Docker GUI Controller

- Requirements
  - python3-tk Package
  - Docker
  - Ubuntu OS

tkinter install

```
$ sudo apt install python3-tk
```

Admin용 GUI Controller

실행 방법

```
$ python ./docker_gui_controller/admin/run.py
```

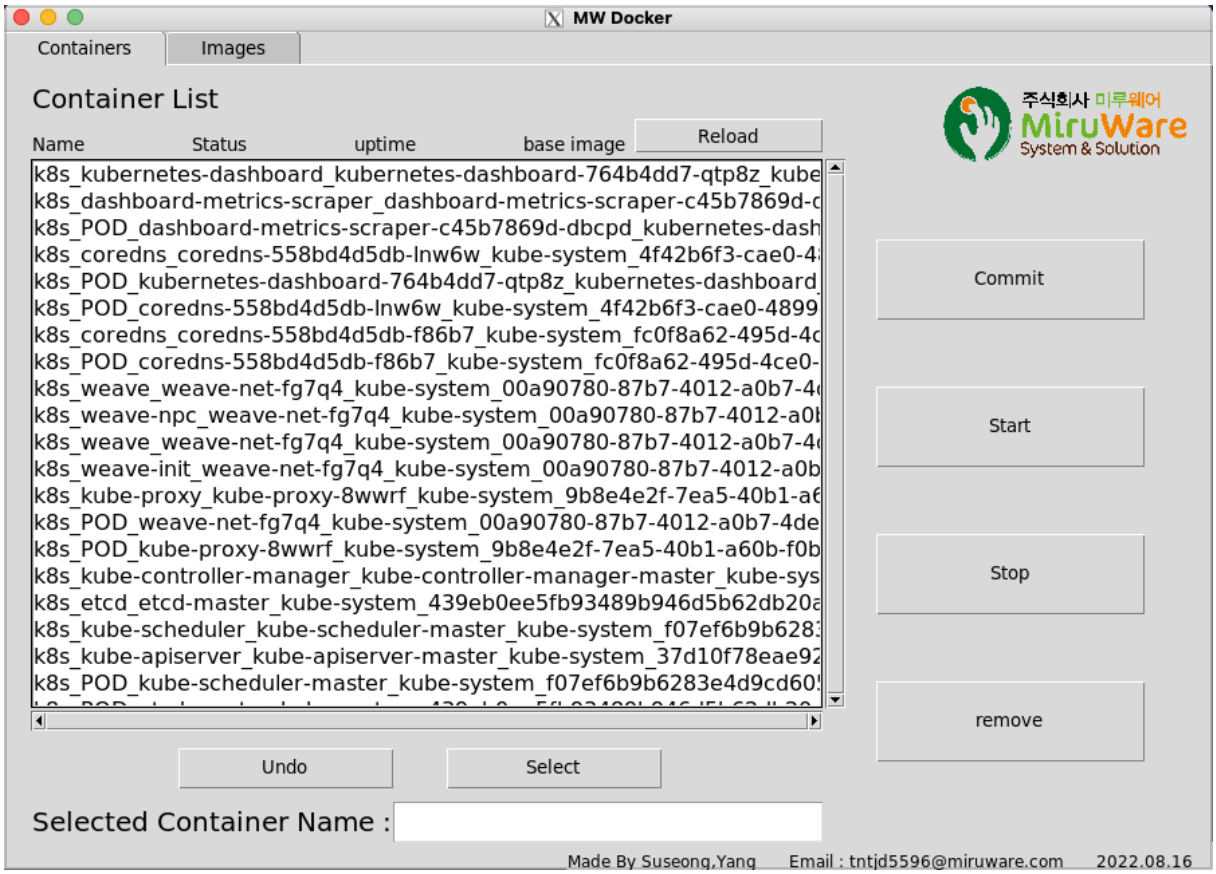
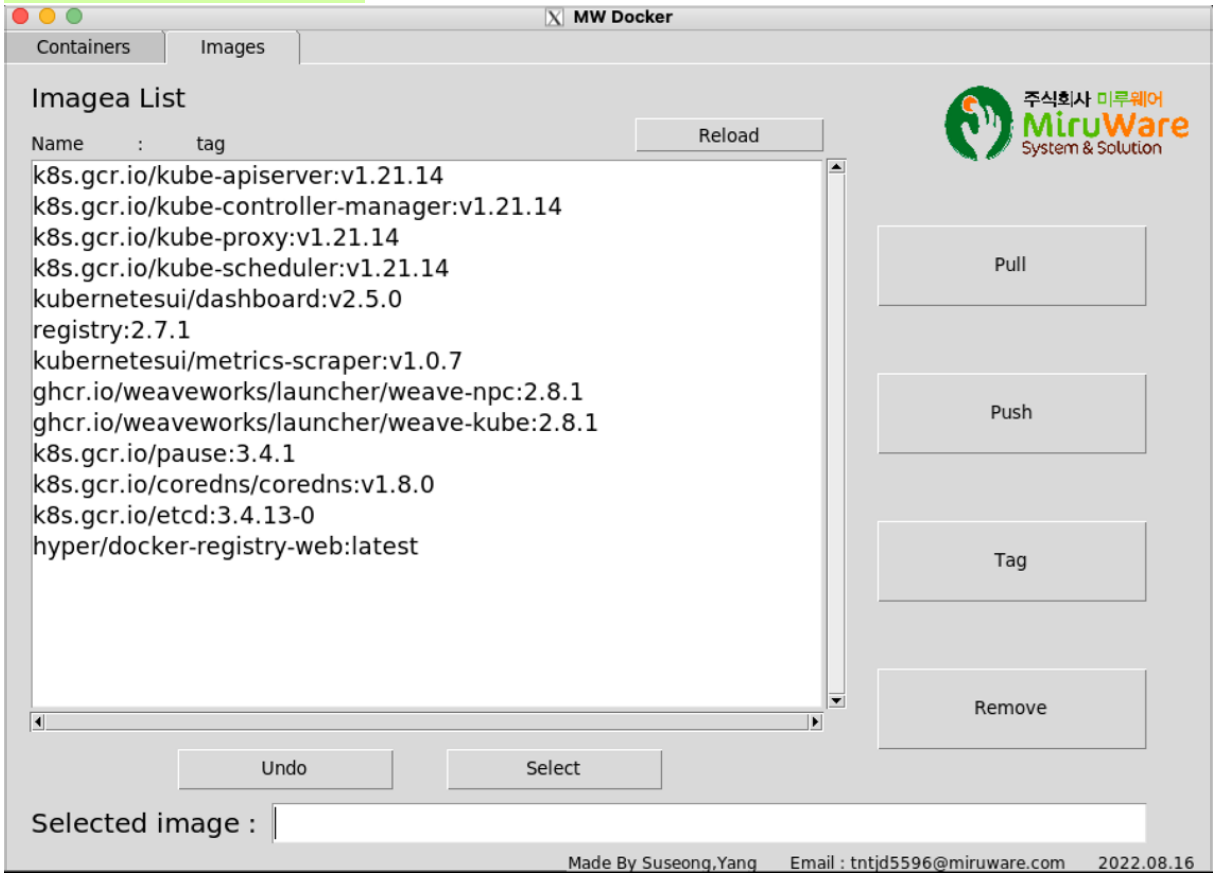


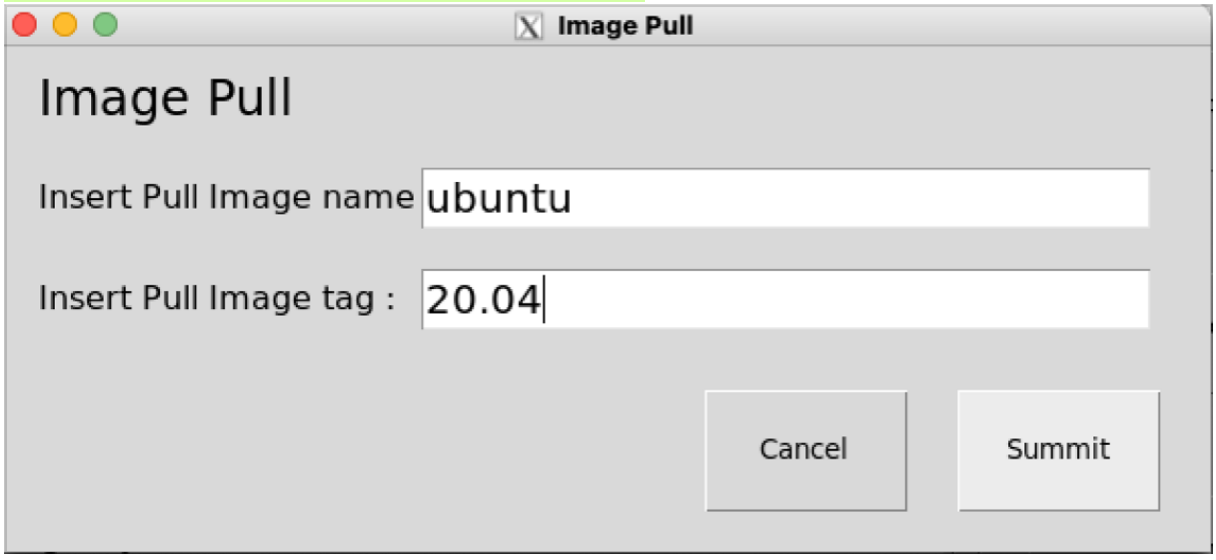
Image 관리

Docker pull

Images 탭 에서 Pull 버튼 클릭



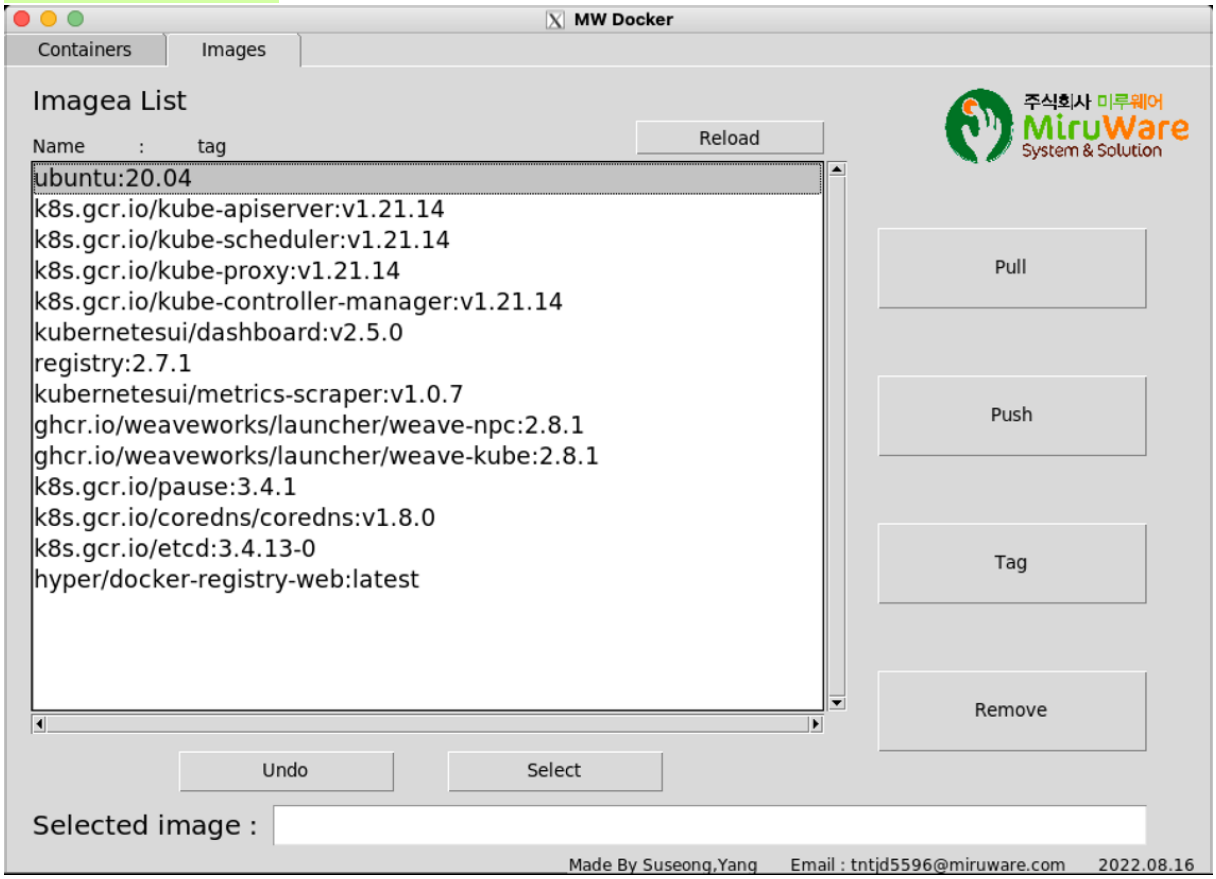
받아올 이미지 이름과 Tag를 입력 후 Summit 버튼 클릭



터미널에서 이미지 Pull 진행 확인

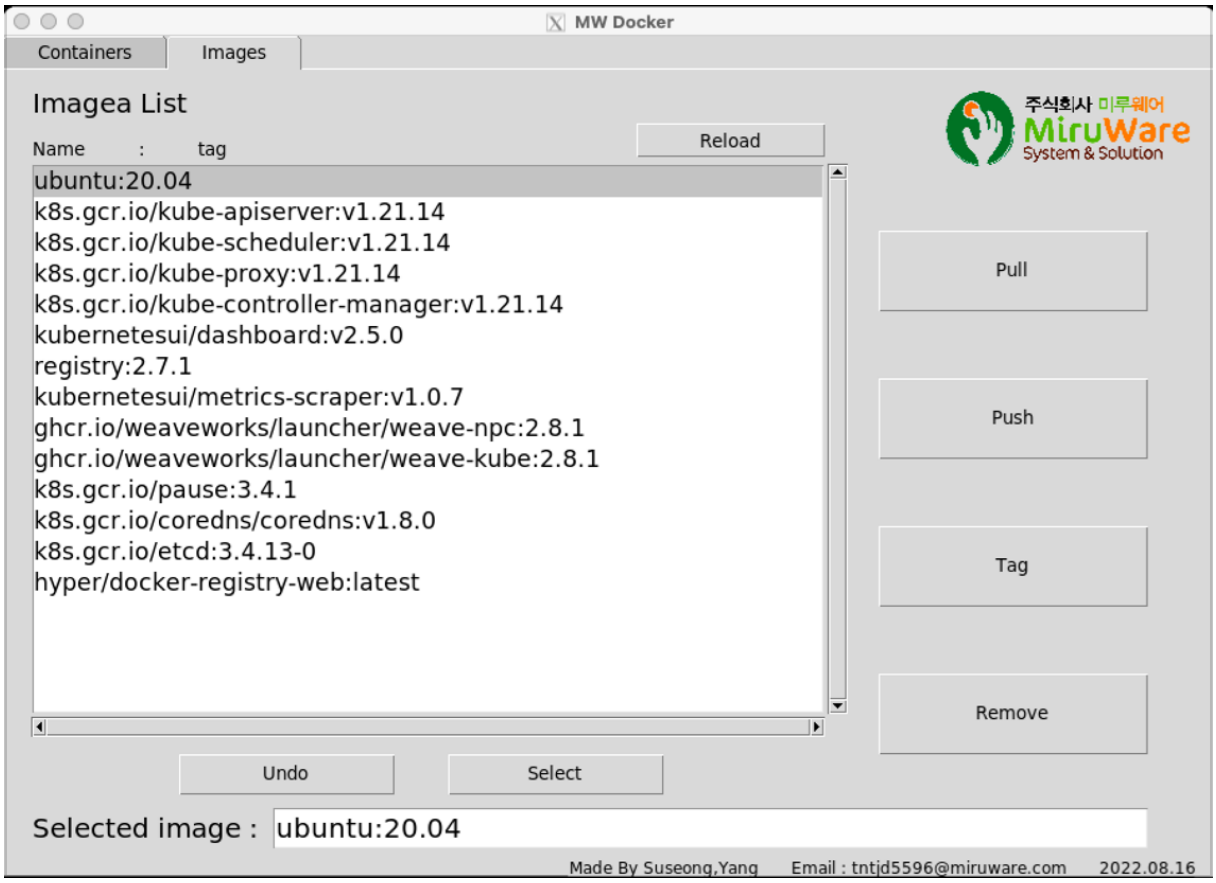
```
20.04: Pulling from library/ubuntu
3b65ec22a9e9: Pull complete
Digest: sha256:af5efa9c28de78b754777af9b4d850112cad01899a5d37d2617bb94dc63a49aa
Status: Downloaded newer image for ubuntu:20.04
docker.io/library/ubuntu:20.04
```

Pull 완료된 이미지 확인

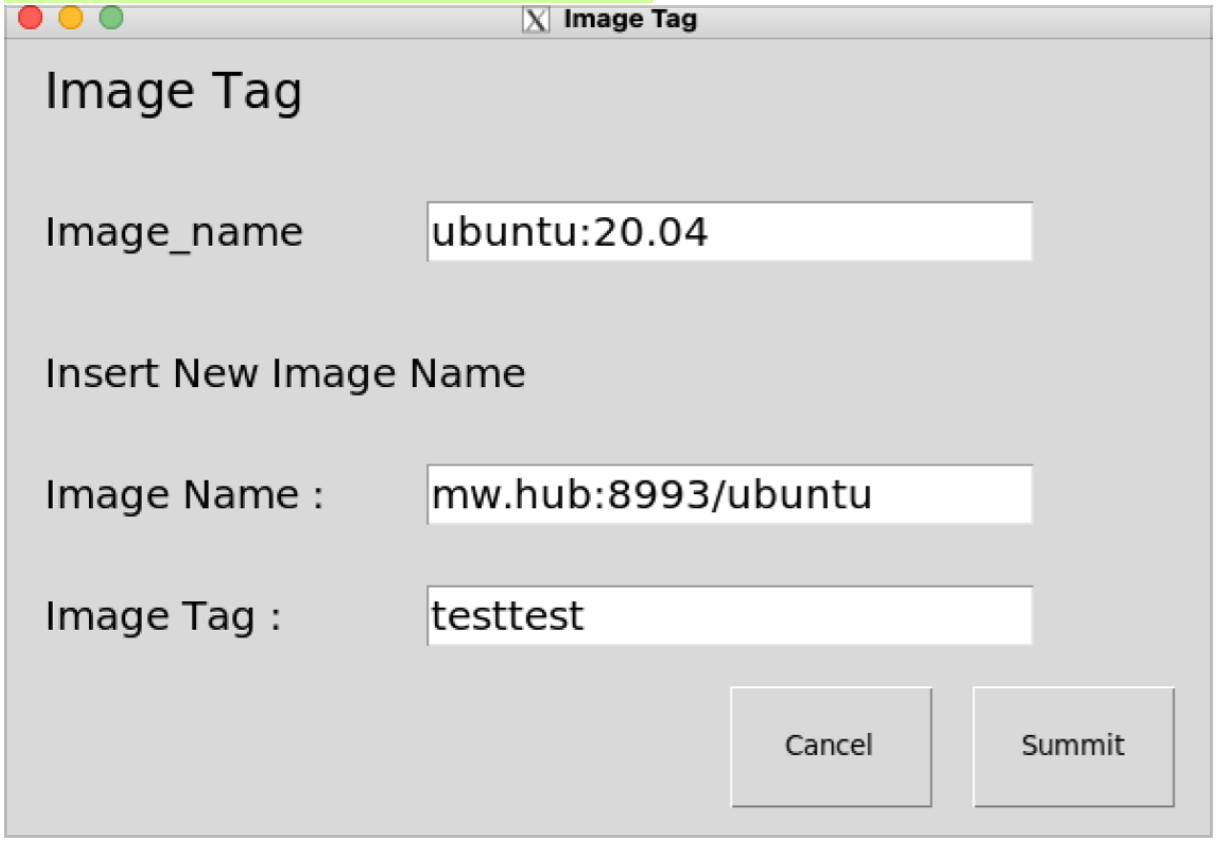


Docker tag

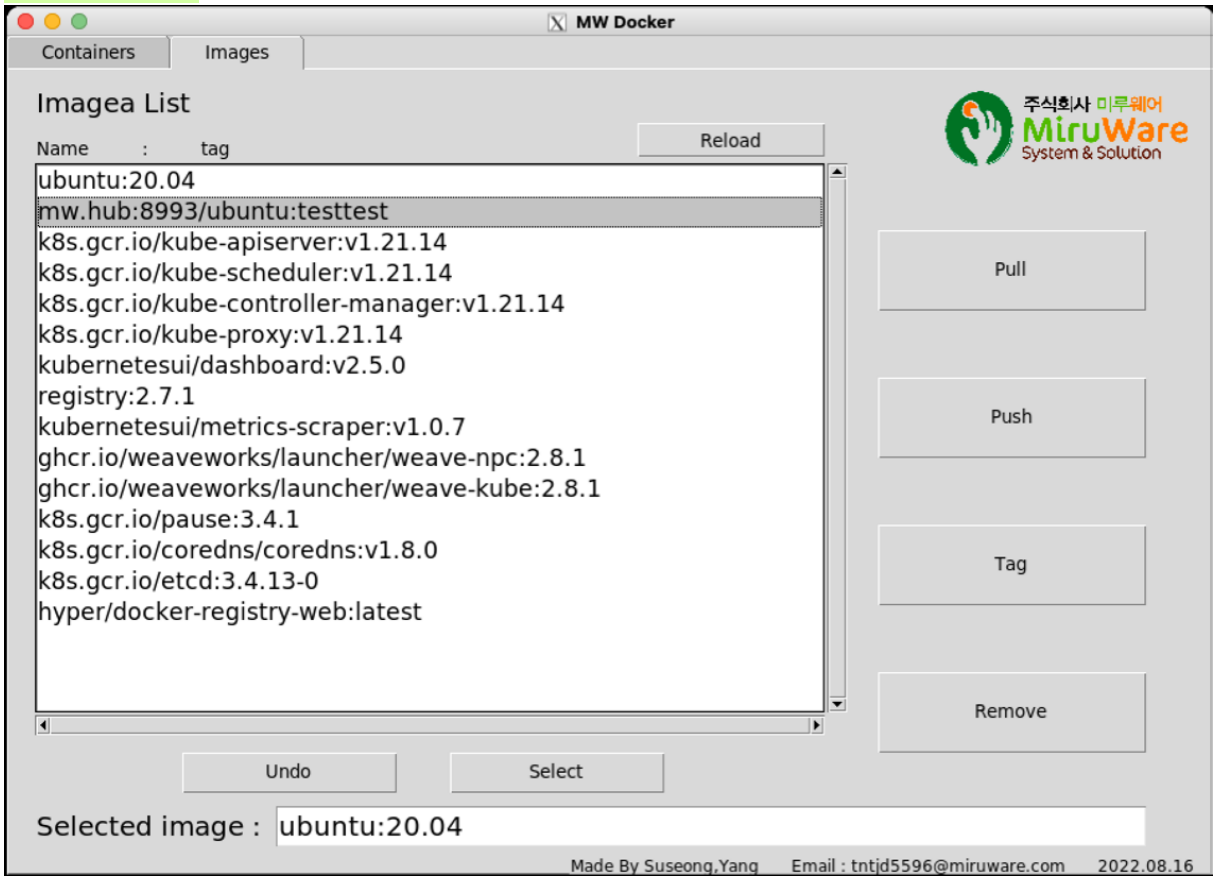
Images 탭 에서 원하는 이미지 Select 한 이후 Tag 버튼 클릭



생성할 이미지 이름과 Tag 를 입력 후 Summit 버튼 클릭



이미지 생성 확인



이미지 Push

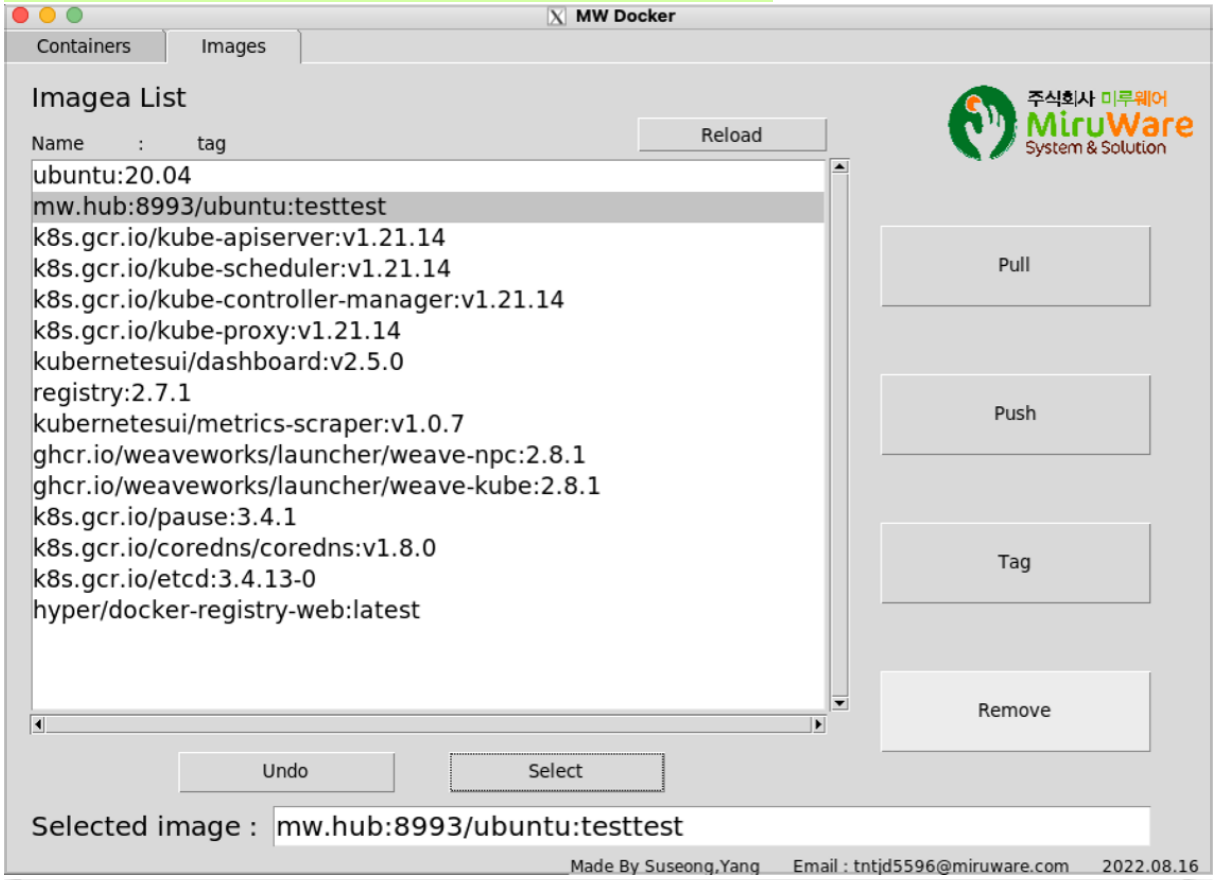
Images 탭 에서 원하는 이미지 Select 한 이후 Push 버튼 클릭

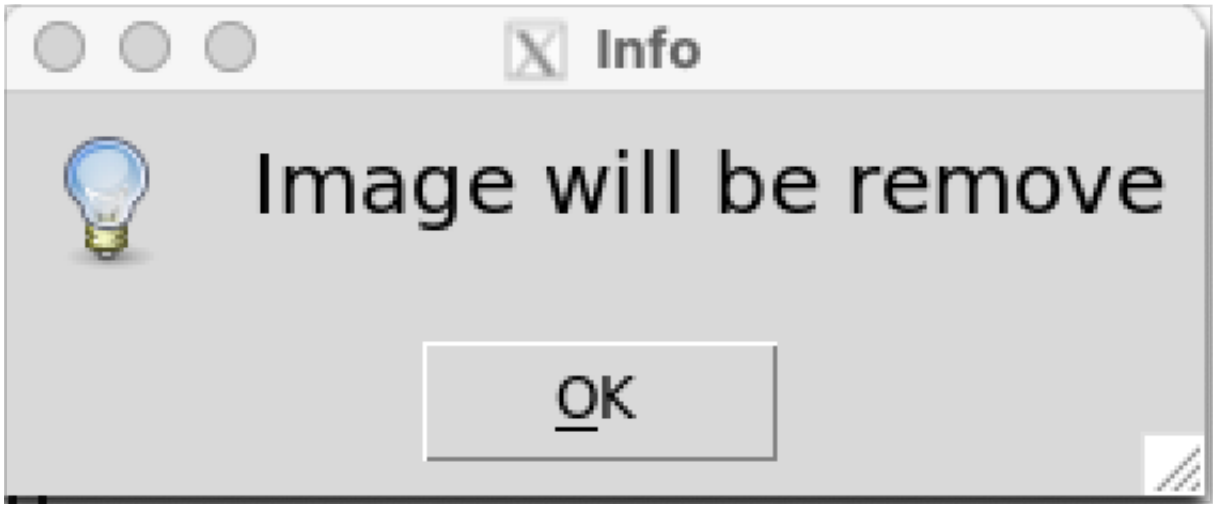

```
The push refers to repository [mw.hub:8993/ubuntu]
c3f11d77a5de: Layer already exists
testtest: digest: sha256:a06ae92523384c2cd182dcfe7f8b2bf09075062e937d5653d7d0db0375ad2221 size: 529
```

이미지 업로드 확인

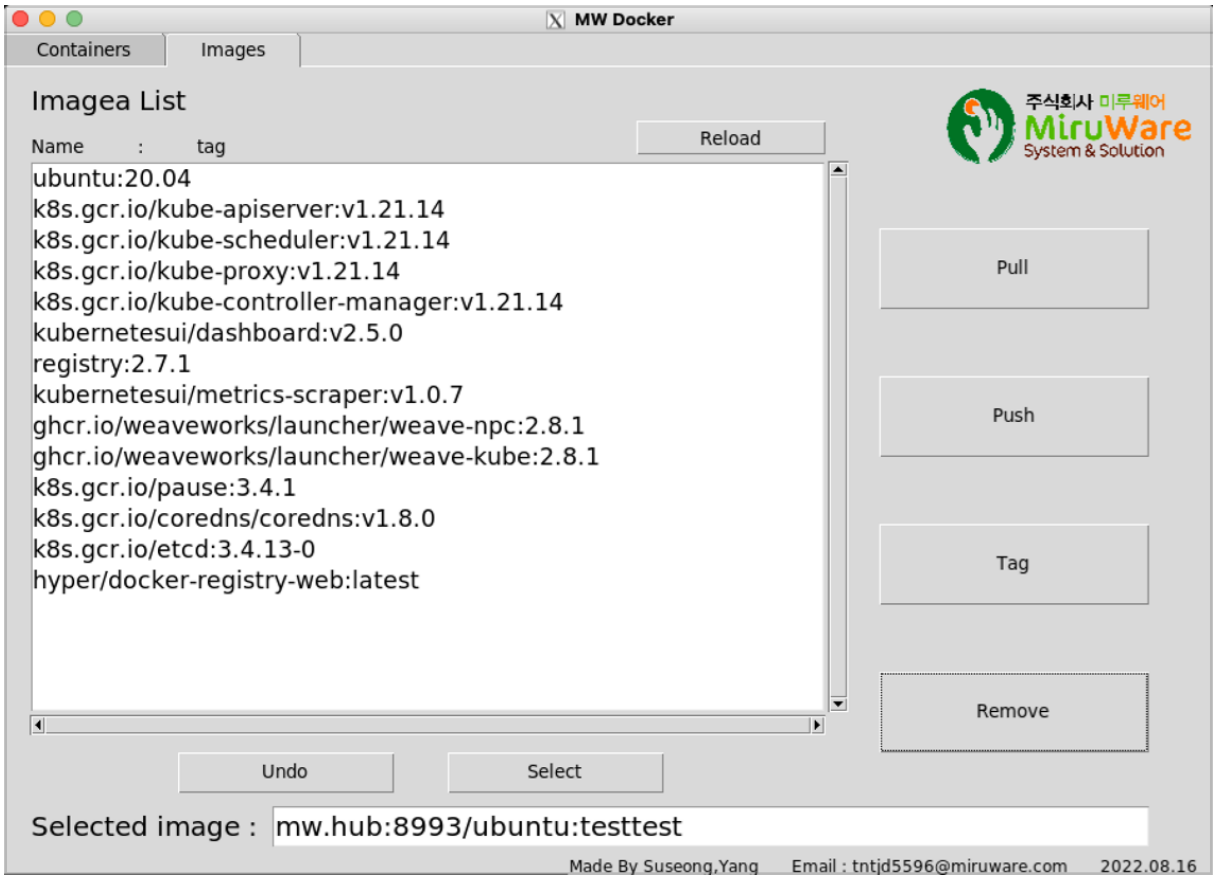
이미지 Remove

Images 탭 에서 원하는 이미지 Select 한 이후 Remove 버튼 클릭





이미지 삭제 확인

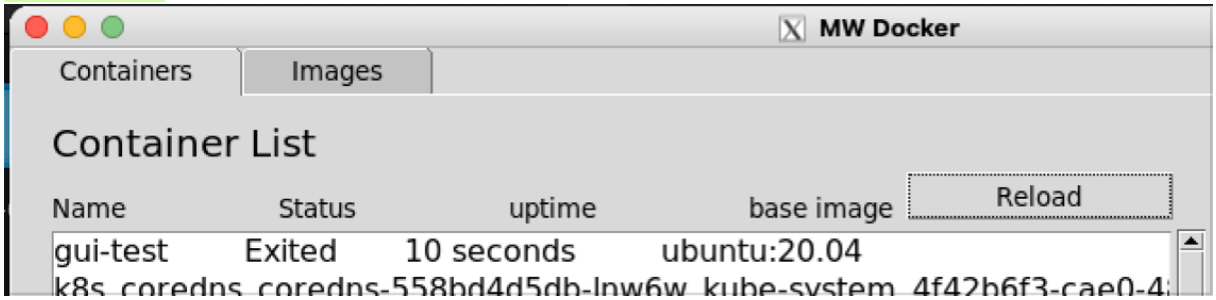


Container 관리

예시를 위한 컨테이너 생성

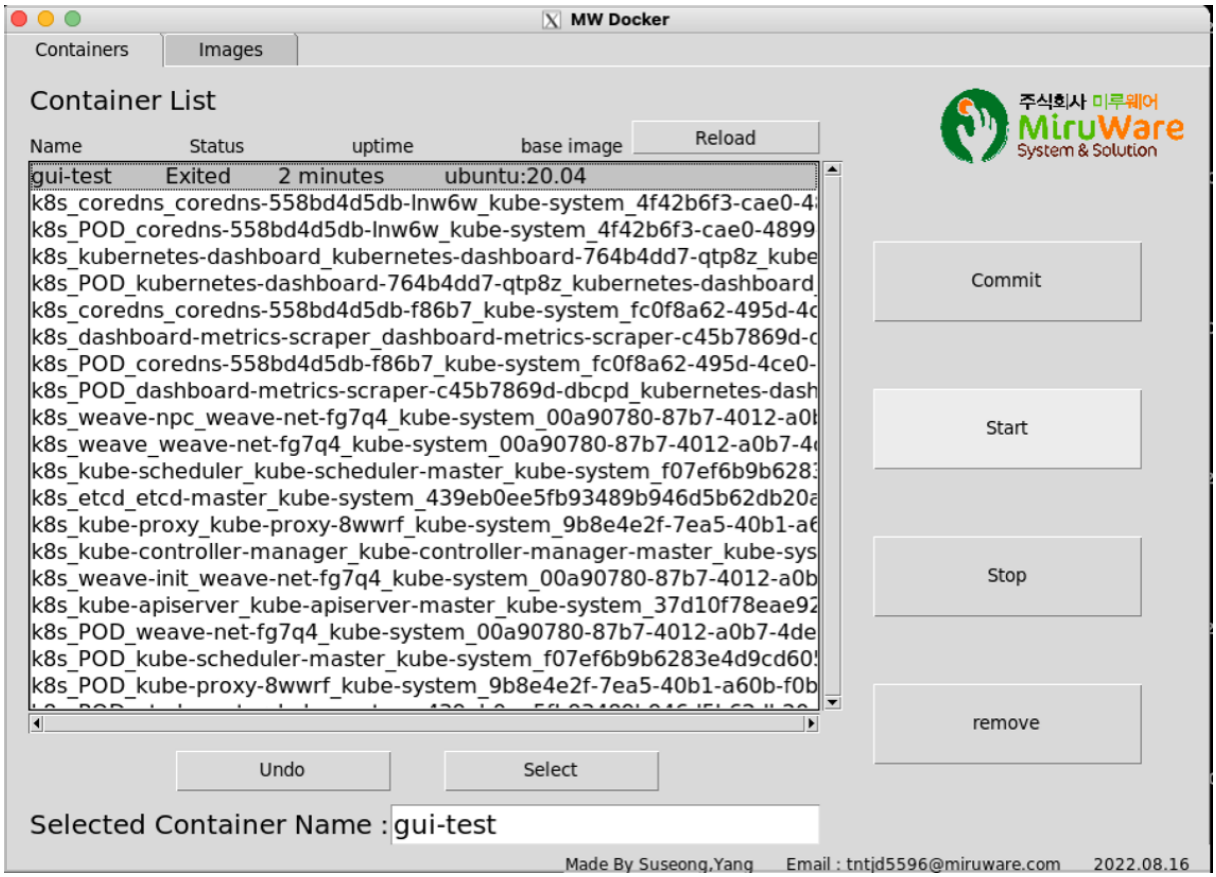
```
$ docker run -d --name gui-test ubuntu:20.04
```

컨테이너 확인

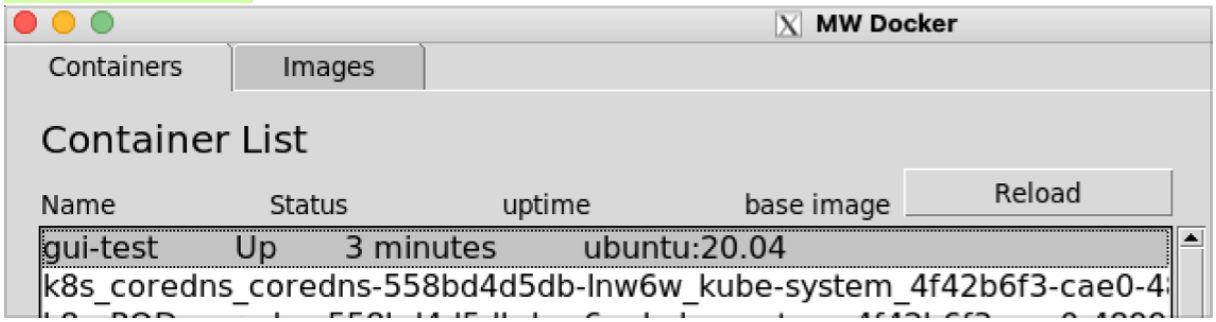


컨테이너 Start

Containers 탭 에서 원하는 컨테이너 Select 한 이후 Start 버튼 클릭

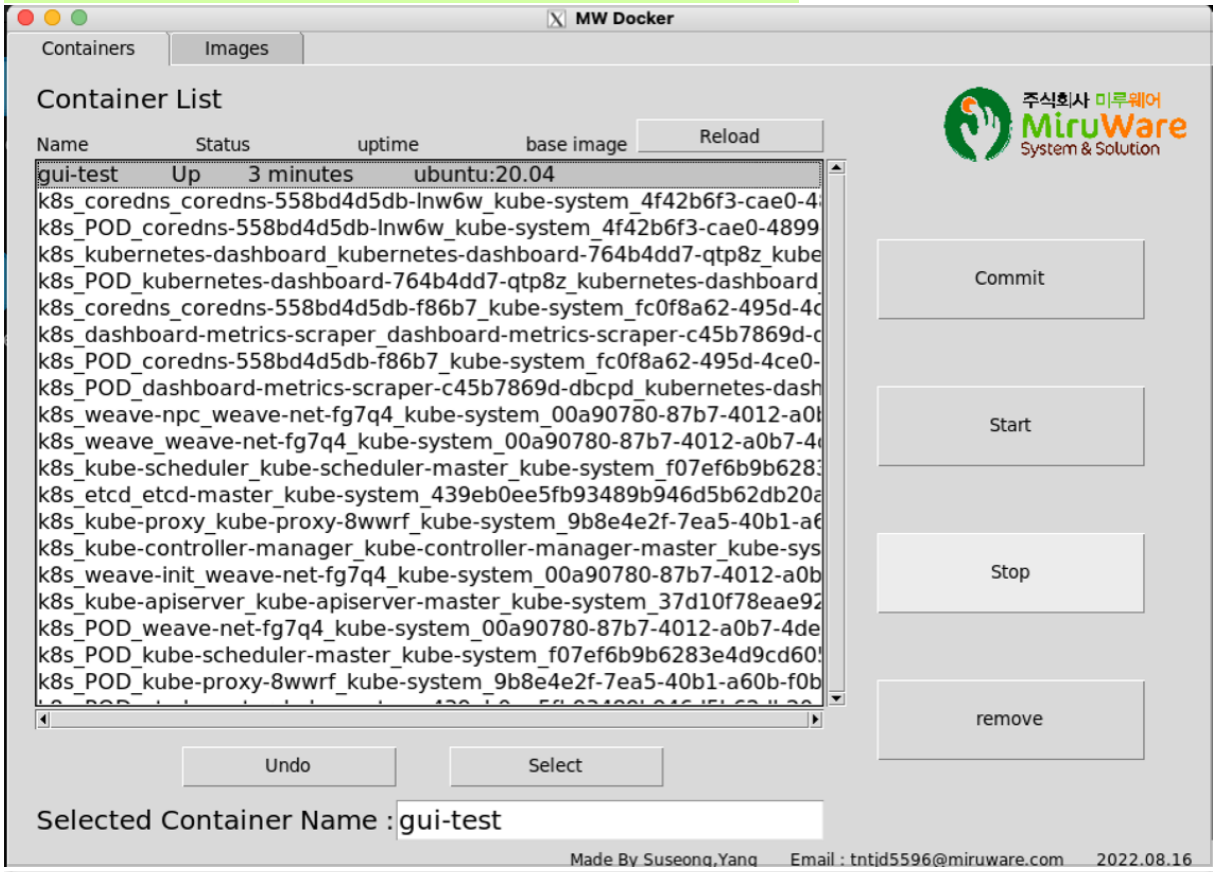


컨테이너 Status 확인

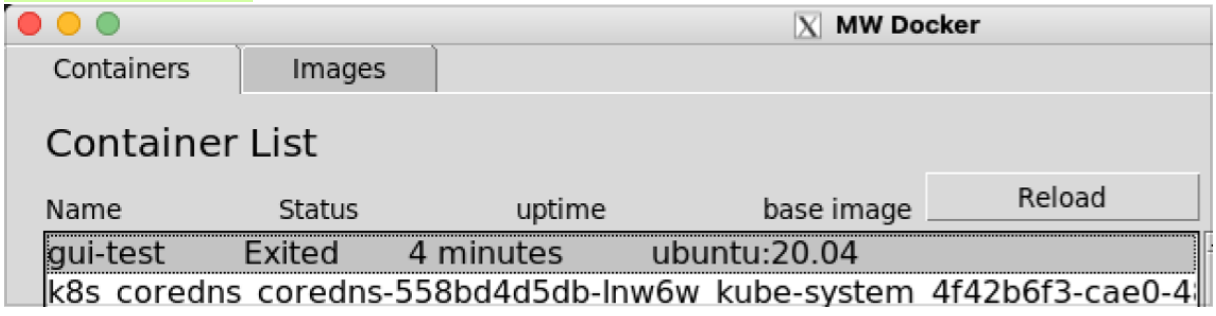


컨테이너 Stop

Containers 탭 에서 원하는 컨테이너 Select 한 이후 Stop 버튼 클릭

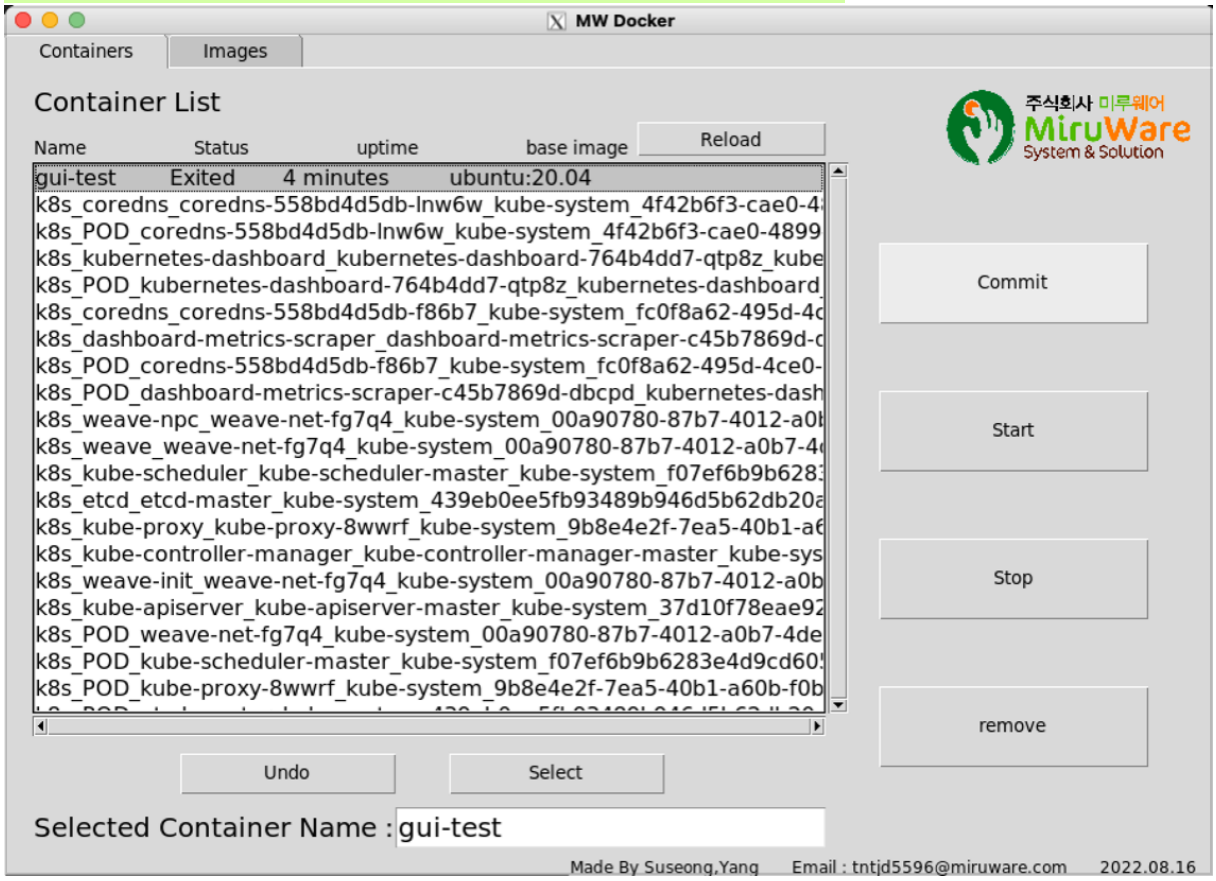


컨테이너 Status 확인

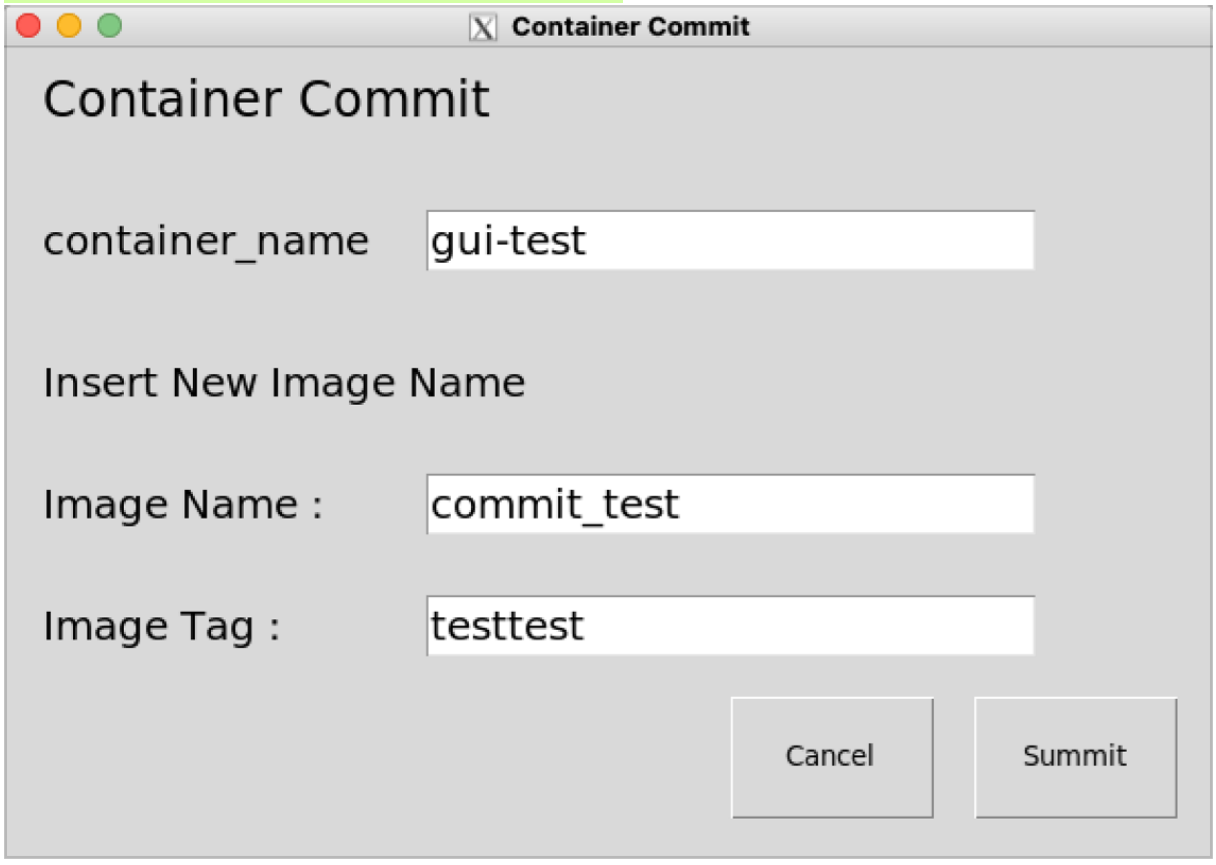


컨테이너 Commit

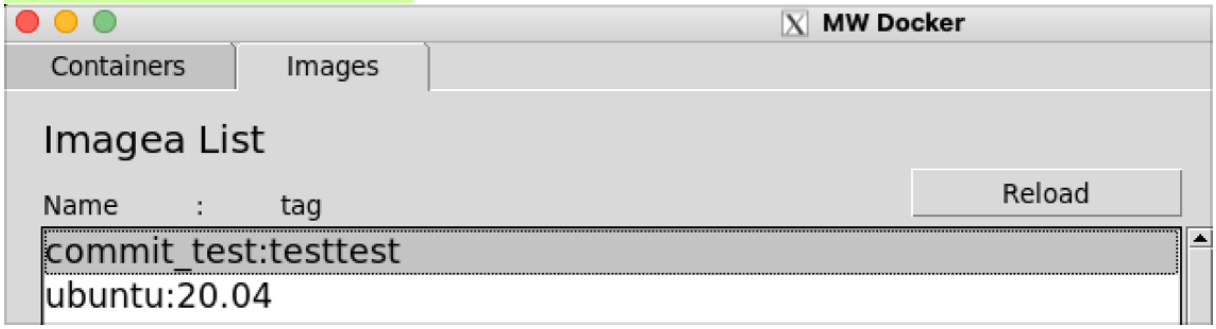
Containers 탭 에서 원하는 컨테이너 Select 한 이후 Commit 버튼 클릭



커밋할 이미지 이름과 태그 입력 후 Summit 버튼 클릭

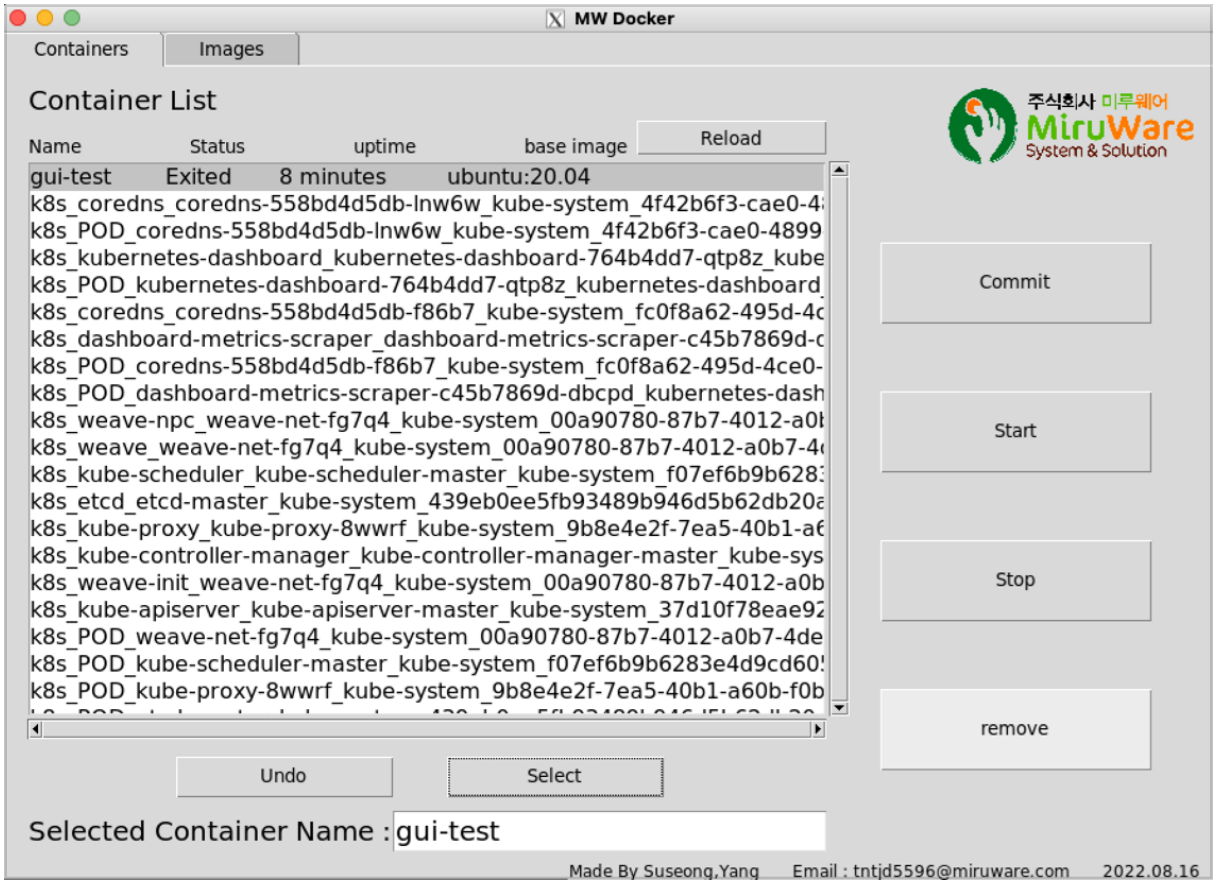


Images 탭에서 커밋한 이미지 확인



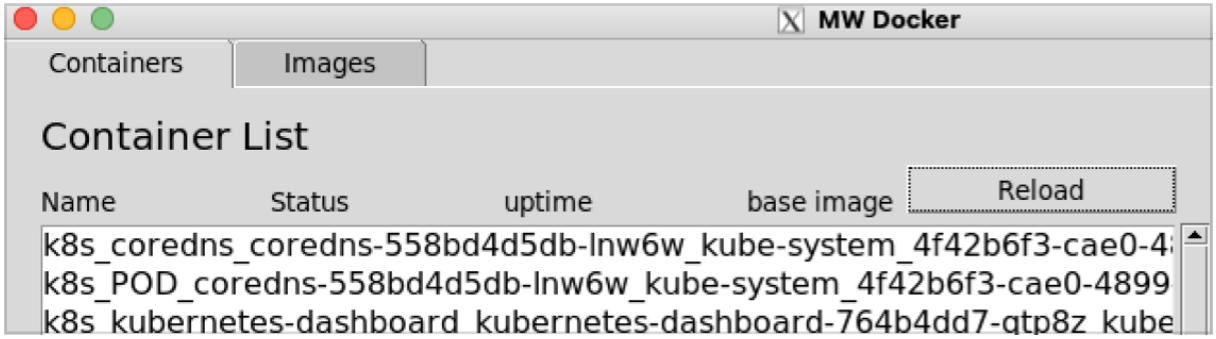
컨테이너 삭제

Containers 탭 에서 원하는 컨테이너 Select 한 이후 Remove 버튼 클릭



확인 단계에서 Summit 버튼 클릭

컨테이너 삭제 확인

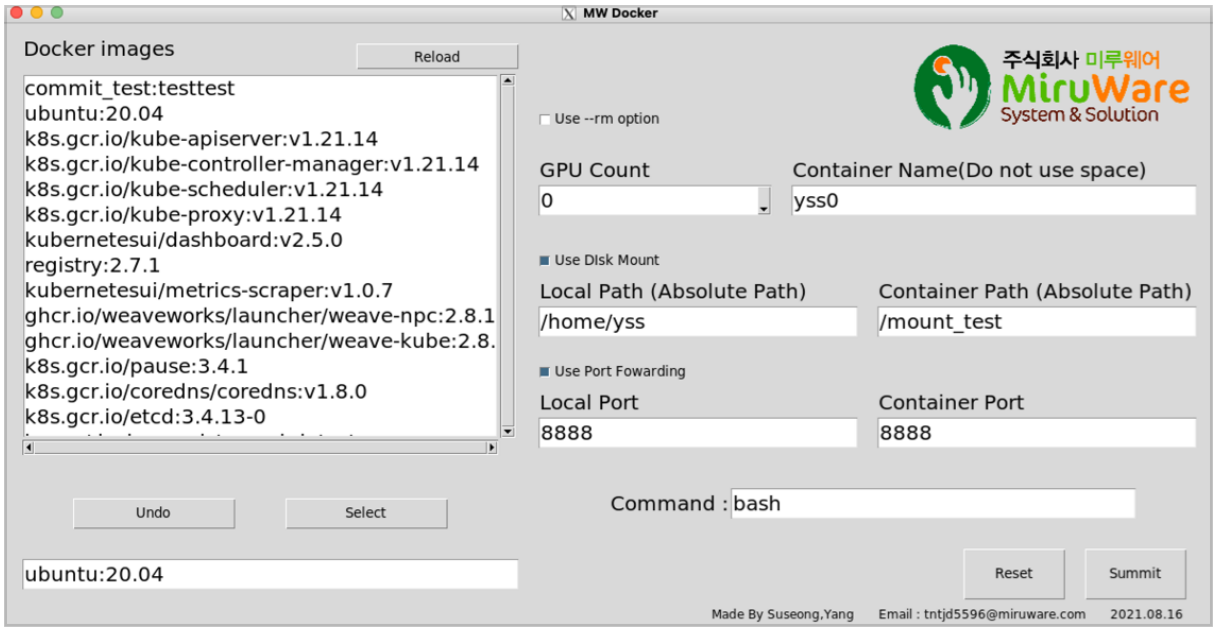


User 용 GUI Controller

실행 방법

```
$ python3 ./docker_gui_controller/Common_user/run.py
```

실행 예시



Docker images

- 사용할 이미지를 선택하여 Select 버튼 클릭

–rm Option

- 컨테이너에 --rm 옵션을 사용할 시 체크박스 체크

GPU Count

- 컨테이너에 사용할 GPU의 개수를 선택
- 자동으로 사용 가능한 최대 개수를 제한함
- CPU 만 사용할 경우 GPU Count를 0으로 선택

Container Name

- 컨테이너의 이름을 지정
- 코드 실행 시 계정명+Index 로 자동으로 이름이 겹치지 않게 지정됨

Mount Option

- 컨테이너에 디스크를 마운트 할 경우 체크박스 체크
- 로컬 디렉토리 경로와 컨테이너 디렉토리 경로를 입력하여 마운트

Port Fowarding Option

- 컨테이너에 포트포워딩을 할 경우 체크박스 체크
- 로컬 포트와 컨테이너 포트를 입력하여 포트포워딩

Command

- 컨테이너 실행 시 실행할 Command를 입력

실행 결과

```
yss@master:~/mw_ost_tools/docker_gui_controller/Common_user$ python3 run.py
sh: 1: nvidia-smi: not found
root@88d837e35a07:/#
```

Mount 동작 확인

컨테이너에서 마운트 포인트에 aaa.txt 파일 생성

```
root@88d837e35a07:/# cd /mount_test && touch aaa.txt
root@88d837e35a07:/mount_test# ls
>>> aaa.txt  mw_ost_tools  yss-kubeconfig
```

로컬 마운트 포인트에 파일 생성 확인

```
$ ls /home/yss
>>> aaa.txt  mw_ost_tools  yss-kubeconfig
```

Trouble Shooting

Docker 권한이 없는 경우

```
Got permission denied while trying to connect to the Docker daemon socket at
unix:///var/run/docker.sock: Get "http://%2Fvar%2Frun%2Fdocker.sock/v1.24/
images/json": dial unix /var/run/docker.sock: connect: permission denied
```

Tool 을 사용하는 계정을 Docker 그룹에 추가하여 해결

```
$ sudo usermod -aG docker $USER
```

Nvidia driver 가 없는 경우

```
sh: 1: nvidia-smi: not found
```

GPU 서버가 아닌 경우 무시해도 상관 없고

GPU 서버인 경우 NVIDIA Driver 를 설치하여 해결