

MAYAN EDMS WITHIN DOCKER

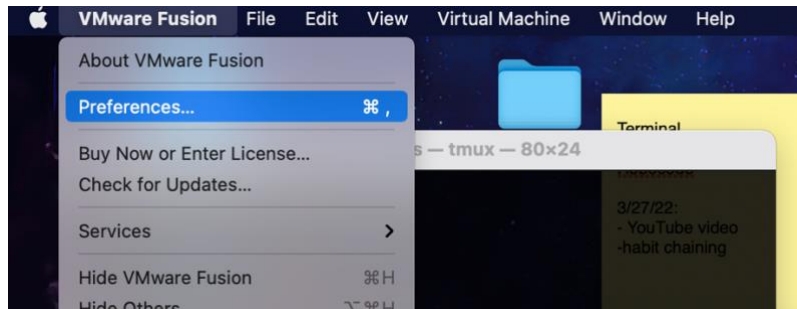
Table of Contents

<i>Setting up Ubuntu 20.04 Virtual machine</i>	3
<i>Configure a Virtual Network</i>	3
<i>Install Ubuntu 20.04 Virtual Machine</i>	4
<i>Setting up SSH</i>	6
<i>Installing Docker</i>	7
<i>Installing Mayan EDMS</i>	8
<i>Pulling Docker Images</i>	8
<i>Starting the Containers</i>	9
<i>Write-Up</i>	10
<i>References</i>	11

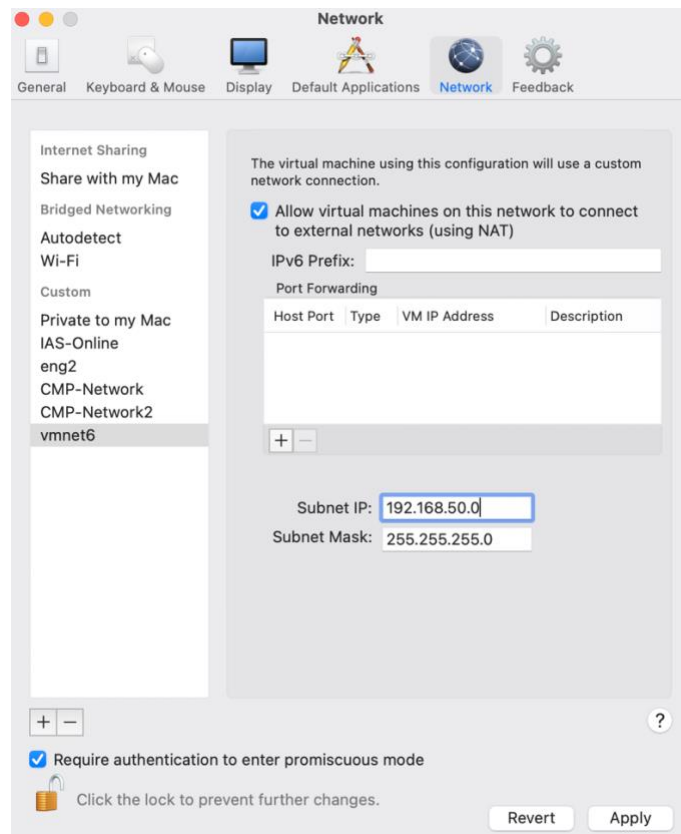
Setting up Ubuntu 20.04 Virtual machine

Configure a Virtual Network

Before anything else, our first step should be to create a virtual network for use by our virtual machine. We can accomplish this by opening VMware Fusion and navigating to the “Preferences...” tab from the drop-down menu.



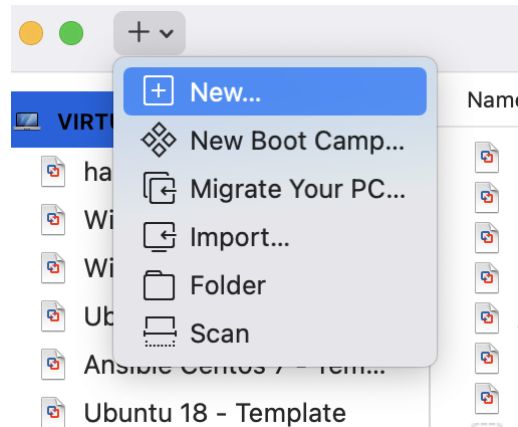
Within Preferences, navigate to the Network tab and click the lock in the bottom-left corner to make changes. Our settings for this network will be a Subnet IP of 192.168.50.0/24 (subnet mask of 255.255.255.0). We will also be ticking “Allow virtual machines on this network to connect to external networks using NAT” (MacOS Big Sur). Correct settings are shown below:



Click “Apply” to save the settings, and then click the lock again to prevent further changes.

Install Ubuntu 20.04 Virtual Machine

Alright, time to set up the virtual machine itself. In the top left of the VMware Fusion window, click on the + and select “New...”

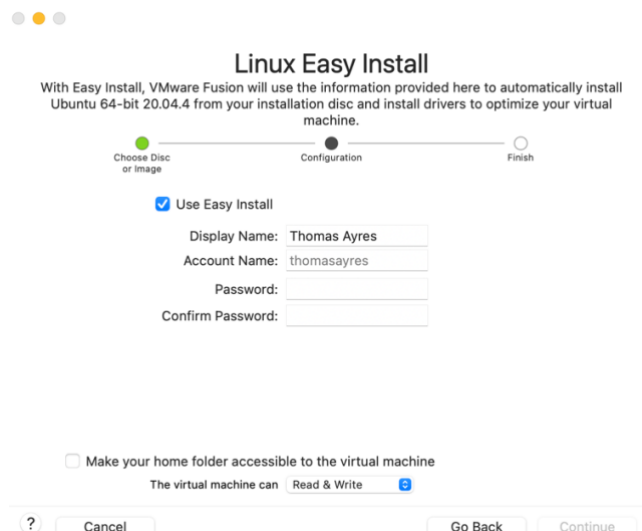


In the new window that pops up, click *Continue* and then you will be able to browse for your Ubuntu image by clicking the following button:

Use another disc or disc image...

After locating your Ubuntu ISO, Click *Continue*.

From this point, we will go ahead with the “Easy Install” option, which will be perfectly fine for the purposes of this assignment. Make sure that you punch in credentials that you will remember for later, and then click “Continue”.

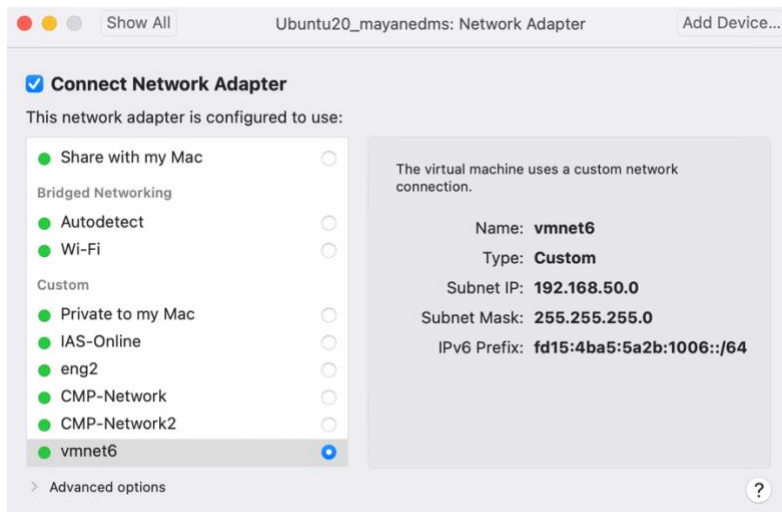
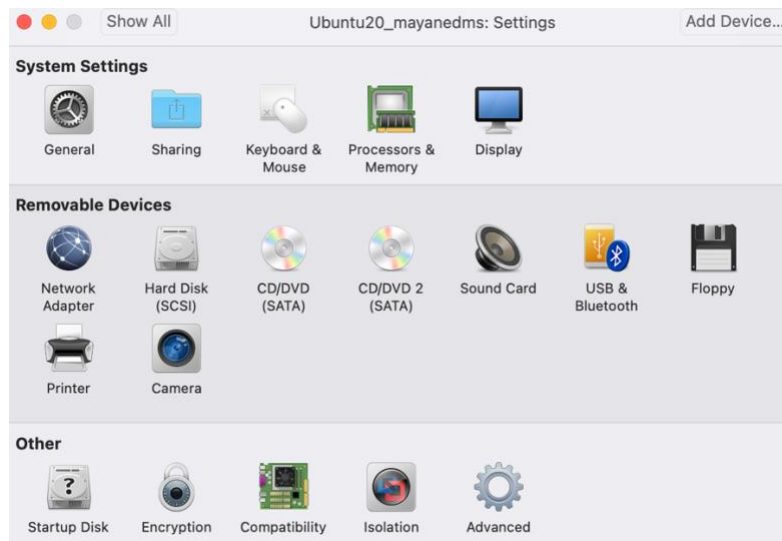


On the next screen, you will see a bunch of hardware information about your virtual machine. Click the button “Customize Settings” and save your VM under a recognizable name.

To change the default virtual machine settings, click Customize Settings. To run the virtual machine now, click Finish.

Customize Settings

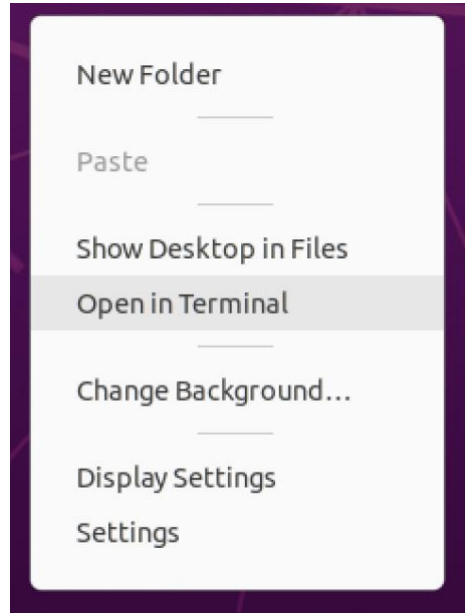
A settings window will pop up. Select “Network Adapter” and then select the virtual network that we created earlier.



If you would like to change other settings (Like storage and processor core quantity) feel free. Otherwise, feel free to start the virtual machine by clicking on the play button!

Setting up SSH

After the virtual machine finishes booting, log in using the credentials set up during the install. When on the Ubuntu desktop, right-click and select “Open in Terminal” from the drop-down menu.



After the terminal pops up, our first step will be to install OpenSSH Server. Type in the following:

sudo apt install openssh-server -y

Note: *The installer will ask for a password when “sudo” is used. Enter in the password created for the account whenever this happens now and in the future.*

After the installation completes, our next step is to open port 22 within the firewall. To do this we will type:

sudo ufw allow ssh

```
tayres@ubuntu:~/Desktop$ sudo ufw allow ssh
Rules updated
Rules updated (v6)
tayres@ubuntu:~/Desktop$
```

After the firewall is configured as such, we can then SSH into our virtual machine from the host, if desired.

Installing Docker

We will be using Docker to run a few programs. Time to install! Run the following command:

`wget -qO- https://get.docker.com/ | sh`

Your output will look similar to this:

```
Server: Docker Engine - Community
Engine:
  Version:      20.10.14
  API version:  1.41 (minimum version 1.12)
  Go version:   go1.16.15
  Git commit:   87a90dc
  Built:        Thu Mar 24 01:45:53 2022
  OS/Arch:      linux/amd64
  Experimental: false
containerd:
  Version:      1.5.11
  GitCommit:    3df54a852345ae127d1fa3092b95168e4a88e2f8
runc:
  Version:      1.0.3
  GitCommit:    v1.0.3-0-gf46b6ba
docker-init:
  Version:      0.19.0
  GitCommit:    de40ad0

=====

To run Docker as a non-privileged user, consider setting up the
Docker daemon in rootless mode for your user:

    dockerd-rootless-setuptool.sh install

Visit https://docs.docker.com/go/rootless/ to learn about rootless mode.

To run the Docker daemon as a fully privileged service, but granting non-root
users access, refer to https://docs.docker.com/go/daemon-access/

WARNING: Access to the remote API on a privileged Docker daemon is equivalent
to root access on the host. Refer to the 'Docker daemon attack surface'
documentation for details: https://docs.docker.com/go/attack-surface/

=====

tayres@ubuntu:~/Desktop$
```

Installing Mayan EDMS

Pulling Docker Images

There are three images that we will need to prepare for the next step. These images consist of Mayan EDMS and its' dependencies Redis and Postgres. Let's install those by running these commands one at a time [ref. 2]:

```
sudo docker pull mayanedms/mayanedms:s4
```

```
sudo docker pull postgres:12.9-alpine
```

```
sudo docker pull redis:6.2-alpine
```

To verify that the installations of these images are in place, run

```
sudo docker images
```

The output should be similar to:

```
tayres@ubuntu:~/Desktop$ sudo docker images
REPOSITORY          TAG                 IMAGE ID            CREATED             SIZE
mayanedms/mayanedms s4                 bae9a3affba8       4 days ago         1.19GB
redis                6.2-alpine         6d12d0de5a46       6 days ago         32.4MB
postgres             12.9-alpine        71f18539f112       4 months ago       204MB
tayres@ubuntu:~/Desktop$
```

Our final step will be to start up all the containers! Run these (rather large) commands in order:

```
sudo docker run -d --name mayan-edms-postgres --restart=always -p 5432:5432 -e  
POSTGRES_USER=mayan -e POSTGRES_DB=mayan -e POSTGRES_PASSWORD=mayanuserpass -v  
/docker-volumes/mayan-edms/postgres:/var/lib/postgresql/data postgres:12.9-alpine
```

```
sudo docker run -d --name mayan-edms-redis --restart=always -p 6379:6379 -v /docker-  
volumes/mayan-edms/redis:/data redis:6.2-alpine redis-server --databases "3" --maxmemory-policy  
allkeys-lru --save "" --requirepass mayanredispassword
```

```
docker run -d --name mayan-edms --restart=always -p 80:8000 -e  
MAYAN_CELERY_BROKER_URL="redis://:mayanredispassword@172.17.0.1:6379/0" -e  
MAYAN_CELERY_RESULT_BACKEND="redis://:mayanredispassword@172.17.0.1:6379/1" -e  
MAYAN_DATABASES="{ 'default': { 'ENGINE': 'django.db.backends.postgresql', 'NAME': 'mayan', 'PASSWO  
RD': 'mayanuserpass', 'USER': 'mayan', 'HOST': '172.17.0.1' } }" -e  
MAYAN_LOCK_MANAGER_BACKEND="mayan.apps.lock_manager.backends.redis_lock.RedisLock" -e  
MAYAN_LOCK_MANAGER_BACKEND_ARGUMENTS="{ 'redis_url': 'redis://:mayanredispassword@172.  
17.0.1:6379/2' }" -v /docker-volumes/mayan-edms/media:/var/lib/mayan  
mayanedms/mayanedms:s4
```

To verify that all three containers are up and running, type the command:

```
sudo docker ps -a
```

Output should be similar to:

```
tayres@ubuntu:~/Desktop$ sudo docker ps -a
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
ed54d53524e3	mayanedms/mayanedms:s4	"entrypoint.sh run_a..."	8 seconds ago	Up 7 seconds	0.0.0.0:80->8000/tcp, :::80->8000/tcp	mayan-edms
70dd3cc1ec09	redis:6.2-alpine	"docker-entrypoint.s..."	29 seconds ago	Up 29 seconds	0.0.0.0:6379->6379/tcp, :::6379->6379/tcp	mayan-edms-redis
496dcd9a1b23	postgres:12.9-alpine	"docker-entrypoint.s..."	44 seconds ago	Up 44 seconds	0.0.0.0:5432->5432/tcp, :::5432->5432/tcp	mayan-edms-postgres

```
tayres@ubuntu:~/Desktop$
```

On the VM, open up Firefox and navigate to the URL <http://localhost> for the final result.

Automatic credentials

You have just finished installing Mayan EDMS, congratulations!

Login using the following credentials:

Username: admin
Email: autoadmin@example.com
Password: q8MCHrVcPh

Be sure to change the password to increase security and to disable this message.

Sign in

Username: (required)

Password: (required)

Submit

[Forgot your password?](#)

Write-Up

Document versioning [Ref. 3], of which the goal is to provide a complete history of document revisions. It is a useful feature, as there is always a chance of data corruption or human error that can occur when working with important documents. Having this system in place allows for quick roll-back to earlier versions of a document and is especially helpful when working within a potentially chaotic group setting, where any number of things could go wrong. I like to think of this feature as “in-house Git”.

Lossless image transformation is a helpful feature within Mayan EDMS which allows for manipulation of a file’s thumbnail. According to the Mayan documentation [ref. 4], the current supported features are rotation, zoom, crop, and resize. I can see this being useful within a corporate or team setting since the same transformation rules can be applied to multiple files at once. Imagine scanning in a set of documents and being able to flip them into portrait mode and zoom in on a common feature. Sounds like fast document parsing to me!

Document categorization [ref. 5] is an entire toolset dedicated to structuring files in many ways. Some of the baseline methods to organize documents would be via tags, “cabinets” (folders), and by type. This is very similar to how it works on most computer operating systems, so the people behind Mayan stepped it up a notch: they added a few automated systems called Indexing and Smart Links! These two in combination are great, as indexing allows for grouping via metadata and smart links allow for documents to be categorized together outside of typical hierarchy; they add references between separate files. Indexing is especially cool, as document hierarchy can be structured with rulesets based around the metadata of the documents themselves. As you could imagine, this would provide an extremely valuable way to structure a virtual team-workspace, having all files dumped in one location and then found via an interface that organizes them via specific ruleset.

References

- [1.] <https://linuxize.com/post/how-to-enable-ssh-on-ubuntu-20-04/>
- [2.] https://docs.mayan-edms.com/chapters/docker/install_simple.html
- [3.] <https://docs.mayan-edms.com/chapters/versioning.html>
- [4.] <https://docs.mayan-edms.com/chapters/transformations.html>
- [5.] <https://docs.mayan-edms.com/parts/categorization.html>