

```
#!/bin/bash
```

```
# this is the install script
```

```
# install_script = "/opt/mcs/api/rundocker.sh"
```

```
# called on boot.
```

```
# this is the refresh script called from ssm for a refresh
```

```
# #refresh_script = "/opt/mcs/api/docker-boot.sh"
```

```
# file not found
```

```
#
```

```
pwd
```

```
ls -latr
```

```
. ./env # for secrets
```

```
set -e # stop on any error
```

```
#export ROOT="" # empty
```

```
export WORKSOURCE="/opt/mcs/api"
```

```
adduser --disabled-password --gecos "" mcs --home "/home/mcs" || echo ignore
```

```
git config --global --add safe.directory "/opt/mcs"
```

```
git config --global --add safe.directory "/opt/mcs-memory"
```

```
cd "/opt/mcs/" || exit 1 # "we need mcs"
```

```
git log -2 --patch | head -1000
```

```
mkdir -p "/var/mcs/agent_workspace/"
```

```
mkdir -p "/home/mcs"
```

```
cd "/opt/mcs/" || exit 1 # "we need mcs"
```

```
mkdir -p "/var/mcs/logs"
```

```
chown -R mcs:mcs "/var/mcs/" "/home/mcs" "/opt/mcs"
```

```
#if [ -f "/var/mcs/agent_workspace/boot_fast.sh" ];
```

```
#then
```

```
#  chmod +x "/var/mcs/agent_workspace/boot_fast.sh" || echo faild
```

```
#  # user install but do not start
```

```
#  su -c "bash -e -x /var/mcs/agent_workspace/boot_fast.sh" mcs
```

```
#fi
```

```
cd "/opt/mcs/" || exit 1 # "we need mcs"
```

```
mkdir -p "/var/run/mcs/secrets/"
```

```
mkdir -p "/home/mcs/.cache/huggingface/hub"
```

```
set +x
```

```
OPENAI_KEY=$(aws ssm get-parameter --name "mcs_openai_key" | jq .Parameter.Value -r )
```

```
export OPENAI_KEY
```

```
echo "OPENAI_KEY=${OPENAI_KEY}" > "/var/run/mcs/secrets/env"
```

```
set -x
```

```
## append new homedir

# check if the entry exists already before appending pls
if ! grep -q "HF_HOME" "/var/run/mcs/secrets/env"; then
    echo "HF_HOME=/home/mcs/.cache/huggingface/hub" >> "/var/run/mcs/secrets/env"
fi

if ! grep -q "^HOME" "/var/run/mcs/secrets/env"; then
    echo "HOME=/home/mcs" >> "/var/run/mcs/secrets/env"
fi

if ! grep -q "^HOME" "/var/run/mcs/secrets/env"; then
# attempt to move the workspace
    echo "WORKSPACE_DIR=${STATE_DIRECTORY}" >> "/var/run/mcs/secrets/env"
fi

# setup the systemd service again
cp "${WORKSOURCE}/nginx/site.conf" /etc/nginx/sites-enabled/default
cp "${WORKSOURCE}/systemd/mcs-docker.service" /etc/systemd/system/mcs-docker.service
grep . -h -n /etc/systemd/system/mcs-docker.service

chown -R mcs:mcs /var/run/mcs/

mkdir -p /opt/mcs/api/agent_workspace/try_except_wrapper/

chown -R mcs:mcs /opt/mcs/api/

# always reload
```

```
# might be leftover on the ami,  
systemctl stop swarms-unicorn || echo ok  
systemctl disable swarms-unicorn || echo ok  
rm /etc/systemd/system/swarms-unicorn.service
```

```
systemctl daemon-reload  
systemctl start mcs-docker || journalctl -xeu mcs-docker  
systemctl enable mcs-docker || journalctl -xeu mcs-docker  
systemctl enable nginx  
systemctl start nginx
```

```
journalctl -xeu mcs-docker | tail -200 || echo oops  
systemctl status mcs-docker || echo oops2
```

```
# now after mcs is up, we restart nginx
```

```
HOST="localhost"
```

```
PORT=8000
```

```
while ! nc -z $HOST $PORT; do
```

```
    sleep 1
```

```
    echo -n "."
```

```
done
```

```
echo "Port ${PORT} is now open!"
```

```
systemctl restart nginx
```