

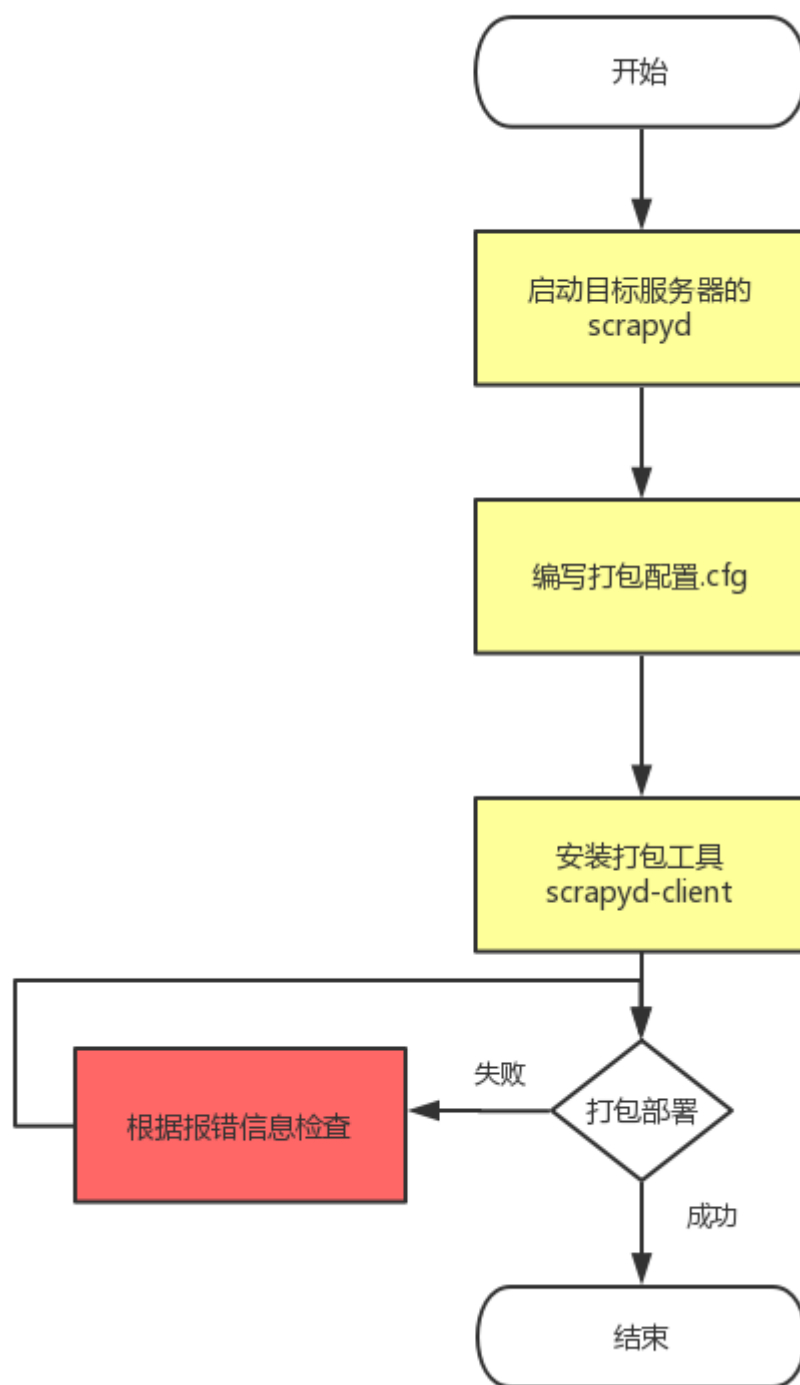
通过 Scrapyd-client 打包并部署爬虫

当爬虫代码编写完毕后，你可以选择直接运行启动文件来启动爬虫，也可以将爬虫部署到 Scrapyd 后，通过 Scrapyd 的 API 来启动爬虫。两种方法各自的优缺点以及应用场景会在后面的小节知识中讲解，这里我们先学会如何将爬虫项目打包并部署到 Scrapyd 上。

本小节将通过两个具体的部署例子（部署到本地以及部署到云服务器）以熟悉 Scrapy 爬虫项目打包、Scrapyd-client 的安装、使用以及爬虫项目部署过程。

爬虫项目打包

Scrapyd 打包部署的整个流程为：



打包前期

当你使用 Scrapy 框架编写完爬虫代码之后，你需要将项目进行打包，才能够将其部署到 Scrapyd 上。[官方文档](#)对项目的打包有介绍：

- 1 Deploying your project involves eggifying it and uploading the egg to Scrapyd via the
- 2 addversion.json endpoint. You can do this manually, but the easiest way is to use the scrapyd-
- 3 deploy tool provided by scrapyd-client which will do it all for you.

Scrapy 项目需要使用 [Scrapyd-client 工具](#) 进行打包。

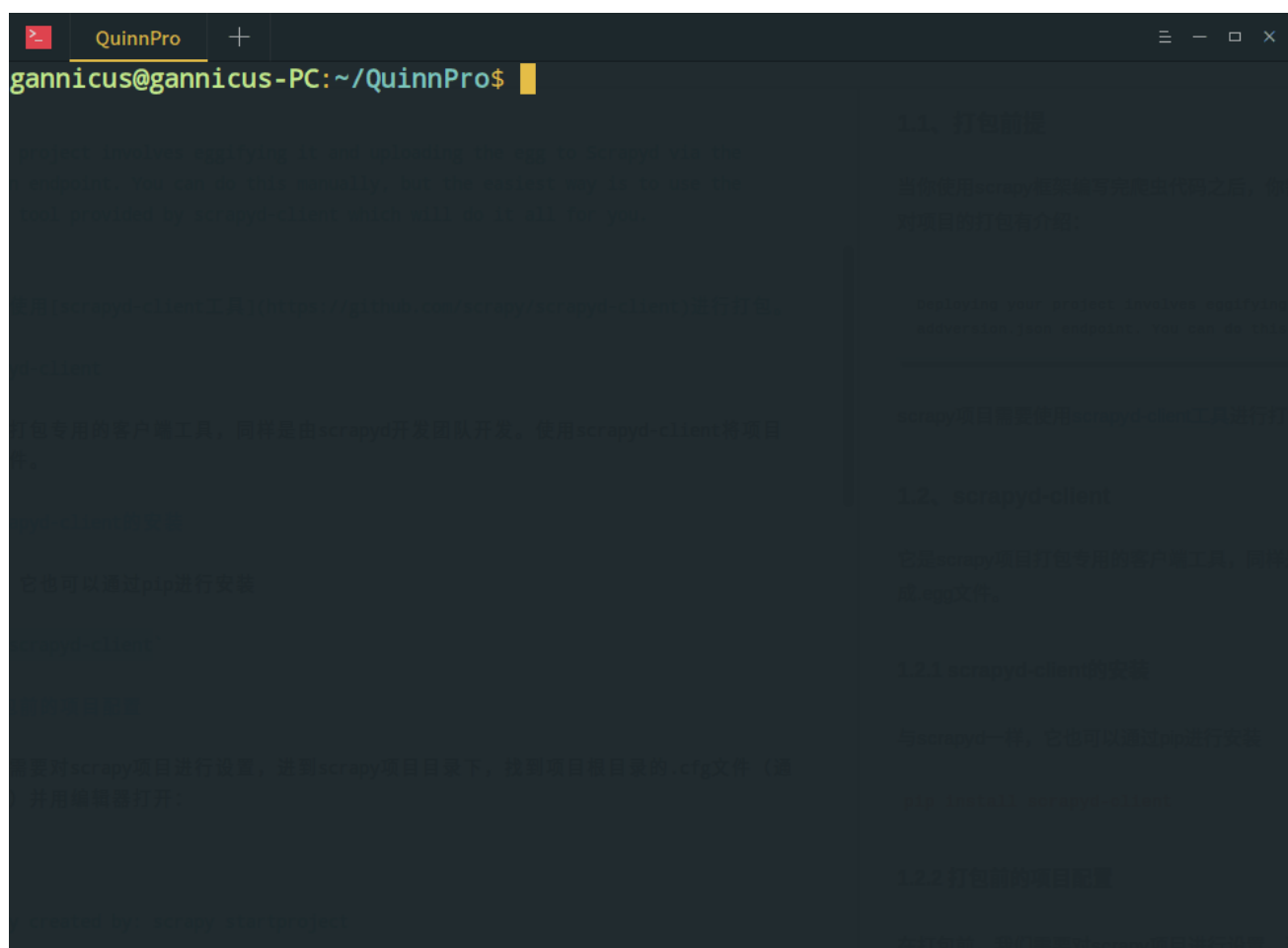
Scrapyd-client

它是 Scrapy 项目打包专用的客户端工具，同样是由 Scrapy 开发团队开发。使用 Scrapyd-client 将项目打包生成 `.egg` 文件。

Scrapyd-client 的安装

与 Scrapyd 一样，它也可以通过 pip 进行安装：

```
1 pip install scrapyd-client
2
```



打包前的项目配置

在打包前，我们需要对 Scrapy 项目进行设置。在 Scrapy 项目目录下，找到项目根目录的 `.cfg` 文件（通常是 `scrapy.cfg`）并用编辑器打开：

```
1 # Automatically created by: scrapy startproject
2 #
3 # For more information about the [deploy] section see:
4 # https://scrapyd.readthedocs.io/en/latest/deploy.html
5
6 [settings]
7 default = arts.settings
8
9 [deploy]
10 #url = http://localhost:6800/
11 project = arts
12
13
```

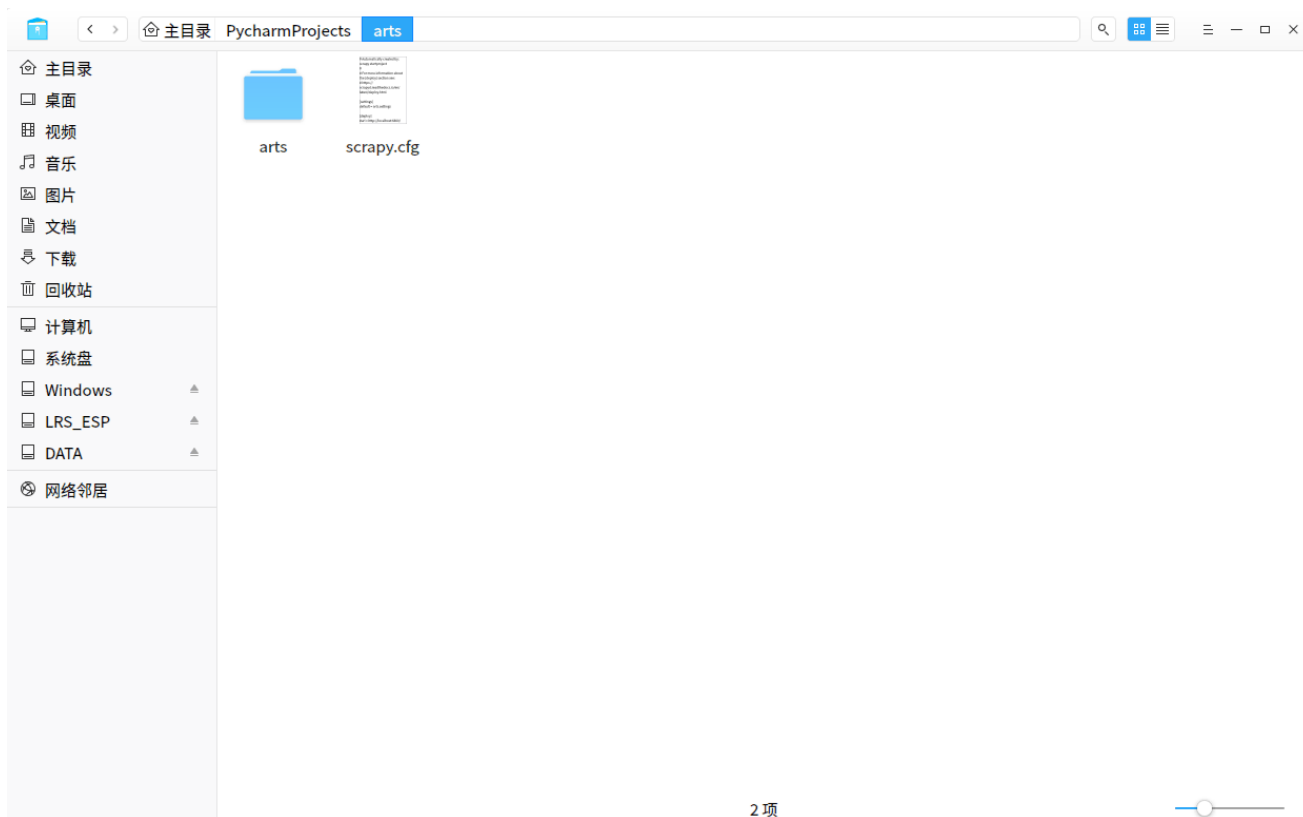
配置文件分为 Settings 级和 Deploy 级。Settings 中指定了项目所用的配置文件，而 Deploy 中指定项目打包的设置。

- URL - 指定部署的目标地址
- Project - 指定打包的项目
- Deploy - 指定项目别名

本小节，使用的项目为 `arts`，Scrapyd 服务为本地服务即 `localhost:6800`，所以这里以此作为基础进行演示。

可以看到 `.cfg` 文件中 URL 处默认是有注释的，这里将注释去掉，并且为项目添加别名 `locals`：

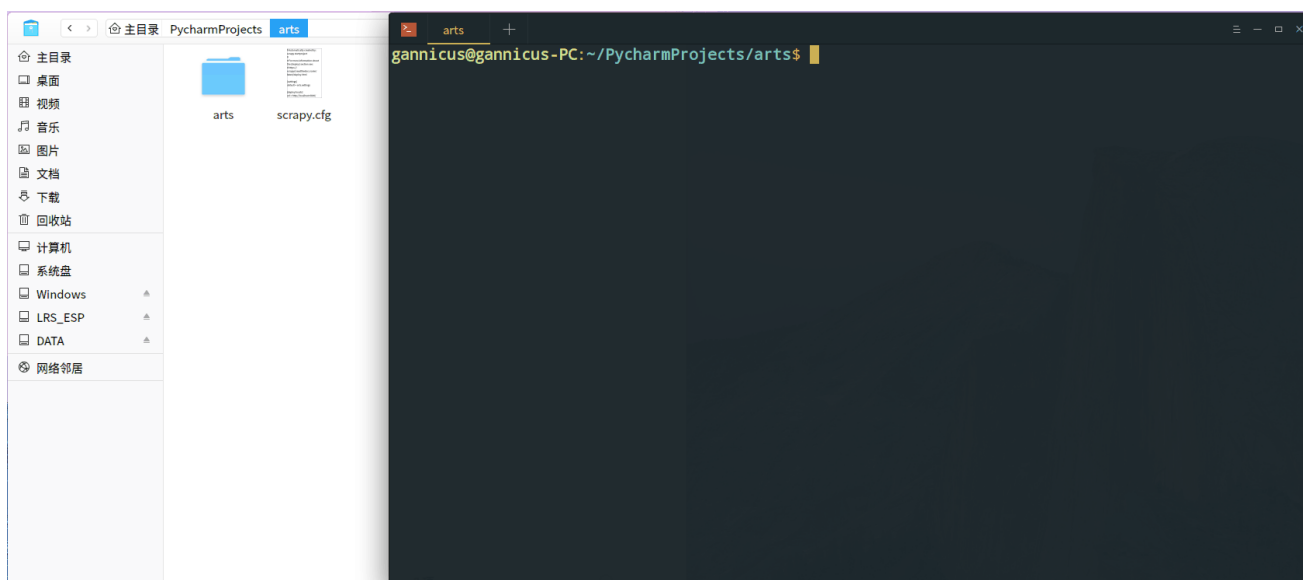
```
1 [settings]
2 default = arts.settings
3
4 [deploy:locals]
5 url = http://localhost:6800/
6 project = arts
7
```



打包部署

而后在 arts 项目的根目录(.cfg 同级目录)下使用命令(此时必须保证 Scrapy 服务是正常运行的):

```
1 scrapy-deploy locals -p arts
2
```



将项目打包并部署到指定的目标服务上, Scrapy 服务会将请求结果以 json 格式返回:

```
1 node-name:arts$ scrapyd-deploy locals -p arts
2 Packing version 1538645094
3 Deploying to project "arts" in http://localhost:6800/addversion.json
4 Server response (200):
5 {"node_name": "node-name", "status": "ok", "project": "arts", "version": "1538645094",
6  "spiders": 1}
```

返回信息中包含了此次打包的版本号、目标服务地址、nodeName、项目状态、项目名称以及其中所包含的爬虫数量。并且在 Web 界面上也可以看到项目 arts 的名称，如下图所示：



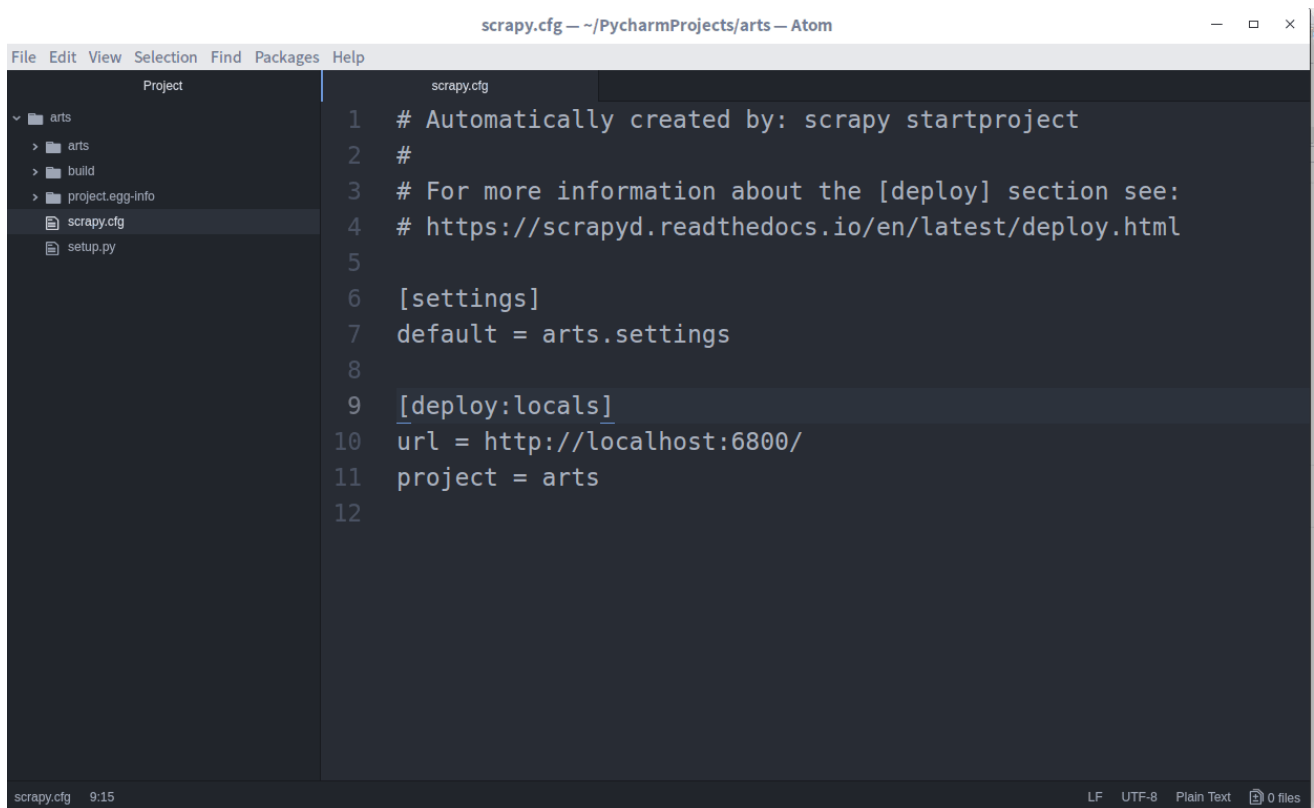
百度一下

思考题

scrapy.cfg 文件中 Deploy 级设置里，Deploy 的名称是必须设置的吗？如果不设置会怎么样？可以有多个 Deploy 级配置吗？

我们可以通过动手实验，来验证这些问题。

若 Deploy 不设置名称



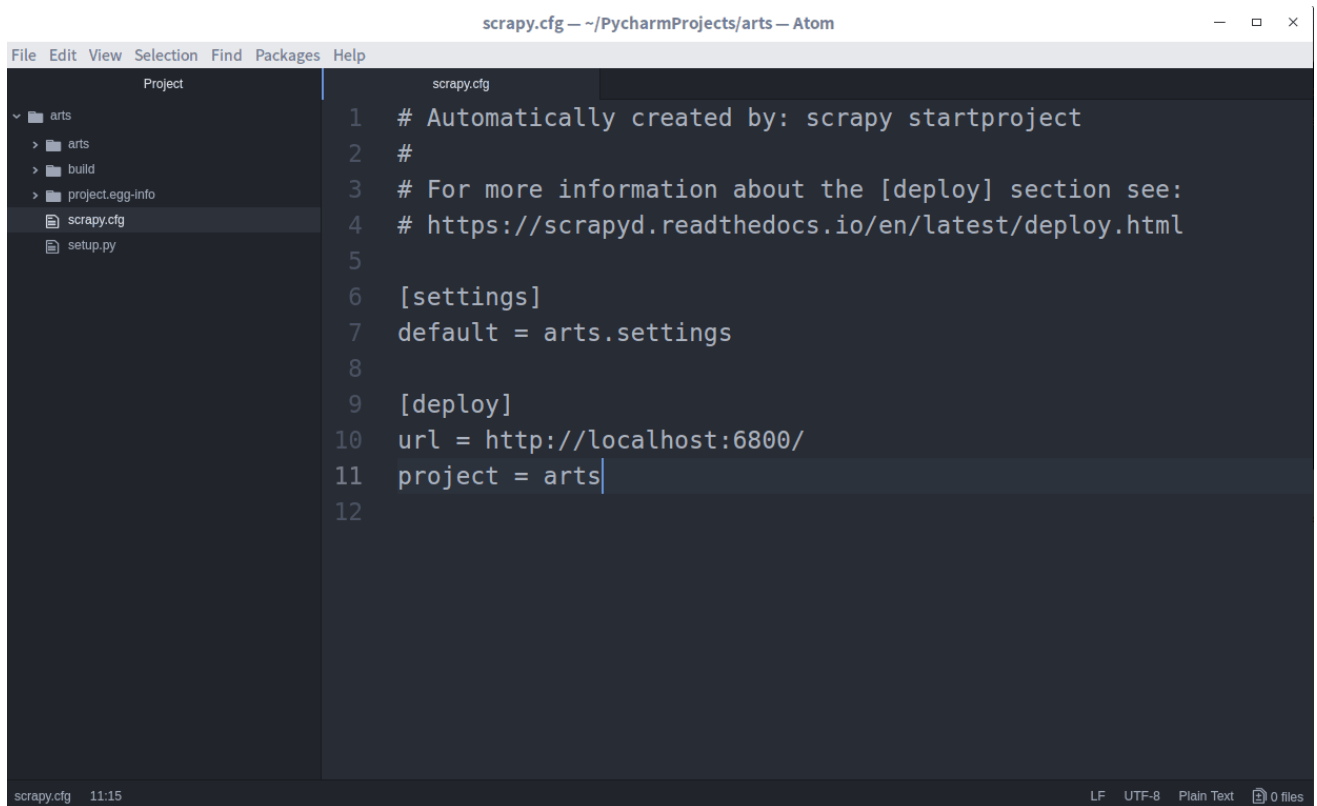
```
scrapy.cfg
1 # Automatically created by: scrapy startproject
2 #
3 # For more information about the [deploy] section see:
4 # https://scrapyd.readthedocs.io/en/latest/deploy.html
5
6 [settings]
7 default = arts.settings
8
9 [deploy:locals]
10 url = http://localhost:6800/
11 project = arts
12
```

可以看到，Deploy 级配置不设置名称的话，在命令行中也无需使用名称，同样可以完成项目的打包。

若多个 Deploy 配置

笔者在 192.168.0.61 服务器启动了 Scrapyd，并且在 `scrapy.cfg` 文件中设置两组 Deploy 级别配置，其中一个 Deploy 不设置名称且 URL 指向本地 Scrapyd；另一个 Deploy 设置名称为 `servers` 且 URL 指向服务器的 Scrapyd。 `cfg` 代码为：

```
1 [settings]
2 default = arts.settings
3
4 [deploy]
5 url = http://localhost:6800/
6 project = arts
7
8 [deploy:servers]
9 url = http://192.168.0.61:6800/
10 project = arts
11
```



```
scrapy.cfg
1 # Automatically created by: scrapy startproject
2 #
3 # For more information about the [deploy] section see:
4 # https://scrapyd.readthedocs.io/en/latest/deploy.html
5
6 [settings]
7 default = arts.settings
8
9 [deploy]
10 url = http://localhost:6800/
11 project = arts
12
```

可以看到，多个 Deploy 级别的配置是允许的，并且我们可以使用 Deploy 的名称来区分它们。

小结

本小节通过 Scrapy 项目的部署案例，我们学会了 Scrapyd-client 的安装、使用以及打包前 `.cfg` 配置文件的相关配置，并且成功的将一个 Scrapy 项目打包部署到目标服务器上。