

Cloud Computing and Deployment

1. N01-Linux-VM

Create one instances of a virtual environment (e.g. using Docker or VirtualBox) on one physical host computer.

Then install linux and the OpenSSH server (and <https://www.openssh.com/>) and create OpenSSH access keys for one remote user.

Then, show that you can connect to this hosted virtual environment using the OpenSSH client from the host computer, passwordless using SSH key authentication.

Submit one merged shell script, renaming it with extension .txt (not .ZIP). This .txt should contain: 1) a script / code (which are not comments), 2) printout of the running of the script (specify this as the commented part of the shell script) and 3) a description of the steps (also as comments, with UTF-8 encoding).

FOR HALF POINTS: without client demonstration for OpenSSH

Tips:

<https://help.ubuntu.com/community/SSH/OpenSSH/Keys>

<https://www.virtualbox.org/>

<https://www.ubuntu.com/download/server>

<https://www.alpinelinux.org/about/>

<http://www.damnsmalllinux.org/>

<https://cdimage.debian.org/debian-cd/current/amd64/iso-cd/debian-9.2.1-amd64-netinst.iso> (install without X)

<https://2nwiki.2n.cz/pages/viewpage.action?pageId=75202968>

<https://azure.microsoft.com/en-us/free/services/virtual-machines/>

Live USB ubuntu+Docker: <https://docs.docker.com/storage/storagedriver/overlayfs-driver/> into /etc/docker/daemon.json:

```
{  
  "storage-driver": "overlay2"  
}
```

```
#!/bin/bash
```

```
cat SUBMITTED_FILE.txt | sed -n 100,350p > executable_script.sh; bash -c executable_script.sh
```

```
echo "The script prints:"
```

```
cat SUBMITTED_FILE.txt | sed -n 500,550p
```

```
#!/bin/bash
```

```
cat prva.txt druga.txt > ODDAJA.txt
```

```
#!/bin/bash
```

```
mkdir extract
```

```
sed < ODDAJA.txt > extract/prva.txt -e 1,$(wc -l prva.txt | awk '{print $1}')p -n
```

```
sed < ODDAJA.txt > extract/druga.txt -e 1,$(wc -l druga.txt | awk '{print $1}')p -n
```

We create in Virtual Box an Ubuntu machine, and then there we install Docker

1. Prerequisites

```
sudo apt-get update
```

```
sudo apt-get install \  
  apt-transport-https \  
  ca-certificates \  
  curl \  
  gnupg-agent \  
  software-properties-common
```

2. The key

```
curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo apt-key  
add -
```

3. The repository

```
sudo add-apt-repository \  
  "deb [arch=amd64] https://download.docker.com/linux/ubuntu \  
  $(lsb_release -cs) \  
  stable"
```

4. The packages

```
sudo apt-get update
```

```
sudo apt-get install docker-ce docker-ce-cli containerd.io
```

5. Its all

Now im going to install in windows, with the installation helper

Once installed, I create a docker image with alpine doing a docker pull:

```
docker pull ubuntu:latest
```

I run the image in a container with:

```
docker run -it -p 80:80 c059bfaa849c
```

Now, we are in of the container, and we are going to install the OpenSsh server

```
sudo apt-get install openssh-server openssh-client.
```

For use our ip, we need to install another packages

```
apt-get install apt-file
```

```
apt-file update
```

```
apt-file search --regexp 'bin/ip$'
```

```
iproute2: /bin/ip
```

```
iproute2: /sbin/ip
```

```
apt install -y iproute2
```

Now, we have installed the ip command, so we can see our interfaces with ip a, and we see that our ip os 172.17.0.2/16

```
root@4ff813f7d35d:/# 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
2: tunl0@NONE: <NOARP> mtu 1480 qdisc noop state DOWN group default qlen 1000
    link/ipip 0.0.0.0 brd 0.0.0.0
3: sit0@NONE: <NOARP> mtu 1480 qdisc noop state DOWN group default qlen 1000
    link/sit 0.0.0.0 brd 0.0.0.0
10: eth0@if11: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UP group default
    link/ether 02:42:ac:11:00:02 brd ff:ff:ff:ff:ff:ff link-netnsid 0
    inet 172.17.0.2/16 brd 172.17.255.255 scope global eth0
        valid_lft forever preferred_lft forever
root@4ff813f7d35d:/#
```

We install ifconfig with

```
apt -y install net-tools
```

So, in this point, we create a new image from this container, with

```
docker commit 4ff813f7d35d ubuntu_basic
```

With this, we can instance containers in different terminals, but for me, working on windows, is more difficult to create a .sh doc with the instructions to instance docker all the times that I want, so I am going to create two new containers from the ubuntu basic, in two different terminals

```
docker run -it ubuntu_basic
```

In this point, we install the command ping, for use the networks in docker:

```
apt-get update && apt-get install -y iputils-ping
```

Now, we can try the connection between both containers with:

In the first container: ifconfig for see the ip (172.17.0.3)

In the second container: ping 172.17.0.3

```
root@7a8d529faedf:/# ping 172.17.0.3
PING 172.17.0.3 (172.17.0.3) 56(84) bytes of data.
 64 bytes from 172.17.0.3: icmp_seq=1 ttl=64 time=3.18 ms
 64 bytes from 172.17.0.3: icmp_seq=2 ttl=64 time=0.056 ms
 64 bytes from 172.17.0.3: icmp_seq=3 ttl=64 time=0.069 ms
 64 bytes from 172.17.0.3: icmp_seq=4 ttl=64 time=0.046 ms
 64 bytes from 172.17.0.3: icmp_seq=5 ttl=64 time=0.043 ms
 64 bytes from 172.17.0.3: icmp_seq=6 ttl=64 time=0.078 ms
 64 bytes from 172.17.0.3: icmp_seq=7 ttl=64 time=0.046 ms
^C
```

Now, in the first container, create the Ubuntu hosted System, using the key generation process OpenSSH. First of all, in the first container we enter to the user efrain, and in the second, we delete this one and create another:

```
sudo userdel efrain
```

```
sudo sueradd -m usuarionuevo
```

```
sudo passwd usuarionuevo
```

For delete the folders with the info of the first users:

```
rm -rf efrain/
```

For move to the users

```
su container1
```

We install nano

```
apt-get install nano
```

Now we are in the good users, so we try ssh connection

```
$ mkdir ~/.ssh
```

```
$ chmod 700 ~/.ssh
```

```
$ ssh-keygen -t rsa
```

```
root@5fbdb8c8219c:/etc/ssh# chmod 700 ~/.ssh
root@5fbdb8c8219c:/etc/ssh# ssh-keygen -t rsa
Generating public/private rsa key pair.
Enter file in which to save the key (/root/.ssh/id_rsa): key_container2
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in key_container2
Your public key has been saved in key_container2.pub
The key fingerprint is:
SHA256:P3qeUDUaS4g448nWxR9Xp4w0Xff8o7gctRRGYRMvQgQ root@5fbdb8c8219c
The key's randomart image is:
+---[RSA 3072]-----+
|           Eoo *= o|
|      . o ...o=o=.|
|    + . + =.*+o.o|
|   o = . o Xo.o .|
|    = . S = .o ..|
|    .   o + o .|
|      . oo o   |
|      oooo    |
|      .ooo    |
+-----[SHA256]-----+
root@5fbdb8c8219c:/etc/ssh#
```

In the container2, we start the ssh server with

```
service ssh start
```

We could start ssh connection with password or with the public keys

And we try the ssh connection from the container1: `ssh container2@172.17.0.3`

Finally, I can't work in Windows with docker confortly, so I moved to a partition with Ubuntu installed

We download the Ubuntu Server 18.04.2 in the Ubuntu web.

Later, we install Virtual box, we create a new machine and then we install here the ubuntu

```
Ubuntu 18.04 LTS n01-linux-vm tty1
n01-linux-vm login: efrain
Password:
Login incorrect
n01-linux-vm login: efra
Password:
Welcome to Ubuntu 18.04 LTS (GNU/Linux 4.15.0-166-generic x86_64)

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

System information disabled due to load higher than 1.0

294 packages can be updated.
188 updates are security updates.

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

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

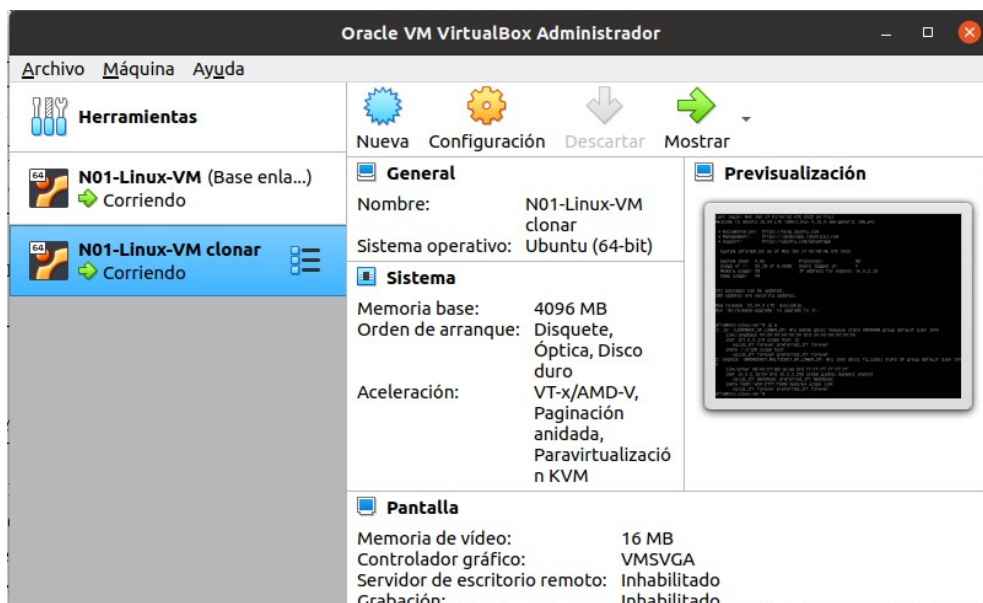
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

efra@n01-linux-vm:~$
```

We entry in the root user, and install ssh

sudo apt-get install ssh

We create another machine, cloning the first one



Then, we entry to the second machine

```
efra@n01-linux-vm:~$ ssh efra
ssh: Could not resolve hostname efra: Temporary failure in name resolution
efra@n01-linux-vm:~$ ssh efra@10.0.2.15
The authenticity of host '10.0.2.15 (10.0.2.15)' can't be established.
ECDSA key fingerprint is SHA256:fL67SVYVF57yd25Xfn6X7Aqp5yFhrBbq2dIzn1wkObI.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '10.0.2.15' (ECDSA) to the list of known hosts.
efra@10.0.2.15's password:
Welcome to Ubuntu 18.04 LTS (GNU/Linux 4.15.0-166-generic x86_64)

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

System information as of Mon Jan 17 02:11:16 UTC 2022

System load:  0.08               Processes:            93
Usage of /:   25.1% of 8.05GB    Users logged in:     1
Memory usage: 3%                IP address for enp0s3: 10.0.2.15
Swap usage:   0%

291 packages can be updated.
185 updates are security updates.

New release '20.04.3 LTS' available.
Run 'do-release-upgrade' to upgrade to it.

Last login: Mon Jan 17 02:08:40 2022
efra@n01-linux-vm:~$ _
```

We do ssh logout with exit