

# 1. Objective

จุดประสงค์คือแยก worker ไว้อีก 1 เครื่อง

## 2. Current Condition

IP: 10.4.29.208 = **ก้อนสีแดง** + **ก้อนเหลือง**

LogService

ResourceService

Checker

EvaluationService

**Worker**

ScoringService

ProxyService

PrintingService

ContestWebServer

AdminWebServer

**RankingWebServer**

IP: 10.4.29.209 = worker **ก้อนสีน้ำเงิน**

ResourceService

Worker: actually runs the jobs in a sandboxed environment;

OS: Ubuntu 18.04LTS

User: graderadmin

Pass: 1029&&c\$2022

## 3. Proposed Setup for TOI18

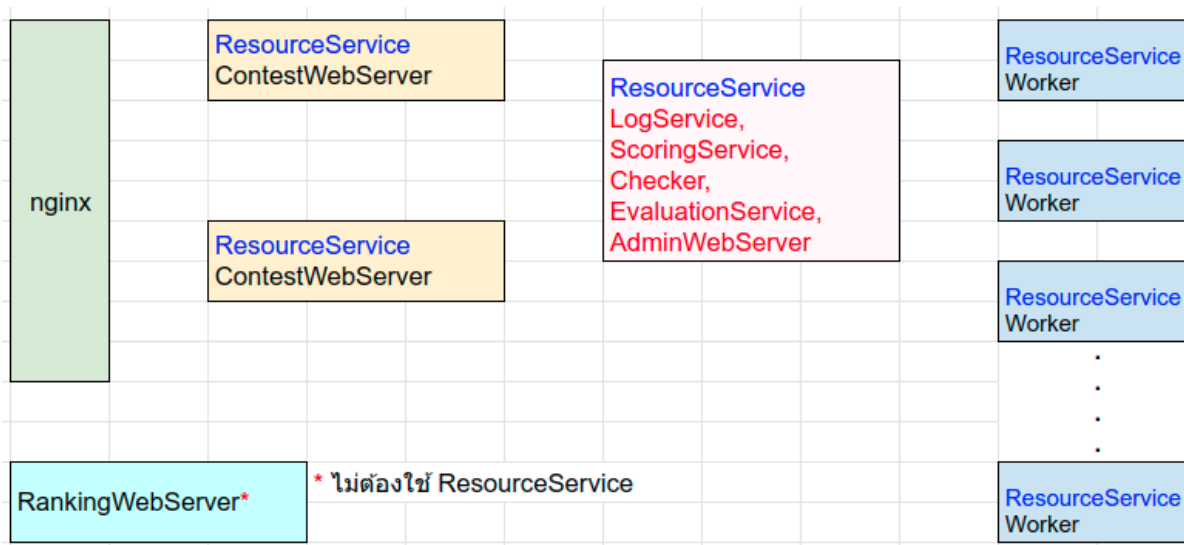


Diagram:

[https://docs.google.com/spreadsheets/d/1URtBJ4ip2\\_VeJ5nu6HO1YKbLXoHOKvrejscrOtVqSEU/](https://docs.google.com/spreadsheets/d/1URtBJ4ip2_VeJ5nu6HO1YKbLXoHOKvrejscrOtVqSEU/)

มีคนเคยทำแบ่งแล้ว doc ไว้ที่

<https://bytefreaks.net/applications/contest-management-system/installing-cms-contest-management-system-on-multiple-servers>

## Recommended setup

Of course, the number of servers one needs to run a contest depends on many factors (number of participants, length of the contest, economical issues, more technical matters...). We recommend that, for fairness, each Worker runs on a dedicated machine (i.e., without other CMS services beyond ResourceService).

As for the distribution of services, usually there is one **ResourceService** for each machine, one instance for each of **LogService**, **ScoringService**, **Checker**, **EvaluationService**, **AdminWebServer**, and one or more instances of **ContestWebServer** and **Worker**. Again, if there are more than one Worker, we recommend to run them on different machines.

The developers of isolate (the sandbox CMS uses) provide a script, **isolate-check-environment** that verifies your system is able to produce evaluations as fair

and reproducible as possible. We recommend to run it and follow its suggestions on all machines where a Worker is running. You can download it [here](#).

We suggest using CMS over Ubuntu. Yet, CMS can be successfully run on different Linux distributions. Non-Linux operating systems are not supported.

We recommend using nginx in front of the (one or more) `cmsContestWebServer` instances serving the contestant interface. Using a load balancer is required when having multiple instances of `cmsContestWebServer`, but even in case of a single instance, we suggest using nginx to secure the connection, providing an HTTPS endpoint and redirecting it to `cmsContestWebServer`'s HTTP interface.

See [config/nginx.conf.sample](#) for a sample nginx configuration. This file probably needs to be adapted to your distribution if it is not Ubuntu: try to merge it with the file you find installed by default. For additional information see the official nginx [documentation](#) and [examples](#). Note that without the `ip_hash` option some CMS features might not always work as expected.

## 4. Ubuntu version

a. แนะนำให้ใช้ 18.04

b. ถ้าจำเป็นต้องลงบน 20.04 ต้องลง python 3.6

```
$ sudo add-apt-repository ppa:deadsnakes/ppa
$ sudo apt update
$ sudo apt install python3.6
$ sudo apt-get install python3.6-distutils
$ sudo apt-get install python3.6-dev
$ sudo apt install libpq-dev
$ sudo apt install python3.6-venv
$ sudo python3.6 -m ensurepip --default-pip --user
$ sudo python3.6 -m pip install --upgrade pip
```

requirement.txt บน 20.04 ที่ test แล้ว

```
tornado==4.5.3
psycpg2==2.7.7
SQLAlchemy==1.1.18
netifaces==0.10.9
pycrypto==2.6.1
psutil==5.4.8
six==1.11.0
requests==2.18.4
gevent==1.2.0
Werkzeug==0.14.1
patool==1.12
bcrypt==3.1.7
chardet==3.0.4
ipaddress==1.0.23
future==0.15.2
Babel==2.4.0
pyxdg==0.25
Jinja2==2.10.3
httplib2==0.14.0
keyring==18.0.1
```

แล้วลง requirement ของ python (พวก `sudo apt install python-XXXX`) แบบ อ่าน python requirements ผ่าน pip แล้วเรียกทุกอย่างด้วย python3.6 ตรงๆ ดูตัวอย่างจากกรอบล่าง (ยังไม่ต้องทำตอนนี้ รอทำตาม step กรอบนี้แค่นี้ก่อน)

```
$ sudo python3.6 -m pip install -r requirements.txt --no-use-pep517
....
$ sudo python3.6 prerequisites.py install
```

```
...  
$ sudo python3.6 setup.py install
```

## 5. Log การติดตั้ง

-ssh เข้าเครื่อง grader

- เริ่มต้น update

```
$ sudo apt update; sudo apt upgrade
```

ลง build essential กรอบแรกใน diagram

```
$ sudo apt-get install build-essential openjdk-8-jdk-headless fp-compiler \  
    postgresql postgresql-client python3.6 cppreference-doc-en-html \  
    cgroup-lite libcap-dev zip python3-pip  
  
$ sudo apt-get install python2.7 php7.2-cli php7.2-fpm \  
    phppgadmin
```

```
sudo apt-get install build-essential openjdk-8-jdk-headless fp-compiler \  
    postgresql postgresql-client python3.6 cppreference-doc-en-html \  
    cgroup-lite libcap-dev zip
```

อันตัวสีน้ำเงินนี้น่าจะลงเฉพาะ main ถ้าเป็น worker อาจจะต้องลง python2.7 ถ้าใช้ แต่เครื่องอื่น ๆ นอกจากนั้นอาจจะไม่จำเป็น (20.04 จะหา python2.7 ไม่เจอ)

```
sudo apt-get install nginx-full python2.7 php7.2-cli php7.2-fpm \  
    phppgadmin \ ๓๑ nginx-full texlive-latex-base a2ps haskell, rusts mono  
gcj-jdk ออก
```

ดาวน์โหลด

```
$ cd  
$ wget https://github.com/cms-dev/cms/releases/download/v1.4.rc1/v1.4.rc1.tar.gz
```

แตกไฟล์

```
$ tar xvf v1.4.rc1.tar.gz
```

run script เตรียม env ต่างๆ

```
$ cd ~/cms
```

```
$ sudo python3 prerequisites.py install
```

ตอบ Y ด้วย

Reboot

ลง requirement ที่ต้องการ

```
$ sudo apt-get install python3-setuptools python3-tornado python3-psycpg2 \
python3-sqlalchemy python3-psutil python3-netifaces python3-crypto \
python3-six python3-bs4 python3-coverage python3-mock python3-requests \
python3-werkzeug python3-gevent python3-bcrypt python3-chardet patool \

python3-babel python3-xdg python3-future python3-jinja2
```

Optional

```
$ sudo apt-get install python3-yaml python3-sphinx python3-cups python3-pypdf2
```

-----

จะเริ่ม install แล้ว อยู่ใน folder cms

```
$ cd ~/cms
$ sudo python3 setup.py install
```

-----

Config database (ทำที่เครื่อง main เครื่องเดียว เครื่องอื่นๆ ไม่ต้อง)

```
sudo su - postgres

createuser --username=postgres --pwprompt cmsuser //แล้วกด password ไว้

createdb --username=postgres --owner=cmsuser cmsdb

psql --username=postgres --dbname=cmsdb --command='ALTER SCHEMA public OWNER TO
cmsuser'
```

```
psql --username=postgres --dbname=cmsdb --command='GRANT SELECT ON
pg_largeobject TO cmsuser'

exit
```

จากนั้นไปแก้ database password

```
$ sudo nano /usr/local/etc/cms.conf
```

แก้ line database

```
"database":
"postgresql+psycopg2://cmsuser:<PASSSSWORD>@10.4.29.208:5432/cmsdb",
```

จากนั้นเรียก

```
$ cmsInitDB
```

```
*-----
```

## Configuring CMS

เข้าไปที่ พวก file .sample config ranking

แก้ cms.conf แก้ database

ตัว secret\_key คัดไว้ก่อน

```
$ cd ~/cms/config
$ cp cms.conf.sample cms.conf
$ cp cms.ranking.conf.sample cms.ranking.con
```

```
-----
```

## Running CMS

ย้ายมาที่ cms เพื่อ copy สิ่งที่เกี่ยวข้อง ขึ้นไป ให้เรียก

```
$ cd ~/cms
$ sudo python3 prerequisites.py install
```

สร้าง screen script เพื่อ start service ต่างๆ เป็น bg

```
$ mkdir ~/CMU_scripts
```

สร้าง ไฟล์ startAll.sh

```
$ cd ~/CMU_scripts
$ nano startAll.sh
```

```
#!/bin/bash

# Add CMS Location to Environment (for crontab)
PATH=$PATH:/usr/local/bin
export PATH

/usr/bin/screen -wipe

# Run CMS
/usr/bin/screen -S cmsLogService -d -m /bin/bash -c
'/usr/local/bin/cmsLogService'

/usr/bin/screen -S cmsResourceService -d -m /bin/bash -c
'/usr/local/bin/cmsResourceService -a 1'
```

เซฟและ exit

สำคัญ เครื่องหลักต้อง start cmsLogService ด้วย

เครื่องอื่น เอาแค่ cmsResourceService

ทำให้เป็น executable

```
$ chmod +x startAll.sh
```

สร้าง admin



```
$ cmsAddAdmin cmuadmin
2022-03-09 08:09:09,895 - INFO [<unknown>] Using configuration file
/usr/local/etc/cms.conf.
2022-03-09 08:09:10,119 - INFO [<unknown>] Creating the admin on the database.
2022-03-09 08:09:10,343 - INFO [<unknown>] Admin with complete access added.
Login with username cmuadmin and password <GENERATED_PASSWORD>
$ exit
```

จด password ไว้

แล้ว ssh ให้มาใหม่ โดยทำ tunnel ก่อน

---

```
ssh -L 8889:127.0.0.1:8889 graderadmin@10.4.29.208
```

เราจะสร้าง contest ก่อน 1 อัน

```
$ cmsAdminWebServer
```

แล้ว เข้าเว็บไปที่ localhost:8889

Login สร้าง contest ใหม่ com O camp 1

---

เราสร้าง killAll.sh

```
$ nano ~/CMU_scripts/killAll.sh
```

```
#!/bin/bash
screen -wipe
screen -S cmsAdminWebServer -X quit
screen -S cmsResourceService -X quit
screen -S cmsLogService -X quit
```

```
$ chmod +x killAll.sh
```

---

ตั้งให้ admin เข้าได้จาก localhost เท่านั้น

```
$ nano ~/cms/config/cms.conf
```

```
"admin_listen_address" : "127.0.0.1"
```

แล้ว update ให้ค่าสั่ง

```
$ cd ~/cms  
$ sudo ./prerequisites.py install
```

-----