## Lab 1 – Setting up the environment and install Docker-ce

Step 1:	前往docker.install.uhuntu		
500p 1.	https://docs.docker.com/install/linux/docker.co/ubuntu/		
	https://docs.docker.com/install/infux/docker-ce/ubuntu/		
	Get Docker Engine - Community for Ubuntu Estimated reading time: 12 minutes		
	To get started with Docker Engine - Community on Ubuntu, make sure you meet the prerequisites, then install Docker.		
	Prerequisites		
	Docker EE customers		
	To install Docker Enterprise Edition (Docker EE), go to Get Docker EE for Ubuntu instead of this topic.		
	To learn more about Docker EE, see Docker Enterprise Edition.		
	OS requirements		
	To install Docker Engine - Community, you need the 64-bit version of one of these Ubuntu versions:		
	• Disco 19.04		
	<ul> <li>Cosmic 18.10</li> <li>Bionic 18.04 (LTS)</li> </ul>		
	• Xenial 16.04 (LTS)		
	Docker Engine - Community is supported on x86_64 (or amd64), armhf, arm64, s390x (IBM Z), and ppc64le (IBM Bower) architectures		
	Power) architectures.		
Step 2:	移除舊版,確定未安裝可跳過		
	\$ sudo apt-get remove docker docker-engine docker.io containerd runc		
Step 3:	安裝repository資源庫		
	\$ sudo apt-get update		
	\$ sudo apt-get install \		
	ca-certificates \		
	curl		
	gnupg-agent \		
	\$ curl -fsSL https://download.docker.com/linux/ubuntu/gpg   sudo apt-key add –		
	\$ sudo apt-key fingerprint 0EBFCD88		
	\$ sudo add-apt-repository \ "deb [oreh=omd64] https://download.docker.com/linux/ubuntu \		
	(lsb release -cs) \		
	stable"		
Step 4:	安裝Docker Engine – Community Edition		
	\$ sudo apt-get update		
Sten5:	S sudo apt-get install docker-ce docker-ce-cli containerd.io		
Steps.	M武又表灯的Docker-CE Enginge		
	5 Sudo docker run nello-world		

nckucc@nckucc-virtual-machine:~\$ sudo docker run hello-world Unable to find image 'hello-world:latest' locally latest: Pulling from library/hello-world 1b930d010525: Pull complete Digest: sha256:c3b4ada4687bbaa170745b3e4dd8ac3f194ca95b2d0518b417fb47e5879d9b5f Status: Downloaded newer image for hello-world:latest Hello from Docker! This message shows that your installation appears to be working correctly. To generate this message, Docker took the following steps: The Docker client contacted the Docker daemon.
 The Docker daemon pulled the "hello-world" image from the Docker Hub. (amd64) 3. The Docker daemon created a new container from that image which runs the executable that produces the output you are currently reading. 4. The Docker daemon streamed that output to the Docker client, which sent it to your terminal. To try something more ambitious, you can run an Ubuntu container with: \$ docker run -it ubuntu bash Share images, automate workflows, and more with a free Docker ID: https://hub.docker.com/ For more examples and ideas, visit: https://docs.docker.com/get-started/

#### Lab 2 – 啟動 Container

Please refer the following steps to complete the Lab2.

Step 1:	使用 docker run https://docs.docker.com/engine/reference/commandline/run/
	docker run
	Description
	Run a command in a new container
	Usage
	docker run [OPTIONS] IMAGE [COMMAND] [ARG]
Step 2:	啟動一個可以互動的容器 \$ sudo docker run -i -t ubuntu:16.04 /bin/bash

	<pre>nckucc@nckucc-virtual-machine:~\$ sudo docker run -i -t ubuntu:16.04 /bin/bash [sudo] password for nckucc: Unable to find image 'ubuntu:16.04' locally 16.04: Pulling from library/ubuntu e80174c8b43b: Pull complete d1072db285cc: Pull complete 858453671e67: Pull complete 3d07b1124f98: Pull complete Digest: sha256:bb5b48c7750a6a8775c74bcb601f7e5399135d0a06de004d000e05fd25c1a71c Status: Downloaded newer image for ubuntu:16.04 root@7886d3f8b59c:/#</pre>
Step 3:	查看容器內的版本訊息及檔案操作
	\$ cat /proc/version
	File Edit View Search Terminal Help root@7886d3f8b59c:/# cat /proc/version
	Linux version 5.0.0-32-generic (buildd@lgw01-amd64-015) (gcc versi on 7.4.0 (Ubuntu 7.4.0-1ubuntu1~18.04.1)) #34~18.04.2-Ubuntu SMP T hu Oct 10 10:36:02 UTC 2019 root@7886d3f8b59c:/#
	使用 Linux 指令進行互動
	\$ ls \$ 11 \$ apt update
Step 4:	停止互動式容器
	\$ exit
	nckucc@nckucc-virtual-machine: ~
	也可以使用 Ctl+D
Step5:	列出運轉中的容器
	\$ sudo docker ps \$ sudo docker ps -a
	nckucc@nckucc-virtual-machine:~\$ sudo docker ps
	CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
	339856e14288 portainer/portainer "/portainer" 5 hours ago Up 5 hours 0.0.0.0:8000->8000/tcp, 0.0.0:9000->9000/tcp portainer nckucc@nckucc-virtual-machine:~\$
	CONTAINER ID: 容器 ID IMAGE: 使用的映像檔
	COMMAND: 啟動容器時的指令 CREATED: 容器的啟動的時間

	STATUS: 容器狀態 created restarting running removing paused exited dead	
	PORTS: 網路連接的類型 NAMES: 自動分配的容器名稱	
	停止及重啟容器	
	\$ sudo docker stop 容器名稱或 ID	
	\$ sudo docker start 容器名稱或 ID	
	\$ sudo docker restart 容器名稱或 ID	
	查看容器內的紀錄	
	\$ sudo docker logs 容器名稱或 ID	
	進入執行中的容器	
	\$ sudo docker attach 容器名稱或 ID /bin/bash.(離開後會停止容器)	
	\$ sudo docker exec -it 容器名稱或 ID /bin/bash	
	更多參數說明 docker exec ––help	
Step 6:	一一一一一一一一一一一一一一一一一一一一一一一一一一一一一一一一一一一一一	
	\$ sudo docker rm –f 容器名稱或 ID	
	將所有狀態是停止的容器刪除掉	
	\$ sudo docker prune	

#### Lab 3 – 匯出及匯入 Container

Please refer the following steps to complete the Lab3.

I lease len	if the following steps to complete the Luos.		
Step 1:	使用 docker export		
	https://docs.docker.com/engine/reference/commandline/export/		
	docker export		
	Description		
	Export a container's filesystem as a tar archive		
	Usage		
	docker export [OPTIONS] CONTAINER		
	Options		
	Name, shorthand Default Description		
	output , -o Write to a file, instead of STDOUT		
Step 2:	將容器匯出到磁碟空間內		

	\$ sudo docker export 容器名稱或 ID > ./docker/ubuntu.tar
	nckucc@nckucc-virtual-machine:~\$ sudo docker ps CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
	<pre>339856e14288 portainer/portainer "/portainer" 10 hours ago Up 10 hours 0.0.0.8000-&gt;8000/tcp, 0.0.0.9000-&gt;9000/tcp portainer nckucc@nckucc-virtual-machine:~\$ sudo docker export 339856e14288 &gt; ./portainer.tar nckucc@nckucc-virtual-machine:~\$ ll total 79020 drwxr-xr-x 15 nckucc nckucc</pre>
	$\frac{d_{\text{max}}}{d_{\text{max}}} = \frac{11 \text{ ackucc}}{11 \text{ ackucc}} = \frac{1006 \text{ +-} 6 2232 \text{ confin}}{1006 \text{ +-} 6 2232 \text{ confin}}$
Step 3:	將 tar 文件導入為 image
	\$ sudo cat 檔案名稱   docker import 一 映像檔:版本
	此外,也可透過指定的 URL 或者某個目錄來導入
	\$ docker import http://xxxxxxx

# Lab 4 – 啟動一個網頁的 container

Please refer the following steps to complete the Lab4.

啟動一個 Python Flask 的 web container
\$ sudo docker run –d –P training/webapp python app.py
<pre>nckucc@nckucc-virtual-machine:~\$ sudo docker run -d -P training/webapp python app.py Unable to find image 'training/webapp:latest' locally Latest: Pulling from training/webapp:latest uses outdated schema1 manifest format. Please upgrade t o a schema2 image for better future compatibility. More information at https://docs.docker.com /registry/spec/deprecated-schema-v1/ e190868d63f8: Pull complete 909cd34c6fd7: Pull complete 909cd34c6fd7: Pull complete 10bbbc0fc0ff: Pull complete 10bbbc0fc0ff: Pull complete e7ae2541b15b: Pull complete 9dd97ef58ce9: Pull complete a4c1b0cb7af7: Pull complete Digest: sha256:06e9c1983bd6d5db5fba376ccd63bfa529e8d02f23d5079b8f74a616308fb11d Status: Downloaded newer image for training/webapp:latest 17424fd9c152bc71df289a15305c7f08f1992d621c717c04e0174d2e37e1c9cd</pre>
ckucc@nckucc-virtual-machine:~\$ sudo docker psCONTAINER ID IMAGE COMMAND CREATED STATUSPORTS NAMES17424fd9c152 training/webapp "python app.py" 5 minutes ago Up 5 minutes 0.0.0.0:32768->5000/tcp affectionate_ardinghelli339856e14288 portainer/portainer "/portainer" 10 hours ago Up 10 hours 0.0.0.0:8000->8000/tcp, 0.0.0.0:9000->9000/tcp portainerMozilla Firerox© Get Docker Engine - Com × II Deployment — Portainer × Portainer× 127.0.0.1:32768/× 127.0.0.1:32768/***********************************



#### Lab 5 – Image 的使用

Please refer the following steps to complete the Lab5.

Step 1:	啟動容器或下載映像檔時,如果映像檔在本地中不存在,docker engine 就會自動從 docker hub				
	中下載,如果沒有 tag 標記版本的將會下載最近版 latest				
	https://hub.docker.c	<u>com/</u>			
Step 2:	2: 列出本地的映像檔				
	\$ sudo docker	images			
	nckucc@nckucc-virtua REPOSITORY portainer/portainer ubuntu hello-world training/webapp nckucc@nckucc-virtua REPOSITORY 映像權 CREATE 映像檔創建	L-machine:~\$ sudo do TAG latest 16.04 14.04 latest latest L-machine:~\$ 當的資源庫 TAG 映作 時間 SIZE 映像檔力	ocker images IMAGE ID d1219c88aa21 Sf2bf26e3524 2c5e00d77a67 fce289e99eb9 6fae60ef3446 象檔的標籤 IMAGE I	CREATED 30 hours ago 6 days ago 5 months ago 10 months ago 4 years ago	SIZE 80.8MB 123MB 188MB 1.84kB 349MB

Step 3:	下載一個映像檔			
	\$ sudo docker pull 映像檔資源/映像檔名稱:版本			
	<pre>nckucc@nckucc-virtual-machine:~\$ sudo docker pull nvidia/cuda Using default tag: latest latest: Pulling from nvidia/cuda 35c102085707: Pull complete 251f5509d51d: Pull complete 8e829fe70a46: Pull complete 6001e1789921: Pull complete 9f0a21d58e5d: Pull complete 47b91ac70c27: Pull complete 23bff6dcced5: Pull complete 2137cd2bcba9: Pull complete Digest: sha256:68efc9bbe07715c54ff30850aeb2e6f0d0b692af3c8dd40f13c0b Status: Downloaded newer image for nvidia/cuda:latest docker.io/nvidia/cuda:latest</pre>	9179bfc0bc15		
Step 4:	使用指令查詢映像檔資源			
	<pre>\$ sudo docker search xoops docker.to/nvtota/cuda:tatest nckucc@nckucc-virtual-machine:~\$ sudo docker search xoops NAME DESCRIPTION</pre>	STARS		
	OFFICIAL AUTOMATED kujiy/kc-xoops xoops image for kc services	1		
	[OK] jahaulin/xoops Xoops 2.5.10	1		
	[OK] lyshie/php-fpm-xoops      XOOPS 校園網路輕鬆架 (php-fpm, nginx, n 「or1	mariad 1		
	lyshie/lamp-xoops XOOPS 校園網路輕鬆架 (Docker)	1		
	t301000/xoops.easy.dock.php-fpm XOOPS 輕鬆架 in Docker 之 php-fpm	Θ		
	turnkeylinux/xoops-13.0 TurnKey Xoops - Web Application System	0		
	NAME 資源庫名稱 DESCRIPTION 鏡像的描述 OFFICIAL 是否為 do	cker 官方		
Stor 5.	STARS 推薦 AUTOMATED 自動構建			
Step 5:	刪除映像檔			
	\$ SUGO FMI 映像隘凸柟			
	nckucc@nckucc-virtual-machine:~\$ sudo docker rmi hello-world Untagged: hello-world:latest Untagged: hello-world@sha256:c3b4ada4687bbaa170745b3e4dd8ac3f194ca95b2d051 Deleted: sha256:fce289e99eb9bca977dae136fbe2a82b6b7d4c372474c9235adc174167 Deleted: sha256:af0b15c8625bb1938f1d7b17081031f649fd14e6b233688eea3c548399	.8b417fb47e5879d9b5f /5f587e /4a66a3		

## Lab 6-建立自己的 Image 來使用

Please refer the following steps to complete the Lab6.

Step 1:	當我們需要建立自己的映像檔時,有兩個主要的方法		
	1. 從己經建立好的映像檔中加入需要的原件或程式		
	2. 使用 Dockerfile 指令來創建一個新的映像		

Step 2:	在第一種方法中,使用建立好的映像檔來更新,要先創建一個容器
	\$ sudo docker run −i −t ubuntu:16.3 /bin/bash
	對容器中的程式進行更新後退出
	\$ apt-get update
	\$ apt-get upgrade -y \$ exit
	<pre>nckucc@nckucc-virtual-machine:~\$ sudo docker run -i -t ubuntu:16.04 /bin/bash root@0c8bdacfe074:/# apt-get update</pre>
	Get:1 http://security.ubuntu.com/ubuntu xenial-security InRelease [109 kB]
	Get:3 http://security.ubuntu.com/ubuntu xenial-security/main amd64 Packages [985 kB] Get:4 http://security.ubuntu.com/ubuntu xenial-updates ToPelesee [109 kB]
	Get:5 http://archive.ubuntu.com/ubuntu xenial-backports InRelease [107 kB]
	Get:7 http://security.ubuntu.com/ubuntu xental-security/universe amd64 Packages [12.7 kB] Get:7 http://security.ubuntu.com/ubuntu xental-security/universe amd64 Packages [589 kB]
	Get:8 http://archive.ubuntu.com/ubuntu xenial/main amoo4 Packages [1558 kB] Get:9 http://security.ubuntu.com/ubuntu xenial-security/multiverse amd64 Packages [6281 B]
	Get:10 http://archive.ubuntu.com/ubuntu xental/restricted amd64 Packages [14.1 kB] Get:11 http://archive.ubuntu.com/ubuntu xental/universe amd64 Packages [9827 kB]
	Get:12 http://archive.ubuntu.com/ubuntu xenial/multiverse amd64 Packages [176 kB] Get:13 http://archive.ubuntu.com/ubuntu xenial-updates/main amd64 Packages [1363 kB]
	Get:14 http://archive.ubuntu.com/ubuntu xenial-updates/restricted amd64 Packages [13.1 kB] Get:15 http://archive.ubuntu.com/ubuntu xenial-updates/universe amd64 Packages [993 kB]
	Get:16 http://archive.ubuntu.com/ubuntu xenial-updates/multiverse amd64 Packages [19.3 kB] Get:17 http://archive.ubuntu.com/ubuntu xenial-backports/main amd64 Packages [7942 B]
	Get:18 http://archive.ubuntu.com/ubuntu xenial-backports/universe amd64 Packages [8807 B] Fetched 16.1 MB in 21s (758 kB/s)
	Reading package lists Done root@0c8bdacfe074:/# apt-get upgrade
	Reading package lists Done Building dependency tree
	Reading state information Done Calculating upgrade Done
	The following packages will be upgraded: gcc-5-base libstdc++6
	2 upgraded, 0 newly installed, 0 to remove and 0 not upgraded. Need to get 410 kB of archives.
	After this operation, 0 B of additional disk space will be used. Do you want to continue? [Y/n] Y
Step 3:	使用 Docker commit 來將剛剛更新完的容器以副本的方式進行更新
	\$ sudo docker commit -m="updated" -a="redhsu" c8bdacfe074
	redhsu/ubuntu:v2
	使用 image 指令查詢新建的映像檔
	\$ sudo docker images
	<pre>nckucc@nckucc-virtual-machine:~\$ sudo docker commit -m="updated" -a="redhsu" 0c8bdacfe074 redhsu/ubuntu:v2 sha256:d207e3d6db368b2cc7b2660d225f699867f598a82acf4fba3f1031738d3dedbe</pre>
	nckuccenckucc-virtual-machine:~\$ sudo docker images REPOSITORY TAG IMAGE ID CREATED SIZE
	portainer/portainer latest d1219c88aa21 32 hours ago 80.8MB
	httpd latest d3017f59d5e2 7 days ago 165MB nvidia/cuda latest 946e78c7b298 2 months ago 2.83GB
	ubuntu 14.04 2c5e00d77a67 5 months ago 188MB training/webapp latest _ 6fae60ef3446 4 years ago 349MB
Step 4:	使用新版本的映像檔來創建容器
	\$ sudo docker run -t -l redhsu/ubuntu:v2 /bin/bash

	<pre>nckucc@nckucc-virtual-machine:~\$ sudo docker run -i -t redhsu/ubuntu:v2 /bin/bash root@ece4ea93abfb:/# apt-get update Get:1 http://security.ubuntu.com/ubuntu xenial-security InRelease [109 kB] Hit:2 http://archive.ubuntu.com/ubuntu xenial-updates InRelease [109 kB] Get:4 http://archive.ubuntu.com/ubuntu xenial-backports InRelease [107 kB] Fetched 325 kB in 3s (82.2 kB/s) Reading package lists Done root@ece4ea93abfb:/# apt-get upgrade Reading package lists Done Building dependency tree Reading state information Done Calculating upgrade Done 0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded. root@ece4ea93abfb:/# exit exit</pre>
Step 5:	使用 docker build 及 dockerfile 來建立一個新的映像檔
	\$ vi Dockerfile
	FROM centos:6.7
	MAINTAINER red "red.hsu@ap.logicalis.com"
	RUN /bin/echo 'root:123456'  chpasswd
	RUN useradd redhsu
	RUN /bin/echo 'redhsu:123456'  chpasswd
	RUN /bin/echo -e "LANG=\"en_US.UTF-8\"" >/etc/default/local
	EXPOSE 22
	EXPOSE 80
	CMD /usr/sbin/sshd –D
	\$ sudo docker build -t redhsu/centos:6.7 .

	<pre>nckucc@nckucc-virtual-machine:~\$ sudo docker build -t redhsu/centos:6.7 .</pre>
	Sending build context to Docker daemon 326MB
	Step 1/9 : FROM centos:6.7
	> 9f1de3c6ad53
	Step 2/9 : MAINTAINER rednsu "red.nsu@ap.logiCalis.com"
	Step 3/9 : RUN /bip/echo 'cont:123456' lchpasswd
	> Using cache
	> 499473aee2c7
	Step 4/9 : RUN useradd redhsu
	> Using cache
	> b4fe3ae02dc2
	Step 5/9 : RUN /bin/echo 'redhsu:123456'  chpasswd
	> Running in bcbf1828afc4
	Removing intermediate container bcbf1828afc4
	> 80a8Ce20eTC2
	Step 0/9 : ROW /Dtil/echo -e LAWG=\ en_05.01F-8\ > /etc/delautt/tocat
	Removing intermediate container 29b616bcb892
	> 982e559d155b
	Step 7/9 : EXPOSE 22
	> Running in fc458d67e2b0
	Removing intermediate container fc458d67e2b0
	> eb53be1e38d5
	Step 8/9 : EXPOSE 80
	> Running in 40a4e288b750
	Removing intermediate container 40a4e288b750
	> C9DDZDala9ZT
	Step 9/9 : CMD / USI/SSID -D
	Removing intermediate container 04cd98ab099a
	> 8dd6473764c6
	Successfully built 8dd6473764c6
	Successfully tagged redhsu/centos:6.7
<u> </u>	
Step 6:	查詢映像檔狀態並創建容器
	\$ sudo docker images
	\$ sudo docker run -t -l redhsu/centos:6.7 /bin/bash
	nckucc@nckucc.victual.machine:"\$ sudo docker images
	REPOSITORY TAG IMAGE ID CREATED SIZE
	redhsu/centos 6.7 8dd6473764c6 3 minutes ago 191MB
	redhsu/ubuntu v2 d207e3d6db36 32 minutes ago 151MB
	portainer/portainer latest d1219c88aa21 33 hours ago 80.8MB
	ubuntu 16.04 ST2DT26e3524 6 days ago 123MB httpd latest d3017f59d5e2 7 days ago 165MP
	nvidia/cuda latest 946e78c7b298 2 months ago 2.83GB
	ubuntu 14.04 2c5e00d77a67 5 months ago 188MB
	centos 6.7 9f1de3c6ad53 7 months ago 191MB
	training/webapp latest 6fae60ef3446 4 years ago 349MB
	froot@78c3e28e3500 /]# id_redbsu
	uid=500(redhsu) gid=500(redhsu) groups=500(redhsu)
	[root@78c3e28e3500 /]#

## Lab 7 – 使用 DockerHub 來存放自建的映像檔

Please refer the following steps to complete the Lab7.

Step 1:	當我們需要使用 DockerHub 來存放自己的映像檔時,可以	人放在 Public Warehouse 裡
	1. 在 <u>https://hub.docker.com</u> 註冊一個帳號	
	2. 使用申請完成的帳號及密碼,我們就可以上傳或下	載帳號下的映像檔
	\$ sudo docker login	
Step 2:	nckucc@nckucc-virtual-machine:~\$ sudo docker login         [sudo] password for nckucc:         Login with your Docker ID to push and pull images from         Username: redzhong         Password:         WARNING! Your password will be stored unencrypted in /h         Configure a credential helper to remove this warning. S         https://docs.docker.com/engine/reference/commandline/log         Login Succeeded         nckucc@nckucc-virtual-machine:~\$         在 dockerhub 上找 ubuntu 的映像檔,下載到本地	Docker Hub. If you don't have a D nome/nckucc/.docker/config.json. see ngin/#credentials-store
	\$ sudo docker serach ubuntu	
	\$ sudo docker pull ubuntu	
	<pre>nckucc@nckucc-virtual-machine:~\$ sudo docker search ubunt NAME ubuntu dorowu/ubuntu-desktop-lxde-vnc rastasheep/ubuntu-sshd consol/ubuntu-xfce-vnc ubuntu-upstart ansible/ubuntu14.04-ansible neurodebian land1internet/ubuntu-16-nginx-php-phpmyadmin-mysql-5 ubuntu-debootstrap i386/ubuntu land1internet/ubuntu-16-apache-php-5.6 land1internet/ubuntu-16-apache-php-7.0 ppc64le/ubuntu land1internet/ubuntu-16-nginx-php-phpmyadmin-mariadb-10 land1internet/ubuntu-16-nginx-php-5.6 land1internet/ubuntu-16-nginx-php-5.6 land1internet/ubuntu-16-nginx-php-5.6-wordpress-4 land1internet/ubuntu-16-nginx-php-7.1 darksheer/ubuntu land1internet/ubuntu-16-nginx-php-7.0 pivotaldata/ubuntu land1internet/ubuntu-16-shd pivotaldata/ubuntu land1internet/ubuntu-16-shd pivotaldata/ubuntu land1internet/ubuntu-16-shd pivotaldata/ubuntu land1internet/ubuntu-16-php-7.1 pivotaldata/ubuntu-gpdb-dev nckucc@nckucc-virtual-machine:~\$ sudo docker pull ubuntu Using default tag: latest latest: Pulling from library/ubuntu 7dbc47eeb70: Pull complete 8c3b70e39044: Pull complete 8c3b70e39044: Pull complete Digest: sha256:6e9f67fa63b0323e9a1e587fd71c561ba48a034504 Status: Downloaded newer image for ubuntu:latest docker.io/library/ubuntu:latest</pre>	DESCRIPTION Ubuntu is a Debian-based Linux Docker image to provide HTML5 Dockerized SSH service, built Ubuntu container with "headles Upstart is an event-based repl Ubuntu 14.04 LTS with ansible NeurODebian provides neuroscie ubuntu-16-nginx-php-phpmyadmin debootstrapvariant=minbase Ubuntu is a Debian-based Linux ubuntu-16-apache-php-5.6 ubuntu is a Debian-based Linux ubuntu-16-nginx-php-phpmyadmin ubuntu-16-nginx-php-5.6 ubuntu-16-nginx-php-5.6 ubuntu-16-nginx-php-7.0 Base Ubuntu Image Updated h ubuntu-16-nginx-php-7.0 A quick freshening-up of the b ubuntu-16-shd Ubuntu 16.04 image for GPDB co ubuntu with smartentry ubuntu-16-php-7.1 Ubuntu images for GPDB develop
Step 3:		ckerHub
	\$ sudo docker tag ubuntu:latest 使用者名稱/ub	untu:18.04
	使用 image 指令查詢新建的映像檔	

ickucc@nckucc-virtual ickucc@nckucc-virtual locker: 'imakes' is n iee 'dockerhelp' ickucc@nckucc-virtual iEPOSITORY	-machine:~\$ sudo -machine:~\$ sudo ot a docker comma	docker tag ubuntu:lat docker imakes	est redzhong/ubuntu:18	3.04
redhsu/centos redhsu/ubuntu portainer/portainer pbuntu pbuntu redzhong/ubuntu rttpd nvidia/cuda pbuntu rentos rraining/webapp	-machine:~\$ sudo TAG 6.7 v2 latest 16.04 latest 18.04 latest latest latest 14.04 6.7 latest	nd. docker images IMAGE ID 8dd6473764c6 d207e3d6db36 d1219c8aa21 Sf2bf26e3524 775349758637 775349758637 d3017f59d5e2 946e78c7b298 2c5e00d77a67 9f1de3c6ad53 6fae60ef3446	CREATED 49 minutes ago About an hour ago 34 hours ago 6 days ago 6 days ago 6 days ago 7 days ago 2 months ago 5 months ago 7 months ago 4 years ago	SIZE 191MB 151MB 80.8MB 123MB 64.2MB 64.2MB 165MB 2.83GB 188MB 191MB 349MB
使用 docker push 來 \$ sudo docker <b>ckucc@nckucc-virtual-m</b> he push refers to repos 0b3afb09dc3: Mounted fi c01b5a53aac: Mounted fi c6ac8e5063e: Mounted fi c967c529ced: Mounted fi 8.04: digest: sha256:13	表映像檔推送到 d er push 使用者名 chine:~\$ sudo dock jitory [docker.io/r om library/ubuntu om library/ubuntu om library/ubuntu om library/ubuntu om library/ubuntu	ockerhub 稱/ubuntu:18.04 er push redzhong/ubuntu edzhong/ubuntu]	:18.04 523f849e580a89a685e5d si;	ze: 1152
使用 docker build 及 \$ vi Dockerfi FROM cen MAINTAINER RUN /bin/ RUN usera RUN /bin/ RUN /bin/ EXPOSE 22 EXPOSE 80 CMD /usr	dockerfile 來建立 le tos:6.7 red "red.hsu 'echo 'root:1234 add redhsu 'echo 'redhsu:12 'echo –e "LANG /sbin/sshd –D	z—個新的映像檔 u@ap.logicalis.com'' 56'  chpasswd 3456'  chpasswd =\"en_US.UTF-8\"''	>/etc/default/local	
	cwcc@nckucc-vtrtual         EPOSITORY         edhsu/ubuntu         optainer/portainer         ountu         edhsu/ubuntu         optainer/portainer         ountu         edhsu/ubuntu         ttpd         //idia/cuda         ountu         entos         raining/webapp         使用 docker push 來導         \$ sudo docker         cwc@nckucc-virtual-mail         re push refers to reposobbash09dc3: Mounted fr         cac8e5063e: Mounted fr	cuccenckucc-virtual-machine:-\$ sudo         edhsu/centos       6.7         edhsu/centos       6.7         edhsu/ubuntu       v2         portainer/portainer       latest         pontu       latest         <	Chucequecture         TAG         IMAGE ID           edhsu/centos         6.7         8dd6473764c6           edhsu/centos         16.04         5f2bf26e3524           pointu         latest         775349758637           edhsu/cuda         latest         775349758637           tipd         latest         d3017f59d5e2           vidia/cuda         latest         940e78c7b298           vidia/cuda         latest         940e78c7b298           entos         6.7         9f1de3c6ad53           raining/webapp         latest         6fae60e73446	Characteristics         Sudo docker         Tunges           Checkstor         6.7         8 dd6473754c6         49 minutes ago           Schsu/Uburtu         2.2         d207836db36         About an hour ago           Schsu/Uburtu         16.04         572b726e3524         6 days ago           Suntu         16.04         572b726e3524         6 days ago           Suntu         16.04         5739756637         6 days ago           Suntu         14 test         475349756637         6 days ago           Suntu         14 test         436775945e2         7 days ago           ridia/cuda         1a test         946e78c7b298         2 months ago           entos         6.7         9 fide3c6ad53         7 months ago           etation         f docker push 來裝映像         f dockerhub         \$ sudo docker file           f docker<

	nckucc@nckucc-virt	ual-machine:~\$	sudo docker build -	t_redbsu/centos:6.7	7
	Sending build cont	ext to Docker d	aemon 326MB	e realisa/centos.o./	•
	Step 1/9 : FROM	centos:6.7			
	> 9f1de3c6ad53				
	Step 2/9 : MAINTAI	NER redhsu "	red.hsu@ap.logicali	s.com"	
	> Using cache				
	> 096eb8c04102				
	Step 3/9 : RUN	/bin/echo 'root	:123456'  chpasswd		
	> Using cache				
	> 499473aee2c7				
	Step 4/9 : RUN	useradd redhsu			
	> Using cache				
	> b4fe3ae02dc2				
	Step 5/9 : RUN	/bin/echo 'redh	isu:123456'  chpassw	d	
	> Running in b	cbf1828afc4			
	Removing intermedi	ate container b	ocbf1828afc4		
	> 80a8ce20efc2				
	Step 6/9 : RUN	/bin/echo -e "l	ANG=\"en_US.UIF-8\"	<pre>&gt; /etc/default/lo</pre>	ocal
	> Running in 2	9D616DCD892			
	Removing intermedi	ate container 2	9D616DCD892		
	> 98285590155D	22			
	Step 7/9 . EAFOSE	22 c458d67a2h0			
	Removing intermedi	ate container f	c458d67e2b0		
	> eb53be1e38d5		04300070200		
	Step 8/9 : EXPOSE	80			
	> Running in 4	0a4e288b750			
	Removing intermedi	ate container 4	10a4e288b750		
	> c9bb2ba1a92f				
	Step 9/9 : CMD	/usr/sbin/sshd	-D		
	> Running in 0	4cd98ab099a			
	Removing intermedi	ate container 0	)4cd98ab099a		
	> 8dd6473764c6				
	Successfully built	8dd6473764c6			
	Successfully tagge	d redhsu/centos	:6.7		
Chan (					
Step 6:	查詢映像檔狀態並創	建容器			
	\$ sudo dock	er images			
	φ σάασ άσσι	ter intageo			
	\$ sudo dock	ker run -t -l red	hsu/centos:6.7 /bin/	bash	
	REPOSITORY	TAC TAC	TMACE TD	CREATED	ST7E
	redhsu/centos	6.7	8dd6473764c6	3 minutes ano	191MB
	redhsu/ubuntu	v2	d207e3d6db36	32 minutes ago	151MB
	portainer/portainer	latest	d1219c88aa21	33 hours ago	80.8MB
	ubuntu	16.04	5f2bf26e3524	6 days ago	123MB
	httpd	latest	d3017f59d5e2	7 days ago	165MB
	nvidia/cuda	latest	946e78c7b298	2 Months ago	2.83GB
	centos	6.7	9f1de3c6ad53	7 months ago	191MB
	training/webapp	latest	6fae60ef3446	4 vears ago	349MB
	nckucc@nckucc-virtual	L-machine:~\$ sudo	docker run -t -i red	hsu/centos:6.7 /bin/b	ash
	[root@78c3e28e3500 /]	]# id redhsu			
	uid=500(redhsu) gid=	500(redhsu) group	os=500(redhsu)		
	[root@78c3e28e3500 /]	#			

#### Lab 8 – Docker GUI工具 Portainer

Step 1:       前往Portainer.io         https://github.com/portainer/portainer         userself       userself	I lease leit	a the following steps to compre		
https://github.com/portainer/portainer         https://github.com/portainer/portainer         importainer/inclusion         importainer         inclusion         inclusin         inclusion	Step 1:	前往Portainer.io		
Step 3:         建立管理者報號密碼           127.0.0.1:9000         建立管理者和影響           127.0.0.1:9000         建立管理者和影響           127.0.0.1:9000         正言管理者和影響		https://github.com/portainer/p	ortainer	
Step 2:       快速啟動 Portaner \$ docker volume create portainer_data \$ docker run -d -p 9000:9000 -p 8000:8000name portainerrestart always -v /var/run/docker.sock -v portainer data:/data portainer/portainer         Step 3:       建立管理者帳號密碼 127.0.0.1:9000         Please create the initial administrator user. Username       admin         Password		docker pulls       1.26       08       6 layers       docs pass         Portainer is a lightweight management Ul v         hosts or Swarm clusters).       Portainer is means         can run on any Docker engine (can be depled platforms too).       Portainer allows you to mark more) ! It is compatible with the standalone	And the second s	
S docker volume create portainer_data         \$ docker volume create portainer_data         \$ docker run -d -p 9000:9000 -p 8000:8000name portainerrestart always -v         /var/run/docker.sock:/var/run/docker.sock -v portainer data:/data portainer/portainer         Step 3:       建立管理者帳號密碼         127.0.0.1:9000         Please create the initial administrator user.         Username       admin         Password       x         The password must be at least 8 characters long	Step 2:	快速啟動 Portaner		
\$ docker run -d -p 9000:9000 -p 8000:8000name portainerrestart always -v         /var/run/docker.sock -v portainer data:/data portainer/portainer         Step 3:       建立管理者帳號密碼         127.0.0.1:9000         Please create the initial administrator user.         Username       admin         Password         Confirm password         X The password must be at least 8 characters long	Stop 2.	\$ docker volume create por	tainer_data	
Step 3:       建立管理者帳號密碼         127.0.0.1:9000       Image: Constrained add://data/portained/portaine		\$ docker run -d -p 9000:90	00 -p 8000:8000name portainerrestart always -v	
127.0.0.1:9000  Please create the initial administrator user. Username admin Password Confirm password X The password must be at least 8 characters long	Step 3:	·····································	un/docker.sock -v portainer_data./data portainer/portainer	
127.0.0.1:9000	500p 51			
Please create the initial administrator user.   Username   admin   Password   Confirm password   X The password must be at least 8 characters long		127.0.0.1:9000		
Please create the initial administrator user.   Username   admin   Password   Confirm password   × The password must be at least 8 characters long			portainer.io	
Username admin   Password		Please create the initial administrator use	er.	
Password       Confirm password       × The password must be at least 8 characters long		Username	admin	
Password       Confirm password       × The password must be at least 8 characters long				
Confirm password × The password must be at least 8 characters long		Password		
Confirm password     ×       × The password must be at least 8 characters long				
★ The password must be at least 8 characters long		Confirm password	×	
		★ The password must be at least 8 chara	cters long	
2+ Create user		<b>≗</b> + Create user		

Step 4:	Portainer 存放資訊						
	Information						
	Manage the Docker environment where Portainer is running.						
	Insure that you have started the Portainer container with the following Docker flag:						
	-v "/var/run/docker.sock:/var/run/docker.sock" (Linux).						
	or						
	-v \\.\pipe\docker_engine:\\.\pipe\docker_engine (WINdOWS).						
Step5:	使用 Dashboard 來管理 Docker 的資源						
	Dashboard     O Portainer support     O admin       Endpoint summary <ul> <li> </li> <li> </li></ul>						
	Endpoint info						
	Endpoint local # 2 📟 8.4 GB - Standalone 19.03.4						
	URL /var/run/docker.sock						
	Tags -						
	Stacks Container						
	2 C 80.8 MB						
	Volume						
	3						
	Networks						
Step 6.	使用 Create container 巫師動—個 lup/terNeteback 服務						
500p 0.	反用 Greate container 米成動  回 JupyterNotebook 版務						

Containers > Add	container		O Portainer	support e admi
Name	e.g. myContainer			
Image cor	nfiguration			
Image	e.g. mylmage:myTag	Registry	DockerHub	•
🛕 Image na	ame is required.			
Aiways pu				
Network	ports configuration			
Network Publish al	ports configuration	8		
Network Publish al	ports configuration	9		
Network Publish al Manual ne	ports configuration I exposed network ports to random host ports etwork port publishing O publish a new network po	2		
Network Publish al Manual ne	ports configuration I exposed network ports to random host ports etwork port publishing O publish a new network po untrol	9 🔵		
Network Publish al Manual ne Access co Enable ac	ports configuration I exposed network ports to random host ports etwork port publishing O publish a new network po introl cess control 3	2		
Network Publish al Manual ne Access co Enable ac	ports configuration I exposed network ports to random host ports stwork port publishing O publish a new network po introl cess control 2	9 🔵		
Network Publish al Manual ne Access co Enable ac	Ports configuration I exposed network ports to random host ports etwork port publishing  publish a new network po introl cess control	2 t	문 Restricted	

# Lab 9-建立一個 Node.js 的 webapp

Please refer the following steps to complete the Lab9.

Step 1:	前往 github 下載檔案 \$ git clone <u>https://github.com/HcwXd/docker-tutorial.git</u> \$ cd docker-tutorial <b>nckucc@nckucc-virtual-machine:~</b> \$ git clone https://github.com/HcwXd/docker-tutorial.git Cloning into 'docker-tutorial' remote: Enumerating objects: 29, done. remote: Counting objects: 100% (29/29), done. remote: Counting objects: 100% (22/22), done.
	remote: Total 29 (delta 3), reused 26 (delta 3), pack-reused 0 Unpacking objects: 100% (29/29), done.
Step 2:	檢查 Dockerfile,前往 docker-tutorial/docker-demo-app 目錄 \$ cat Dockerfile
	<pre>nckucc@nckucc-virtual-machine:~/docker-tutorial/docker-demo-app\$ cat Dockerfile FROM node:10.15.3-alpine WORKDIR /app ADD . /app RUN npm install EXPOSE 3000 CMD node index.isnckucc@nckucc-virtual-machine:~/docker-tutorial/docker-demo-app\$ FROM node:10.15.3-alpine</pre>
	這行會載入 Node.js 需要的執行環境,每個不同的程式需要的環境可能都不同,這裏下載的是
	node:10.15.3-alpine,詳細的其他版本可以在 Dockerhub 上看到 WORK DIR /app
	在這個 Docker 的環境之中建立一個工作目錄 /app

	ADD . /app				
	把跟 Dockerfile 同	1個資料夾的程式	北加到剛建立的	工作目錄 /app 中	
	RUN npm install				
	運行 npm install ,	<sub>镩 mm</sub> 诱渦讀]	IV nackage ison -	下載相依的 marchage	e
		或 IIpIII 边边镇I	4X package.json		
	指定 container 到分	个開放的 port			
	CMD node index.j	S			
	我們透過 node ind	lex.js 來執行我個	門的 Server		
Step 3:	使用 docker buil	d 來打包程式(因	國為使用 npm 所	以要加裝 npm 套f	件)
	\$sudo aj	ot-get install n	ipm		
	\$ sudo c	locker build . –	-t docker-dem	o–app	
	nckucc@nckucc-virtu Sending build conte Step 1/6 : FROM nod > 56bc3a1ed035 Step 2/6 : WORKDIR > Using cache > bf3cf5d6a1f6 Step 3/6 : ADD . /a > Using cache > 4e8e548ffad2 Step 4/6 : RUN npm > Using cache > 7fbfe9862a5b Step 5/6 : EXPOSE 3 > Using cache > 7ca8db505f00 Step 6/6 : CMD node > 675551500255 Successfully built Successfully tagged	al-machine:~/docke xt to Docker daemo e:10.15.3-alpine /app install 900 index.js 675551500255 docker-demo-app:l	er-tutorial/docker n 13.31kB	-demo-app\$ sudo docke	er buildt docker-demo-app
Step 4:	查詢映像檔的位置	置後,開始執行	程式		
	\$ sudo do	cker images			
	\$ sudo do	cker run -p 3000	:3000 -it 映像檔:	名稱	
	nckucc@nckucc-virtual	-machine:~/docker-1	tutorial/docker-demo	-app\$ sudo docker imag	ges
	docker-demo-app	TAG latest	IMAGE ID 675551500255	CREATED 5 minutes ago	51ZE 74.2MB
	redhsu/centos	6.7 v2	8dd6473764c6	About an hour ago	191MB
	portainer/portainer	latest	d1219c88aa21	2 hours ago 34 hours ago	80.8MB
	ubuntu	16.04 latest	5f2bf26e3524 775349758637	6 days ago 6 days ago	123MB
	redzhong/ubuntu	18.04	775349758637	6 days ago	64.2MB
	httpd	latest	d3017f59d5e2	7 days ago	165MB
	ubuntu	14.04	2c5e00d77a67	2 months ago 5 months ago	188MB
	node	10.15.3-alpine	56bc3a1ed035	6 months ago	71MB
	centos training/webapp	6./ latest	9f1de3c6ad53 6fae60ef3446	7 months ago 4 years <u>ago</u>	191MB 349MB
	nckucc@nckucc-virtual	-machine:~/docker-1	tutorial/docker-demo	-app\$ sudo run -p 3000	0:3000 -it 675551500255
	<pre>sudo: run: command no nckucc@nckucc-virtual</pre>	machine:~/docker-1	tutorial/docker-demo	-app\$ sudo docker run	-p 3000:3000 -it 67 <u>5551500255</u>
	listening on port 300	0			
Q. 7				· · · · · · · · · · · · · · · · · · ·	
Step5:	用瀏灠器開始 loc	alhost:3000,	就能看到下面的	l小鯨魚圖檔	



## Lab 10 – 在 ubuntu 上安裝 Minikube

Please refe	er the following steps to complete the Lab10.
Step 1:	更新作業系統
	\$ sudo apt-get update \$ sudo apt-get install apt-transport-https \$ sudo apt-get upgrade
Step 2:	安裝 VirtualBox 作為虛擬化平台
	\$ sudo apt-get install virtualbox virtualbox-ext-pack 同意使用者條款 OK
Step 3:	下載 Minikube
	\$ wget https://storage.googleapis.com/minikube/releases/latest/minikube-linux- amd64
	\$ chmod +x minikube-linux-amd64
	\$ sudo mv minikube-linux-amd64 /usr/local/bin/minikube
	\$ minikube version
	<pre>nckucc@nckucc-virtual-machine:-/docker-tutorial/docker-demo-app\$ wget https://storage.googleapis.com/minikube/releases -2019-11-08 01:05:55 https://storage.googleapis.com/minikube/releases/latest/minikube-linux-amd64 Resolving storage.googleapis.com (storage.googleapis.com) 172.217.160.80, 2404:6800:4012::2010 Connecting to storage.googleapis.com (storage.googleapis.com) 172.217.160.80 :443 connected. HTTP request sent, awaiting response 200 OK Length: 48571328 (46M) [application/octet-stream] Saving to: 'minikube-linux-amd64' hinikube-linux-amd64</pre>
	2019-11-08 01:05:56 (80.3 MB/s) - 'minikube-linux-amd64' saved [48571328/48571328]
	<pre>hckucc@nckucc-virtual-machine:~/docker-tutorial/docker-demo-app\$ chmod +x minikube-linux-amd64 hckucc@nckucc-virtual-machine:~/docker-tutorial/docker-demo-app\$ sudo mv minikube-linux-amd64 /usr/local/bin/minikube hckucc@nckucc-virtual-machine:~/docker-tutorial/docker-demo-app\$ minikube version minikube version: v1.5.2 commit: 792dbf92a1de583fcee76f8791cff12e0c9440ad-dirty hckucc@nckucc-virtual-machine:~/docker-tutorial/docker-demo-app\$</pre>
Step 4:	安裝 kubectl 在 ubuntu 18.04
	<pre>\$ curl -s https://packages.cloud.google.com/apt/doc/apt-key.gpg   sudo apt-key add - \$ echo "deb http://apt.kubernetes.io/ kubernetes-xenial main"   sudo tee /etc/apt/sources.list.d/kubernetes.list \$ sudo apt update \$ sudo apt -y install kubectl \$ kubectl version -o json</pre>
	<pre>ckucc@nckucc-virtual-machine:~/docker-tutorial/docker-demo-app\$ kubectl version -o json</pre>
	<pre>"clientVersion": {     "major": "1",     "minor": "16",     "gitVersion": "v1.16.2",     "gitCommit": "c97fe5036ef3df2967d086711e6c0c405941e14b",     "gitTreeState": "clean",     "buildDate": "2019-10-15T19:18:23Z",     "goVersion": "go1.12.10",     "compiler": "gc",     "platform": "linux/amd64" }</pre>
	The connection to the server localhost:8080 was refused - did you specify the right host or port?

Step5:	啟動 Minikube
	\$ sudo minikube start -vm-driver=none
	<pre>nckucc@nckucc-virtual-machine:~/docker-tutorial/docker-demo-app\$ sudo minikube startvm-driver=none minikube v1.5.2 on Ubuntu 18.04 Running on localhost (CPUs=2, Memory=7974MB, Disk=100278MB) 05 release is Ubuntu 18.04.3 LTS Preparing Kubernetes v1.16.2 on Docker '19.03.4' kubelet.resolv-conf=/run/systemd/resolve/resolv.conf Downloading kubeadm v1.16.2 Pulling images Launching Kubernetes Configuring local host environment</pre>
	A The 'none' driver provides limited isolation and may reduce system security and reliability. For more information, see: Thtps://minikube.sigs.k8s.io/docs/reference/drivers/none/
	▲ kubectl and minikube configuration will be stored in /home/nckucc ▲ To use kubectl or minikube commands as your own user, you may need to relocate them. For example, to overwrit
	■ sudo mv /home/nckucc/.kube /home/nckucc/.minikube \$HOME ■ sudo chown -R \$USER \$HOME/.kube \$HOME/.minikube
	<pre>P This can also be done automatically by setting the env var CHANGE_MINIKUBE_NONE_USER=true</pre>
	Z Done! kubectl is now configured to use "minikube"
Step 6:	Minikube 基礎操作
	\$ sudo kubectl cluster-info
	<pre>nckucc@nckucc-virtual-machine:~/docker-tutorial/docker-demo-app\$ sudo kubectl cluster-info</pre>
	Kubernetes master <b>is running at</b> https://140.116.68.82:8443 KubeDNS <b>is running at</b> https://140.116.68.82:8443/api/v1/namespaces/kube-system/services/kube-dns:dns/proxy
	To further debug and diagnose cluster problems, use 'kubectl cluster-info dump'. nckucc@nckucc-virtual-machine:~/docker-tutorial/docker-demo-app\$
	\$ sudo kubectl get nodes
Step 7:	啟用 Kubernetes Dashboard
	\$ minikube addons list
	<pre>nckucc@nckucc-virtual-machine:~/docker-tutorial/docker-demo-app\$ minikube addons list addon-manager: enabled default-storageclass: enabled efk: disabled freshpod: disabled gvisor: disabled heapster: disabled ingress: disabled ingress: disabled logviewer: disabled nwtita-grue-device-plugin: disabled nvidia-gpu-device-plugin: disabled registry: disabled storage-provisioner: enabled storage-provisioner-gluster: disabled nckucc@nckucc-virtual-machine:~/docker-tutorial/docker-demo-app\$</pre>
	\$ sudo minikube dashboard

← → C' û	🖲 🛈 127.0.0.1:38459/api	/v1/namespaces/kubernet	es-dashboard/services/http:kuber	
🛞 kubernetes	Q	Search		
Cluster Roles	Namespaces		Labela	
Cluster	Namespaces			
Namespaces	Name		Labels	F
Nodes			addonmanager.kubernetes.io/mod	
Persistent Volumes	kubernetes-	lashboard	e: Reconcile	4
Storage Classes			kubernetes.io/minikube-addons: dash board	
Namespace	efault			ŀ
default	kube-node-le	ase	-	F
Overview	kube-public			A
Workloads	🔗 kube-system	1		ŀ

# Lab 11 – 在 Kubenetes 上使用 yaml 架設 Wordpress

#### 網站

Please refer the following steps to complete the Lab11.

Step 1:	下載 Lab 所需的 yaml 檔案								
	https://kubernetes.io/examples/application/wordpress/mysql-deployment.yaml								
	https://kubernetes.io/examples/application/wordpress/wordpress-deployment.yaml								
	建立帳號密碼保全的 kustomization.yaml 檔(要把密碼換成自訂的) \$ cat < <eof>./kustomization.yaml secretGenerator: - name: mysql-pass literals: - password=YOUR_PASSWORD EOF</eof>								
	下載組態檔案								

	\$ curl -LO https://k8s.io/examples/application/wordpress/mysql-deployment.yaml								
	\$ curl -LO https://k8s.io/examples/application/wordpress/wordpress-deployment.yaml								
	加入 kustomization.yaml 檔案								
	<pre>\$ cat &lt;<eof>&gt;&gt;./kustomization.yaml</eof></pre>								
	resources:								
	- mysql-deployment.yaml								
	- wordpress-deproyment.yann FOF								
	201								
Step 2:	開始確認及應用								
-	\$ sudo kubectl apply -k ./								
	<pre>nckucc@nckucc-virtual-machine:~\$ sudo kubectl apply -k ./</pre>								
	[sudo] password for nckucc:								
	secret/mysql-pass-21466cdddc created service/wordpress-mysql created								
	service/wordpress created								
	deployment.apps/wordpress-mysql created deployment.apps/wordpress created								
	persistentvolumeclaim/mysql-pv-claim created								
	persistentvolumeclaim/wp-pv-claim created								
	\$ sudo kubectl get secrete								
	nckucc@nckucc-virtual-machine:~\$ sudo kubectl get secrets NAME TYPE DATA AGE								
	default-token-7tlh2 kubernetes.io/service-account-token 3 42m								
	mysql-pass-2f466cdddc Opaque 1 103s								
	\$ kubectl get pvc								
	NAME STATUS VOLUME CAPACITY ACCESS MODES								
	mysql-pv-claim Bound pvc-4487c5ed-2810-43b6-a824-c3030bc395e7 20Gi RWO wp-pv-claim Bound pvc-ae7aef31-54f5-40d4-a3e1-609ff47be0ce 20Gi RWO								
	nckucc@nckucc-virtual-machine:~S								
	\$ sudo kubectl get pods								
	nckucc@nckucc-virtual-machine:~\$ sudo kubectl get pods								
	NAME READY STATUS RESTARTS AGE wordpress-d89c954fd-7b9hz 1/1 Running 1 2m53s								
	wordpress-mysql-bbc9bfddb-9f4kt 1/1 Running 0 2m53s								
	\$ sudo kubectl get services wordpress								
	nckucc@nckucc-virtual-machine:~\$ sudo kubectl get service wordpress NAME TYPE CLUSTER-IP EXTERNAL-IP PORT(S) AGE								
	wordpress LoadBalancer 10.99.198.152 <pending> 80:32373/TCP 3m36s</pending>								
	\$ sudo minikube service wordpressurl								
	<pre>nckucc@nckucc-virtual-machine:~\$ sudo minikube service wordpressurl http://140_116_68_82:22272</pre>								
	nctp.//140.110.08.82.32373								
Step 3:	登入網站開始使用								



$(\leftarrow) \rightarrow$ C' $\textcircled{a}$	0 3 127.0.	0.1:38459/api/v1/names	paces/kuberne	etes-dashboard/services	/http://	⊠ ☆	<u>≁</u> ∥	\ 🗉 🔹
🛞 kubernetes		Q Search						+
$\equiv$ Workloads								
Cluster Cluster Roles	Wo	rkload Status						
Namespaces Nodes Persistent Volumes Storage Classes		100.0%		100.0	%		100.0%	
Namespace default	•	Deployments		Pod	5		Replica Set	S
Overview	De	ployments						$\overline{\Xi}$
Workloads		Name	Namespace	e Labels	Pods	Age 个	Images	
Cron Jobs	0	wordpress	default	app: wordpress	1/1	22 minutes	wordpress:4.8	-apache
Daemon Sets	0	wordpress-mysql	default	app: wordpress	1/1	22 minutes	mysql:5.6	
Deployments						1 - 2 of 2	1< <	$\rightarrow$ $\rightarrow$