Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING

Pivotal Cloud Foundry

Table of Contents

Getting Started with Command Line Interface

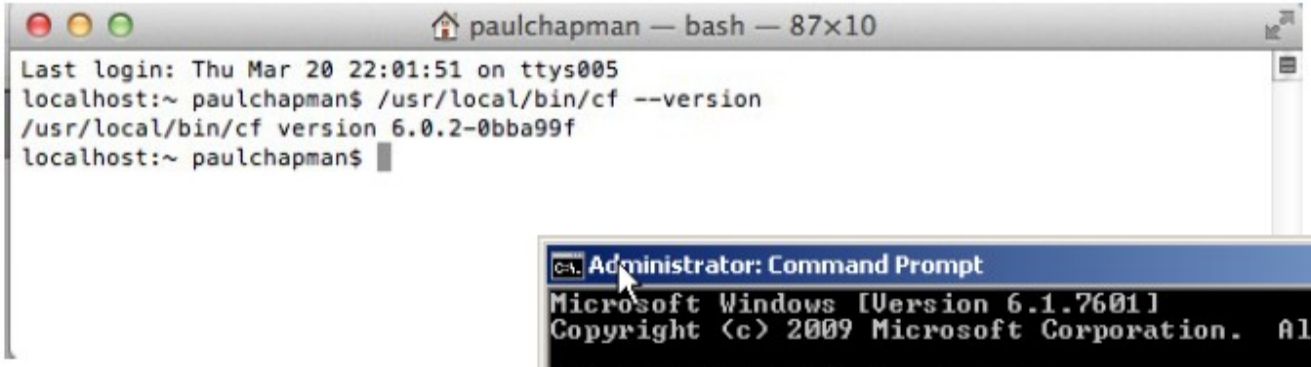
The Command Line Interface

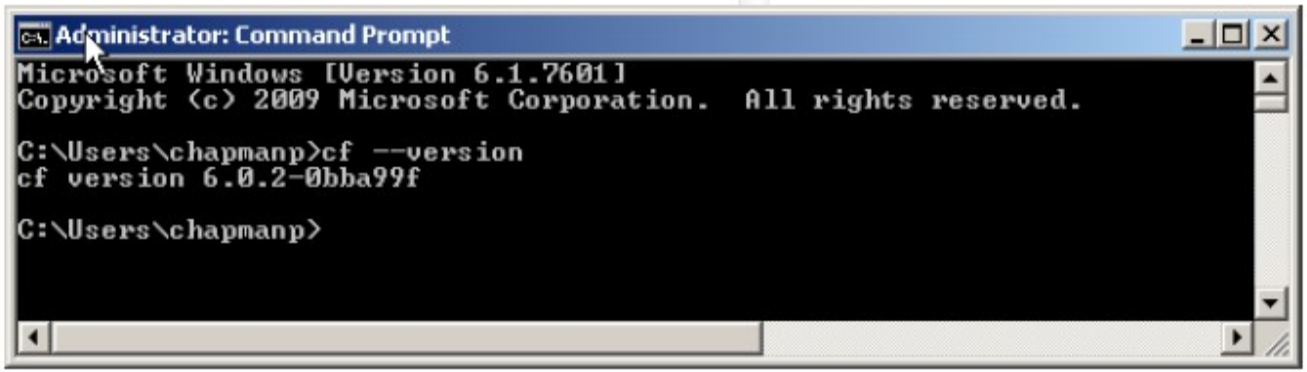
- Several interface options exits for Cloud Foundry
  - Command Line Interface (CLI)
  - Web-based Application Manager Console
  - Eclipse / STS plug-in
- Primary access is done via the CLI
  - Make sure you have it installed
    - Installation was covered in the “welcome” module

Test the CLI Utility

- It is called **cf**
  - Open a command/Shell window
  - At the prompt type : **cf --version**

- Version *must* be **6** or more
- *Must* remove any earlier (Ruby) version





Getting Help

- Get help at any time via **cf help**

- Or **cf help <command>**

DO NOW – Get Help on a Command

- Perform these steps on your computer:
  - Open a command prompt
  - Issue the **cf help** command
  - Get help on the login command: **cf help login**
- Answer these questions:
  - What option do you use to specify username?
  - Is specifying the password option encouraged?

Login

Login to Cloud Foundry

- Need to tell cf
  - What cloud foundry instance you are using
  - What your account details are
  - Use **cf login**

Color highlighting
MacOS, Linux only

```
> cf login -a api.run.pivotal.io -u <username>
API endpoint: api.run.pivotal.io
Authenticating...
OK

Targeted org Cloud Foundry Course

Targeted space development

API endpoint: https://api.run.pivotal.io (API version: 2.0.0)
User:         qzqz2020@yahoo.com.au
Org:          Cloud Foundry Course
Space:        development
```

Will prompt for anything you don't specify
No -p? Prompts for password

Cloud Foundry URLs

- To access CF you need to know 3 URLs
  - API Endpoint
    - Identifies your CF instance
    - Used to deploy applications, manage spaces, routes…
    - The cf utility makes *RESTful* requests to this URL

      Actually to the Cloud Controller
  - Apps Manager
    - Application management dashboard
    - Pivotal CF only
  - Apps Domain
    - Used to access deployed applications
    - May be same as System Domain

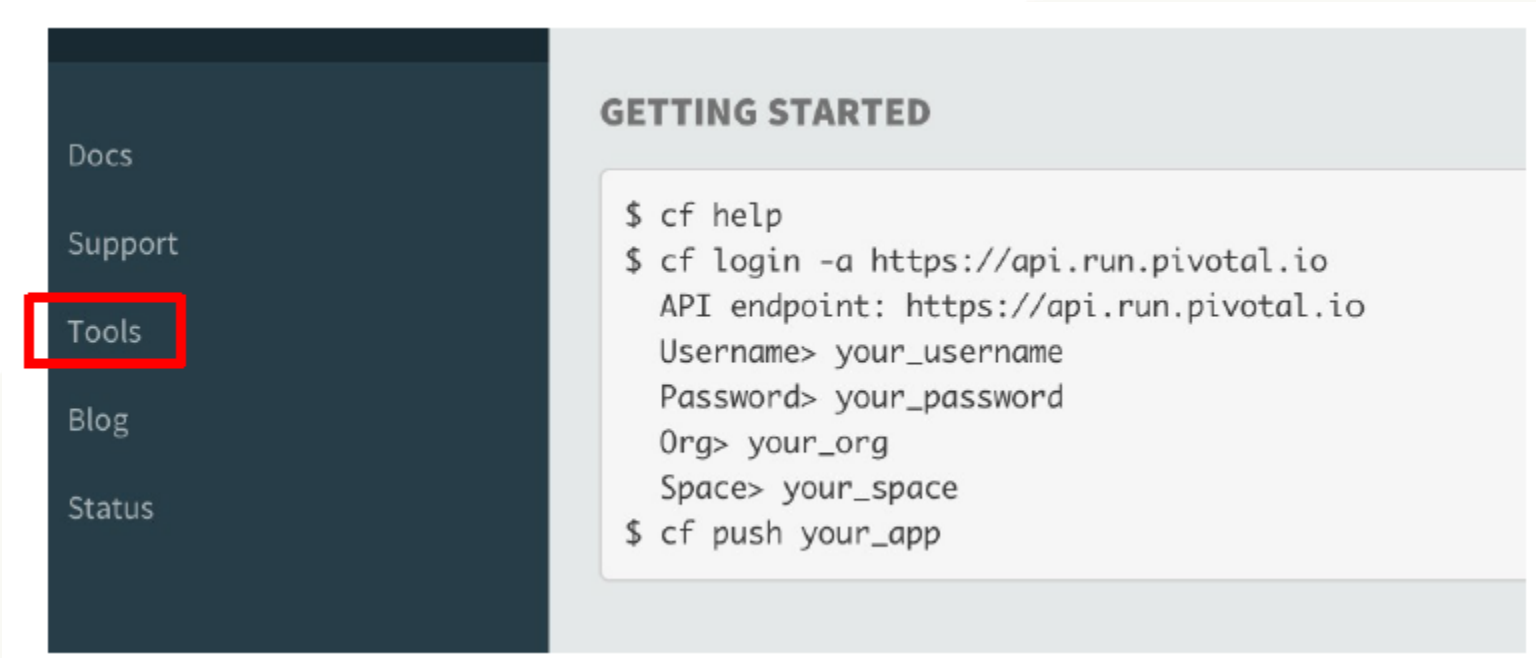
Cloud Foundry URLs

For simplicity, most examples
In this session show PWS URLs

- System & App domains defined when CF was installed
- If using PWS
  - System domain: **run.pivotal.io**
    - API Endpoint: **api.run.pivotal.io**
    - Apps Manager: **console.run.pivotal.io**
  - Apps domain: **cfapps.io**
- Your own CF installation
  - System domain: **<your-cf-system-domain>**
    - API Endpoint: **api. <your-cf-system-domain>**
    - Apps Manager: **console. <your-cf-system-domain>**
  - Apps domain: **<your-cf-system-domain>**

Finding the API Endpoint URL

- URL of Cloud Controller in our Cloud Foundry instance
  - On Apps Manager home-page on first login
  - Or click Tools
  - Shows how to run **cf login**, including the API Endpoint



DO NOW – Login

- Perform these steps on your computer
  - Login with **cf login** command
    - Specify CF instance using **–a <API-URL>**

      For PWS: **-a api.run.pivotal.io**
    - Specify email / password
    - If prompted, select an organization and space

```
$> cf login -a <API-URL> -u <your-email-or-username>
API endpoint: api.run.pivotal.io
...
```

  - Firewall issues?

http://docs.cloudfoundry.org/devguide/installcf/http-proxy.html

The .cf Directory

- **cf** creates a **.cf** directory in your home directory
  - Stores context, logs, crash reports …
  - Remembers your CF API Endpoint
    - Don't need to specify **–a** option at next login

```
localhost:dev$ ls -l ~/.cf
total 48
-rw-------  1 paulchapman  staff   2491 15 Aug 14:06 config.json
-rw-r--r--  1 paulchapman  staff  11737 29 Nov  2013 crash
drwxr-xr-x  3 paulchapman  staff    102 12 Sep  2013 logs
-rw-r--r--  1 paulchapman  staff     26 12 Sep  2013 target
-rw-r--r--  1 paulchapman  staff   2084 29 Nov  2013 tokens.yml
```

DO NOW - .cf folder

- Perform these steps on your computer:
  - Find the **.cf** folder / directory on your computer:
    - You won't (yet) have all the files shown on previous slide
  - Open the **config.json** file, observe the contents

Current Targets

- When your first login you see output like this
  - Notice it shows current organization and space
  - At any time, run **cf target** to get same information

```
API endpoint: https://api.run.pivotal.io (API version: 2.6.0)
User:         pchapman@pivotal.io
Org:          pivotaledu
Space:        development
```

- By default your organization only has one space
  - Development
- **Note:** on PWS you are setup as your own oraganization

Viewing Organization

- Commands
  - **cf orgs** — All orgs for current user
  - **cf org <org-name>** — Shows specified org

```
>$ cf org pivotaledu
Getting info for org myorg as user@somedomain.com
OK

pivotaledu:
    domains:          cfapps.io
    quota:            paid (10240M memory limit, Unlimited instance
                        memory limit, 1000 routes, -1 services,
                        paid services allowed)
    spaces:           development, production, staging
    space quotas:
>$
```

Managing Space

- To see all the spaces in an organization
  - **cf spaces**
- Create a new space (in current organization by default)
  - **cf create-space <space-name>**
  - **cf create-space <space-name> -o <org-name>**
- Use ***target*** command to change space (or organization)
  - **cf target –s <space-name>**
  - **cf target –o <org-name>**

Deploy An Application

Deploy Using the CLI

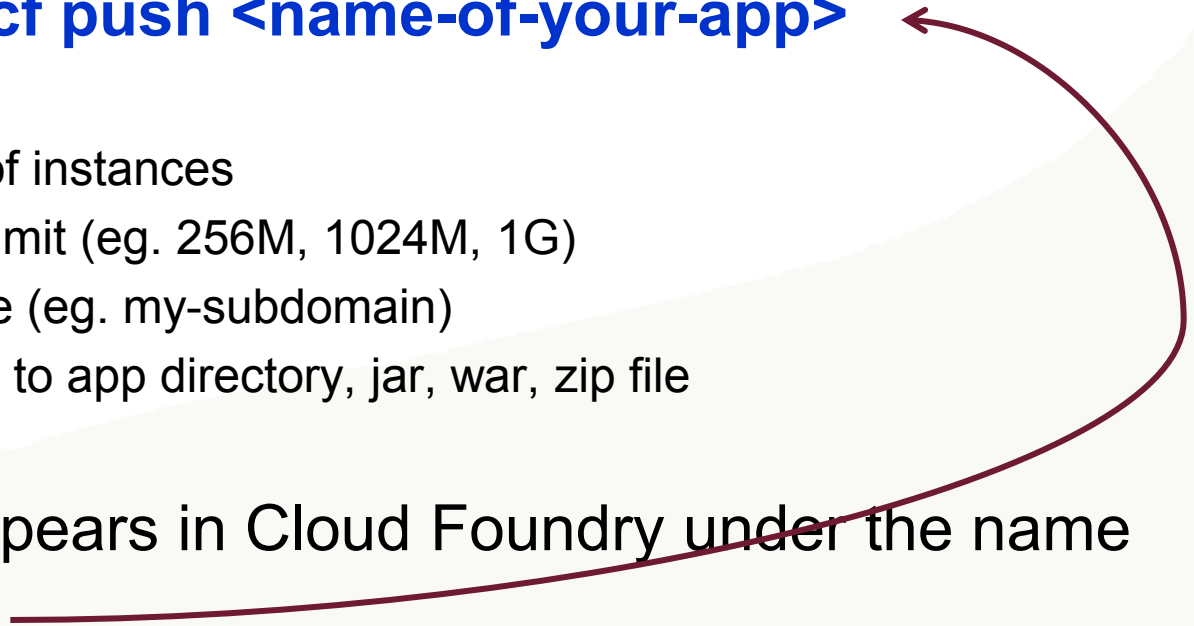
- You need a deployable application
  - For example with Java: a jar or war
    - Ant, maven or Gradle built-tools can make it for us
    - Cloud Foundry doesn't care how you build your application
  - Other languages (Ruby, Node.js, etc …) : the course will do

The cf push Philosophy

- Onsi Fakhouri (Cloud Foundry PM)
  - Here is my source code
  - Run it on the cloud for me
  - I do not care how
- The architecture of CF is fascinating
  - And we will cover it
  - But ultimately irrelevant
- I just want to push an application
  - I no longer need to know: how that happens, how it is packaged or how it is run?

Deploy push to Cloud Foundry

- Deploy by running **cf push <name-of-your-app>**
  - Many options
    - -i Number of instances
    - -m Memory limit (eg. 256M, 1024M, 1G)
    - -n Hostname (eg. my-subdomain)
    - -p local path to app directory, jar, war, zip file
- Your application appears in Cloud Foundry under the name you specify here

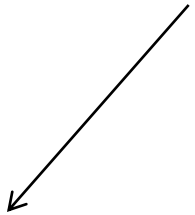


Domains and URLs

- Every CF instance is assigned a domain at installation
  - Known as the *Apps Domain*
  - For PWS this is **cfapps.io**
- When you deploy, your application gets a unique route (URL) to access it: **hostname + app domain name**
  - By default, hostname = application name
  - Make sure hostname is **unique**
    - cf push returns an HTTP 400 error if not
- PWS example:
  - **cf push spring-music …**
  - gets route: **spring-music.cfapps.io**

Example of Using cf push

Specify unique sub-domain by adding numbers, initials …



- Fully specified (recommended)

```
cf push spring-music -i 1
                     -m 512M
                     -n spring-music-678
                     -p build/libs/spring-music.war
```

  - Deploys war file (specify path if needed)
  - 1 instance, 512M memory
  - Name: **spring-music**
    - Appears as **spring-music** in Cloud Foundry
  - Hostname: **spring-music-678**
    - Creates URL (PWS): **spring-music-678.cfapps.io**

What Happens?

- cf connects to Cloud Foundry using your credentials
- It ‘pushes’ your application to CF and tells it to deploy it
  - The whole application is uploaded – takes a while
  - CF “stages” your application
    - Recognizes Java WAR file, prepares a “droplet” with a JRE and Tomcat server
    - “Droplet” is deployed to a container and starts running
    - All requests to the deployed URL route to your application
- Whole processed logged on screen
  - Next 3 slides

What Happens – 1

URL: spring-music-678.cfapps.io

```
cf push spring-music -n spring-music-678 -i 1 -m 512M
                     -p pre-built/spring-music.war
```

```
> cf push spring-music -n spring-music-678 -p build/libs/spring-music.war -i 1 -m 512M

Updating app spring-music in org your-org / space development as your-id@company.io...
OK

Using route spring-music-678.cfapps.io
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 your-org / space development as your-id@company.io...
...
```

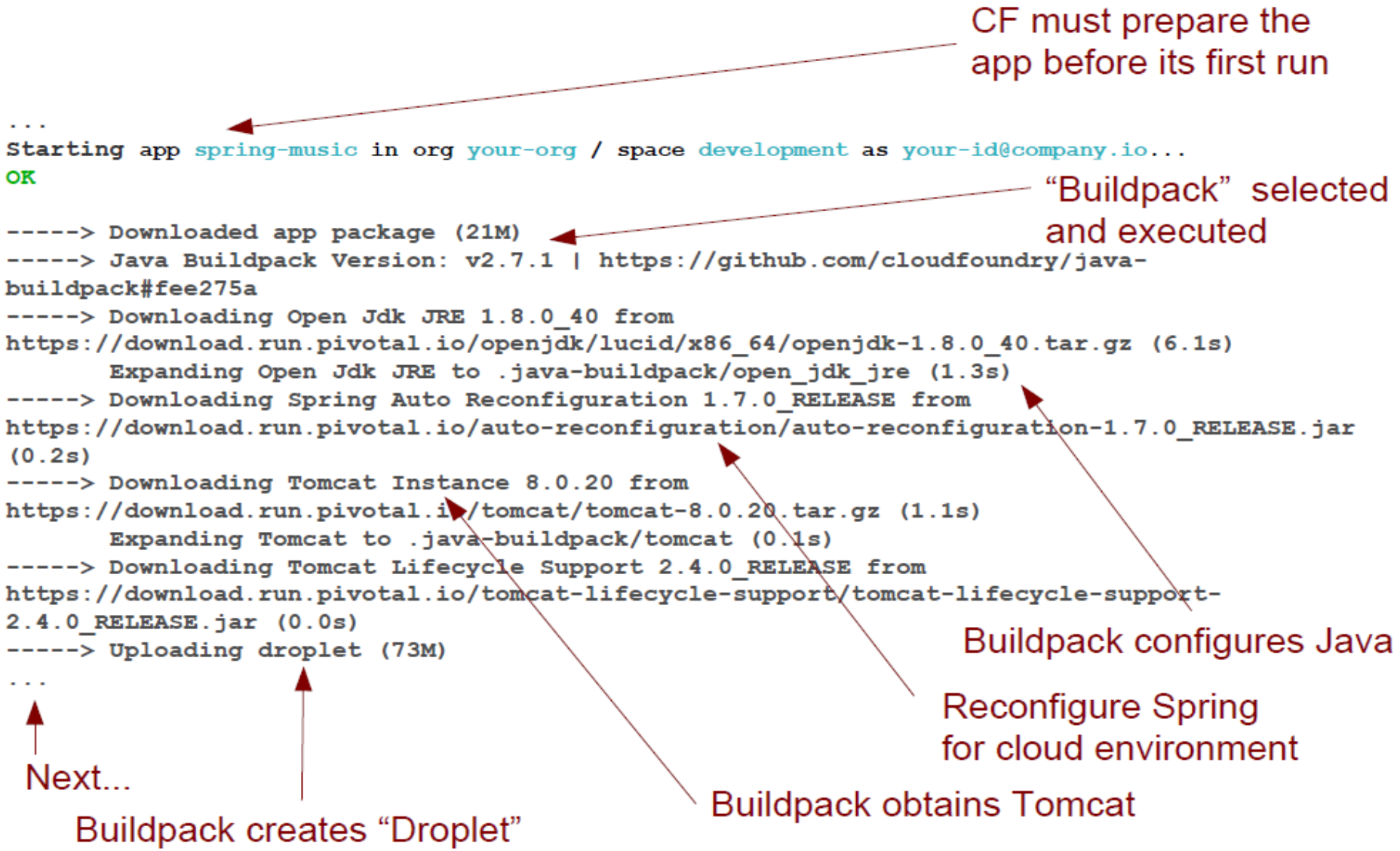
Updates CF metadata (app name, instances, memory)

Establish route

Uploads war

Next...

What Happens – "Staging"



What Happens – start

Cloud Foundry runs the "Droplet" on a "container"

```
...
0 of 1 instances running, 1 starting
0 of 1 instances running, 1 starting
1 of 1 instances running

App started

OK

App spring-music was started using this command `JAVA_HOME=$PWD/.java-buildpack/open_jdk_jre JAVA_OPTS="-
Djava.io.tmpdir=$TMPDIR -XX:OnOutOfMemoryError=$PWD/.java-buildpack/open_jdk_jre/bin/killjava.sh
-Xmx382293K -Xms382293K -XX:MaxMetaspaceSize=64M -XX:MetaspaceSize=64M -Xss995K
-Daccess.logging.enabled=false -Dhttp.port=$PORT" $PWD/.java-buildpack/tomcat/bin/catalina.sh run`

Showing health and status for app spring-music in org your-org as your-id@company.io...
OK

requested state: started
instances: 1/1
usage: 512M x 1 instances
urls: spring-music-678.cfapps.io
last uploaded: Tue Mar 17 17:58:35 UTC 2015

     state     since                    cpu    memory           disk
#0   running   2015-03-17 01:59:35 PM   0.0%   474.4M of 512M   150.3M of 1G
```

Health Check

Done! 1 application instance running on spring-music-678.cfapps.io

Application State and Logs

- Run `cf apps`

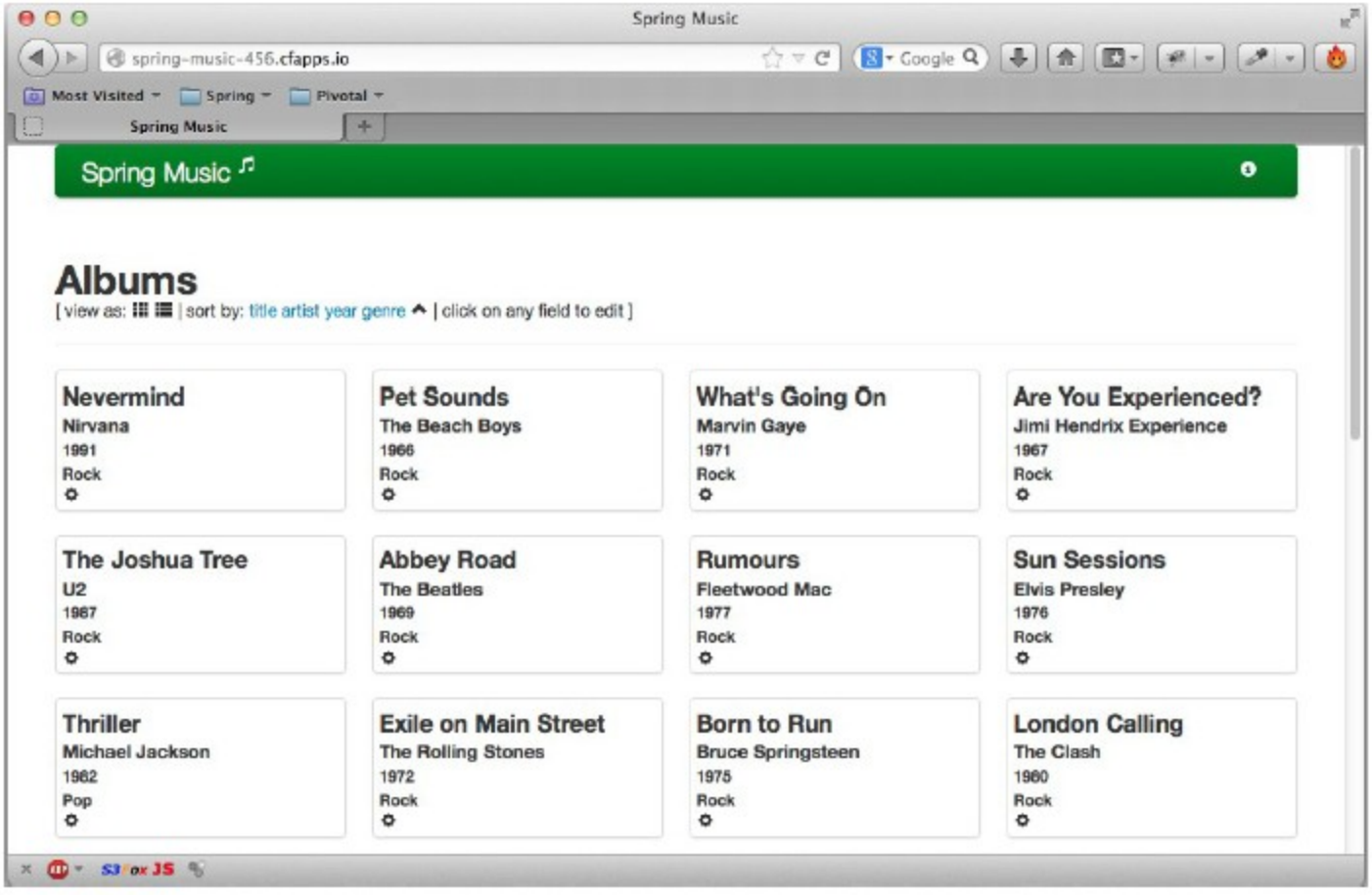
```
> cf apps
Getting apps in org pivotaledu / space development as kkrueger@pivotal.io...
OK

name           requested state   instances   memory   disk   urls
spring-music    started          1/1         512M     1G     spring-music-678.cfapps.io
```

- `cf logs spring-music`

```
> cf logs spring-music
Connected, tailing logs for app spring-music in org pivotaledu / space development as
kkrueger@gopivotal.com...
2014-06-07T23:01:47.68-0400 [RTR]     OUT spring-music-678.cfapps.io -
[08/06/2014:03:01:47 +0000] "GET /assets/js/status.js HTTP/1.1" 200 844 "http://spring-
music-678.cfapps.io/" "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_7_5)
AppleWebKit/537.73.11 (KHTML, like Gecko) Version/6.1.1 Safari/537.73.11"
10.10.66.34:64401 vcap_request_id:73037523-63ef-498f-6cd8-d3b48fe69e84
response_time:0.003693009 app_id:314f0434-d2c9-446c-ab4a-6c310878ca80
2014-06-07T23:01:48.47-0400 [RTR]     OUT spring-music-678.cfapps.io -
[08/06/2014:03:01:48 +0000] "GET /assets/templates/header.html HTTP/1.1" 200 1060
"http://spring-music-678.cfapps.io/" "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_7_5)
AppleWebKit/537.73.11 (KHTML, like Gecko) Version/6.1.1 Safari/537.73.11"
10.10.66.34:64324 vcap_request_id:39fbb3f2-46fb-4bd7-78d6-8994fafade9f
response_time:0.004132254 app_id:314f0434-d2c9-446c-ab4a-6c310878ca80
```


See the Application Running

- Open a browser window to **spring-music-678.cfapps.io**



Configuring a Deployed Application

- Change the number of instances
  - **cf scale <app> -i <new-value>**
  - Two instances: **cf scale spring-music –i 2**
  - New instances added, or some existing instances stopped
- Change the memory allocation
  - **cf scale <app> -m <new-value>**
  - 1024M: **cf scale spring-music –m 1024M**
  - Requires a restart to take effect

Stopping and Starting

- cf stop - Sends SIGTERM message to application
  - Sends SIGKILL 10 seconds later if still running
- cf start - Starts existing application
- cf restart - cf stop followed by cf start
- cf restage - Repeats the staging process, and starts the app.
  - Useful when environment variables / bound services change

    * (Covered later)

Adding / Removing Routes

- Add a new domain mapping
  - **cf map-route <app> <domain> -n <hostname>**
  - **cf map-route spring-music cfapps.io –n mymusic**
    - **mymusic.cfapps.io** **also** maps to spring-music
- Remove mapping
  - **cf unmap-route <app> <domain> -n <hostname>**
  - **cf unmap-route spring-music cfapps.io –n spring-music-678**
    - **Spring-music-678.cfapps.io** **no longer** maps to spring-music

Cleaning Up Unused Routes

- Routes tend to accumulate over time
  - Application in other Orgs / Spaces cannot use these routes
- Find all other routes used in a space:
  - **cf routes**
- Remove route:
  - **cf delete-route**
- Very Useful! Remove unused routes:
  - **cf delete-orphaned-routes**

Managing Application Instances

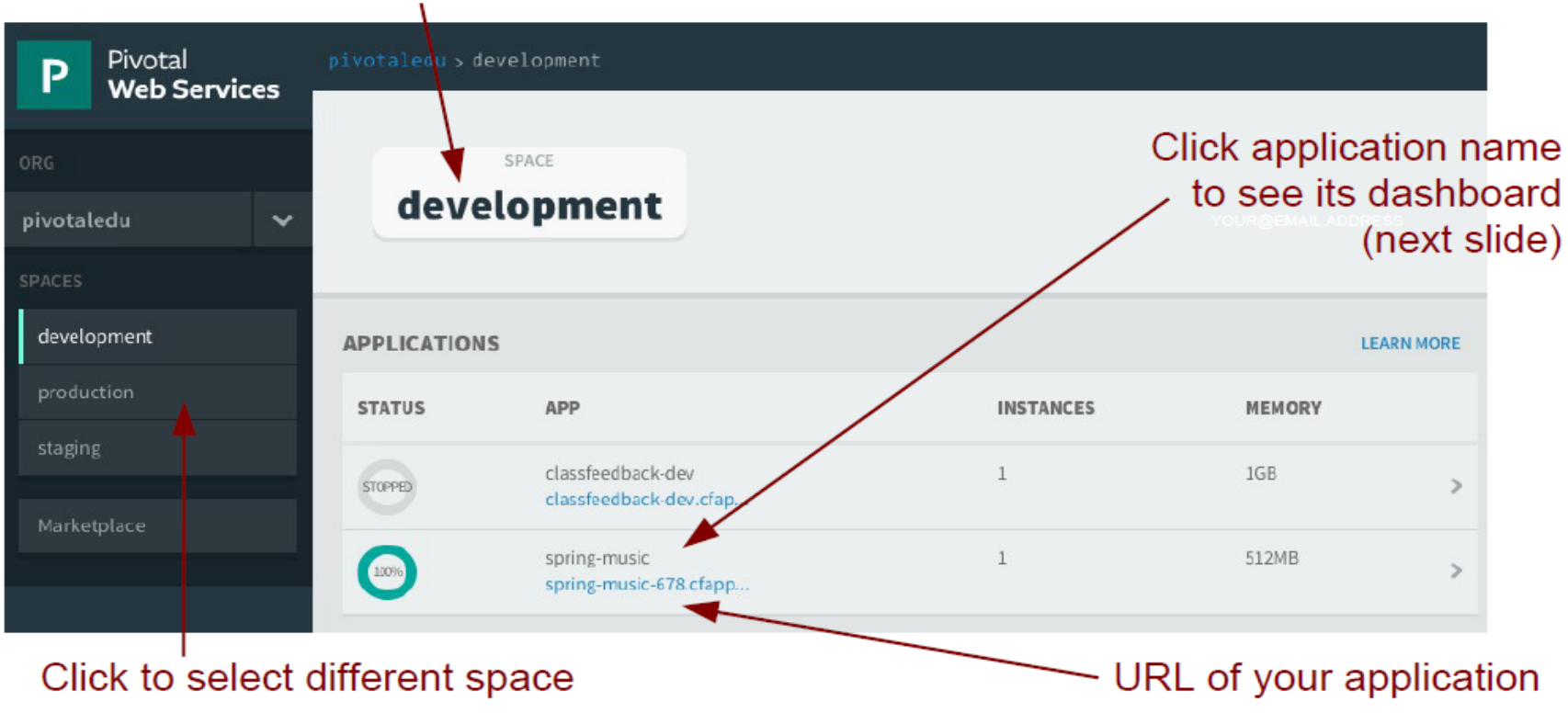
Apps Manager

- Login to Cloud Foundry using your web-browser
  - Pivotal Web Services : http://run.pivotal.io
    - Your Cloud Foundry instance URL will be different
    - console.<your-cf-domain>
  - Use the username and password you registered with
  - Our new application should show green in the Apps Manager
- Next slide

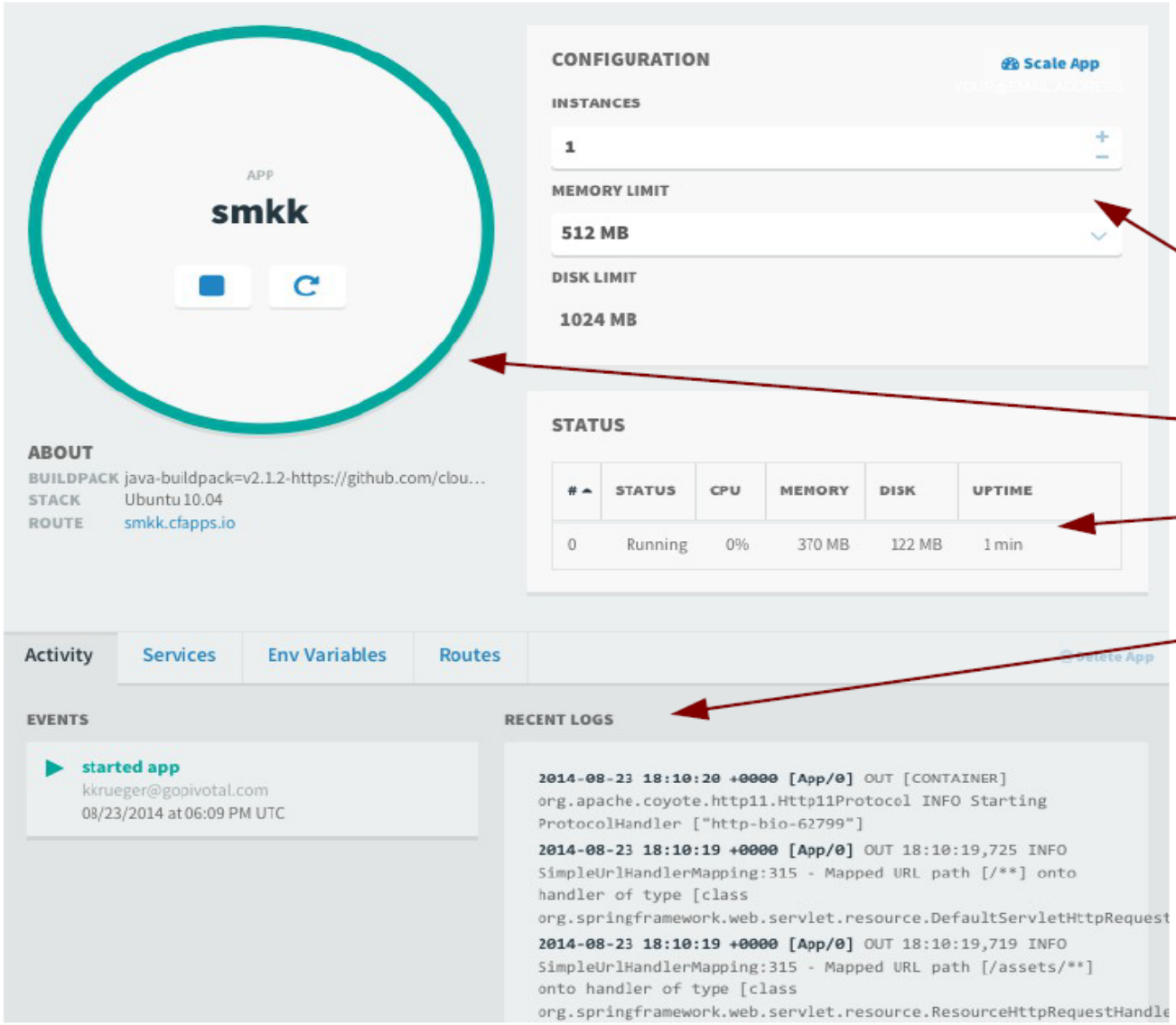
Note: *only pivotal CF comes with the apps manager*
Open Source Cloud Foundry does not

Apps Manager Home Page

- At a glance view of all your applications
  - Shows current space



Application Dashboard



Change instances, memory

Application state

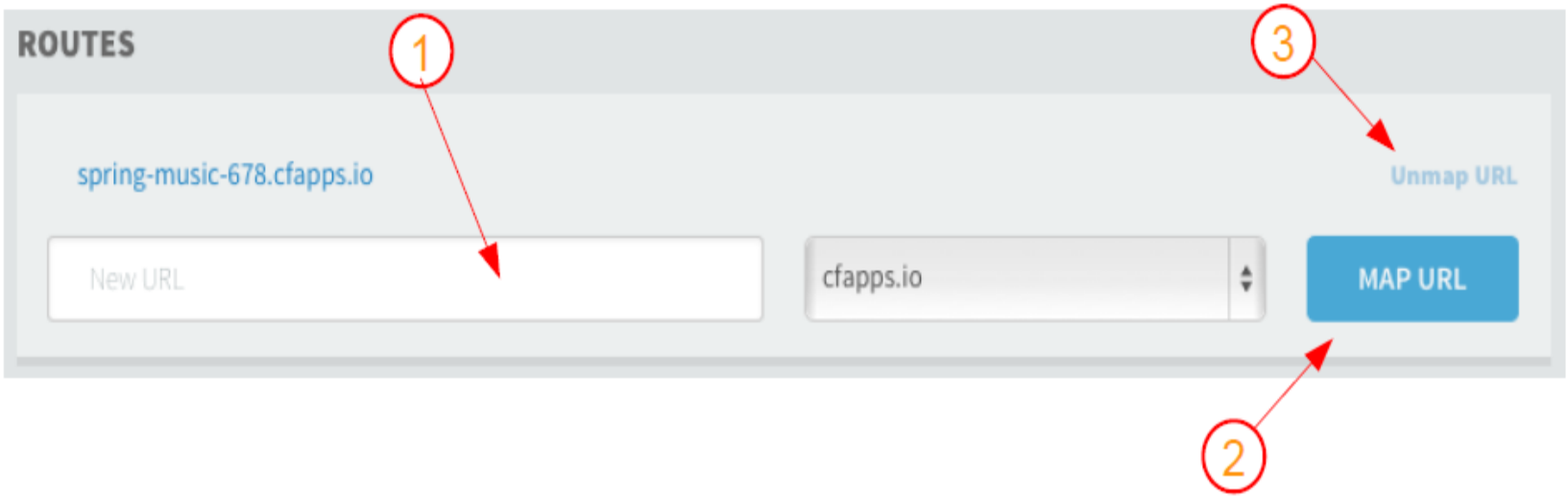
Instance Statistics

Logs and Events

All work described was performed by Capgemini or a Capgemini affiliate

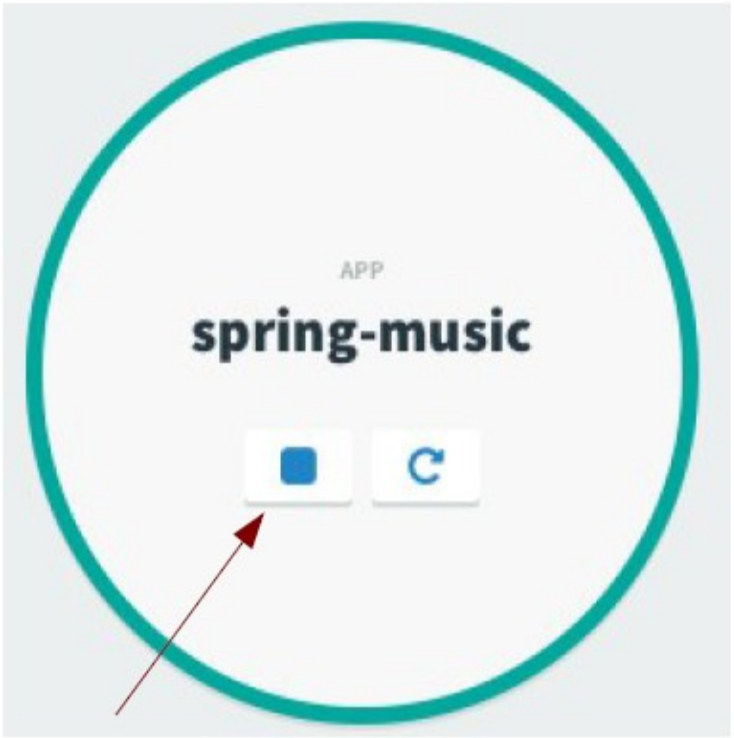
Change Mapped URL

- Enter new domain (1)
  - Remember to make it Unique
- Click MAP URL (2)
- To remove a mapping (3)
  - Click UNMAP URL at far right of same line



Stopping and Starting

- Just click the square to stop
- Click play to start



Click to Stop

Click to Start

Monitoring Instances

- The very bottom panel shows all your instances
  - Provides statistics
  - Updated live (slight time – lag)

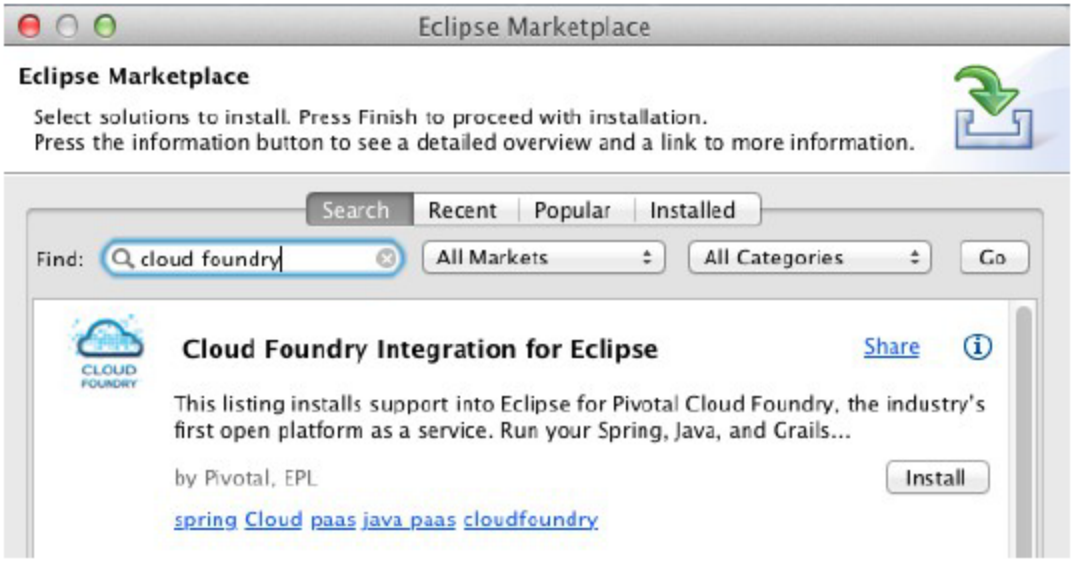
INSTANCES

INSTANCE	CPU	MEMORY	DISK	UPTIME	STATE
1	0%	312MB	121MB		Running
0	0%	314MB	121MB	31min	Running

Deploying via Eclipse or Spring Tool Suite

Setting up Eclipse / Spring Tool Suite

- Select Help → Eclipse Marketplace
- Search for Cloud Foundry plugin and install

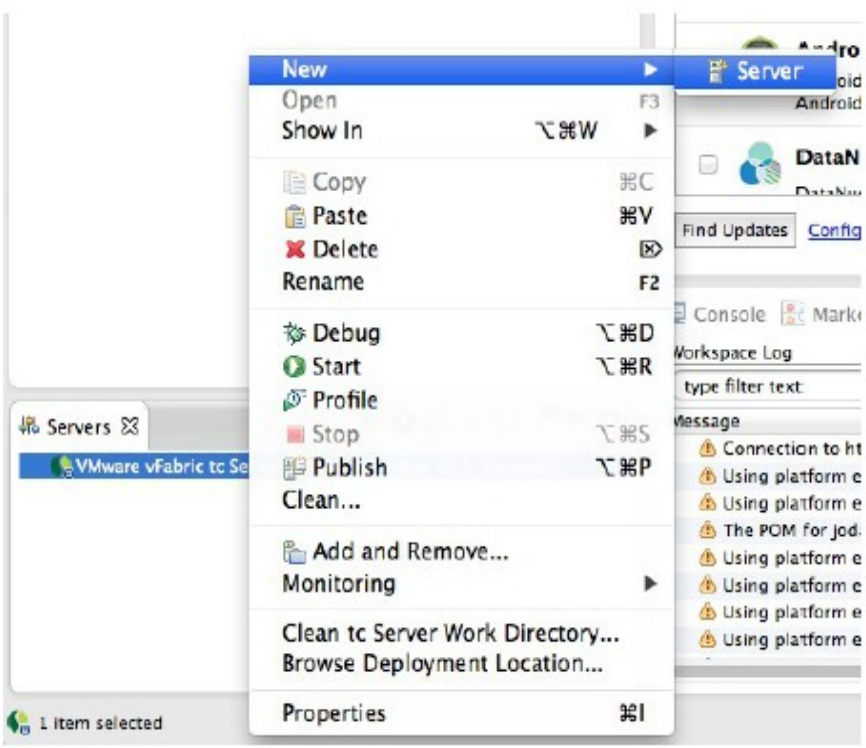


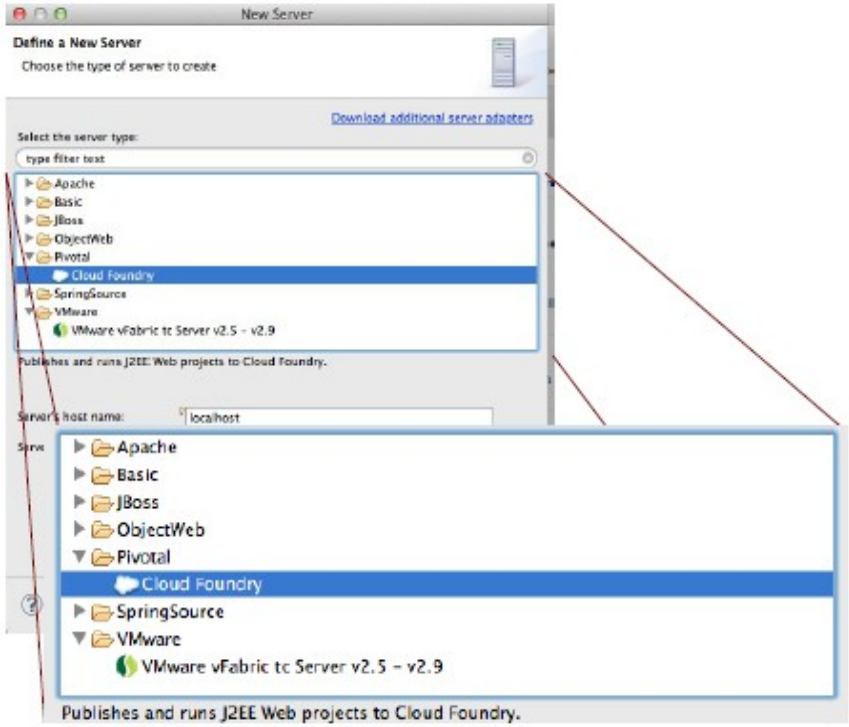
Do this now…

- The install takes time to run.

Setting up a New Server

- In white-area of Servers panel, right click New → Server
- In popup, under Pivotal select Cloud Foundry



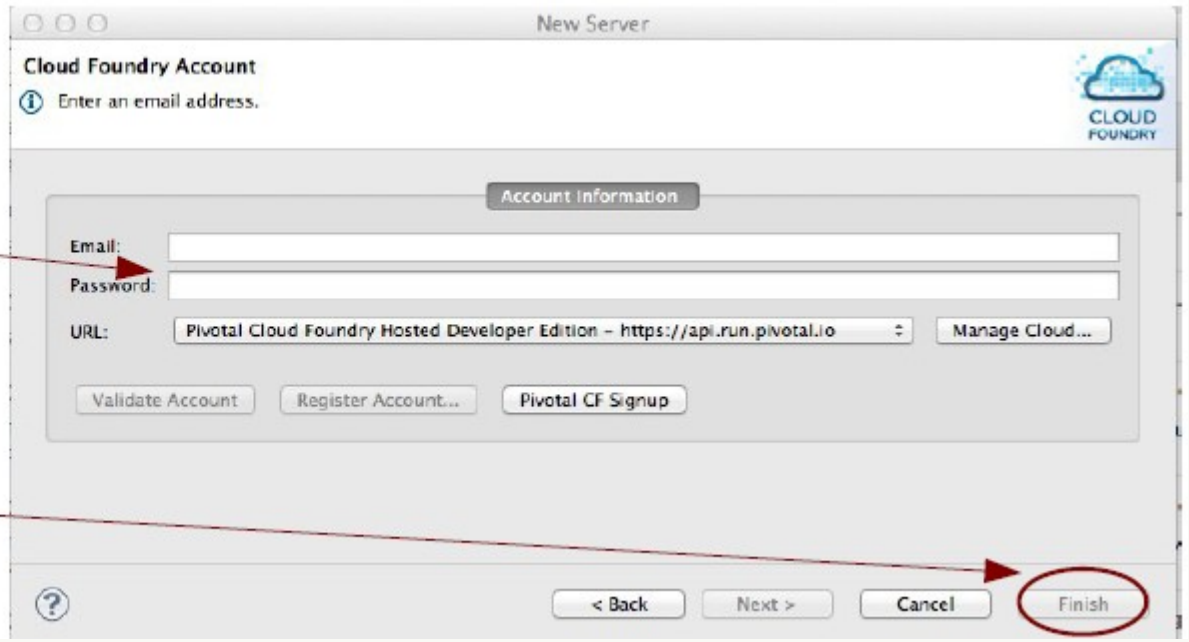


Fill in Details of your CF Account

- Fill in registration dialog
  - Use Manage Cloud to specify a different URL
    - Another public PaaS or for your private cloud
  - Note there is a Signup button here.

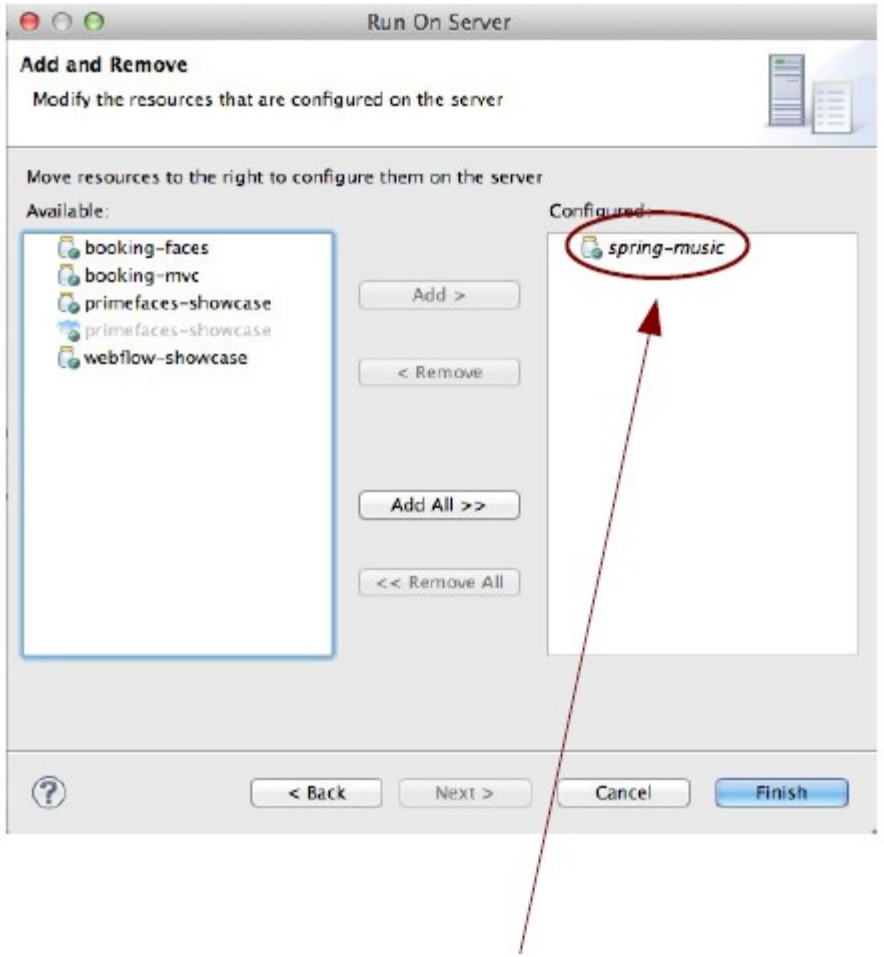
Same email and password
That you signed-up with

Click Finish when done



Deployment

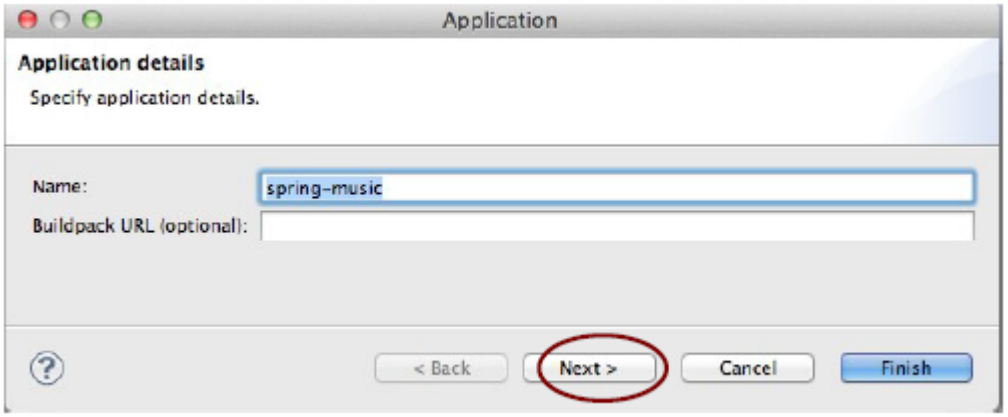
- We are now ready to deploy an application
- Select a project in Eclipse
  - Right click and select
    - Run As … → Run on server
    - *Just like any other server*
  - Select Cloud Foundry server
  - Click Next
  - In next dialog make sure your project is in the RHS list
    - Just as you would normally
  - Click Finish to deploy

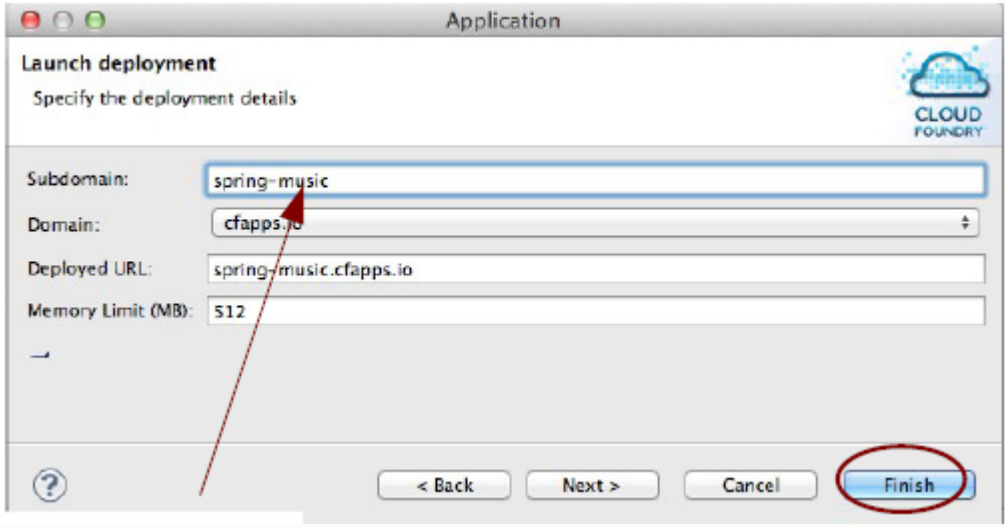


Now things get different …

Application details

- Two dialogs appear
  - Application details … just click next for now
  - Launch deployment… Pick a unique URL
    - All apps deploy to <appname>.cfapps.io
  - For now, just click Finish to deploy





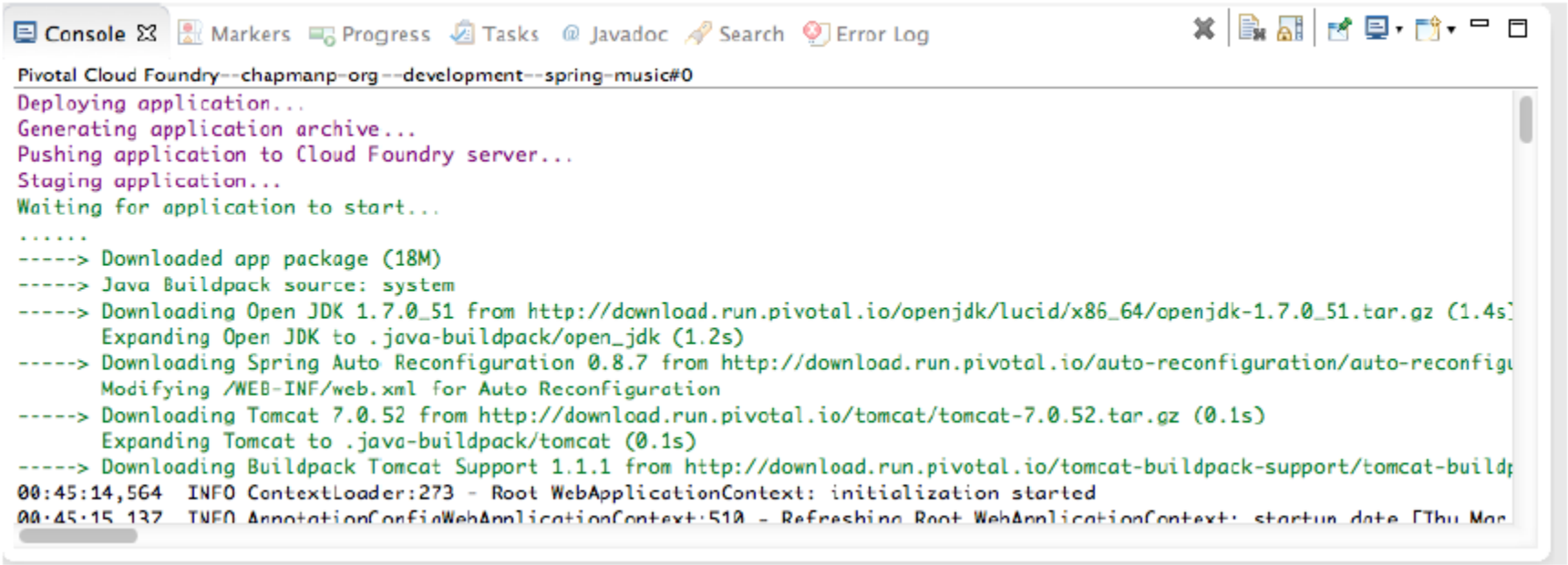
Will this be unique? Change Sub-domain to make sure

What just happened?

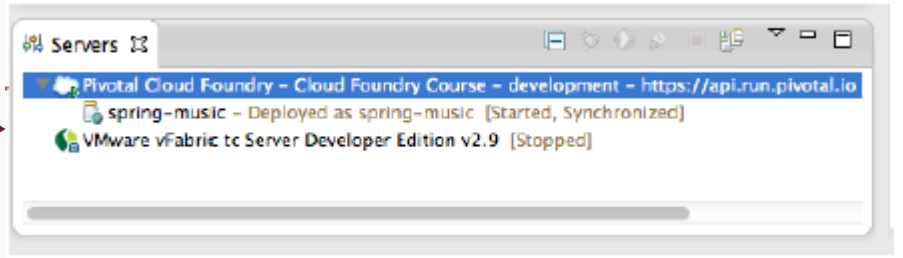
- Eclipse Connected to Cloud Foundry using your credentials
- It 'pushed' your application to CF and told it to deploy it
  - The whole application is uploaded – takes a while
  - CF "staged" your application
    - Recognized Java/ WAR, prepared a "droplet" containing JRE and Tomcat server
  - "Droplet" was deployed to a container and began running
  - All requests to the Deployed URL route to your application
- Same process as when using the CLI

Watching it Run

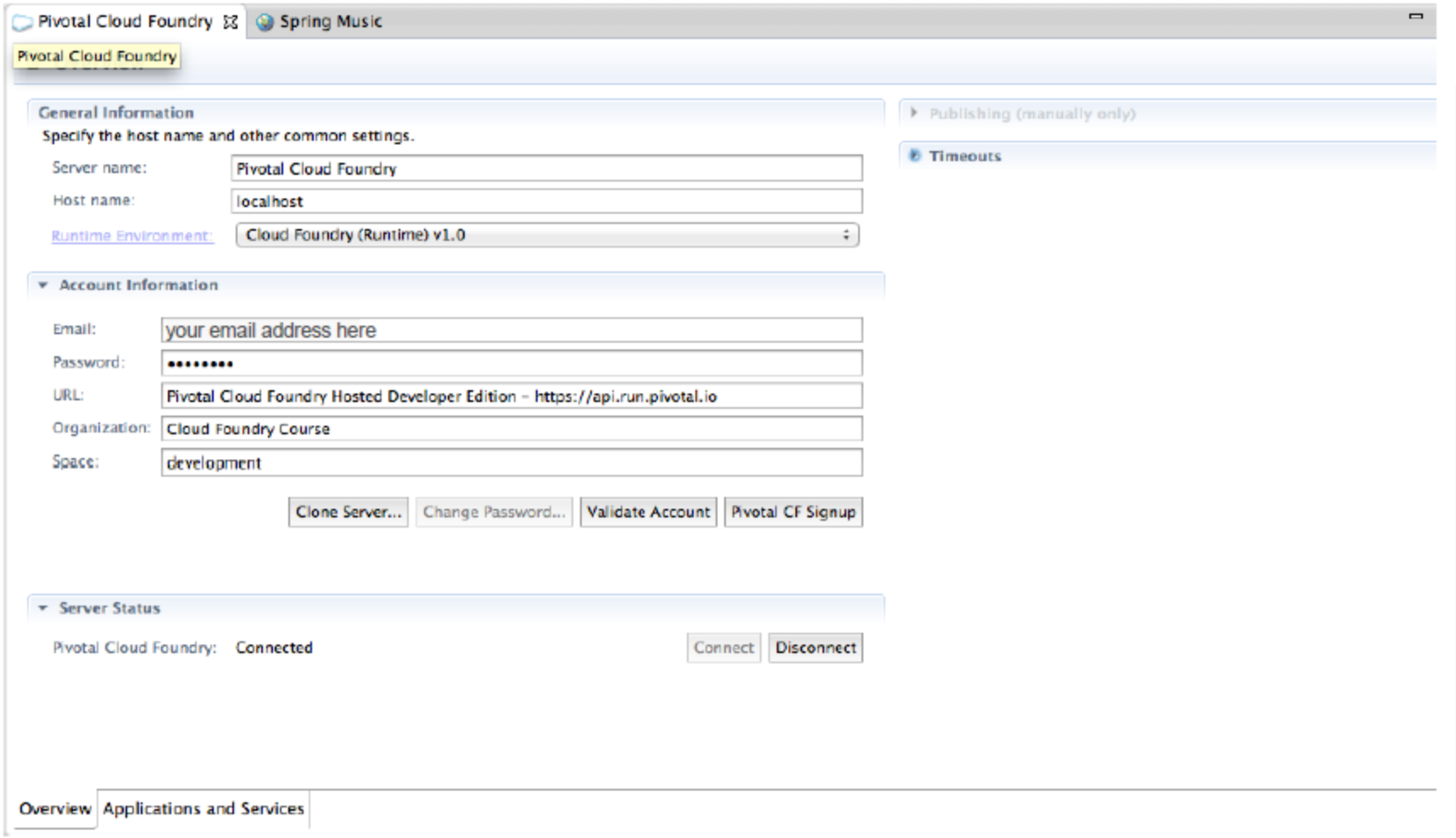
- In the Console View



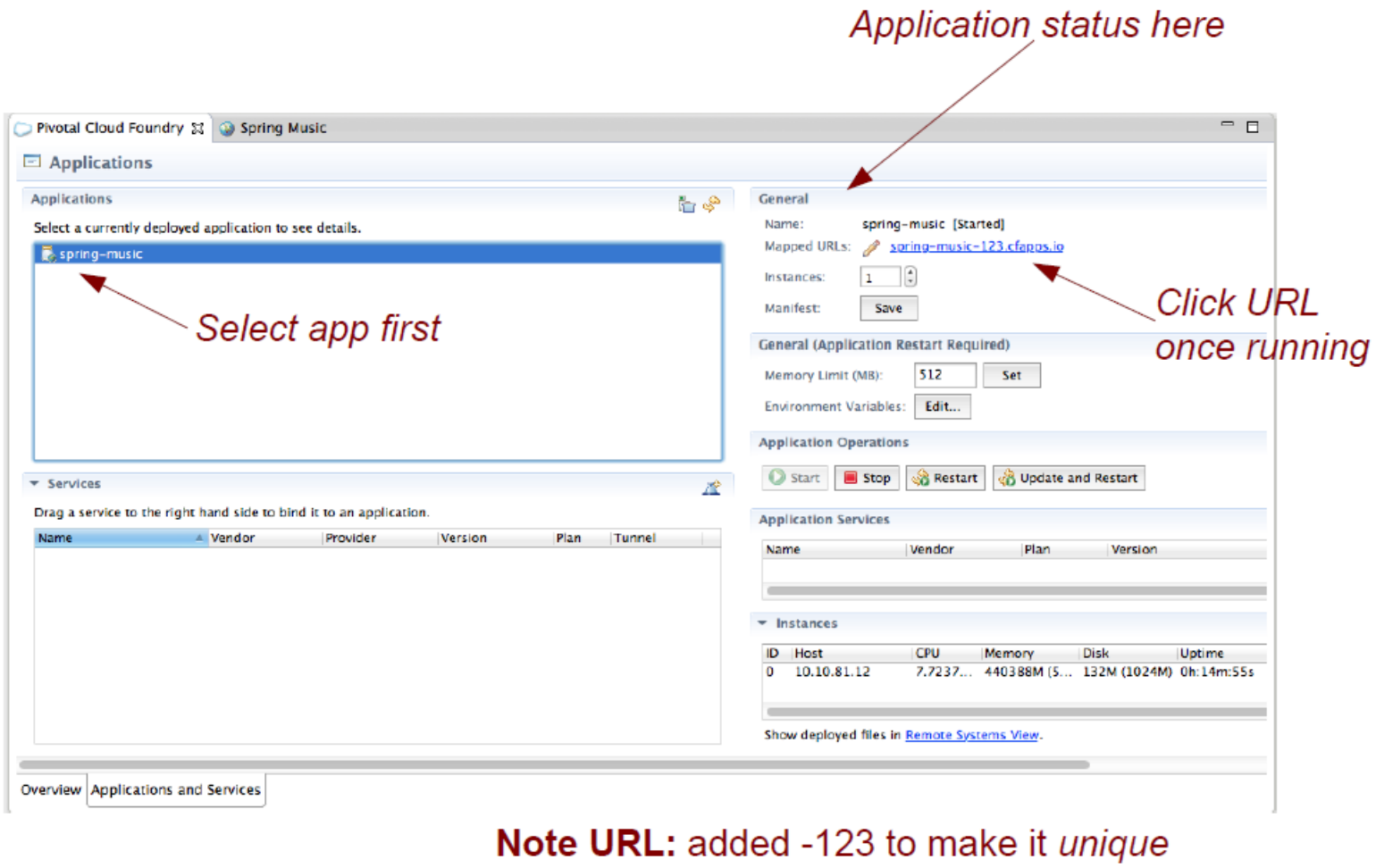
- In the Dashboard
  - Double click



Overview Dashboard Tab

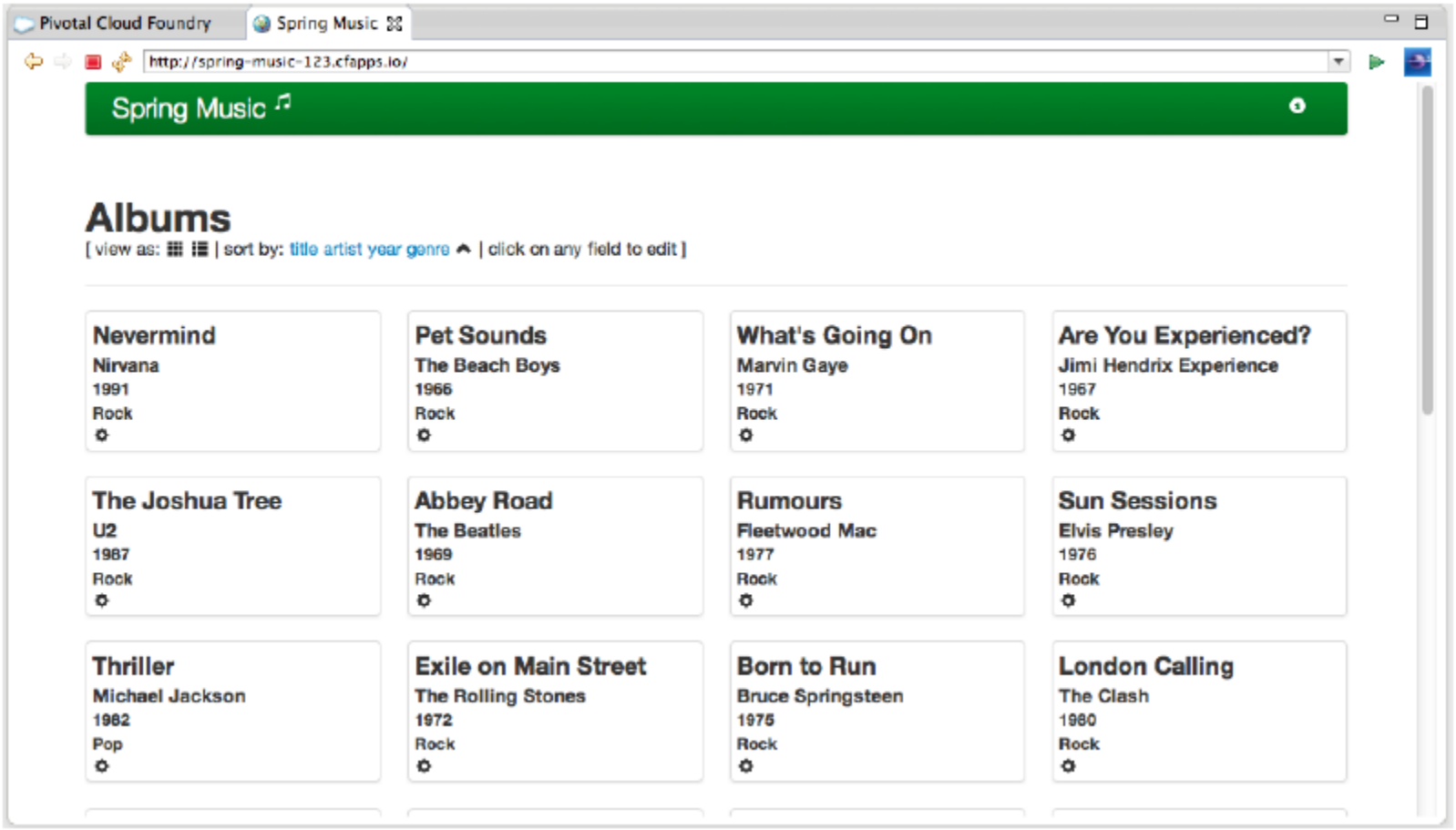


Application Tab



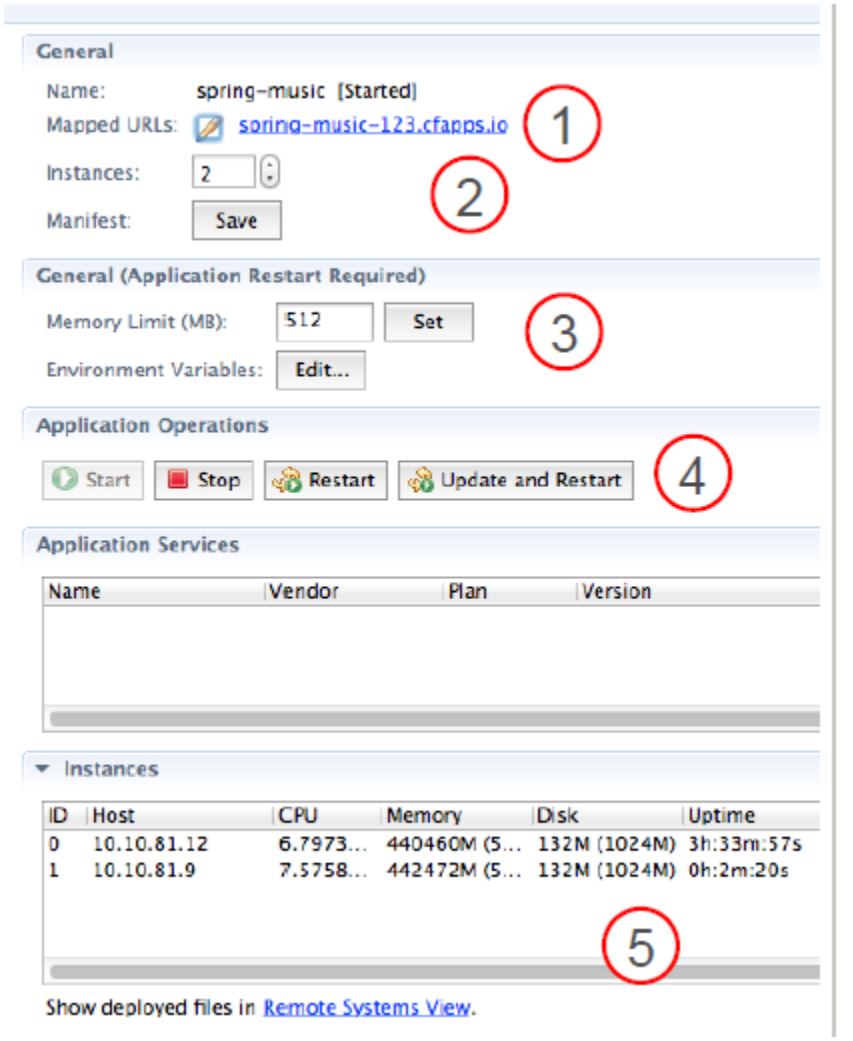
See your Application Running in Eclipse

- Eclipse pops up a browser window open at your URL
  - Or use the browser of your choice



Managing Application Instances

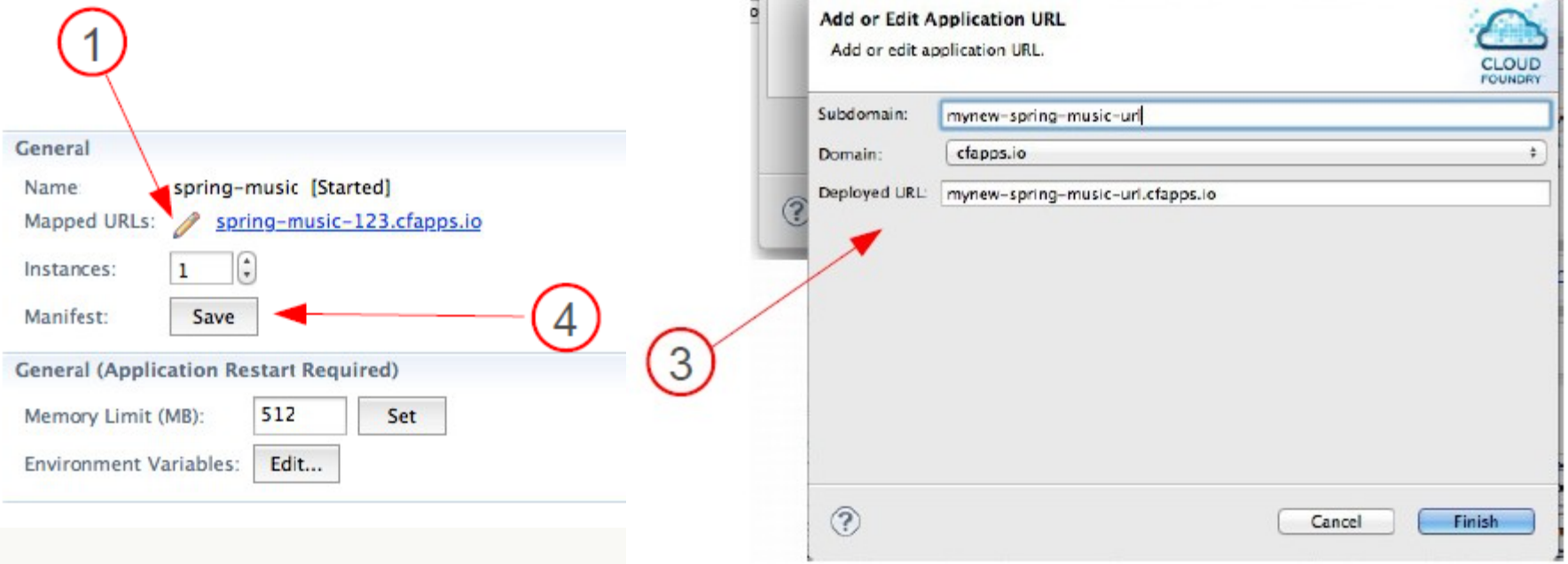
Cloud Foundry Dashboard



- The right-side panel of Applications and Services tab
  - Below General
- Control your application
  1. Change mapped URL
  2. Add/remove instances
  3. Change memory
  4. Stop/start
  5. Monitor instances

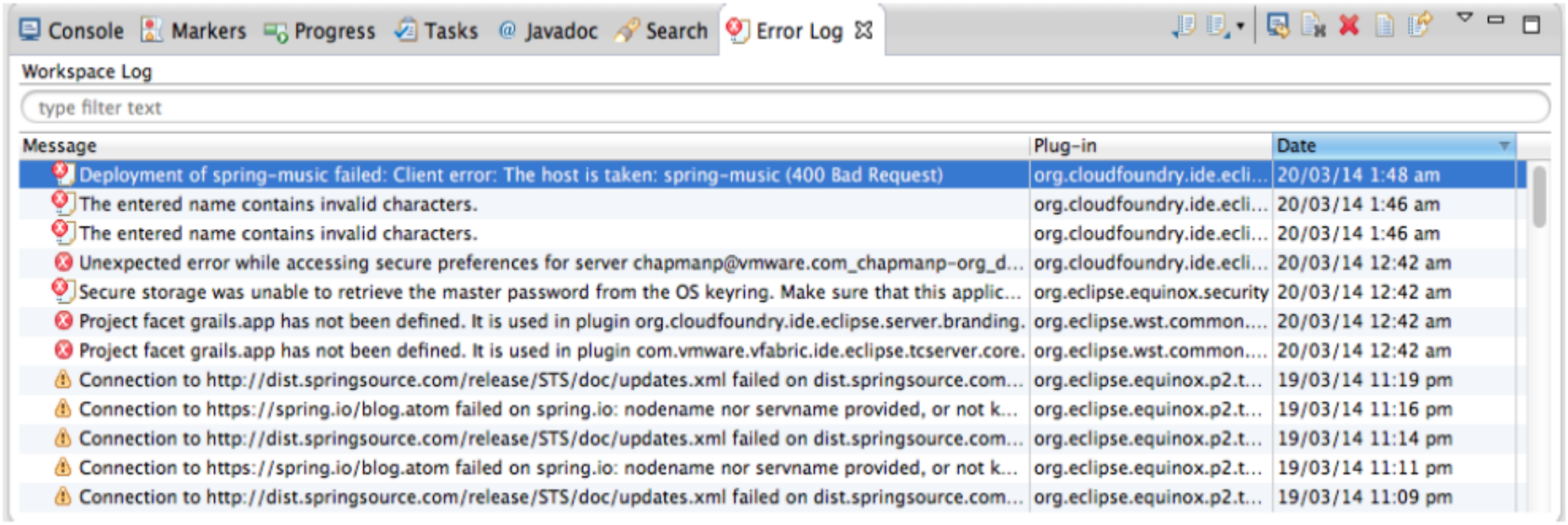
Change Mapped URL

- Click pencil(edit) icon
- Can add, edit or remove URLs
- Save when done



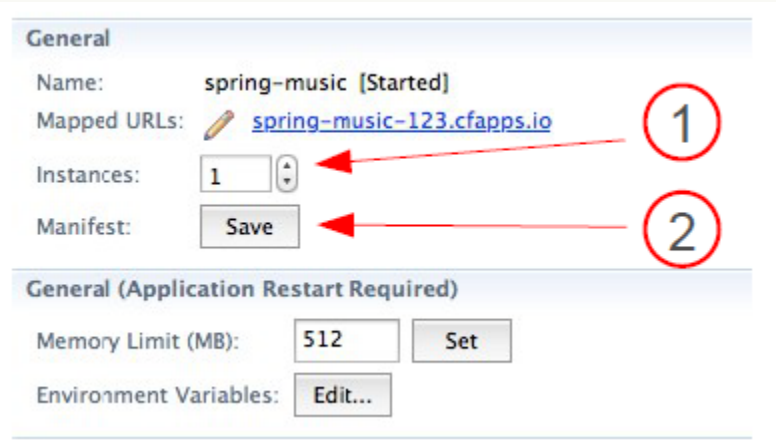
Choosing Your URL

- All applications mapped to cfapps.io domain
- Your URL must be unique
  - Get a Bad Request 400 if you try to use an existing URL



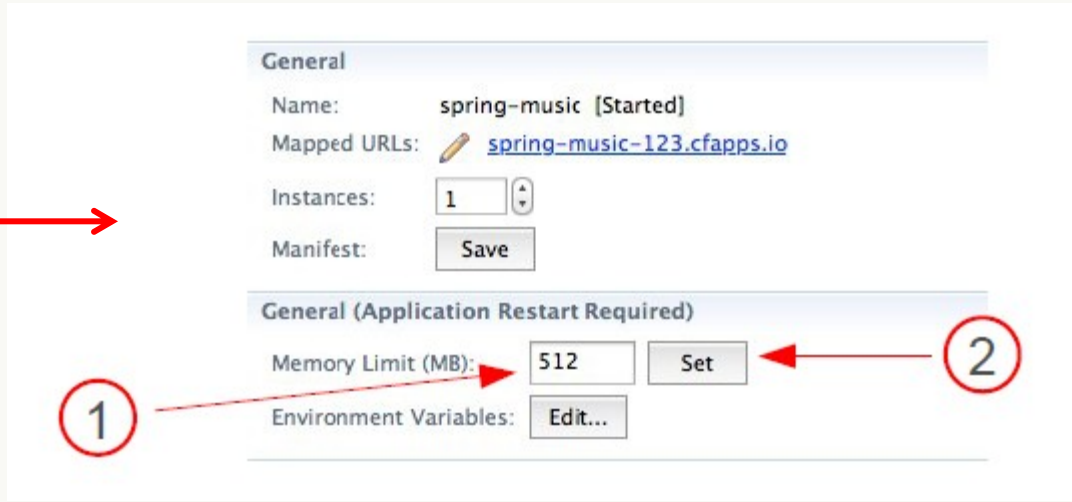
Instances

- By default one instance of your application runs up
  - Typically a Tomcat server
- To handle large loads you need multiple servers
  - Known as instances
  - Run behind load balancer
- How many instances do I need?
  - Design issue – covered later
- Modify as shown



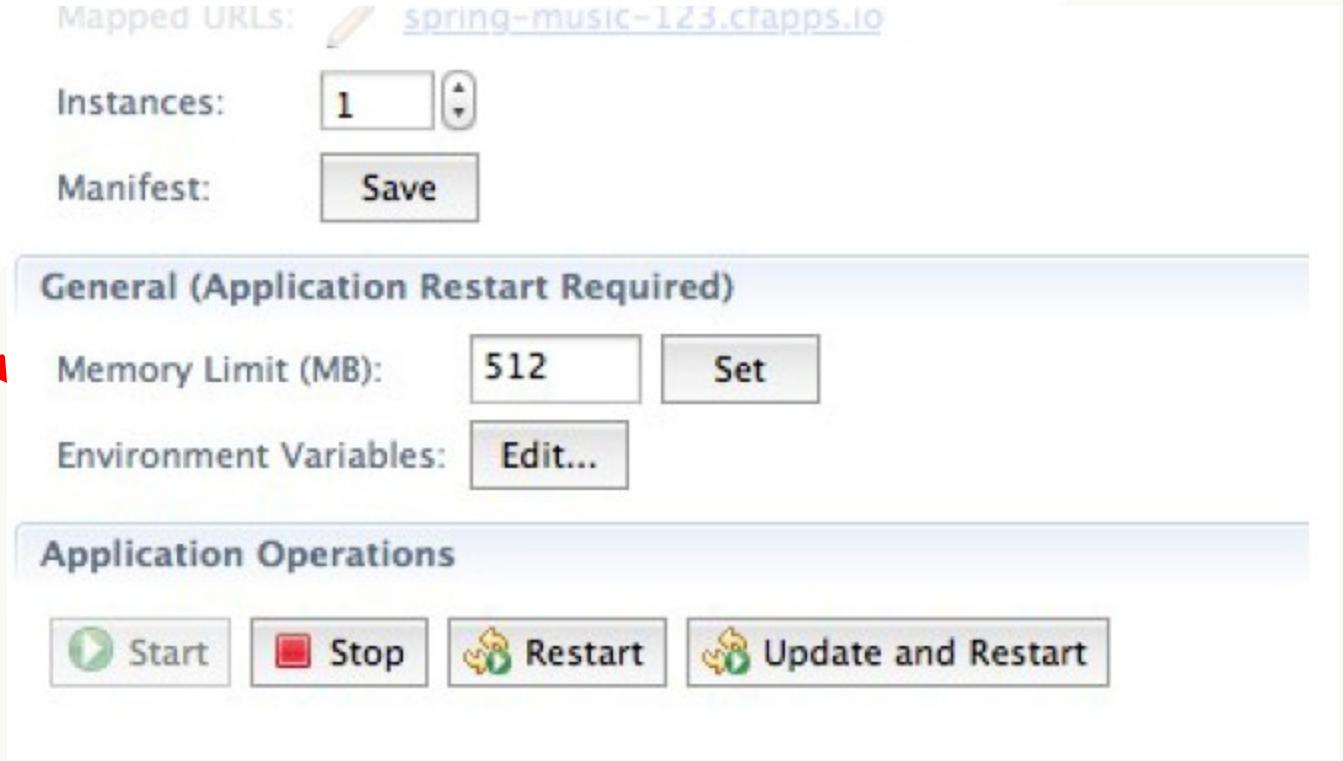
Memory Allocation

- Define how much memory our process gets to run in
  - 512 is the default
    - Good for a typical application under test (1 or 2 users)
  - How much memory do I need in production
    - Another good question for later!
- Easy to configure …
  - But the server has to be started



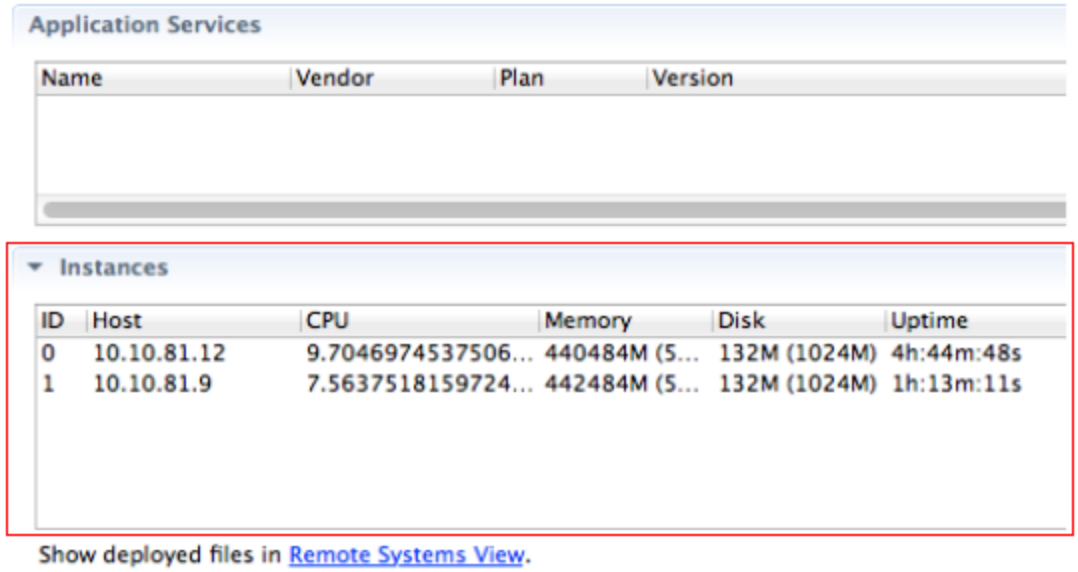
Stopping and Starting

- Normally this happens in the Servers view
  - Those buttons are grayed out
- Instead use the Dashboard



Monitoring Instances

- The very bottom panel shows all your instances
  - Provides statistics
  - Not real-time
- To refresh
  - Click refresh icon on the application list



Summary: Cloud Foundry Dashboard

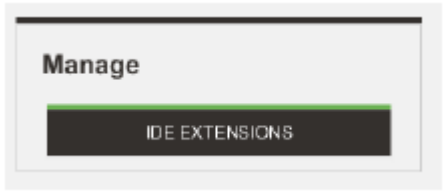
- In the Application and Services tab
  - Configure your application (below general on right-side)
  - Options
    - Modify mapped URL
    - Change number of instances
    - Change the amount of memory allocated
    - Start and stop the application
    - Monitor instances
  - You may have noticed we missed two options (later)
    - Add or remove services
    - Set environment variables

Appendices

- Describe installation of Cloud Foundry plug-in into
  - Appendix A : STS
  - Appendix B : Standard Eclipse
- For full details see:
  - http://docs.cloudfoundry.org/devguide/deploy-apps/sts.html

Appendix A: Installing CF Plug in into STS

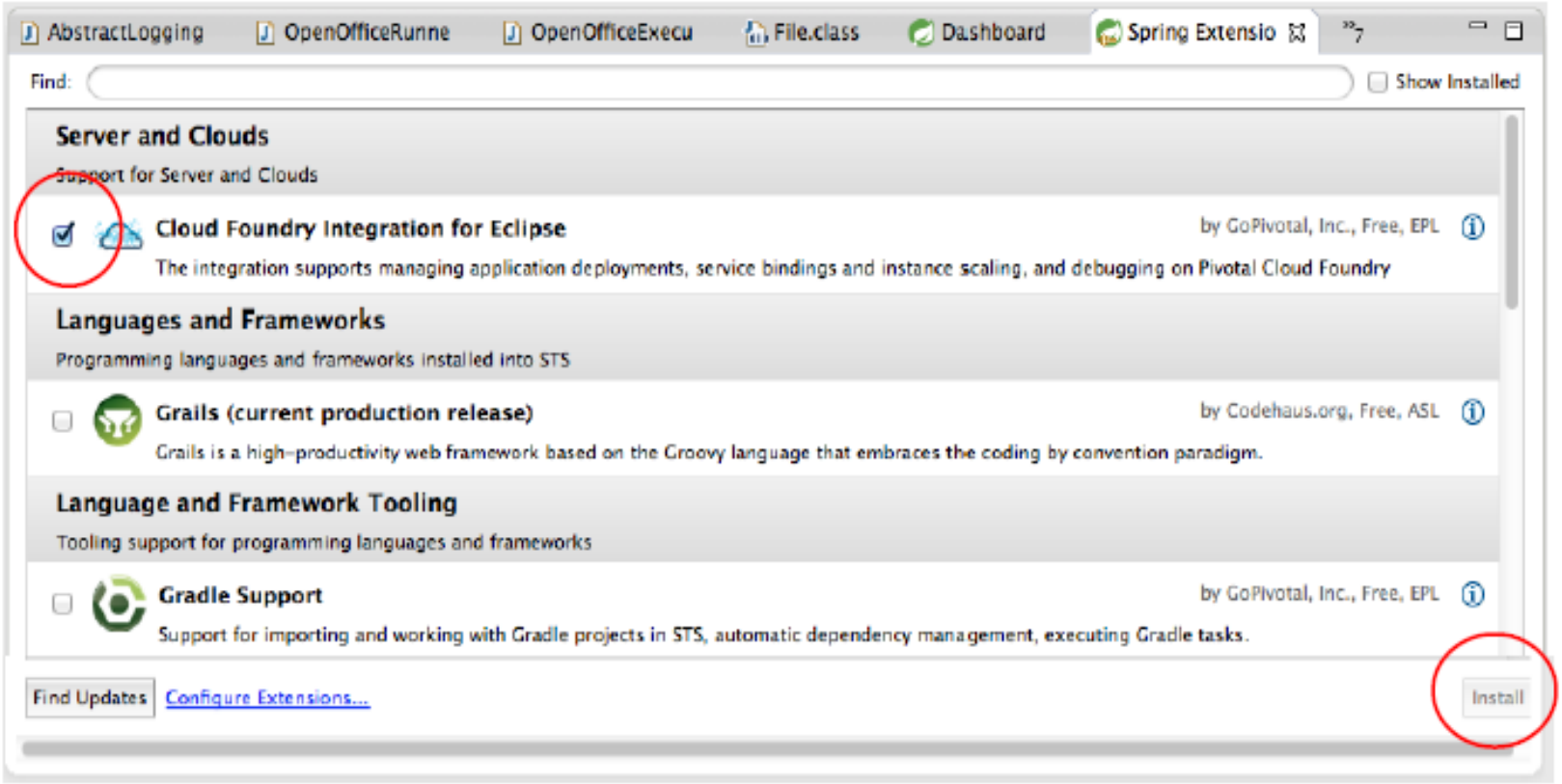
- Click Spring leaf icon in STS
  - Displays dashboard
- At bottom right under Manage
  - Click “IDE Extensions”



See : http://docs.cloudfoundry.org/devguide/deploy-apps/sts.html#install-to-sts

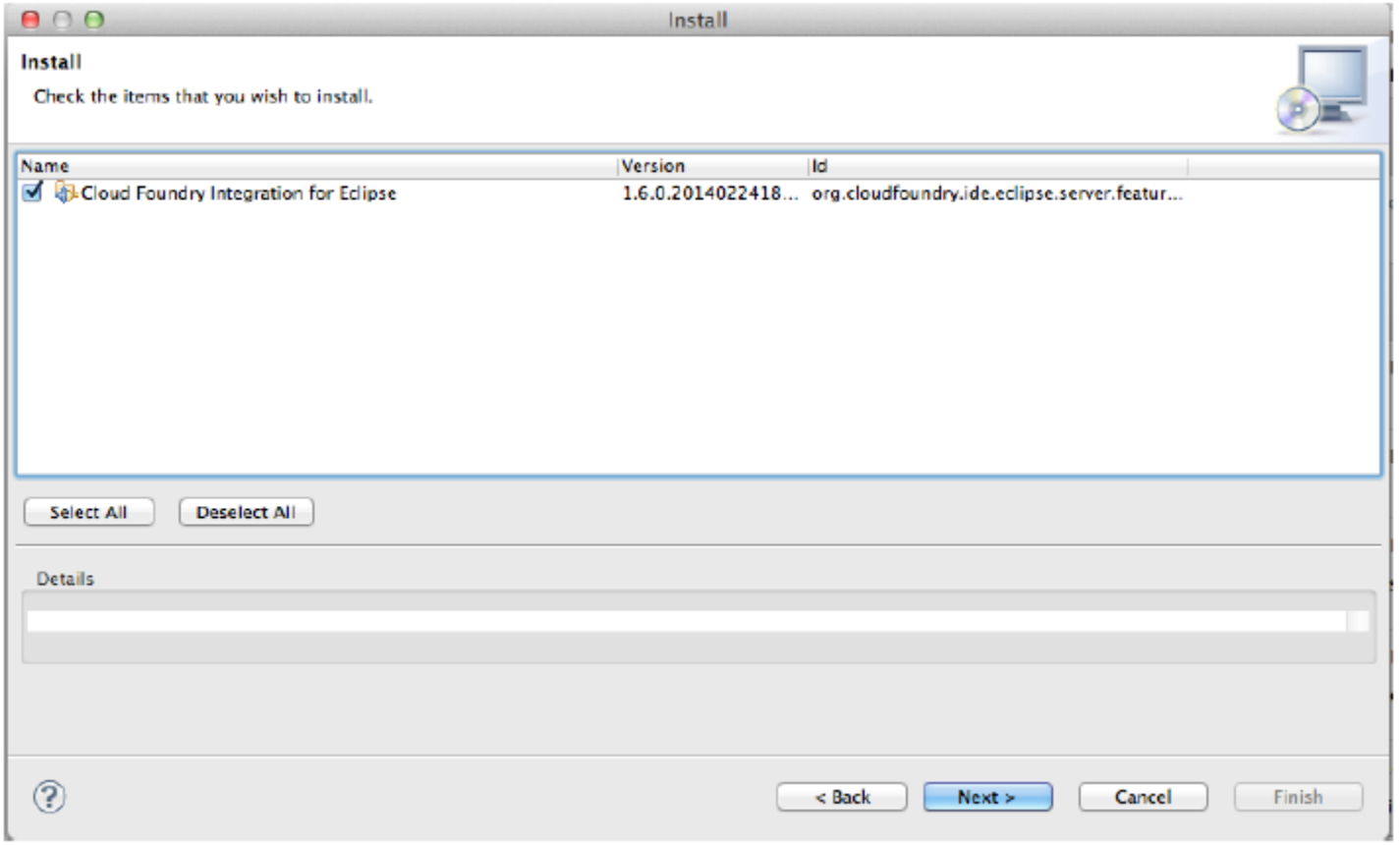
See Cloud Foundry Integration for Eclipse

- Select checkbox and click install button
  - If not listed, enter “cloud foundry” in Find and hit enter



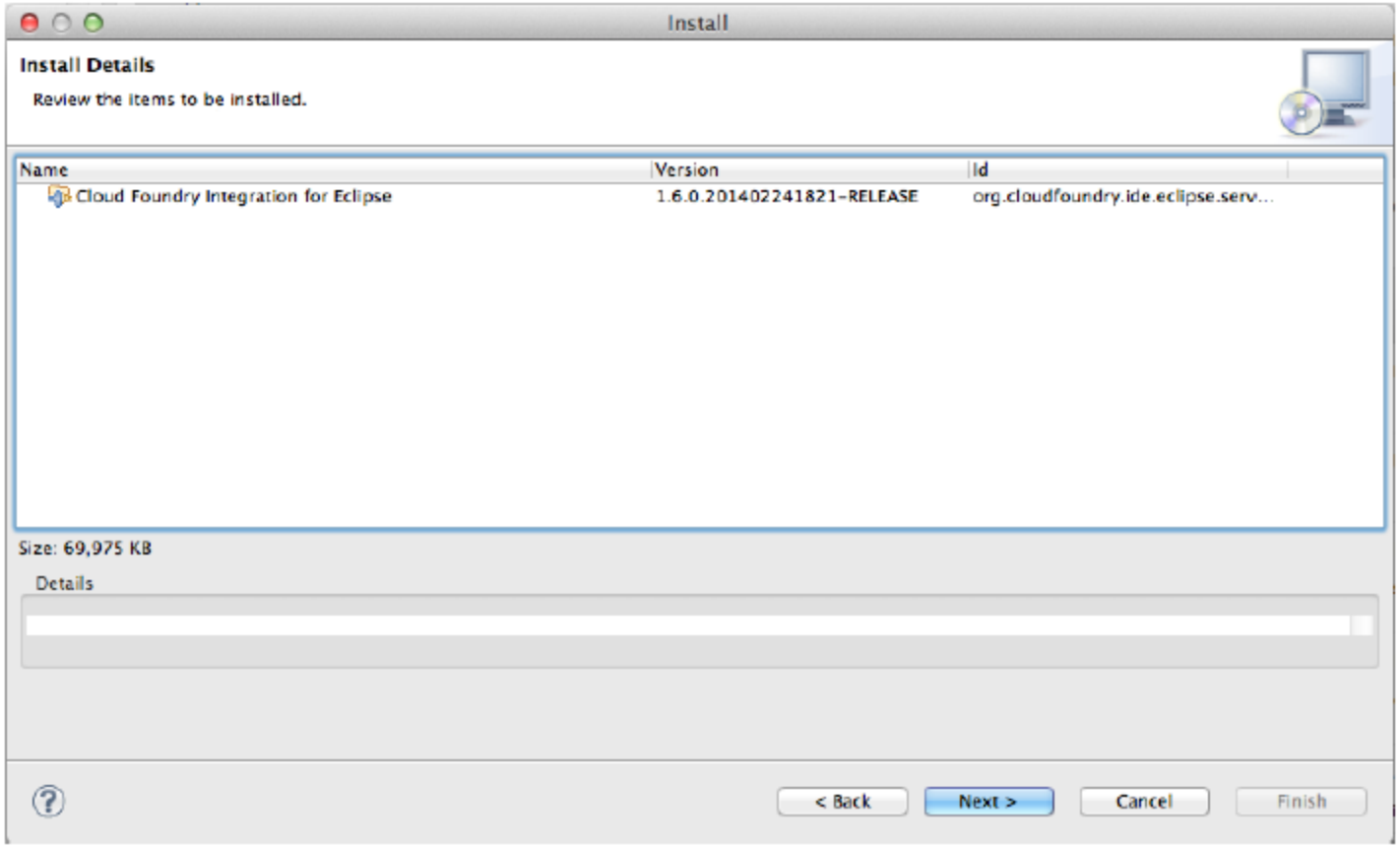
Runs up a wizard

- Click next



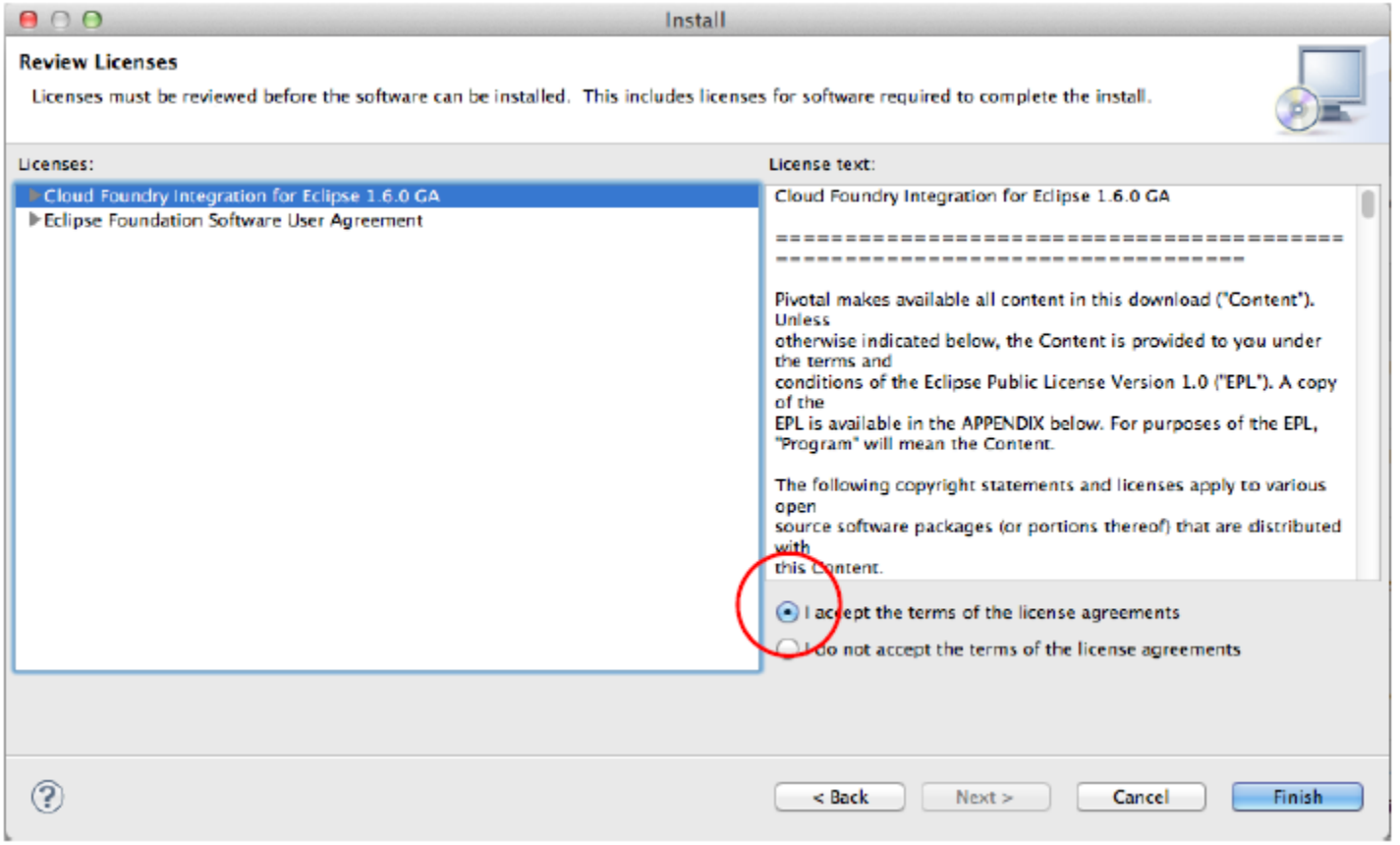
Wizard – Step 2

- Click next again



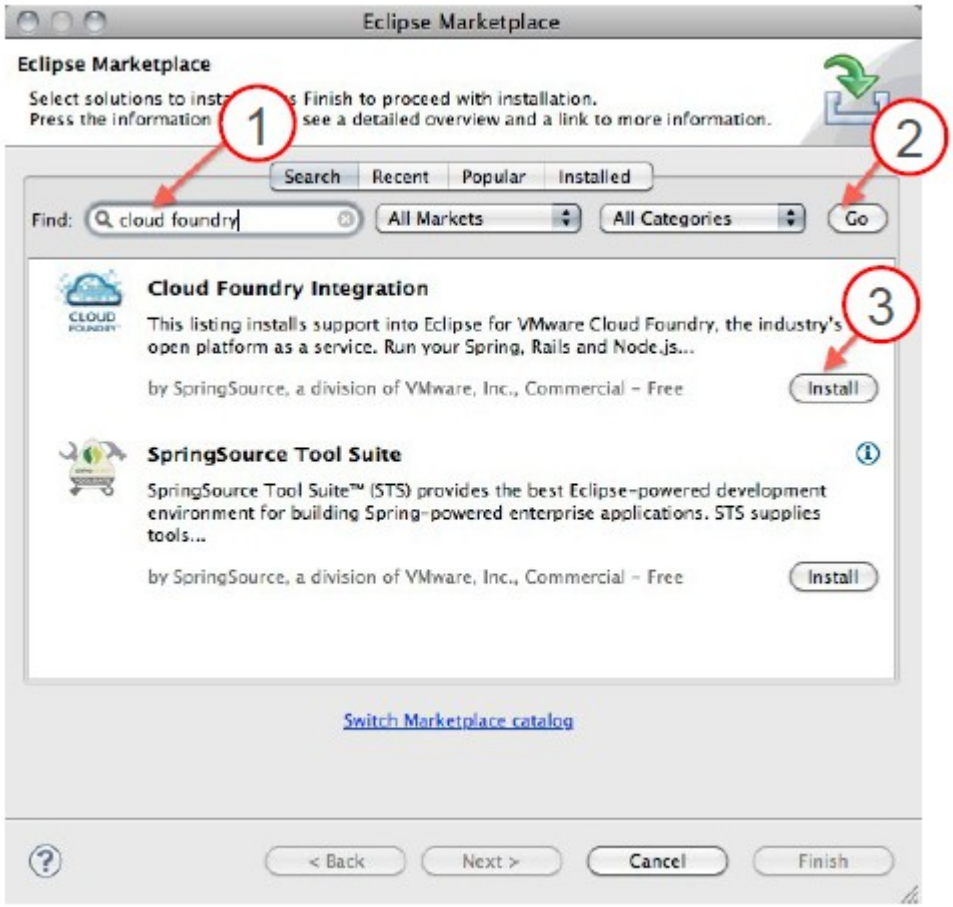
Wizard – Step 3

- Accept license agreement, and the installer will run



Appendix B : Installing into Eclipse

- For Help menu
  - Eclipse Marketplace
- Enter “Cloud Foundry” into find box
- Click “Go”
- Click “Cloud Foundry Integration” *Install* button

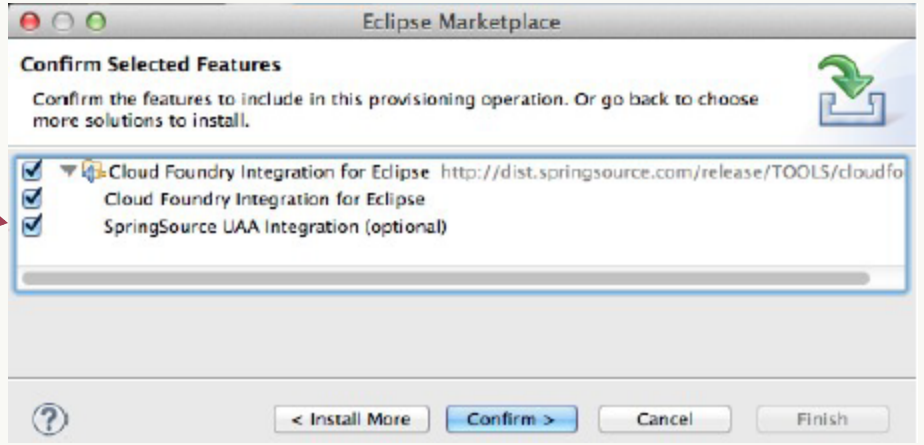


See : http://docs.cloudfoundry.org/devguide/deploy-apps/sts.html#install-to-eclipse

Confirm Selected Features

- Popup window lists what will be installed
  - "Cloud Foundry Integration for Eclipse"
  - "SpringSource UAA Integration" (optional)
    - Reports tool usage data, anonymously
    - Helps us track usage of free software
    - Deselect to stop plug in usage statistics being sent
  - Click Confirm.

Optional, deselect it you wish

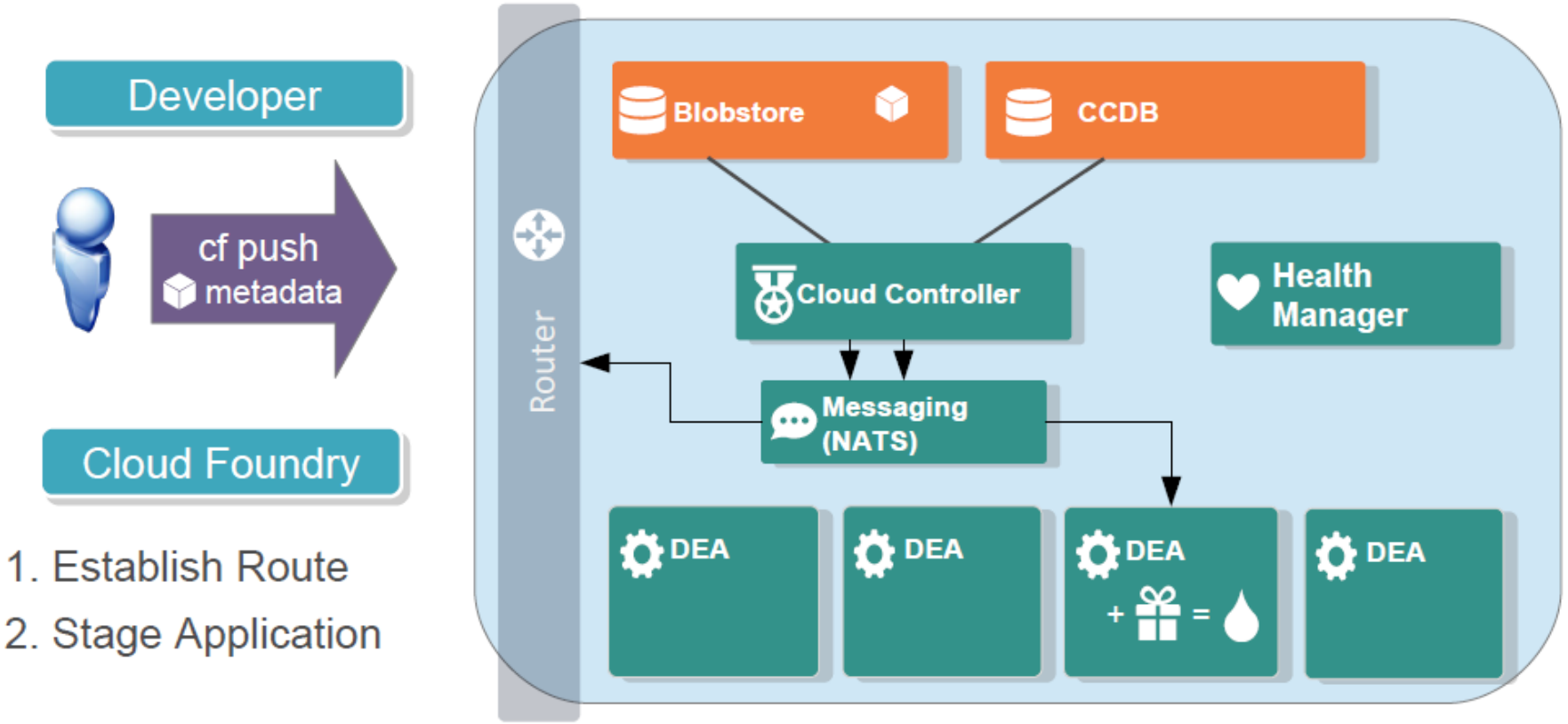


Last Few Steps

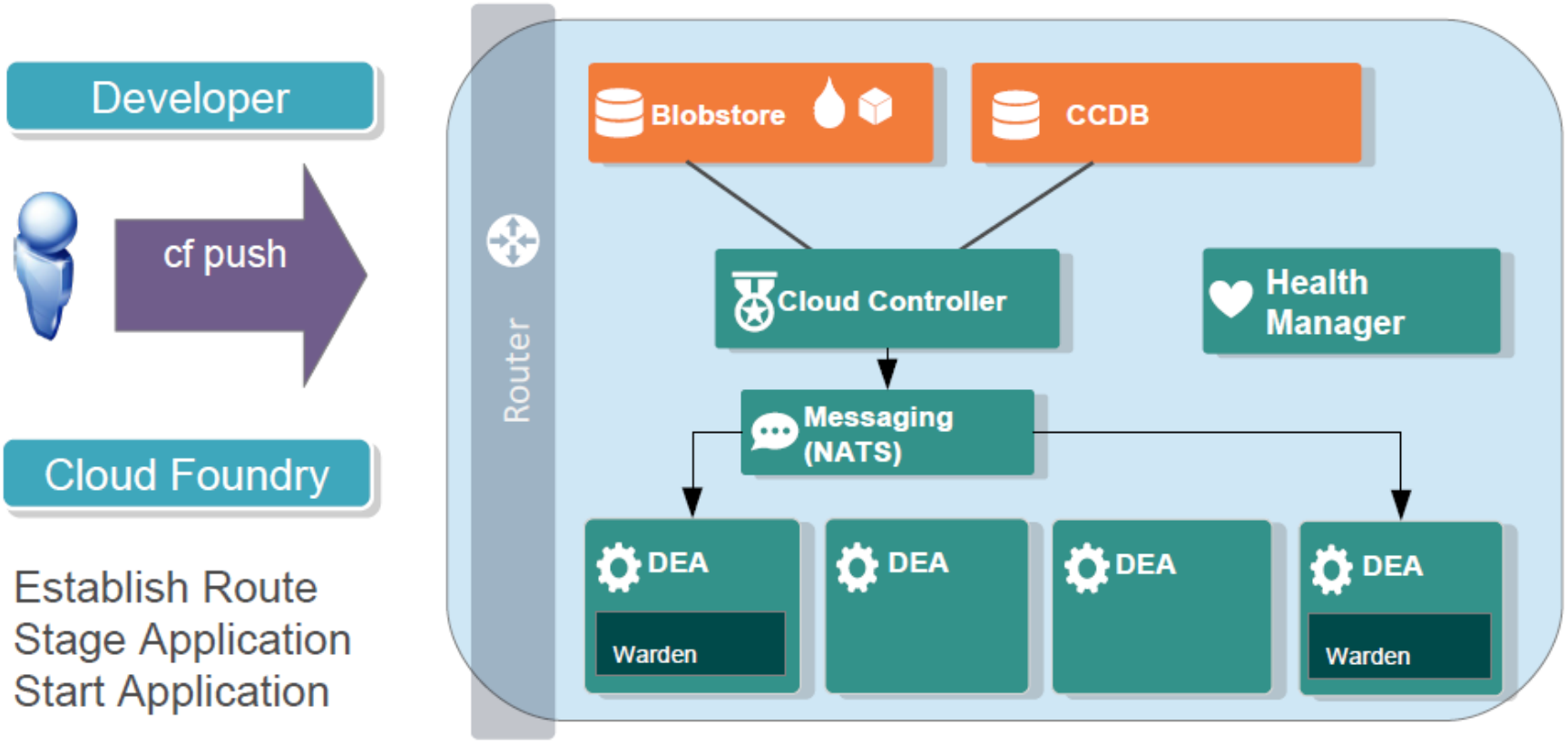
- Accept the license agreement
- Click finish
- Installer runs (takes a while)
- Eventually you are asked to restart Eclipse

How Deployment Happens

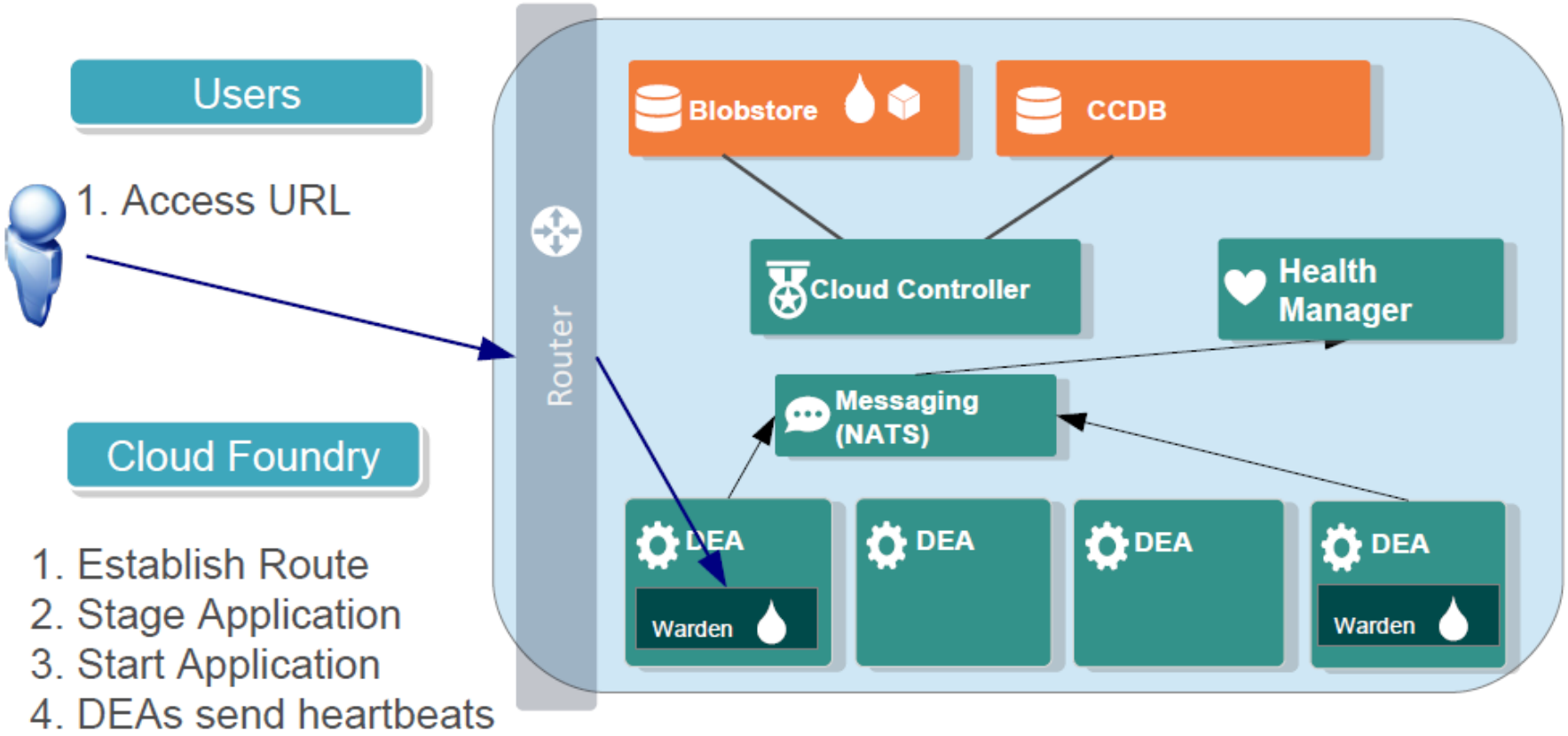
Deploying App to Cloud Foundry *(Push & Staging)*



Deploying App to Cloud Foundry *(Starting)*



Deploying App to Cloud Foundry *(Health Management & Running)*



Recap

manage

Web-based

install

STS

Eclipse

Dashboard

Report

Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING

People matter, results count.

About Capgemini

With more than 130,000 people in 44 countries, Capgemini is one of the world's foremost providers of consulting, technology and outsourcing services. The Group reported 2012 global revenues of EUR 10.3 billion.

Together with its clients, Capgemini creates and delivers business and technology solutions that fit their needs and drive the results they want. A deeply multicultural organization, Capgemini has developed its own way of working, the Collaborative Business Experience™, and draws on Rightshore®, its worldwide delivery model.

www.capgemini.com