# Creating Config-Server Instance And Binding To PCF Apps

Open Platform Engineering

Exported on 02/22/2018

# Table of Contents

# 1 Introduction

Config Server[1] for Pivotal Cloud Foundry (PCF) is an externalized application configuration service, which gives you a central place to manage an application's external properties across all environments. The properties are kept in a repo preferably github. NBCUniversal has enterprise Github and upon request, the github repo is provided by the team.

1 https://docs.pivotal.io/spring-cloud-services/1-3/common/config-server/

# 2 Pre-Requisite

Please note that, when you are trying to use config-server, you have a github account. If you do not have, please raise a ticket to NBC Github team.

Open a generic request and edit as:

- Configuration Item: Software
- Subcategory: Other
- Configuration Item: Github Enterprise
- Assignment Group: Platform Software Tools

Once you get the repo from the team you need to create a config-server instance, bind it to app and verify if you have got the account online. If not configured properly, you might get an error “Config Server not initialized properly”.

# 3 Create an Instance

Creating config-server instance can be done in 2 ways:

1. Using CF-CLI
2. From Apps Manager Dashboard

# 4 Using cf cli

- Target your org and space from cf cli.
- Check if the config-server is available in your foundation by typing: `cf m`

```
[root@ashapld00004 openplatform]# cf t

API endpoint:   https://api.devsysash.inbcu.com (API version: 2.65.0)
User:           admin
Org:            SRA-Linear
Space:          Software-Eng-sandbox
[root@ashapld00004 openplatform]# cf m
Getting services from marketplace in org SRA-Linear / space Software-Eng-sandbox as admin...
OK

service                         plans                   description
app-autoscaler                  standard                Scales bound applications in response to load (beta)
newrelic                        NBCU_NewRelic           Manage and monitor your apps
p-circuit-breaker-dashboard     standard                Circuit Breaker Dashboard for Spring Cloud Applications
p-config-server                 standard                Config Server for Spring Cloud Applications
p-mysql                         pre-existing-plan       MySQL databases on demand
p-rabbitmq                      standard                RabbitMQ is a robust and scalable high-performance multi-protocol messa
ging broker.
p-redis                         shared-vm, dedicated-vm Redis service to provide a key-value store
p-service-registry              standard                Service Registry for Spring Cloud Applications

TIP:  Use 'cf marketplace -s SERVICE' to view descriptions of individual plans of a given service.
[root@ashapld00004 openplatform]#
```

- Ideally, we would recommend you to put the git repo info in a json file and create an instance by referring it. For an example, for my repo, the details are as follows and kept in a file named config.json. The json file must be valid one and all the properties have accurate information. The content of config.json is as follows:

```
{     "git":{
       "uri":"https://github.inbcu.com/<YOUR_GIT_PATH>[2]",
       "username":"<SSO_ID>",
       "password":"xxxxxxxxxxxxxxxxxxxxx",
       "searchPaths":"<PATH_IF_ANY>",
       "proxy":{
          "http":{
             "host":"proxy.inbcu.com",
             "port":"80",
             "nonProxyHosts":"uaa.sysush.inbcu.com|p-spring-cloud-
services.uaa.sysush.inbcu.com| *.uaa.sysush.inbcu.com|
*.login.sysush.inbcu.com"
          },
          "https":{
             "host":"proxy.inbcu.com",
             "port":"80",
             "nonProxyHosts":"uaa.sysush.inbcu.com|p-spring-cloud-
services.uaa.sysush.inbcu.com| *.uaa.sysush.inbcu.com|
*.login.sysush.inbcu.com"
          }
       }
    }
}
```

- When we create an instance from cf cli, we need to provide below command.

```
cf create-service p-config-server standard myConfigServer -c
config.json
```

- I have kept my config.json file in the location from where I am executing the command. You can provide the path, where your config file is located.

2 https://github.inbcu.com/%3cYOUR_GIT_PATH%3e

```
[root@ashapld00004 openplatform]# cf create-service p-config-server standard myConfigServer -c config.json
Creating service instance myConfigServer in org SRA-Linear / space Software-Eng-sandbox as admin...
OK

Create in progress. Use 'cf services' or 'cf service myConfigServer' to check operation status.
[root@ashapld00004 openplatform]#
```

- Check the status of the service by typing: ***cf service myConfigServer***

```
[root@ashapld00004 openplatform]# cf service myConfigServer

Service instance: myConfigServer
Service: p-config-server
Bound apps:
Tags:
Plan: standard
Description: Config Server for Spring Cloud Applications
Documentation url: http://docs.pivotal.io/spring-cloud-services/
Dashboard: https://spring-cloud-broker.devash.inbcu.com/dashboard/p-config-server/d7261cfe-5ece-4f77-8350-2d86b52cd2b2

Last Operation
Status: create succeeded
Message:
Started: 2017-04-25T13:31:21Z
Updated: 2017-04-25T13:32:26Z
[root@ashapld00004 openplatform]#
```

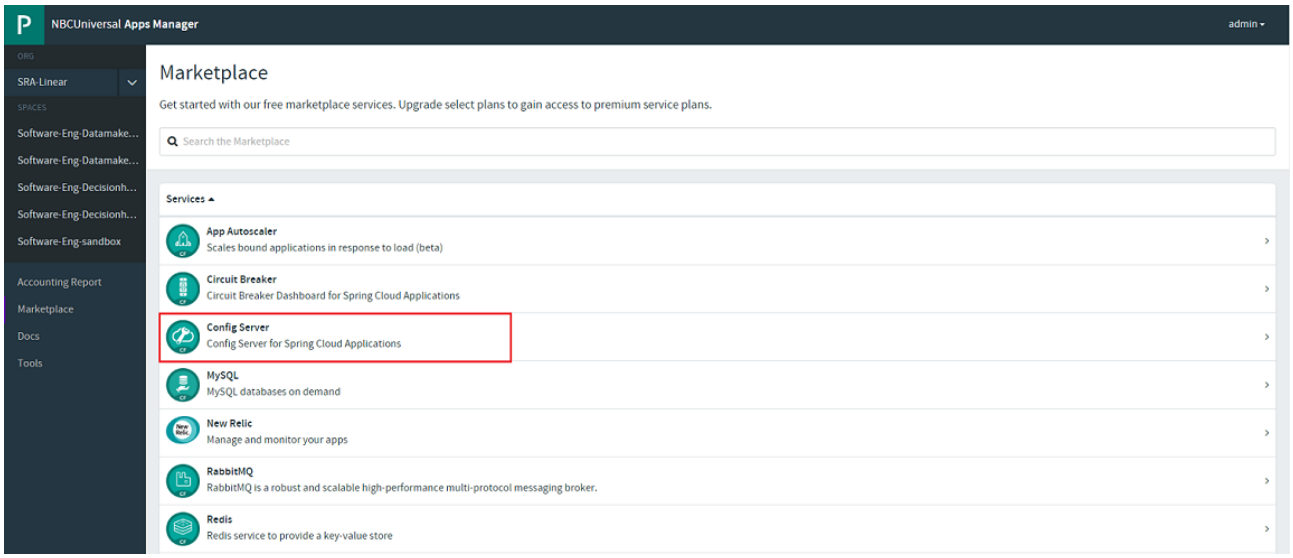
- If the Status and only after the Status: ***create succeeded***, go ahead and bind the service instance with the app that you want to use. The bind can be done via cf cli or mentioning the instance in manifest.yml file.
    1. Bind via cli - ***cf bind-service myapp myConfigServer***
    2. Bind in the manifest.yml

```
---
applications:
- name: myapp
  random-route: true
  …
 …..
  services:
   - myConfigServer
```

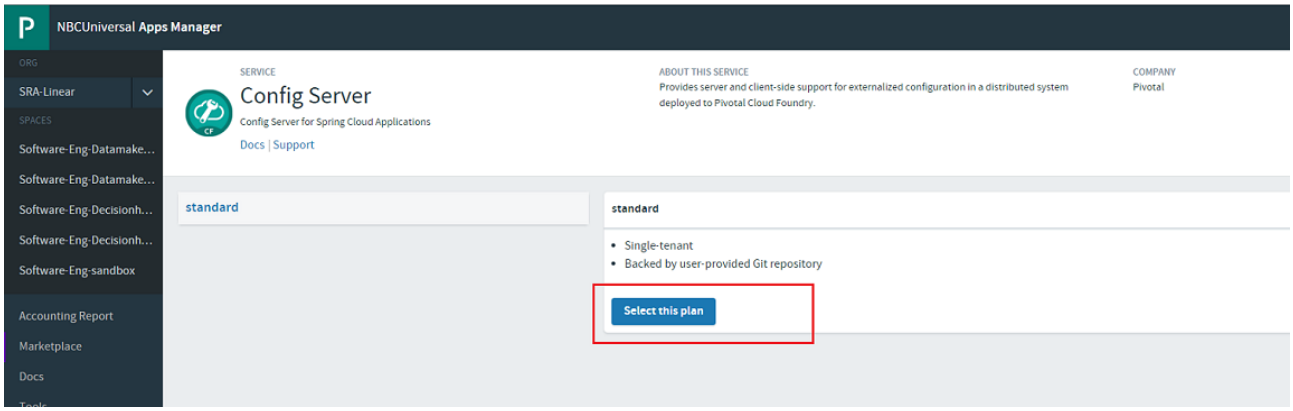
- Once it is done, you are good to use your config-server instance and refer the files from the configured repo.

# 5 using APPS Manager

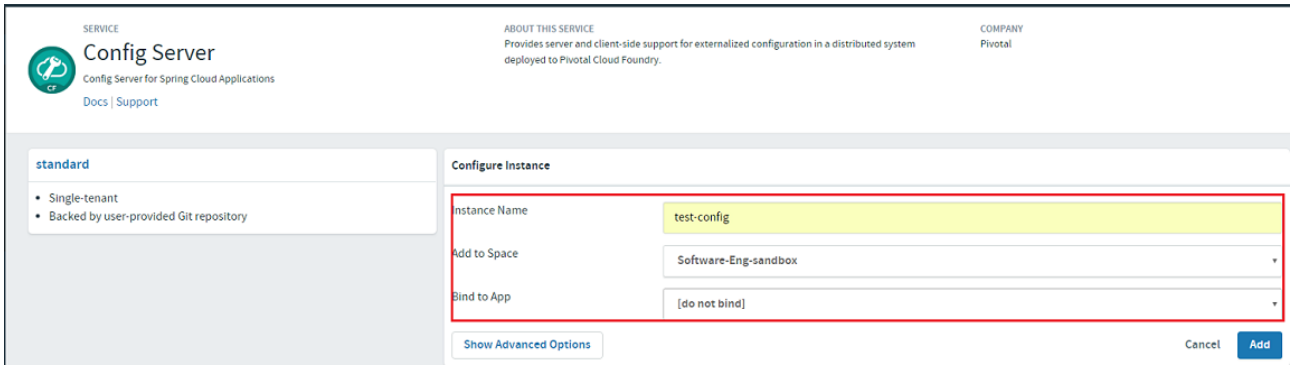
- Log into Apps Manager as a Space Developer. In the Marketplace, select **Config Server**.




- Select the desired plan for the new service instance.




- Provide a name for the service instance (for example, “test-config”). Click the **Add** button.




- In the **Services** list, click the **Manage** link under the listing for the new service instance.

It may take a few minutes to provision the service instance; while it is being provisioned, you will see a “The service instance is initializing” message. When the instance is ready, its dashboard will load automatically.

**Note**: If you create a Config Server service instance using Apps Manager, you will need to use the cf CLI to update the service instance and provide settings (including one or more configuration sources) before the instance will be usable. For information on updating a Config Server service instance, see the Updating an Instance[3] topic.

To update the service instance you need to provide all the details that needs to be put in cf cli like below:

```
#cf update-service myConfigServer -c '{"git": {"uri": "https://
github.inbcu.com/OpenPlatform/Workshop", "username": "20####67",
"password": "a$@$$#*#" } }'
```

You can provide a lot of information like if you are using proxies and other information.

For further assistance, please contact pcf.operations@nbcuni.com[4] .

```
{
  "git":{
    "uri":"https://github.inbcu.com/<YOUR_GIT_PATH>[5]",
    "username":"<SSO_ID>",
    "password":"xxxxxxxxxxxxxxxxxxxxx",
    "searchPaths":"<PATH_IF_ANY>",
    "proxy":{
      "http":{
        "host":"proxy.inbcu.com",
        "port":"80",
        "nonProxyHosts":"uaa.sysush.inbcu.com|p-spring-cloud-services.uaa.sysush.inbcu.com|
*.uaa.sysush.inbcu.com| *.login.sysush.inbcu.com"
      },
      "https":{
        "host":"proxy.inbcu.com",
        "port":"80",
        "nonProxyHosts":"uaa.sysush.inbcu.com|p-spring-cloud-services.uaa.sysush.inbcu.com|
```

3 https://docs.pivotal.io/spring-cloud-services/config-server/updating-an-instance.html
4 mailto:pcf.operations@nbcuni.com
5 https://github.inbcu.com/%3cYOUR_GIT_PATH%3e

```
*.uaa.sysush.inbcu.com| *.login.sysush.inbcu.com"
      }
    }
  }
}
```