

# 高效部署和监控分布式爬虫项目

## 应用场景

- 有 N 台云主机，通过 Scrapy-Redis 构建分布式爬虫
- 希望集成身份认证
- 希望在页面上直观地查看所有云主机的运行状态
- 希望能够**自由选择部分云主机，批量部署和运行爬虫项目，实现集群管理**
- 希望自动执行日志分析，以及爬虫进度可视化
- 希望在出现特定类型的异常日志时能够及时通知用户，包括自动停止当前爬虫任务

**优点：**能够通过浏览器直接部署和运行项目，能够查看日志

## 安装和配置

1、确保所有主机都已经安装和启动 Scrapy

2、如需远程访问 Scrapy，需将 Scrapy 配置文件中的 bind\_address 修改为：

```
bind_address = 0.0.0.0
```

3、开发主机安装 ScrapyWeb：

```
pip install scrapyweb
```

4、运行命令： scrapyweb -h，

**将在当前工作目录生成配置文件 scrapyweb\_settings.py**，可用于下文的自定义配置。

5、启用 HTTP 基本认证：

```
ENABLE_AUTH = True
```

```
USERNAME = 'username'
PASSWORD = 'password'
# The default is False, set i
ENABLE_AUTH = False
# In order to enable basic au
USERNAME = ''
PASSWORD = ''
```

6、添加 Scrapy server，支持字符串和元组两种配置格式，支持添加认证信息和分组/标

签：

```
SCRAPYD_SERVERS = [
    '127.0.0.1',
    # 'username:password@localhost:6801#group',
    ('username', 'password', 'localhost', '6801', 'group'),
]

SCRAPYD_SERVERS = [
    '127.0.0.1:6800',
    # 'username:password@localhost:6801#group',
    # ('username', 'password', 'localhost', '6801', 'group'),
]
```

7、配置 SCRAPY\_PROJECTS\_DIR 指定 Scrapy 项目开发目录：

```
# ScrapyWeb is able to locate projects in the SCRAPY_PROJECTS_D
# so that you can simply select a project to deploy, instead of
# e.g. 'C:/Users/username/myprojects' or '/home/username/myproje
SCRAPY_PROJECTS_DIR = r'D:\edu\python\code\JobSpider'
```

8、启动 ScrapyWeb

```
(erw1) D:\edu\python1904\20190807\code\JobSpider>scrapyweb
[2019-08-23 09:29:23,203] INFO in apscheduler.scheduler: Scheduler started
[2019-08-23 09:29:23,351] INFO in scrapyweb.run: Scrapyweb version: 1.4.0
[2019-08-23 09:29:23,351] INFO in scrapyweb.run: Use scrapyweb -h to get help
[2019-08-23 09:29:23,351] INFO in scrapyweb.run: Main pid: 11684
[2019-08-23 09:29:23,352] DEBUG in scrapyweb.run: Loading default settings from d:/python/virtualenv/erw1/lib/site-packages/scrapyweb/default_settings.py

*****
Overriding custom settings from D:/edu/python1904/20190807/code/JobSpider/scrapyweb_settings_v10.py
*****

[2019-08-23 09:29:23,661] DEBUG in scrapyweb.run: Reading settings from command line: Namespace(bind='0.0.0.0', debug=False, disable_auth=False, disable_logparser=False, disable_monitor=False, port=5000, scrapy_server=None, switch_scheduler_state=False, verbose=False)
[2019-08-23 09:29:23,662] DEBUG in scrapyweb.utils.check_app_config: Checking app config
[2019-08-23 09:29:23,664] INFO in scrapyweb.utils.check_app_config: Setting up URL_SCRAPYWEB: http://127.0.0.1:5000
[2019-08-23 09:29:23,665] DEBUG in scrapyweb.utils.check_app_config: Checking connectivity of SCRAPYD_SERVERS...

Index Group          Scrapy IP:Port      Connectivity Auth
=====
1 None               127.0.0.1:6800     True       None
=====

[2019-08-23 09:29:23,783] DEBUG in scrapyweb.utils.check_app_config: Created 1 tables for JobsView
[2019-08-23 09:29:23,783] INFO in scrapyweb.utils.check_app_config: Locating scrapy logfiles with SCRAPYD_LOG_EXTENSIONS: ['.log', '.log.gz', '.txt']
[2019-08-23 09:29:23,788] INFO in scrapyweb.utils.check_app_config: Scheduler for timer tasks: STATE_RUNNING
[2019-08-23 09:29:23,893] INFO in scrapyweb.utils.check_app_config: create_jobs_snapshot (trigger: interval[0:05:00], next run at: 2019-08-23 09:34:28 CST)

*****
Visit Scrapyweb at http://127.0.0.1:5000 or http://IP-OF-THE-CURRENT-HOST:5000
*****

[2019-08-23 09:29:23,906] INFO in scrapyweb.run: For running Flask in production, check out http://flask.pocoo.org/docs/1.0/deploying/
* Serving Flask app "scrapyweb" (lazy loading)
* Environment: production
  WARNING: Do not use the development server in a production environment.
  Use a production WSGI server instead.
* Debug mode: off
[2019-08-23 09:29:29,463] INFO in werkzeug: * Running on http://0.0.0.0:5000/ (Press CTRL+C to quit)
```

## 访问 Web UI

通过浏览器访问并登录 <http://127.0.0.1:5000>

- Overview 页面自动输出所有 Scrapy server 的运行状态
- 通过分组和过滤可以自由选择若干台 Scrapy server，调用 Scrapy 提供的所有

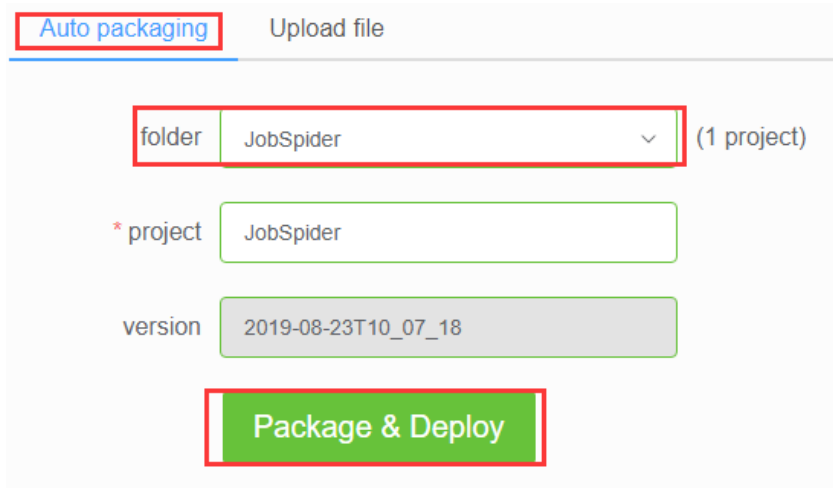
HTTP JSON API，实现一次操作，批量执行



## 部署项目

- 支持指定若干台 Scrapy server 部署项目
- 通过配置 SCRAPY\_PROJECTS\_DIR 指定 Scrapy 项目开发目录, *ScrapydWeb* 将

自动列出该路径下的所有项目, 选择项目后即可自动打包和部署指定项目:



The image shows a web interface for deploying a Scrapy project. It features two tabs: 'Auto packaging' (highlighted with a red box) and 'Upload file'. Below the tabs, there are three input fields: 'folder' (a dropdown menu showing 'JobSpider' with a red box around it), '\* project' (a text input field showing 'JobSpider' with a green box around it), and 'version' (a text input field showing '2019-08-23T10\_07\_18' with a green box around it). To the right of the 'folder' dropdown, it says '(1 project)'. At the bottom, there is a green button labeled 'Package & Deploy' with a red box around it.

## 运行爬虫

- 通过下拉框直接选择 project, version 和 spider
- 支持传入 Scrapy settings 和 spider arguments
- 同样支持指定若干台 Scrapy server 运行爬虫



The image shows a web interface for configuring a Scrapy server. It includes several dropdown menus for 'Scrapy server' (127.0.0.1:6800), 'project' (JobSpider), '\_version' (2019-08-23T10\_07\_18), and 'spider' (pythonPosition). A 'timer task' toggle switch is highlighted with a red box and an arrow pointing to it with the text '可实现定时任务' (Can achieve scheduled tasks). Below this is a 'curl command' text area containing a command to schedule a job. At the bottom are two buttons: 'Check CMD' and 'Run Spider'.

Scrapy server: 127.0.0.1:6800

\* project: JobSpider

\* \_version: 2019-08-23T10\_07\_18

\* spider: pythonPosition

settings & arguments: ☐

timer task: ☐ 可实现定时任务

\* curl command: 

```
curl http://127.0.0.1:6800/schedule.json \
-d project=JobSpider \
-d _version=2019-08-23T10_07_18 \
-d spider=pythonPosition \
-d jobid=2019-08-23T10_10_12
```

Check CMD Run Spider

可以控制爬虫任务的执行，暂停等



The image shows a web interface for managing Scrapy jobs. It includes a table with columns for 'id', 'project', 'spider', 'jobid', 'status', 'start', 'end', 'update time', and 'delete'. There are two rows of data. The first row has a status of 'Running' and the second row has a status of 'Completed'.

Get the list of jobs of all projects in database. Classic

'pip install logparser' on host '127.0.0.1:6800' and run command 'logparser' to show crawled\_pages and scraped\_items.

id	project	spider	jobid	status	start	end	update time	delete
1	JobSpider	pythonPosition	2019-08-23T10_14_11	Running			2019-08-23 10:14:11	
2	JobSpider	pythonPosition	2019-08-23T10_14_11	Completed			2019-08-23 10:14:11	

## 日志分析和可视化

默认情况下，*ScrapydWeb* 将在后台定时自动读取和分析 *Scrapy log* 文件并生成

**Stats** 页面

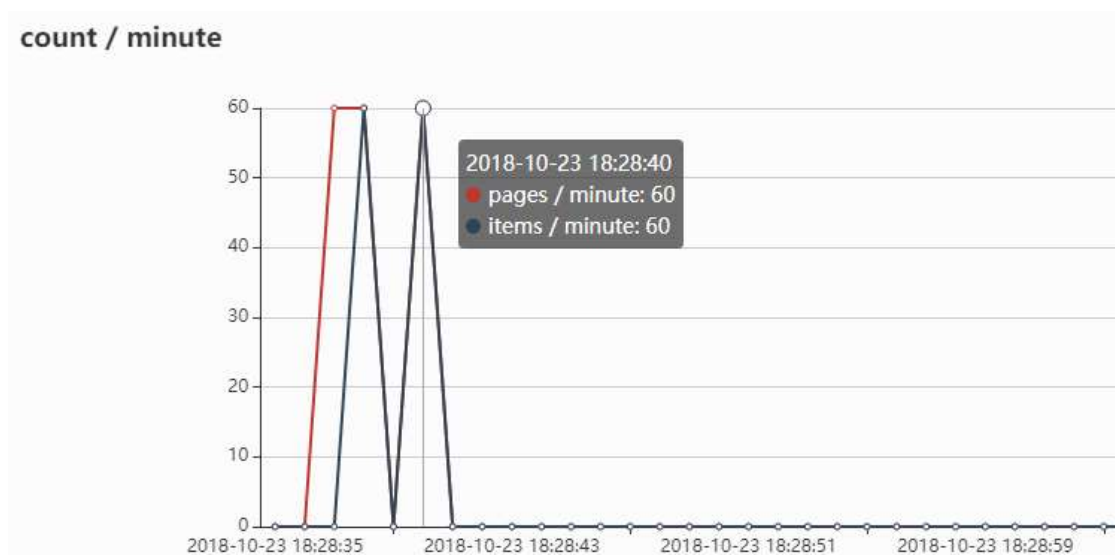
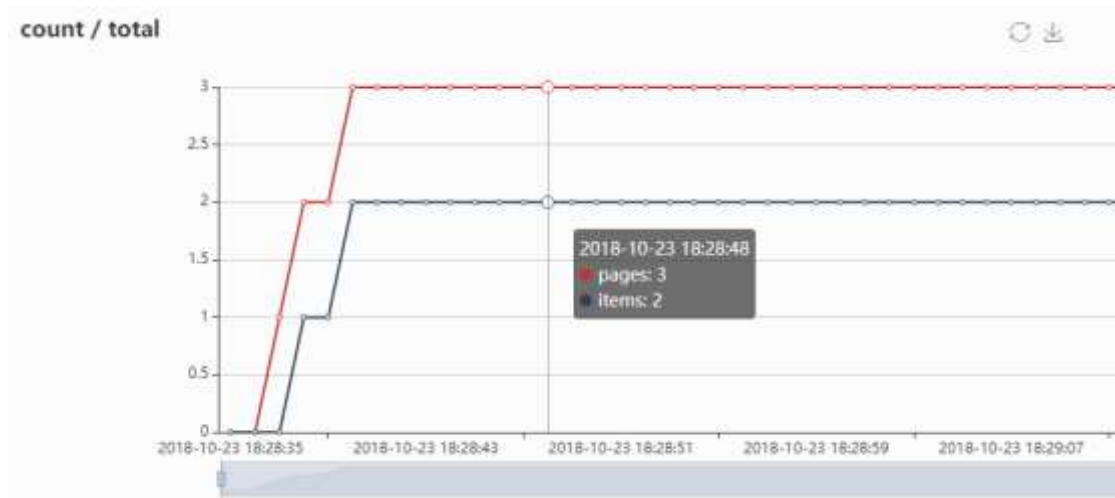
## Stats 页面

### PROJECT (JobSpider), SPIDER (pythonPosition)

[Log analysis](#)[Log categorization](#)[View log](#)

project	JobSpider
spider	pythonPosition
job	2019-08-23T10_14_17
first_log_time	2019-08-23 10:14:45
latest_log_time	2019-08-23 10:15:09
runtime	0:00:24
crawled_pages	0
scraped_items	0
shutdown_reason	N/A
finish_reason	N/A
log_critical_count	0
log_error_count	0
log_warning_count	0
log_redirect_count	0
log_retry_count	0
log_ignore_count	0
latest_crawl	2 minutes ago
latest_scrape	2 minutes ago
latest_log	2 minutes ago
current_time	Fri Aug 23 2019 10:17:58 GMT+0800 (中国标准时间)
latest_item	N/A

## 爬虫进度可视化



## 日志分析

Log analysis	Log categorization	Progress visualization	<a href="#">View log</a>	C
Head				
<pre>2018-10-23 18:28:34 [scrapy.utils.log] INFO: Scrapy 1.5.0 started (bot: demo 2018-10-23 18:28:34 [scrapy.utils.log] INFO: Versions: lxml 4.2.1.0, libxml: 13:32:41) [MSC v.1900 64 bit (AMD64)], pyOpenSSL 17.5.0 (OpenSSL 1.0.2o 27 2018-10-23 18:28:34 [scrapy.crawler] INFO: Overridden settings: {'BOT_NAME' 'file:///C:/Users/win7/items/demo/test/2018-10-23_182826.jl', 'LOGSTATS_INTI ['demo.spiders'], 'USER_AGENT': 'Mozilla/5.0'} 2018-10-23 18:28:34 [scrapy.middleware] INFO: Enabled extensions: ['scrapy.extensions.corestats.CoreStats', 'scrapy.extensions.telnet.TelnetConsole', 'scrapy.extensions.feedexport.FeedExporter', 'scrapy.extensions.logstats.LogStats'] 2018-10-23 18:28:35 [scrapy.middleware] INFO: Enabled downloader middlewares: ['scrapy.downloadermiddlewares.httpauth.HttpAuthMiddleware', 'scrapy.downloadermiddlewares.downloadtimeout.DownloadTimeoutMiddleware', 'scrapy.downloadermiddlewares.defaultheaders.DefaultHeadersMiddleware', 'scrapy.downloadermiddlewares.useragent.UserAgentMiddleware', 'scrapy.downloadermiddlewares.retry.RetryMiddleware', 'scrapy.downloadermiddlewares.redirect.MetaRefreshMiddleware', 'scrapy.downloadermiddlewares.httpcompression.HttpCompressionMiddleware', 'scrapy.downloadermiddlewares.redirect.RedirectMiddleware', 'scrapy.downloadermiddlewares.httpproxy.HttpProxyMiddleware', 'scrapy.downloadermiddlewares.stats.DownloaderStats'] 2018-10-23 18:28:35 [scrapy.middleware] INFO: Enabled spider middlewares: ['scrapy.spidermiddlewares.httperror.HttpErrorMiddleware', 'scrapy.spidermiddlewares.offsite.OffsiteMiddleware', 'scrapy.spidermiddlewares.referer.RefererMiddleware', 'scrapy.spidermiddlewares.urllength.UrlLengthMiddleware', 'scrapy.spidermiddlewares.depth.DepthMiddleware']</pre>				



```
{'city': '北京-海淀区',
 'corp': '慧影医疗科技(北京)有限公司',
 'name': 'Python开发工程师',
 'pub_date': '08-23',
 'salary': '2-4万'}
2019-08-23 10:14:46 [scrapy.dupefilters] DEBUG: Filtered duplicate request: <GET https://search.51job.com/
lang=c&stype=1&postchannel=0000&workyear=99&cotype=99&degreefrom=99&jobterm=99&companysize=99&lonlat=0%2C
- no more duplicates will be shown (see DUPEFILTER_DEBUG to show all duplicates)
2019-08-23 10:14:46 [pythonPosition] DEBUG: csv path:D:\edu\python1904\scrapyd_test
2019-08-23 10:14:46 [pythonPosition] DEBUG: csv path:D:\edu\python1904\scrapyd_test
2019-08-23 10:14:46 [scrapy.core.scrapers] DEBUG: Scraped from <200 https://search.51job.com/list/010000,0
lang=c&stype=1&postchannel=0000&workyear=99&cotype=99&degreefrom=99&jobterm=99&companysize=99&lonlat=0%2C

{'city': '北京-大兴区',
 'corp': '达内时代科技集团有限公司',
 'name': '少儿编程讲师(python)',
 'pub_date': '08-23',
 'salary': '0.8-1.3万'}
2019-08-23 10:14:46 [pythonPosition] DEBUG: csv path:D:\edu\python1904\scrapyd_test
2019-08-23 10:14:46 [pythonPosition] DEBUG: csv path:D:\edu\python1904\scrapyd_test
2019-08-23 10:14:46 [scrapy.core.scrapers] DEBUG: Scraped from <200 https://search.51job.com/list/010000,0
lang=c&stype=1&postchannel=0000&workyear=99&cotype=99&degreefrom=99&jobterm=99&companysize=99&lonlat=0%2C
```

## 邮件通知

基于后台定时读取和分析 Scrapy log 文件, *ScrapydWeb* 将在满足特定触发器时发送通知邮件, 邮件正文包含当前运行任务的统计信息。

### 1、添加邮箱帐号:

```
SMTP_SERVER = 'smtp.qq.com'
SMTP_PORT = 465
SMTP_OVER_SSL = True
SMTP_CONNECTION_TIMEOUT = 10
FROM_ADDR = 'username@qq.com'
EMAIL_PASSWORD = 'password'
TO_ADDRS = ['username@qq.com']
```

### 2、设置邮件工作时间和基本触发器, 以下示例代表: 每隔 1 小时或某一任务完成时, 并且

当前时间是工作日的 9 点, 12 点和 17 点, *ScrapydWeb* 将会发送通知邮件。

```
EMAIL_WORKING_DAYS = [1, 2, 3, 4, 5]
EMAIL_WORKING_HOURS = [9, 12, 17]
ON_JOB_RUNNING_INTERVAL = 3600
ON_JOB_FINISHED = True
```

### 3、除了基本触发器, *ScrapydWeb* 还提供了多种触发器用于处理不同类型的 log, 包括

'CRITICAL', 'ERROR', 'WARNING', 'REDIRECT', 'RETRY' 和 'IGNORE'等。

```
LOG_CRITICAL_THRESHOLD = 3
LOG_CRITICAL_TRIGGER_STOP = True
LOG_CRITICAL_TRIGGER_FORCESTOP = False
LOG_IGNORE_TRIGGER_FORCESTOP = False
```

以上示例表示：当发现 3 条或 3 条以上的 critical 级别的 log 时，*ScrapyWeb* **自动停**

**止当前任务**，如果当前时间在邮件工作时间内，则同时发送通知邮件。