

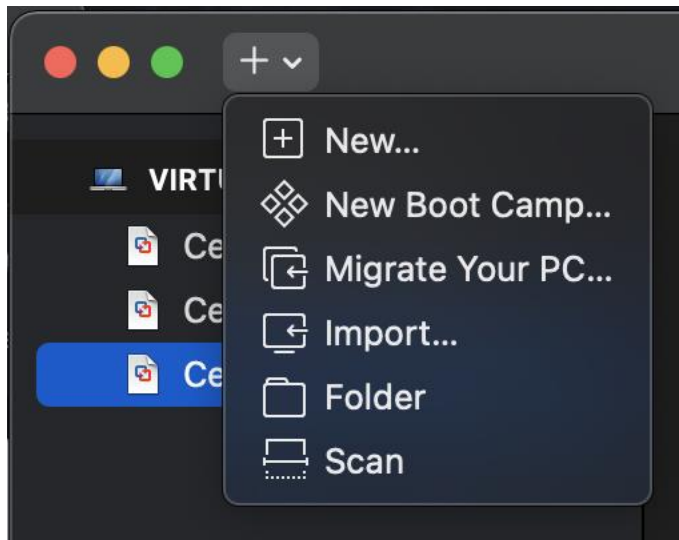
This document will provide step by step instructions on how to install Ubuntu Desktop 20.04, enable SSH, Docker and Mayan EDMS.

Prerequisites before starting this process:

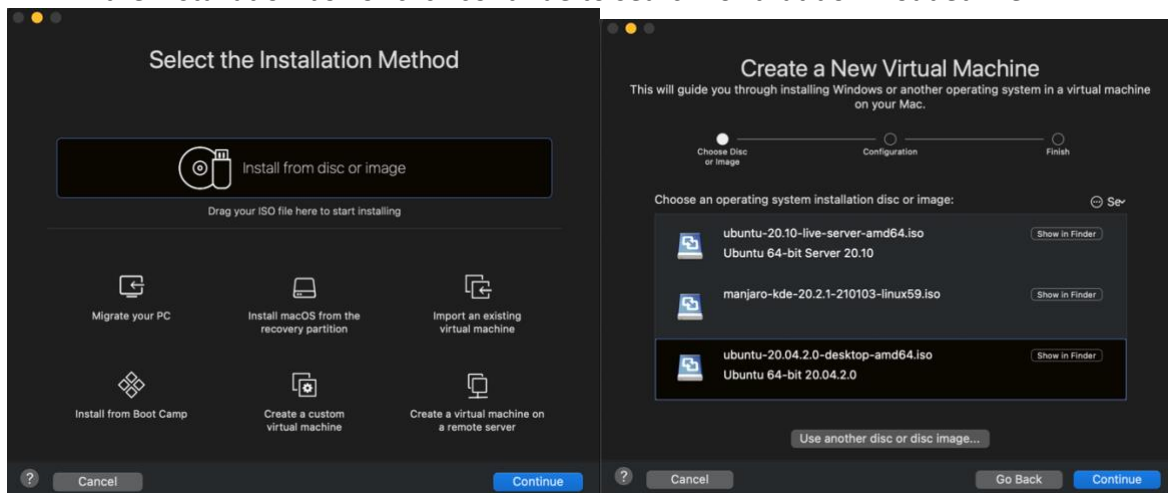
1. Must have a hypervisor software in order to install a Virtual Machine. (Ex. VMware Fusion)
2. Ubuntu Desktop ISO file which can be found at this [website](#). *An ISO file is a file that contains an image of data found on an optical disc, like a CD or DVD.

Let's Begin installing Ubuntu Desktop (I am using **VMware Fusion v12**).

1. Create a new Virtual Machine.



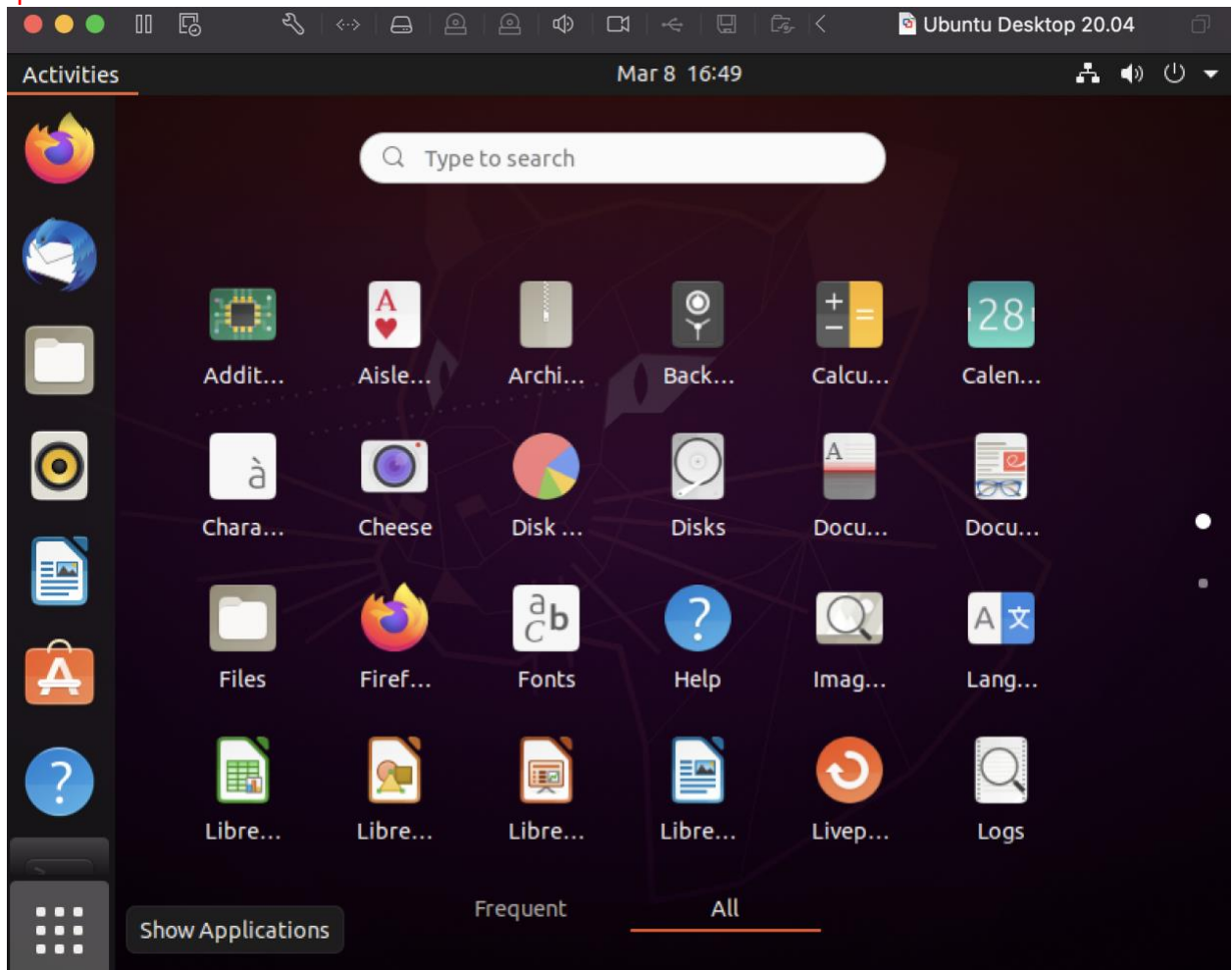
2. Once your Ubuntu Desktop ISO file is downloaded you can drag and drop the file into the installation box or click continue to search for that downloaded file.



3. Once that file is selected click on continue and now you should be at the **Linux Easy Install** screen. You will want to set up a password for the admin. Click Continue.
4. Next, should be the finish screen and to rename your Virtual Machine click on customized settings. Click Finish. All DONE!

Enabling SSH on Ubuntu

1. Click menu dots at the bottom left of the screen to show applications. Search for 'Terminal' and open it up. ***Terminal is a command line tool where it makes it easier and quicker to install software.**



2. Type command – **sudo apt update** (This updates any packages/software on your Ubuntu system) **You will need to enter your admin password that was created earlier in the document.*

```
damarkusharris@ubuntu:~$ sudo apt update
[sudo] password for damarkusharris:
Hit:1 http://us.archive.ubuntu.com/ubuntu focal InRelease
Hit:2 http://security.ubuntu.com/ubuntu focal-security InRelease
Hit:3 http://us.archive.ubuntu.com/ubuntu focal-updates InRelease
Hit:4 http://us.archive.ubuntu.com/ubuntu focal-backports InRelease
Reading package lists... Done
Building dependency tree
Reading state information... Done
53 packages can be upgraded. Run 'apt list --upgradable' to see them.
```

3. Now let's install openssh by using command – **sudo apt install openssh-server**

```
damarkusharris@ubuntu:~$ sudo apt install openssh-server
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  ncurses-term openssh-sftp-server ssh-import-id
Suggested packages:
  molly-guard monkeysphere ssh_askpass
The following NEW packages will be installed:
  ncurses-term openssh-server openssh-sftp-server ssh-import-id
0 upgraded, 4 newly installed, 0 to remove and 53 not upgraded.
Need to get 688 kB of archives.
After this operation, 6,010 kB of additional disk space will be used.
Do you want to continue? [Y/n] y
```

4. Press y for yes if prompted to continue through the installation. Once the installation is complete, the SSH service will start automatically. To verify enter command – **sudo systemctl status ssh**

```
damarkusharris@ubuntu:~$ sudo systemctl status ssh
● ssh.service - OpenBSD Secure Shell server
   Loaded: loaded (/lib/systemd/system/ssh.service; enabled; vendor preset: ☐)
   Active: active (running) since Mon 2021-03-08 16:46:34 PST; 3min 35s ago
     Docs: man:sshd(8)
           man:sshd_config(5)
    Main PID: 5147 (sshd)
      Tasks: 1 (limit: 4619)
     Memory: 1.3M
    CGroup: /system.slice/ssh.service
           └─5147 sshd: /usr/sbin/sshd -D [listener] 0 of 10-100 startups

Mar 08 16:46:34 ubuntu systemd[1]: Starting OpenBSD Secure Shell server...
Mar 08 16:46:34 ubuntu sshd[5147]: Server listening on 0.0.0.0 port 22.
Mar 08 16:46:34 ubuntu sshd[5147]: Server listening on :: port 22.
Mar 08 16:46:34 ubuntu systemd[1]: Started OpenBSD Secure Shell server.
```

5. Last step is to make sure to open the SSH port since Ubuntu ships with a firewall configuration tool called UFW. Use command – **sudo ufw allow ssh**

```
damarkusharris@ubuntu:~$ sudo ufw allow ssh
[sudo] password for damarkusharris:
Rules updated
Rules updated (v6)
damarkusharris@ubuntu:~$
```

Installing Docker

Docker uses a technology call container where you are able to install software by creating images and shipping them to other devices if needed without having to install dependencies on your local machine or another machine to use the container.

***Note to ssh into this machine to copy long commands and execute them quicker open of your terminal (MAC) or CMD (Windows) and type 'ssh username@ip-address' replace username with your username and IP address with your actual IP address of your Ubuntu system. Find your IP address by using command – ip r on your Ubuntu and after link src is your IP address. Type yes to continue connecting and then type in your password.**

```
damarkusharris@ubuntu:~$ ip r
default via 10.10.5.1 dev ens33 proto dhcp metric 100
10.10.5.0/24 dev ens33 proto kernel scope link src 10.10.5.4 metric 100
169.254.0.0/16 dev ens33 scope link metric 1000
[damarkusharris@Damarkuss-MacBook-Pro ~ % ssh damarkusharris@10.10.5.4
The authenticity of host '10.10.5.4 (10.10.5.4)' can't be established.
ECDSA key fingerprint is SHA256:Q0fmVIt0jNXyby2yIFfZlUmlJjGXJIjzXTcRtLxuAaI.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '10.10.5.4' (ECDSA) to the list of known hosts.
damarkusharris@10.10.5.4's password:
Welcome to Ubuntu 20.04.2 LTS (GNU/Linux 5.8.0-44-generic x86_64)
```

1. First, we need to remove any previous version of Docker we may have. Use command – **sudo apt-get remove docker docker-engine docker.io containerd runc**
2. We will set up a Docker repository and since we already updated are system earlier, we can skip that step and get right to setting up are repository. Use command in screenshot below. ***Do not use the '\$' also you can type out command fully in one big line or use the backslash like in the picture just hit enter after the backslash and it will go to a new line with a greater than sign as the indicator '>' and type 'y' to continue through installation.**


```
$ sudo apt-get install \
  apt-transport-https \
  ca-certificates \
  curl \
  gnupg
```

3. Add Docker's official GPG key using command –

```
curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo gpg --dearmor -o
/usr/share/keyrings/docker-archive-keyring.gpg
```

4. Now to install docker use command – **sudo apt install docker.io**
5. Check for docker version using command – **docker --version**
6. You will need to start and enable docker at startup by using 2 commands – **sudo systemctl start docker** and **sudo systemctl enable docker**.
 *If start command for docker is successful it will not output any information use status command below to make sure it worked!
7. To check status use command - **sudo systemctl status docker**

```
[damarkusharris@ubuntu:~]$ sudo systemctl start docker
[damarkusharris@ubuntu:~]$ sudo systemctl enable docker
Created symlink /etc/systemd/system/multi-user.target.wants/docker.service → /li
b/systemd/system/docker.service.
[damarkusharris@ubuntu:~]$ sudo systemctl status docker
● docker.service - Docker Application Container Engine
   Loaded: loaded (/lib/systemd/system/docker.service; enabled; vendor preset
   Active: active (running) since Mon 2021-03-08 18:08:05 PST; 23s ago
   TriggeredBy: ● docker.socket
     Docs: https://docs.docker.com
    Main PID: 11189 (dockerd)
      Tasks: 10
     Memory: 36.8M
    CGroup: /system.slice/docker.service
            └─11189 /usr/bin/dockerd -H fd:// --containerd=/run/containerd/con>
```

Installing Mayan EDMS

Make sure Docker is installed and active (running) and use **sudo** in front of commands if permission is denied. Mayan is a web-based free document management system for managing documents within an organization.

1. Download the Mayan EDMS Docker image using command:

```
docker pull mayanedms/mayanedms:3.5.5
```

2. Download the PostgreSQL Docker image using command:

```
docker pull postgres:9.6-alpine
```

3. Download the Redis Docker image using command:

```
docker pull redis:5.0-alpine
```

4. Create and run a PostgreSQL container using command:

```
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:9.6-alpine
```

5. Create and run a Redis container using command:

```
docker run \
-d \
--name mayan-edms-redis \
--restart=always \
-p 6379:6379 \
-v /docker-volumes/mayan-edms/redis:/data \
redis:5.0-alpine \
redis-server \
--databases \
"2" \
--maxmemory-policy \
allkeys-lru \
--save \
"" \
--requirepass mayanredispassword
```

6. Create and run a Mayan EDMS container using command:

```
docker run \
-d \
--name mayan-edms \
--restart=always \
-p 80:8000 \
-e
MAYAN_DATABASES="{\"default\":{\"ENGINE\":\"django.db.backends.postgresql\",\"NAME\":\"mayan\",\"PASSWORD\":\"mayanuserpass\",\"USER\":\"mayan\",\"HOST\":\"172.17.0.1\"}}\" \
-e MAYAN_CELERY_BROKER_URL=\"redis://:mayanredispasword@172.17.0.1:6379/0\" \
-e MAYAN_CELERY_RESULT_BACKEND=\"redis://:mayanredispasword@172.17.0.1:6379/1\" \
-v /docker-volumes/mayan-edms/media:/var/lib/mayan \
mayanedms/mayanedms:3.5.5
```

7. All done with the installations. Next open up a browser and go to your Ubuntu Desktop IP address. You should see this screen. Sign in with the information provided

Automatic credentials

You have just finished installing **Mayan EDMS**, congratulations!

Login using the following credentials:

Username: **admin**
Email: **autoadmin@example.com**
Password: **b8akpLcyFZ**

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

Sign in

Username: (required)

Password: (required)

☐ Remember me

→ Sign in

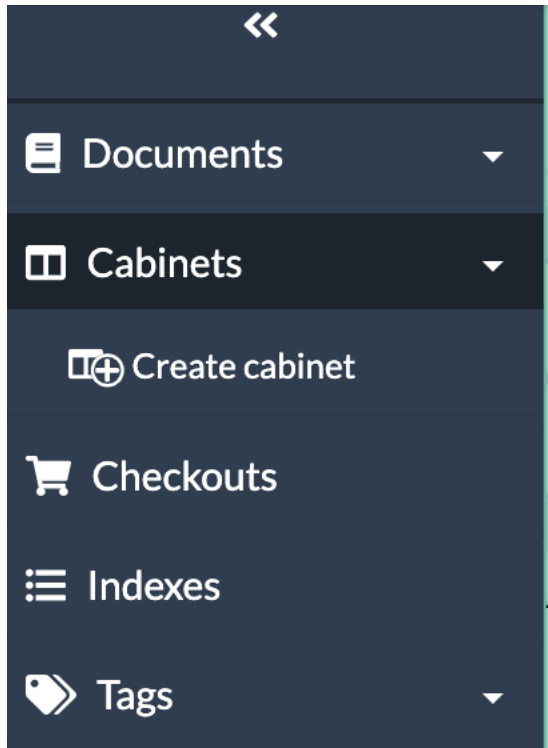
Forgot your password?

Write Up Section

Explore Mayan EDMS: Utilize internet resources to research key functions within the software. Select three (3) notable features, describe their function, and explain how these features would be configured and used in a real-world environment.

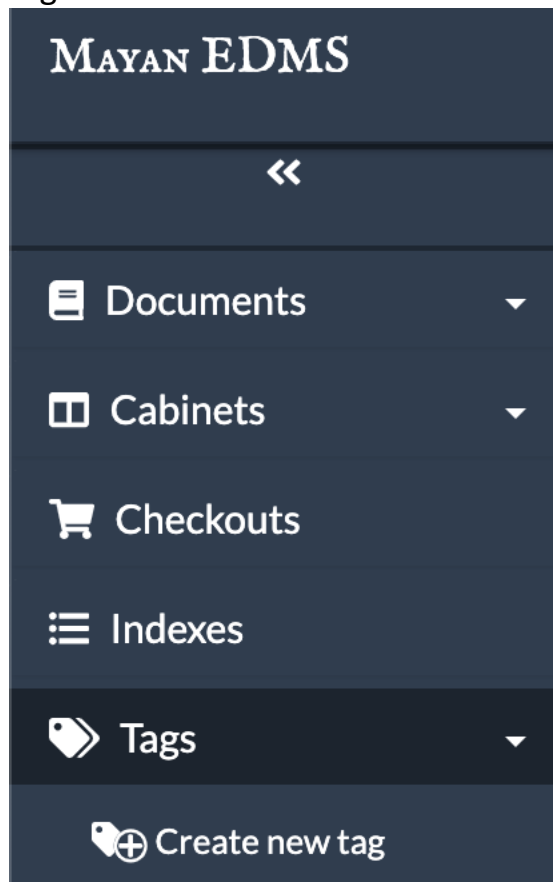
1. **Cabinets** are a multi-level method to organize documents. Each cabinet can contain documents as well as other sub level cabinets. Cabinets can be configured by selecting the menu to the top left and selecting Cabinets and select Create cabinet. Cabinets can be used in a real-world

environment by allowing businesses to set up a hierarchy on their documents.



2. **Tags** are color coded properties that can be attached or removed from documents. Tags allow giving documents a binary property. Also, you can use more than one tag on a document. You can add a tag by click on the menu in the top left and selecting tags and Create new tag. Tags can be used in a real-world environment by allowing searching of documents easier and to show all documents tagged with a particular

tag via the Documents link of each tag.



3. **Smart links** are a way to link documents without changing how they are organized in their respective indexes. Configuration is automatic and are rule based. Smart links just show the documents that match the rules as evaluated against the metadata or properties of the currently displayed document. Smart links are useful when two documents are related somehow but are of different type or different hierarchical units.