

UpCloud Tutorial

一、前言

感谢使用 **upcloud**，这是我的又拍云开发者大赛参赛作品。如果你在使用过程中发现BUG，欢迎在 [Github](#) 或 [gitcafe](#) 上给我提issue。如果你对这个工具有更好的建议和想法，也非常欢迎fork。

英语写作还需学习，就不整英文文档了。目前中文能让我更好的介绍这个工具的特性和使用方法。如果你觉得自己的英文水平不错，希望你能抽出一点宝贵的时间帮助我翻译这篇使用说明，感激不尽！

二、安装

第一次安装，在终端执行：

```
sudo pip install upcloud
```

升级你的程序版本：

```
sudo pip install upcloud --upgrade
```

注意：新版会更正BUG，完善帮助文档，建议升级你的程序到最新版本获得更好的体验。

安装完成后使用 `upcloud -v` 查看当前安装版本

```
→ ~ > upcloud -v
upcloud 0.1.3

→ ~ > upcloud -h
usage: upcloud [-h] [-v] -b BUCKET -u USERNAME [-p] [-t TIMEOUT]
```

```
[-e {auto,telecom,cnc,ctt}]
```

Remote terminal management client for UpYun !

optional arguments:

```
-h, --help            show this help message and exit
-v, --version          show program's version number and exit
-b BUCKET, --bucket BUCKET
                        The name of your bucket (default: None)
-u USERNAME, --username USERNAME
                        The operator's name of this bucket (default: None)
-p, --passwd           The operator's password of this bucket (default:
                        False)
-t TIMEOUT, --timeout TIMEOUT
                        The HTTP request timeout time (default: 30)
-e {auto,telecom,cnc,ctt}, --endpoint {auto,telecom,cnc,ctt}
                        The network access point (default: auto)
```

三、使用

1. 登陆

```
→ ~ > upcloud -b kehrspace -u test01 -p
Password for test01:
=====
+ Bucketname: kehrspace
+ Username: test01
+ Timeout: 30
+ EndPoint: v0.api.upyun.com
+ Workspace: /
=====
login ...

+ Space usage: 31.45MB

login sucess ! Have a nice day !

Welcome to use upcloud ! version: 0.1.3

You can use this tool to manage your remote space easily. Enjoy it !

Type "man" show this message again.
Type "help" or "?" for help.
Type "-h" or "--help" behind a command for help.
Type "![command]" or "shell [command]" run a shell command. example: !ls
Type "bash" enter the local bash environment.
```

```
Type "cls" or "clear" to clear the terminal screen.  
Type double <Tab> key to get a command list.
```

```
test01@kehrspace >
```

程序的启动命令是 `upcloud` , 参数: `-h` , `-v` , `-b` , `-u` , `-p` , `-t` , `-e`

- `-h` , 使用帮助
- `-v` , 显示程序版本
- `-b` , 又拍云空间名称【必填】
- `-u` , 空间管理员 (操作员) 姓名【必填】
- `-p` , 后无参数, 回车后输入管理员 (操作员) 密码, 输入时密码不会明文显示【必填】
- `-t` , 连接又拍云的HTTP请求超时时间, 默认是30s
- `-e` , 连接线路, 可选: `auto` 自动, `telecom` 电信, `cnc` 联通网通, `ctt` 移动铁通, 默认是 `auto`

2. 获取帮助

程序提供两种获取帮助的方式:

- 获取详细使用说明: `man [command]` 或者 `help [command]` 或者 `? [command]`
- 获取简单使用说明: `command -h|--help`

详细说明中会附带该命令的使用例子

```
test01@kehrspace > ls -h  
usage: ls [-h] [-l] [-d] [-r] [path [path ...]]  
  
positional arguments:  
  path                the directory path that you want to see (default: /)  
  
optional arguments:  
  -h, --help          show this help message and exit  
  -l, --long          use a long listing format (default: False)  
  -d, --directory     list directory entries instead of contents (default: False)  
  -r, --refresh       refresh the file list of current workspace (default: False)  
Type "ls -h" for help.  
test01@kehrspace > man ls  
NAME  
    ls - list directory content  
  
SYNOPSIS  
    ls [OPTIONS] [PATH ...]  
  
DESCRIPTION  
    1. List the information about the FILES, The default directory is current  
    working directory.  
       example: ls  
    2. -d, list directory entries instead of contents.
```

```
example: ls -d dir1 dir2 ...
If you want to get more information about the directory,
type: ls -ld dir1, dir2 ...
3. -l, Use a long listing format.show file's detail information.
example: ls -l dir1 dir2 ...
4. -r, Refresh the file list of current working directory.
example: ls -r

test01@kehrspace >
```

直接输入 **help** 或 **?** 会得到所有命令的名称列表

```
test01@kehrspace > ?

All commands you can use (type help <command> get more info):
=====
bash cd      exit help man   put  quit shell
cat  clear  get  ls   mkdir pwd  rm   usage

All alias command:
=====
cls ll q
```

3. 切换目录

- 切换当前工作目录使用命令：**cd**
- cd 后不加路径默认回到根目录“/”
- 支持相对路径和绝对路径
- 你可以使用 cd .. 返回上一级目录

```
test01@kehrspace > ll
/:
2014-05-10 15:14:26 <dir>      test
2014-04-16 21:31:54 <dir>      data
2014-04-17 16:03:27 <dir>      imgs
2014-05-01 19:49:31 <dir>      tmp
2014-05-04 17:21:00 <dir>      Users
2014-04-15 16:35:54 <dir>      img

6 directories, 0 files
test01@kehrspace > cd img
/img/

test01@kehrspace > cd ../imgs
/imgs/

test01@kehrspace > cd
/
```

所有涉及到云空间路径的地方，都支持相对路径和绝对路径。路径处理函数只能处理常用的路径，功能较弱，但是基本满足日常操作需要。

4. 浏览文件

浏览文件或目录信息，使用命令：`ls` 或者 `ll`

命令：`ll` 等价于 `ls -l`

```
test01@kehrspace > ls -h
usage: ls [-h] [-l] [-d] [-r] [path [path ...]]

positional arguments:
  path                the directory path that you want to see (default: /)

optional arguments:
  -h, --help          show this help message and exit
  -l, --long          use a long listing format (default: False)
  -d, --directory    list directory entries instead of contents (default: False)
  -r, --refresh      refresh the file list of current workspace (default: False)
Type "ls -h" for help.
```

`ls` 命令的使用方法和 Linux 的 `ls` 命令一样，但只支持参数 `-l` 和 `-d`，并增加了参数 `-r`

- `-l`，列出文件详细信息，如果是目录，则列出目录中的文件详细信息。
- `-d`，只显示目录信息，而不显示目录内容，需要配合 `-l` 参数才能获取详细信息
- `-r`，刷新当前目录的文件列表

```
test01@kehrspace > ls
/:
Users data img imgs tmp

5 directories, 0 files
test01@kehrspace > ls -l
/:
2014-04-16 21:31:54 <dir>          data
2014-04-17 16:03:27 <dir>          imgs
2014-05-01 19:49:31 <dir>          tmp
2014-05-04 17:21:00 <dir>          Users
2014-04-15 16:35:54 <dir>          img

5 directories, 0 files
test01@kehrspace > ls -l img
/img/:
2014-04-15 16:36:02 <file> 1003.50KB terminal.png
2014-04-16 21:35:59 <file> 151.96KB  domain.png

0 directories, 2 files
test01@kehrspace > ls -ld img
2014-04-15 16:35:54 <dir>          img
```

```
test01@kehrspace >
```

5. 创建目录

创建目录使用命令：`mkdir`

```
test01@kehrspace > mkdir -h
usage: mkdir [-h] [-p] path [path ...]

positional arguments:
  path                Create one or more directories

optional arguments:
  -h, --help          show this help message and exit
  -p, --parents        make parent directories as needed (default: False)
```

- `mkdir` 可以同时创建多个目录，和多级目录
- `-p`，创建多级目录

```
test01@kehrspace > mkdir dir1 dir2
creating directory /dir1/ ...
creating directory /dir2/ ...
All the files to create success!
test01@kehrspace > mkdir -p a/b/c/d/e
recursive create the directory: /a/b/c/d/e/
creating directory /a/ ...
creating directory /a/b/ ...
creating directory /a/b/c/ ...
creating directory /a/b/c/d/ ...
creating directory /a/b/c/d/e/ ...
All the files to create success!
test01@kehrspace > ls
/:
Users a data dir1 dir2 img imgs tmp

8 directories, 0 files
test01@kehrspace >
```

6. 删除文件

删除文件或目录使用命令：`rm`

```
test01@kehrspace > rm -h
usage: rm [-h] [-r] path [path ...]

positional arguments:
```

```
path                remove one or more files or directories

optional arguments:
  -h, --help          show this help message and exit
  -r, --recursive     remove directories and their contents recursively (default:
False)
```

- 可以同时删除多个文件和目录
- -r, 删除非空目录, 删除前会列出每个目录中的文件

```
test01@kehrspace > rm dir1 dir2
removing file /dir1 ...
removing file /dir2 ...
All files deleted successfully!
test01@kehrspace > rm -r a
Recursive delete the directory: /a/
+++++
/a/:
2014-05-10 14:37:52 <dir>          b

1 directories, 0 files
removing /a/:
+++++
/a/b/:
2014-05-10 14:37:52 <dir>          c

1 directories, 0 files
removing /a/b/:
+++++
/a/b/c/:
2014-05-10 14:37:46 <dir>          d

1 directories, 0 files
removing /a/b/c/:
+++++
/a/b/c/d/:
2014-05-10 14:37:52 <dir>          e

1 directories, 0 files
removing /a/b/c/d/:
+++++
/a/b/c/d/e/:

0 directories, 0 files
sucess!
removing file /a/b/c/d/e ...
sucess!
removing file /a/b/c/d ...
sucess!
removing file /a/b/c ...
sucess!
removing file /a/b ...
```

```
sucess!  
removing file /a ...  
All files deleted successfully!  
test01@kehrspace >
```

7. 空间信息

查看空间的使用情况，使用命令：`usage`

```
test01@kehrspace > usage  
Your space has been used: 31.45MB
```

8. 查看文件内容

在线查看文件内容，使用命令：`cat`

```
test01@kehrspace > cat -h  
usage: cat [-h] [-n NUMBER | -t TAIL] FILE  
  
positional arguments:  
  FILE                  The file path that you want to see  
  
optional arguments:  
  -h, --help            show this help message and exit  
  -n NUMBER, --number NUMBER  
                        Get first N rows content file (default: None)  
  -t TAIL, --tail TAIL  N lines after access to files (default: None)
```

- 默认显示所有文件内容

```
cat file1 #显示文件所有内容
```

- `-n`，指定显示文件的前n行

```
cat -n 10 file1 #显示文件前10行
```

- `-t`，指定显示文件的后n行

```
cat -t 5 file1 #显示文件后5行
```

例如：

```
test01@kehrspace > cat -n 10 __init__.py
```



```

1 import os
2
3 # IGNORE_FILES is a list of strings, which describe the file names to
be
4 # ignored.
5 #     i.e. file.name
6 IGNORE_FILES = ['__init__.py', 'Test_Temp.py']
7
8 __all__ = []
9 for root, dirs, files in os.walk(os.getcwd()):
10     for file in IGNORE_FILES:

```

test01@kehrspace >

注意： **cat** 的实现原理是缓存文件到本地。查看文件第一次时联网获取数据，以后的查看直接从本地缓存中读取。建议不要打开比较大的文件，受网络传输速度的限制，可能需要等待较长时间。

9. 上传文件

上传文件使用命令：**put**

```

test01@kehrspace > put -h
usage: put [-h] -s SOURCE [SOURCE ...] [-d DESTINATION] [-l LEVEL]

optional arguments:
  -h, --help            show this help message and exit
  -s SOURCE [SOURCE ...], --source SOURCE [SOURCE ...]
                        The file path of your local system (default: None)
  -d DESTINATION, --destination DESTINATION
                        The file path of your bucket<UpYun space> (default: /)
  -l LEVEL, --level LEVEL
                        remove The file path level(Support negative number),
                        begin with local path "/". save the name of files by
                        default (default: 0)

```

支持文件/文件夹批量上传，保留原目录结构，自定义目录级别

- **-s**，指定本地待上传的文件或文件夹路径，路径的写法和本地文件操作一样，不受特殊限制【必填】
- **-d**，指定存放文件的云空间位置。上传的文件/文件夹将会保持其原目录结构，放在指定目录下，默认是根目录“/”。
- **-l**，指定上传后保存原目录结构的级别，默认为0，保存原目录结构（从根目录开始），如果设置为-1，则只保存文件名。忽略空目录。

```
test01@kehrspace > ls #查看当前云空间工作目录下的文件
```

```

/: #当前查看的目录
Users data img imgs tmp

5 directories, 0 files
test01@kehrspace > !ls #查看当前本地目录下的文件（运行shell命令参考第11节）
cli.py
cli.pyc
color.py
color.pyc
__init__.py
__init__.pyc
main.py
manual.py
manual.pyc
upcloud.py
upcloud.pyc

test01@kehrspace > put -s cli.pyc -d /test #将本地文件上传到/test目录下，保留原目录结构
level: 0
from: /home/kehr/Github/upcloud/upcloud/cli.pyc #文件本地路径
to: /test/home/kehr/Github/upcloud/upcloud/cli.pyc #文件云空间路径
Uploading file ...
All files uploaded successfully!
count: 0 directories, 1 files
test01@kehrspace > put -s cli.pyc -d /test -l -1 #将本地文件上传到/test目录下，只保留文件名
level: -1
from: /home/kehr/Github/upcloud/upcloud/cli.pyc #文件本地路径
to: /test/cli.pyc #文件云空间路径
Uploading file ...
All files uploaded successfully!
count: 0 directories, 1 files
test01@kehrspace > ll #查看云空间文件，自动创建目录test
/:
2014-05-10 15:14:26 <dir> test
2014-04-16 21:31:54 <dir> data
2014-04-17 16:03:27 <dir> imgs
2014-05-01 19:49:31 <dir> tmp
2014-05-04 17:21:00 <dir> Users
2014-04-15 16:35:54 <dir> img

6 directories, 0 files
test01@kehrspace > ls -l test #确认文件是否上传成功
/test/:
2014-05-10 15:14:43 <file> 25.41KB cli.pyc
2014-05-10 15:14:31 <dir> home

1 directories, 1 files
test01@kehrspace >

```

关于 `-l` 参数的说明：

注意：-l 参数支持负数，指定目录级别。当值为负数时，意味着从后向前数。

假设本地有待上传文件，绝对路径为 /home/kehr/dev/file.md，需要上传到云空间的 /test 目录下。

1. file.md 上传到 '/test' 目录下后，保留目录结构为 /test/home/kehr/dev/file.md

```
put -s /home/kehr/dev/file.md -d /test
```

2. file.md 上传到 '/test' 目录下后，保留目录结构为 /test/dev/file.md

```
put -s /home/kehr/dev/file.md -d /test -l 2
#或者
put -s /home/kehr/dev/file.md -d /test -l -2
```

3. file.md 上传到 '/test' 目录下后，保留目录结构为 /test/file.md，即只保留文件名

```
put -s /home/kehr/dev/file.md -d /test -l -1
```

10. 下载文件

下载文件使用命令：`get`

```
test01@kehrspace > get -h
usage: get [-h] -s SOURCE [SOURCE ...] [-d DESTINATION] [-l LEVEL]

optional arguments:
  -h, --help                show this help message and exit
  -s SOURCE [SOURCE ...], --source SOURCE [SOURCE ...]
                           The file path of your bucket<UpYun space> (default:
                           None)
  -d DESTINATION, --destination DESTINATION
                           The file path of your local system (default: .)
  -l LEVEL, --level LEVEL
                           Remove The file path level(Support negative
                           number),begin with bucket path "/". save the name of
                           files by default (default: 0)
```

支持文件/文件夹批量下载，保留原目录结构，自定义目录级别

- -s，指定待下载的云空间文件或文件夹路径，路径尽量使用常用的相对路径或绝对路径【必填】
- -d，指定存放文件的本地目录位置。下载的文件/文件夹将会保持其原目录结构，放在指定目录下，默认是程序运行的当前目录（可以通过命令 `!cd PATH` 修改，使用shell命令参考第11小结）。

- `-l`，指定下载后保存原目录结构的级别，默认为0，保存原目录结构（从根目录开始），如果设置为-1，则只保存文件名。忽略空目录。

`get` 的使用方法和 `put` 类似，可以参考上一节的例子熟悉 `get` 命令的使用方法。

11. shell环境

使用命令：`bash`

或者：`![command]`

为了方便在本地shell环境和云空间的shell环境之间切换，程序添加了本地shell的入口。你可以使用命令 `bash` 进入本地shell环境。

```
test01@kehrspace > ls          #查看云空间文件
/:
Users data img imgs test tmp

6 directories, 0 files
test01@kehrspace > pwd          #查看云空间的工作路径
/

test01@kehrspace > !pwd         #查看程序的本地工作目录
/home/kehr/Github/upcloud/upcloud

test01@kehrspace > bash        #进入本地shell环境
→ upcloud git:(master) > pwd    #查看当前工作目录
/home/kehr/Github/upcloud/upcloud
→ upcloud git:(master) > cd     #切换当前工作目录
→ ~ > pwd                      #查看当前工作目录
/home/kehr
→ ~ > exit                     #退出本地shell环境
test01@kehrspace > !pwd         #查看程序的本地工作目录
/home/kehr/Github/upcloud/upcloud

test01@kehrspace >
```

注意：在切换到本地shell环境的目的是执行一些本地操作，比如文件的复制、移动、删除等等，但是在本地shell中改变工作目录，并不影响程序的本地工作目录。如果希望修改程序的本地工作目录，以便使用相对路径上传当前工作目录（本地）中的文件，需要使用命令：`! cd PATH`

```
test01@kehrspace > !pwd
/home/kehr/Github/upcloud/upcloud

test01@kehrspace > !cd /home/kehr
/home/kehr
test01@kehrspace > !pwd
```

```
/home/kehr
```

前面的介绍中你可以见到以感叹号 ! 开始, 后接shell命令的用法。当你不想为了一个简单的操作而大费周章的切换到本地shell环境的时候, 可以使用这种方式暂时执行一个shell命令。

```
test01@kehrspace > !ls -l      #查看程序本地工作目录的文件
total 56
drwx-----  5 kehr kehr 4096 Feb 12 12:45 Algorithm
drwxr-xr-x   8 kehr kehr 4096 May 10 14:34 Desktop
drwx----- 17 kehr kehr 4096 May  4 22:14 Development
drwxr-xr-x  16 kehr kehr 4096 Apr 15 22:39 Documents
drwxr-xr-x   3 kehr kehr 4096 May 10 11:03 Downloads
drwx-----  7 kehr kehr 4096 May 10 12:44 Dropbox
drwx----- 16 kehr kehr 4096 Apr 30 13:30 Github
drwxr-xr-x   9 kehr kehr 4096 Jan 29 18:11 Music
drwxr-xr-x  12 kehr kehr 4096 May  5 13:53 Pictures
drwxr-xr-x   2 kehr kehr 4096 Jan 12 19:46 Public
drwxrwxr-x   3 kehr kehr 4096 May  9 13:26 temp
drwxr-xr-x   2 kehr kehr 4096 Feb 27 17:32 Templates
drwxrwxr-x   3 kehr kehr 4096 Feb 17 22:15 Ubuntu One
drwxr-xr-x   2 kehr kehr 4096 Jan 12 19:46 Videos
```

```
test01@kehrspace > ls -l      #查看云空间工作目录的文件
/ :
2014-05-10 15:14:26 <dir>      test
2014-04-16 21:31:54 <dir>      data
2014-04-17 16:03:27 <dir>      imgs
2014-05-01 19:49:31 <dir>      tmp
2014-05-04 17:21:00 <dir>      Users
2014-04-15 16:35:54 <dir>      img

6 directories, 0 files
```

12. 查看工作目录

查看云空间工作目录：

```
pwd
```

查看本地工作目录：

```
! pwd
```

13. 退出程序

使用命令：exit 或 quit 或 q

14. 高级特性

1. 清屏

使用命令: `clear`

清屏操作是最常用的操作之一，你可以使用命令 `clear` 或者 `cls` 清除冗余信息。

2. 文件名补全

程序支持当前目录（云空间）下文件名自动补全，连续按下两次 `<tab>` 键触发该操作。

```
test01@kehrspace > ll
/ :
2014-05-10 15:14:26 <dir>      test
2014-04-16 21:31:54 <dir>      data
2014-04-17 16:03:27 <dir>      imgs
2014-05-01 19:49:31 <dir>      tmp
2014-05-04 17:21:00 <dir>      Users
2014-04-15 16:35:54 <dir>      img

6 directories, 0 files
test01@kehrspace > cd
Users data img imgs test tmp
test01@kehrspace > cd t
test tmp
test01@kehrspace > cd tmp
tmp
test01@kehrspace > cd tmp
/tmp/
test01@kehrspace >
```

3. 命令补全

- 输入命令开头字母，按下 `<tab>` 键，补全命令
- 没有任何输入，连续按下两次 `<tab>` 键，显示可用命令列表

4. 历史命令回滚

使用上下键查看前一条或后一条命令

四、问题

程序仍旧存在一些已知或潜在的Bug，我正在努力的测试和完善。

- Github地址：<https://github.com/kehr/upcloud>
- Gitcafe地址：<https://gitcafe.com/kehr/upcloud-for-UPYUN>
- Pypi地址：<https://pypi.python.org/pypi/upcloud>

我的联系方式：

- mail：kehr.china@gmail.com
- 微信号：kehr036

有问题就铺天盖地的扔 issue 吧！非常希望能够与志同道合的你交流！