Lab 3b: Binding services with manifest files

Description: This is the same lab as “Lab 3a: Services” but it uses manifest files to bind the application to the desired service.

1. Using the same application and services you already have created on the previous lab, check the manifest file at the “manifest” directory of “Sample Apps”.

*$ more manifest.yml*

*---*

*applications:*

*- name: PaaSTutorial*

*memory: 256M*

*services:*

*- ocdadb*

Update the manifest file using your text editor of choice to reflect the application name you want, number of instances desired, path of the war file, hostname, etc.

The manifest for the app being pushed by user 10 could look like:

*---*

*applications:*

*- name: PaaSTutorial10*

*memory: 256M*

*instances: 2*

*services:*

*- mymongo\_user10*

The manifest above assumes you’ve already created the service

“MyRabbit\_user10”.

1. Now from the manifest file directory, just execute the cf push command:

*cf push*

The CloudFoundry CLI should verify the manifest file and re-push the app

using the options provided and bind to the service specified.

1. Check your application is running fine.

Login to the Web Console and verify the App health or just use the command line:

*cf app PaasTutorial-userX*

Acccess the application URL and verify it was deployed and bound to the service specified.

<http://paastutorial.cfapps.io/>

Check the logs to ensure there are no errors in the logs in the following way; 1) from the command line and 2) from the developer console.

Command Line:

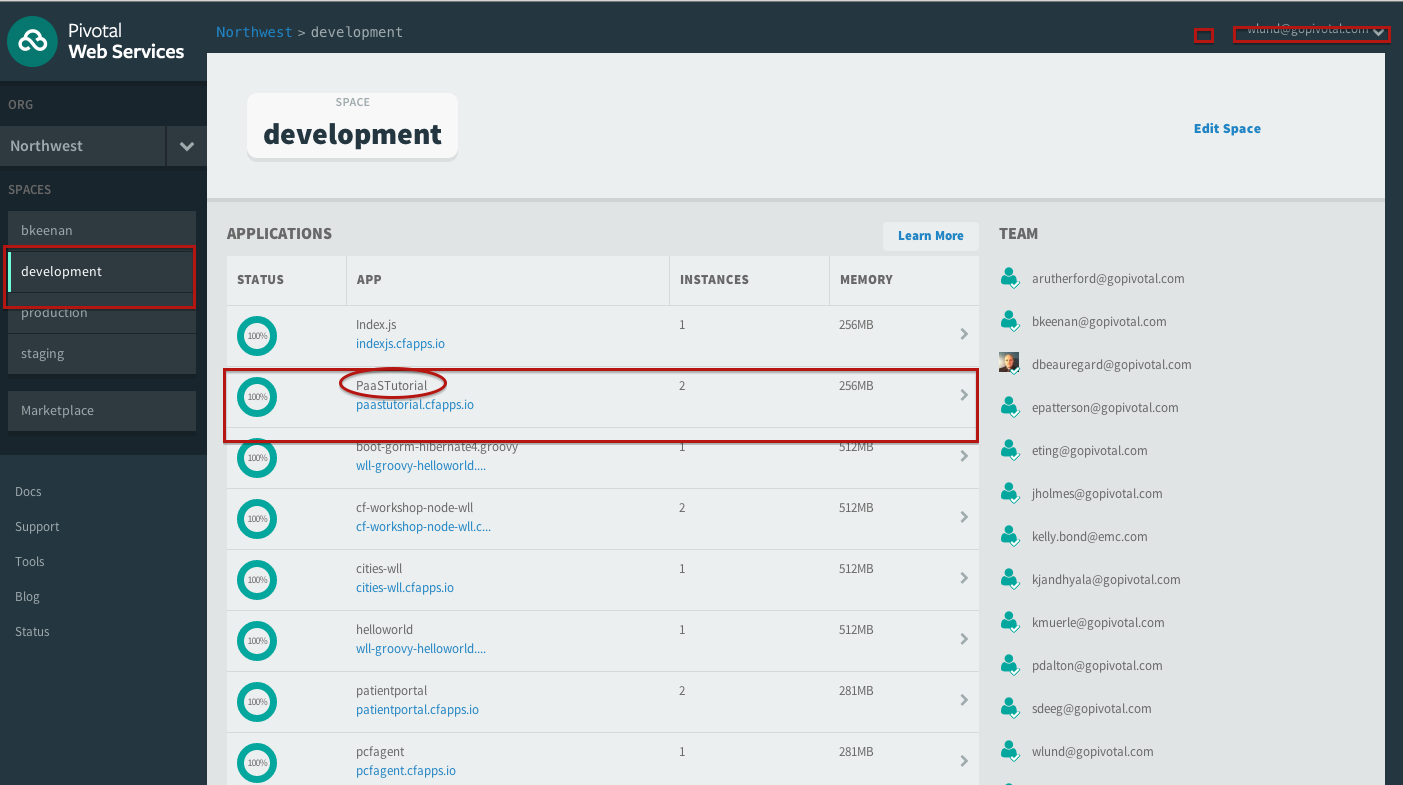
You can get the most recent activity in the log through the following command:

*cf logs paastutorial –recent*

and you can tail the log while its running by dropping –recent.

*cf logs paastutorial*

You can also view the logs via the developer console by ensuring you are logged into your account and space and select the application to manage.



In the developer console you can see the following:

