ShipShuk

Prepared by

Esabil Bulbul

1. Index

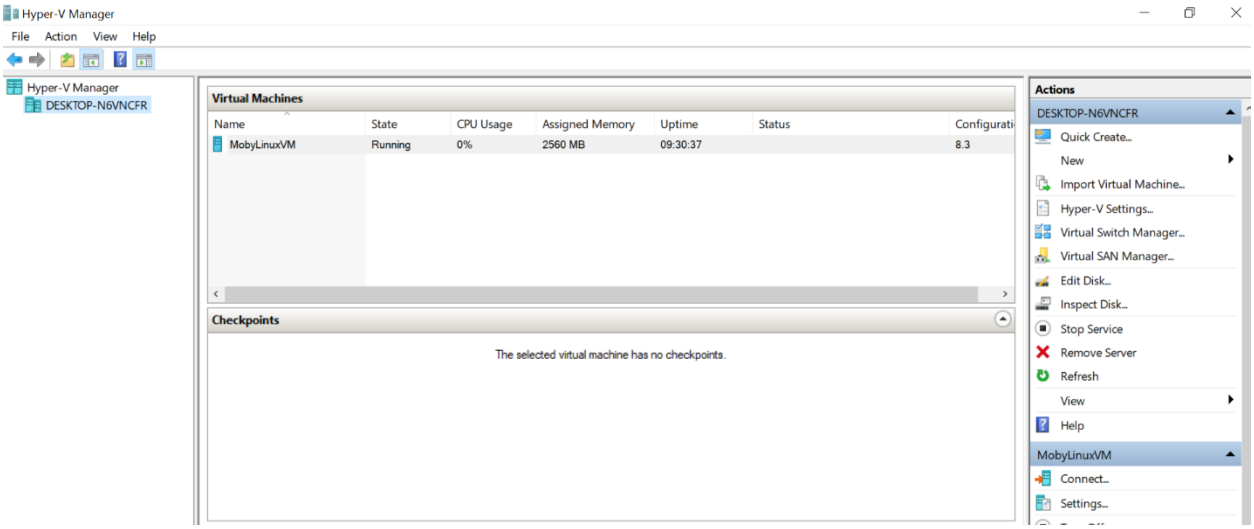
Contents

2. Terminology

Bid	Offering for ads
CPM	Cost Per Impression. This is the bidding cost per 1k view aka Mille

3. Installation

This document explains how docker can be installed on windows machine for mac computers. Once the docker for windows installed the next needs to be done is to run Hyper-V (MoyLinuxVM) virtual machine as Docker is run on a Linux VM on windows. This is where things got starting to spiral for mac & windows machine. As Hyper-V requires hardware virtualization to be enabled, Mac computers are not coming with BIOS as the conventional PCs. MAC uses its own EFI at least they start using it after sometime (guess later 2012 versions)



The screenshot above is Hyper-V screen. The item MobyLinuxVM won't start after first install of Docker for Windows. We will do some manual operations for it explained in following statements.

To enable virtualization on mac pc for windows side is somewhat tricky operation. You may ended up crashing your harddrives and recovery your OSX side as well. However, luckily we managed to activated the virtualization from MAC side. So follow the following steps carefully.

General Steps

1. Deactivate Mac Integrity on Recovery Mode using by Terminal (Cmd + R on restart) gets you to HD Recovery of Mac
2. Download refind
3. Change redif.config-sample file as described on the link
4. Run redif executable to activate the changes
5. Restart to windows

The link for more details

https://apple.stackexchange.com/questions/120361/how-to-turn-on-hardware-virtualization-on-late-2013-macbook-pro-for-windows-8-1

go to the section

12

EDIT: I found a better way to get this working instead of the boot dance originally suggested (it's below for reference). Basically set `enable_and_lock_vmx true` in rEFInd and that's it. Details below ...

Suggested method

1. Disable macOS System Integrity Protection/SIP (Reboot Mac, hold down Command + R keys, at "OS X Utilities Utilities" pick "Terminal" menu item -> type in terminal `csrutil disable; reboot`)
2. Back inside macOS after the reboot, get rEFInd and extract it anywhere (desktop, downloads etc)
3. Open a terminal window, cd to where you extracted it and edit via `sudo nano refind/refind.conf-sample`
4. Uncomment `enable_and_lock_vmx` and set to `true` i.e. the whole line should read `enable_and_lock_vmx true` . **<= This is what really fixes the issue!**
5. **[optional]** While here, change the timeout to something quick, like 4 or 5 i.e. `timeout 4`
6. Install rEFInd by running `./refind-install` from terminal. The sample config you edited is used as the installed config.
7. **[optional]** Enable SIP again. Follow #1 above but run `csrutil enable; reboot` instead

This method is very smooth - it works on normal as well as encrypted disks (FileVault2, VeraCrypt or BitLocker) and really takes a few minutes to install. Best of all, it works in every reboot.

Old method

By default, sudo ./refind-install won't work. You may need to find and download the package. Then when you are under the package folder run the command after changing the conf-sample file.

To install the refind the link is also useful

https://www.deranged-society.com/hard-and-software/how-to-install-refind-0-10-4-on-mac-os-sierra/

The link above has the refind package link to download or here

https://sourceforge.net/projects/refind/

Note: The last step enabling back Mac os, we skipped this step. However, can be done.

IMPORTANT: If this doesn't work, reboot pc and switch to OSX then before loging into OSX restart back to Windows will work. Usually it starts when the machine rebooted however there are cases not starting in that case switch OSX and restart then it will start.

4. Docker Commands

To compile/build Image

This command will be used to compile the docker file ("Dockerfile" – no extension) that will be running on our container. In our case of nsfw, we will be running ubuntu (with the apps installed) on our docker container.

```
docker build  -t ImageName:TagName dir

For example
Docker build -t img_ubuntu .
```

Parameters

t: Tag

img_ubuntu: Name of our image

Should be ended with "." For the directory

For more details: https://www.tutorialspoint.com/docker/building_docker_files.htm

To Run Image on Container

This command will run the image we have compiled in previous step. Say we compiled an ubuntu image when we run the image we will have ubuntu OS running on our container. This command will lead us to ubuntu terminal where we can browse, configure among the files.

```
docker run –it centos /bin/bash

docker run -it img_ubuntu /bin/bash
```

REMEMBER: docker run command creates run a new container each time when it run. To access to an container already existing you should try docker exec -it <container ID> bash

To Remove Image

This command will list the images exist

```
Docker images
```

The following command will remove the image

```
docker rmi Image Image
```

or

```
docker rmi $(docker images -a -q)
```

(q: Image Id)

To force to delete of the image you can use -f option.

```
docker images -f Image Image
```

To Remove Containers

The following command will list the containers

```
docker ps -a
```

To remove the container

```
docker rm ID_or_Name ID_or_Name
```

to force use -f option

```
docker rm -f ID_or_Name ID_or_Name
```

to remove the containers exited

```
docker rm $(docker ps -a -f status=exited -q)
```

to remove all containers

```
docker rm $(docker ps -a -q)
```

to stop containers

```
docker stop $(docker ps -a -q)
```

To Remove Volumes

To list volumes

```
docker volume ls
```

To remove volumes

```
docker volume rm volume_name volume_name
```

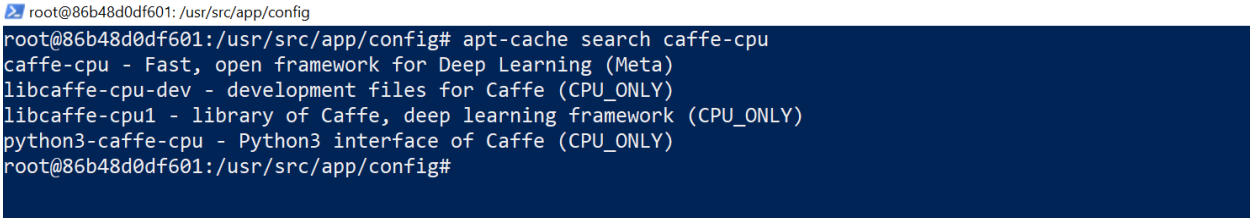
To Check Python Package Versions

On python editor (terminal or on pycharm)

```
>>> import statlib
>>> print statlib.__version__
```

To search for packages on ubuntu

apt-cache search <packagename>



Interactive Build Mode

https://askubuntu.com/questions/523962/how-to-install-a-package-with-apt-without-the-do-you-want-to-continue-y-n-p

check ctrl q / p combination as well

To Commit Changes on Image

This command is used to reflect the changes of packages or other operations made on a container. The command is

Docker commit <containerId> <packagename>

To leave the container

Use **Ctrl-p** + **Ctrl-q** consecutively

Interactive Image Building

Expect command will be used.

This is used to build images interactively. Expect command is a type of sniffer that runs as base application that runs the build package. In expect script, the interactions (questions) from the package installations are sniffed within expect and answered.

The hiearchy of the package is base: Expect then app:Package(dockerfile)

Therefore, the answers implemented ahead of the package installations without need of a user interactions.

For instance;

Say we are installing vim (editor) in our Docker file as

RUN apt-get install vim

The installation will ask if we want to proceed before the final installation

In our expect script

We will implement

Expect -exact “Do you want to continue?”

Send “Y\r”

Once the question detected by expect will send by auto. This way the installation won’t stop. Remember, docker file installations are static in other words not interactive.

One of the problems you can face is the file format if you are a windows user. In that case you might need to install doc2unix to convert to file into unix format as windows has its own line endings which creates “No file or directory found” problem.

To convert the file

$> doc2unix <sourcefile> <targetfile>

To Run Commands at Startup on Attachment

docker exec 9ede507fa042 /bin/bash -c '<command>'

For example;

docker exec 9ede507fa042 /bin/bash -c 'uwsgi --ini /usr/src/app/nsfw/config/nsfw_uwsgi.ini --callable app && service nginx start'

5. Linux Commands

To List Process

Ps ax

To Kill Process

Kill <pid> <pid2>...