DCOS 应用仓库之Universe

2016年4月，Mesosphere开源了他们开发的DC/OS（数据中心操作系统），作为一个数据中心操作系统而言，基于其之上的各种各样的应用是不可或缺的。在DC/OS系统中，Cosmos扮演了App Store的作用，Cosmos提供的API可以方便的通过DC/OS的UI以及DC/OS的CLI来完成应用仓库的管理，应用的部署，应用的卸载等功能。

Universe简介：

Cosmos的应用仓库管理是通过Universe和Cosmos共同完成的，Universe是mesosphere的一个开源项目，其目的是为DC/OS提供一个应用软件包仓库，目前在mesosphere github(https://github.com/mesosphere/universe) 上的应用包提供了一些能跑在Marathon之上的App以及一些mesos上层的framework。

今天我们就先从介绍universe开始。

准备工作：

首先，我们需要把mesosphere github 上的universe克隆到本地。

# git clone https://github.com/mesosphere/universe

# ll universe

-rw-r--r-- 1 hchenxa staff 10834 Jun 3 14:40 LICENSE

-rw-r--r-- 1 hchenxa staff 583 Jun 3 14:40 Makefile

-rw-r--r-- 1 hchenxa staff 11852 Jun 3 14:40 README.md

drwxr-xr-x 5 hchenxa staff 170 Jun 3 14:40 docker/

drwxr-xr-x 3 hchenxa staff 102 Jun 3 14:40 hooks/

drwxr-xr-x 6 hchenxa staff 204 Jun 3 14:40 local/

drwxr-xr-x 4 hchenxa staff 136 Jun 14 15:32 repo/

drwxr-xr-x 14 hchenxa staff 476 Jun 14 17:48 scripts/

在本机安装jsonschema的python包，这个包从https://pypi.python.org/pypi就能找到，或者你也可以通过以下命令进行安装。

#sudo pip install jsonschema

然后执行以下命令来完成pre-commit hook，你本地的universe开发环境就已经准备就绪了:

bash /universe\_path/scripts/install-git-hooks.sh

应用包介绍：

Universe的所有应用包都存放在repo 底下，repo下面有2个目录，meta里面存放了repo的meta用于Cosmos对包进行检索，以及定义应用包内容的schema; packages底下就是各种各样的应用，这些应用按照首字母分别归类到不同的目录下面，以nginx为例，它的应用包就会放到repo/packages/N/nginx 下面。

Universe支持应用的多版本控制，不同的版本的相通应用放到不同的以数字0开始命名的文件夹里。

如下面的组织图所示，一个nginx有2个版本并且每个版本的文件夹中包涵5个文件：

nginx

├── 0

│   ├── command.json

│   ├── config.json

│   ├── marathon.json.mustache

│   ├── package.json

│   └── resource.json

├── 1

│   ├── command.json

│   ├── config.json

│   ├── marathon.json.mustache

│   ├── package.json

│   └── resource.json

package.json:

package.json里面主要是一些应用包的介绍，包括应用包的版本，维护着的email, 应用包名称，描述等信息，这些信息都会被Cosmos收集并且展示到DC/OS的UI上面。

config.json:

config.json里面主要是一些应用部署时候所需要的一些参数，不如应用所需要创建的Marathon instance的个数已经每个instance所占用的cpu和mem的大小等等。当然，在用户通过DC/OS 来提交应用的时候，这些参数都会作为应用的高级配置展示给用户，用户可以根据自己的需求来提交这些应用。

command.json

Command.json并不是每个应用都需要用到，这个文件只有当应用需要一些脚本完成按安装的时候才会被使用，而且目前只支持pip这一种方式。

resource.json

Resource.json里面定义了应用的在DC/OS上面所显示的图片，图标，帮助信息等等，

marathon.json.mustache

marathon.json.mustache是用来创建marathon APP所需要的App json文件，里面是对marathon app的定义，具体参数定义和格式可以参考marathon的rest-api文档（<https://mesosphere.github.io/marathon/docs/rest-api.html>），当然，cosmos会去合并config.json里面的一些用户自定义的属性最后在发给marathon去创建app.

制作应用仓库

上面说了这么多，怎么能够使DC/OS系统能够使用我本地的仓库呢？

当你完成了以上这些工作并且也创建好了自己的应用包，执行下面命令开始build属于你的应用仓库。

# bash scripts/build.sh

Building the universe!

Validating version...

OK

Validating package definitions...

- /N/ngnix/1/config.json

- /N/nginx/1/package.json

- /N/nginx/1/resource.json

- /N/nginx/1/command.json

OK

Building index...

OK

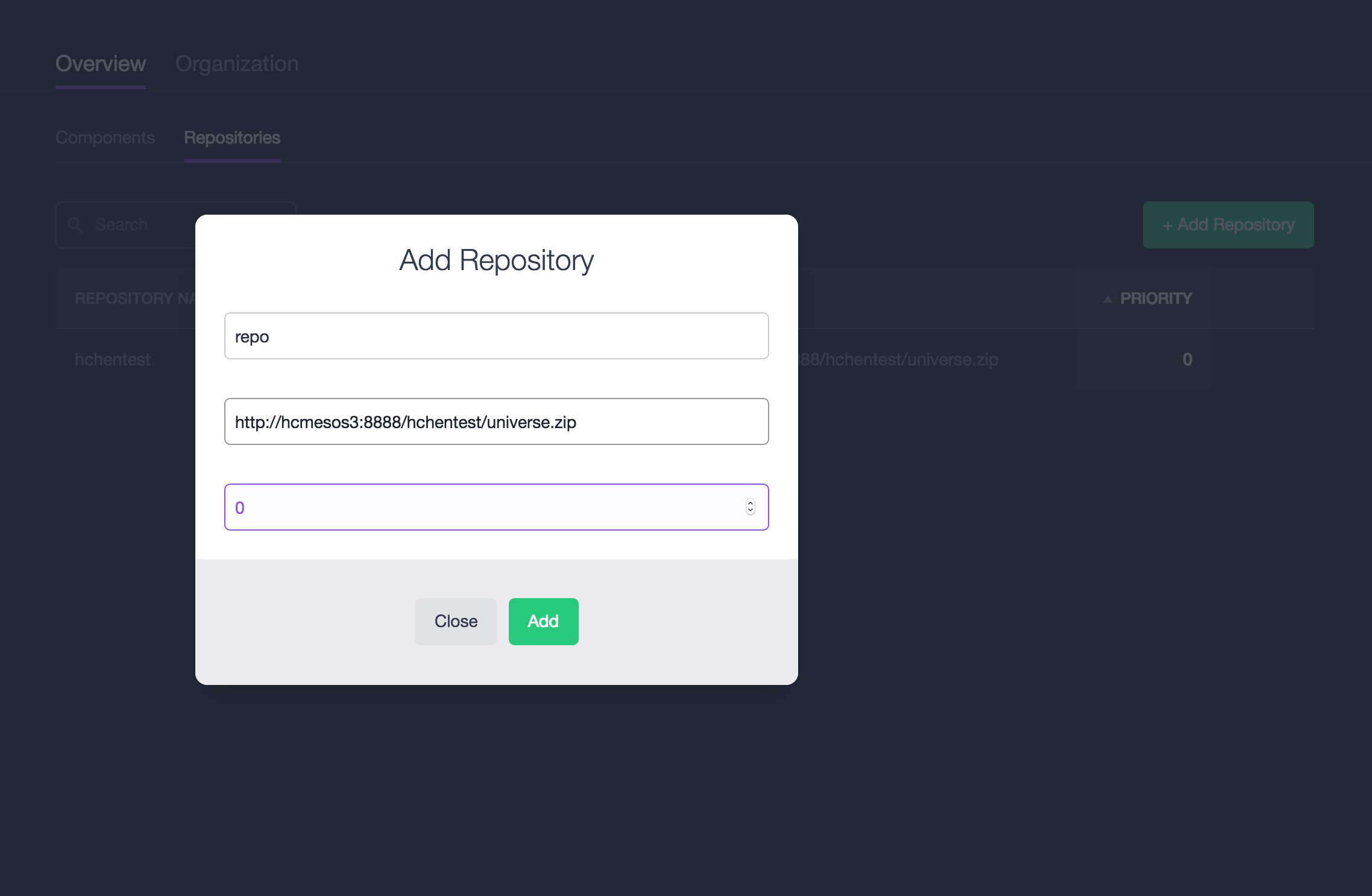
Validating index...

OK

Build会去校验版本以及应用包里面的文件，然后回去重新刷新index 并且对index进行校验。

然后打包你本地的universe并且把它存放在一个http/https的服务器上面（universe的repositories目前只支持这两种协议）

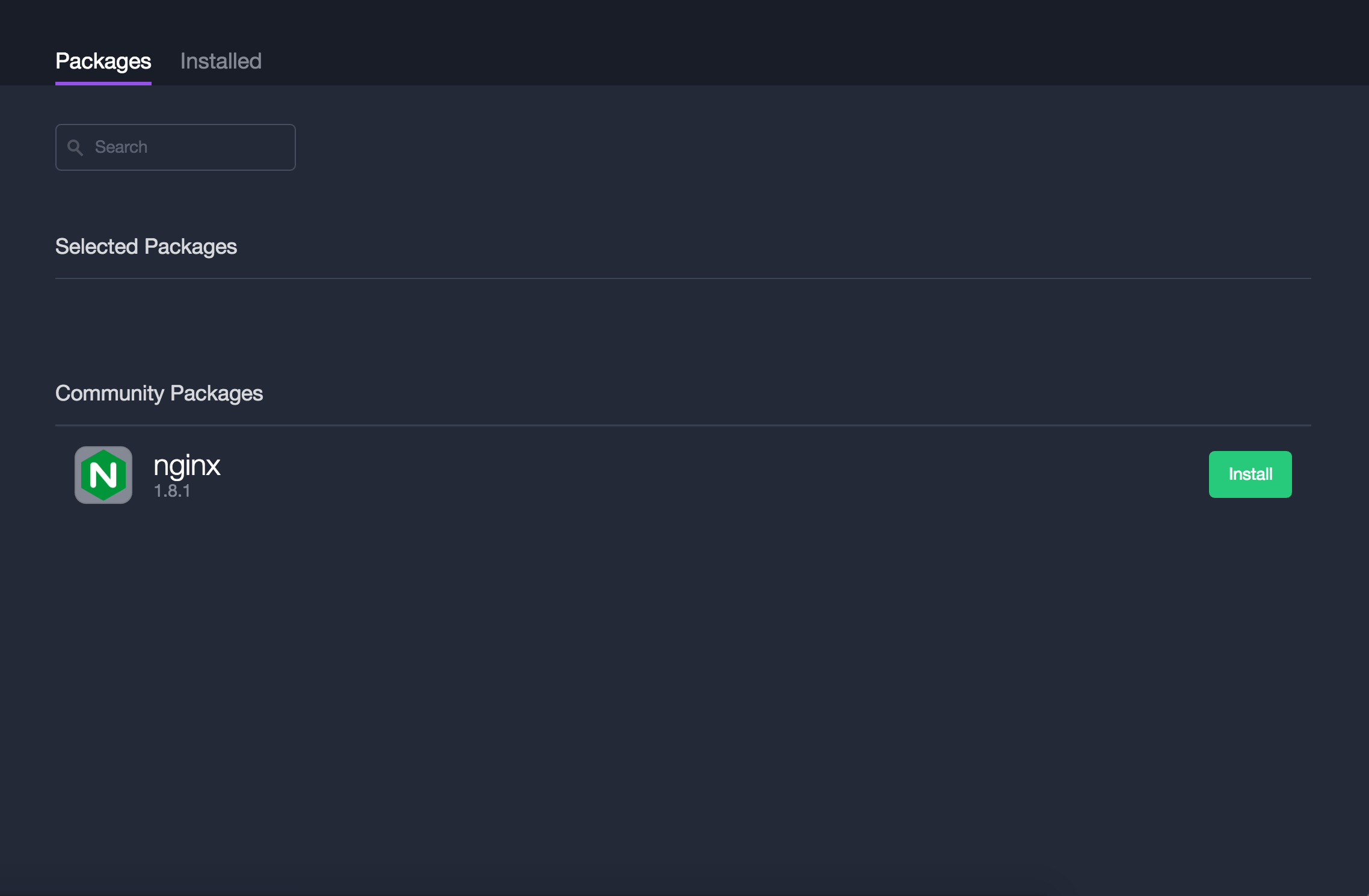
打开你的DC/UI, 点击System -> Overview -> Repositories 来开始添加你的repositories.



点击Add，你的仓库就添加成功了，当然，如果你有dcos CLI的话，也可以通过下面的命令添加仓库。

dcos package repo add Development http://your\_universe\_URI

然后，在Universe -> Packages页面里面，你就能看到自己制作的应用了。



作者简介：陈晖，供职于西安IBM, 主要从事云计算方面工作，涉及Openstack,Mesos等开源项目，在资源管理 ，资源调度等方面有比较丰富的经验 。