

NASA hw5

Author: B12705014陳泊華

Question 1

建立虛擬磁碟檔案

```
$ qemu-img create -f qcow2 /tmp2/b12705014/nasahw5/ubuntu.qcow2 20G
```

利用virt-install安裝VM(man virt-install發現unprivileged的network設定要用user)

```
$ virt-install --virt-type kvm \
--name b12705014 \
--vcpus=2 \
--ram 8192 \
--disk ubuntu.qcow2,format=qcow2 \
--network user,model=virtio,mac=52:54:F8:70:50:14 \
--graphics vnc,password=nasa2024 \
--noautoconsole \
--cdrom=/tmp2/nasa-hw5/ubuntu.iso
```

```
$ virsh list
```

```
b12705014@ws1 [~] virsh list
  Id   名稱      狀態
-----
 10   b12705014    執行中
```

```
b12705014@ws1 [~] █
```

```
$ ip a
```

```
b12705014@nasa-hw5:~$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
        inet 127.0.0.1/8 scope host lo
            valid_lft forever preferred_lft forever
        inet6 ::1/128 scope host
            valid_lft forever preferred_lft forever
2: enp1s0: <BROADCAST,MULTICAST> mtu 1500 qdisc noop state DOWN group default qlen 1000
    link/ether 52:54:f8:70:50:14 brd ff:ff:ff:ff:ff:ff
3: br-6f6c9b42f3ce: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc noqueue state DOWN group default
    link/ether 02:42:d0:b7:0e:fc brd ff:ff:ff:ff:ff:ff
        inet 192.168.49.1/24 brd 192.168.49.255 scope global br-6f6c9b42f3ce
            valid_lft forever preferred_lft forever
4: docker0: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc noqueue state DOWN group default
    link/ether 02:42:64:e2:f2:44 brd ff:ff:ff:ff:ff:ff
        inet 172.17.0.1/16 brd 172.17.255.255 scope global docker0
            valid_lft forever preferred_lft forever
```

開機畫面

```
[ OK ] Finished Set console scheme.
[ OK ] Finished Terminate Plymouth Boot Screen.
[ OK ] Created slice Slice /system/getty.
[ OK ] Started Getty on tty1.
[ OK ] Reached target Login Prompts.
[ OK ] Started LSB: automatic crash report generation.
[ OK ] Finished Record successful boot for GRUB.
    Starting GRUB failed boot detection...
[ OK ] Started Authorization Manager.
    Starting Modem Manager...
[ OK ] Finished GRUB failed boot detection.
[ OK ] Finished Remove Stale Onl...ext4 Metadata Check Snapshots.
[ OK ] Started OpenBSD Secure Shell server.
[ OK ] Started Modem Manager.
[ OK ] Started User Login Management.
[ OK ] Started Unattended Upgrades Shutdown.
[ OK ] Finished Ubuntu FAN network setup.
[ OK ] Started Disk Manager.
[ OK ] Started Dispatcher daemon for systemd-networkd.
[ OK ] Started containerd container runtime.
    Starting Docker Application Container Engine...
[ OK ] Started libcontainer conta...45dfc0f9cf254c08df8e46a06c0ff.
[ OK ] Started libcontainer conta...2ee67dc10a3fc37a5c5fcdf1f8005.
[ OK ] Started Docker Application Container Engine.

Ubuntu 22.04.4 LTS nasa-hw5 ttyS0

nasa-hw5 login: [ 21.222390] cloud-init[1629]: Cloud-init v. 23.3.3-0ubuntu0~22.04.1 running 'modules:config' at Fri, 29 Mar 2024 12:16:25 +0000. Up 21.15 seconds.
[ 21.726533] cloud-init[1643]: Cloud-init v. 23.3.3-0ubuntu0~22.04.1 running 'modules:final' at Fri, 29 Mar 2024 12:16:25 +0000. Up 21.66 seconds.
[ 21.797424] cloud-init[1643]: Cloud-init v. 23.3.3-0ubuntu0~22.04.1 finished at Fri, 29 Mar 2024 12:16:25 +0000. Datasource DataSourceNone.
Up 21.79 seconds
[ 21.804482] cloud-init[1643]: 2024-03-29 12:16:25,965 - cc_final_message.py[WARNIN]: Used fallback datasource
```

reference:

(i) virt-install command <https://blog.gtwang.org/linux/kvm-qemu-virt-install-command-tutorial/> (<https://blog.gtwang.org/linux/kvm-qemu-virt-install-command-tutorial/>)

(ii) networking modes

<https://docs.oracle.com/en/virtualization/virtualbox/6.0/user/networkingmodes.html> (<https://docs.oracle.com/en/virtualization/virtualbox/6.0/user/networkingmodes.html>)

(iii) <https://serverfault.com/questions/534484/libvirt-network-error-no-default-network-device-found> (<https://serverfault.com/questions/534484/libvirt-network-error-no-default-network-device-found>).

Question 2

serial console

```
$ virsh vncdisplay b12705014
```

```
$ nano /etc/default/grub
```

修改grub檔中 GRUB_CMDLINE_LINUX=""為

GRUB_CMDLINE_LINUX="console=ttyS0,115200n8"

```

root@nasa-hw5:/home/b12705014# cat /etc/default/grub
# If you change this file, run 'update-grub' afterwards to update
# /boot/grub/grub.cfg.
# For full documentation of the options in this file, see:
#   info -f grub -n 'Simple configuration'

GRUB_DEFAULT=0
GRUB_TIMEOUT_STYLE=hidden
GRUB_TIMEOUT=0
GRUB_DISTRIBUTOR=`lsb_release -i -s 2> /dev/null || echo Debian`
GRUB_CMDLINE_LINUX_DEFAULT=""
GRUB_CMDLINE_LINUX="console=ttyS0,115200n8"

# Uncomment to enable BadRAM filtering, modify to suit your needs
# This works with Linux (no patch required) and with any kernel that obtains
# the memory map information from GRUB (GNU Mach, kernel of FreeBSD ...)
#GRUB_BADRAM="0x01234567,0xfefefefe,0x89abcdef,0xefefefef"

# Uncomment to disable graphical terminal (grub-pc only)
#GRUB_TERMINAL=console

# The resolution used on graphical terminal
# note that you can use only modes which your graphic card supports via VBE
# you can see them in real GRUB with the command `vbeinfo'
#GRUB_GFXMODE=640x480

# Uncomment if you don't want GRUB to pass "root=UUID=xxx" parameter to Linux
#GRUB_DISABLE_LINUX_UUID=true

# Uncomment to disable generation of recovery mode menu entries
#GRUB_DISABLE_RECOVERY="true"

# Uncomment to get a beep at grub start
#GRUB_INIT_TUNE="480 440 1"
root@nasa-hw5:/home/b12705014# █

```

\$ update-grub

```

$ virsh reboot b12705014
$ virsh console b12705014 成功進入VM

```

```

Last login: Thu Mar 28 16:50:25 UTC 2024 on ttyS0
b12705014@nasa-hw5:~$
b12705014@ws1 [~] virsh console b12705014
Connected to domain 'b12705014'
Escape character is ^] (Ctrl + ])

```

b12705014@nasa-hw5:~\$ █

ssh

```
$ virsh qemu-monitor-command --hmp b12705014 'hostfwd_add ::8099-:22'
```

在work station上設定ssh的port forwarding

```
$ ssh localhost -p 8099
b12705014@ws1 [~] ssh localhost -p 8099
b12705014@localhost's password:
Welcome to Ubuntu 22.04.4 LTS (GNU/Linux 5.15.0-101-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/pro

 System information as of Fri Mar 29 09:34:41 AM UTC 2024

System load:                      0.25
Usage of /:                         95.0% of 9.75GB
Memory usage:                      18%
Swap usage:                        0%
Processes:                          203
Users logged in:                  1
IPv4 address for br-28116f69c338: 192.168.49.1
IPv4 address for br-d995b17418c4: 172.18.0.1
IPv4 address for docker0:          172.17.0.1
IPv4 address for enp1s0:            10.0.2.15
IPv6 address for enp1s0:           fec0::5054:f8ff:fe70:5014

=> / is using 95.0% of 9.75GB

* Strictly confined Kubernetes makes edge and IoT secure. Learn how MicroK8s just raised the bar for easy, resilient and secure K8s cluster deployment.

https://ubuntu.com/engage/secure-kubernetes-at-the-edge

Expanded Security Maintenance for Applications is not enabled.

12 updates can be applied immediately.
To see these additional updates run: apt list --upgradable

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

Last login: Fri Mar 29 09:31:31 2024 from 10.0.2.2
b12705014@nasa-hw5:~$ █
```

references:

- (i) <https://help.ubuntu.com/community/SerialConsoleHowto>
(<https://help.ubuntu.com/community/SerialConsoleHowto>).
- (ii) <https://serverfault.com/questions/170079/forwarding-ports-to-guests-in-libvirt-kvm>
(<https://serverfault.com/questions/170079/forwarding-ports-to-guests-in-libvirt-kvm>).

Question 3

```
$ docker version
root@nasa-hw5:/home/b12705014# docker version
Client:
  Version:          24.0.5
  API version:      1.43
  Go version:       go1.20.3
  Git commit:       24.0.5-0ubuntu1~22.04.1
  Built:            Mon Aug 21 19:50:14 2023
  OS/Arch:          linux/amd64
  Context:          default

Server:
  Engine:
    Version:          24.0.5
    API version:      1.43 (minimum version 1.12)
    Go version:       go1.20.3
    Git commit:       24.0.5-0ubuntu1~22.04.1
    Built:            Mon Aug 21 19:50:14 2023
    OS/Arch:          linux/amd64
    Experimental:    false
  containerd:
    Version:          1.7.2
    GitCommit:
  runc:
    Version:          1.1.7-0ubuntu1~22.04.2
    GitCommit:
  docker-init:
    Version:          0.19.0
    GitCommit:
```

```
$ docker-compose version
root@nasa-hw5:/home/b12705014# docker-compose version
docker-compose version 1.29.2, build unknown
docker-py version: 5.0.3
CPython version: 3.10.12
OpenSSL version: OpenSSL 3.0.2 15 Mar 2022
```

Question 4

使用Container的時機

1. 應用程序的容易部署和移植性
2. 資源隔離和管理
3. 微服務架構的支持
4. 快速開發和部署

使用VM的時機

1. 需要完全隔離的環境
2. 需要多個作業系統的支持
3. 需要更高的性能隔離、空間和資源

reference: <https://aws.amazon.com/tw/compare/the-difference-between-containers-and-virtual-machines/> (<https://aws.amazon.com/tw/compare/the-difference-between-containers-and-virtual-machines/>).

Question 5

在 Linux 上，Docker 可以直接使用宿主機的內核並在其中運行容器，因此效能通常會比較好。然而，在 macOS 和 Windows 上，由於缺少類似於 Linux 內核的內建容器支援，Docker 需要使用虛擬機來運行容器，導致性能下降的問題。

reference: <https://www.docker.com/blog/the-magic-behind-the-scenes-of-docker-desktop/> (<https://www.docker.com/blog/the-magic-behind-the-scenes-of-docker-desktop/>).

Question 6

- (a) \$ docker stop CONTAINER
- (b) \$ docker rmi IMAGE rmi代表remove image
- (c) \$ docker system prune -a -a代表包含所有系統資源
- (d) \$ docker inspect NAME|ID
- (e) \$ docker stats --no-stream

reference: <https://hackmd.io/@joshhu/SJtUggLKS> (<https://hackmd.io/@joshhu/SJtUggLKS>).

Question 7

```
$ docker run -d -p 8763:80 --name nginx-1 nginx:1.24.0
```

```
$ curl localhost:8763
```

```
root@nasa-hw5:/home/b12705014# docker run -d -p 8763:80 --name nginx-1 nginx:1.24.0
Unable to find image 'nginx:1.24.0' locally
1.24.0: Pulling from library/nginx
c0edef2937fa: Pull complete
4fdaa9ff0d0b0: Pull complete
697f6b054c93: Pull complete
9e4cc9a5ba09: Pull complete
55f9a23afdb0: Pull complete
5981c066c964: Pull complete
Digest: sha256:25b1dd75ab9caf2f84bc35cc82c0924c93a2b5b2495e280bb0f3bad826d5fb37
Status: Downloaded newer image for nginx:1.24.0
a98593d4328cf8556ac86bd9e81d8cb40658d5a5a33b4860769432e5d50c81bd
root@nasa-hw5:/home/b12705014# docker ps
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
a98593d4328c nginx:1.24.0 "/docker-entrypoint..." 2 minutes ago Up 2 minutes 0.0.0.0:8763->80/tcp, :::8763->80/tcp nginx-1
root@nasa-hw5:/home/b12705014# curl localhost:8763
<!DOCTYPE html>
<html>
<head>
<title>Welcome to nginx!</title>
<style>
html { color-scheme: light dark; }
body { width: 35em; margin: 0 auto;
font-family: Tahoma, Verdana, Arial, sans-serif; }
</style>
</head>
<body>
<h1>Welcome to nginx!</h1>
<p>If you see this page, the nginx web server is successfully installed and working. Further configuration is required.</p>
<p>For online documentation and support please refer to
<a href="http://nginx.org/">nginx.org</a>.<br/>
Commercial support is available at
<a href="http://nginx.com/">nginx.com</a>.</p>
<p><em>Thank you for using nginx.</em></p>
</body>
</html>
```

Question 8

```
$ docker exec -it nginx-1 /bin/sh
```

```
root@nasa-hw5:/home/b12705014# docker exec -it nginx-1 /bin/sh
#
# ls
bin boot dev docker-entrypoint.d docker-entrypoint.sh etc home lib lib64 media mnt opt proc root run sbin srv sys tmp usr var
# pwd
/
```

Question 9

```
$ docker cp nginx-1:/etc/nginx/nginx.conf nginx.conf
root@nasa-hw5:/home/b12705014# docker cp nginx-1:/etc/nginx/nginx.conf nginx.conf
Successfully copied 2.56kB to /home/b12705014/nginx.conf
root@nasa-hw5:/home/b12705014# cat nginx.conf

user    nginx;
worker_processes  auto;

error_log  /var/log/nginx/error.log notice;
pid      /var/run/nginx.pid;


events {
    worker_connections  1024;
}

http {
    include      /etc/nginx/mime.types;
    default_type application/octet-stream;

    log_format  main  '$remote_addr - $remote_user [$time_local] "$request" '
                      '$status $body_bytes_sent "'.$http_referer" '
                      '"$http_user_agent" "'.$http_x_forwarded_for'"';

    access_log  /var/log/nginx/access.log  main;

    sendfile      on;
    #tcp_nopush    on;

    keepalive_timeout  65;
    #gzip  on;

    include /etc/nginx/conf.d/*.conf;
}
root@nasa-hw5:/home/b12705014# _
```

Question 10

差異

1. 用於定義默認命令

ENTRYPOINT 用於設置容器的默認命令，該命令將始終作為容器運行時的主要命令
CMD 用於定義容器運行時的默認命令和參數，它提供了一種更靈活的方式來定義命令

2. 處理命令行參數的方式（覆蓋性）

對於 ENTRYPOINT，如果容器在啟動時指定了命令，則該命令將被附加到 ENTRYPOINT 指定的命令之後作為參數傳遞

對於 CMD，如果 Dockerfile 中有多個 CMD 指令，僅最後一個 CMD 指令生效，如果在 docker run 命令中提供了命令，它將覆蓋 CMD 指定的命令

3. 參數傳遞方式

在 ENTRYPOINT 中使用 JSON 數組或 Shell 格式指定命令時，命令和參數都必須被引用為 JSON 數組或字符串。這使得傳遞複雜的命令和參數變得更加困難。

在 CMD 中指定的命令和參數可以更自由地使用 Shell 語法，例如通配符和管道等，使得命令的撰寫和使用更加靈活。

Dockerfile 應用

```
FROM python:3.8-slim

# 定義 ENTRYPOINT 為 Python 解釋器
ENTRYPOINT ["python"]

# 定義 CMD 為默認運行的腳本和參數
CMD ["app.py", "arg1", "arg2"]
```

如此運行容器時如果不提供任何額外的命令，它將默認運行 `python app.py`。
`arg1 arg2`；如果希望運行不同的腳本或者提供不同的參數，可以直接在 `docker run` 命令中覆蓋 CMD 指定的默認參數

reference: https://www.google.com/search?q=entrypoint+vs+cmd&rlz=1C5CHFA_enTW978TW978&oq=entrypoint+vs+cmd&gs_lcp=EgZjaHJvbWUqDggAEEUYJxg7GIAEGloFMg4IABBFGCcYOxiABBiKBTIHCAEQABiABDIGCAIQABgeMgYIAxAAGB4yBggEEAYHjIGCAUQABgeMgYIBhAAGB4yBggHEAYHjIGCAgQABgeMgYICRAAGB6oAgCwAgA&sourceid=chrome&ie=UTF-8
https://www.google.com/search?q=entrypoint+vs+cmd&rlz=1C5CHFA_enTW978TW978&oq=entrypoint+vs+cmd&gs_lcp=EgZjaHJvbWUqDggAEEUYJxg7GIAEGloFMg4IABBFGCcYOxiABBiKBTIHCAEQABiABDIGCAIQABgeMgYIAxAAGB4yBggEEAYHjIGCAUQABgeMgYIBhAAGB4yBggHEAYHjIGCAgQABgeMgYICRAAGB6oAgCwAgA&sourceid=chrome&ie=UTF-8

Question 11

Docker 是一個用於創建和管理單個容器的工具，而 Docker Compose 則是一個用於定義和管理多容器應用程序的配置的工具，並且使用 YAML 檔來讀入 configuration。在開發和部署多容器應用程序時，通常會使用 Docker Compose 來簡化配置和管理過程。

reference: <https://www.theserverside.com/blog/Coffee-Talk-Java-News-Stories-and-Opinions/Docker-run-vs-docker-compose-Whats-the-difference>
<https://www.theserverside.com/blog/Coffee-Talk-Java-News-Stories-and-Opinions/Docker-run-vs-docker-compose-Whats-the-difference>

Question 12

- (a) ````\$ docker-compose up -d```
- (b) ````\$ docker-compose pause```
- \(c) ````\$ docker-compose down --volumes```

Question 13

```
$ cat Dockerfile
```

```
root@nasa-hw5:/home/b12705014# cat Dockerfile
FROM alpine:latest

RUN apk add --no-cache sl

CMD ["sl"]
root@nasa-hw5:/home/b12705014# _
```

```
$ docker build -t sl .
```

```
$ docker run -rm -it sl
```

即可看到sl小火車畫面

Question 14

```
$ cat docker-compose.yaml
b12705014@nasa-hw5:~$ cat docker-compose.yaml
version: '3.8'

services:
  mysql:
    image: mysql:latest
    #container_name: mysql
    networks:
      - nasa-net
    environment:
      MYSQL_HOST: b12705014
      MYSQL_ROOT_PASSWORD: secret
    #command: --default-authentication-plugin=mysql_native_password
    restart: always

  b12705014:
    image: phpmyadmin/phpmyadmin:latest
    #container_name: phpmyadmin
    networks:
      - nasa-net
    ports:
      - "8888:80"
    environment:
      PMA_HOST: mysql
      PMA_USER: root
      PMA_PASSWORD: secret
    restart: always

networks:
  nasa-net:
```

在工作站上設定至VM的port forwarding (使用還沒有人用過的9019)

```
$ virsh qemu-monitor-command --hmp b12705014 'hostfwd_add ::9019-:8888'
```

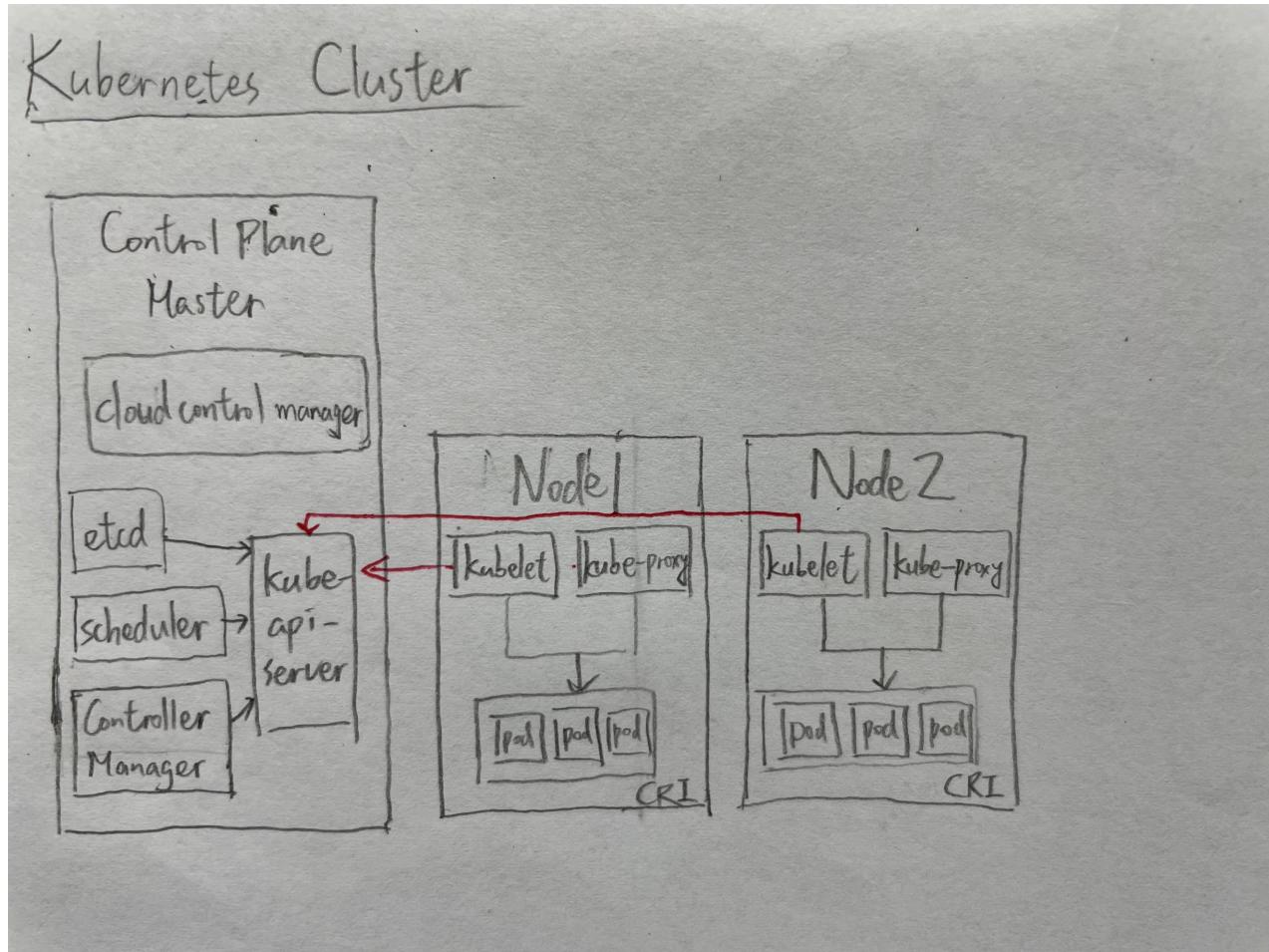
```
$ docker-compose up
```

到瀏覽器輸入 ws1.csie.ntu.tw:9019 (<http://ws1.csie.ntu.tw:9019>)

The screenshot shows the phpMyAdmin interface for a MySQL server. The left sidebar lists databases: New, information_schema, mysql, performance_schema, and sys. The main area has tabs for Databases, SQL, Status, User accounts, Export, Import, Settings, Binary log, Replication, Variables, and More.

- General settings:** Server connection collation: utf8mb4_unicode_ci
- Appearance settings:** Language: English, Theme: pmahomme
- Database server:**
 - Server: mysql via TCP/IP
 - Server type: MySQL
 - Server connection: SSL is not being used
 - Server version: 8.3.0 - MySQL Community Server - GPL
 - Protocol version: 10
 - User: root@172.18.0.3
 - Server charset: UTF-8 Unicode (utf8mb4)
- Web server:**
 - Apache/2.4.57 (Debian)
 - Database client version: libmysql - mysqld 8.2.8
 - PHP extension: mysqli curl mbstring sodium
 - PHP version: 8.2.8
- phpMyAdmin:**
 - Version information: 5.2.1 (up to date)
 - Documentation
 - Official Homepage
 - Contribute
 - Get support
 - List of changes
 - License

Question 15



a. Master node

- Kube-apiserver (Kubernetes API Server)：負責提供 Kubernetes API 服務，接收來自用戶和其他系統組件的請求，以及與 etcd 進行通信
- etcd: 分佈式鍵值存儲，存儲叢集的所有配置數據，包括集群狀態、配置信息等
- Kube-scheduler (Kubernetes Scheduler)：負責根據容器的需求和節點的資源狀況，將容器部署到適合的節點上
- Kube-controller-manager (Kubernetes Controller Manager)：包含多個控制器，用於管理叢集中的各種資源，如部署、ReplicaSet、Service 等
- Cloud-controller-manager (雲端控制器管理器)：用於與雲提供商集成，管理雲資源

b. Worker nodes

- Kubelet: 是主節點通過 Kubernetes API 發送指令給工作節點的代理，負責管理容器的生命週期、健康狀態等
- Kube-proxy (Kubernetes Proxy)：負責維護節點上的網絡規則，實現網絡的轉發和負載均衡
- Container Runtime: 負責運行容器，如 Docker、containerd 等

c. Advantages

- 自動擴展：Kubernetes 可以根據應用程序的負載和需求自動擴展容器，以應對流量峰值，並在負載減少時自動縮減容器數量，從而節省成本
- 容器隔離和安全性：Kubernetes 提供強大的容器隔離機制，使不同應用程序和服務之間能夠相互隔離，從而提高安全性
- 高可用性：Kubernetes 可以在叢集中自動管理容器的部署和健康狀態，並且具有自動恢復能力，從而確保應用程序始終可用

reference: <https://kubernetes.io/docs/concepts/architecture/>
[\(https://kubernetes.io/docs/concepts/architecture/\)](https://kubernetes.io/docs/concepts/architecture/).

Question 16

*本題使用第一題所架設的VM繼續進行

```
$ snap install kubectl --classic
$ sudo apt-get update
$ sudo apt-get install apt-transport-https ca-certificates curl software-properties-common
$ sudo usermod -aG docker b12705014 && newgrp docker
```

```
$ minikube start
$ kubectl get nodes
b12705014@nasa-hw5:~$ minikube status
minikube
type: Control Plane
host: Running
kubelet: Running
apiserver: Running
kubeconfig: Configured

b12705014@nasa-hw5:~$ kubectl get nodes
NAME      STATUS    ROLES          AGE      VERSION
minikube   Ready     control-plane  65s     v1.28.3
b12705014@nasa-hw5:~$ █
```

touch q16 YAML檔進行相關設定（見附檔q16.yaml）

部署kubectl service

```
$ kubectl apply -f q16
```

檢查

```
$ kubectl get services
b12705014@nasa-hw5:~$ kubectl get services
NAME        TYPE        CLUSTER-IP      EXTERNAL-IP      PORT(S)        AGE
kubernetes  ClusterIP  10.96.0.1      <none>          443/TCP       4h2m
mysql-b12705014  ClusterIP  10.99.227.73  <none>          3306/TCP      3h58m
nginx-b12705014  NodePort    10.97.9.133    <none>          8888:30001/TCP 3h58m
phpmyadmin-b12705014  NodePort    10.99.66.46    <none>          8887:30471/TCP 3h58m
```

成功運行三種服務的deployment及service

```
$ kubectl describe Deployment nginx-b12705014
```

```
b12705014@nasa-hw5:~$ kubectl describe Deployment nginx-b12705014
Name:          nginx-b12705014
Namespace:     default
CreationTimestamp: Fri, 29 Mar 2024 09:08:11 +0000
Labels:        <none>
Annotations:   deployment.kubernetes.io/revision: 1
Selector:      app=nginx-b12705014
Replicas:      1 desired | 1 updated | 1 total | 1 available | 0 unavailable
StrategyType:  RollingUpdate
MinReadySeconds: 0
RollingUpdateStrategy: 25% max unavailable, 25% max surge
Pod Template:
  Labels:  app=nginx-b12705014
  Containers:
    nginx:
      Image:      nginx:latest
      Port:       80/TCP
      Host Port:  0/TCP
      Environment: <none>
      Mounts:    <none>
      Volumes:   <none>
  Conditions:
    Type     Status  Reason
    ----  -----
    Progressing  True   NewReplicaSetAvailable
    Available   True   MinimumReplicasAvailable
  OldReplicaSets: <none>
  NewReplicaSet:  nginx-b12705014-65d687567d (1/1 replicas created)
  Events:
    Type     Reason           Age   From             Message
    ----  -----           ----  ----             -----
    Normal  ScalingReplicaSet  88m   deployment-controller  Scaled up replica set nginx-b12705014-65d687567d to 1
b12705014@nasa-hw5:~$
```

```
$ kubectl describe Deployment mysql-b12705014
```

```
b12705014@nasa-hw5:~$ kubectl describe Deployment mysql-b12705014
Name:          mysql-b12705014
Namespace:     default
CreationTimestamp: Fri, 29 Mar 2024 09:08:11 +0000
Labels:        <none>
Annotations:   deployment.kubernetes.io/revision: 1
Selector:      app=mysql-b12705014
Replicas:      1 desired | 1 updated | 1 total | 1 available | 0 unavailable
StrategyType:  RollingUpdate
MinReadySeconds: 0
RollingUpdateStrategy: 25% max unavailable, 25% max surge
Pod Template:
  Labels:  app=mysql-b12705014
  Containers:
    mysql:
      Image:      mysql:latest
      Port:       <none>
      Host Port: <none>
      Environment:
        MYSQL_ROOT_PASSWORD: <set to the key 'password' in secret 'mysql-b12705014-password'>  Optional: false
      Mounts:    <none>
      Volumes:   <none>
  Conditions:
    Type     Status  Reason
    ----  -----
    Progressing  True   NewReplicaSetAvailable
    Available   True   MinimumReplicasAvailable
  OldReplicaSets: <none>
  NewReplicaSet:  mysql-b12705014-566f6b7d7 (1/1 replicas created)
  Events:
    Type     Reason           Age   From             Message
    ----  -----           ----  ----             -----
    Normal  ScalingReplicaSet  91m   deployment-controller  Scaled up replica set mysql-b12705014-566f6b7d7 to 1
b12705014@nasa-hw5:~$
```

```
$ kubectl describe Deployment phpmyadmin-b12705014
```

```
b12705014@nasa-hw5:~$ kubectl describe Deployment phpmyadmin-b12705014
Name:          phpmyadmin-b12705014
Namespace:     default
CreationTimestamp: Fri, 29 Mar 2024 09:08:12 +0000
Labels:        <none>
Annotations:   deployment.kubernetes.io/revision: 1
Selector:      app=phpmyadmin-b12705014
Replicas:      1 desired | 1 updated | 1 total | 1 available | 0 unavailable
StrategyType:  RollingUpdate
MinReadySeconds: 0
RollingUpdateStrategy: 25% max unavailable, 25% max surge
Pod Template:
  Labels:  app=phpmyadmin-b12705014
  Containers:
    phpmyadmin:
      Image:      phpmyadmin/phpmyadmin:latest
      Port:       80/TCP
      Host Port:  0/TCP
      Environment:
        PMA_HOST:           mysql-b12705014
        MYSQL_ROOT_PASSWORD: <set to the key 'password' in secret 'mysql-b12705014-password'> Optional: false
      Mounts:        <none>
      Volumes:       <none>
  Conditions:
    Type        Status  Reason
    ----        ----   -----
    Available   True    MinimumReplicasAvailable
    Progressing True    NewReplicaSetAvailable
  OldReplicaSets: <none>
  NewReplicaSet:  phpmyadmin-b12705014-7fb4b77558 (1/1 replicas created)
  Events:
    Type  Reason     Age   From            Message
    ----  -----     ---   ----            -----
    Normal  ScalingReplicaSet  92m   deployment-controller  Scaled up replica set phpmyadmin-b12705014-7fb4b77558 to 1
b12705014@nasa-hw5:~$
```

```
$ kubectl describe Service nginx-b12705014
```

```
b12705014@nasa-hw5:~$ kubectl describe Service nginx-b12705014
```

```
Name:          nginx-b12705014
Namespace:     default
Labels:        <none>
Annotations:   <none>
Selector:      app=nginx-b12705014
Type:          ClusterIP
IP Family Policy: SingleStack
IP Families:   IPv4
IP:             10.97.9.133
IPs:            10.97.9.133
Port:          <unset>  8888/TCP
TargetPort:    80/TCP
Endpoints:     10.244.0.9:80
Session Affinity: None
Events:        <none>
b12705014@nasa-hw5:~$
```

```
$ kubectl describe Service mysql-b12705014
```

```
b12705014@nasa-hw5:~$ kubectl describe Service mysql-b12705014
Name: mysql-b12705014
Namespace: default
Labels: <none>
Annotations: <none>
Selector: app=mysql-b12705014
Type: ClusterIP
IP Family Policy: SingleStack
IP Families: IPv4
IP: 10.99.227.73
IPs: 10.99.227.73
Port: <unset> 3306/TCP
TargetPort: 3306/TCP
Endpoints: 10.244.0.8:3306
Session Affinity: None
Events: <none>
b12705014@nasa-hw5:~$ █
```

```
$ kubectl describe Service phpmyadmin-b12705014
```

```
b12705014@nasa-hw5:~$ kubectl describe Service phpmyadmin-b12705014
Name: phpmyadmin-b12705014
Namespace: default
Labels: <none>
Annotations: <none>
Selector: app=phpmyadmin-b12705014
Type: ClusterIP
IP Family Policy: SingleStack
IP Families: IPv4
IP: 10.99.66.46
IPs: 10.99.66.46
Port: <unset> 8888/TCP
TargetPort: 80/TCP
Endpoints: 10.244.0.6:80
Session Affinity: None
Events: <none>
b12705014@nasa-hw5:~$ █
```

```
$ kubectl get services --all-namespaces
```

```
b12705014@nasa-hw5:~$ kubectl get services --all-namespaces
NAMESPACE NAME TYPE CLUSTER-IP EXTERNAL-IP PORT(S) AGE
default kubernetes ClusterIP 10.96.0.1 <none> 443/TCP 127m
default mysql-b12705014 ClusterIP 10.99.227.73 <none> 3306/TCP 124m
default nginx-b12705014 NodePort 10.97.9.133 <none> 8888:30001/TCP 124m
default phpmyadmin-b12705014 ClusterIP 10.99.66.46 <none> 8888/TCP 124m
kube-system kube-dns ClusterIP 10.96.0.10 <none> 53/UDP,53/TCP,9153/TCP 127m
```

Browser輸入 <http://ws1.csie.ntu.edu.tw:9020/> (<http://ws1.csie.ntu.edu.tw:9020/>).

The screenshot shows the phpMyAdmin interface for a MySQL server. The left sidebar lists databases: New, information_schema, mysql, performance_schema, and sys. The main content area has four panels:

- General settings:** Shows the server connection collation set to utf8mb4_unicode_ci.
- Database server:** Displays server details: Server: mysql via TCP/IP, Server type: MySQL, Server connection: SSL is not being used, Server version: 8.3.0 - MySQL Community Server - GPL, Protocol version: 10, User: root@172.18.0.3, and Server charset: UTF-8 Unicode (utf8mb4).
- Web server:** Shows Apache/2.4.57 (Debian), Database client version: libmysql - mysqlnd 8.2.8, PHP extension: mysqli, curl, mbstring, sodium, and PHP version: 8.2.8.
- phpMyAdmin:** A footer menu with links to Version information: 5.2.1 (up to date), Documentation, Official Homepage, Contribute, Get support, List of changes, and License.