



# Backend Celery task manager dashboard via Flower

Date: 2023-10-29  
By: csr13

Download as PDF

Sometimes deploying systems with asynchronous tasks is required per client specifications.

For these instances I tend to dockerize celery redis and monitor them with Flower, which provides visualization for the performance of system asynchronous tasks. version: '3.9'

```
services:
  ... rest of services (redis, etc)
tasks:
  build:
    context: .
    dockerfile: ./docker/tasks/Dockerfile
  command: >
    sh -c "python3 -m celery -A my_system flower --address=0.0.0.0 --port=5566 & python3 -m celery -A my_system worker -l INFO -E"
  env_file:
    - ./docker/tasks/.env
  depends_on:
    - redis
  restart: unless-stopped
  network:
  my-network:
    ipv4_address: 10.10.10.2
  expose:
    - "5566"

networks:
  ... network config
```

On the command: section of the dockerfile, putting flower on the background so the worker can operate normally, notice the ampersand & and the ; if the previous command fails, the container won't be marked as exited by the docker daemon.

Usually setting up a shared task monitoring tool over public http is not the best idea.

I pipe to my remote workstation machine all flower instances for all systems using ssh

```
~$ ssh -L 10.10.10.2:5566:127.0.0.1:8001 -i my_system.pem user@$MY_SYSTEM_IP
```

Open browser at 127.0.0.1:8001 and monitor my systems shared tasks, configure rate limits, etc .... You can fine tune flower, check out their documentation

## Related Notes

[1\) Real Time Language Translation Agent System for Call Centers](#)  
[Download PDF](#)  
Date published: 2023-11-16  
[voip telephony python systems](#)

[2\) Using ec2 instances as sneaker bid bots pt 2.](#)  
[Download PDF](#)  
Date published: 2023-11-27  
[bots python aws](#)

[3\) Whatsapp chatbot with Python and Twilio](#)  
[Download PDF](#)  
Date published: 2023-11-15  
[bots python whatsapp business](#)

[4\) Using ec2 instances as sneaker bid bots pt 1.](#)  
[Download PDF](#)  
Date published: 2023-11-21  
[bots python aws](#)