**Spring Cloud Pivotal**

**Cloud Foundry:**

It is a [PaaS service](https://en.wikipedia.org/wiki/Platform_as_a_service) where we can easily deploy and manage our applications and the Cloud Foundry will take care of the rest of the cloud based offerings like scalability, high availability etc.

**What is Cloud Foundry?**

Cloud foundry is an open source platform as service offering provided by pivotal

Let’s discuss what is platform as service offering?

We have several cloud platform like Google cloud, Amazon Cloud and Spring Cloud and pivotal Cloud where we can deploy our micro services Or any application but each cloud factory have their own patterns to configure, deploy, managing etc.

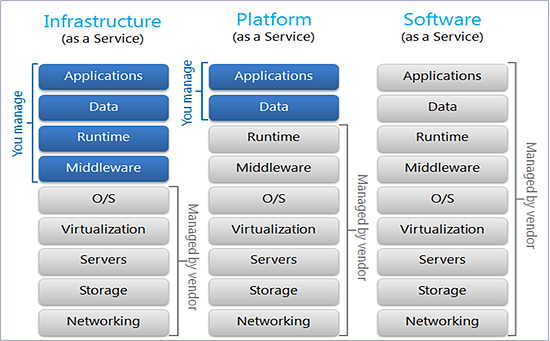
So each of them offering service us to deploy application.

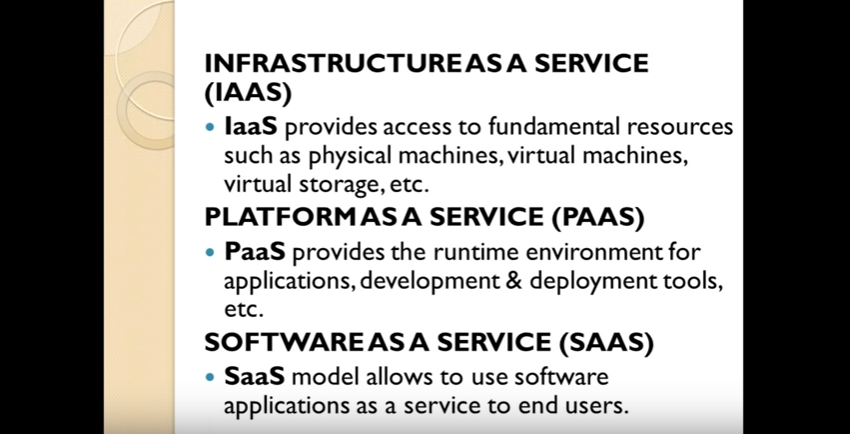
So let’s see what are the service they providing and how they categorized, normally it divided in 3 types

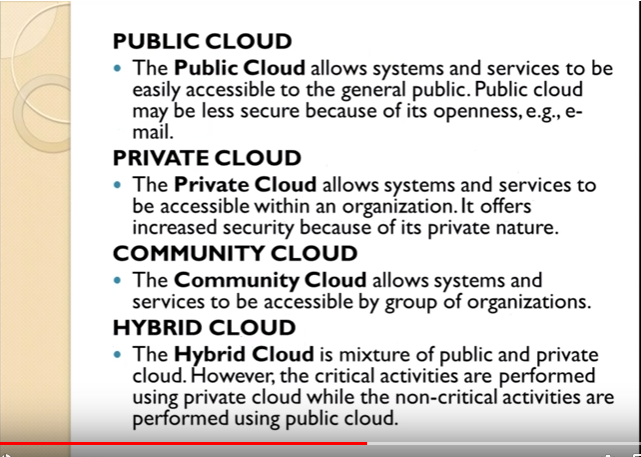
1. Infrastructure

2. Platform

3. Software.







So discuss one by one.  
As per architecture normally we required few support to deploy our application in any environment

1.    Application  
2.    Data  
3.    Runtime  
4.    Middleware  
5.    Operating system  
6.    Virtualization  
7.    Servers  
8.    Storage  
9.    Networking

Now let’s see what each service offering

**Infrastructure Service Offering**

1.    Application  
2.    Data  
3.    Runtime  
4.    Middleware  
5.    Operating system  
6.    Virtualization  
7.    Servers  
8.    Storage  
9.    Networking

Infrastructure provides only green mark fields Rest all we need to manually configure in that cloud environment

Like: workspace, runtime environment and API related configure that we need to do manually

The Best Example is AWS (Amazon Web service) if any one worked on AWS they marked everything we need to set there like Virtual box

**Platform Service Offering**

1.    Application  
2.    Data  
3.    Runtime  
4.    Middleware  
5.    Operating system  
6.    Virtualization  
7.    Servers  
8.    Storage  
9.    Networking

Platform provides only green mark fields Rest all we need to manually configure in that cloud environment

So here you need to bother about only your application set up and data set rest all inbuilt by platform service provider like you can choose your own platform like java, python or any tech they have multi environment supports

The best Example is GCP and Pivotal Cloud Foundry and Cloud Foundry have to great support to Our Spring Cloud

**Software Service Offering**

1.    Application  
2.    Data  
3.    Runtime  
4.    Middleware  
5.    Operating system  
6.    Virtualization  
7.    Servers  
8.    Storage  
9.    Networking

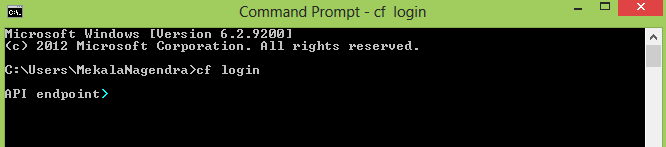
Software service provides all the required environment   
Best Example is Google docs

So now in this article am going to explain Platform Offering Service that with Pivotal Cloud Foundry So what we need to manage for this only our application and data. Rest all will provide by Cloud Foundry

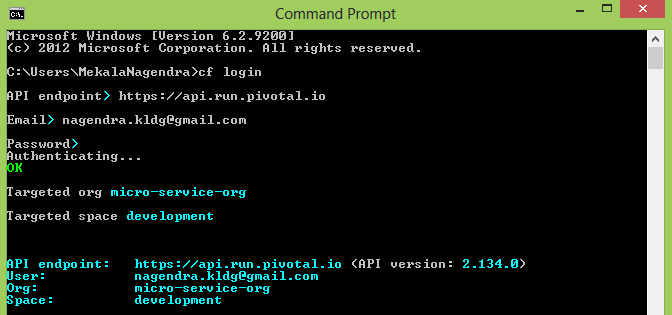
**Login to Pivotal Webservices Account using CF CLI**

Open the command terminal and use the following command

cf login



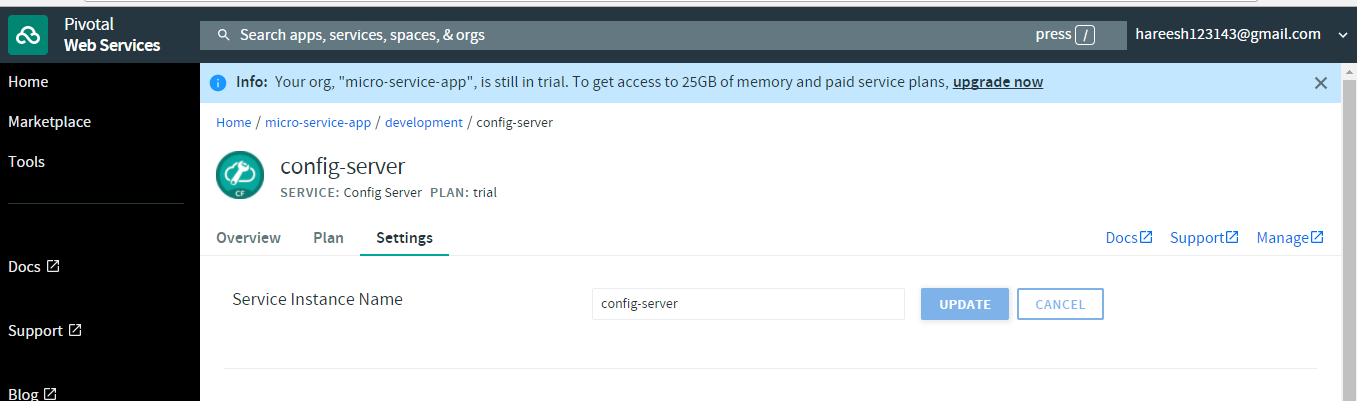
It will ask for Cloud Foundry API. Enter The following API value-



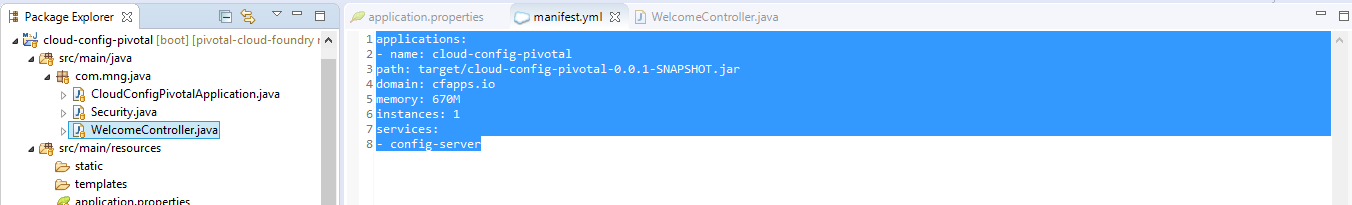
<https://javagyanmantra.wixsite.com/website/single-post/2018/02/14/Pivotal-Cloud-Foundry>

**Config client using Pivotal config server Application:**

Step 1 :

****

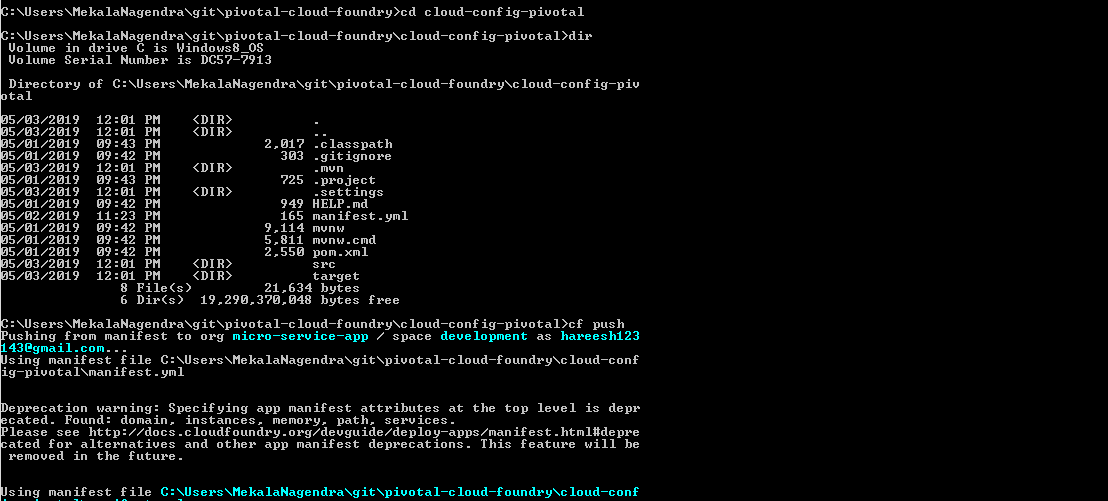
**Created manifest file pivotal deployment.**

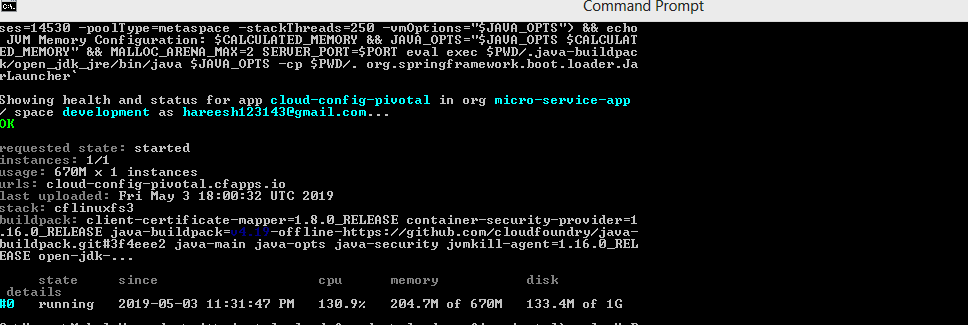


**Push the application into pivotal:**

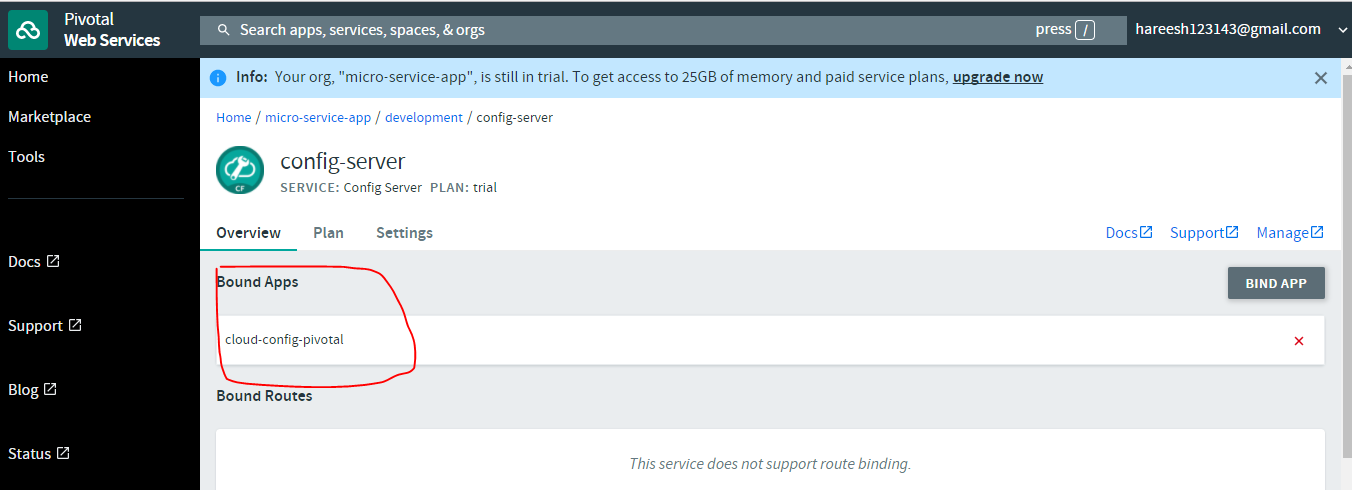
Step 1: go to project directory.

Step 2: push application jar into pivotal



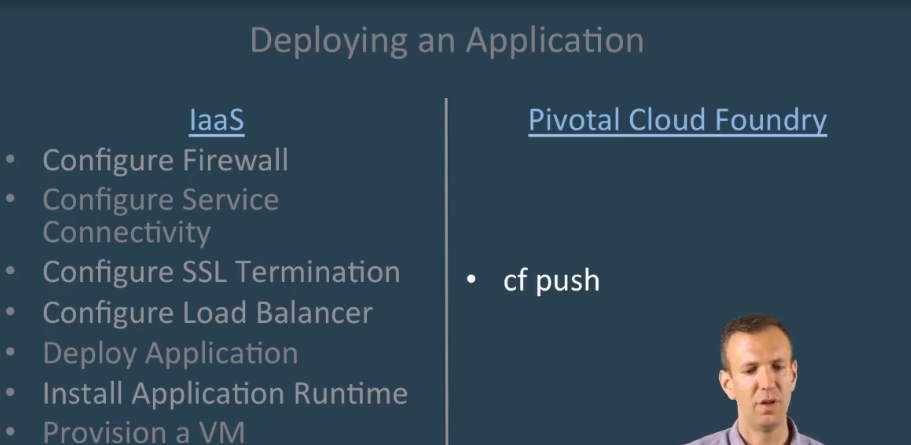
****

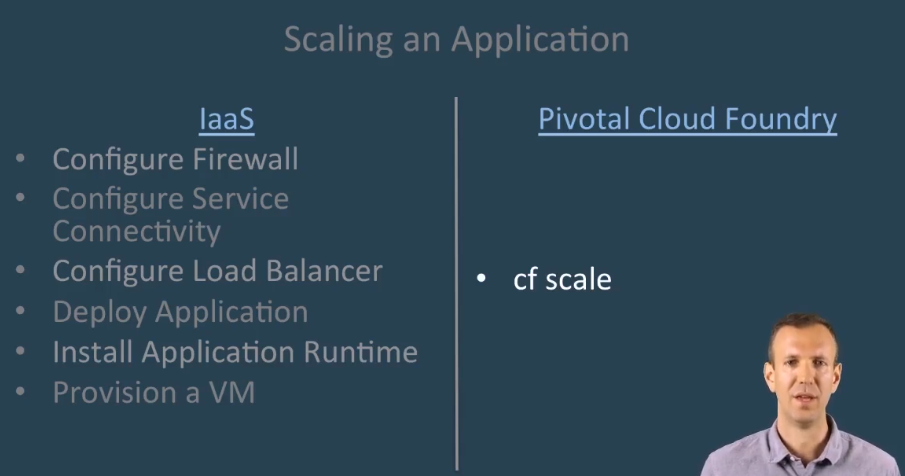
**After pushing application internally pivotal config server bounded into cloud-config-pivotal application**

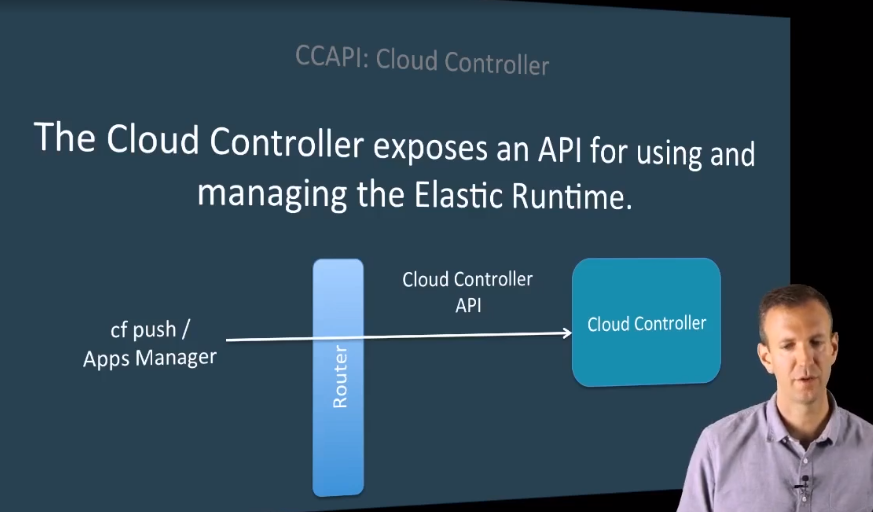
****

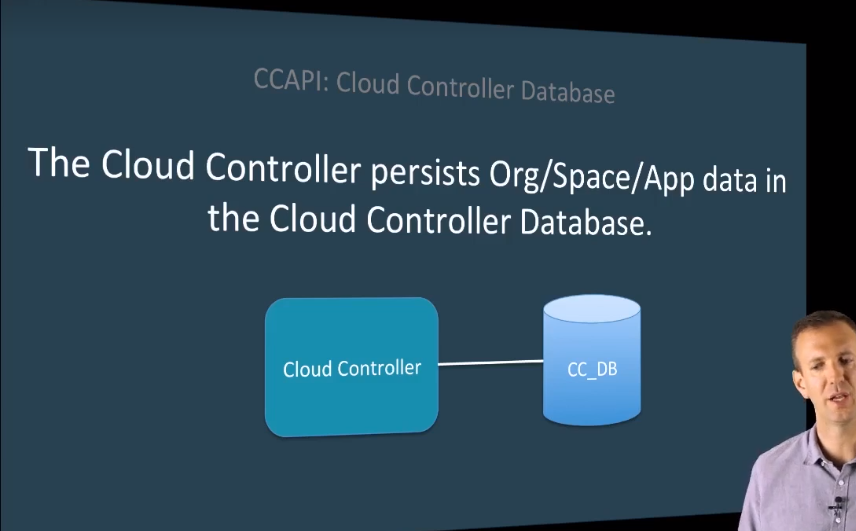
Step 3: hit the application url

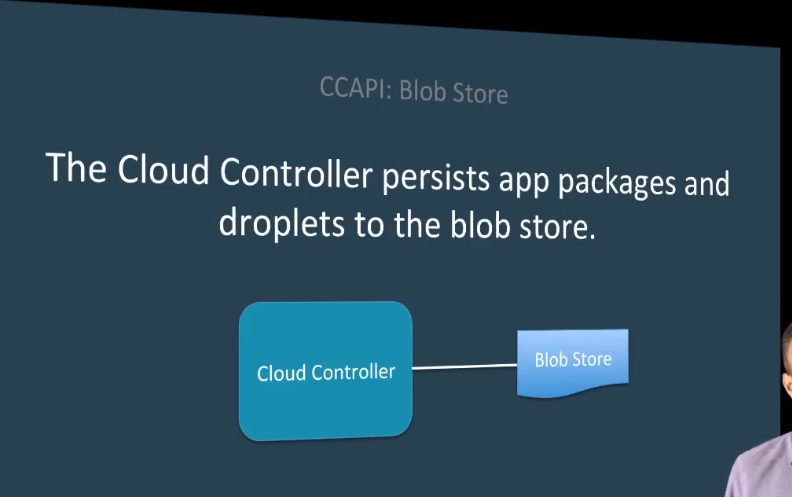


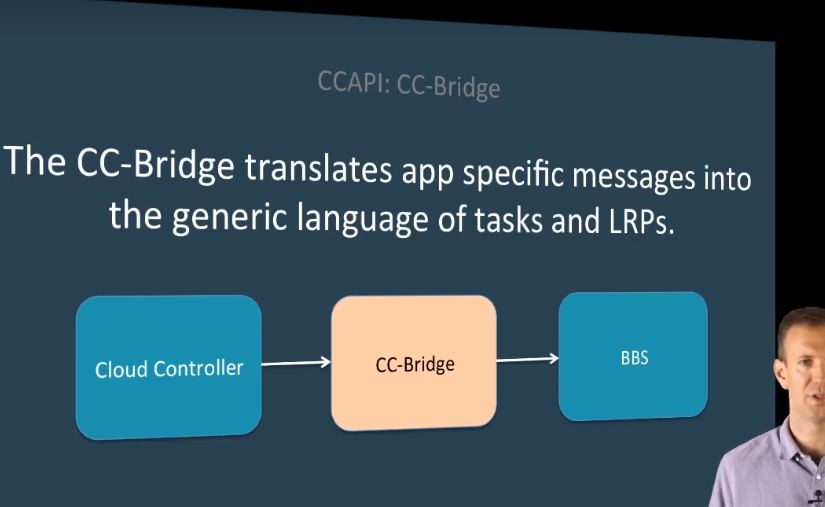


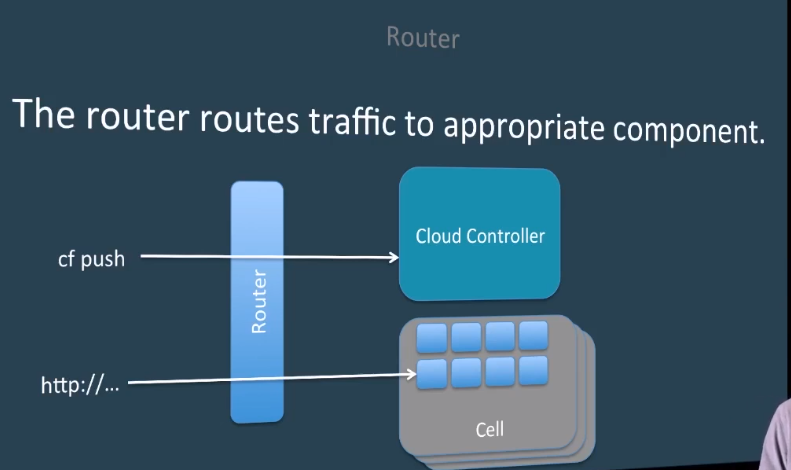


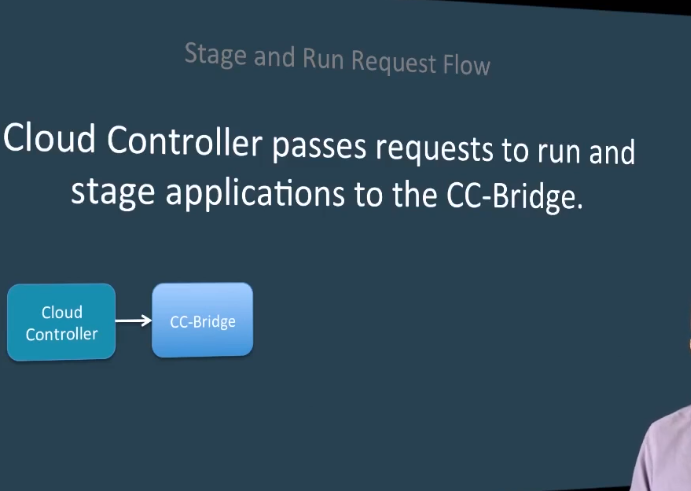












|  |
| --- |
| **Commands :**  cf apps // display the apps information from cloud foundry.  cf stop my-example-app // stop the app  cf start my-example-app // start the app  cf push my-example-ap -p ./my-example-ap.jar -m 512  cf logs my-example-ap  cf scale my-example-ap –I 2  Use cf scale APP -i INSTANCES to horizontally scale your app. Cloud Foundry will increase or decrease the number of instances of your app to match INSTANCES.  **cf scale myApp -k 512M**  Use cf scale APP -k DISK to change the disk space limit applied to all instances of your app. DISK must be an integer followed by either an **M**, for megabytes, or **G**, for gigabytes.  **cf scale myApp -m 1G**  Use cf scale APP -m MEMORY to change the memory limit applied to all instances of your app. MEMORY must be an integer followed by either an **M**, for megabytes, or **G**, for gigabytes.  cf |

# Difference between stateless protocol and statefull protocol

**Stateless Protocol:-**

* When stateless protocol is used between a server and the client, the server does not remember anything. It treats any message from a client as the client's first message and responds with the same effects every time
* A stateless server does not keeps state between connections.=>So,When you send a request to a stateless server, it does not create any objects that track information regarding your requests. If you "open" something on the server, the server retains no information at all that you have something open. A "close" operation would make no sense, since there would be nothing to close.

**Statefull Protocol :-**

* Stateful protocol means the server remembers what a client has done before.
* Stateful server keeps state between connections.=> when you send a request to a stateful server, it may create some kind of connection object that tracks what information you request. When you send another request, that request operates on the state from the previous request. So you can send a request to "open" something. And then you can send a request to "close" it later. In-between the two requests, that thing is "open" on the server.
* A protocol which requires the keeping of internal state is known as a stateful protocol.
* statefull Protocol :- maintain the state ,but cannot maintain the persistance, once we shutdown the system the values stored in local hard disc, can have the passivate and activate states. ex: shopping cardExample = SMB, FTP,Telnet