Lab 3a: Services

Description: This lab will familiarize you with provisioning Cloud Foundry services, binding the service to an application and auto-configuration of your application.

Note that commands are in *italics* and the X should be replaced with a unique number or participant’s initials.

We can start our application without it being bound to the mongo service so the manifest file should look like:

*---*

*applications:*

*- name: PaaSTutorial*

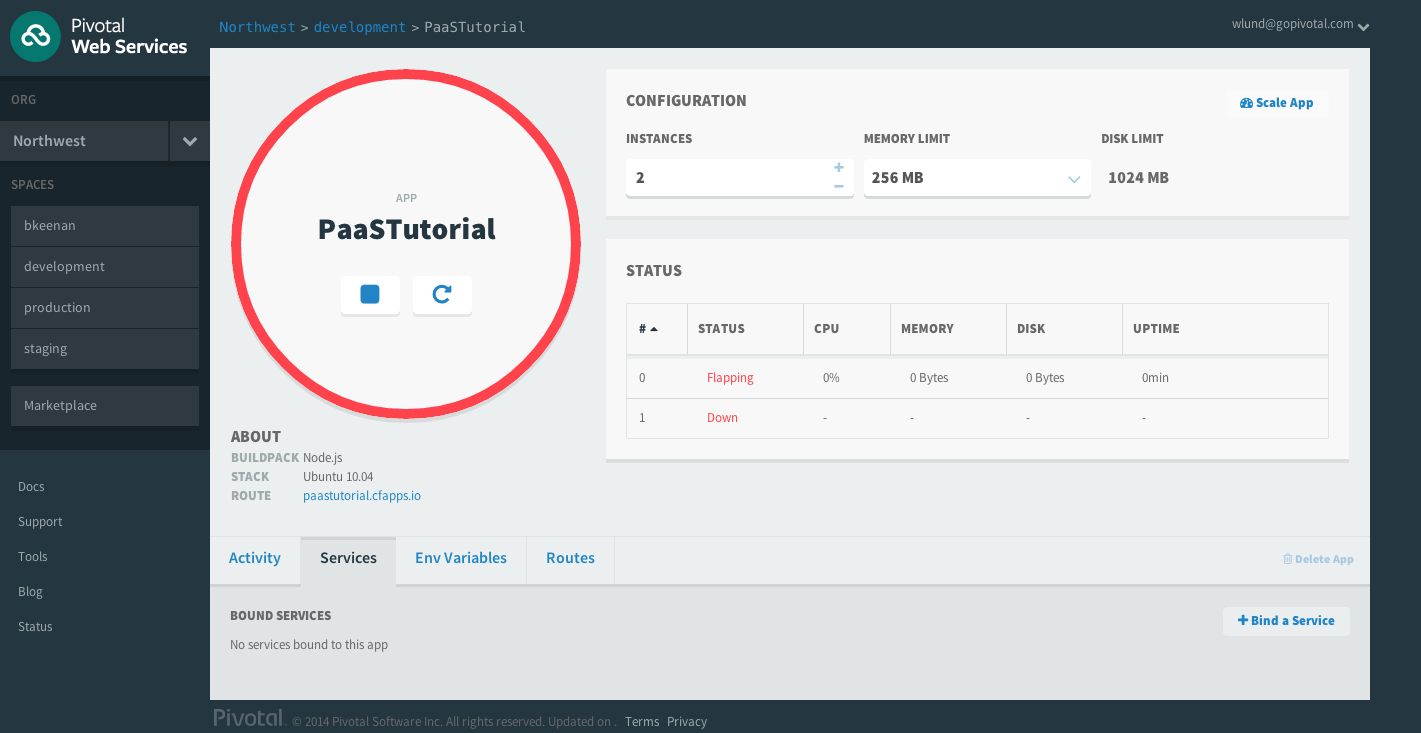
*memory: 256M*

and unbind the service we used for lab2

*cf unbind-service PaaSTutorial-userX mymongo-userX*

*cf delete-service mymongo-userX*

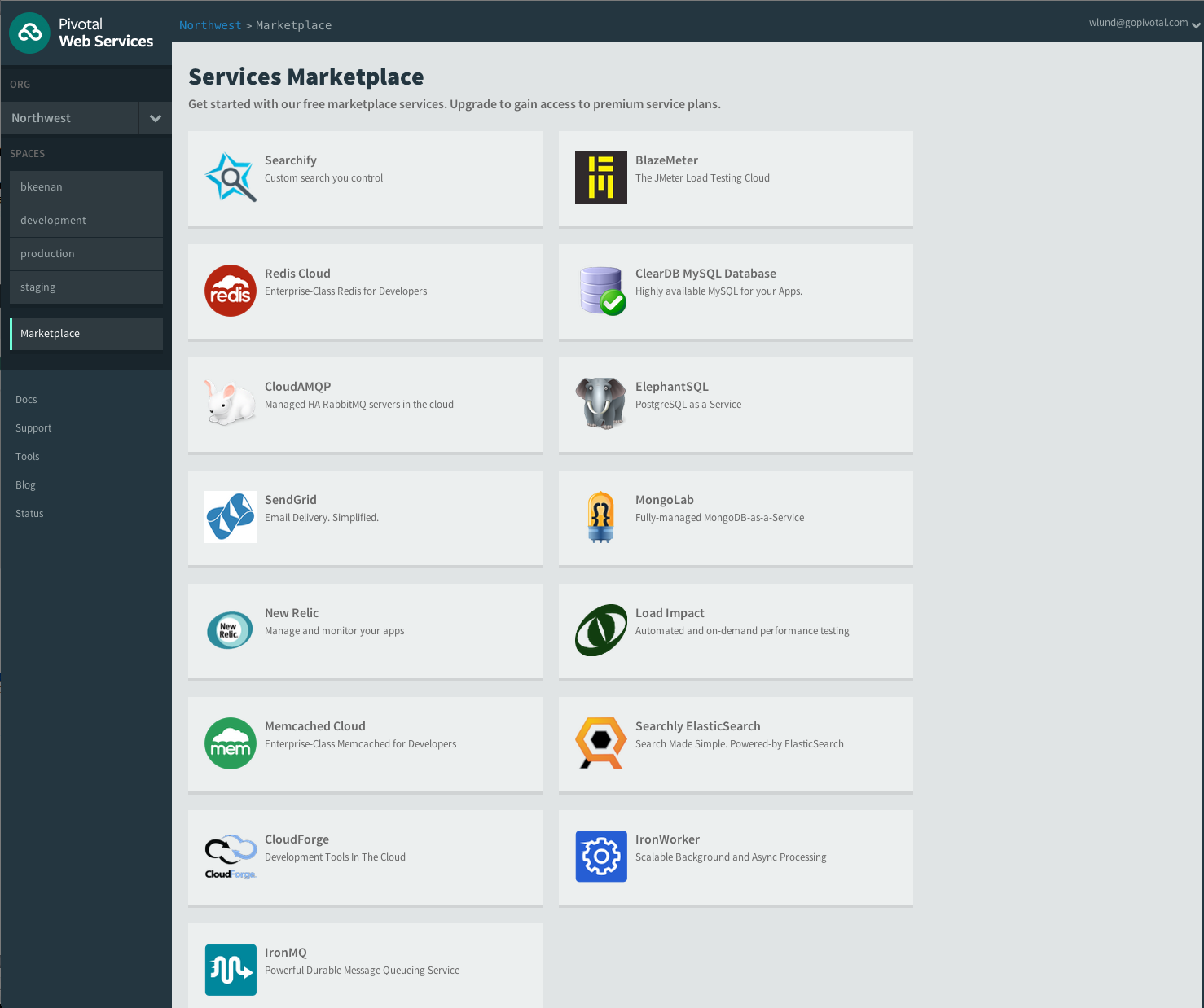
If you push the PaaSTutorial application before its bound to a service (e.g. cf push from odca-paas-workshop root) it will fail the deployment. This can be confirmed by checking the developer console at Pivotal Web Services (<https://run.pivotal.io>).



We will fix this now.

1. Check the services available in the marketplace by selecting “Marketplace” in the left nav bar . These services are installed and maintained from the Operations Manager console. Services have associated plans that the developer selects during the creation process. You will need to log into your account at Pivotal Web Services (<https://run.pivotal.io>).

From the Web Console, click “Marketplace” and select “MongoLab”.

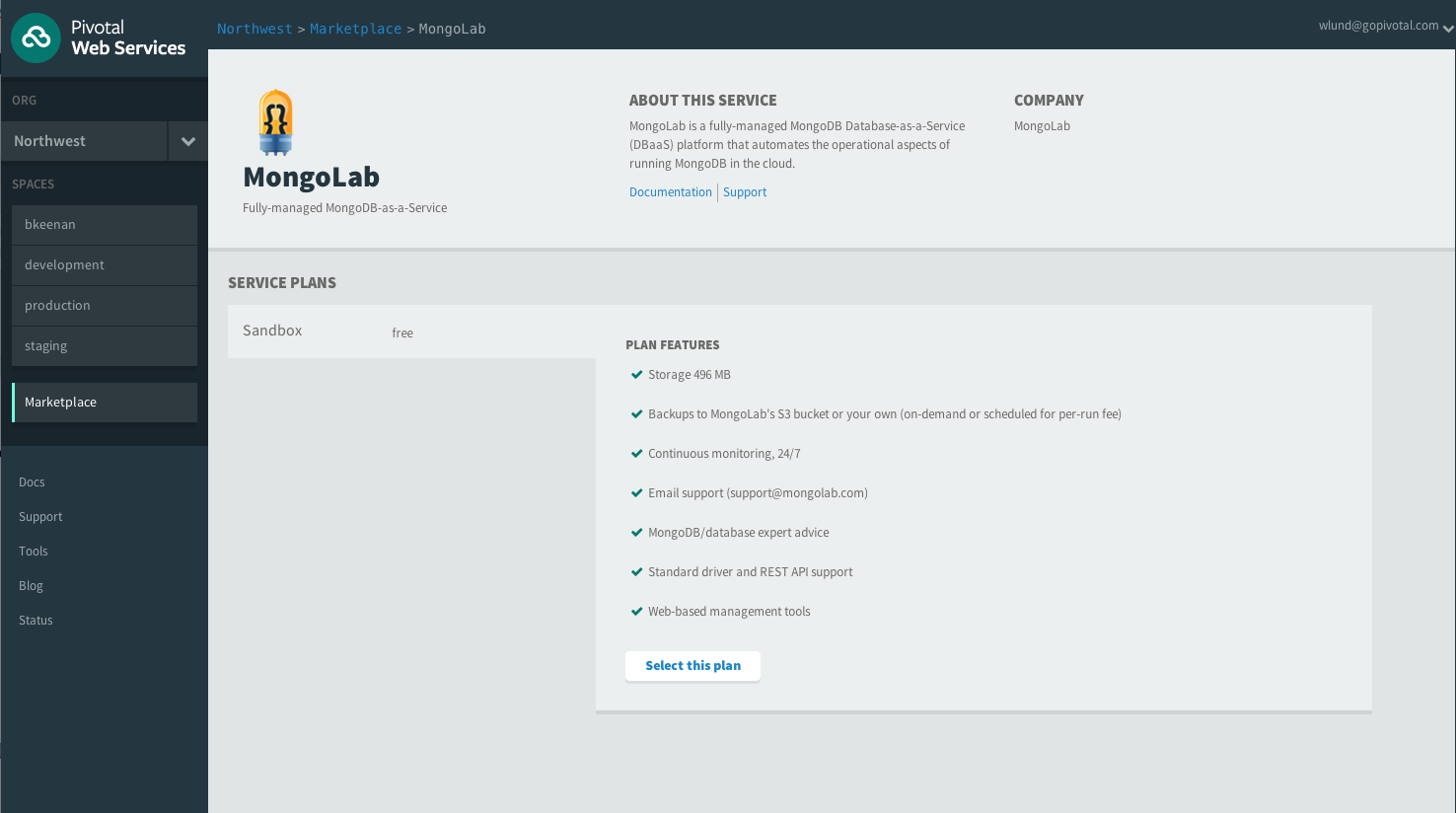


You can also use the command-line:

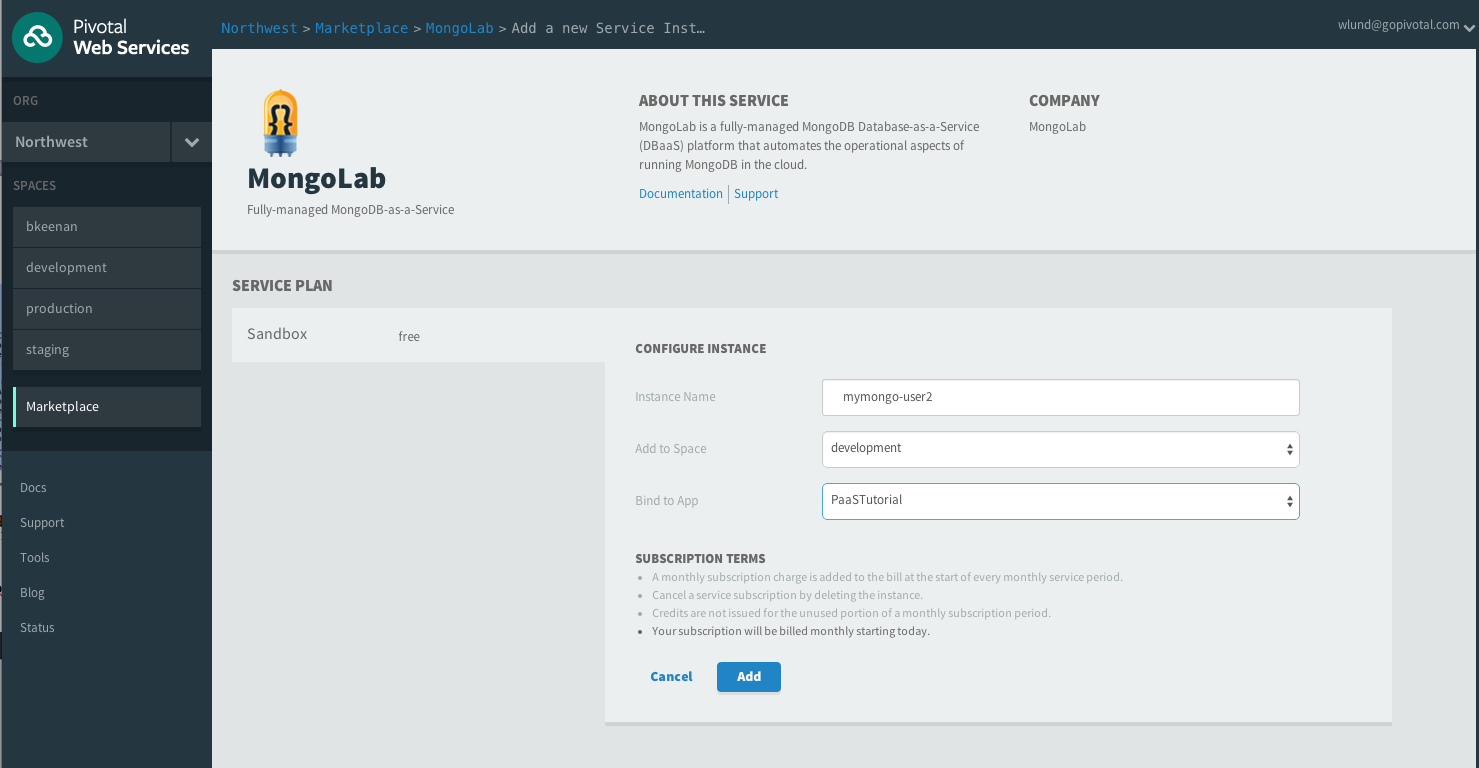
*cf marketplace*

**

1. Create a service to bind to our previous application. Service creation requires a developer to select a plan and a name. Service designers have complete freedom to design a service and make it easy to consume.



Select the “Free” plan and click “Create a Service”



Give the service a unique name such as “mymongo-userX”, select the space you’re deploying your applications to and bind it to your app. In lab 2 we took the short cut listed below so you will want to select another way to identify your service

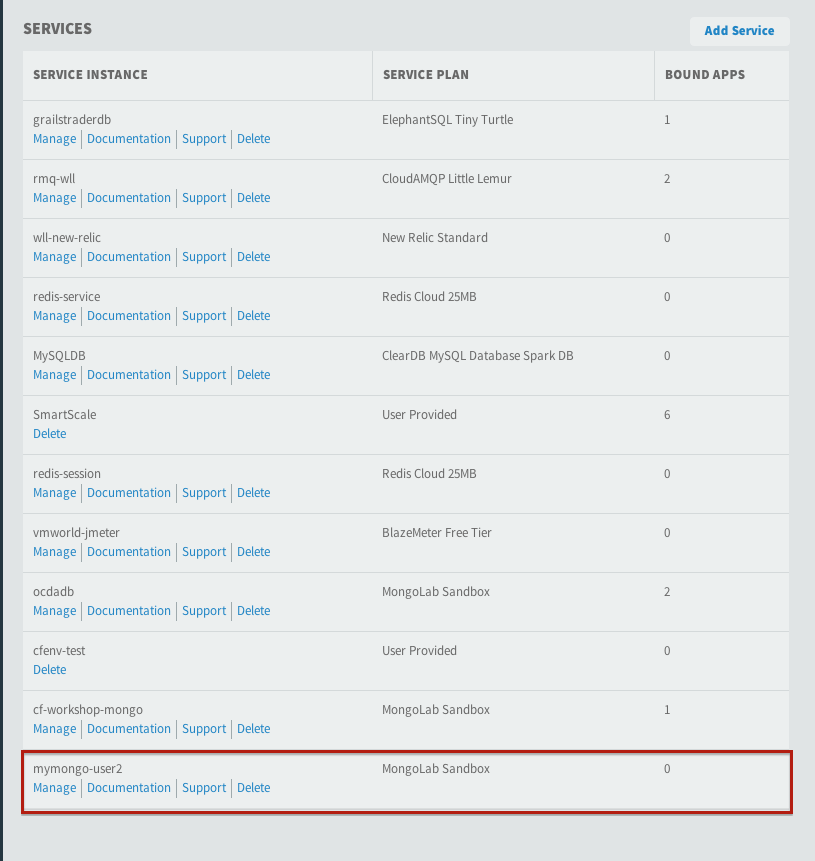
You can also use the command line to create the service

*cf create-service mongolab sandbox mymongo-userX*

and bind it to the app

*cf bind-service PaaSTutorial-userX mymongo\_userX*

1. We now have the required service available to be able to deploy our sample app. On the application list page you will see your new service at the bottom of the page.



We need to restart the application to allow auto-configuration to bind the application variables to the service instance. We can either do that from the developer console by stopping the app and then restarting from the app details or from the command line.

From the command line simply execute  
  
*cf push*

Normally we could do cf restart and ensure that the service is back in the manifest.yml because it was not successfully deployed a start is necessary.

*---*

*applications:*

*- name: PaaSTutorial*

*memory: 256M*

*services:*

*- mymongo-userX*

The mongo service in your lab should be something like mymongo-userX.

1. In the next lab we will tell our application what the service is via a manifest file and prepare to launch the application without having had to install mongo. Its simply available as a service.