Lab 1: Push an Application

Description: During this lab we will push an application, customize our push and review the files generated during the pushing process. We will also ensure that the developer prerequisites are installed to enable the application to run both locally and within cloudfoundry.

1. Verify node is installed and acquire the application.

Node can be obtained from <http://nodejs.org/download/>. Select the right installer for your platform. After you have installed you can verify the installation with:

*node –version*

*output v0.10.29*

*npm –version*

*output:*

*{ http\_parser: '1.0',*

*node: '0.10.29',*

*v8: '3.14.5.9',*

*ares: '1.9.0-DEV',*

*uv: '0.10.27',*

*zlib: '1.2.3',*

*modules: '11',*

*openssl: '1.0.1h',*

*npm: '1.4.14',*

*'PaaS-Workshop': '0.0.1' }*

Now you can checkout the sample application from github with the following command to a directory of your choice (e.g. for example /home/<username>/git-projects (linux) or /Users/<username)/git-projects.

*Git clone https://github.com/wxlund/odca-paas-workshop.git*

Change your directory to the git project and install the packages required by the application.

*cd odca-paas-workshop*

*npm install.*

You can test the application by running locally with

*node web.js*

and checking for “Hello World!” in your browser at *<http://localhost:4000>.*

1. Verify your access to Pivotal Web Services. You will set the API endpoint to Pivotal Web Services (api.run.pivotal.io) and login in with your credentials.

*cf api httsp://api.run.pivotal.io*

*cf login*

1. Change to the odca-paas-workshop directory. We will push a simple “Hello World” Node.js application called web.js (web.js).

During this first installation we will see how easy it is to push applications to Cloud Foundry. The command below pushes an application named web.js to Pivotal Web Services (PWS).

All applications pushed to PWS share the same domain name (cfapps.io), so we need to provide a unique name for our application. We suggest adding your initials or a unique number to your application name. The URL to your application will be the appName.cfapps.io. Custom domains can be added to your account and reference the internal URL.

The first step is to open the manifest-web.yml file using your favorite editor and modify the application name to be unique.

---

applications:

- name: webjs-<user number or initials>

memory: 256M

Then the app can be pushed to your account in cloud foundry with the following command:

*cf push -f manifest-web.yml –i 2*

Open a web browser and visit http://web.js-<user number or initials>.cfapps.io. There is not much to see yet but you’ll notice that for the developer there was no environment to install. Node.js is a supported “buildpack” and was ready and available for deployment. We only had to push our app to enjoy the benefits.

1. When an application is pushed, a number of steps are completed and we want to see what the platform has configured. You can verify that you’re application is running within your space by the following command:

*cf apps*

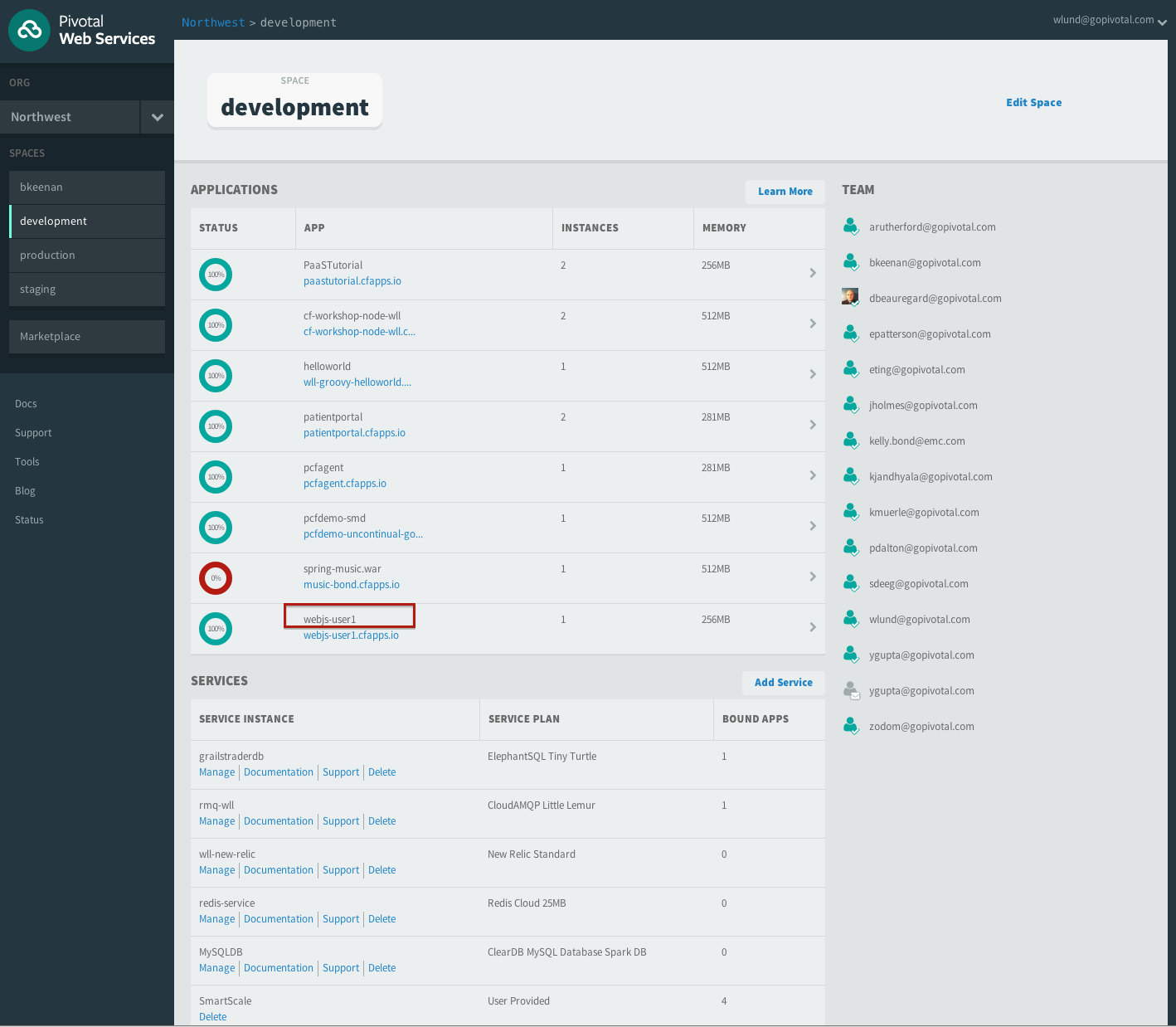
This is only a “Hello World” application but you can view the details of your app with:

*cf app webjs-user1*

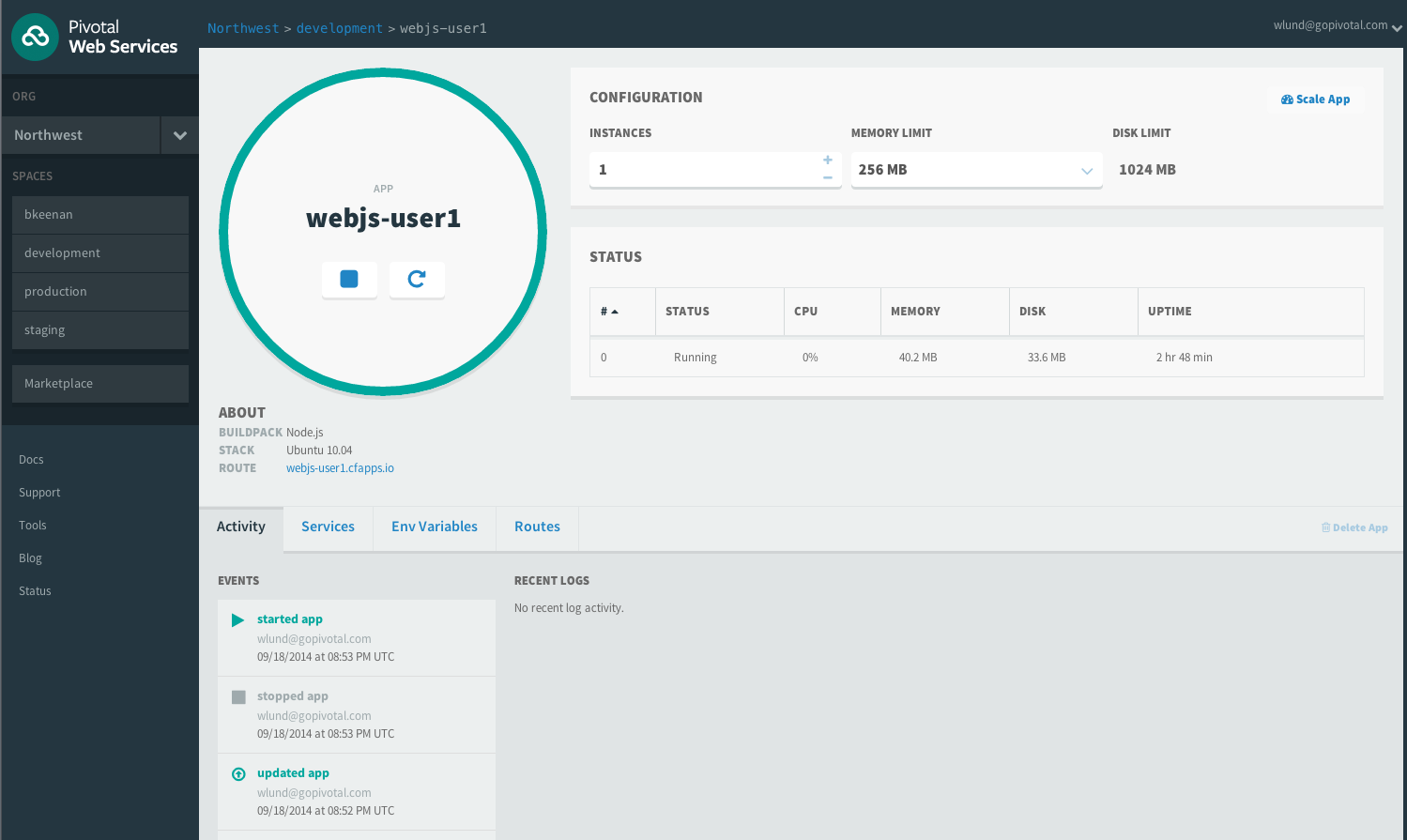
Tuck this command way as you’ll use it later to view the services that we will be binding to it.

1. Now log into “run.pivotal.io” and let’s check the health, running instances, route and other details for the application we’ve pushed. Access http://run.pivital.io with your credentials.

You’ll need to select the same space you’ve first targeted your environment to.



Now you can select your app as highlighted and see the details.



That completes how to push an application to cloudfoundry.