

**Pivotal Cloud Foundry & Cloud-Native Architecture Workshop Prerequisites**

[PCF Connectivity Verification](#h.m67momio3j46)

[Local Machine Setup](#h.ze1vyoyflker)

[Required](#h.23p6dp3vnzh7)

[Optional (but recommended)](#h.9fatnp8f23vr)

[Alternative: Windows VM](#h.g0b394k6zof4)

[Spring Labs Setup](#h.9tazzjs82tom)

[Spring Boot Labs Setup](#h.82do8vj9ff7w)

[Spring Cloud Services Labs Setup](#h.svod8bteouvb)

[Recap: Local Labs Directories](#h.u2dqc91vr9t1)

# 

# 

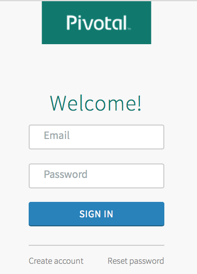
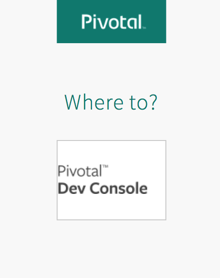
# PCF Connectivity Verification

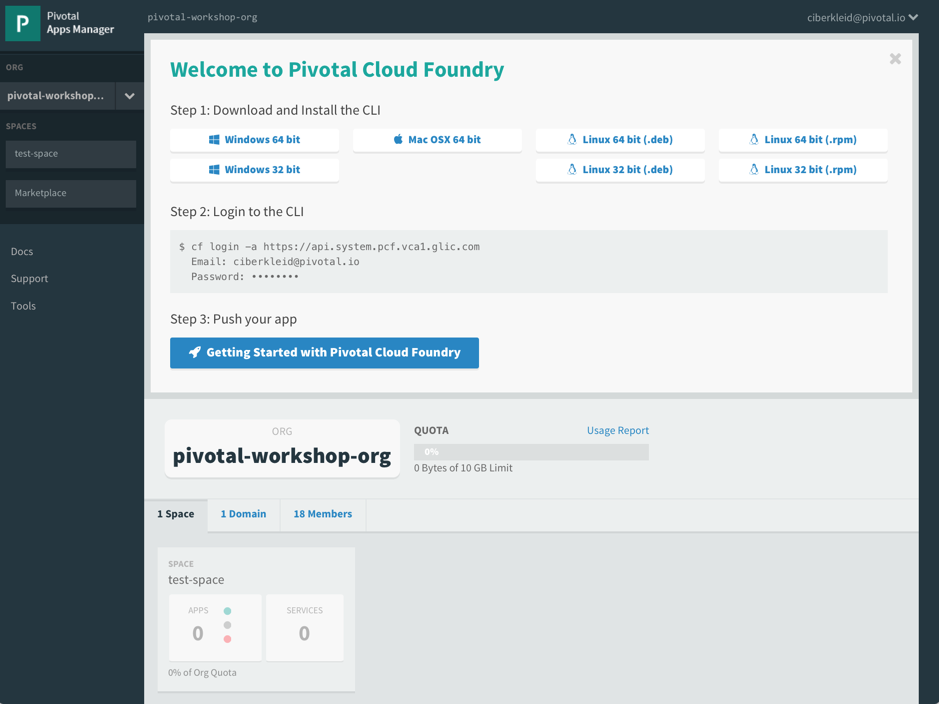
1. Log in to **Apps Manager**, the Pivotal Cloud Foundry Developer UI

URL: <https://login.system.appledevs.com/login>

Username: your email address

Password: password



1. Download and install the Cloud Foundry command line interface (cf CLI). The link to the binary/installer is in the ‘Welcome to Pivotal Cloud Foundry’ frame or in the ‘Tools’ section of the UI
2. From a command/terminal window, test the CLI using ‘cf --version’ or ‘cf --help’



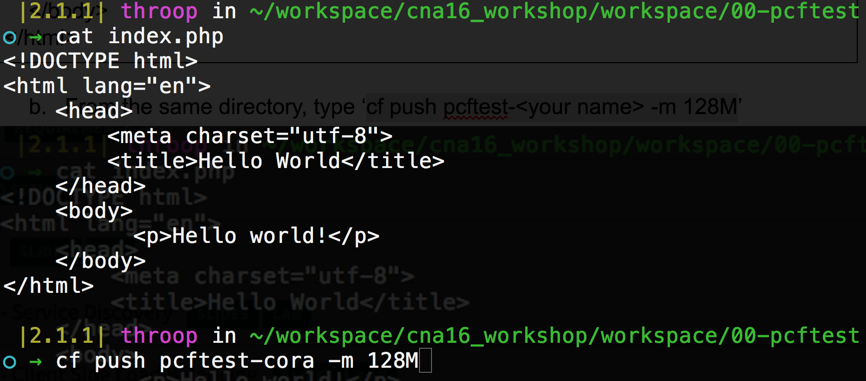
1. Use the cf CLI to log in to Pivotal Cloud Foundry

Command: cf login --skip-ssl-validation -a https://api.system.appledevs.com Credentials: same as those used for Apps Manager

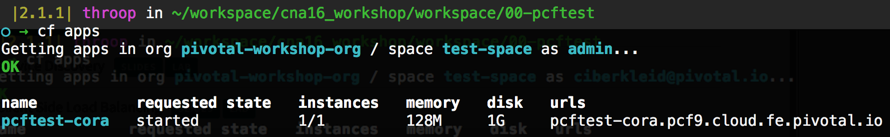
1. Note that you are targeting the same org (**tau-workshop**) and space (the part of your email id before the @) as you see through Apps Manager. The cf CLI and Apps Manager are two clients to the REST interface of the same instance of Pivotal Cloud Foundry.
2. Using the cf CLI, push an app as follows:
   1. On your local machine, create a file called index.php and copy the following text into it:

|  |
| --- |
| <!DOCTYPE html>  <html lang="en">  <head>  <meta charset="utf-8">  <title>Hello World</title>  </head>  <body>  <p>Hello world!</p>  </body>  </html> |

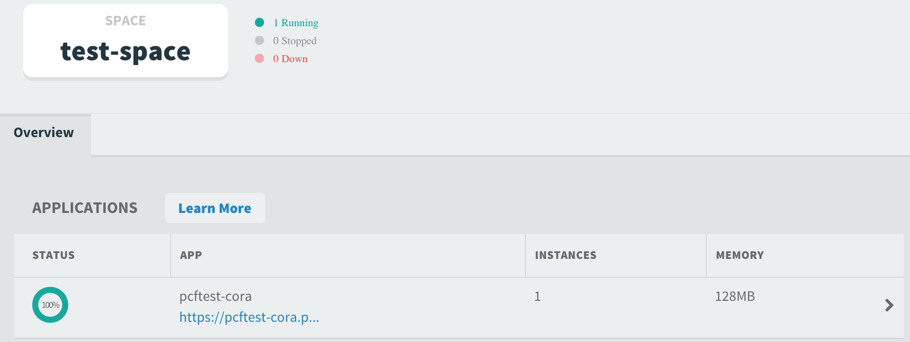
* 1. From the same directory, type ‘cf push pcftest-<your name> -m 128M’.



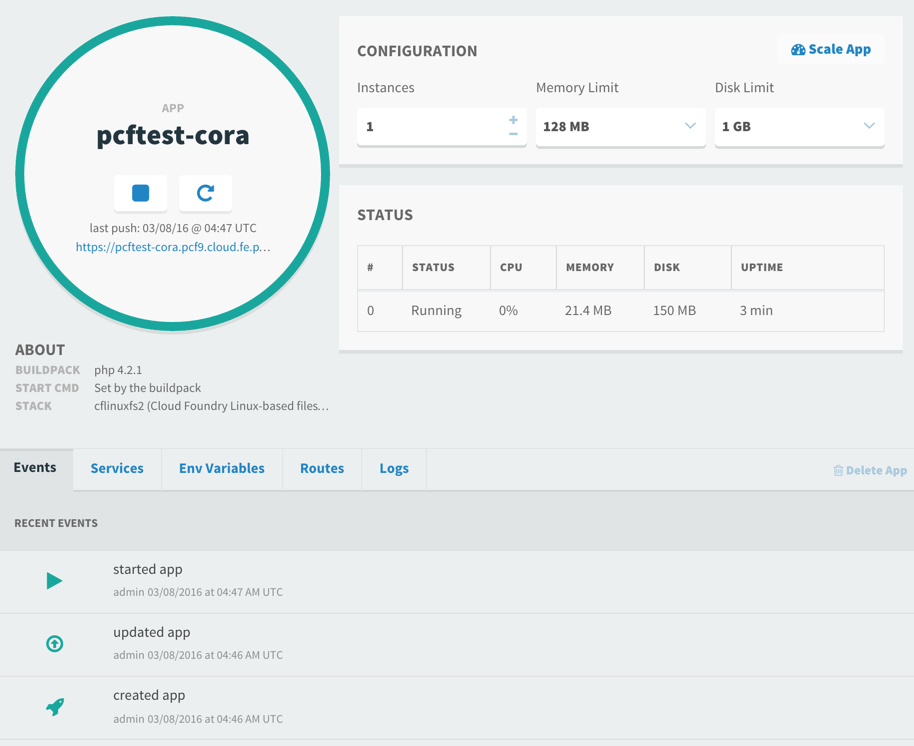
* 1. ‘cf push’ will upload your PHP application to Pivotal Cloud Foundry, which will stage and start your app. When it is done, type ‘cf apps’ to get a snapshot of the state of all of the apps in the space ‘test-space’:



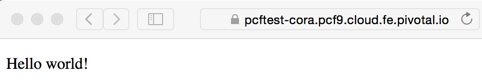
* 1. Return to the Apps Manager UI. Click on your space and verify that you see your app deployed.



* 1. Click on the arrow (**>**) to the right of your app to see detailed information about your app.



* 1. Click on the link to your app and verify that your app is working as expected



1. Congrats! You’ve pushed your first app to Pivotal Cloud Foundry! If you make changes to your HTML test app, simply push the app again using the CLI. Your changes will be uploaded and your app will be automatically updated and restarted.

# 

# 

# Local Machine Setup

## Required

1. 4GB Memory
2. Installation of [JDK 1.8](http://www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.html)
3. Installation of [Maven](https://maven.apache.org/)
4. Installation and experience with [git](https://git-scm.com/) and
5. [Github Account](https://github.com/)
6. Installation of [curl](http://curl.haxx.se/download.html)
7. Installation of [cf](https://apps.system.pcf.vca1.glic.com/tools)

## Optional (but recommended)

1. [SpringSource Tool Suite](https://spring.io/tools) (Eclipse, IntelliJ, or other of your IDE of your choice is fine)
2. [Json Formatter](https://chrome.google.com/webstore/detail/json-formatter/bcjindcccaagfpapjjmafapmmgkkhgoa?hl=en)

## Alternative: Windows VM

An alternative to installing and configuring the software on your local machine is to use the Windows VM. Please email [unadkarni@pivotal.io](mailto:unadkarni@pivotal.io) if you would like instructions for setting up a Windows VM on AWS for the workshop.

# 

# 

# Spring Labs Setup

## Spring Boot Labs Setup

1. Log into [GitHub](https://github.com/) (create an account if necessary)
2. Open the Spring Boot Labs repository

<https://github.com/pivotal-enablement/spring-boot-labs>

1. Fork the repo to your account



1. Your fork of the repo will display on the screen. From your fork of the repo, copy the HTTPS clone URL. It will be formatted as:

https://github.com/<YOUR-ACCOUNT>/spring-boot-labs.git

1. On your local machine, create a directory in which to download your fork of the labs repo. The directory can be called repos.
2. From a command/terminal window, go to your repos directory and clone your fork of the labs repo using the HTTPS clone URL:

**git clone** <https://github.com/><YOUR-ACCOUNT>/spring-boot-labs.git

This will create a spring-boot-labs directory inside of your repos directory. This directory will be used for your work for the Spring Boot portion of the workshop.

1. Optionally, import the applications in spring-boot-labs into your IDE (STS or other).

For STS, select File > Import…

Then select Maven > Existing Maven Projects

On the Import Maven Projects page, browse to your spring-boot-labs directory.

Make sure all projects are selected and click Finish.

## Spring Cloud Services Labs Setup

Repeat the steps in the Spring Boot Labs Setup section for the Spring Cloud Services Labs repo:

<https://github.com/pivotal-enablement/spring-cloud-services-labs>

## Recap: Local Labs Directories

The following convention will be used throughout the labs to refer to your cloned git repos:

* repos → $REPOS\_HOME
* repos/spring-boot-labs → $SPRING\_BOOT\_LABS\_HOME
* repos/spring-cloud-labs → $SPRING\_CLOUD\_LABS\_HOME

It is not necessary to create these environment variables; this convention is only a convenience for the lab instructions.