新建一個test-Iris的資料夾

照著網址建立train\_model.py server.py client.py(需修改ip)

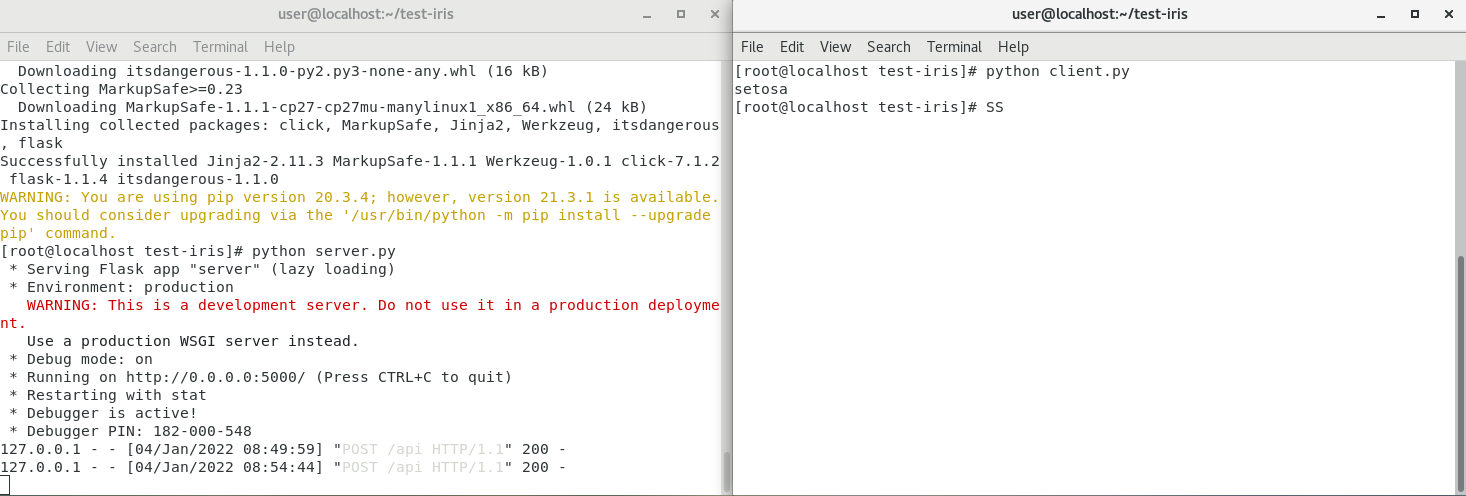
需要pip install sklearn與 flask

若無法pip install 則需要

wget <https://bootstrap.pypa.io/pip/2.7/get-pip.py>

python get-pip.py

完成後運行server.py 與client.py



註冊gitlab

設置sshkey

輸入cd ~

輸入cd .ssh/

輸入ssh-keygen 生成sshkey

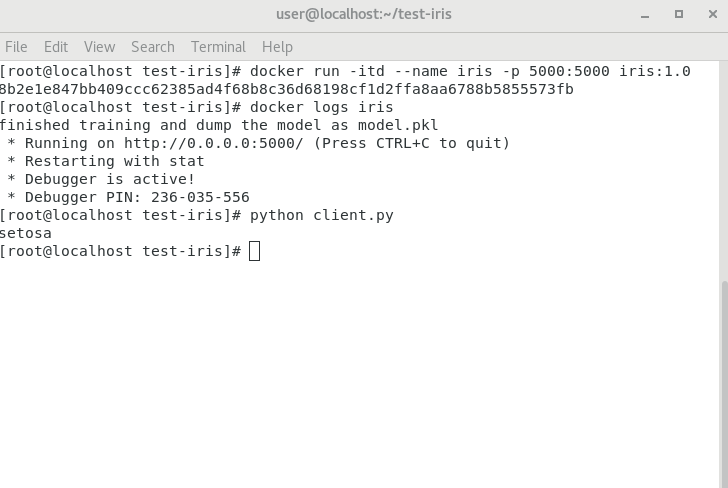
輸入 cat id\_rsa.pub 查看並複製貼上至gitlab

製作成容器

在test-Iris的資料夾建立Dockerfile

在test-Iris的資料夾建立requirements.txt

編輯完後 輸入docker build -t iris:1.0 . 建立鏡像

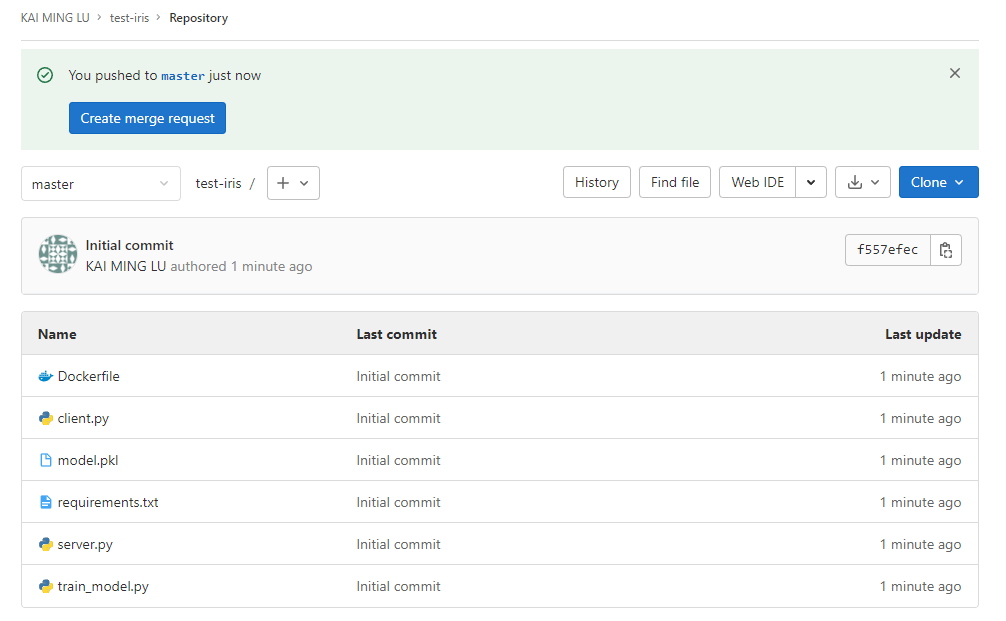


用gitlab自動化部屬

在gitlab創建project

進行gitlab設置

將iris檔案push到gitlab



安裝gitlab-runner

輸入sudo curl -L --output /usr/local/bin/gitlab-runner <https://gitlab-runner-downloads.s3.amazonaws.com/latest/binaries/gitlab-runner-linux-amd64>

下載完後輸入sudo chmod +x /usr/local/bin/gitlab-runner 變成可執行檔案

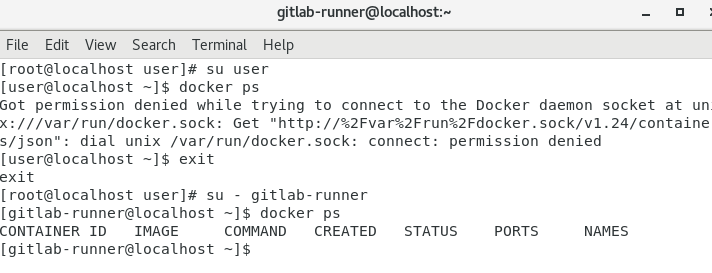
創建使用者

sudo useradd --comment 'GitLab Runner' --create-home gitlab-runner --shell /bin/bash

新增一個可執行docker的user

usermod -aG docker gitlab-runner

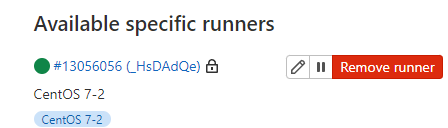
測試



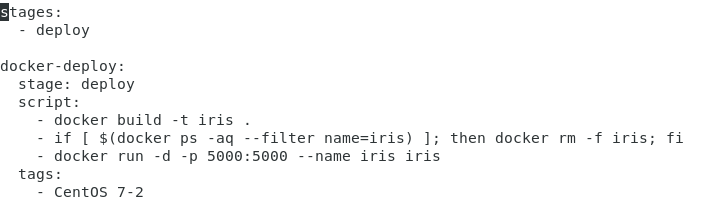
進行gitlab-runner註冊

可以在Gitlab的setting裡的CI/CD 找到runner 若為紅色則輸入

gitlab-runner run --working-directory /home/gitlab-runner --config /etc/gitlab-runner/config.toml --service gitlab-runner --syslog --user gitlab-runner



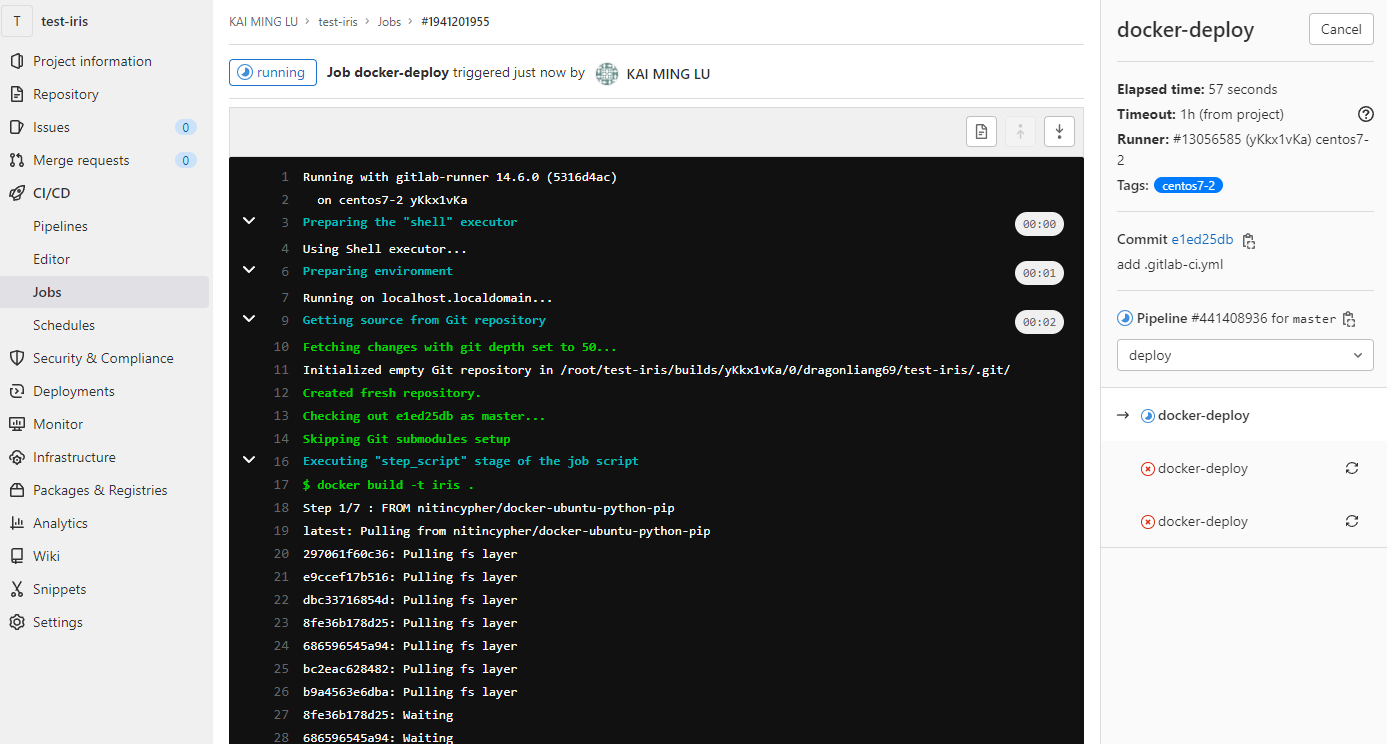
設置.gitlab-ci.yml 檔案



輸入git add .gitlab-ci.yml

輸入git commit -m "add .gitlab-ci.yml"增加剛剛的檔案

輸入git push -u origin master 上傳



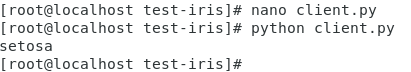
在CI/CD Jobs中可以看到正在部屬

在第二台虛擬機可以看到正在運行



在第一台虛擬機修改客戶端ip

並執行



參考網址:

<https://github.com/linjiachi/Linux_note/blob/master/109-1%20Docker/W7-20201027.md#%E5%B0%87-iris-%E6%89%93%E5%8C%85%E6%88%90-docker>

<https://wulc.me/2019/04/19/%E9%80%9A%E8%BF%87%20Flask,%20Docker,%20Jenkins%20%E5%92%8C%20Kubernets%20%E9%83%A8%E7%BD%B2%E6%9C%BA%E5%99%A8%E5%AD%A6%E4%B9%A0%E6%A8%A1%E5%9E%8B/>

<https://hackmd.io/@110510549/HkRlhLHuP>

https://stackoverflow.com/questions/52595158/gitlab-runner-not-picking-up-jobs-automatically-after-reboot