Network Administration and System Administration
Midterm Examination

Time: 2020/4/20 (Mon.) 09:10 - 12:10

Instructions and Announcements

- 考試時間共三小時，三人一組遠端考試。
- 所有題目均可在自己的筆電上完成。
- 題目所需 VM 或其他檔案已於昨晚公告載點，解壓縮密碼: `CA6o2XuCQg2J`
- 為避免發生重大意外，請不要一次開太多 VM，且請斟酌定期備份 VM 至 USB，避免當機時損失過多進度。
- 助教將會在 CSIE Workstation 上面測試 shell script 題目，請確保程式能正確地在上面執行。
- 完成題目時請上傳至 https://tinyurl.com/nasa2020-midterm-form，每組每題最多上傳 **5** 次。
- 有些題目會自動批改、有些題目上傳後助教會使用 Google Meet 連入、以分享螢幕方式檢查、有些題目助教會線上批改並給予評語。
  (若該題有規定特定上傳格式，請務必遵照該格式)
- 上傳結果與各組目前分數見：https://tinyurl.com/nasa2020-midterm-board
- 組與組間禁止討論。考試期間禁止使用手機、電話、任何通訊軟體等與同組成員外任何人聯繫。如被發現將視為作弊行為，期中考以 **0** 分計。
- 如需發問，可使用 slido。請在問題開頭標記發問題號。如果想用語音和助教發問，請在問題中附上開好的 Google Meet 連結，助教會點進連結和同學交流。
- 各題後面黑色星號數目代表我們估計的難度。請參考，可用來決定解題順序。
- 滿分 230 pts。

Submission Format

- 根據題目需求，有些題目需使用 google meet 請同學當場 demo ，另外有些僅需要同學上傳螢幕截圖或程式碼。
- 題目都會說明需上傳螢幕截圖、程式碼或 google meet 連結，同學依照 submission format 填寫 google 表單即可
- 為確保程式碼的格式不會跑掉，都請上傳 base64 encode 後的程式碼，相關教學如下：https://linuxhint.com/bash_base64_encode_decode/
- 螢幕截圖可上傳在任何助教可以看到的外部空間
- Google Meet demo 的題目請同學上傳會議網址，助教收到後就會加入會議檢查。

Shell Scripting

For problems that involve shell scripts:

- **Writing a script that just print sample output will get 0 points**
- Start your script with a *shebang* (i.e,. `#!`) and make it executable (`chmod +x`).
- Your script should be executed with one of the following shells: `bash`, `sh`, `zsh`, `tcsh` or `ksh`.
- Only the following programs can be used in your shell script:
  - All built-in syntax and commands **provided by the shell itself** (e.g. `read`, `while`, ...).
  - Text processing: `head`, `tail`, `sed`, `tr`, `paste`, `grep`, `cut`, `awk`, `sort`, `uniq`, `wc`
  - Output: `echo`, `printf`
  - File manipulation: `touch`, `mkdir`, `mktemp`, `mkfifo`, `rm`, `mv`, `cp`, `ls`, `find`, `cat`, `stat`, `pwd`
  - Calculation: `bc`
  - Process manipulation: `ps`, `kill`, `lsof`
  - Remote connection: `ssh`-related commands
  - Utilities: `xargs`, `tar`-related commands (including `gzip`, `xz`, etc.)
  - System information: `date`, `uptime`
  - Programs explicitly allowed in the problem description.

1 Wireshark (35 points)

Resources

- Problem 1.1: p1-1.pcapng
- Problem 1.2: p1-2.pcapng

1.1 Tons of packets ★★★★★ (20 points)

The Address Resolution Protocol (ARP) is a communication protocol used to discover the link layer address, such as a MAC address, associated with a given Internet Protocol address, typically an IPv4 address. This mapping is crucial for the Internet to function correctly.

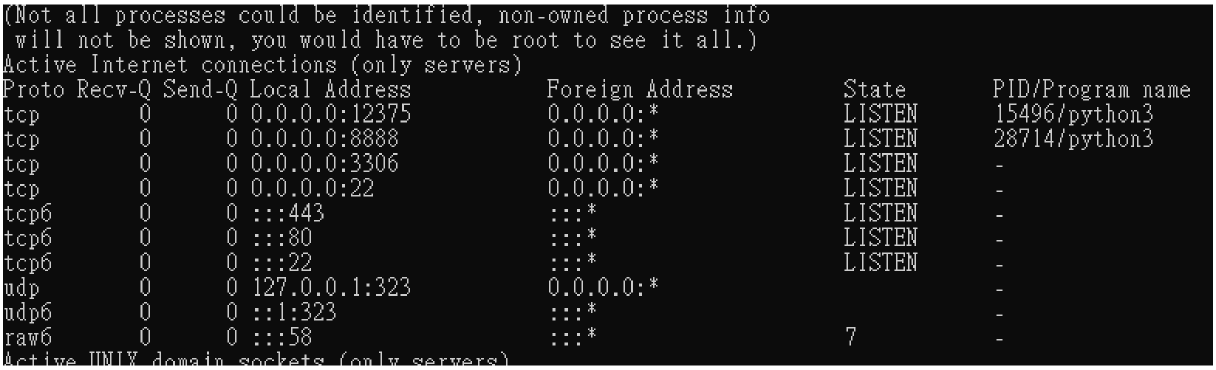
Pie's linux host with IP(10.1.23.45) got tons of packets which were recorded in `p1-1.pcapng`. Assume that the ARP table on Pie's host is initially empty. Your job is to determine the expected ARP table on Pie's host after receiving all packets captured in `p1-1.pcapng`. Output it as a `b64encode` file `arp-table.txt`. The output format should be

```
[IP] [MAC-Address]
[IP] [MAC-Address]
...
```

- Note 1: The order of the entries in the ARP table does not matter; sorting will be performed prior to judging.
- Note 2: The use of tools other than Wireshark IS permitted.
- Note 3: You can assume that the entries in the ARP table do not expire for this problem.

1.2 Pie messed up everything ★★★★☆ (15 points)

(a) (8 points) Pie tries to run an HTTP server at http://140.112.30.243:12375. However, he found that, except on the same host which the HTTP server is running (i.e., 140.112.30.243), no other host is able to connect to this web server at the URL indicated above. To debug this situation, he runs `netstat` on 140.112.30.243, whose output is shown in the screenshot below. (Another unrelated HTTP server is running on the same host at http://140.112.30.243:8888, which is the HW1 server you have seen before and can be seen in the output)



```
(Not all processes could be identified, non-owned process info
 will not be shown, you would have to be root to see it all.)
Active Internet connections (only servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State       PID/Program name
tcp        0      0 0.0.0.0:12375           0.0.0.0:*               LISTEN      15496/python3
tcp        0      0 0.0.0.0:8888            0.0.0.0:*               LISTEN      28714/python3
tcp        0      0 0.0.0.0:3306            0.0.0.0:*               LISTEN      -
tcp        0      0 0.0.0.0:22              0.0.0.0:*               LISTEN      -
tcp6       0      0 :::443                  :::*                    LISTEN      -
tcp6       0      0 :::80                   :::*                    LISTEN      -
tcp6       0      0 :::22                   :::*                    LISTEN      -
udp        0      0 127.0.0.1:323           0.0.0.0:*                           -
udp6       0      0 ::1:323                 :::*                                -
raw6       0      0 :::58                   :::*                    7           -
Active UNIX domain sockets (only servers)
```

To be able to connect to the HTTP server, instead of figuring out the reason of not being able to connect to `http://140.112.30.243:12375` from other host, Pie tries some tricks on his own computer (not the same host where he runs the HTTP server). Upon doing this, he was able to connect to the HTTP server (though not using the same URL). Please examine the Wireshark file `p1-2.pcapng` captured from Pie's computer, and determine what he had done to be able to connect to the HTTP server. Also, please briefly explain the reason that we could not reach the HTTP server at `http://140.112.30.243:12375` initially. Note that there could be multiple possible reasons, and you only need to give one.

(b) (7 points) A different HTTP server runs on 140.112.30.243 at `http://140.112.30.243:8887`, and encounters the same problem. You have access to a restricted account, which allows you to SSH to 140.112.30.243 with username `piepie01` and password `nasa2020`. Once you SSH into the host, you are only able to run `netstat`, but not any other command.

Please attempt to connect to the HTTP server running on 140.112.30.243 on port 8887, perhaps using the same trick Pie used. Take a screenshot of the browser after you successfully connect to that HTTP server.

Submission Format

- Problem 1.1: upload `b64encode file` named `arp-table.txt` to Google drive or other online file storage, and submit the link via Google form in `screenshot link` section
- Problem 1.2(a): submit `short answer` via Google form (Problem ID on Google form is 1.21)
- Problem 1.2(b): submit `screenshot link` via Google form (Problem ID on Google form is 1.22)

2 Cisco Packet Tracer ★★☆☆☆ (25 points)

Resources

- Please perform the task on the provided pka file `2020-mid.pka`.
- Packet Tracer account: nasa.cisco.pt@yopmail.com .
- Packet Tracer password: Nasa.Cisco.Pt.217 .

Greetings. We have a ~~gift~~ assignment for you, a newly hired network administrator in the IT department. Your assignment is to configure three new switches in a new building, **Core**, **Switch1**, and Switch2, given in the following.

1. First, please perform some basic configurations on the **Core** switch:
   - Name is important. Set the name of the device to be `Core`.
   - Set the username to be `admin`, password to be `nasa`, with privilege `1`, and on `line console 0`.
   - In order to improve security, set enable secret to be `bababibaba`.
2. The switches are already physically connected. (Please examine the connections in the packet tracer.) Configure trunk connections as follows. (Here, a trunk connection means a connection carrying tagged VLAN packets)

- Create a trunk connection between `Core` and `switch0`.
- Create a trunk connection between `switch0` and `switch1` and set port-channel 1 to enable link aggregation using ports GigabitEthernet 0/1-2.

3. Please set each device to be on correct VLANs and create VLAN in the switches when needed. In addition, give proper names to the VLANs, as follows:
   - The admin PC on `VLAN 99` (named `Admin`)
   - PCs are located on `VLAN 52` (named `PC`)
   - Laptops are on `VLAN 5` (named `Laptop`)
   - Servers are on `VLAN 95` (named `Server`)
4. Configure user, password, and login services. Allow the admin to telnet and SSH to Core switch, specified below:
   - Set the domain name of the **Core** switch to be `nasa.midterm.com`.
   - Only allow SSH connection on `line vty 0 3`, with password `nasassh`.
   - Only allow telnet connection on `line vty 4 7`, with password `nasatelnet`.
   - Disable the connection on `line vty 8 15`.
   - Encrypt the passwords for better security.

Submission Format

- Google Meet Demo – briefly show and explain what you did within the packet tracer using the supplied pka file.

3 Data Visualizer ★★★☆☆ (25 points)

Resources

- `example_folder.zip` contains the files used in this problem.

As a CSIE student, knowing how to submit homework/project correctly is important. To avoid missing any files in a submitted folder, we need some useful visualization tools!

Notes

- Only shell script is permitted. Other programming languages are prohibited.
- Ignore hidden files or folders, such as `.git`.

3.1 Basis Visualization (5 points)

Visualize the content of a folder as follows.

- In this sub problem, you always visualize the default path, which is the current folder, represented by a single dot `'.'`.

- The first line of the output is the path to the folder to be visualized. The second line should contain only a character `'|'`.
- For each file or folder in the folder to be visualized, output a `'|'` and a space (`' '`) before the name of the file or folder. Please sort the output lines such that the names are in alphabetical order. Note that for this sub problem, the script only need to list the files and the folders directly within the current folder.

Sample Input/Output

- Input

```
bash visualize.sh
```

- Output

```
.
|
| example_folder
| visualize.sh
```

3.2 Certain path (5 points)

In this sub problem, the script should take the argument `-p`, which specifies the path to the folder to be visualized. Similarly, this should be in the first line of the output.

Sample Input/Output

- Input

```
bash visualize.sh -p example_folder
```

- Output

```
example_folder
|
| file-A.txt
| subfolder_1
| subfolder_2
| subfolder_3
| test_img.png
```

3.3 Tree structure (10 points)

The visualization tool should now include files and folders within the first-level folders directly inside the folder to be visualized. Please use the argument `-l` to specify the number of levels. The default value for this argument is 1, where the tool would have the same output of the previous sub problems. For files or folders at level `n` (the files and folders directly inside the folder to be visualized is at level1), the output of that file or folder should have `n` times of `----` between `"|"` and `space`, followed by the name. See below for an example.

Sample Input/Output

- Input

```
bash visualize.sh -p example_folder -l 3
```

- Output

```
example_folder
|
| file-A.txt
| subfolder_1
|---- file-A.txt
| subfolder_2
| subfolder_3
|---- src
|-------- secret.png
| test_img.png
```

3.4 Do searching ! (5 points)

The tree structure sometimes is too huge, and it becomes quite difficult to find what we want. Here we would like to implement a search function within the tool, using the argument `-s` to pass the search string.

- Only the files and the folders with names containing the search string would be output.
- All parent folders (i.e., ancestor) of a matching file/folder should also be included in the output. See below for an example – the parent folders of `secret.png` are also listed.

Sample Input/Output

- Input

```
bash visualize.sh -p example_folder -l 3 -s .png
```

- Output

```
example_folder
|
| subfolder_3
|---- src
|-------- secret.png
| test_img.png
```

Submission Format

- Google Meet Demo

4 pfSense ★★★☆☆ (25 points)

Resources

- ubuntu.ova (OS for VM in VLAN 99)
- alpine-standard-3.11.5-x86_64.iso (OS for VMs in VLAN 5 and VLAN 8)
- pfSense-CE-2.4.4-RELEASE-p3-amd64.iso

Your task in this problem is to set up a pfSense on your computer, then set up three VLANs with ID 5, 8, and 99, on pfSense. The VLANs should satisfy the requirements below. The three VMs should also be configured to connect to those VLANs, respectively. Note that the OS of the VMs connected to VLAN 5 and VLAN 8 should be **Alpine**. The VM connected to VLAN 99 should use the `ubuntu.ova` file we provide. **Each client machine should have only one IP, assigned by the DHCP server running on the VLANs mentioned above.**

1. (1 pt) Specify the range of IPs you assign to each VLAN, in CIDR notation, i.e., x.x.x.x/xx.
2. (3 pts) The pfSense VM should obtain an IP issued by the same DHCP server that issue your (VM) host an IP. This should be the IP binded to the WAN interface of pfSense. Briefly explain the steps you performed to achieve this and show the IPs of your host and pfSense.
3. (4 pts) Hosts in VLANs other than VLAN 99 cannot initiate connections to hosts in VLAN 99. The opposite, however, should be allowed. Briefly explain the rules you created to implement this.
4. (4 pts) Hosts in VLAN 5 are only allowed to access *linux1.csie.ntu.edu.tw* and DNS server. Show that you can ping linux1.csie.ntu.edu.tw but cannot ping other sites.
5. (7 pts) Set up the VM with the provided Ubuntu ova file in VLAN 99, and make sure that only hosts within VLAN 99 can access to the pfSense administration interfaces (SSH and web GUI). SSH and web GUI connections from those in other VLANs should be refused.

   Briefly explain your steps to implement this, and show that you can SSH to pfSense or connect to the web GUI from the Ubuntu VM. Also, show that SSH or connect to web GUI from hosts within VLAN 5 is refused.
6. (6 pts) Hosts in VLAN 8 and VLAN 99 should be configured to have connectivity to the Internet. Please briefly explain the steps you have taken and show the result of "traceroute www.google.com" from hosts in VLAN 8.

Notes

- The account/password of the provided Ubuntu ova is root/midterm.
- You cannot use the WAN interface IP to connect to pfSense web GUI as what we did in the pfSense lab.

Requirements

- pfSense should be configured in web configurator without using `Diagnostics > Command Prompt`.

Hints

- VirtualBox supports many different modes of networking. (Network Adapter)
- Aliases may be very useful.

Submission Format

- Google Meet Demo

5 Longest Running Processes ★★☆☆☆ (20 pts)

Please write a shell script to find out the process that runs for the longest time on a machine with Linux Operating System.

1. If the script is executed without any argument, please output the effective user ID, process ID, and accumulated CPU time.
2. If the argument `-u` exists, for each user with running processes, output one line which include the user ID, the number of processes, and accumulated CPU time of the his/her longest running process.
3. If both the arguments `-u` and `-t TIME` exist, only the processes that run more than `TIME` should be considered.

Notes

1. There may be a lot of processes running on the system; please make sure that your script runs fast enough (< 1 minute).
2. Creating temporary files is forbidden.
3. You can design your own output format as long as you can document it clearly.

Submission Format

- base64 shell script.
- Google Meet Demo

6 COVID Reporter ★★★☆☆ (25pts)

Create a shell script to report the top infected regions of COVID-19.

Resources

Thanks to European Centre for Disease Prevention and Control, scientists are able to use the latest data analyzing the trend of COVID-19.

For simplicity, we will use the data provided by ECDC on April 10-th. Please use the provided file `COVID19-data-0410.csv`.

Notes

- Please make sure your script **works correctly on CSIE workstations**. We'll test your script there.
- You are allowed to create temporary files in the testing environment, but make sure that **no other files** would be deleted due to the creation of these files (you do not know the content in the directory). Also, your temporary files should be deleted after your script terminates successfully.
- Your script could run for several seconds; however, your script should consume **less than 90** seconds in CPU time.

6.1 Help! (5 pts)

You'll get full credits in this part if your script satisfy the specifications below.

The help message provides information about how to use the script to users, specified below:

```
usage:  ./COVID-reporter.sh CSV_FILE [-h] [-n NUM] [-t | -m] [-i DATE1 DATE2]
-h              Print help message
-n NUM          Print the most infected NUM countries. Default is 20
-t              Sort based on confirmed cases
-m              Sort based on deathes
-i DATE1 DATE2  Use the time interval between DATE1 and DATE2 (inclusive).
                The data format is YYYY-MM-DD. Default is to use all history data.
```

The script takes a csv file as a positional argument and several optional arguments. Please print the help message if one of the following conditions is met:

- No argument is given, or the flag `-h` is given.
  Example: `./COVID-reporter.sh -h`
- The csv file is not given.
  Example: `./COVID-reporter.sh -n 15`
- The two mutually exclusive flags `-t` and `-m` are given simultaneously.
  Example: `./COVID-reporter.sh data.csv -t -m`
- The optional arguments are provided in the wrong way.
  Hint: Using *regex* is would be fine.
  Example: `./COVID-reporter.sh data.csv -n -1`, `./COVID-reporter.sh -i 03-03 2020-4-8` or `./COVID-reporter.sh -n ALL -i lastmonth today`

Submission Format

- base64 shell script.

6.2 Main Task(12 pts)

You'll get 6 points if your script can correctly process the sample test cases. There are another 6 points for additional test cases.

Please implement the three flags, `-n`, `-t`, `-m`. Print the first `NUM` infected regions. For simplicity, please output one region in one line using `"%-30s%8d", region, number` as the format. Mark the first five line in red and the next five line in yellow, and use the default font color for all other lines.

Sample Input/Output

1. - Input
     `./covid-reporter.sh COVID19-data-0410.csv`
   - Output
     Please refer to `sample-2-1.output`
2. - Input
     `./covid-reporter.sh COVID19-data-0410.csv -n 15 -m`
   - Output
     Please refer to `sample-2-1.output`

Submission Format

- base64 shell script.

6.3 Customized Time Interval (8 pts)

You'll get 4 points if your script can correctly process the sample test cases. There is another 4 points for additional test cases.

Implement the `-i` flag, followed by two dates in the format `YYYY-MM-DD`. Note that the original functionality should continue to work.

Sample Input/Output

- Input
  `./covid-reporter.sh COVID19-data-0410.csv -i 2020-03-01 2020-03-31`
- Output
  Please refer to `sample-3.output`

Submission Format

- base64 shell script.

Reference

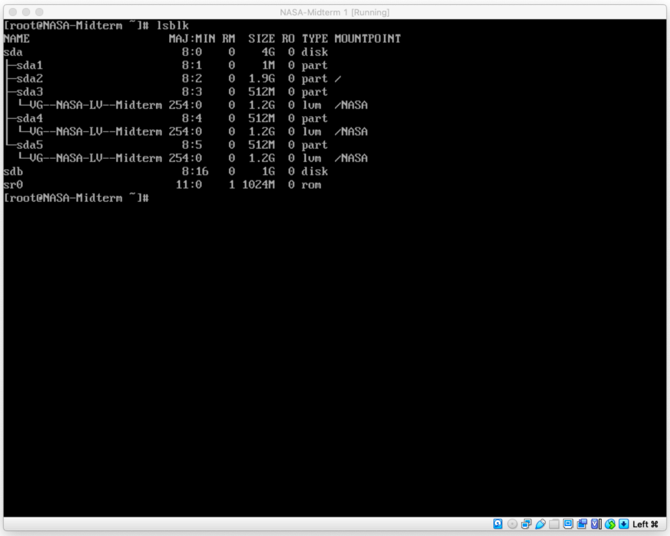
- ECDC - Download today's data on the geographic distribution of COVID-19 cases worldwide
- Google - COVID-19 Map

7 Storage ★★☆☆☆ (25 points)

Resources

- VM Image : `NASA-Midterm.ova`
- OS : Arch Linux
- Username: `root`

- Password: `nasa2020`
- Original Partition Setting:



```
[root@NASA-Midterm ~]# lsblk
NAME                     MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
sda                        8:0    0    4G  0 disk
├─sda1                     8:1    0    1M  0 part
├─sda2                     8:2    0  1.9G  0 part /
├─sda3                     8:3    0  512M  0 part
│ └─VG--NASA-LV--Midterm 254:0    0  1.2G  0 lvm  /NASA
├─sda4                     8:4    0  512M  0 part
│ └─VG--NASA-LV--Midterm 254:0    0  1.2G  0 lvm  /NASA
└─sda5                     8:5    0  512M  0 part
  └─VG--NASA-LV--Midterm 254:0    0  1.2G  0 lvm  /NASA
sdb                        8:16   0    1G  0 disk
sr0                       11:0    1 1024M  0 rom
[root@NASA-Midterm ~]#
```

Figure 1: Original Setting on the VM

7.1 Task 1 - Shrink (7pts)

Andy realizes that the LVM volume size is too big and decide to shrink it. However, he finds that there is data on all three physical volumes. Please help Andy to accomplish the mission.

1. Resize the logical volume to 800 MiB and remove the physical volume `/dev/sda5` from the volume group.
2. Make sure the data is not corrupted after all operations. Please briefly report what you did in the demo.

Submission Format

- Google Meet Demo

7.2 Task 2 - Partition Table (7pts)

Please perform the following tasks step-by-step. You can use either `fdisk` or `parted`.

- Expand `/dev/sda5`, the partition you just free from LVM, with the rest of space on `/dev/sda`.
- Create a partition `/dev/sdb1` with the entire space on `/dev/sdb`.
- The partition type should be `Linux RAID`.

Submission Format

- Google Meet Demo

7.3 Task 3 - RAID (11pts)

Build an RAID 1 volume with `/dev/sda5` and `/dev/sdb1`, and format the volume with `brtfs` file system. Then, mount it on `/NASA-<TEAMID>`, for example, `/NASA-Team01`.

Submission Format

- Google Meet Demo

8 Nested VM ★★★★★ (25 points)

In this problem, you are asked to run a virtual machine M on any CSIE workstation using QEMU-KVM. The virtual machine must satisfy the requirements below:

- M should have 3 GiB of memory and 2 logical CPUs.
- There are 4 virtual machines M_1, M_2, M_3, M_4 running on M (using QEMU-KVM). Each of them has 512 MiB of memory and 1 logical CPU.
- All 5 VMs use Arch Linux as their operating system.
- The total real size (output of `du -B1 [img]`) of the image file(s) of M should be as small as possible. You will get credit if the total size is less than 6 GiB, and more credit for having a size less than 3.5 GiB. (Tricks such as placing some images on `tmpfs` are not allowed.)
- The network of 5 VMs should be set up as follows:
  - M has 2 'real' NICs (which means they are not created within M) and one virtual network interface. The first real NIC is used to access the Internet; the second real NIC and the virtual interface is for creating local networks (described below).
  - Each of M_1, M_2, M_3, M_4 has 1 NIC.
  - M_1 uses NAT (provided by QEMU user networking) to access the Internet.
  - M_2 is in the same network with M's first 'real' interface. As a result, it can use DHCP to get an IP and access the Internet.
  - M's second real interface and M_3 are in the same local network; M's virtual interface and M_4 are in another local network. Packets should not be forwarded between these two networks and the Internet. Their IP should be `192.168.3.100` (M's second real interface), `192.168.3.103` (M_3), `192.168.4.100` (M's virtual interface) and `192.168.4.104` (M_4), respectively. (Use `/24` subnets.)
  - All network settings must be preserved across reboots.
- One can SSH into M, M_1 from the CSIE workstation using a private key.
  (that is, one can use `ssh -i [some_key] -p [some_port] [some_user]@[some_workstation]` to log in to M and M_1; the port number will be different for M and M_1.)
- One can SSH into M_2, M_3, M_4 from M (by `ssh -i [some_key] [IP]`).
- You should not use SSH tunnel or `iptables` rules to satisfy the requirements of SSH or networking. (However, using these when setting up the VMs is allowed.)

Subtasks & Scores

(1) (5 pts) Install M and complete its SSH setup.

(2) (2 pts) Install M_1 and complete its network / SSH setup (and the corresponding setup on M).

(3) (3 pts) Install M_2 and complete its network / SSH setup (and the corresponding setup on M).

(4) (3 pts) Install M_3 and complete its network / SSH setup (and the corresponding setup on M).

(5) (2 pts) Install M_4 and complete its network / SSH setup (and the corresponding setup on M).

(6) (5 pts) Finish all network settings and make all network settings persistent across reboots.

(7) (3 pts) Install all 5 VMs and meet the 6 GiB space requirement.

(8) (2 pts) Install all 5 VMs and meet the 3.5 GiB space requirement.

- You can get the points of subtasks (1)~(5) regardless of whether the settings are persistent across reboots.

Submission Format

- Short answer containing the following information:
  - The workstation where M resides.
  - The process ID and path to the image of all 5 VMs (for M, it is the PID and path on the workstation; for $M_1, \cdots, M_4$, they are the PIDs and paths on M).
  - The commands you used to start $M_1, \cdots, M_4$. (We will test whether the settings are preserved by rebooting M and start $M_1, \cdots, M_4$ again.)
  - The IP of M_2.
  - The port (for M and M_1 only), username and private key file to SSH into all 5 VMs. (The given user must have the permission to execute any commands by `sudo` without password. You may use the same username and private key file for convenience.)
  - If you didn't finish all of the subtasks, specify the subtasks you've done.
- Google Meet Demo (after we checked your VMs)

Hints

- The real size of the image of **one** freshly-installed Arch Linux should be around 1.8 GiB or 2.6 GiB (with QEMU). How can you create 5 VMs in 6 GiB or even 3.5 GiB?
- The previous hint can also save you the time of installing and setting up VM!
- VNC is useful when installing VMs.
- If your VM runs VERY SLOW, double-check your VM settings.
- `netctl` is also useful.
- You are advised to create the VM image in `/tmp2`. Check the remaining disk space and memory first.

9 Bob Needs Your Help ★★☆☆☆ (15 points)

In this problem, you are given a virtual machine image. There are two docker containers, Alice and Bob, in the virtual machine. However, it seems that Bob remains unconscious at exited status. It is time for you to come to his rescue, helping Bob to recover. The main process of Bob is a custom shell script which may give you some instructions. Try to run the container and access it to get the flag inside.

Resources

- VM login credential:
  - username: `root`
  - password: `root`

Hints

- Try to gain more information of the containers and realize the situation.
- If you encounter `Operation not permitted` issue, simply switch to `privileged` mode.
- Item in Bob's bag may help, but don't try to open it. It's a huge file.
- We highly recommend that you run the container in background.

Submission Format

- short answer : `Flag` in the container `Bob`
- Google Meet Demo

10 KVM + Docker ★★☆☆☆ (10 points)

Create a virtual machine guest via `libvirtd` on a virtual machine host just as in HW3. Inside the guest OS, run two containers as simple websites with Apache (called `httpd` on CentOS). Suppose the guest OS uses `GUEST_IP`, configure your VMs so that the following result can be obtained when running commands `curl GUEST_IP:8081/` and `curl GUEST_IP:8082/` on the host:

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="UTF-8">
        <title>KVM + Docker</title>
    </head>
    <body>
        <h1>We are team_TEAMTOKEN.</h1>
    </body>
</html>
```

Subtasks

(1) (4 pts) Create a virtual machine guest named `team_TEAMTOKEN` and login to it via any console. Provide the screenshots taken after running command `virsh list` on the host and the command used to login to the guest.

(2) (6 pts) VM host is able to get the required result with `curl` commands. Provide the following screenshots:

- After running command `ip addr` on both VM host and guest
- After running command `docker ps` on the guest. Also, provide the content of your dockerfile.
- After running commands `curl GUEST_IP:8081/` and `curl GUEST_IP:8082/` on the host

Resources

- VMWare VM image in the `KVM` folder.
  - OS : CentOS 7
  - uid/password: root/nasa
  - installation image for the guest: /var/lib/libvirt/images/CentOS7.iso
  - anaconda kickstart script for the guest: /root/midterm-anaconda-ks.cfg

Notes

- Install Docker and pull CentOS 7 with the following commands:

```
yum install yum-utils device-mapper-persistent-data
yum-config-manager --add-repo https://download.docker.com/linux/centos/docker-ce.repo
yum install docker-ce
systemctl start docker
systemctl enable docker
docker pull centos:7
```

Submission Format

- Screenshots specified above.