

## Cloud Foundry

## 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

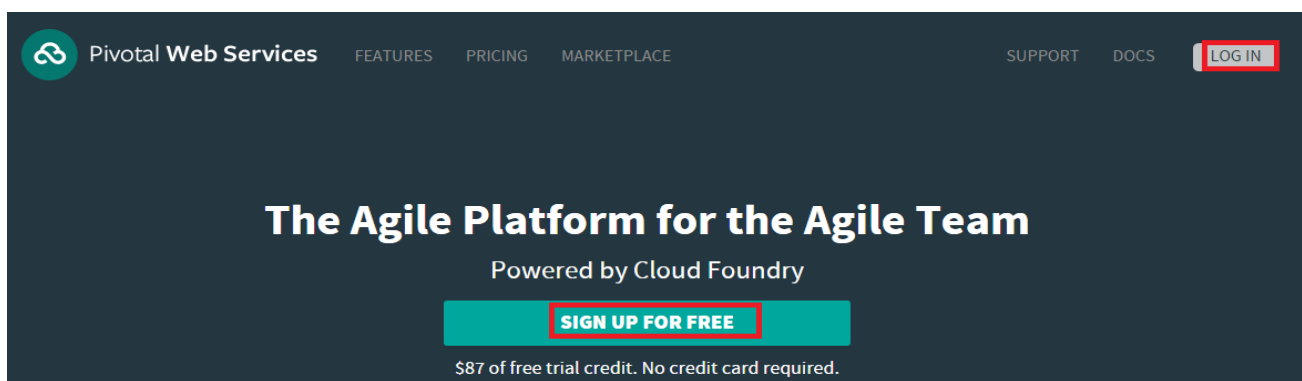
## Introduction

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.

Pivotal Web Services Masthu816@gmail.com

1 SIGN UP 2 CLAIM YOUR TRIAL 3 CREATE ORG

### Sign Up for your free trial

Welcome to Pivotal Web Services! Complete these steps to access your account.

Username  
masthu816@gmail.com

☒ Keep me up to date about Pivotal Web Services.  
☒ Send me occasional news about related Pivotal products

☐ I have read and agree to the [Terms of Service](#) for Pivotal Web Services

**Next: Claim Your Trial**

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.

Pivotal Web Services mshaik1@miraclesoft.com

ORG: Miracle

QUOTA: 1.5 GB / 2 GB 75% Increase Quota

Space (1) Domains (2) Member (1) Settings

Development

APPS: 2 SERVICES: 1

75% of Org Quota

+ Add a Space

ORG: Miracle

MSS

**Create a New Org**

Docs Support Tools Blog Status

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**.

Pivotal Web Services mshalk1@miraclesoft.com

ORG  
Miracle

SPACES  
Development

Marketplace

Docs

Support

Tools

Blog

Status

## Create an Org

Org (or Project) Name

MSSLabs

**Create Org**

Organization (org) is a development account that encompasses computing resources, apps, and services. It can be owned and used by an individual or by multiple collaborators.

Set the org name to be the name of the project you will be working on or the name of your team. Do not worry - you can change this name at any time!

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

Pivotal Web Services

ORG  
MSSLabs

SPACES  
development

Marketplace

Docs

Support

Tools

Blog

Status

ORG: MSSLabs

QUOTA: 0 MB / 0 MB

Increase Quota

NaN%

Space (1) Domain (1) Member (1) Settings

| development |          |
|-------------|----------|
| APPS        | SERVICES |
| 0           | 0        |

100% of Org Quota

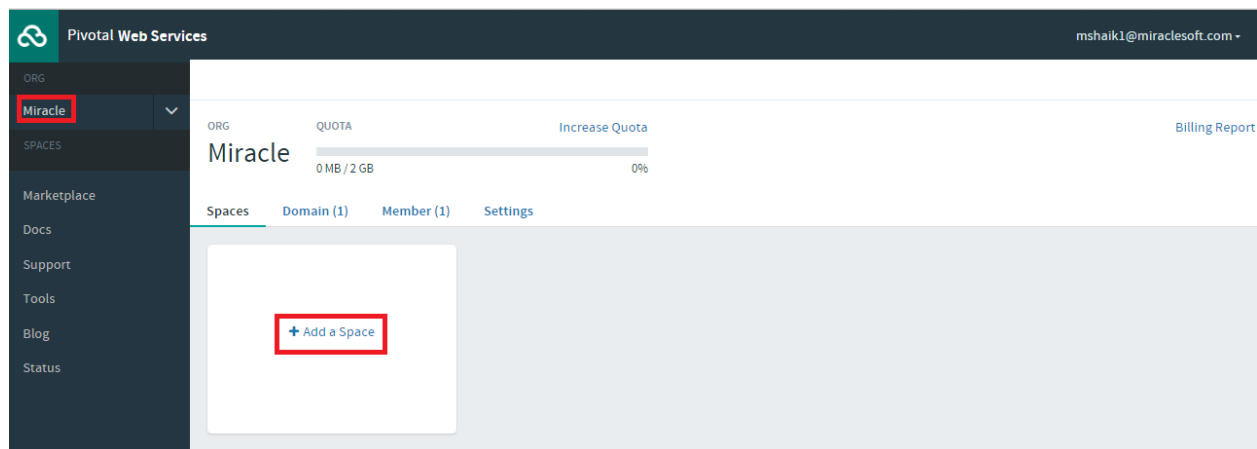
+ Add a Space

## Spaces

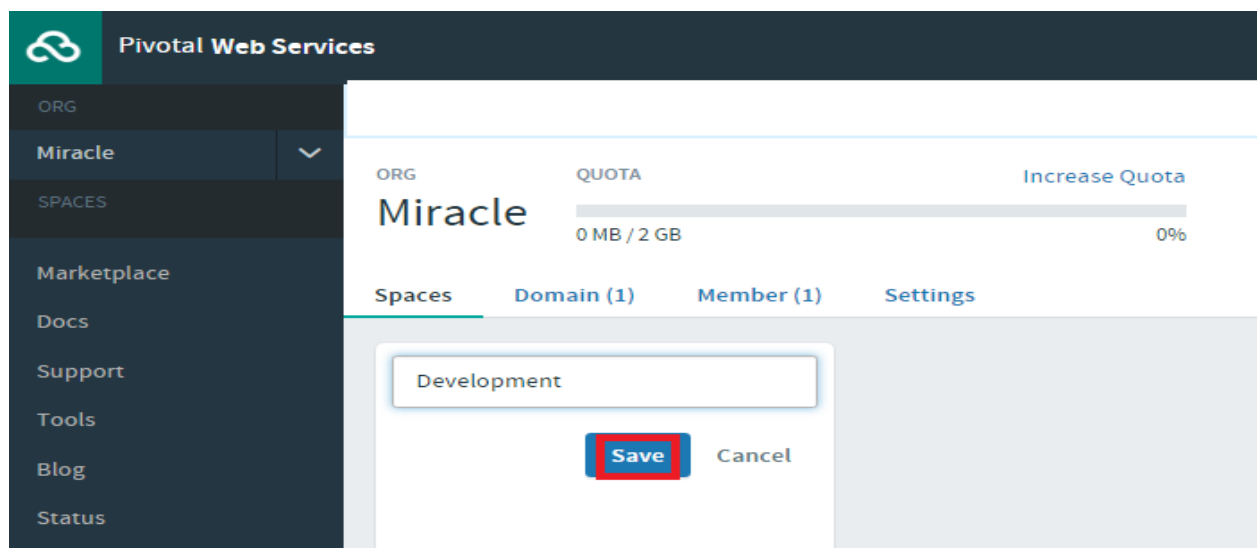
Every application and service is scoped to a space. Each organization contains at least one space. A space provides users with access to a shared location for application development, deployment, and maintenance. Each space role applies only to a particular space.

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.

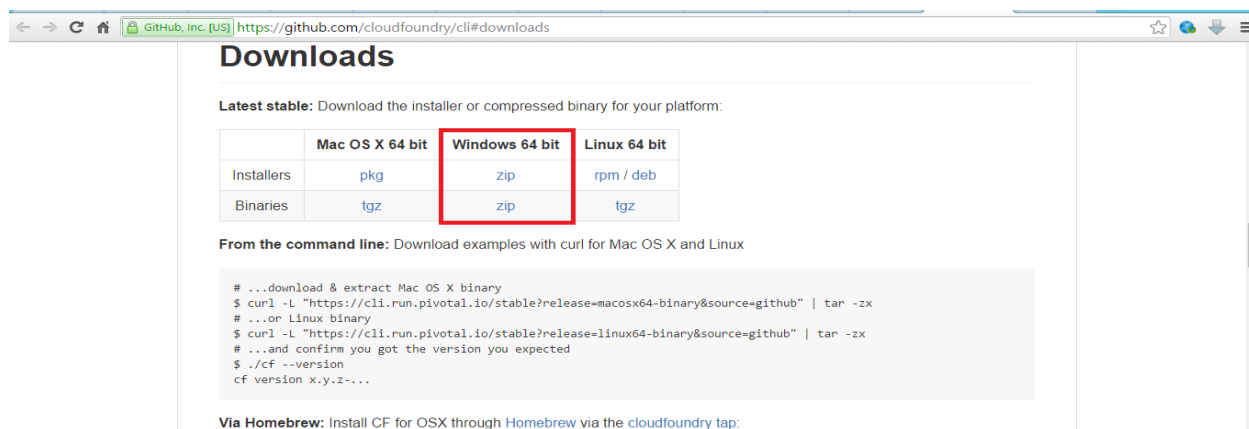


Under the organization that you have created create a space and add users to the space.

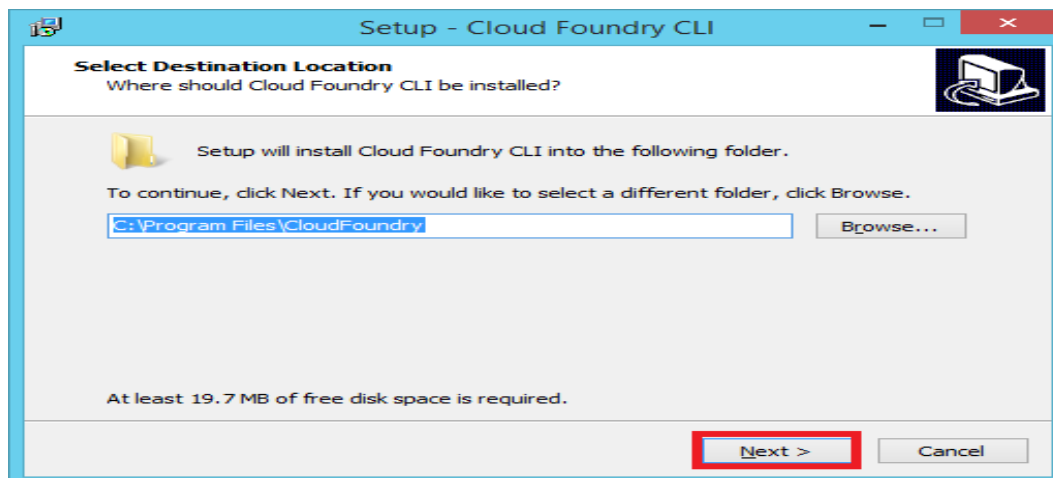
## Installing Cloud Foundry Command Line Interface

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.

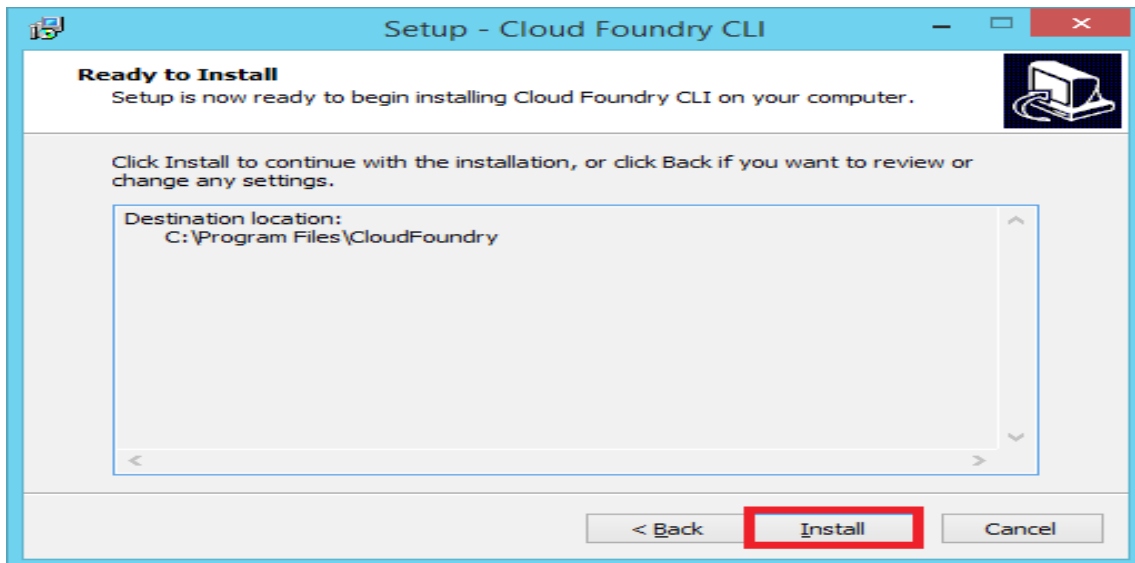
<https://github.com/cloudfoundry/cli#downloads>



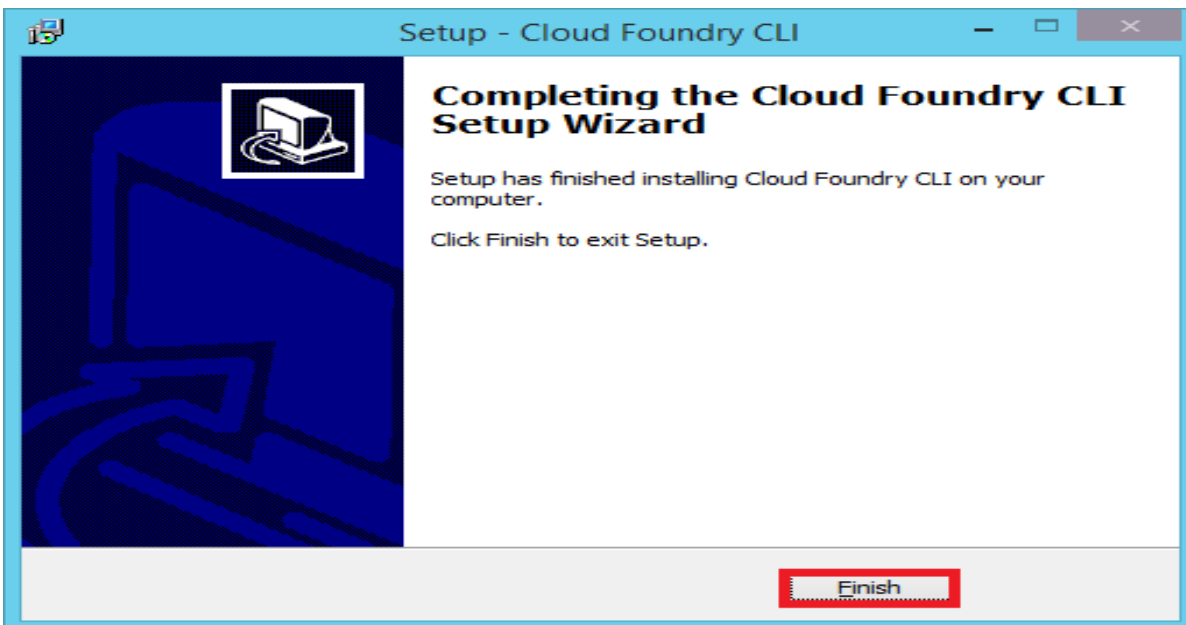
After downloading the executable file we need to run that file by double clicking on it. It will provide you an option to specify the path for the installation.



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.



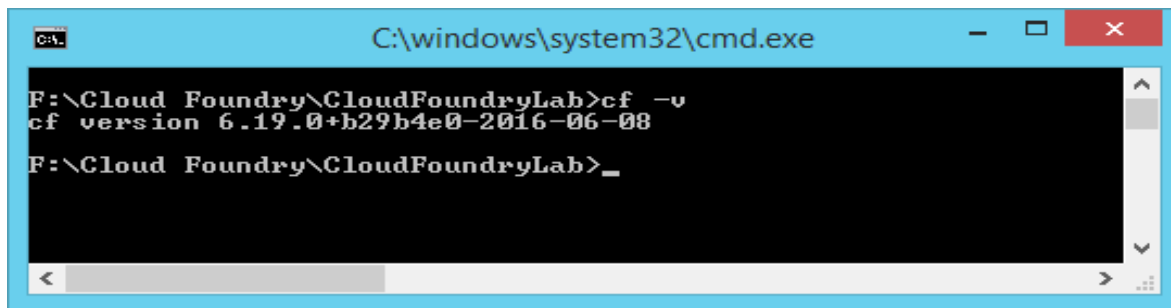
After completing the installation, it will provide you with a finish dialog box, then click on **Finish**.



## Verifying the installation

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

->cf -v



```
C:\windows\system32\cmd.exe
F:\Cloud Foundry\CloudFoundryLab>cf -v
cf version 6.19.0+b29b4e0-2016-06-08
F:\Cloud Foundry\CloudFoundryLab>_
```

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



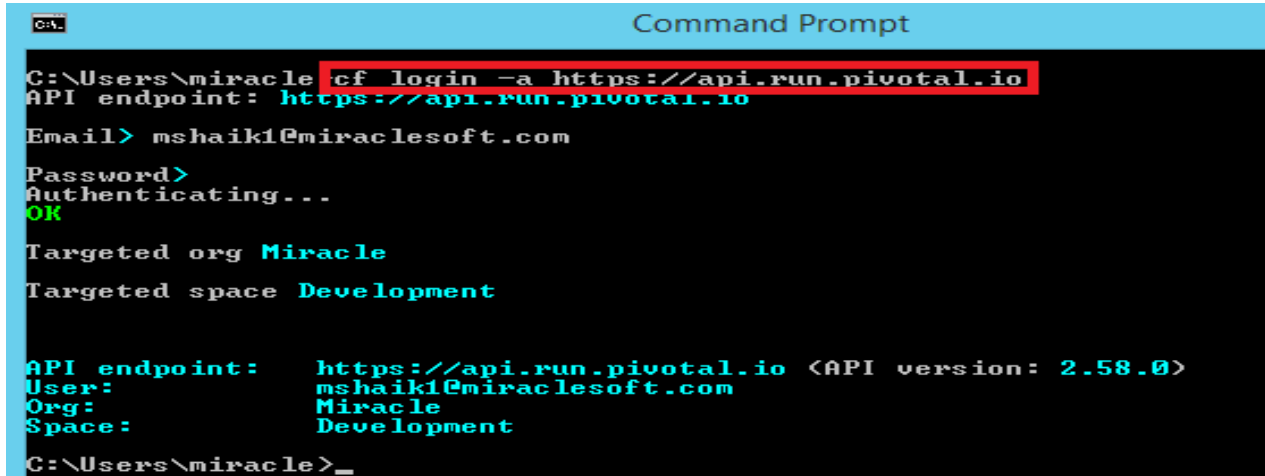
```
C:\windows\system32\cmd.exe
F:\Cloud Foundry\CloudFoundryLab>cf config --locale en_US
F:\Cloud Foundry\CloudFoundryLab>_
```

## 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



```
C:\Users\miracle> cf login -a https://api.run.pivotal.io
API endpoint: https://api.run.pivotal.io
Email> mshaik1@miraclesoft.com
Password>
Authenticating...
OK
Targeted org Miracle
Targeted space Development

API endpoint: https://api.run.pivotal.io <API version: 2.58.0>
User: mshaik1@miraclesoft.com
Org: Miracle
Space: Development
C:\Users\miracle>
```

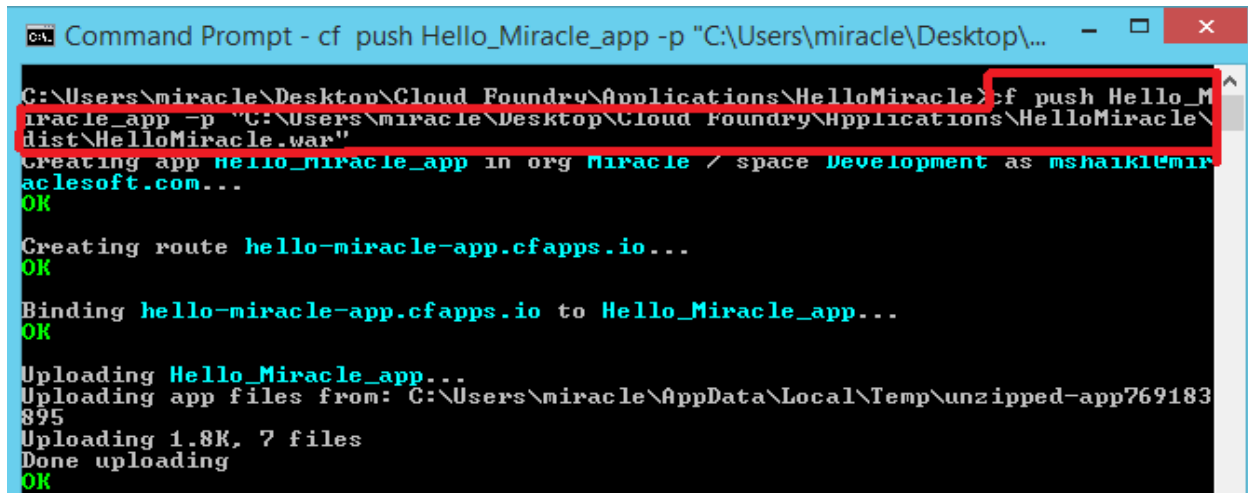
## Buildpacks

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

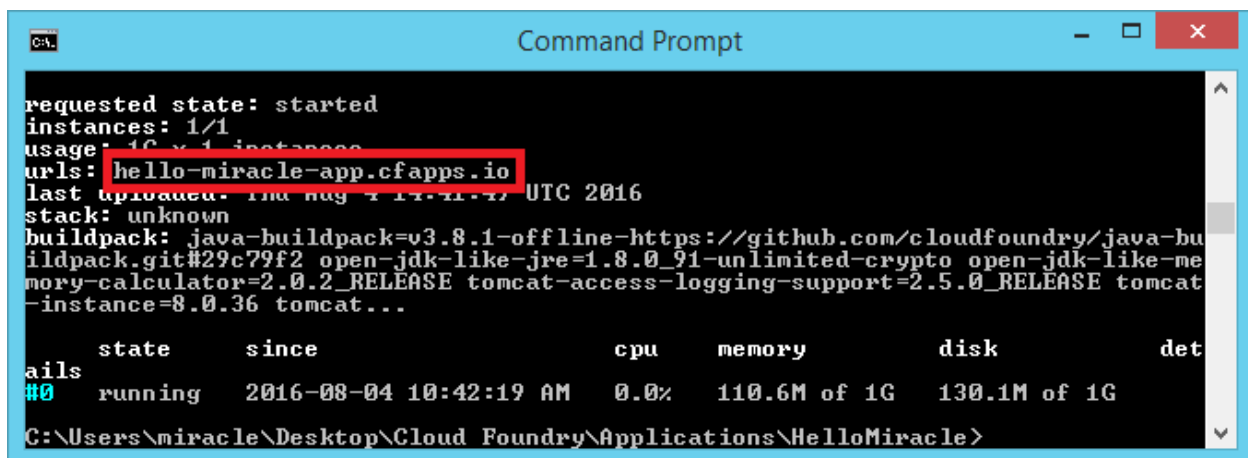
In order to deploy our application in Cloud Foundry, we need to push the application from our local repository to the Cloud Foundry. Before we push an application into Cloud Foundry we need to consider few things. In general we need to follow the buildpacks directory structure for our application otherwise 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.

->cf push application\_name -p path\_to\_warfile



```
C:\Users\miracle\Desktop\Cloud Foundry\Applications\HelloMiracle>cf push Hello_Miracle_app -p "C:\Users\miracle\Desktop\Cloud Foundry\Applications\HelloMiracle\dist\HelloMiracle.war"
Creating app hello-miracle_app in org Miracle / space Development as mshaikl@miraclesoft.com...
OK
Creating route hello-miracle-app.cfapps.io...
OK
Binding hello-miracle-app.cfapps.io to Hello_Miracle_app...
OK
Uploading Hello_Miracle_app...
Uploading app files from: C:\Users\miracle\AppData\Local\Temp\unzipped-app769183895
Uploading 1.8K, 7 files
Done uploading
OK
```

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.

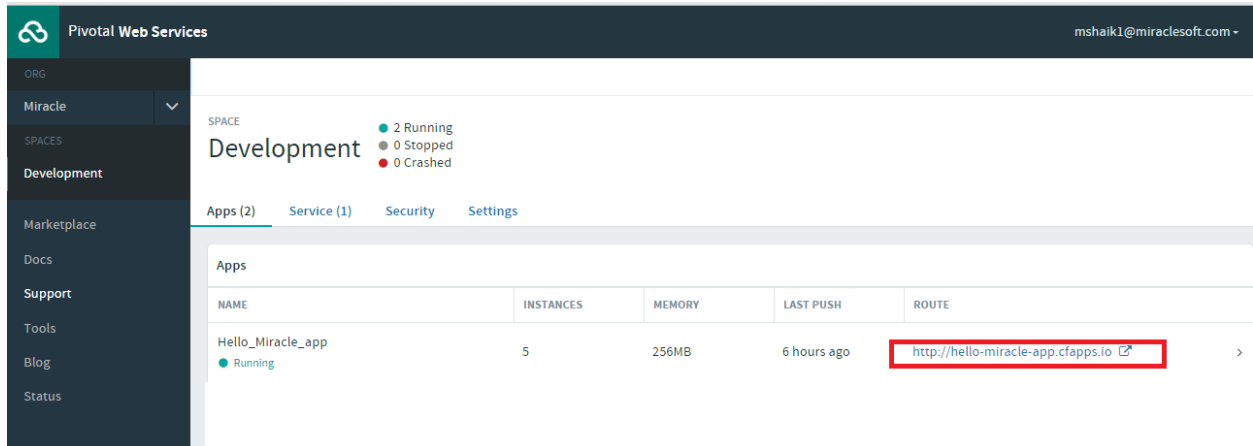
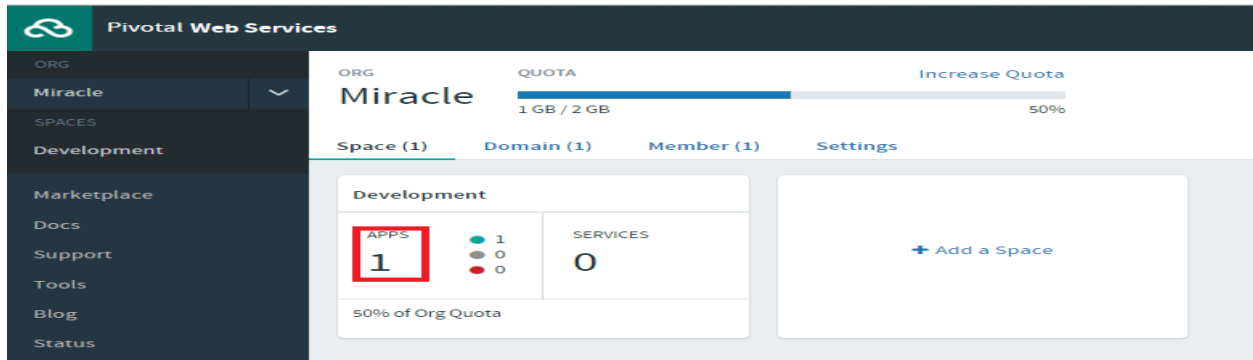


```
requested state: started
instances: 1/1
usage: 1G x 1 instances
urls: hello-miracle-app.cfapps.io
last updated: Thu Aug 11 11:11:17 UTC 2016
stack: unknown
buildpack: java-buildpack=v3.8.1-offline-https://github.com/cloudfoundry/java-buildpack.git#29c79f2 open-jdk-like-jre=1.8.0_91-unlimited-crypto open-jdk-like-memory-calculator=2.0.2_RELEASE tomcat-access-logging-support=2.5.0_RELEASE tomcat-instance=8.0.36 tomcat...

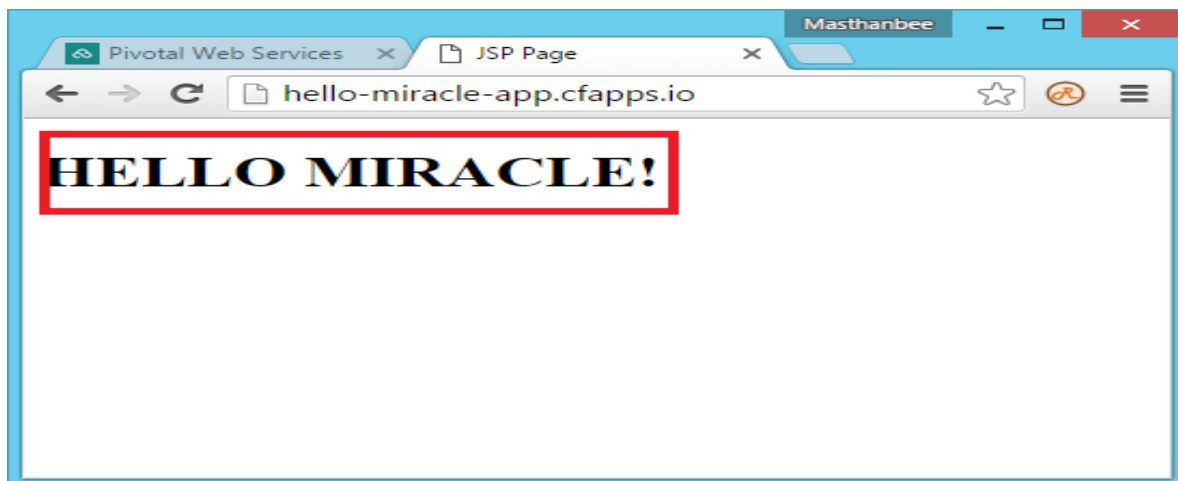
state      since                cpu    memory              disk              det
ails
#0 running  2016-08-04 10:42:19 AM  0.0%   110.6M of 1G       130.1M of 1G

C:\Users\miracle\Desktop\Cloud Foundry\Applications\HelloMiracle>
```

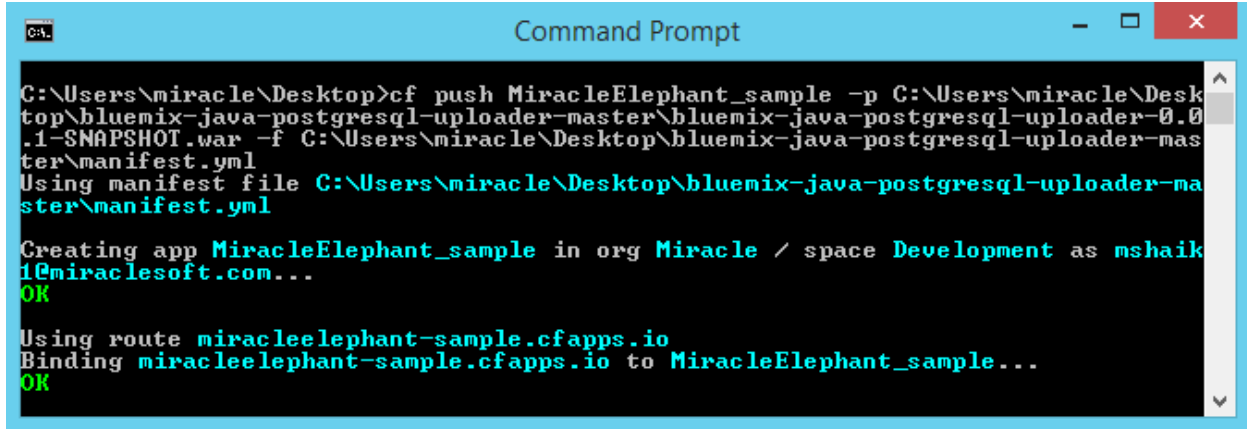
After successful deployment we can also observe our application from the web console.



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



```
C:\Users\miracle\Desktop>cf push MiracleElephant_sample -p C:\Users\miracle\Desktop\bluemix-java-postgresql-uploader-master\bluemix-java-postgresql-uploader-0.0.1-SNAPSHOT.war -f C:\Users\miracle\Desktop\bluemix-java-postgresql-uploader-master\manifest.yml
Using manifest file C:\Users\miracle\Desktop\bluemix-java-postgresql-uploader-master\manifest.yml

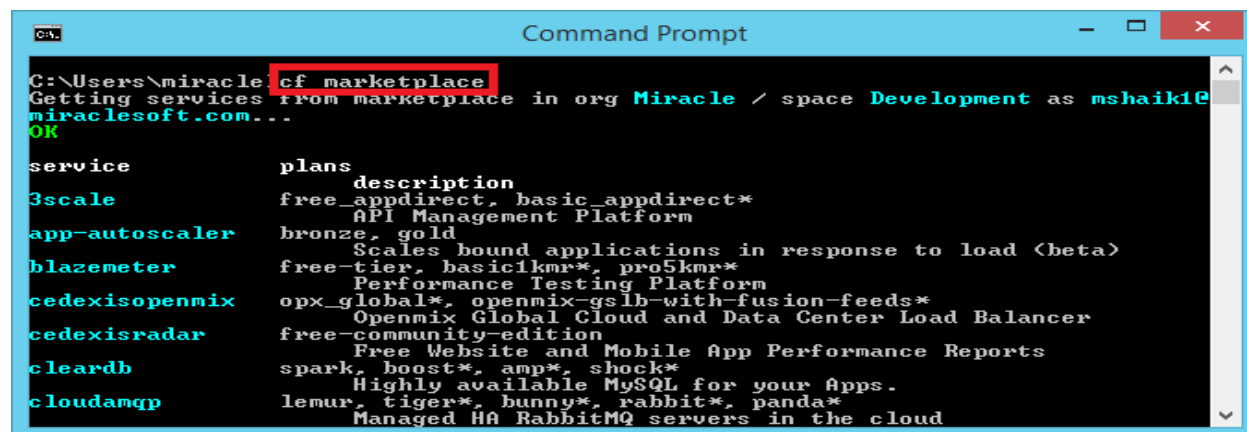
Creating app MiracleElephant_sample in org Miracle / space Development as mshaik1@miraclesoft.com...
OK

Using route miracleelephant-sample.cfapps.io
Binding miracleelephant-sample.cfapps.io to MiracleElephant_sample...
OK
```

## 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

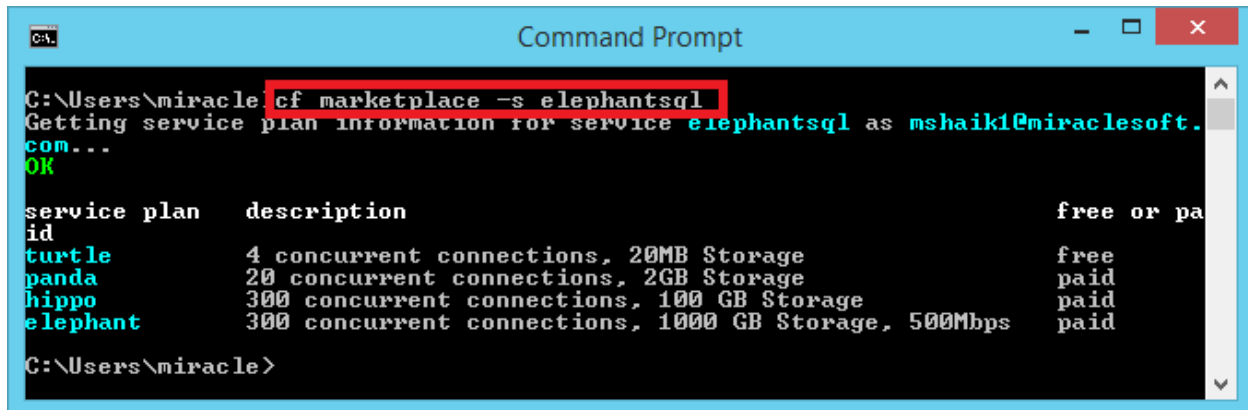


```
C:\Users\miracle>cf marketplace
Getting services from marketplace in org Miracle / space Development as mshaik1@miraclesoft.com...
OK

service      plans      description
3scale        free_appdirect, basic_appdirect* API Management Platform
app-autoscaler bronze, gold Scales bound applications in response to load (beta)
blazemeter    free-tier, basic1krr*, pro5krr* Performance Testing Platform
cedexisopenmix opx_global*, openmix-gslb-with-fusion-feeds* Openmix Global Cloud and Data Center Load Balancer
cedexisradar  free-community-edition Free Website and Mobile App Performance Reports
cleardb       spark, boost*, amp*, shock* Highly available MySQL for your Apps.
cloudamqp     lemur, tiger*, bunny*, rabbit*, panda* Managed HA RabbitMQ servers in the cloud
```

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

->cf marketplace -s service\_name



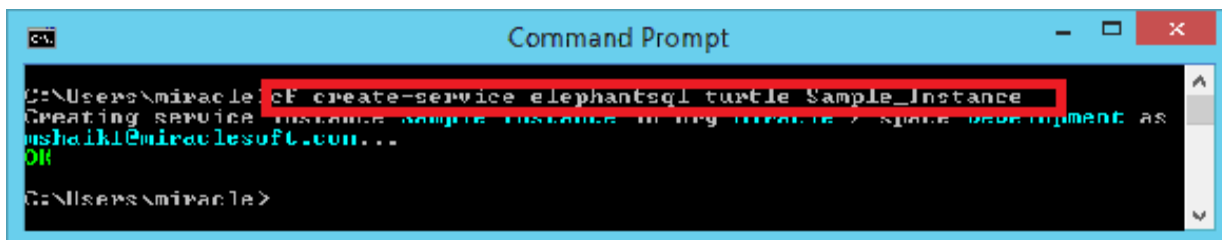
```
C:\Users\miracle>cf marketplace -s elephantsql
Getting service plan information for service elephantsql as mshaik1@miraclesoft.com...
OK

service plan  description                                     free or pa
id
turtle        4 concurrent connections, 20MB Storage                    free
panda         20 concurrent connections, 2GB Storage                     paid
hippo         300 concurrent connections, 100 GB Storage                  paid
elephant      300 concurrent connections, 1000 GB Storage, 500Mbps       paid

C:\Users\miracle>
```

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

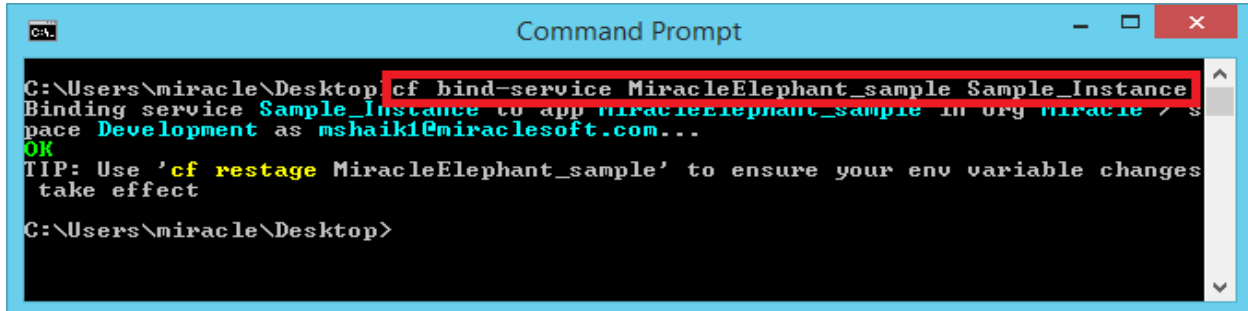


```
C:\Users\miracle>cf create-service elephantsql turtle Sample_Instance
Creating service instance Sample_Instance in my miracle / space deployment as mshaik1@miraclesoft.com...
OK

C:\Users\miracle>
```

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



```
Command Prompt

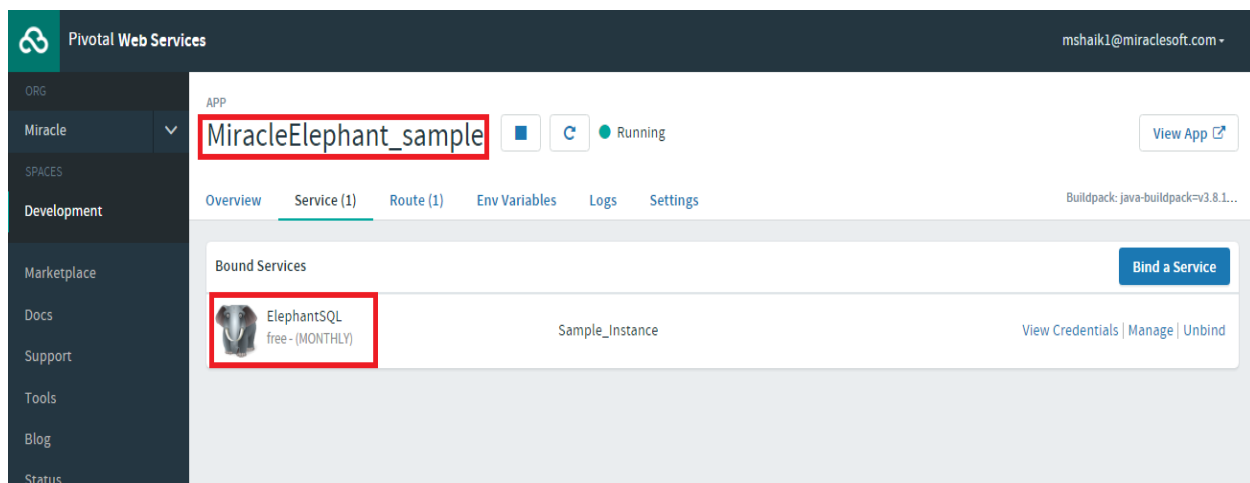
C:\Users\miracle\Desktop>cf bind-service MiracleElephant_sample Sample_Instance
Binding service Sample_Instance to app miracleElephant_sample in org miracle / s
pace Development as mshaik1@miraclesoft.com...
OK
TIP: Use 'cf restage MiracleElephant_sample' to ensure your env variable changes
take effect

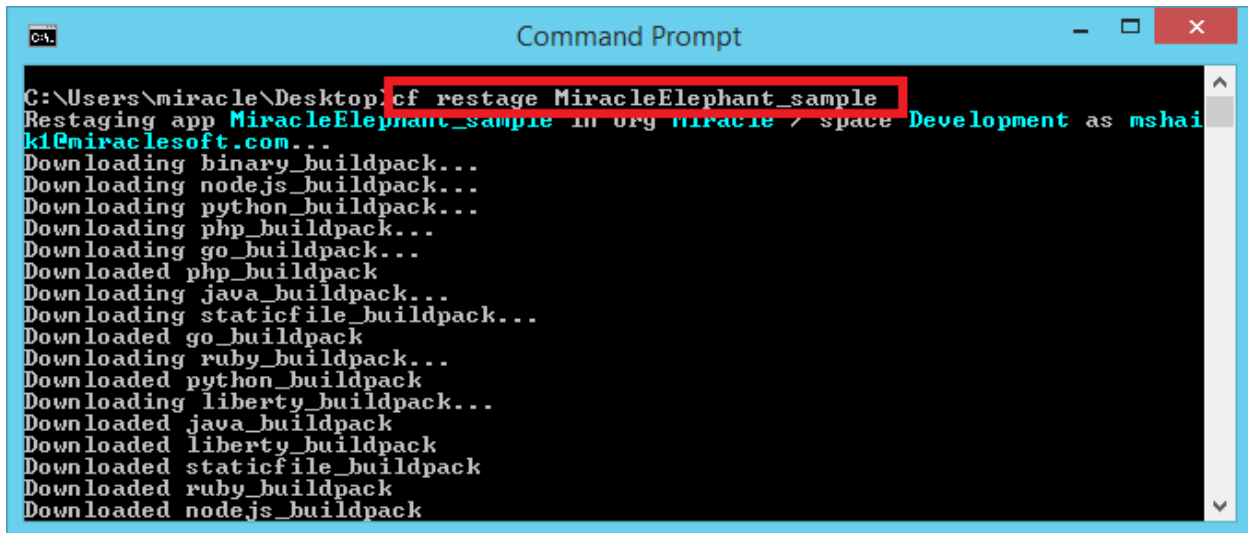
C:\Users\miracle\Desktop>
```

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

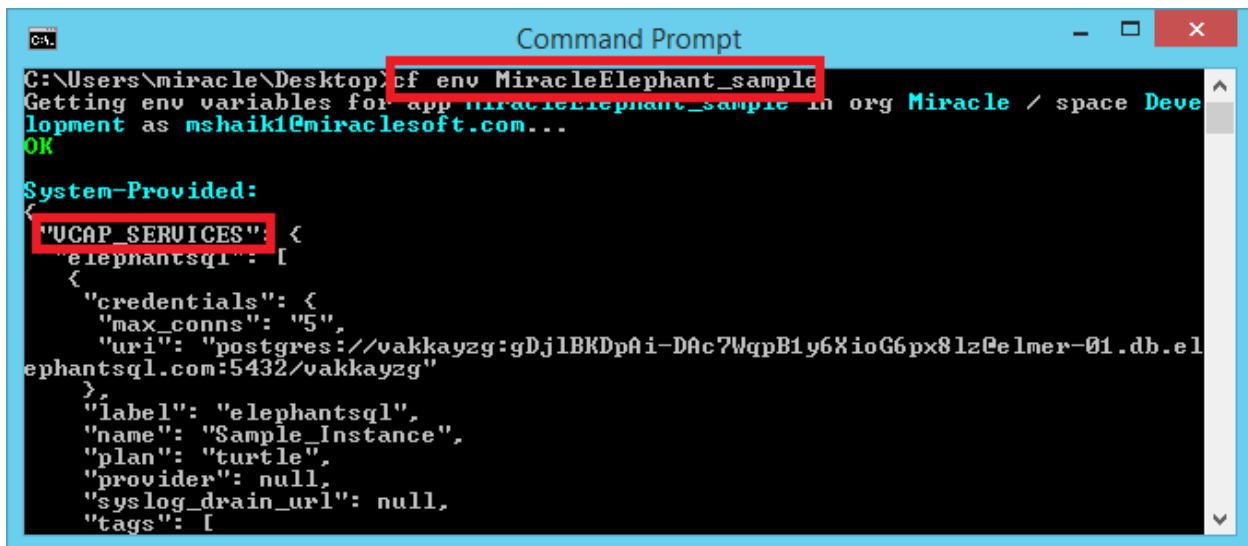
In restart, the application will be stopped and started again. On the other hand restage will create a new droplet and moves our environments into it from the existing droplet. By restaging the application the changes to the environment variables will take place. But in modern architecture the service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





```
CA. Command Prompt
C:\Users\miracle\Desktop>cf restage MiracleElephant_sample
Restaging app MiracleElephant_sample in org miracle / space Development as mshaiki@miraclesoft.com...
Downloading binary_buildpack...
Downloading nodejs_buildpack...
Downloading python_buildpack...
Downloading php_buildpack...
Downloading go_buildpack...
Downloaded php_buildpack
Downloading java_buildpack...
Downloading staticfile_buildpack...
Downloaded go_buildpack
Downloading ruby_buildpack...
Downloaded python_buildpack
Downloading liberty_buildpack...
Downloaded java_buildpack
Downloaded liberty_buildpack
Downloaded staticfile_buildpack
Downloaded ruby_buildpack
Downloaded nodejs_buildpack
```

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



```
CA. Command Prompt
C:\Users\miracle\Desktop>cf env MiracleElephant_sample
Getting env variables for app miracle/elephant_sample in org miracle / space Development as mshaiki@miraclesoft.com...
OK

System-Provided:
<
  "UCAP_SERVICES": <
    "elephantsql": [
      <
        "credentials": <
          "max_conns": "5",
          "uri": "postgres://vakayzg:gDj1BKDpAi-DAc7WqpB1y6XioG6px81z0elmer-01.db.elephantsql.com:5432/vakayzg"
        >
      >
      "label": "elephantsql",
      "name": "Sample_Instance",
      "plan": "turtle",
      "provider": null,
      "syslog_drain_url": null,
      "tags": [

```

The screenshot shows the Pivotal Web Services dashboard. The top bar indicates the user is logged in as mshaik1@miraclesoft.com. A notification banner at the top states: "Org 'Miracle' has about \$5 of credit left in its free trial, which includes 2GB of app memory and 10 of the free Marketplace services. Upgrade now". The left sidebar contains navigation links: ORG, Miracle (selected), SPACES, Development, Marketplace, Docs, Support, Tools, Blog, and Status. The main content area shows the 'Development' space with a status summary: 2 Running, 0 Stopped, 0 Crashed. Below this, there are tabs for Apps (2), Service (1), Security, and Settings. The 'Apps' tab is active, displaying a table of applications:

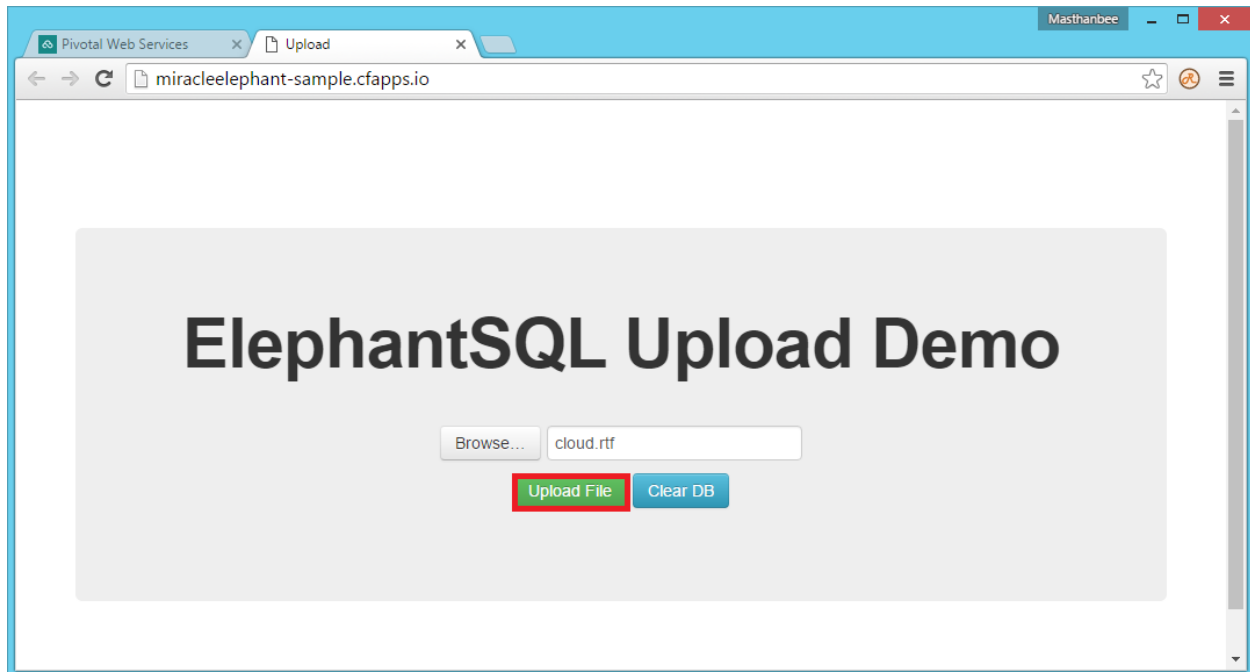
| NAME                                | INSTANCES | MEMORY | LAST PUSH     | ROUTE   |
|-------------------------------------|-----------|--------|---------------|---|
| Hello_Miracle_app<br>● Running      | 5         | 256MB  | 21 hours ago  | <a href="http://hello-miracle-app.cfapps.io">http://hello-miracle-app.cfapps.io</a>           |
| MiracleElephant_sample<br>● Running | 1         | 512MB  | 5 minutes ago | <a href="http://miracleelephant-sample.cfapps.io">http://miracleelephant-sample.cfapps.io</a> |

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

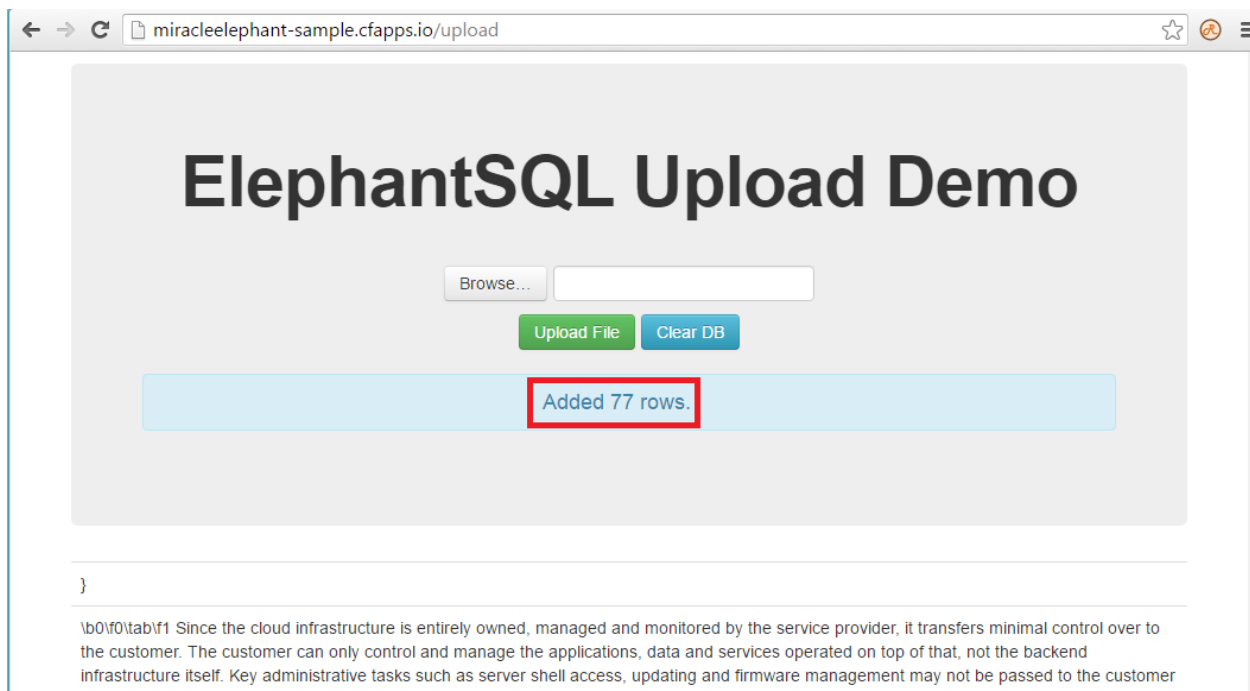
The screenshot shows a web browser window with the address bar displaying 'miracleelephant-sample.cfapps.io'. The page content features a large heading 'ElephantSQL Upload Demo'. Below the heading, there is a 'Browse...' button highlighted with a red box, followed by a text input field. At the bottom of the form, there are two buttons: 'Upload File' (green) and 'Clear DB' (blue).



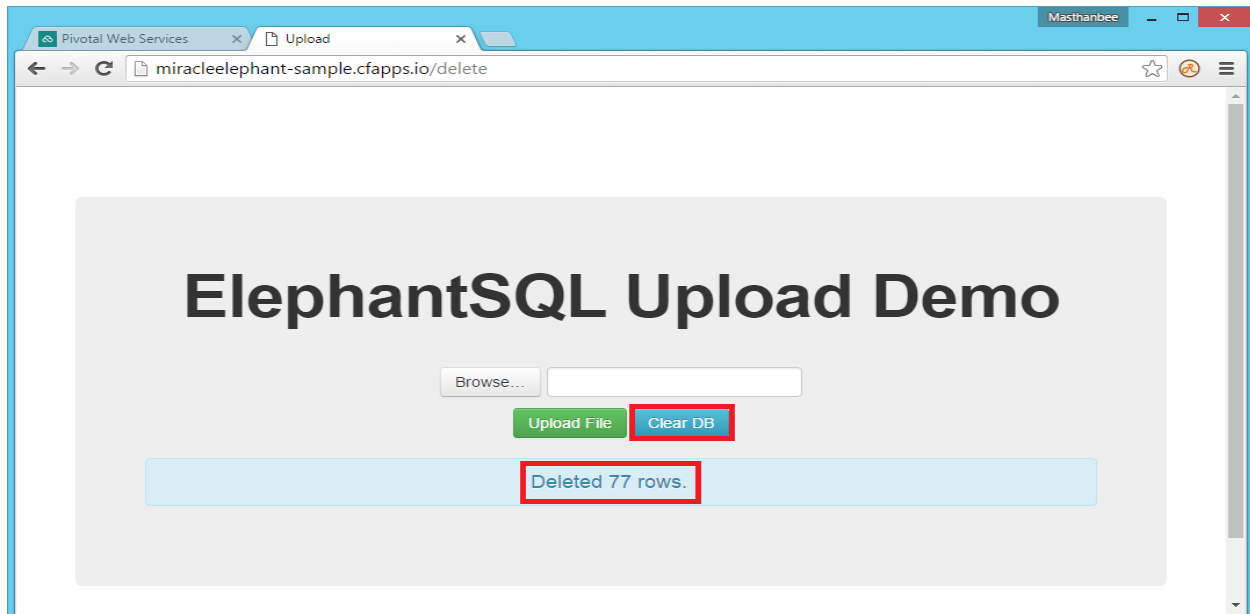
After browsing the application please click on the **Upload File** button.



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



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

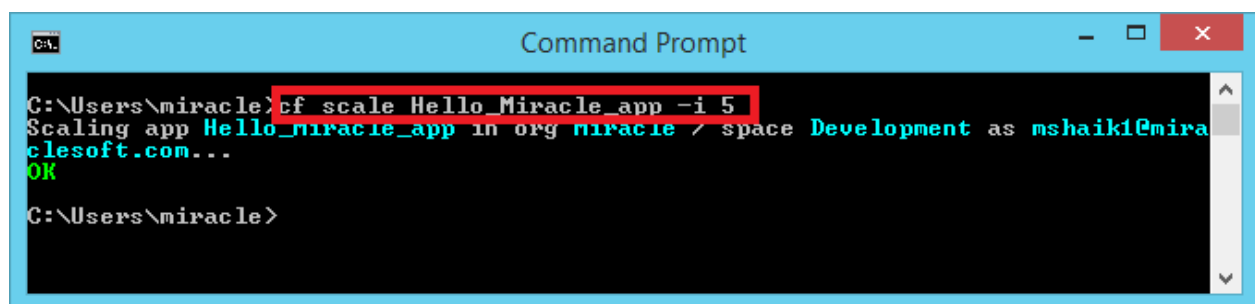


## Scaling

### 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.

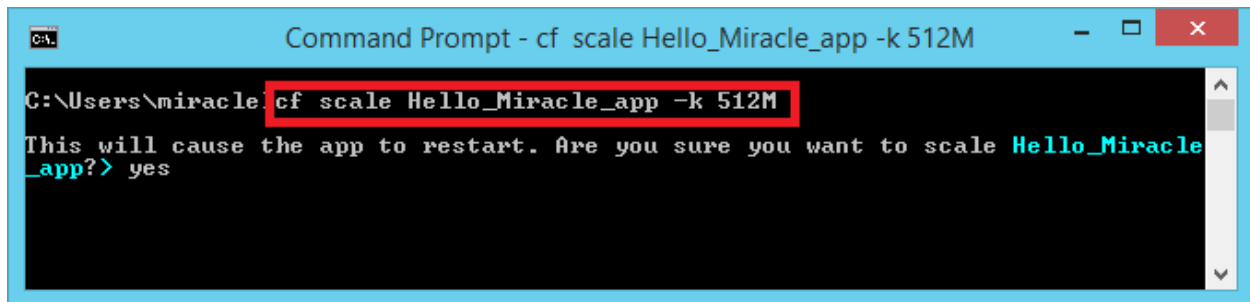
-> cf scale application\_name -i instances\_count



## 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



```
Command Prompt - cf scale Hello_Miracle_app -k 512M

C:\Users\miracle>cf scale Hello_Miracle_app -k 512M
This will cause the app to restart. Are you sure you want to scale Hello_Miracle_app?> yes
```

->cf scale application\_name ram\_size



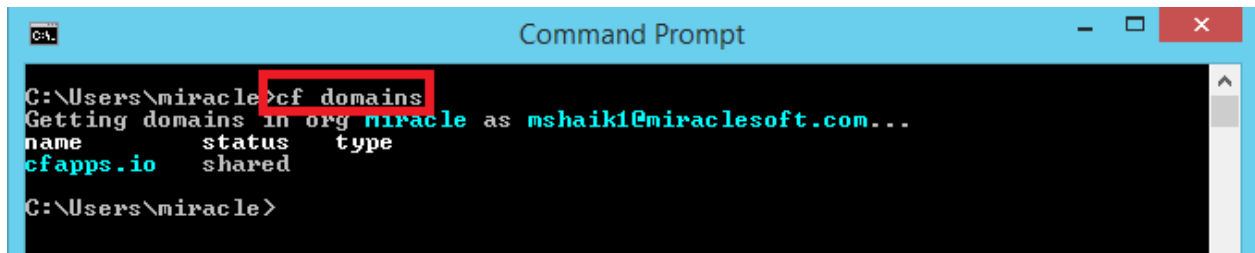
```
Command Prompt - cf scale Hello_Miracle_app -m 1G

C:\Users\miracle>cf scale Hello_Miracle_app -m 1G
This will cause the app to restart. Are you sure you want to scale Hello_Miracle_app?> yes_
```

## 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



```
Command Prompt
C:\Users\miracle>cf domains
Getting domains in org miracle as mshaik1@miraclesoft.com...
name      status  type
cfapps.io shared
```

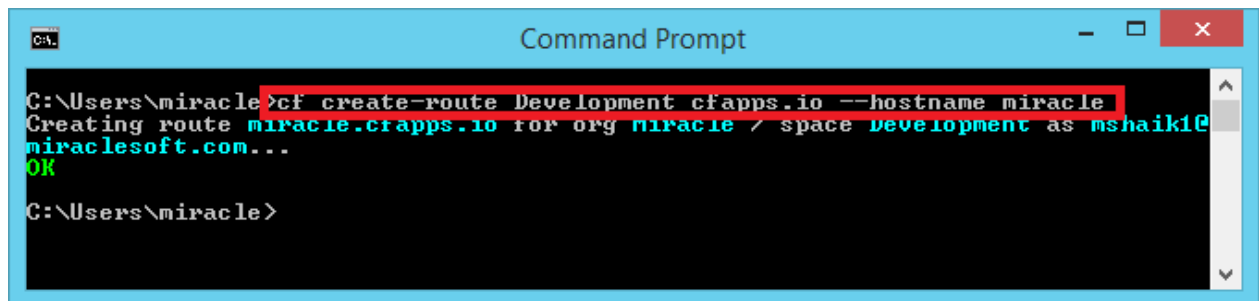
In general our applications will get a URL with the application name followed by domain name. For example if our application name is miracle and the domain name is cfapps.io then our URL will be miracle.cfapps.io

## 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

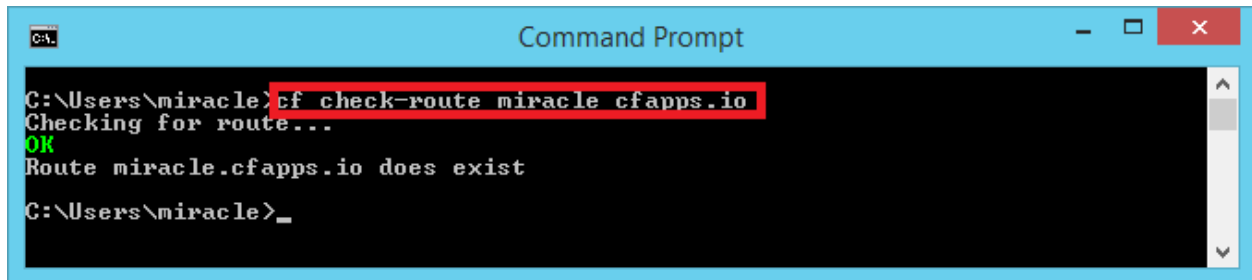
->cf create-route space\_name domain\_name --hostname name\_of\_the\_host



```
Command Prompt
C:\Users\miracle>cf create-route Development cfapps.io --hostname miracle
Creating route miracle.cfapps.io for org miracle / space development as mshaik1@miraclesoft.com...
OK
C:\Users\miracle>
```

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

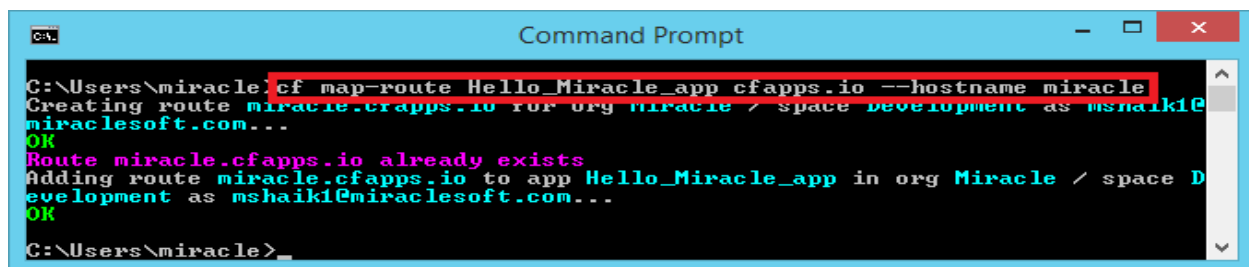
->cf check-route hostname domain\_name



```
Command Prompt
C:\Users\miracle>cf check-route miracle cfapps.io
Checking for route...
OK
Route miracle.cfapps.io does exist
C:\Users\miracle>_
```

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

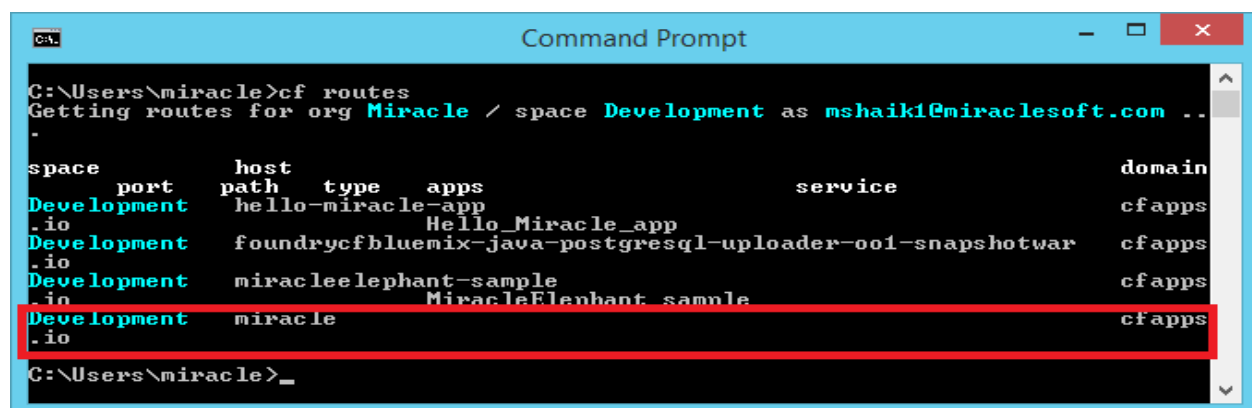
->cf map-route application\_name domain\_name [--hostname]



```
Command Prompt
C:\Users\miracle>cf map-route Hello_Miracle_app cfapps.io --hostname miracle
Creating route miracle.cfapps.io for org miracle / space development as mshaik1@miraclesoft.com...
OK
Route miracle.cfapps.io already exists
Adding route miracle.cfapps.io to app Hello_Miracle_app in org Miracle / space Development as mshaik1@miraclesoft.com...
OK
C:\Users\miracle>_
```

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

->cf routes



```
Command Prompt
C:\Users\miracle>cf routes
Getting routes for org Miracle / space Development as mshaik1@miraclesoft.com ..
-
space      port      host      type      apps      service      domain
Development .io      hello-miracle-app      Hello_Miracle_app      cfapps
Development .io      foundrycfbluemix-java-postgresql-uploader-001-snapshotwar      cfapps
Development .io      miracleelephant-sample      MiracleElephant sample      cfapps
Development .io      miracle      cfapps
C:\Users\miracle>_
```