Ubuntu-KVM-K8S 研究心得

一、摘要：

本文的目的是提供一個在 Ubuntu 系統環境下，先使用 KVM (Kernel-based Virtual Machine) 建置虛擬主機，再在虛擬主機上架設 Kubernetes 的手動安裝程序說明文件。本文分為兩個部分，第一部分是手動安裝 Kubernetes v1.6.8 版，第二部分是手動安裝 Kubernetes v1.8.x 版。第一部分的重點是在於把手動安裝程序寫成 script 檔案，以節省安裝時間。第二部分的重點則是著重在 Kubernetes 中各項服務之間的憑證認證機制的建置。

二、研究動機：

現在許多企業都在研究如何使用 OpenStack 和 Kubernetes 為企業建置更有彈性且更有效率的私有雲架構。但是，就個人而言，能使用的資源有限，無法像企業一樣使用 OpenStack 為雲端架構提供虛擬主機。於是，我提出先在 Ubuntu 實體主機上安裝 KVM，使用 KVM 建置虛擬主機 (Virtual Machine; VM)，再在 VM 上建置 Kubernetes 的架構，稱為 Ubuntu-KVM-K8S 架構。

本文在 Ubuntu-KVM-K8S 架構上佈署了兩種版本的 Kubernetes，第一種是 v1.6.8，第二種是 v1.8.x 版。建置第一版的原因是剛開始接觸 Kubernetes 時碰到的就是 v1.6.8 版，當時的小組負責人 DannyAJ 研讀了許多網路資料，並整理出一套手動安裝的程序和技術資料。到了我參與此案時自然是以 v1.6.8 版做為起手式練習。在我開始建置 Kubernetes 時，感受到建置程序的複雜和極高的重複操作程序，所以想要把這些程序簡化成腳本 (Shell Scripts) 檔案，以方便重複執行和除錯使用，所以此部分的貢獻在於將 DannyAJ 整理出來的手動架設程序轉成以腳本為基礎的安裝程序。

由於 v1.6.8 版的 Kubernetes 限制了許多服務的版本，例如 Docker 必須使用 v1.12.6。以及經由腳本安裝程序所建置的 Kubernetes 在 flannel 服務上不是很穩定，常需要較久時間才能感知到節點的 IP 位址更換。因此，需要一個更穩定的手動安裝程序。再加上 Kubernetes 從 v1.7.x 版

後大幅改版，所以第二次的 Kubernetes 建置就選擇了 v1.8.x 版本。此部分的建置程序是參考 KaiRen's Blog 的文件，再加上一些環境設定程序而完成的。

因此，接下來的章節將說明如何使用第一部分的腳本建置出 Kubernetes v1.6.8。以及在第二部分中如何利用網路資源建置出 Kubernetes v1.8.x。

三、Kubernetes v1.6.8 建置程序：

3.1 KVM 建置：

KVM 的建置程序可以參考兩個網站：

1. help.ubuntu.com 網站：https://help.ubuntu.com/community/KVM/Installation

2. How to Install KVM and Create Virtual Machines on Ubuntu: https://www.howtogeek.com/117635/how-to-install-kvm-and-create-virtual-machines-on-ubuntu/

3.2 Kubernetes v1.6.8 建置：

所有的腳本建置程序儲存於 scripts_install 目錄下。共有三個子目錄，分別是 common、master 和 node 子目錄。顧名思義，common 目錄下的所有腳本是所有虛擬主機都必須要安裝的程序。master 目錄下的安腳本是為了 master VM 撰寫的安裝程序，而 node 目錄下的腳本是為了 node VM 撰寫的程序。

3.2.1 Common 目錄：

在 common 目錄下共有 4 支腳本，名稱和安裝順序如下：

1. docker-1.12.6_install.sh
2. docker_add-user.sh
3. ca-keys_create.sh

4. ca-keys_scp-to-other-server.sh

第 1 支腳本的用途是安裝 Docker v1.12.6 版，第 2 支腳本是賦予使用者帳號執行 docker 的權限。第 3 支腳本是建立 VM 間溝通用的金鑰憑證對。第 4 支腳本的用途是憑證拷貝到其他 VM。

3.2.2 Master 目錄：

master 目錄的內容如下：

1. etcd_install.sh
2. etcd_setup-config.sh
3. etcd_setup-service.sh
4. etcd_start.sh
5. etcd_test.sh

6. kube-apiserver_install.sh
7. kube-apiserver_setup.sh
8. kube-apiserver_start.sh

9. kube-controller-manager_setup.sh
10. kube-controller-manager_start.sh

11. kube-scheduler_setup.sh
12. kube-scheduler_start.sh

13. flannel_install.sh
14. flannel_setup-service.sh
15. flannel_setup-update-docker.sh
16. flannel_start.sh
17. update_docker.sh

18. haproxy_install.sh
19. haproxy_setup.sh
20. haproxy_start.sh
21. haproxy_test.sh

22. k8s-config.json
23. kubernetes_test.sh

可分為七個部分，分別是 ETCD、kube-apiserver、kube-controller-manager、kube-schedule、Flannel、HAProxy 和 test。 k8s-config.json 是建置 Kubernetes 時的組態設定檔，以 json 格式撰寫。

在 ETCD 部分，執行順序為 etcd_install.sh、etcd_setup-config.sh、etcd_setup-service.sh 和 etcd_start.sh 腳本。而 etcd_test.sh 腳本是測試 ETCD 是否正常運作的。至於詳細的原理可以觀看 Install-Kubernetes-on-Ubuntu-16.04 文件。

在 kube-apiserver_install 和以下部分，僅需依編號順序執行即可。特別注意的是，在 flannel 部分，flannel_setup-update-docker.sh 不需要執行，因為其目的是為了產生 /opt/flannel/update_docker.sh 腳本，因為一些安裝上的問題，我直接提供了 update_docker.sh 腳本，只要將它複製到 /opt/flannel/ 目錄即可，就不需執行 flannel_setup-update-docker.sh 了。另外，k8s-config.json 內的 IP 位址和 VM 名稱 (id) 可以任意改變，但必須與 VM 的 hostname 相符。

3.2.3 Node 目錄：

Node 目錄下的內容如下：

1. node-kubelet_install.sh
2. node-kubelet_setup.sh
3. node-kubelet_start.sh

4. node-kube-proxy_setup.sh
5. node-kube-proxy_start.sh

6. node-flannel_install.sh
7. node-flannel_setup_service.sh
8. node-flannel_setup_update-docker.sh
9. node-flannel_start.sh
10. update_docker.sh

11. k8s-config.json

可以分為四個部分，分別是 kubelet、kube-proxy、flannel 和 k8s-config.json。同樣地，在每一部分均依照 install、setup 和 start 三個步驟執行。其中，在 flannel 一樣不需執行 setup_update-docker.sh，而是將 update_docker.sh 複製到 /opt/flannel/ 目錄即可。

在執行完所有的腳本後，就可以執行 Master 目錄下的 kubernetes_test.sh 腳本。如果順利出現 Kubernetes 的版本資訊和 node 資訊就是安裝成功了，如圖 1。

```
root@Master1:~# kubectl version
Client Version: version.Info{Major:"1", Minor:"6", GitVersion:"v1.6.8", GitCommit:"d74e09bb4e4e7026f45becbed8310665ddcb8514", G
itTreeState:"clean", BuildDate:"2017-08-03T18:12:08Z", GoVersion:"go1.7.6", Compiler:"gc", Platform:"linux/amd64"}
Server Version: version.Info{Major:"1", Minor:"6", GitVersion:"v1.6.8", GitCommit:"d74e09bb4e4e7026f45becbed8310665ddcb8514", G
itTreeState:"clean", BuildDate:"2017-08-03T18:01:01Z", GoVersion:"go1.7.6", Compiler:"gc", Platform:"linux/amd64"}
root@Master1:~#
root@Master1:~# kubectl get nodes -o wide
NAME      STATUS    AGE       VERSION   EXTERNAL-IP   OS-IMAGE             KERNEL-VERSION
node1     Ready     34m       v1.6.8    <none>        Ubuntu 16.04.1 LTS   4.4.0-31-generic
node2     Ready     29m       v1.6.8    <none>        Ubuntu 16.04.1 LTS   4.4.0-31-generic
```

圖 1 Kubernetes 測試成功

四、Kubernetes v1.8.x 建置程序：

4.1 建立虛擬主機：

虛擬主機的建置程序和 3.1 節的內容相同。

4.2 建立無密碼的 root SSH 憑證登錄機制：

可以參考下列連結：

1. How to set up passwordless SSH access for root user
2. How can I set up password-less SSH login?

4.3 手動安裝 Kubernetes v 1.8.x

參考網路文件：Kubernetes v1.8.x 全手動苦工安裝教學