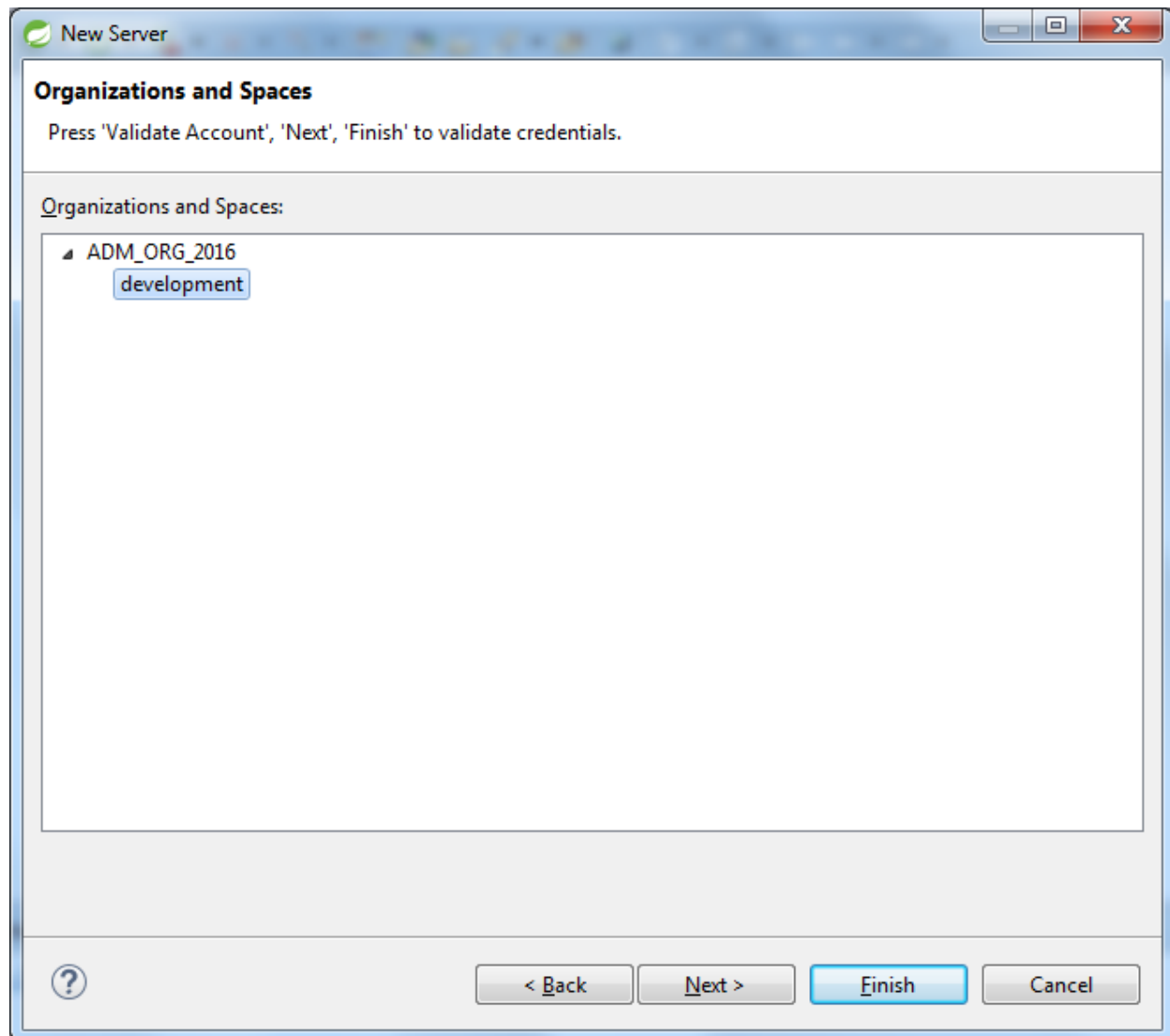
Email:

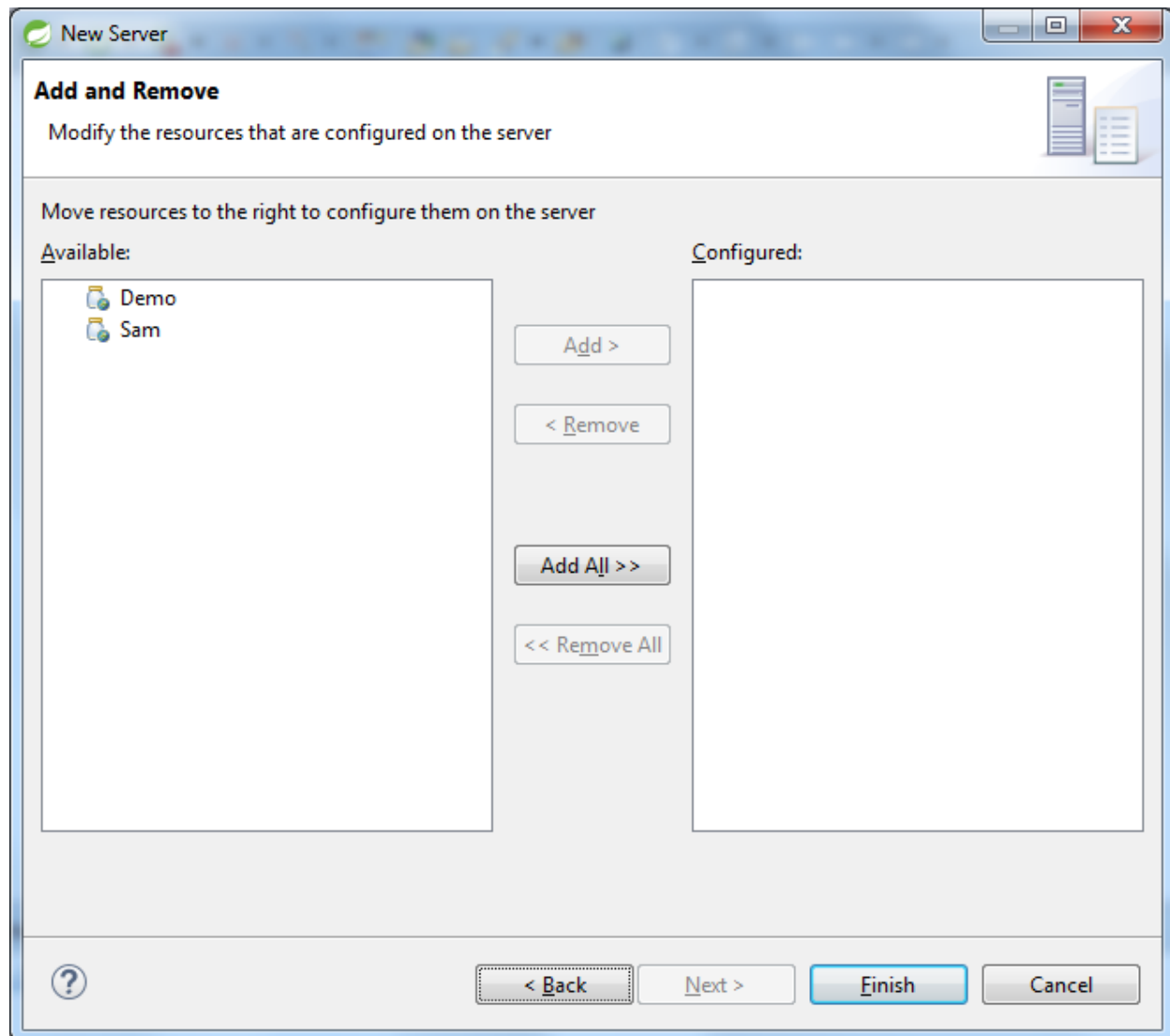
Password:

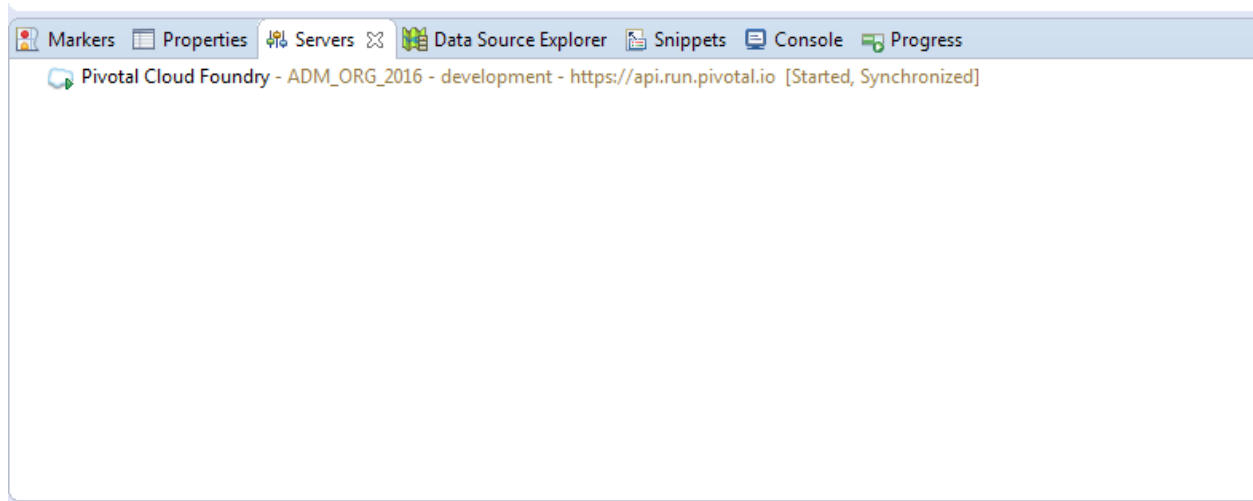
URL: [Manage Cloud...](#)

[Validate Account](#) [Register Account...](#) [Sign Up](#)

[? Help](#) [< Back](#) [Next >](#) [Finish](#) [Cancel](#)







5. Import projects into your IDE
 - a. Eclipse or STS
 - i. From the project explorer, right click, import, Maven, existing maven projects
 - ii. Navigate to the folder and Finish.
 - iii. The projects will be imported.
 - iv. Note that depending on the version of Eclipse / STS, your version of Java, and other factors, you may encounter errors. These will need to be fixed on a case-by-case basis before proceeding.

Congratulations, you have completed this exercise!

Console Log for Lab 2:

```
Checking application - cf-spring-mvc-demo
Generating application archive - cf-spring-mvc-demo
Pushing application - cf-spring-mvc-demo
Creating application - cf-spring-mvc-demo
Application successfully pushed
Starting application - cf-spring-mvc-demo
Updated app with guid 5051caf0-bd03-4bc4-8602-c8b0f2d8efda ({"state"=>"STARTED"})
Downloading java_buildpack...
Downloaded java_buildpack
Creating container
Successfully created container
```

```

Downloading app package...
Downloaded app package (23.6M)
Staging...
-----> Java Buildpack Version: v3.8.1 (offline) |
https://github.com/cloudfoundry/java-buildpack.git#29c79f2
    Expanding Open Jdk JRE to .java-buildpack/open_jdk_jre (1.0s)
-----> Downloading Open JDK Like Memory Calculator 2.0.2_RELEASE from https://java-
buildpack.cloudfoundry.org/memory-calculator/trusty/x86_64/memory-calculator-
2.0.2_RELEASE.tar.gz (found in cache)
    Memory Settings: -Xmx317161K -Xms317161K -XX:MetaspaceSize=64M -
XX:MaxMetaspaceSize=64M -Xss228K
-----> Downloading Container Customizer 1.0.0_RELEASE from https://java-
buildpack.cloudfoundry.org/container-customizer/container-customizer-
1.0.0_RELEASE.jar (found in cache)
-----> Downloading Spring Auto Reconfiguration 1.10.0_RELEASE from https://java-
buildpack.cloudfoundry.org/auto-reconfiguration/auto-reconfiguration-
1.10.0_RELEASE.jar (found in cache)
-----> Downloading Tomcat Instance 8.0.36 from https://java-
buildpack.cloudfoundry.org/tomcat/tomcat-8.0.36.tar.gz (found in cache)
    Expanding Tomcat Instance to .java-buildpack/tomcat (0.1s)
-----> Downloading Tomcat Lifecycle Support 2.5.0_RELEASE from https://java-
buildpack.cloudfoundry.org/tomcat-lifecycle-support/tomcat-lifecycle-support-
2.5.0_RELEASE.jar (found in cache)
-----> Downloading Tomcat Logging Support 2.5.0_RELEASE from https://java-
buildpack.cloudfoundry.org/tomcat-logging-support/tomcat-logging-support-
2.5.0_RELEASE.jar (found in cache)
-----> Downloading Tomcat Access Logging Support 2.5.0_RELEASE from https://java-
buildpack.cloudfoundry.org/tomcat-access-logging-support/tomcat-access-logging-
support-2.5.0_RELEASE.jar (found in cache)
[Application Running Check] - Checking if application is running - cf-spring-mvc-
demo. Please wait...
Staging complete
Exit status 0
Uploading build artifacts cache...
Uploading droplet, build artifacts cache...
Uploading droplet...
Uploaded build artifacts cache (108B)
Uploaded droplet (76.6M)
Uploading complete
Destroying container
Creating container
Successfully created container
Successfully destroyed container
Starting health monitoring of container
[CONTAINER] org.apache.coyote.http11.Http11NioProtocol      INFO    Initializing
ProtocolHandler ["http-nio-8080"]
[CONTAINER] org.apache.catalina.core.StandardEngine        INFO    Starting
Servlet Engine: Apache Tomcat/8.0.36
[CONTAINER] org.apache.catalina.core.StandardService       INFO    Starting
service Catalina

```

```
[CONTAINER] org.apache.catalina.startup.HostConfig      INFO    Deploying web
application directory /home/vcap/app/.java-buildpack/tomcat/webapps/ROOT
[CONTAINER] org.apache.catalina.startup.Catalina        INFO    Initialization
processed in 497 ms
[CONTAINER] org.apache.jasper.servlet.TldScanner       INFO    At least one
JAR was scanned for TLDs yet contained no TLDs. Enable debug logging for this logger
for a complete list of JARs that were scanned but no TLDs were found in them.
Skipping unneeded JARs during scanning can improve startup time and JSP compilation
time.
[CONTAINER] ing.AutoReconfigurationServletContainerInitializer INFO    Initializing
ServletContext with Auto-reconfiguration ApplicationContextInitializers
[CONTAINER] lina.core.ContainerBase.[Catalina].[localhost].[/] INFO    Spring
WebApplicationInitializers detected on classpath:
[com.gopivotal.cf.workshop.WebApplication@15dbcdcb]
2016-08-08 15:36:23.438 INFO 19 --- [ost-startStop-1]
o.c.r.s.CloudProfileApplicationListener : Adding 'cloud' to list of active profiles
/\ / __' _ _ _ _ ( _ ) _ _ _ _ \ \ \ \
.
\ /  _ ) | | _ | | | | ( _ | _ ) _ )
( ( ) \ _ | ' _ | | | | _ \ _ | \ \ \ \
' | _ | . _ | | | | | _ _ , | / / / /
=====|_|=====|_|_/=//_/_/_/
:: Spring Boot ::          (v1.1.7.RELEASE)
2016-08-08 15:36:24.189 INFO 19 --- [ost-startStop-1]
pertySourceApplicationContextInitializer : Adding 'cloud' PropertySource to
ApplicationContext
2016-08-08 15:36:24.236 INFO 19 --- [ost-startStop-1]
nfigurationApplicationContextInitializer : Adding cloud service auto-reconfiguration
Ect.....
```