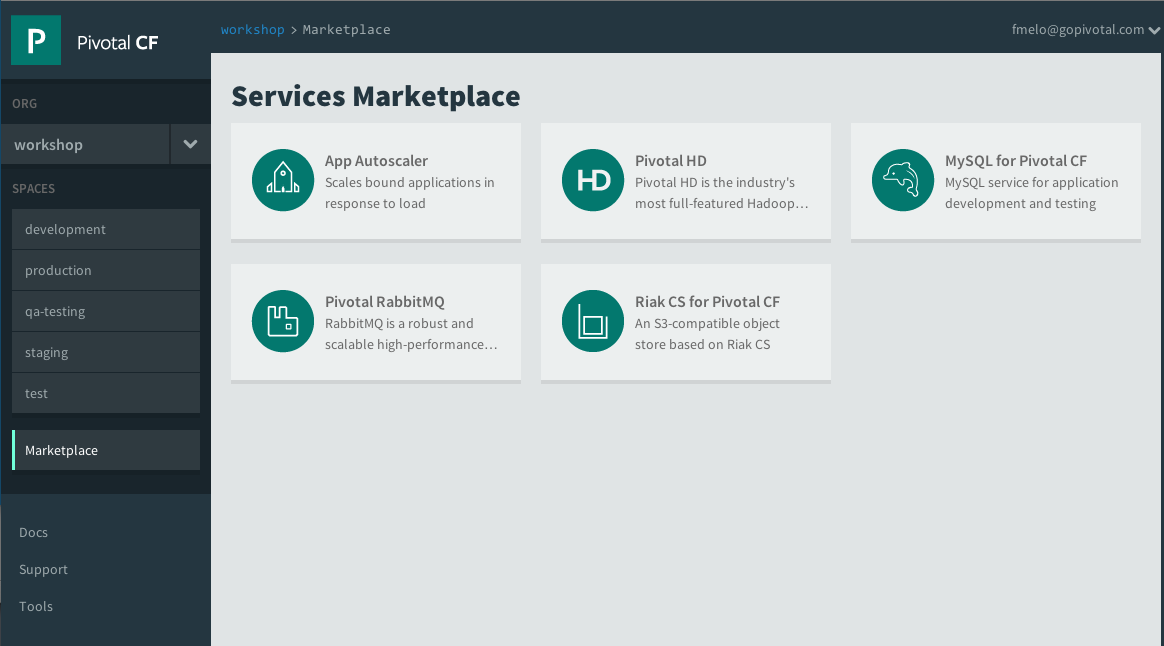
Lab 3: Services

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

1. Check the services available in the marketplace. These services are installed and maintained from the Operations Manager console. Services have associated plans that the developer selects during the creation process.

From the Web Console, click “Marketplace”

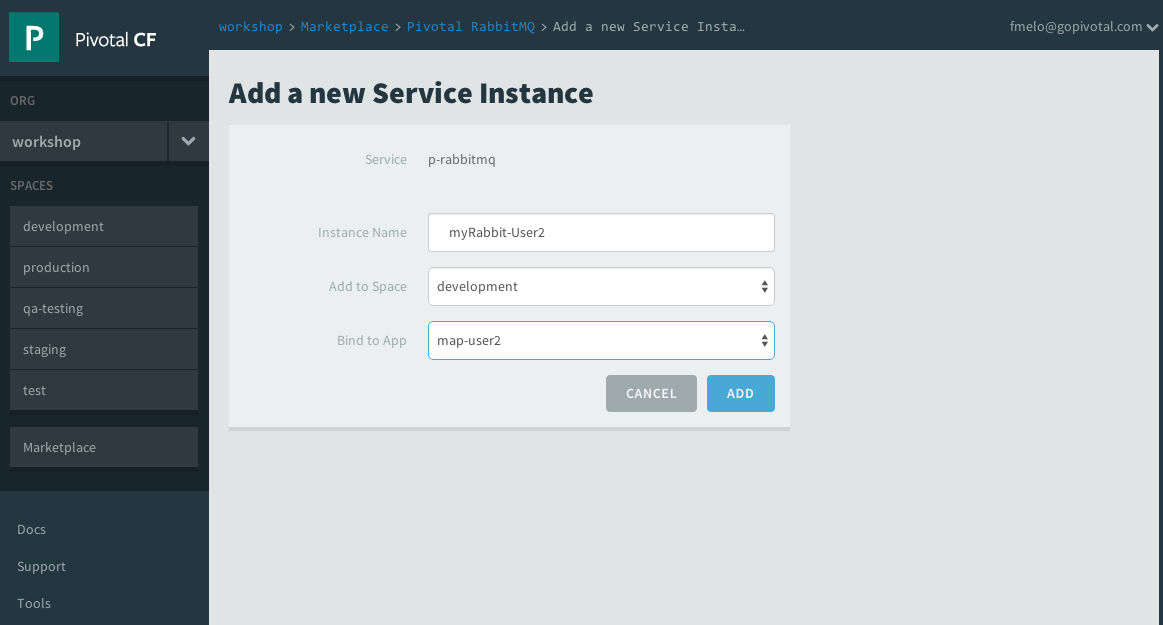


You can also use the command-line:

*cf marketplace*

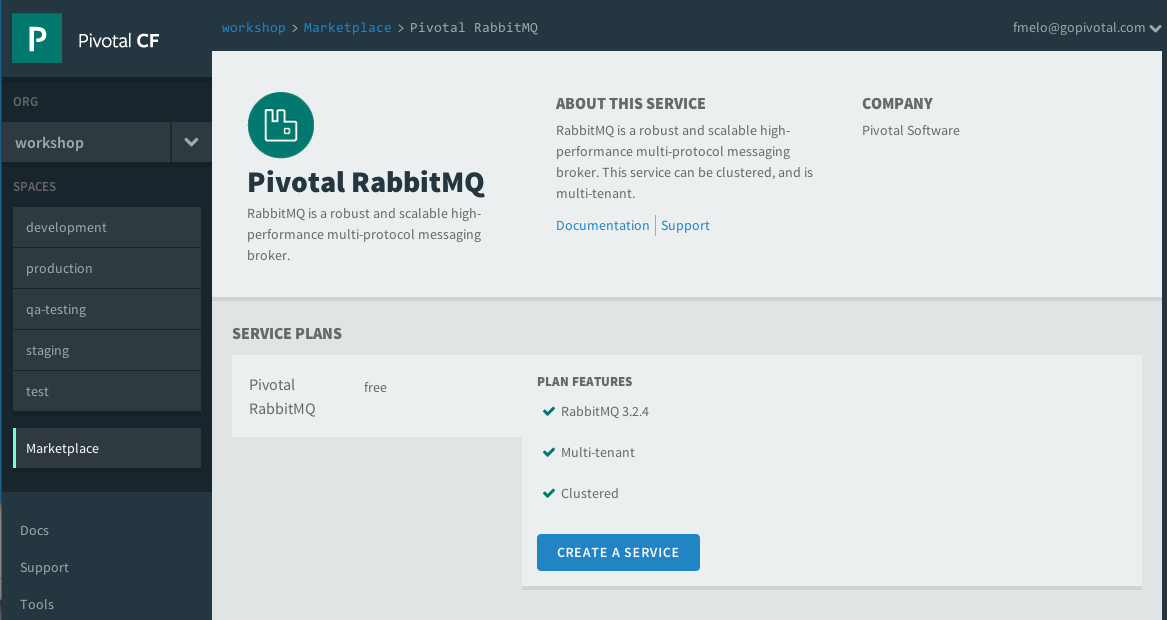
1. Create a service to bind to our previous application. Service creation requires a developer to select a plan and a name. Service designers have complete freedom to design a service and make it easy to consume.

On the Marketplace page of the Pivotal CF Web Console, click “Pivotal RabbitMQ”.



On the “Free” plan, click “Create a Service”

Give the service a unique name such as “myRabbit-userX”, select the space you’re deploying your applications to and bind it to your app:



You can also use the command line to create the service

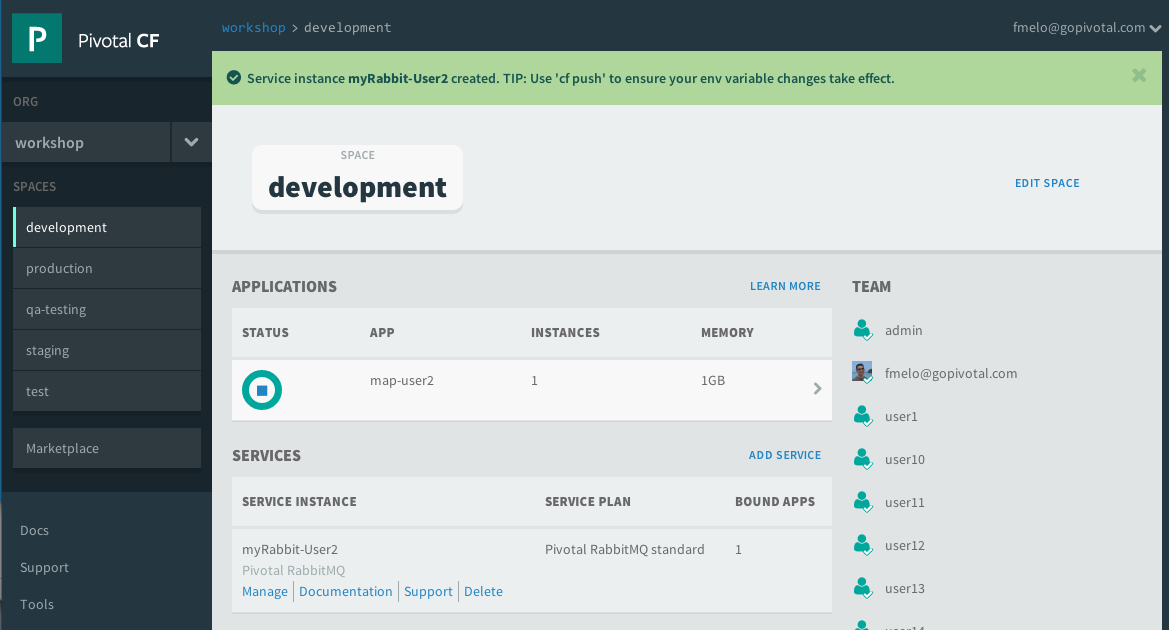
*cf create-service p-rabbitmq standard myRabbit\_userX*

and bind it to the app

*cf bind-service map-userX myRabbit\_userX*

1. We need to restart the application to allow auto-configuration to bind the application variables to the service instance.

On the application detail page, click on the health green circle to stop and then click again to restart the app



Refresh the page once its restarted to get the most updated health information.

The same can be done using the command line as:

*cf restart map-userX*

1. Let visit the application and click the “Data Streaming” button. This will start a flow of order events to the application. Notice the application informs us that a RabbitMQ service is bound to the application. It got this information from the vCAP\_SERVICES environment variable.

Open a browser and visit <http://map-userX.cfdemo5.fe.gopivotal.com>

Select a state to see the order event flow into the application.

