**Installing Celery and creating first task:**

**Select a broker** (RabbitMQ). Visit <https://www.rabbitmq.com/#getstarted> for RabbitMQ documentation:

$ sudo apt-get install rabbitmq-server

Once the command completes, the server starts in the background.

**Install Celery:**

$ pip install celery

**Install Redis as backend**

$ sudo apt-get install redis-server

4. **Celery application file: tasks.py**

from celery import celery

app = Celery('tasks', broker='pyamqp://guest@localhost//' ,backend= 'redis://’)

@app.task

def multiply(x, y):

return x \* y

A single task is defined called multiply which returns a multiplication of two numbers.The first argument ‘tasks’ is the name of the current module. Second argument is the broker keyword argument, specifying the broker url, in this case rabbitmq.Third is the backend storage for the results.

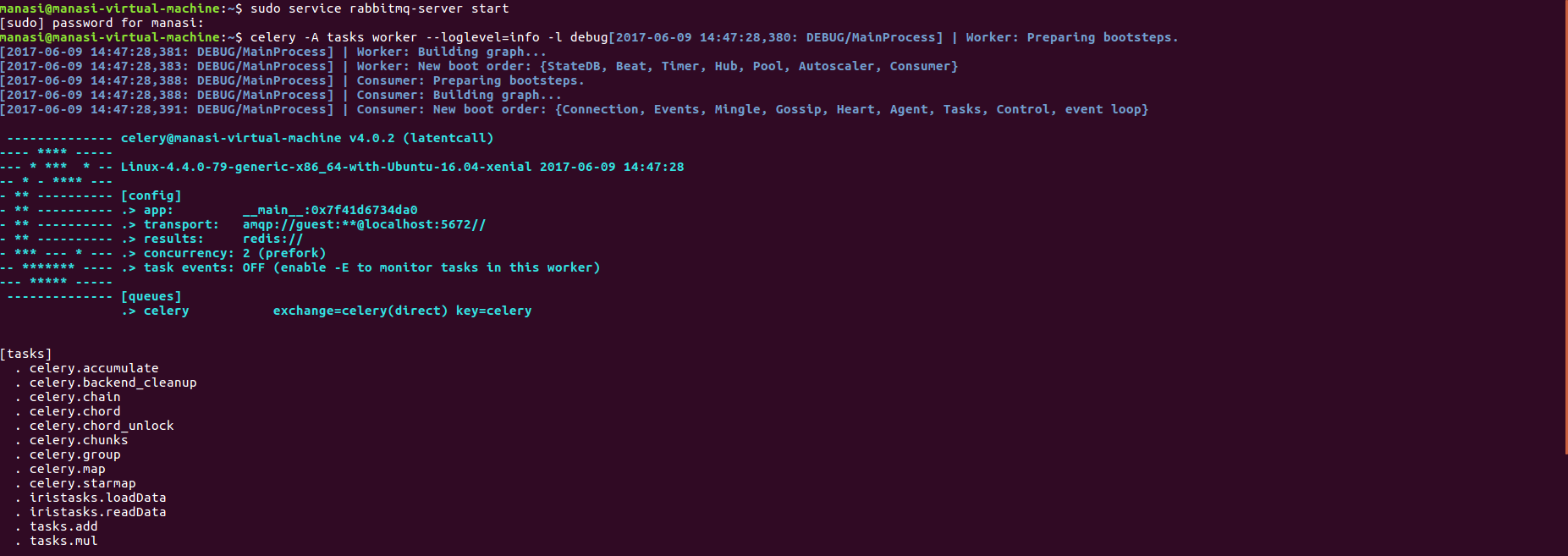
**Run the script**:

5.Start/restart the rabbitmq server

$ sudo service rabbitmq-server start

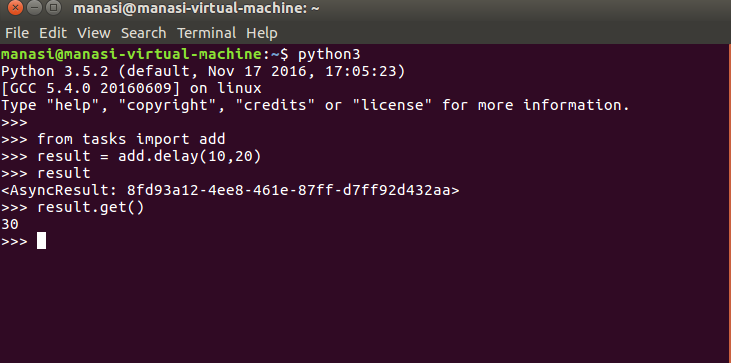
6. Run the celery worker daemon in debug mode

$ celery -A tasks worker --loglevel=info -l debug



**Calling the add function.**

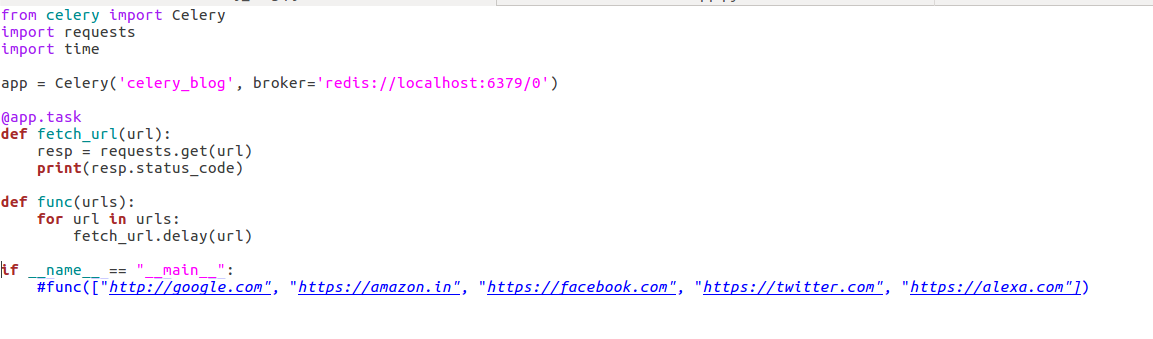
Open a python shell and using the delay() call the task as shown:



**Delay()** is a shorthand of **apply\_async()** which generates async calls and results, which can be viewed on the workers output console.

**For the celery parallel processing example restart the celery worker daemon with:**

$ celery worker -A celery\_blog -l info -c 5 (specify concurrency).



**Thea above python script is available on the** [github link.](https://github.com/vishalsatam/Data-Pipelining)