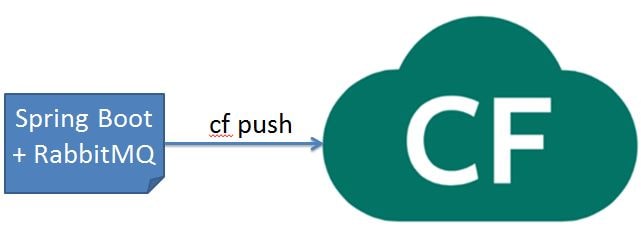
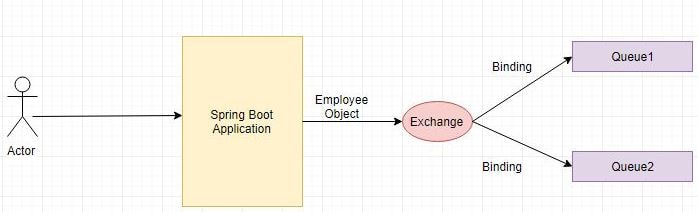
**Pivotal Cloud Foundry Tutorial - Deploying Spring Boot + RabbitMQ Application to PCF**

In previous tutorial we had deployed a [Spring Boot + MySQL application to PCF](https://www.javainuse.com/pcf/pcf-rabbitmq). Let us now look at some PCF concepts like Orgs, Spaces.  
In this tutorial we will develop a Spring Boot + RabbitMQ Application and deploy it to PCF.

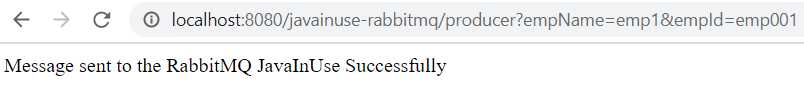


[In a previous post](https://www.javainuse.com/misc/rabbitmq-hello-world) we had seen how to get RabbitMQ up and running.  
  
The project will be as follows-

Define the pom.xml as follows- Add the **spring-boot-starter-amqp** dependency.

Define the RabbitMQSender class which sends the message to the RabbitMQ using AmqpTemplate. We use the exchange and the exchange key.  
Exchanges are message routing agents, defined per virtual host within RabbitMQ. An exchange is responsible for the routing of the messages to the different queues. An exchange accepts messages from the producer application and routes them to message queues with help of header attributes, bindings, and routing keys.  
We will use a direct exchange instead. The routing algorithm behind a direct exchange is simple - a message goes to the queues whose binding key exactly matches the routing key of the message.

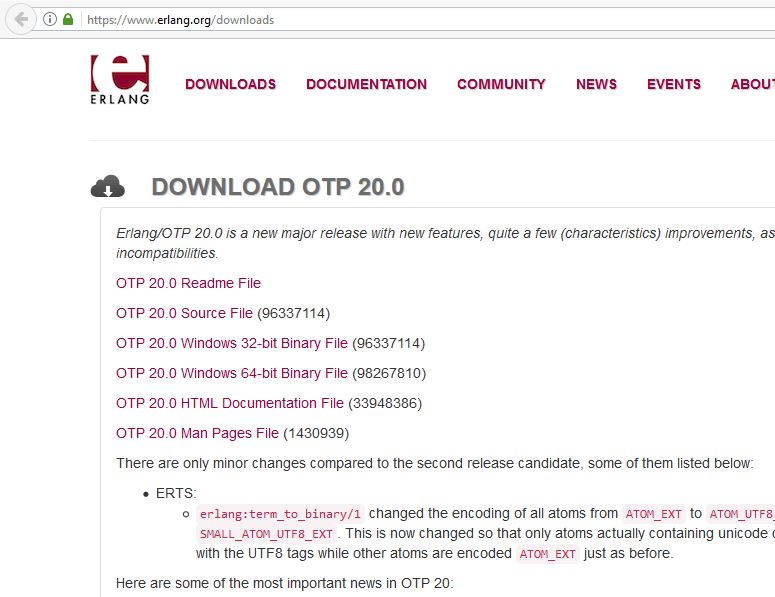
We are done with the required Java code. Now lets start RabbitMQ. As we had explained in detail in the [Getting started with RabbitMQ](https://www.javainuse.com/misc/rabbitmq-hello-world) perform the steps to start the RabbitMQ.  
Next start the Spring Boot Application by running it as a Java Application. Hit the url as follows-

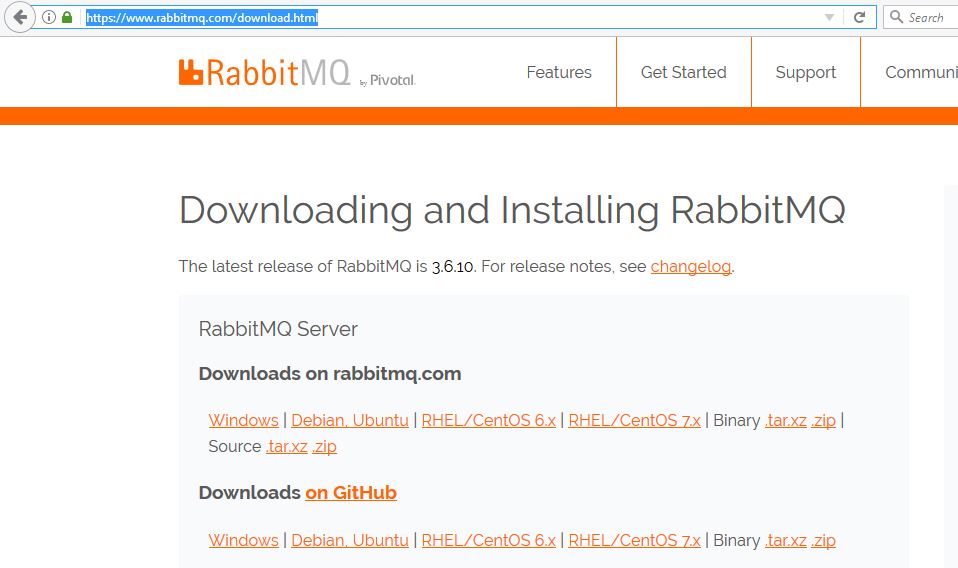
**http://localhost:8080/java-rabbitmq/producer?empName=emp1&empId=emp001**  


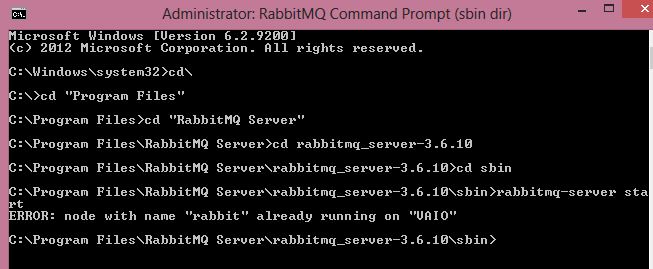
This will trigger the message to be sent to the java queue.

**Lets Begin-**

Since RabbitMQ is built on top of Erlang, we will first need to install Erlang. Got to the [Erlang downloads page](https://www.erlang.org/downloads) and download the erlang binary file for windows which is an executable.

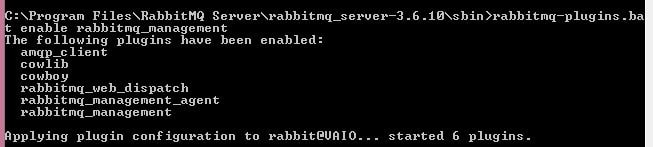


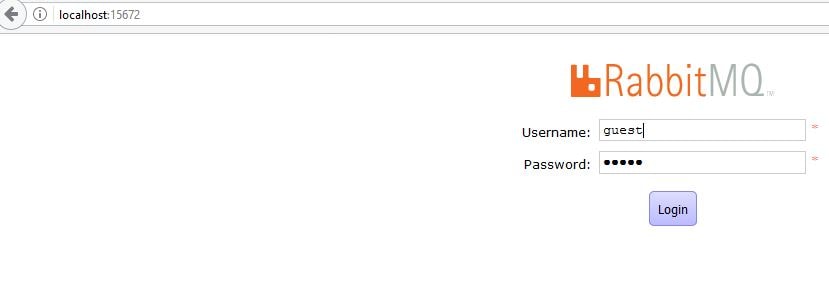
* Next run the binary file downloaded and install erlang on your machine.
* Go to [RabbitMQ downloads page](https://www.rabbitmq.com/download.html) and download RabbitMQ installation.  
    
  This will be a .exe installation file for windows.
* Run this exe and install RabbitMQ on your machine.
* We will now start Rabbit. The above installation should have installed the RabbitMQ command prompt. Open it.  
  Go to the RabbitMQ Server Location and use the command as follows-
* rabbitmq-server start

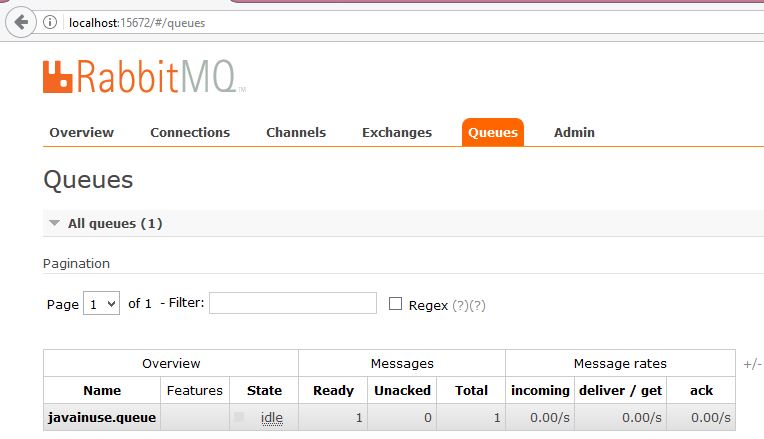
  
If RabbitMQ already running we get the above message.

Next we will install the RabbitMQ plugin which will give use the RabbitMQ Management Console which is accessible using the browser. For this use the command as follows-

* rabbitmq-plugins.bat enable rabbitmq\_management



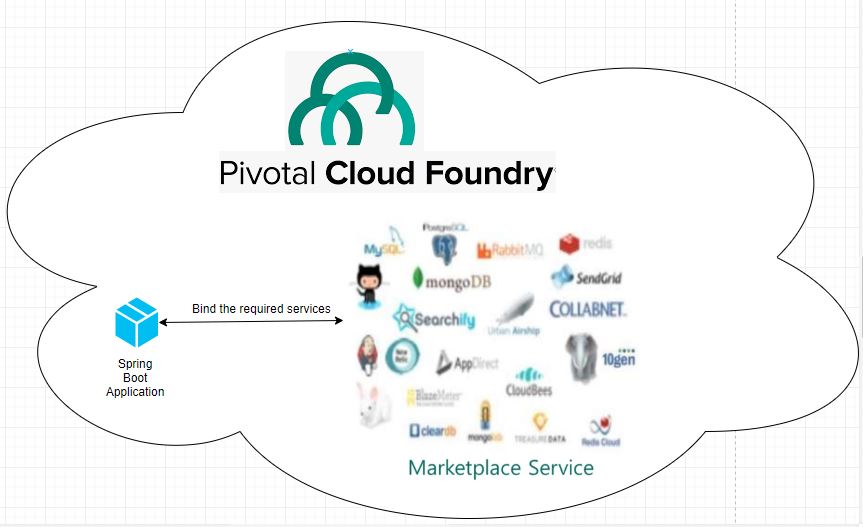
Next go to the RabbitMQ console-**http://localhost:15672/**  


We can see in the Queues section, a queue name java gets created and it has one message.  


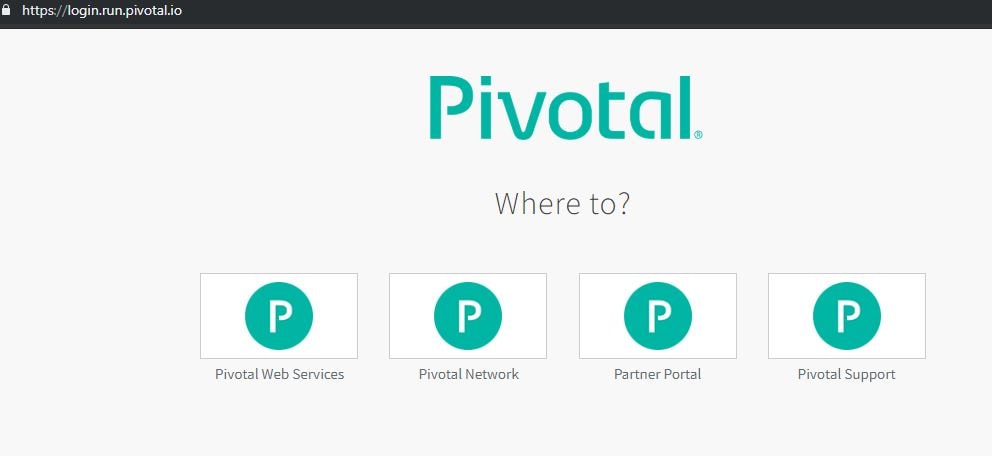
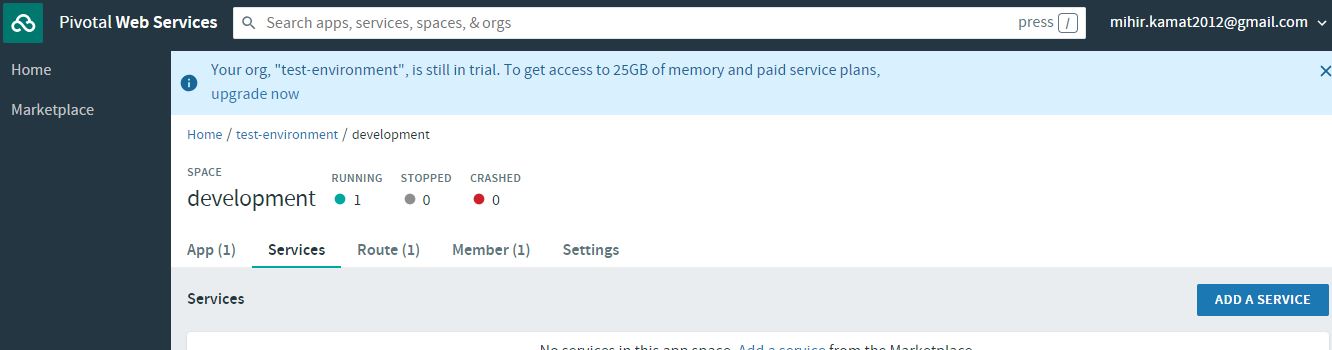
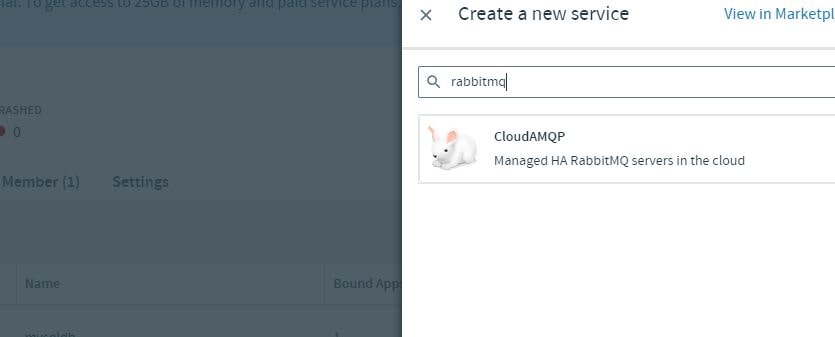
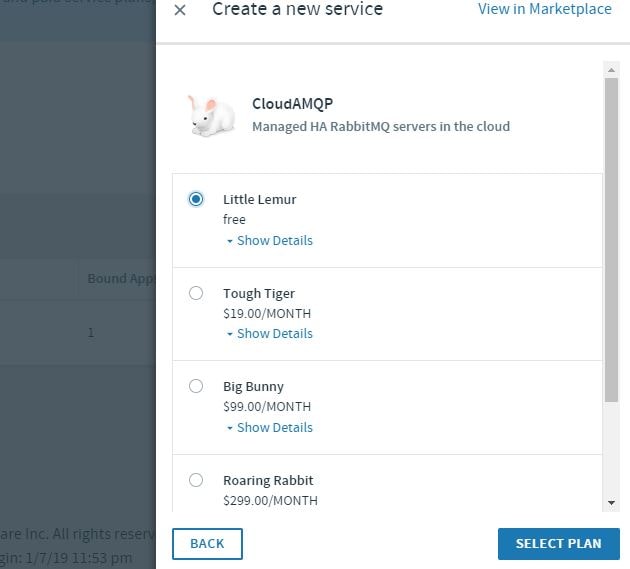
**PCF Deployment-**

In Cloud Foundry, services are on demand resources that users can provision and use for their deployed application.  
Examples of resources services provide include databases on a shared or dedicated server, or accounts on a SaaS application. These resources are known as service instances and the systems that deliver and operate these resources are known as Services. Think of a service as a factory that delivers service instances.  
Services can either be

* User Defined Service
* MarketPlace Service

For this example we will be needing the RabbitMQ MarketPlace service.  


Provisioning RabbitMQ Service using PCF Web Console

* Login to Pivotal Cloud Foundry with your credentials  
  
* Go to Services tab for our development space  
  
* Select ClearDB MySql  
  
* Select the free plan free Spark DB.  
    
  Name the service as **rabbitmq**

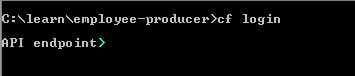
**Creating Manifest Configuration file to Bind our Maven Project to PCF SQL Service**

In our eclipse jdbc project create a manifest configuration file as follows-



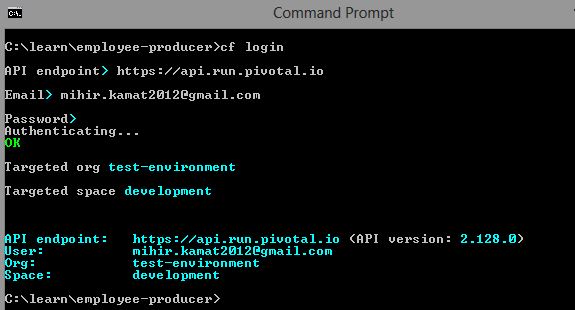
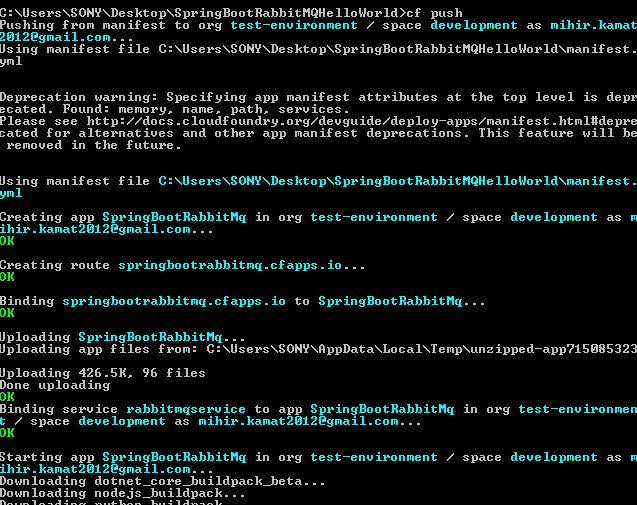
Push the application to PCF

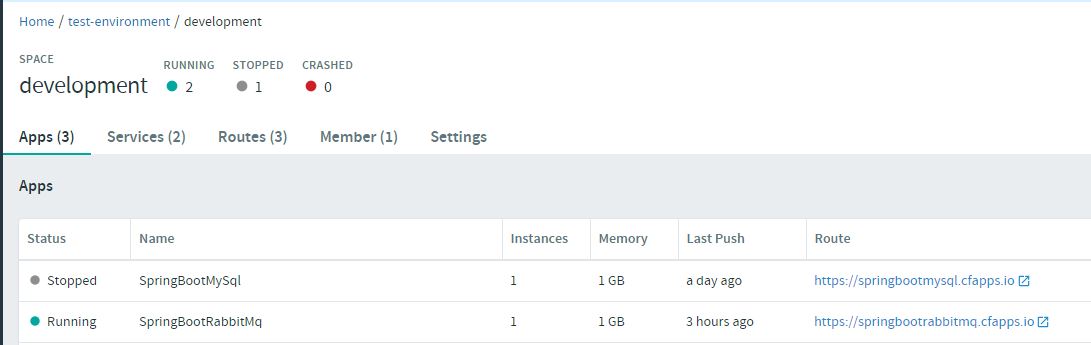
* Open the command terminal and use the following command
* cf login



* It will ask for Cloud Foundry API. Enter The following API value-
* https://api.run.pivotal.io

Cloud Foundry Login API

* Next it will ask you for the Cloud Foundry credentials  
  
* Use the command cf push-
* cf pu   
  

The application is deployed on PCF. Go the PCF Web Console. Our new application will be up and running.  


Select the application and go to the specified route. Append java-rabbitmq/producer?empName=emp1&empId=emp001 to the route url. Our application is up and running.

