链滴

docker-machine 使用（二）

作者：yanxiaonan
原文链接：https://ld246.com/article/1548145699745
来源网站：链滴

docker-machine 二

修改创建虚拟机的位置

环境变量添加 MACHINE_STORAGE_PATH 的值为新的路径

删除 c:/user/用户名/.docker 目录

找到 Docker Toolbox 的安装目录，比如我的是 D:\softSetUp\Docker Toolbox，这个目录下面有个 tart.sh,用 bash 执行这个命令就可以创建 docker 虚拟机，此时虚拟机就是创建在我们设置的目录

但是这个脚本我们只能创建名字为 default 的虚拟机，很多东西是不能定义的，我这里改了一下脚本以自定义几个参数：

```
#!/bin/bash

trap '[ "$?" -eq 0 ] || read -p "Looks like something went wrong in step ´$STEP´... Press any key to continue..."' EXIT

#Quick Hack: used to convert e.g. "C:\Program Files\Docker Toolbox" to "/c/Program Files/Docker Toolbox"
win_to_unix_path(){
    wd="$(pwd)"
    cd "$1"
        the_path="$(pwd)"
    cd "$wd"
    echo $the_path
}

while getopts ":i:n:d:" opt
do
    case $opt in
        i)
        echo "iso-path=$OPTARG"
        ISO_PATH=${OPTARG}
        ;;
        n)
        echo "docker-vm-name=$OPTARG"
        VM=${OPTARG}
        ;;
        d)
        echo "disk-size=$OPTARG"
        DISK_SIZE=${OPTARG}
        ;;
        ?)
        echo "未知参数"
        exit 1;;
    esac
done

#镜像路径
if [ ! $ISO_PATH ]; then
  ISO_PATH=""
```

原文链接：docker-machine 使用（二）

```
else
  ISO_PATH="--virtualbox-boot2docker-url=${ISO_PATH}"
fi
#虚拟机路径
if [ ! $VM_DIR ]; then
  VM_DIR=""
else
  VM_DIR="--storage-path=${VM_DIR}"
fi
#docker名字
if [ ! $VM ]; then
  VM="default"
fi
#镜像路径
if [ ! $DISK_SIZE ]; then
  DISK_SIZE=""
else
  DISK_SIZE="--virtualbox-disk-size ${DISK_SIZE}"
fi

# This is needed  to ensure that binaries provided
# by Docker Toolbox over-ride binaries provided by
# Docker for Windows when launching using the Quickstart.
export PATH="$(win_to_unix_path "${DOCKER_TOOLBOX_INSTALL_PATH}"):$PATH"
#VM=${DOCKER_MACHINE_NAME-default}
DOCKER_MACHINE="${DOCKER_TOOLBOX_INSTALL_PATH}\docker-machine.exe"

STEP="Looking for vboxmanage.exe"
if [ ! -z "$VBOX_MSI_INSTALL_PATH" ]; then
  VBOXMANAGE="${VBOX_MSI_INSTALL_PATH}VBoxManage.exe"
else
  VBOXMANAGE="${VBOX_INSTALL_PATH}VBoxManage.exe"
fi

echo "VBOXMANAGE管理器---------- $VBOXMANAGE"

BLUE='\033[1;34m'
GREEN='\033[0;32m'
NC='\033[0m'

#clear all_proxy if not socks address
if  [[ $ALL_PROXY != socks* ]]; then
  unset ALL_PROXY
fi
if  [[ $all_proxy != socks* ]]; then
  unset all_proxy
fi

if [ ! -f "${DOCKER_MACHINE}" ]; then
  echo "Docker Machine is not installed. Please re-run the Toolbox Installer and try again."
  exit 1
fi

if [ ! -f "${VBOXMANAGE}" ]; then
```

```
  echo "VirtualBox is not installed. Please re-run the Toolbox Installer and try again."
  exit 1
fi

"${VBOXMANAGE}" list vms | grep \""${VM}"\" &> /dev/null
VM_EXISTS_CODE=$?

set -e

STEP="Checking if machine $VM exists"
if [ $VM_EXISTS_CODE -eq 1 ]; then
  "${DOCKER_MACHINE}" rm -f "${VM}" &> /dev/null || :
  rm -rf ~/.docker/machine/machines/"${VM}"
  #set proxy variables if they exists
  if [ "${HTTP_PROXY}" ]; then
    PROXY_ENV="$PROXY_ENV --engine-env HTTP_PROXY=$HTTP_PROXY"
  fi
  if [ "${HTTPS_PROXY}" ]; then
    PROXY_ENV="$PROXY_ENV --engine-env HTTPS_PROXY=$HTTPS_PROXY"
  fi
  if [ "${NO_PROXY}" ]; then
    PROXY_ENV="$PROXY_ENV --engine-env NO_PROXY=$NO_PROXY"
  fi
  echo "创建脚本 ${DOCKER_MACHINE}" $VM_DIR create -d virtualbox --engine-registry-mirro
=https://jdrq8x78.mirror.aliyuncs.com $ISO_PATH $DISK_SIZE $PROXY_ENV "${VM}"
  "${DOCKER_MACHINE}" $VM_DIR create -d virtualbox --engine-registry-mirror=https://jdrq
x78.mirror.aliyuncs.com $ISO_PATH $DISK_SIZE $PROXY_ENV "${VM}"
fi

STEP="Checking status on $VM"
echo "开始检查状态 ------------"
VM_STATUS="$( set +e ; "${DOCKER_MACHINE}" status "${VM}" )"
echo "VM_STATUS状态 ------------ ${VM_STATUS}"

if [ "${VM_STATUS}" != "Running" ]; then
  "${DOCKER_MACHINE}" start "${VM}"
  yes | "${DOCKER_MACHINE}" regenerate-certs "${VM}"
fi

STEP="Setting env"

echo "设置环境"

echo "设置环境111--- $("${DOCKER_MACHINE}" env --shell=bash --no-proxy "${VM}" | sed -e
s/export/SETX/g" | sed -e "s/=/ /g")" &> /dev/null #for persistent Environment Variables, avai
able in next sessions
echo "设置环境2222--- $("${DOCKER_MACHINE}" env --shell=bash --no-proxy "${VM}")" #for
ransient Environment Variables, available in current session

eval "$("${DOCKER_MACHINE}" env --shell=bash --no-proxy "${VM}" | sed -e "s/export/SETX/
" | sed -e "s/=/ /g")" &> /dev/null #for persistent Environment Variables, available in next sess
ons
eval "$("${DOCKER_MACHINE}" env --shell=bash --no-proxy "${VM}")" #for transient Environ
ent Variables, available in current session
```

```
STEP="Finalize"
clear
cat << EOF


                ##        .
          ## ## ##       ==
       ## ## ## ## ##    ===
    /"""""""""""""""""\___/ ===
~~~ {~~ ~~~~ ~~~ ~~~~ ~~~ ~ /  ===- ~~~
    \_____ o           __/
     \    \         __/
      \____\_______/

EOF
echo -e "${BLUE}docker${NC} is configured to use the ${GREEN}${VM}${NC} machine with IP ${GREEN}$("${DOCKER_MACHINE}" ip ${VM})${NC}"
echo "For help getting started, check out the docs at https://docs.docker.com"
echo
echo
#cd #Bad: working dir should be whatever directory was invoked from rather than fixed to the Home folder

docker () {
  MSYS_NO_PATHCONV=1 docker.exe "$@"
}
export -f docker

if [ $# -eq 0 ]; then
  echo "Start interactive shell"
  exec "$BASH" --login -i
else
  echo "Start shell with command"
  exec "$BASH" -c "$*"
fi
```

参数说明：

-i boot2docker.iso 的路径

-n 虚拟机的名字

-d 虚拟机的硬盘大小

已经加入了阿里镜像加速

用法：

./start.sh -i boot2docker -n mydocker -d 30000