

The first image server.js used based on exercise from this course.

```
FROM debian
USER root
COPY package.json ./
WORKDIR /
RUN apt-get update
RUN apt-get install -y nodejs
RUN apt-get install -y npm
RUN npm install
COPY . .
ENTRYPOINT node server.js
```

Build first image.

```
winx@Winx-MBP dockerImage1 % docker build -t server1ssh .
[+] Building 74.1s (11/11) FINISHED
=> [internal] load build definition from Dockerfile
=> => transferring dockerfile: 214B
=> [internal] load .dockerignore
=> => transferring context: 34B
=> [internal] load metadata for docker.io/library/debian:latest
=> [internal] load build context
=> => transferring context: 34B
=> [1/7] FROM docker.io/library/debian@sha256:45ee40a844048c2f6d0105899c1a17733530b56d481612608aab5e2e4048570b
=> CACHED [2/7] COPY package.json ./
=> CACHED [3/7] RUN apt-get update
=> CACHED [4/7] RUN apt-get install -y nodejs
=> [5/7] RUN apt-get install -y npm
=> [6/7] RUN npm install
=> exporting to image
=> => exporting layers
=> => writing image sha256:a0f7ddcf4276f446e58983d3f24d212f8029673ac1169b797d1309bb49051dc0
=> => naming to docker.io/library/server1ssh
```

The second image changed permitrootlogin to yes in the file 'sshd_config' and created ssluser with password eee in dockerfile, the net-tools installed then later can use ifconfig to see the ip address when the iamge is running.

```
FROM server1ssh
RUN apt-get install -y openssh-server
RUN sed -i 's/PermitRootLogin prohibit-password/PermitRootLogin yes/' /etc/ssh/sshd_config
RUN useradd -m -s /bin/bash -G sudo -p $(openssl passwd -1 eee) ssluser

RUN apt-get install -y python3
RUN apt-get install -y sudo
RUN apt-get install -y net-tools
ENV PORT=8081
EXPOSE 22
ENTRYPOINT service ssh start && node server.js
```

Build second image

```
winx@Winx-MBP dockerImage2 % docker build -t sshlogin .
[+] Building 8.5s (11/11) FINISHED
=> [internal] load build definition from Dockerfile
=> => transferring dockerfile: 434B
=> [internal] load .dockerignore
=> => transferring context: 2B
=> [internal] load metadata for docker.io/library/server1ssh:latest
=> [1/7] FROM docker.io/library/server1ssh
=> [2/7] RUN apt-get install -y openssh-server
=> [3/7] RUN sed -i 's/PermitRootLogin prohibit-password/PermitRootLogin yes/' /etc/ssh/sshd_config
=> [4/7] RUN useradd -m -s /bin/bash -G sudo -p $(openssl passwd -1 eee) ssluser
=> [5/7] RUN apt-get install -y python3
=> [6/7] RUN apt-get install -y sudo
=> [7/7] RUN apt-get install -y net-tools
=> exporting to image
=> => exporting layers
=> => writing image sha256:72f8874c988d6c48d3317de161a85235c32e3281bf2aa271ea7f91769023ef47
=> => naming to docker.io/library/sshlogin
```

List both of the docker image:

```
winx@Winx-MBP dockerImage2 % docker image ls
```

REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
sshlogin	latest	72f8874c988d	31 seconds ago	980MB
server1ssh	latest	a0f7ddcf4276	2 minutes ago	959MB

Run docker image and list docker container:

```
winx@Winx-MBP ansible % docker container ls
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
eacf7f43e590	sshlogin	"/bin/sh -c 'service..."	57 seconds ago	Up 56 seconds	22/tcp	loving_hamilton
e70c4a80c97b	server1ssh	"/bin/sh -c 'node se..."	9 minutes ago	Up 9 minutes		inspiring_euler

Get ip address of the container by "docker exec container_id ifconfig"

```
winx@Winx-MBP ansible % docker exec eacf7f43e590 ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 172.17.0.3 netmask 255.255.0.0 broadcast 172.17.255.255
    ether 02:42:ac:11:00:03 txqueuelen 0 (Ethernet)
    RX packets 11 bytes 866 (866.0 B)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 0 bytes 0 (0.0 B)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
    loop txqueuelen 1000 (Local Loopback)
    RX packets 0 bytes 0 (0.0 B)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 0 bytes 0 (0.0 B)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

I run the docker image with port 2222:22, at this point it is possible to start the image and ssh. If using localhost or machine's IP with port 2222, it will communicate to port 22 inside the container.

```
winx@Winx-MBP ansible % docker run -p 2222:22 ba74ae0cb59a
Starting OpenBSD Secure Shell server: sshd.
Listen on 8081
```

The ip address of the second image which installed openssh is 172.17.0.3 which is the docker internal ip. I used "sshsslruser@172.17.0.3" with password "eee" to login:

```
winx@Winx-MBP ansible % docker container ls
CONTAINER ID   IMAGE                                COMMAND                                  CREATED        STATUS        PORTS                               NAMES
043ae4c358e0   ba74ae0cb59a                        "/bin/sh -c 'service..." 13 seconds ago Up 11 seconds 0.0.0.0:2222->22/tcp interesting_mahavira
e70c4a80c97b   bd6ec3151aa                        "/bin/sh -c 'node se..." About an hour ago Up About an hour 0.0.0.0:8080->8080/tcp inspiring_euler
7ec35c499b55   group-41-develop_server-a          "docker-entrypoint.s..." 7 months ago   Up 2 days     0.0.0.0:8080->8080/tcp group-41-develop_server-a_1
509fa75141cc   mongo:4.4                           "docker-entrypoint.s..." 7 months ago   Up 2 days     27017/tcp group-41-develop_db_1

winx@Winx-MBP ansible % docker port 043ae4c358e0 22
0.0.0.0:2222

winx@Winx-MBP ansible % docker exec 043ae4c358e0 ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 172.17.0.3 netmask 255.255.0.0 broadcast 172.17.255.255
    ether 02:42:ac:11:00:03 txqueuelen 0 (Ethernet)
    RX packets 10 bytes 796 (796.0 B)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 0 bytes 0 (0.0 B)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
    loop txqueuelen 1000 (Local Loopback)
    RX packets 0 bytes 0 (0.0 B)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 0 bytes 0 (0.0 B)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

winx@Winx-MBP ansible % ssh sslruser@172.17.0.3
ssh: connect to host 172.17.0.3 port 22: Operation timed out
winx@Winx-MBP ansible % ssh sslruser@localhost -p 2222
The authenticity of host '[localhost]:2222 ([::1]:2222)' can't be established.
ECDSA key fingerprint is SHA256:fpY/W+Z0UVhp7k1qPAy4B1SM7x1d0dSysM2zc26KDLg.
Are you sure you want to continue connecting (yes/no/[fingerprint])? y
Please type 'yes', 'no' or the fingerprint: yes
Warning: Permanently added '[localhost]:2222' (ECDSA) to the list of known hosts.
sslruser@localhost's password:
Linux 043ae4c358e0 5.10.25-linuxkit #1 SMP Tue Mar 23 09:27:39 UTC 2021 x86_64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
sslruser@043ae4c358e0:~$
```

The ansible installed in own machine, and the inventory file was created in the path ~/.ansible/inventory.

```
[winx@Winx-MBP / % cd ~
[winx@Winx-MBP ~ % mkdir ansible
[winx@Winx-MBP ~ % cd ansible
[winx@Winx-MBP ansible % nano inventory
```

Get host from inventory:

```
winx@Winx-MBP ansible % ansible-inventory -i inventory --list
{
  "_meta": {
    "hostvars": {}
  },
  "all": {
    "children": [
      "ungrouped"
    ],
    "ungrouped": {
      "hosts": [
        "192.168.50.155"
      ]
    }
  }
}
```

Since there is no default ansible folder, it is created with “sudo mkdir ansible” . it is also needed to create hosts file in “~/etc/ansible” folder:

```
winx@Winx-MBP /etc % cd ansible
winx@Winx-MBP ansible % ls
winx@Winx-MBP ansible % nano hosts
winx@Winx-MBP ansible % sudo nano hosts
winx@Winx-MBP ansible % sudo nano hosts
```

I used password ssh login, there are settings need to be change in sshd_config file:

“passwordauthentication yes”

“Permitrootlogin no”

“PubKeyAuthentication yes”

Then run the playbook, but seems missing sshpass:

```
winx@Winx-MBP ansible % ansible-playbook playbook/playbookdemo.yml
PLAY [My playbook] *****
TASK [Gathering Facts] *****
fatal: [127.0.0.1]: FAILED! => {"msg": "to use the 'ssh' connection type with passwords or pkcs11_provider, you must install the sshpass program"}
```

Sshpass installed on own machine by the following command:

▲ There are instructions on how to install sshpass here:

177 <https://gist.github.com/arunoda/7790979>

▼ For Mac you will need to install xcode and command line tools then use the unofficial Homewbrew command:



```
curl -L https://raw.githubusercontent.com/kadwanev/bigboybrew/master/Library/Formula
```

Then with the “—ask-pass” appended, it is possible to enter password while do ssh login, but it seems own machine is missing “sshpass” :

```
winx@Winx-MBP ansible % ansible-playbook playbook/playbookdemo.yml --ask-pass
SSH password:
PLAY [My playbook] *****
TASK [Gathering Facts] *****
ok: [192.168.58.155]

TASK [check latest] *****
fatal: [192.168.58.155]: FAILED! => {"changed": true, "cmd": ["apt-get", "update"], "apt-get": "install", "-y", "git"], "delta": "0:00:00.004058", "end": "2021-12-15 23:58:32.437158", "msg": "non-zero r
eturn code", "rc": 100, "start": "2021-12-15 23:58:32.432280", "stderr": "E: Invalid operation update", "stderr_lines": ["E: Invalid operation update"], "stdout": "", "stdout_lines": []}

PLAY RECAP *****
192.168.58.155 : ok=1 changed=0 unreachable=0 failed=1 skipped=0 rescued=0 ignored=0
```

Playbook:

```
playbook > ! playbookdemo.yml
1  ---
2  - name: My playbook
3    hosts: server
4    remote_user: ssluser
5    tasks:
6      - name: check latest
7        #command: "apt-get update; apt-get install -y git"
8        apt:
9          name: git
10         state: present
11         update_cache: yes
12      - name: list uptime
13        shell: uptime
14        register: hostname
15
```

After install " sshpass" locally, it is possible to get the result of playbook:

```
winx@winx-MBP ansible % ansible-playbook playbook/playbookdemo.yml --ask-pass
SSH password:
PLAY [My playbook] *****
TASK [Gathering Facts] *****
ok: [192.168.58.155]
TASK [check latest] *****
```

I dont know why even if I rerun the playbook, the result is the same.

I used macbook but it is not very easy to set up the environment, I spent quite many times for the ansible path, and the host file, and the ansible documentation. I did the login with password, then there were some issue with it, e.g. missing sshpass in own machine. I think my result is not very good, I didn' t attach the O2, O3, O4, because somehow even if I re-run the playbook, it didn' t change the output.