

## 异步，celery，rabbitmq

### 1. 如何在多台机器上部署celery和rabbitmq

假设有机器A， B

(1)

在机器B上安装并且启动rabbitmq， 默认自带guest用户(只能localhost登录)， 自己需要再添加一个用户admin

命令：

```
rabbitmqctl start_app 启动服务
```

```
rabbitmqctl list_users 列出所有用户
```

```
rabbitmqctl add_user admin admin 添加admin用户
```

```
rabbitmqctl set_permissions -p / admin .* .* .* 为admin添加权限
```

之后在B机器上可通过<http://127.0.0.1:15672>访问rabbitmq控制台

(2)

在机器A上建立文件

tasks.py

```
-----  
-----  
# -*- coding:utf-8 -*-
```

```
from celery import Celery
```

```
app = Celery('tasks', broker='amqp://admin:admin@B机器IP:5672',
```

```
backend='amqp://admin:admin@B机器IP:5672') #配置好celery的backend和broker
```

```
@app.task #普通函数装饰为 celery task
```

```
def add(x, y):
```

```
    return x + y  
-----  
-----
```

在机器B上建立文件

tasks.py

```
-----  
-----  
# -*- coding:utf-8 -*-
```

```
from celery import Celery
```

```
app = Celery('tasks', broker='amqp://guest:guest@localhost:5672',
```

```
backend=' amqp://guest:guest@localhost:5672') #配置好celery的backend和broker
```

```
@app.task #普通函数装饰为 celery task
def add(x, y):
    return x * y
```

---

注意这里机器A上的add函数执行加法， 而机器B上的add执行乘法。

(3)

在机器A上tasks.py所在的文件夹执行命令行指令

```
celery -A tasks worker --loglevel=info
```

在机器B上tasks.py所在的文件夹执行命令行指令

```
celery -A tasks worker --loglevel=info
```

(4)

在机器A或机器B上执行指令

```
from tasks import add
result = add.delay(4, 4)
result.get()
```

默认不作修改的情况下两个celery工作节点轮流获取任务， 所以重复执行这个指令会出现8， 16， 8， 16循环地输出， 让两个输出不同只是为了验证是在不同的机器上进行部署， 实际生产环境中函数应该是一样的以实现分布式。