MIRACLE
SOFTWARE SYSTEMS

Cloud Foundry

©2016 Miracle Software Systems, Inc.

Goal

In this document, we will introduce you with the Cloud Foundry which is offered as Platform as a service and provides you with some basic hands on lab for getting started with Cloud Foundry.

Pre-Requisites

- CF Command Line interface
- Pivotal Account

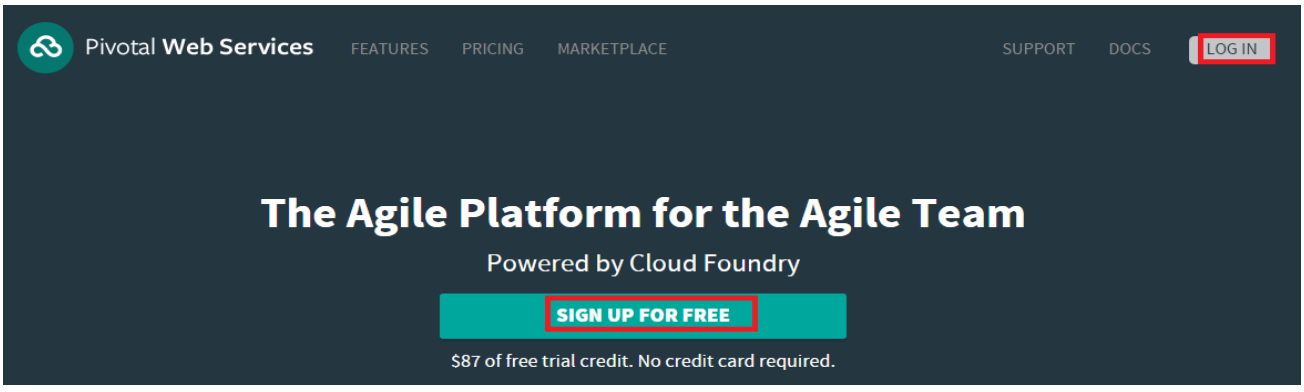
Introduction

Cloud Foundry is an open source cloud platform as a service (PaaS) on which developers can build, deploy, run and scale applications on public and private cloud models. Unlike most of the Cloud Computing platform services, which are tied to particular cloud providers, Cloud Foundry is available as a stand-alone software package, with which we can build Cloud Foundry on any IaaS platform like Amazon Web Services(AWS), OpenStack, VMware,etc. So the developers no need to stick with a particular service provider to run their applications.

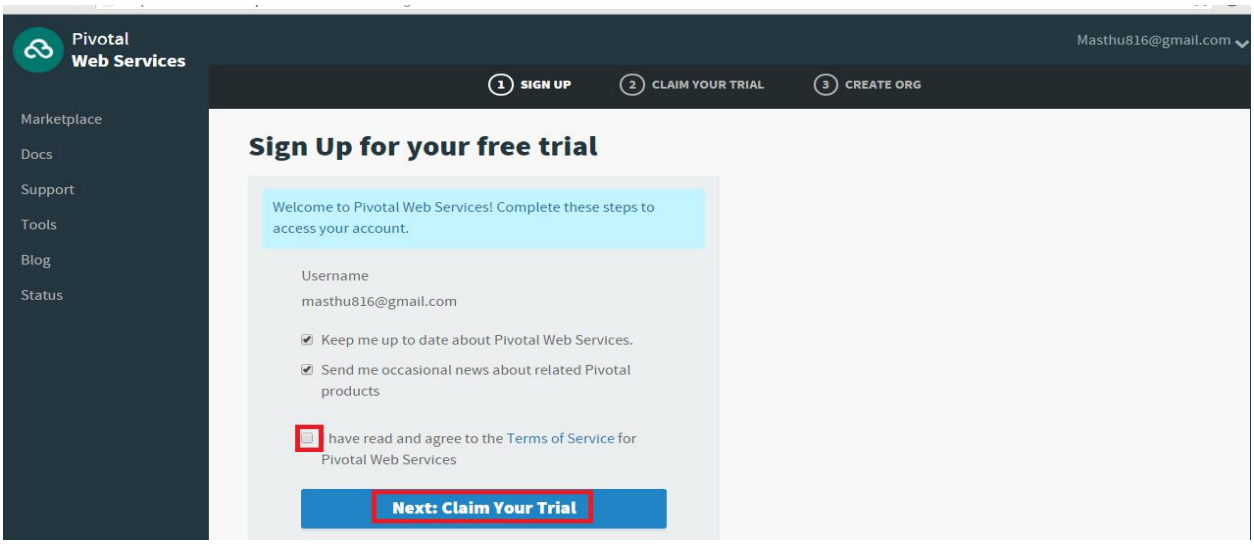
Lab Steps

Creating a Pivotal Account

In order to create an account login to https://run.pivotal.io and click Sign Up . If you already have an account login to the account.



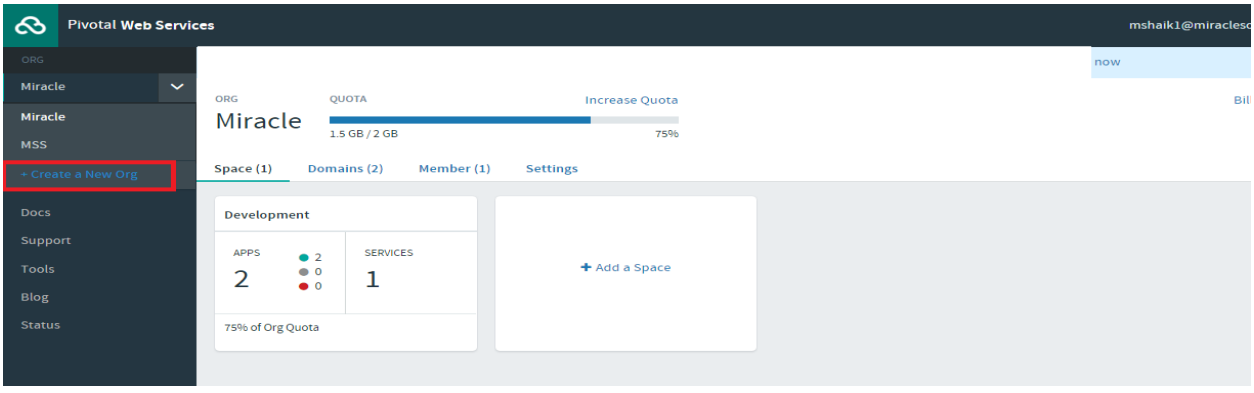
After creating an account, log into it you will be provided with a screen with which you need to claim your free trial account by accepting the agreement. Click on **claim your trial** without providing any credit/debit card details.



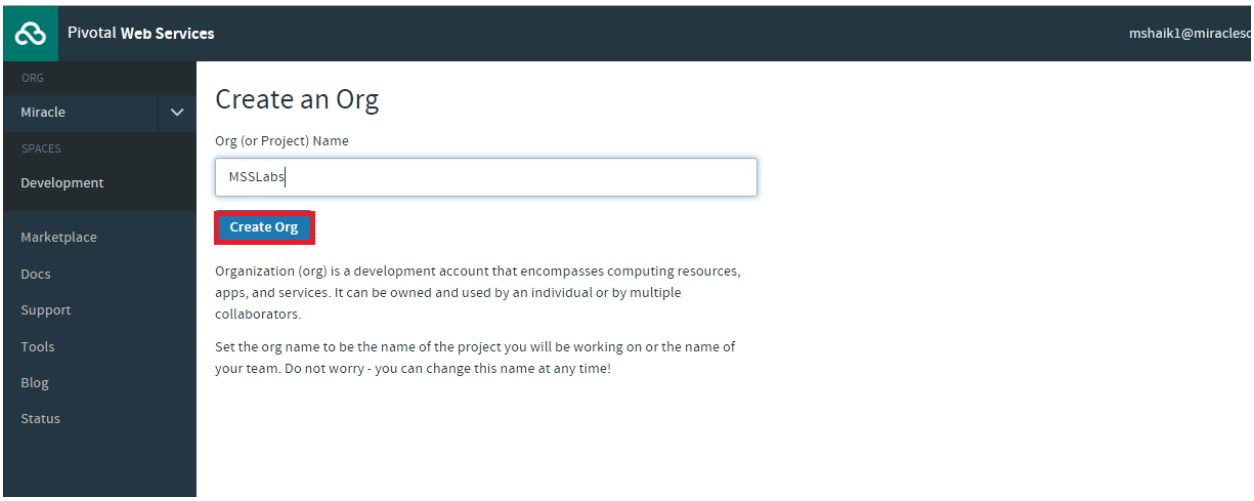
After claiming your trail, you will be allowed to create organizations and spaces within that account.

Organization

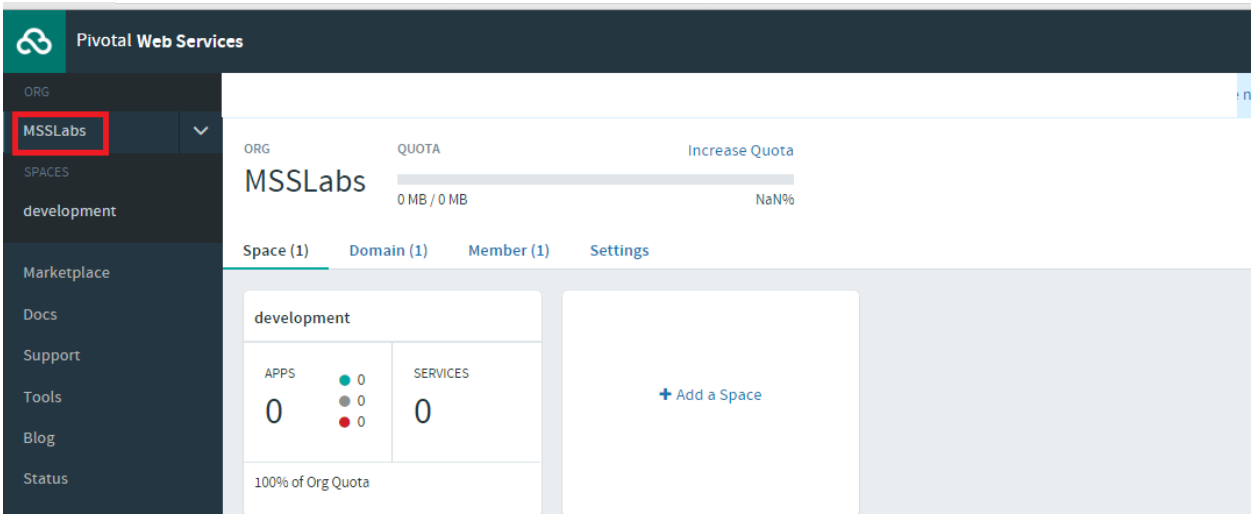
An organization is a development account that an individual or multiple collaborators can own and use. All collaborators access an organization with user accounts. Collaborators in an organization share a resource quota plan, applications, services availability, and custom domains.
Create an organization by clicking on **Create a New Org** option.



After clicking on **Create a New Org** it will ask to enter the name of the organization. Please provide your valid organization name and click on **Create Org**.



After clicking on **Create Org** an organization will be created and you will be able to see the organization.

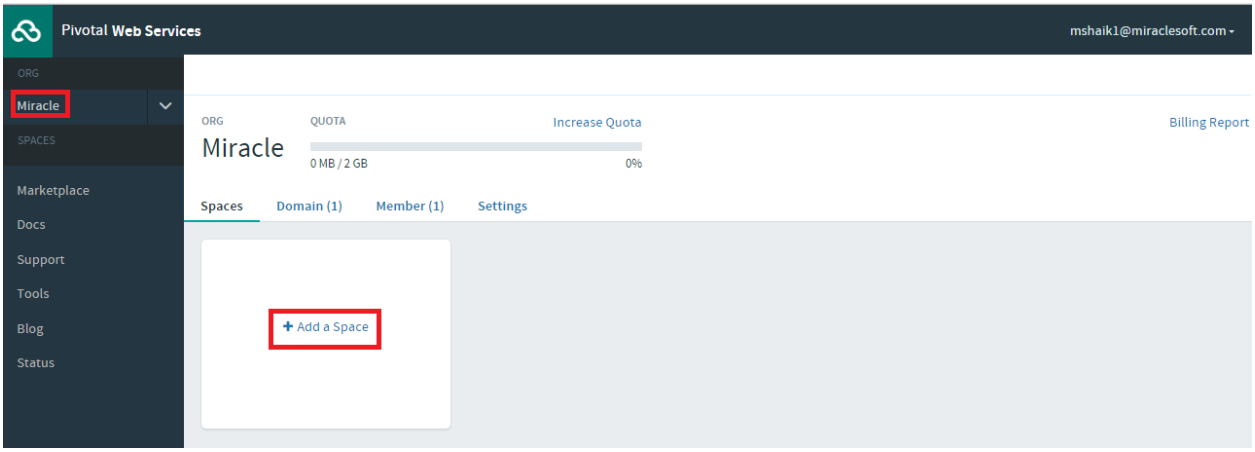


Spaces

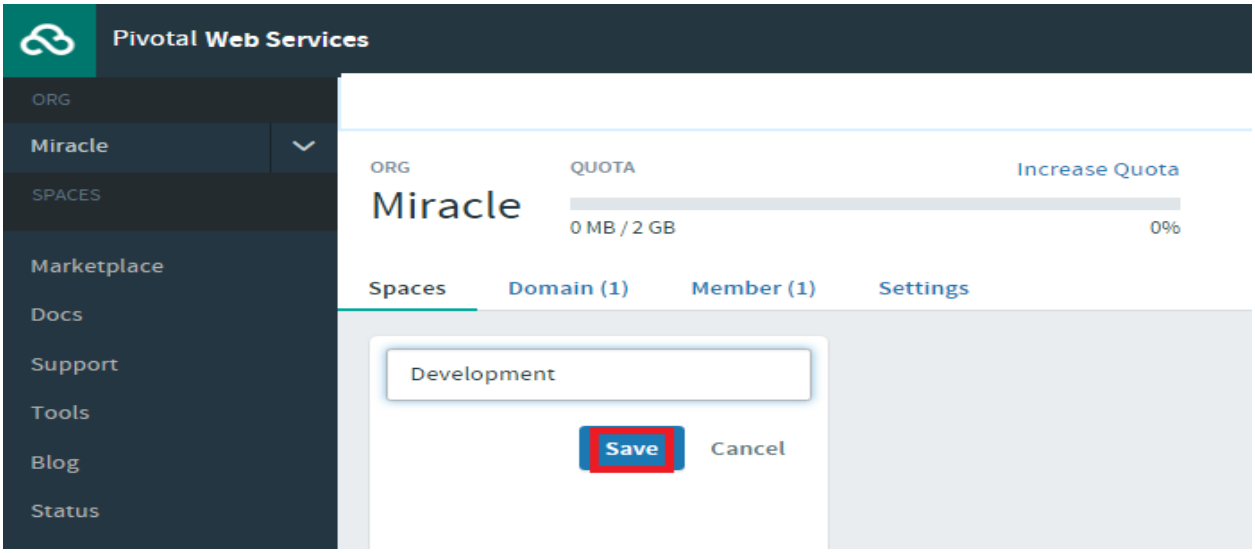
Every application and service is scoped to a space. Each organization contains at least one space. A space provides users with access to a shared location for application development, deployment, and maintenance. Each space role applies only to a particular space.

For suppose, we have a project named **Employee Payroll** then we can consider it as an organization entity. Spaces can be considered as different phases in a project like development, testing, staging. Then we need to add individual user to their respective space and manage their access through roles.

Create a space by clicking on the organization name, in that click on spaces then click on **Add a Space**.



It will prompt you to enter the space name. Please select an appropriate space name.

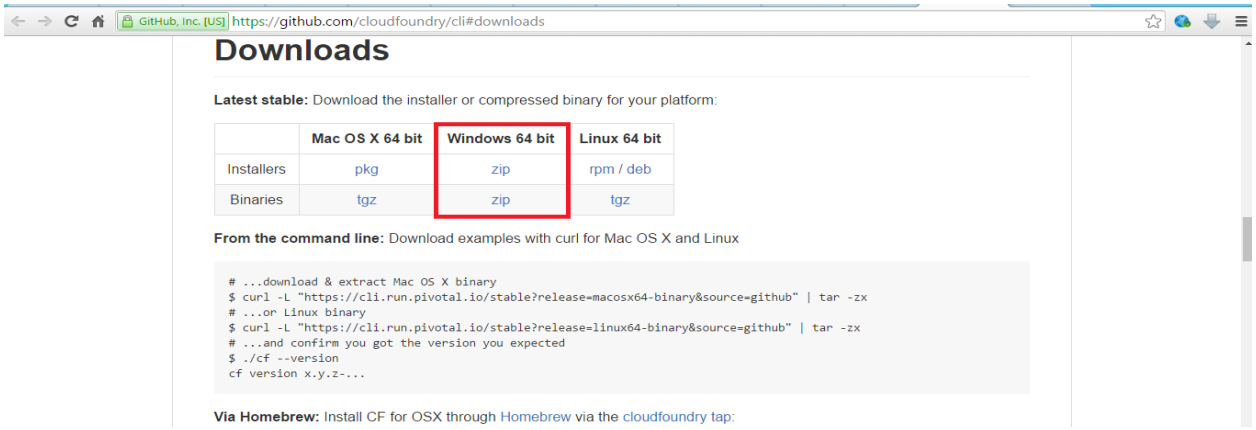


Under the organization that you have created create a space and add users to the space.

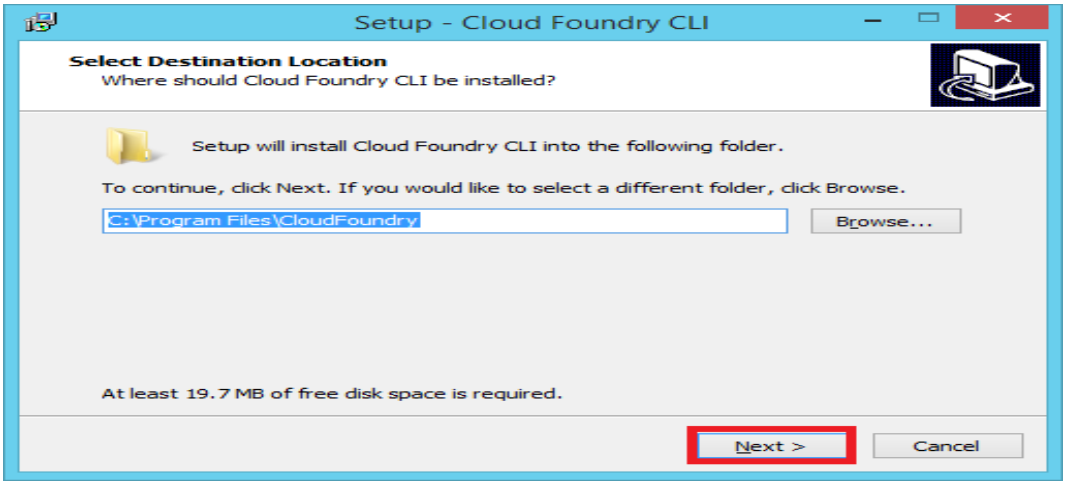
Installing Cloud Foundry Command Line Interface

In order to deploy and configure our applications, services and environment in Cloud Foundry, we need to install Cloud Foundry command line interface and use it to communicate with the Pivotal Cloud Foundry. You can download the executables/binaries from the following link. For demo purpose we are installing CLI in windows.

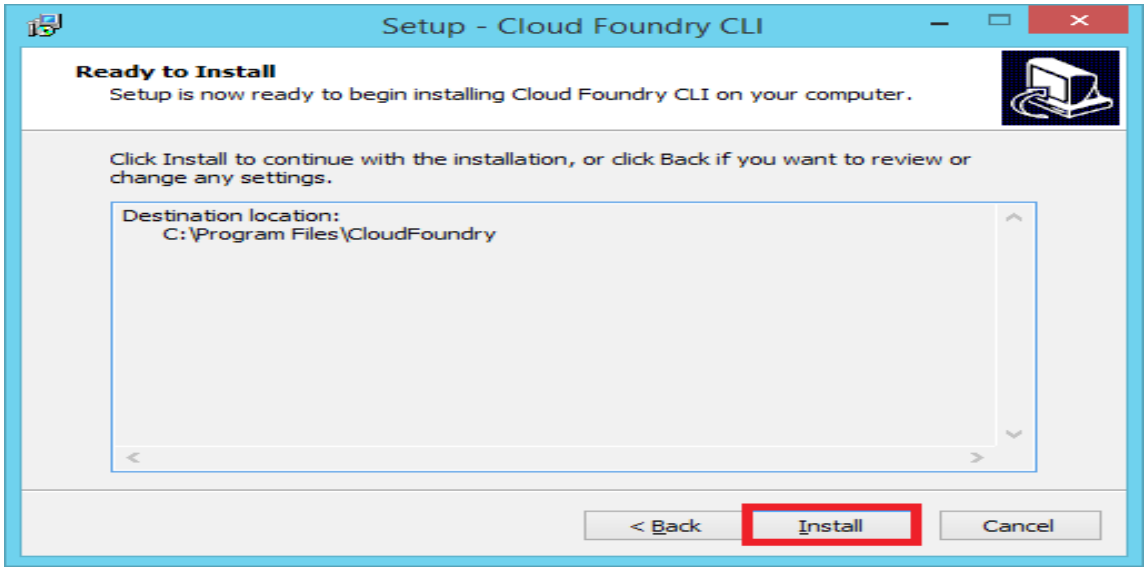
https://github.com/cloudfoundry/cli#downloads



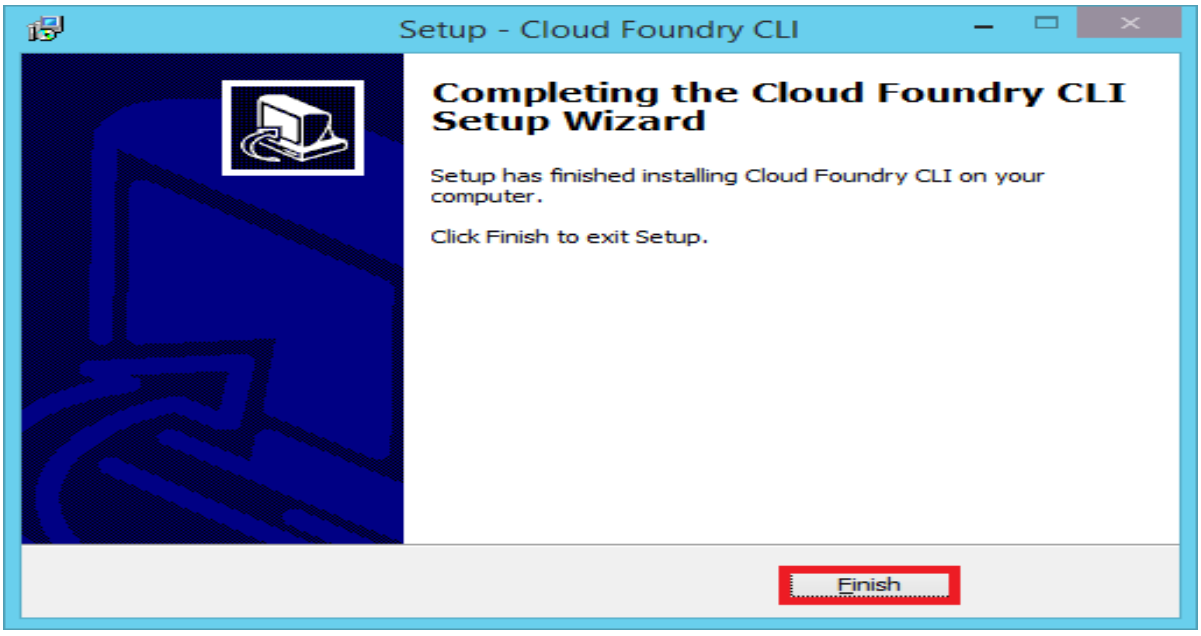
After downloading the executable file we need to run that file by double clicking on it. It will provide you an option to specify the path for the installation.



After selecting the proper installation path, click next and then it will provide you with the option whether to install the application or not. Click on **Install** button.



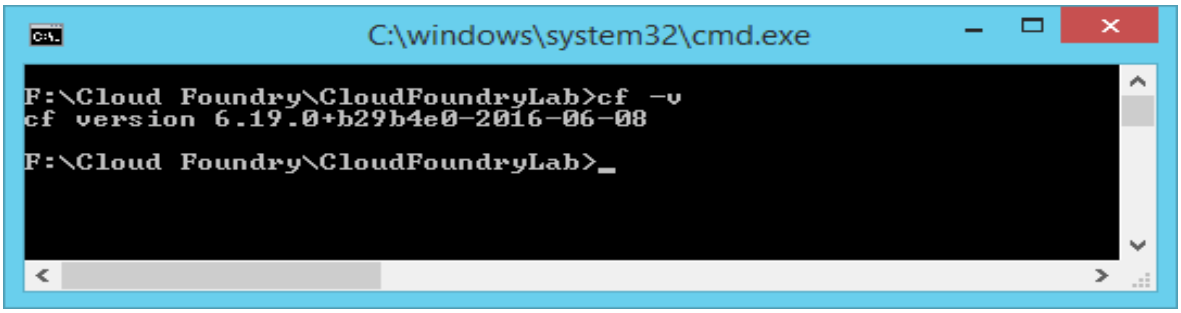
After completing the installation, it will provide you with a finish dialog box, then click on **Finish**.



Verifying the installation

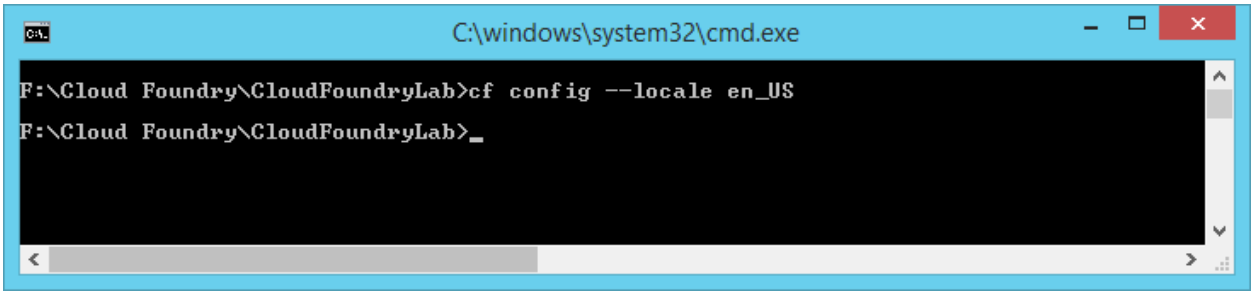
In order verify the completion of installation use the following command, which will show you the version of the Cloud Foundry command line interface.

->cf -v



We need to set our preferred language for the console. The cf CLI translates terminal output into the language that you select. The default language is **en-US**. We have several other optional languages to set. You can find them in https://docs.cloudfoundry.org/cf-cli/getting-started.html

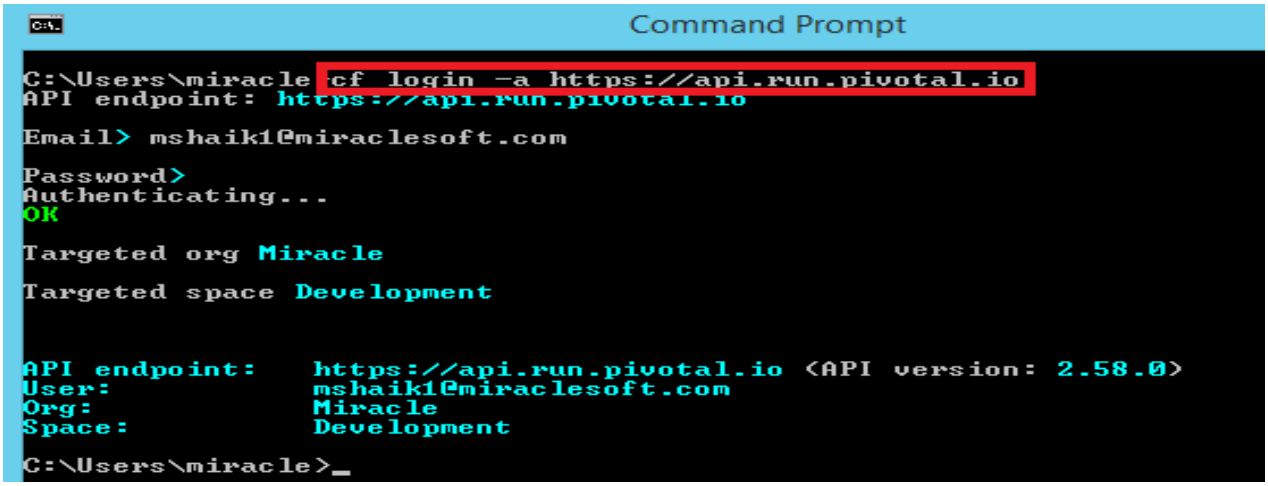
->cf config –locale your_language_code



Logging into Pivotal Cloud Foundry

In order to access the Cloud Foundry environment we need to provide API end point of our Cloud Foundry (https://api.run.pivotal.io) environment then it will prompt for username and password as follows. Generally it will save the credentials under the home directory in Linux environment or under user directory in windows with .cf folder and within that it will store the details in config.json file.

-> cf login -a api_url



Buildpacks

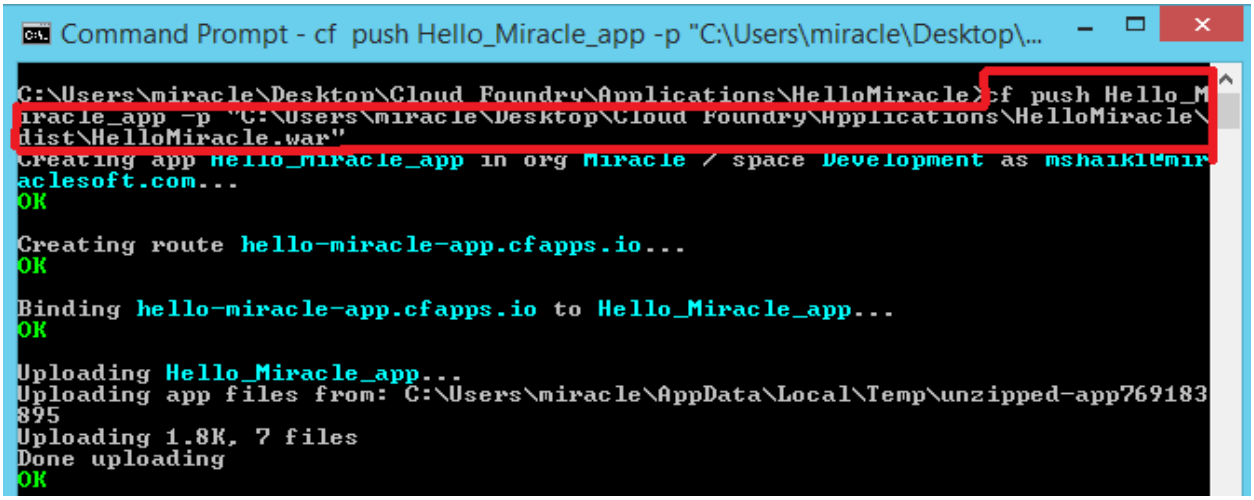
After deploying your application, we need runtime environments for running the application. Cloud Foundry provides these runtime environments with the help of buildpacks. There are several buildpacks like java, ruby, php, etc. Apart from the available buildpacks we can also provide our own custom buildpacks for configuring the runtime environments as per our requirements.

Deploying the application

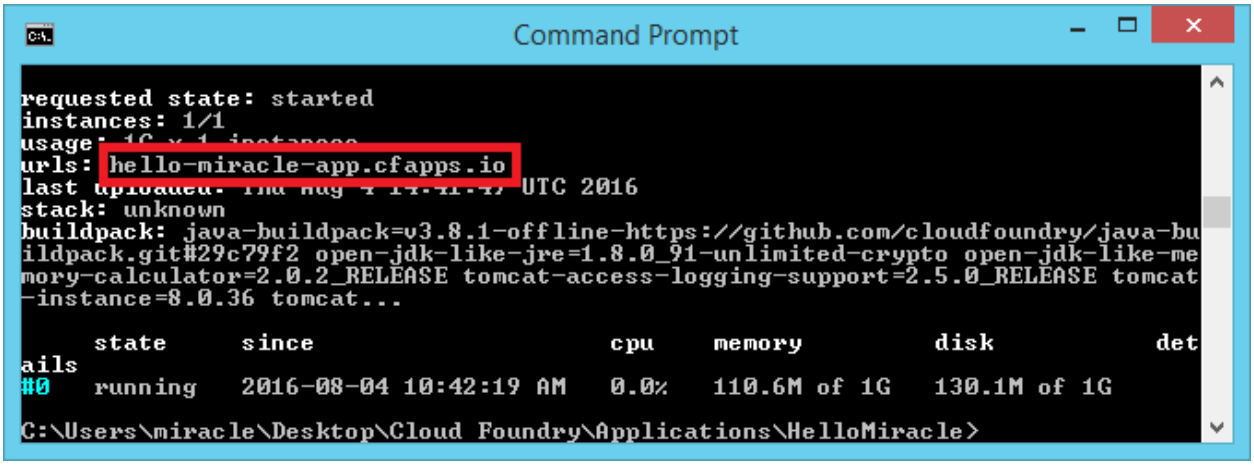
In order to deploy our application in Cloud Foundry, we need to push the application from our local repository to the Cloud Foundry. Before we push an application into Cloud Foundry we need to consider few things. In general we need to follow the buildpacks directory structure for our application otherwise our application cannot be recognized by Cloud Foundry. If we want to follow our own directory structure we need to provide a manifest file for pushing an application in which we need to specify the runtime or else we can push a war file by specifying its path.

©2016 Miracle Software Systems, Inc.

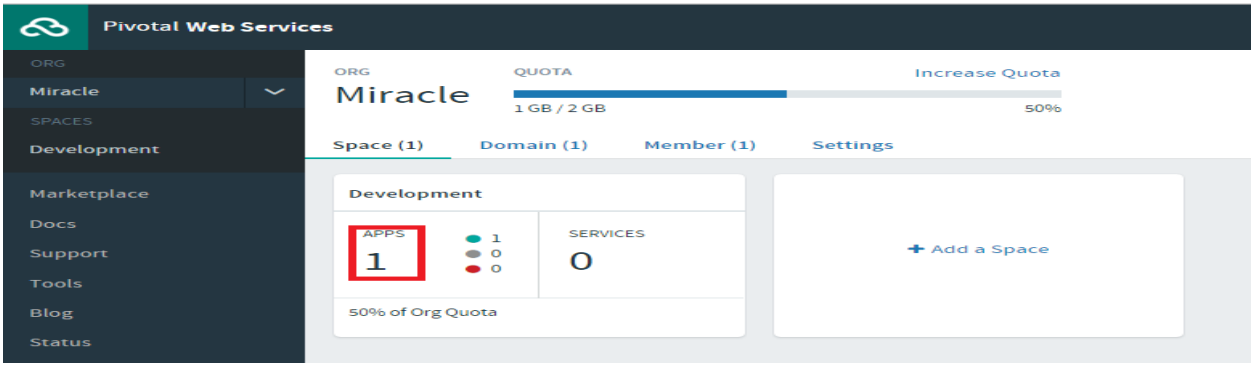
->cf push application_name -p path_to_warfile

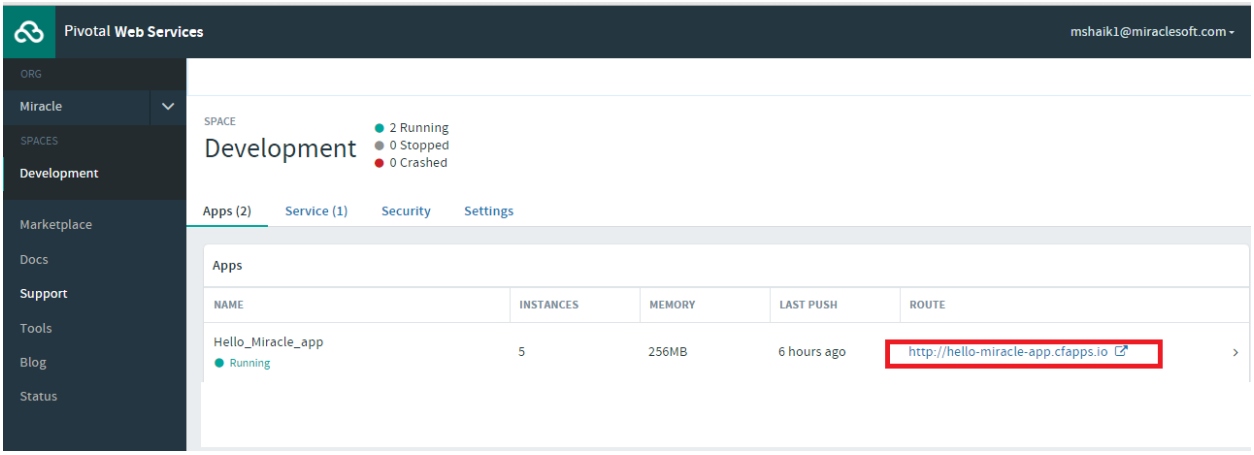


After Successfully deploying the application into Cloud Foundry, you will be provided with an URL for accessing the application along with few other details like number of instances, ram, disk utilization, etc.. as shown below.



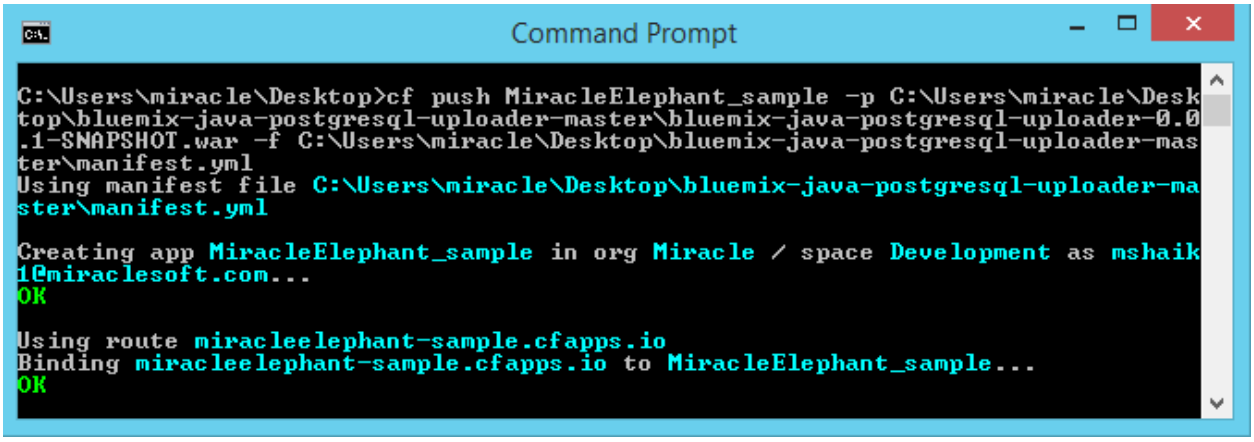
After successful deployment we can also observe our application from the web console.





The following is the output for the application through the web browser with the URL that we got after deploying the application.

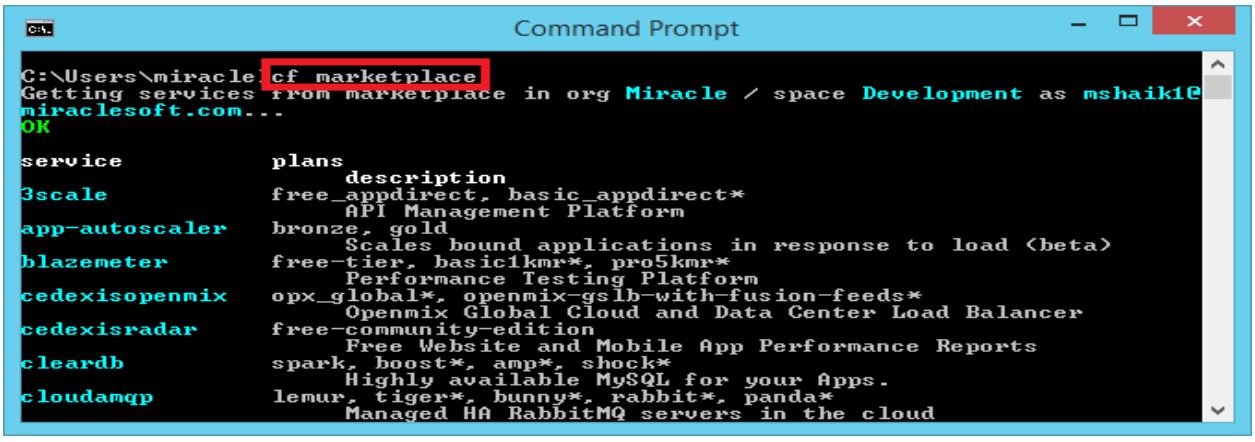
Push another application for binding the service



Services

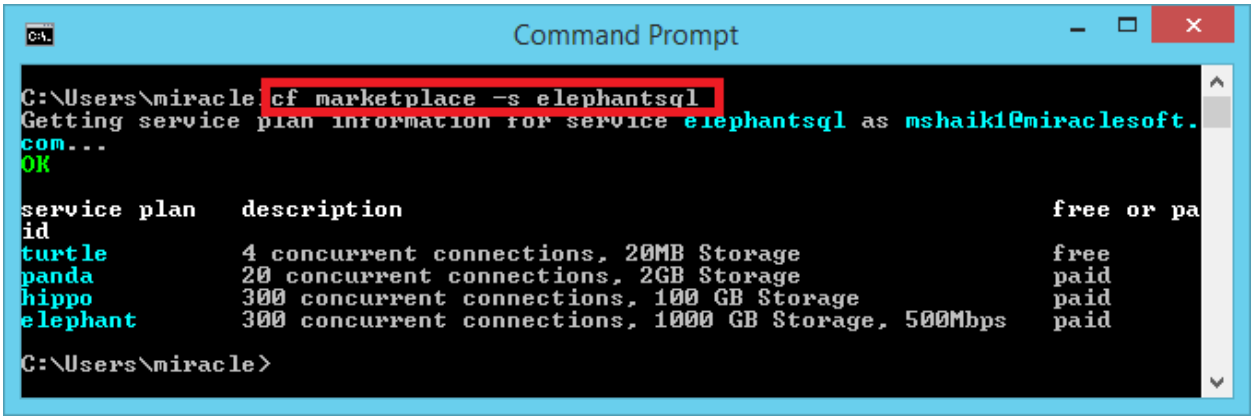
As Cloud Foundry is a PaaS service, it will not allow other services like database, analytics,etc.. to install in the same Infrastructure where our application is running. So, in order to consume those services we need to consume them by binding them to our application by providing appropriate details of the service. In pivotal, they are providing few services from the market place. We can see all the services using the following command.

->cf marketplace



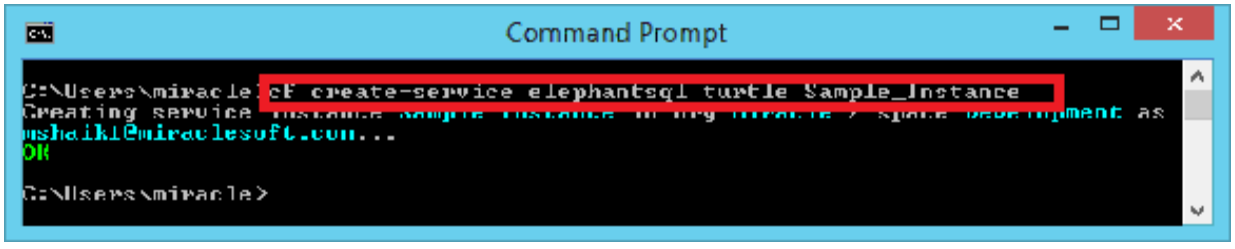
We can get the information about a particular service using the following command.

->cf marketplace -s service_name



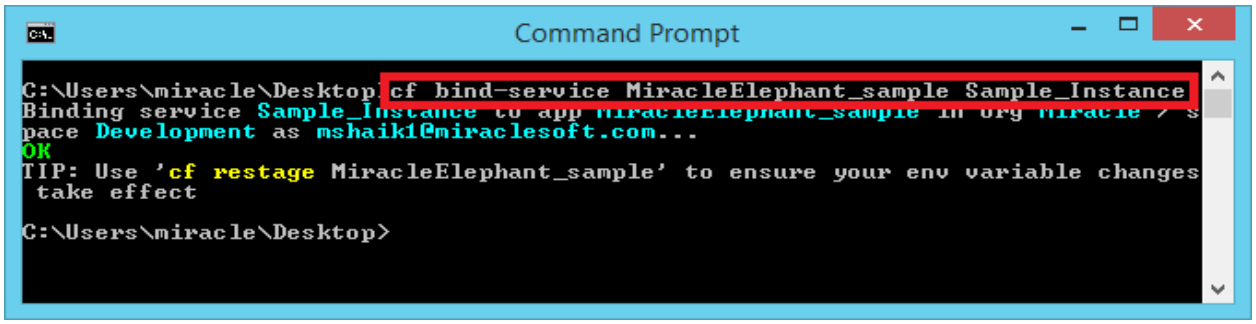
In order to consume a service we need to create it by which the service provider will create an instance of our service. For that we need to provide the appropriate service name which we would like to consume, service plan depending on our requirement and instance name with which name we would like to consume our service.

->cf create-service service_name service_plan instance_name



After creating the service we need to bind it to the application in which we need to consume our service.

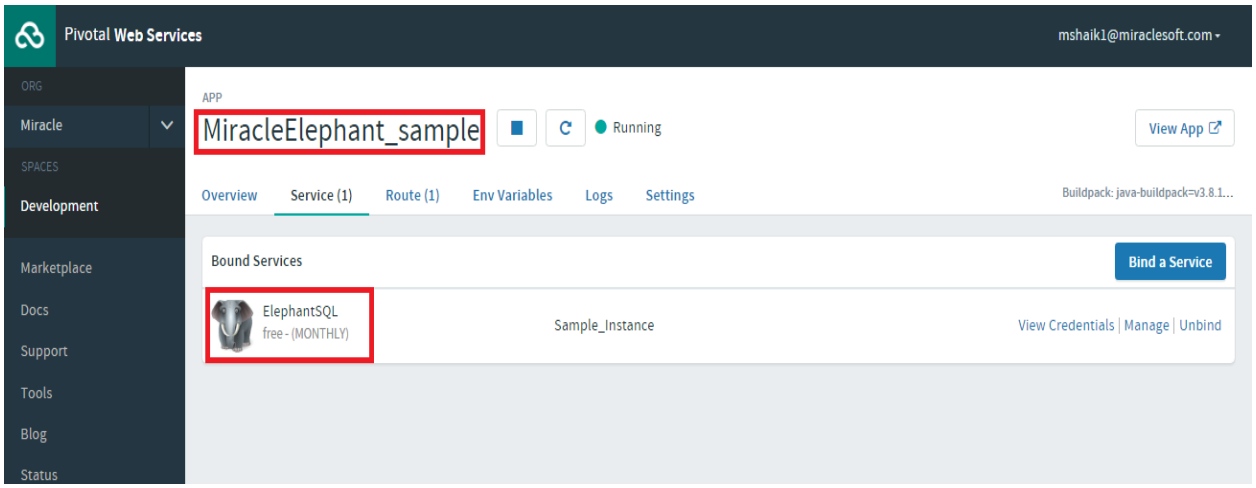
->cf bind-service application_name instance_name

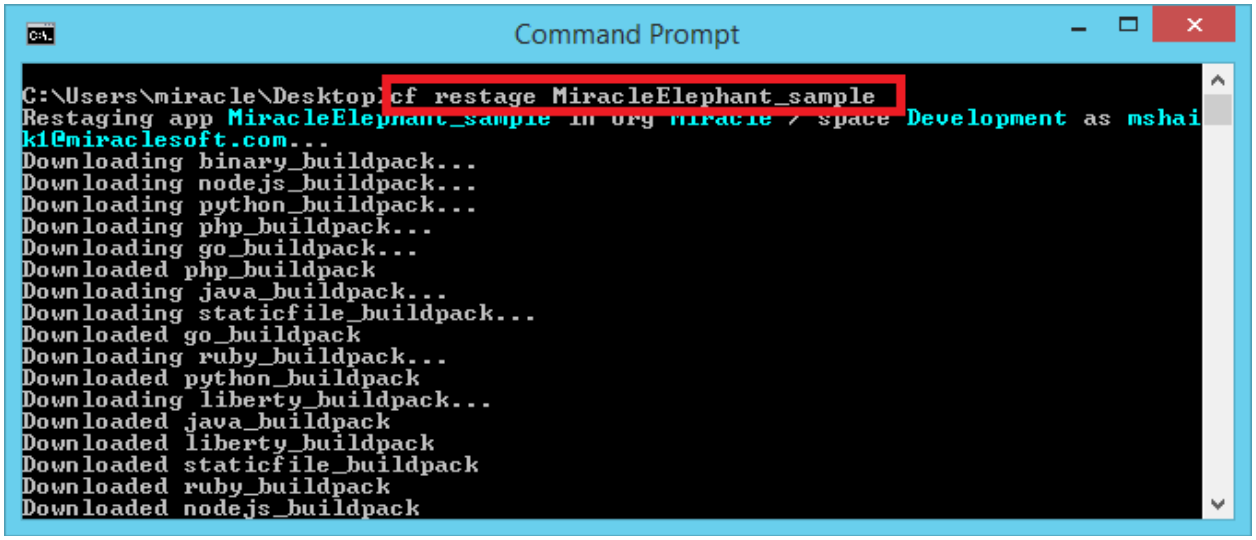


After binding the service to the application it will suggest us to restage the application to let the changes take place.

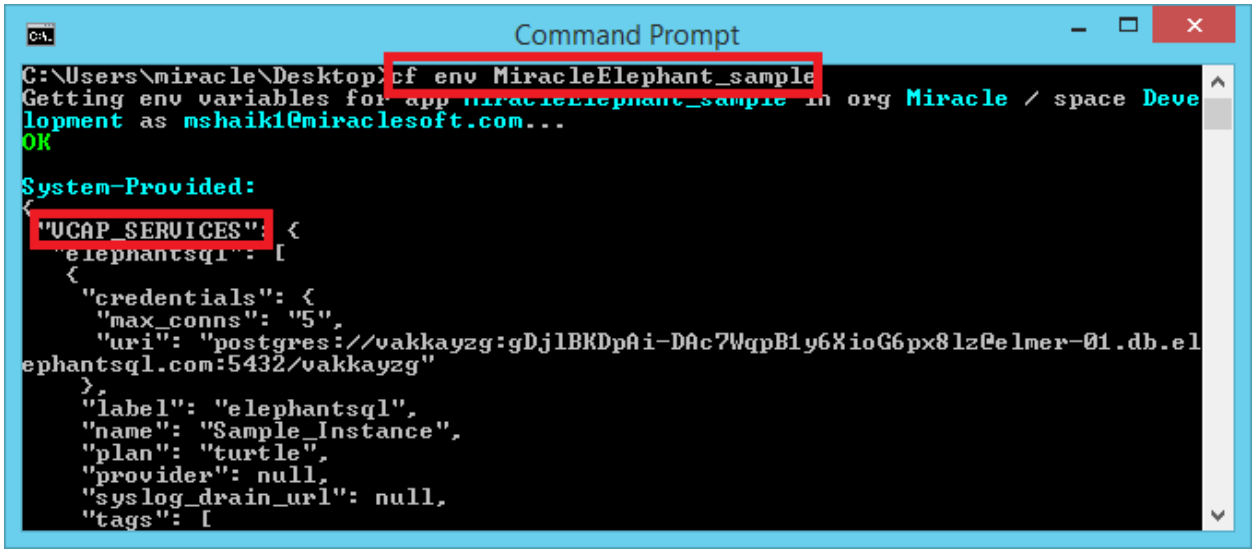
Difference between restage and restart

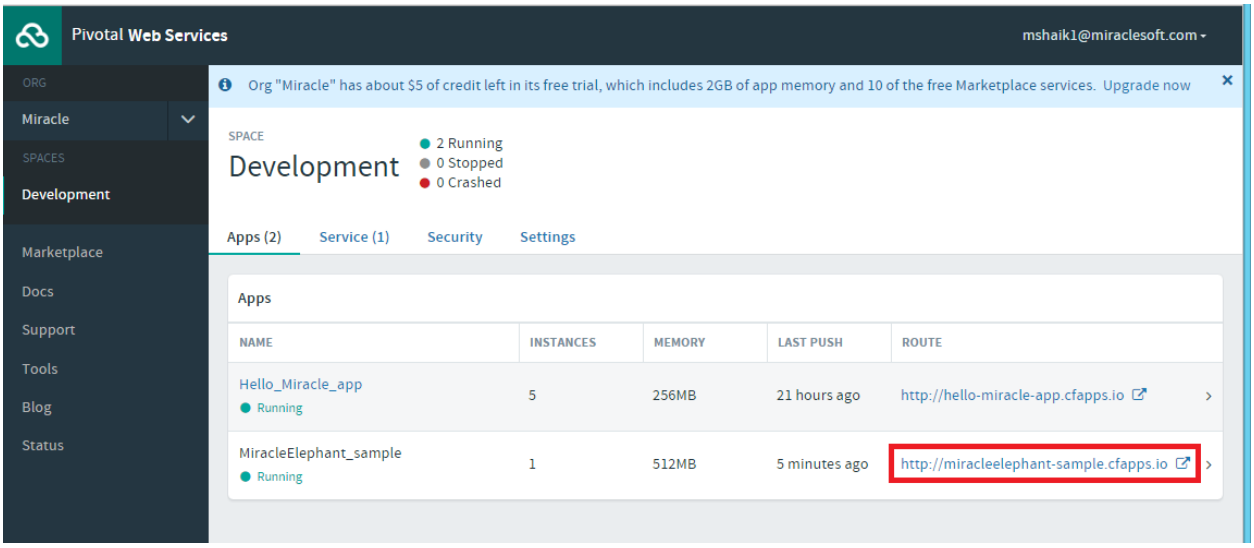
In restart, the application will be stopped and started again. On the other hand restage will create a new droplet and moves our environments into it from the existing droplet. By restaging the application the changes to the environment variables will take place. But in modern architecture the service details will be added to the environments without restaging with some services.



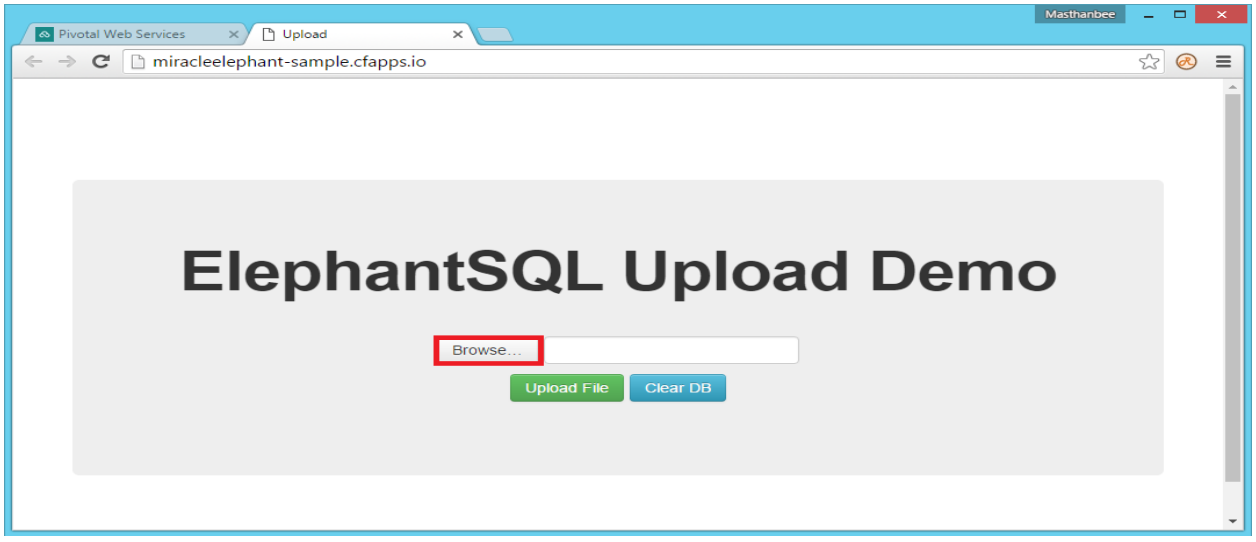


After binding the service and restaging the application, we can consume the service by accessing the details from the environment variables.

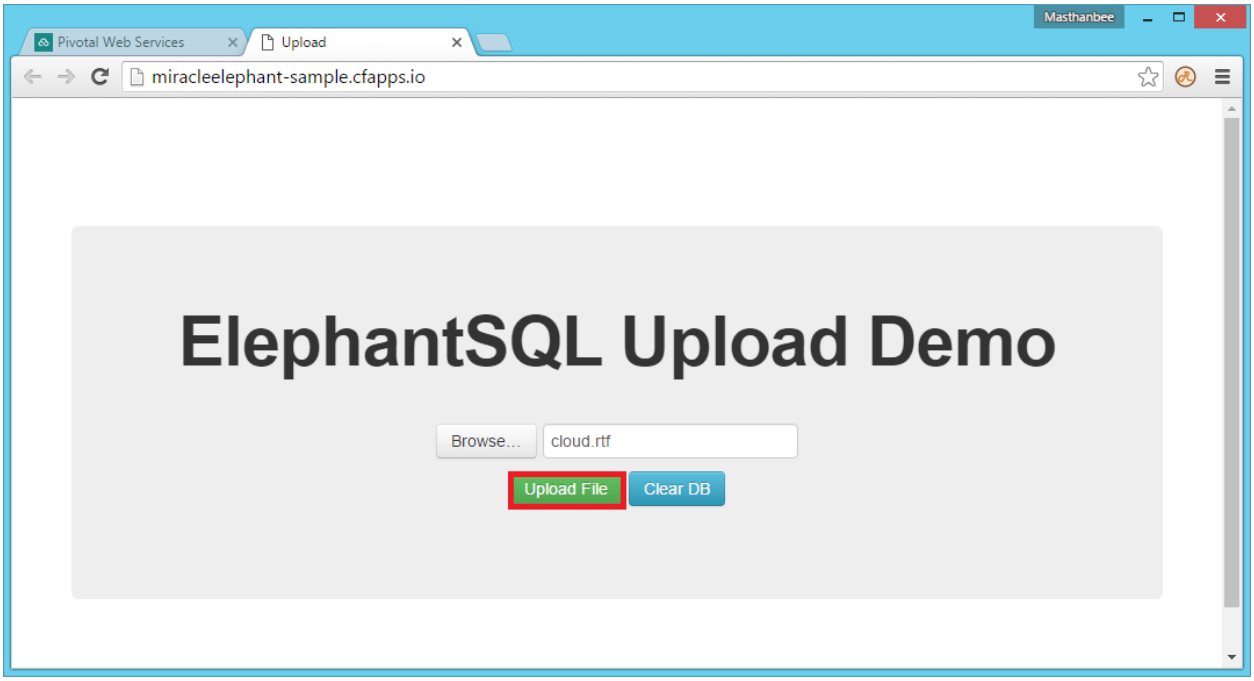




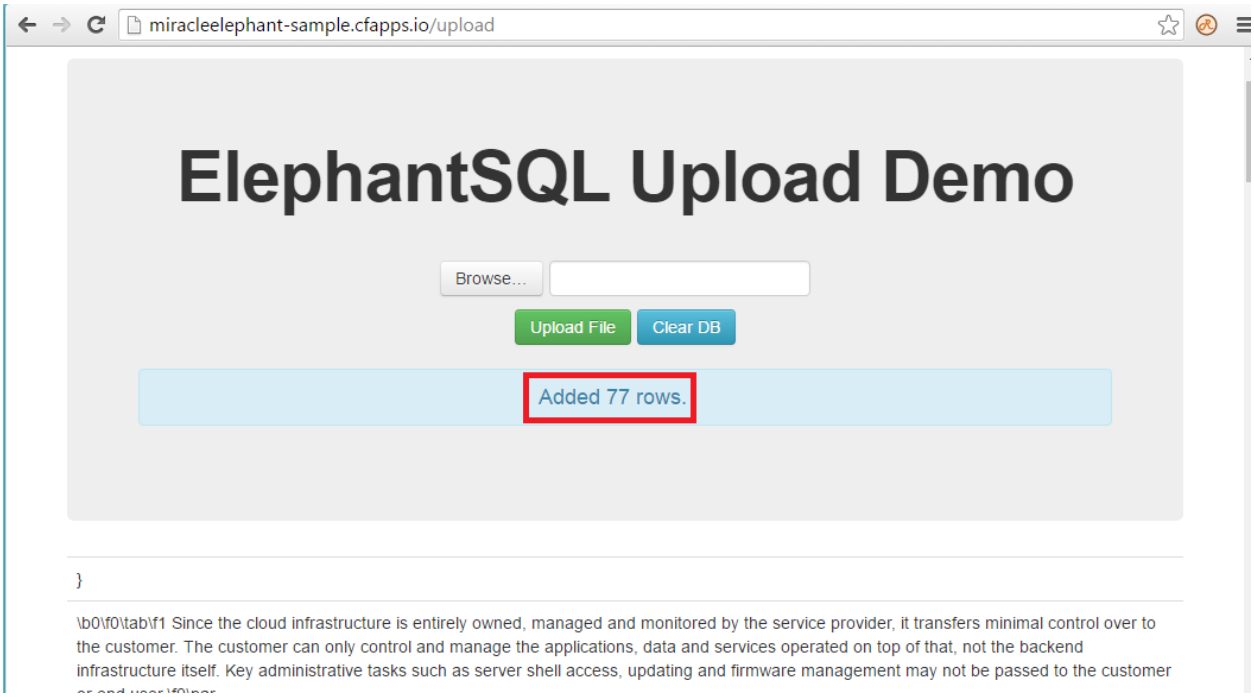
Browse the path of the file which you would like to upload into the database



After browsing the application please click on the **Upload File** button.

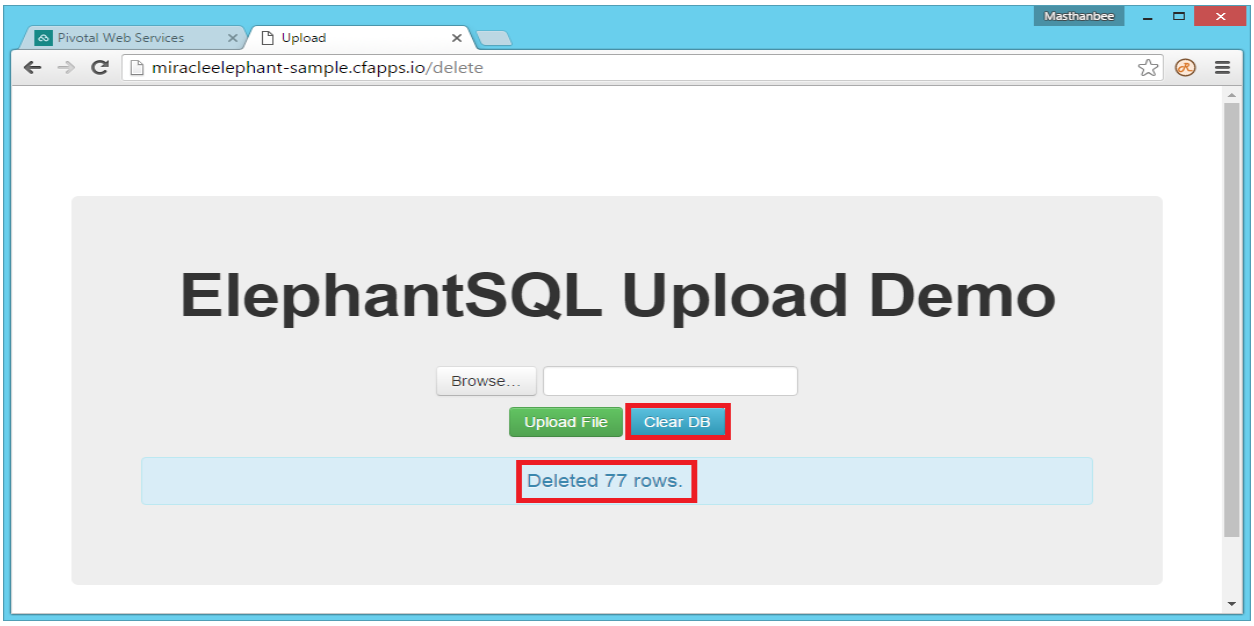


After clicking on the upload file button the data will be inserted into the database and the output can be observed as shown below.



©2016 Miracle Software Systems, Inc.

We can clear the existing database by clicking on the **Clear DB** button.



Scaling

Horizontal Scaling

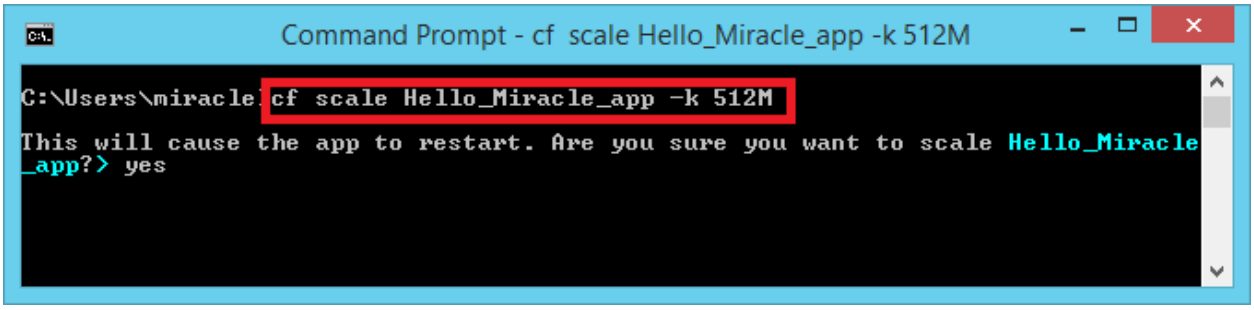
Horizontally scaling an application creates or destroys instances of your application. Incoming requests to your application are automatically load balanced across all instances of your application, and each instance handles tasks in parallel with every other instance. Adding more instances allows your application to handle increased traffic and demand.

->cf scale application_name -i instances_count

Vertical Scaling

Vertically scaling an application changes the disk space limit or memory limit that Cloud Foundry applies to all instances of the application.

->cf scale application_name -k disk_size

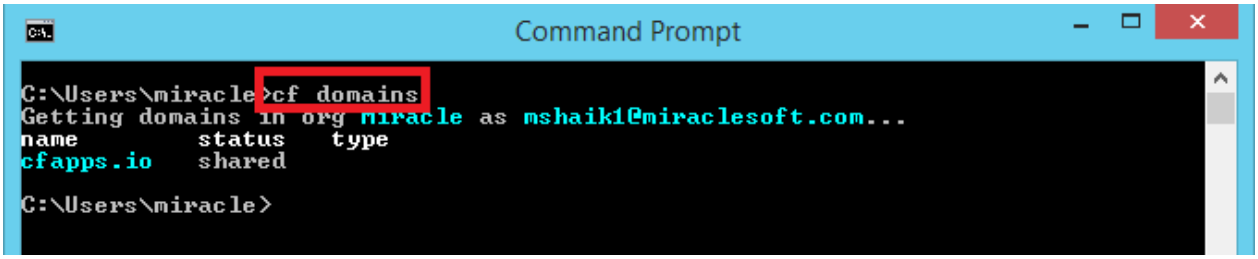


->cf scale application_name ram_size

Domains

In order to identify our application in a network, domains names are a source, which is a collection of one or more IP Addresses which uniquely identify our system in a network. To list all the available domains in an organization enter the following command.

->cf domains

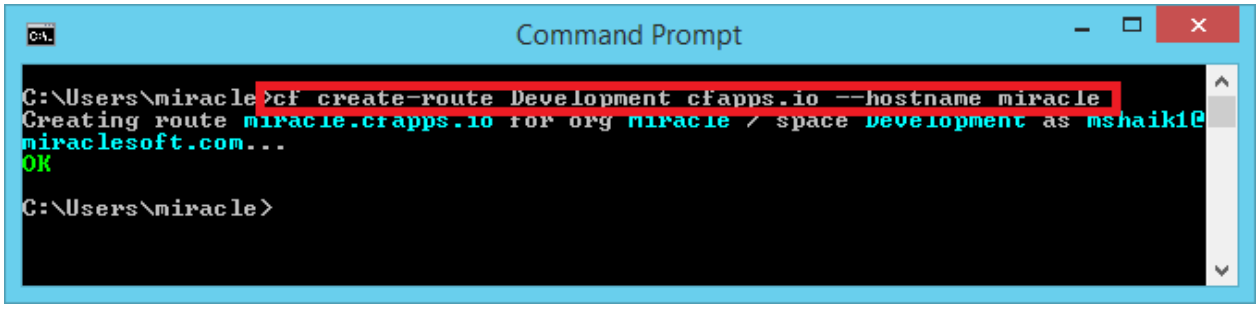


In general our applications will get a URL with the application name followed by domain name. For example if our application name is miracle and the domain name is cfapps.io then our URL will be miracle.cfapps.io

Routes

If we want our application to be accessible with more than one URL then we need to create a route and map the route to the respective application.
creating a route for the application

->cf create-route space_name domain_name –hostname name_of_the_host

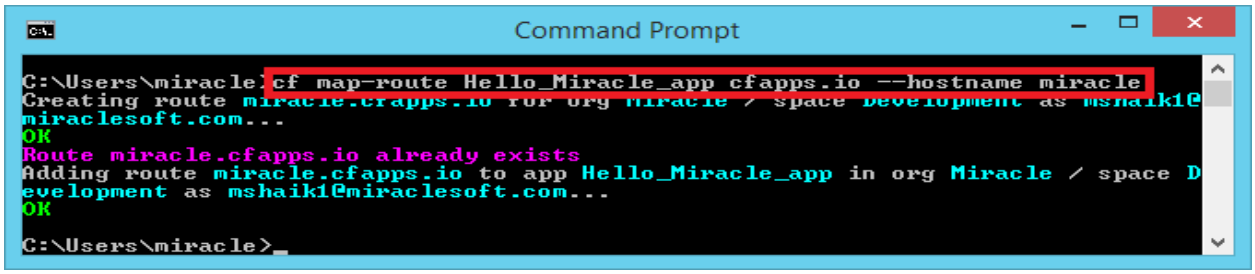


You can verify whether the route has been created or not using the command

->cf check-route hostname domain_name

Even though we have created a route no traffic will be diverted to the application until we map the route to the application.

->cf map-route application_name domain_name [--hostname]



You can view the list of all the routes that are mapped to the application using the command

->cf routes