

Humberto Gonzalez

February 10, 2024

# Docker and Mayan EDMS Installation on Ubuntu 22.04

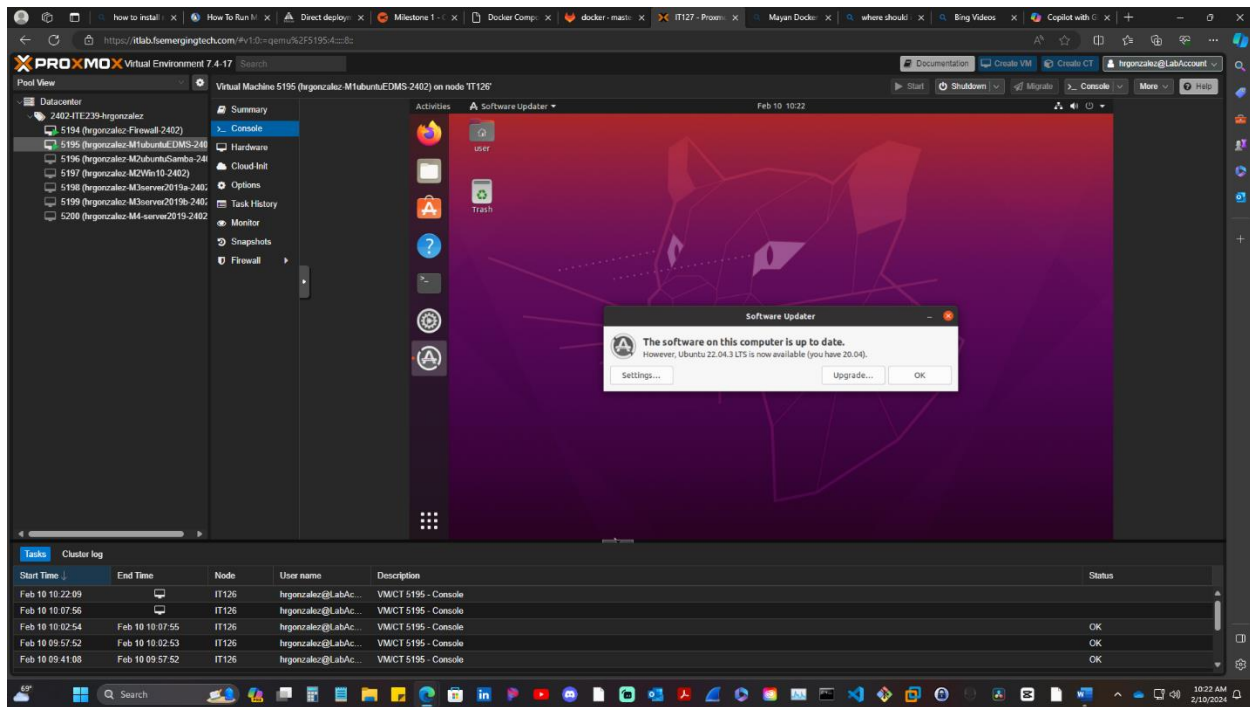
## Table of Contents

Docker and Docker compose Prerequisite .....	3
Software up to date .....	3
Remove old versions of Docker .....	3
Allow apt to use repo over HTTPS.....	5
Adding Docker's official GPG key .....	5
Stable repository.....	6
Installing docker engine with docker compose.....	6
Check if Docker is working correctly .....	7
Check version of Docker installed .....	8
Check version of docker compose .....	8
Mayan Installation using Docker compose .....	9
Mayan files needed for the installation .....	9
Navigate to docker directory.....	9
Launch Mayan EDMS docker containers.....	10
Check status of the running containers .....	11
Verification of Mayan EDMS Installation.....	12
Mayan Log In .....	12
Change Mayan EDMS logging password .....	13
Mayan EDMS Notable Features Write Up.....	14
Document Versioning.....	14
Document Configuration.....	14
Document Versioning in real world scenario .....	14
Workflow Automation.....	14
Configuration of Workflow Automation .....	15
Workflow Automation in real world scenario .....	15
Color coded tags .....	15
Configuration of color coded tags.....	15
Color coded tags in real world scenario.....	15
References.....	16

## Docker and Docker compose Prerequisite

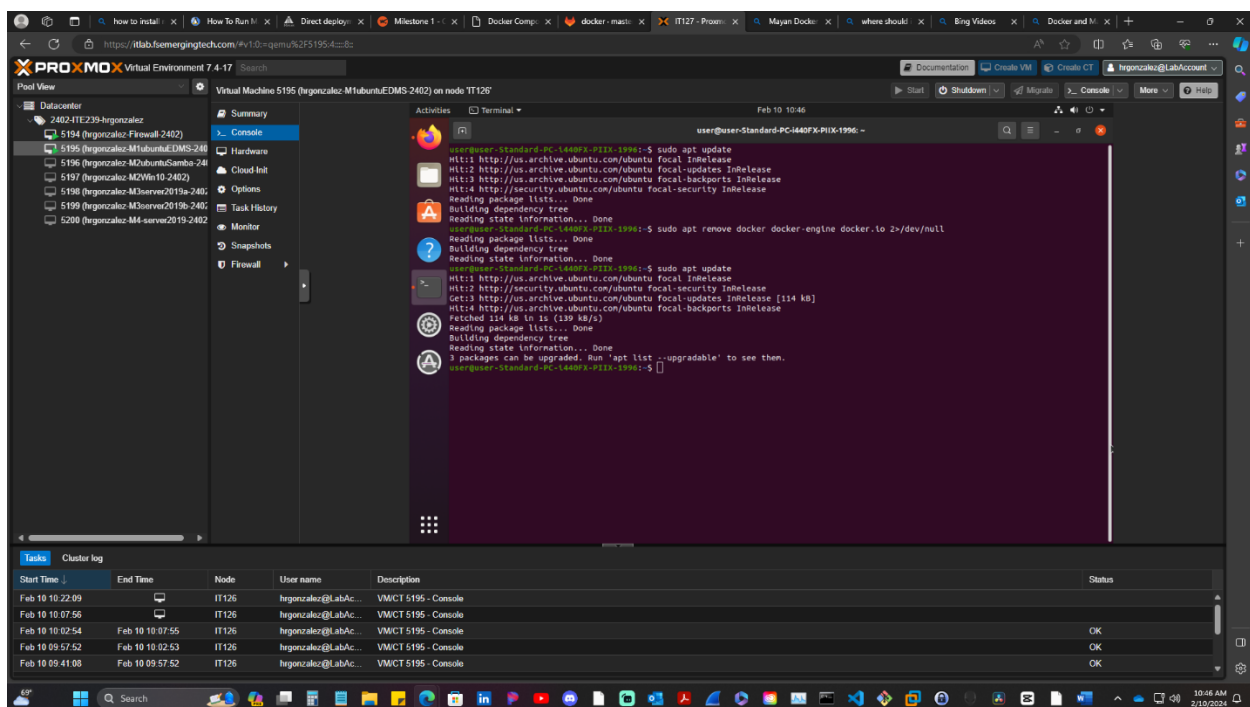
### Software up to date

Make sure OS system is up to date. Do this by going to `> show applications (9 dot image) > software updater > update` if necessary.



### Remove old versions of Docker

First, we need to update the package list by using command `> sudo apt update` \*If prompted for password use your created password\*



Let's remove any old version of docker. Use command `> sudo apt remove docker docker-engine docker.io 2>/dev/null`

The screenshot shows a Proxmox VE terminal window with the following commands and output:

```

user@user-Standard-PC-1440FX-P1XX-19961:~$ sudo apt update
Hit:1 http://us.archive.ubuntu.com/ubuntu focal InRelease
Hit:2 http://us.archive.ubuntu.com/ubuntu focal-updates InRelease
Hit:3 http://us.archive.ubuntu.com/ubuntu focal-backports InRelease
Hit:4 http://security.ubuntu.com/ubuntu focal-security InRelease
Reading package lists... Done
Building dependency tree
Reading state information... Done
user@user-Standard-PC-1440FX-P1XX-19961:~$ sudo apt remove docker docker-engine docker.io 2>/dev/null
Reading package lists... Done
Building dependency tree
Reading state information... Done
3 packages can be upgraded. Run 'apt list --upgradable' to see them.
user@user-Standard-PC-1440FX-P1XX-19961:~$

```

The terminal output shows the successful removal of Docker and the subsequent system update. The bottom of the screenshot shows a table of tasks:

Start Time	End Time	Node	User name	Description	Status
Feb 10 10:22:09		IT126	hegencales@LabAc	VMCT 5195 - Console	
Feb 10 10:07:56		IT126	hegencales@LabAc	VMCT 5195 - Console	
Feb 10 10:02:54	Feb 10 10:07:55	IT126	hegencales@LabAc	VMCT 5195 - Console	OK
Feb 10 09:57:52	Feb 10 10:02:53	IT126	hegencales@LabAc	VMCT 5195 - Console	OK
Feb 10 09:41:05	Feb 10 09:57:52	IT126	hegencales@LabAc	VMCT 5195 - Console	OK

Let's update the package list again by using command `> sudo apt update`

The screenshot shows a Proxmox VE terminal window with the following commands and output:

```

user@user-Standard-PC-1440FX-P1XX-19961:~$ sudo apt update
Hit:1 http://us.archive.ubuntu.com/ubuntu focal InRelease
Hit:2 http://us.archive.ubuntu.com/ubuntu focal-updates InRelease
Hit:3 http://us.archive.ubuntu.com/ubuntu focal-backports InRelease
Hit:4 http://security.ubuntu.com/ubuntu focal-security InRelease
Reading package lists... Done
Building dependency tree
Reading state information... Done
user@user-Standard-PC-1440FX-P1XX-19961:~$

```

The terminal output shows the successful update of the package list. The bottom of the screenshot shows a table of tasks:

Start Time	End Time	Node	User name	Description	Status
Feb 10 10:22:09		IT126	hegencales@LabAc	VMCT 5195 - Console	
Feb 10 10:07:56		IT126	hegencales@LabAc	VMCT 5195 - Console	
Feb 10 10:02:54	Feb 10 10:07:55	IT126	hegencales@LabAc	VMCT 5195 - Console	OK
Feb 10 09:57:52	Feb 10 10:02:53	IT126	hegencales@LabAc	VMCT 5195 - Console	OK
Feb 10 09:41:05	Feb 10 09:57:52	IT126	hegencales@LabAc	VMCT 5195 - Console	OK

## Allow apt to use repo over HTTPS

Next, we need to allow apt to use repositories over HTTPS. Use command `> sudo apt -y install lsb-release gnupg apt-transport-https ca-certificates curl software-properties-common`

```

user@user-Standard-PC-1440FX-PIIX-1996:~$ sudo apt -y install lsb-release gnupg apt-transport-https ca-certificates curl software-properties-common
Reading package lists... Done
Building dependency tree
Reading state information... Done
lsb-release is already the newest version (11.1ubuntu2).
ca-certificates is already the newest version (20230311ubuntu2.04.1).
gnupg is already the newest version (2.2.39-3ubuntu2.2).
software-properties-common is already the newest version (0.99.9.12).
The following packages were automatically installed and are no longer required:
  gpgv
Use 'sudo apt autoremove' to remove them.
The following additional packages will be installed:
  libcurl4
The following NEW packages will be installed:
  apt-transport-https curl libcurl4
0 upgraded, 3 newly installed, 0 to remove and 3 not upgraded.
Need to get 398 kB of archives:
  After this operation, 1,288 kB of additional disk space will be used.
Get:1 http://us.archive.ubuntu.com/ubuntu focal-updates/universe amd64 apt-transport-https all 2.0.10 [1,704 B]
Get:2 http://us.archive.ubuntu.com/ubuntu focal-updates/main amd64 libcurl4 amd64 7.68.0-1ubuntu2.21 [235 kB]
Get:3 http://us.archive.ubuntu.com/ubuntu focal-updates/main amd64 curl amd64 7.68.0-1ubuntu2.21 [165 kB]
Fetched 398 kB in 0s (1,864 kB/s)
Selecting previously unselected package apt-transport-https.
(Reading database ... 157171 files and directories currently installed.)
Preparing to unpack .../apt-transport-https_2.0.10_all.deb ...
Unpacking apt-transport-https (2.0.10) ...
Selecting previously unselected package libcurl4:amd64.
Preparing to unpack .../libcurl4_7.68.0-1ubuntu2.21_amd64.deb ...
Unpacking libcurl4:amd64 (7.68.0-1ubuntu2.21) ...
Selecting previously unselected package curl.
Preparing to unpack .../curl_7.68.0-1ubuntu2.21_amd64.deb ...
Unpacking curl (7.68.0-1ubuntu2.21) ...
Setting up apt-transport-https (2.0.10) ...
Setting up libcurl4:amd64 (7.68.0-1ubuntu2.21) ...
Setting up curl (7.68.0-1ubuntu2.21) ...

```

## Adding Docker's official GPG key

Now we need to add Docker's GPG key. Use command `> curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo gpg --dearmor -o /etc/apt/trusted.gpg.d/docker.gpg`

```

user@user-Standard-PC-1440FX-PIIX-1996:~$ curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo gpg --dearmor -o /etc/apt/trusted.gpg.d/docker.gpg
user@user-Standard-PC-1440FX-PIIX-1996:~$ sudo apt-add-repository "deb [arch=$(dpkg --print-architecture)] https://download.docker.com/linux/ubuntu $(lsb_release -cs) stable"
Hit:1 http://us.archive.ubuntu.com/ubuntu focal InRelease
Hit:2 http://us.archive.ubuntu.com/ubuntu focal-updates InRelease
Hit:3 http://security.ubuntu.com/ubuntu focal-security InRelease
Hit:4 http://us.archive.ubuntu.com/ubuntu focal-backports InRelease
Get:5 https://download.docker.com/linux/ubuntu focal InRelease [57.7 kB]
Get:6 https://download.docker.com/linux/ubuntu focal/stable amd64 Packages [37.8 kB]
Fetched 95.5 kB in 1s (203 kB/s)
Reading package lists... Done

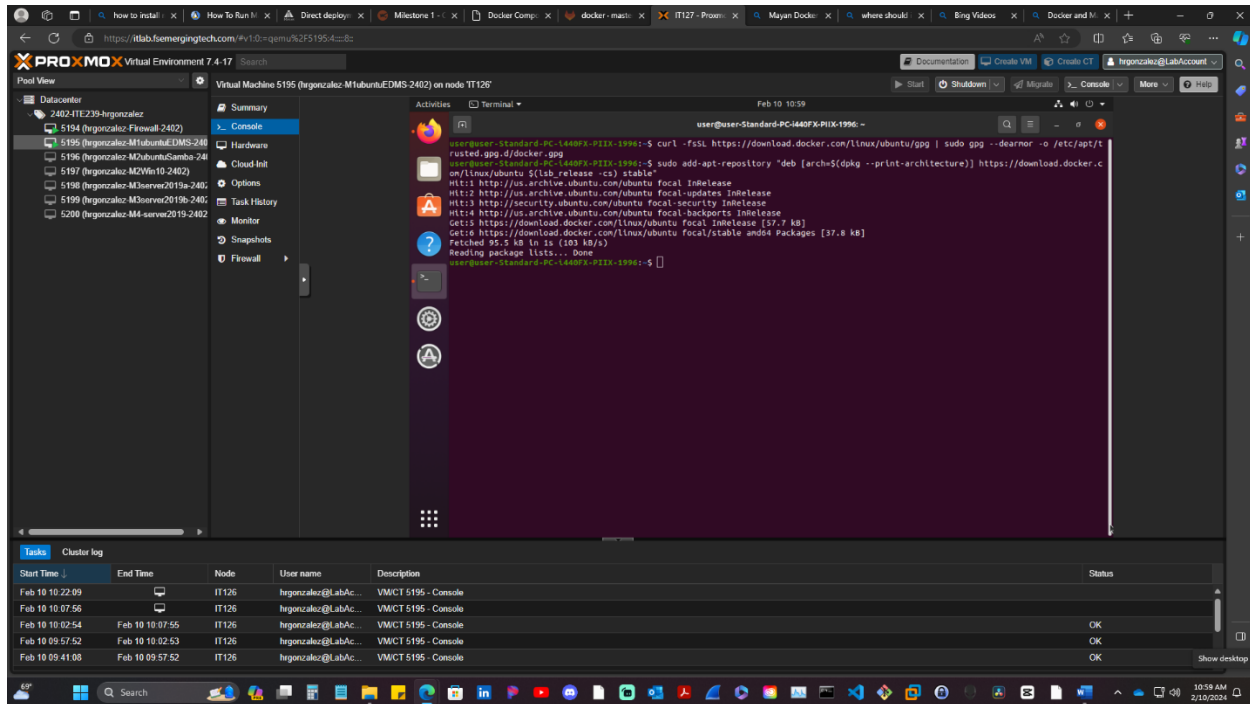
```



After command it should automatically take you back to your command line.

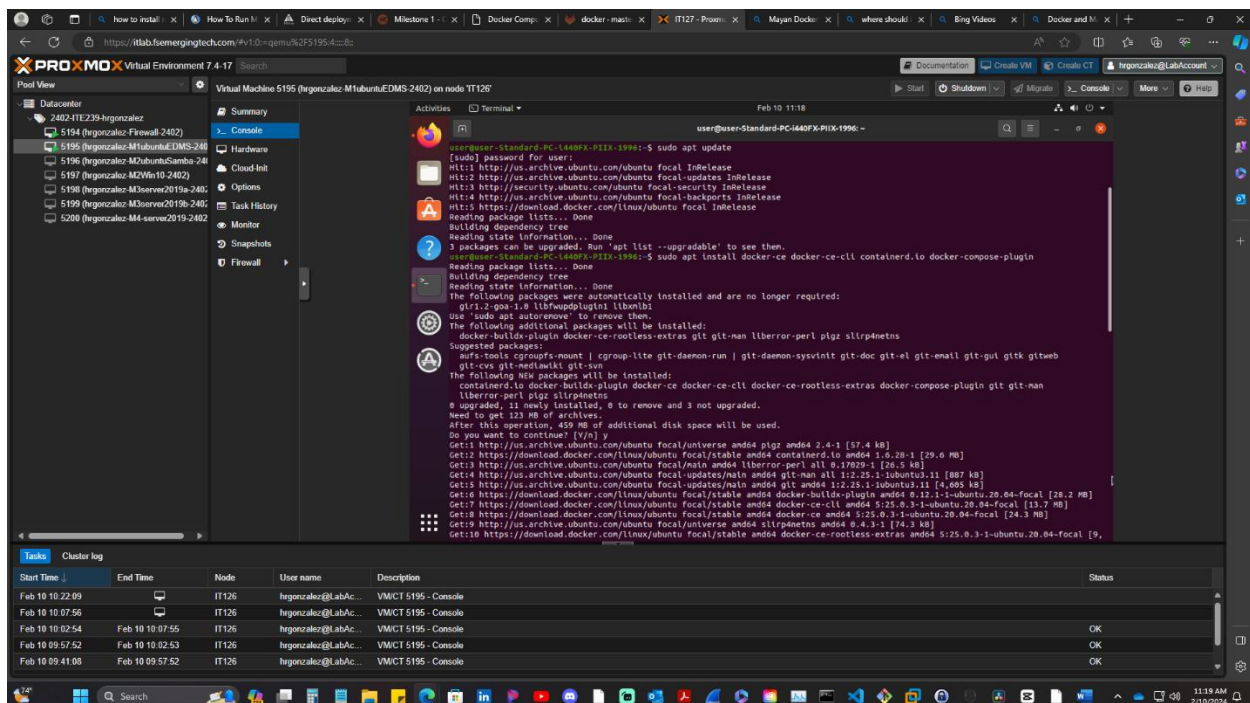
## Stable repository

Let's add a stable repository. Use command `> sudo add-apt-repository "deb [arch=$(dpkg --print-architecture)] https://download.docker.com/linux/ubuntu $(lsb_release -cs) stable"`



## Installing docker engine with docker compose

Use command `> sudo apt update`



Followed by command `> sudo apt install docker-ce docker-ce-cli containerd.io docker-compose-plugin` > when prompted press y to continue with installation.

```

user@user-Standard-PC-1440FX-PIIX-1996:~$ sudo apt update
[sudo] password for user:
Hit:1 http://us.archive.ubuntu.com/ubuntu focal InRelease
Hit:2 http://us.archive.ubuntu.com/ubuntu focal-updates InRelease
Hit:3 http://security.ubuntu.com/ubuntu focal-security InRelease
Hit:4 http://us.archive.ubuntu.com/ubuntu focal-backports InRelease
Hit:5 https://download.docker.com/linux/ubuntu focal InRelease
Reading package lists... Done
Building dependency tree
Reading state information... Done
3 packages can be upgraded. Run 'apt list --upgradable' to see them.
user@user-Standard-PC-1440FX-PIIX-1996:~$ sudo apt install docker-ce docker-ce-cli containerd.io docker-compose-plugin
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following packages were automatically installed and are no longer required:
  git-2-gpg-1.8 libhttpd-authn1 libnss1
Use 'sudo apt autoremove' to remove them.
The following additional packages will be installed:
  docker-buildx-plugin docker-ce-rootless-extras git git-man liberror-perl pigz slurp4mtns
Suggested packages:
  aufs-tools cgroupfs-mount | cgroup-lite git-daemon-run | git-daemon-symlinks git-doc git-el git-email git-gui gitk gitweb
  git-cvs git-mediawiki git-svn
The following NEW packages will be installed:
  containerd.io docker-buildx-plugin docker-ce docker-ce-cli docker-ce-rootless-extras docker-compose-plugin git git-man
  liberror-perl pigz slurp4mtns
0 upgraded, 11 newly installed, 0 to remove and 3 not upgraded.
Need to get 123 MB of archives.
After this operation, 459 MB of additional disk space will be used.
Do you want to continue? [y/n] y
Get:1 http://us.archive.ubuntu.com/ubuntu focal/universe amd64 pigz amd64 2.4.3 [57.4 kB]
Get:2 https://download.docker.com/linux/ubuntu focal/stable amd64 containerd.io amd64 1.0.28-1 [29.6 MB]
Get:3 http://us.archive.ubuntu.com/ubuntu focal/main amd64 liberror-perl all 0.9.2029-1 [26.3 kB]
Get:4 http://us.archive.ubuntu.com/ubuntu focal-updates/main amd64 git-man all 1:2.25.1-1ubuntu3.11 [887 kB]
Get:5 https://download.docker.com/linux/ubuntu focal/stable amd64 git amd64 1:2.25.1-1ubuntu3.11 [4,485 kB]
Get:6 https://download.docker.com/linux/ubuntu focal/stable amd64 docker-buildx-plugin amd64 0.12.1-1-ubuntu.20.04-focal [28.2 MB]
Get:7 https://download.docker.com/linux/ubuntu focal/stable amd64 docker-ce-cli amd64 5:25.0.3-1-ubuntu.20.04-focal [13.7 MB]
Get:8 https://download.docker.com/linux/ubuntu focal/stable amd64 docker-ce amd64 5:25.0.3-1-ubuntu.20.04-focal [24.3 MB]
Get:9 http://us.archive.ubuntu.com/ubuntu focal/universe amd64 slurp4mtns amd64 0.4.3-1 [74.3 kB]
Get:10 https://download.docker.com/linux/ubuntu focal/stable amd64 docker-ce-rootless-extras amd64 5:25.0.3-1-ubuntu.20.04-focal [9,
  
```

These commands will install docker engine with docker compose as a plug in. **\*If prompted for password use your created password\***

Check if Docker is working correctly

Use command line `> sudo docker run hello-world`

```

user@user-Standard-PC-1440FX-PIIX-1996:~$ sudo docker run hello-world
Unable to find image 'hello-world:latest' locally
latest: Pulling from library/hello-world
c1efk8e544: Pull complete
Digest: sha256:4bd7811b6914a99dc56de6a2beab577f655a4a4b5c50bc5491968bcb2e6
Status: Downloaded newer image for hello-world:latest

Hello from Docker!
This message shows that your installation appears to be working correctly.

To generate this message, Docker took the following steps:
1. The Docker client contacted the Docker daemon.
2. The Docker daemon pulled the "hello-world" image from the Docker Hub.
   (amd64)
3. The Docker daemon created a new container from that image which runs the
   executable that produces the output you are currently reading.
4. The Docker daemon streamed that output to the Docker client, which sent it
   to your terminal.

To try something more ambitious, you can run an Ubuntu container with:
$ docker run -it ubuntu bash

Share images, automate workflows, and more with a free Docker ID:
https://hub.docker.com/

For more examples and ideas, visit:
https://docs.docker.com/get-started/

user@user-Standard-PC-1440FX-PIIX-1996:~$
  
```

If docker is working correctly you should be able to see a reply saying hello-world with confirmation that docker was installed successfully.

Check version of Docker installed

Use command line > `sudo docker version`

The screenshot shows the Proxmox VE interface with a terminal window open. The terminal displays the output of the command `sudo docker version`. The output shows the Docker Engine version (25.0.3) and the Docker Client version (25.0.3). The Docker Engine version is 25.0.3, API version is 1.44, Go version is go1.21.6, Git commit is 4d6f435, Built: Tue Feb 6 21:14:17 2024, OS/Arch: linux/amd64, Experimental: false. The Docker Client version is 25.0.3, API version is 1.44 (minimum version 1.24), Go version is go1.21.6, Git commit is 4d6f435, Built: Tue Feb 6 21:14:17 2024, OS/Arch: linux/amd64, Experimental: false. The containerd version is 1.6.28, GitCommit: aad7eda36d25f8a1b98dfb587313b99c0190bb, Version: 1.1.12, GitCommit: v1.1.12-0-g51de94, docker-init version is 0.19.0, GitCommit: de40ad0.

Start Time	End Time	Node	User name	Description	Status
Feb 10 11:30:02		IT126	hgonzalez@labAc...	VMCT 5195 - Console	
Feb 10 11:30:02		IT126	hgonzalez@labAc...	VMCT 5195 - Console	
Feb 10 22:09	Feb 10 11:30:02	IT126	hgonzalez@labAc...	VMCT 5195 - Console	OK
Feb 10 10:56	Feb 10 11:30:02	IT126	hgonzalez@labAc...	VMCT 5195 - Console	OK
Feb 10 10:254	Feb 10 10:56	IT126	hgonzalez@labAc...	VMCT 5195 - Console	OK

Check version of docker compose

Use command line > `docker compose version`

The screenshot shows the Proxmox VE interface with a terminal window open. The terminal displays the output of the command `docker compose version`. The output shows the Docker Compose version v2.24.5.

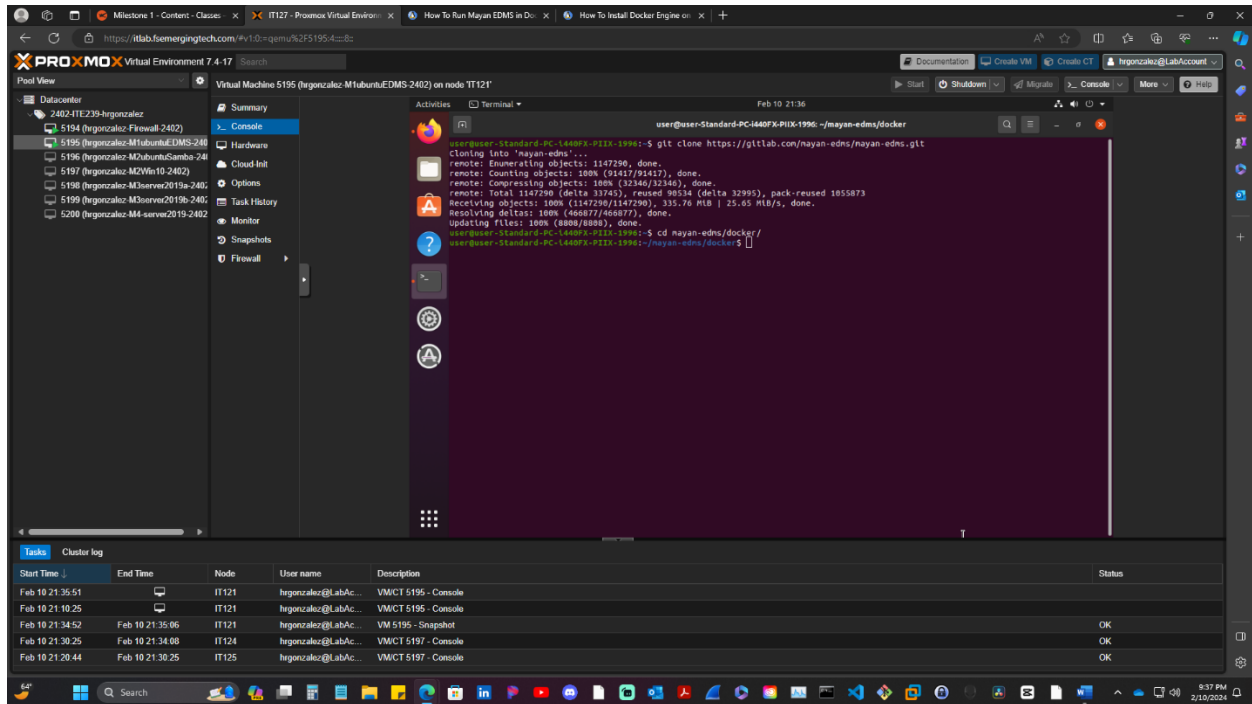
Start Time	End Time	Node	User name	Description	Status
Feb 10 11:30:02		IT126	hgonzalez@labAc...	VMCT 5195 - Console	
Feb 10 11:30:02		IT126	hgonzalez@labAc...	VMCT 5195 - Console	
Feb 10 22:09	Feb 10 11:30:02	IT126	hgonzalez@labAc...	VMCT 5195 - Console	OK
Feb 10 10:56	Feb 10 11:30:02	IT126	hgonzalez@labAc...	VMCT 5195 - Console	OK
Feb 10 10:254	Feb 10 10:56	IT126	hgonzalez@labAc...	VMCT 5195 - Console	OK



## Mayan Installation using Docker compose

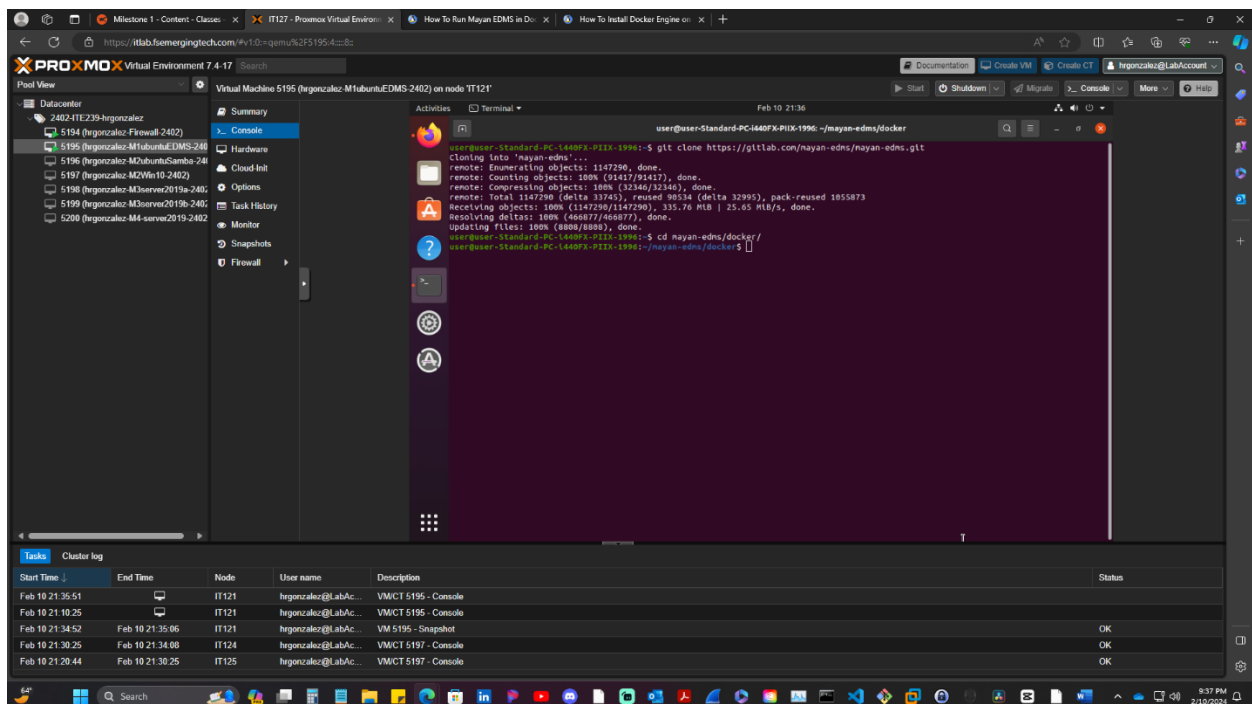
### Mayan files needed for the installation

In terminal use command line > `git clone https://gitlab.com/mayan-edms/mayan-edms.git`



Navigate to docker directory (\*this directory and it's subdirectory are included and created when we imported the files from gitlab\*).

Use command line > `cd mayan-edms/docker/`



In this directory we can pull up our .env file and make changes from updating compose project name or updating variables such as passwords for DB and users. For the sake of keeping assignment straightforward and simple this step is going to be skipped.

## Launch Mayan EDMS docker containers

To run Mayan EDMS docker containers (while still being inside the directory from previous step) we are going to use command line > `docker compose up -d`

The screenshot shows the Proxmox Virtual Environment (VE) interface. The main window displays the terminal output of the command `docker compose up -d` executed in the directory `~/mayan-edms/docker`. The output shows the pulling of various Docker images for the Mayan EDMS application stack, including postgresql, rabbitmq, and redis. The Proxmox interface also shows a list of tasks at the bottom.

Start Time	End Time	Node	User name	Description	Status
Feb 10 21:46:11		IT121	hgonzalez@LabAc...	VMCT 5195 - Console	
Feb 10 21:10:25		IT121	hgonzalez@LabAc...	VMCT 5195 - Console	
Feb 10 21:45:55		IT124	hgonzalez@LabAc...	VMCT 5197 - Console	
Feb 10 21:45:54	Feb 10 21:46:11	IT124	hgonzalez@LabAc...	VMCT 5197 - Console	OK
Feb 10 21:45:57	Feb 10 21:45:53	IT121	hgonzalez@LabAc...	VMCT 5195 - Console	OK

Check status of the running containers

Use command line > `docker ps`

Another alternative way to check status is by using command line > `docker compose ps`

The screenshot shows the Proxmox VE interface with a terminal window open for VM 5196. The terminal displays the output of the `docker ps` command, showing a list of containers and their status. Below the terminal output, a table summarizes the container status.

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS
9e5a8f86803	mayanedms/mayanedms:54.6	"/usr/local/bin/entr..."	8 seconds ago	Up 3 seconds	0.0.0.0:80->8000/tcp, ::
b03c41754e29	postgres:13.13-alpine	mayan-app-1	8 seconds ago	Up 3 seconds	5432/tcp
cc0f67dfc701	redis:7.0.15-alpine	mayan-postgresql-1	8 seconds ago	Up 3 seconds	6379/tcp
6fc3fc99f35	rabbitmq:3.12.12-management-alpine	mayan-rabbitmq-1	8 seconds ago	Up 3 seconds	4369/tcp, 5671-5672/tcp, ::

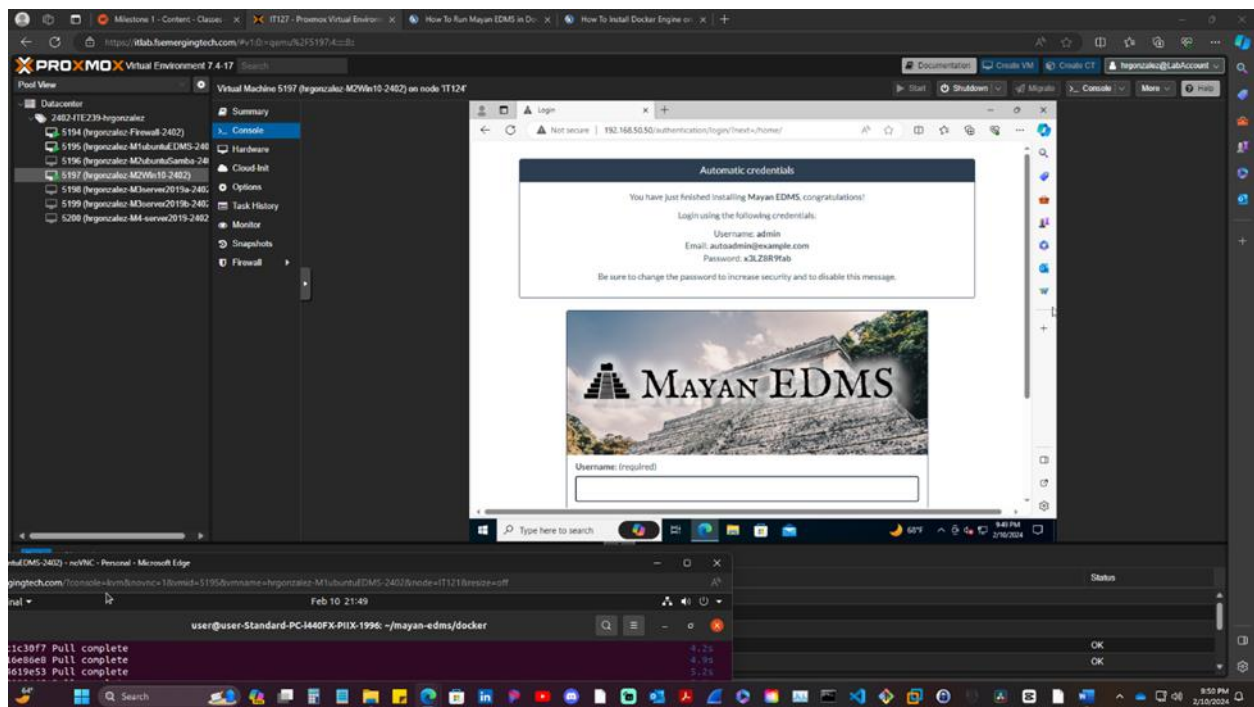
## Verification of Mayan EDMS Installation

Verification can be done 2 ways. One way to do this is by going to a host machine, open web browser and input the Ubuntu VM IP address. The second way it can be done is using `http://localhost/` on your Ubuntu VM.

To find IP address of your Ubuntu VM use command line `> ip addr`

*\*I used Windows 10 VM to do this due to lack of hardware specifications on the Ubuntu VM needed to run smoothly while running Mayan EDMS\*.*

Open Windows 10 web browser and used IP address `> 192.168.50.50` (yours is most likely different). If successfully installed your page should look like the picture below.



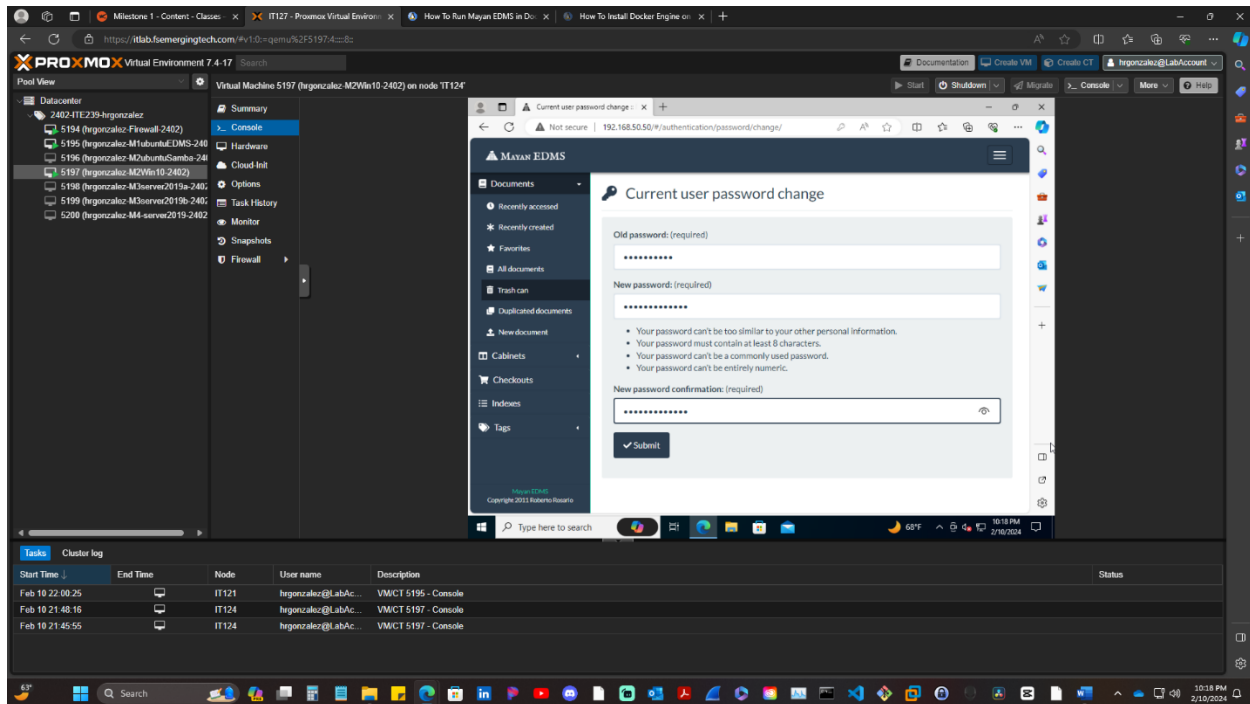
## Mayan Log In

Use the automatic created credentials as shown on Mayan EDMS page.

Log in username is “admin” > Password is “password” replace “password” with your given password in the Mayan credential box > sign in.

## Change Mayan EDMS logging password

Inside Mayan EDMS go to the upper right icon and click on it (icon has 3 dash lines) > user > change password.





## Mayan EDMS Notable Features Write Up

### Document Versioning

When using document versioning an associated document version object is created. This occurs each time a document file is submitted. These document versions function as a presentation layer, showing the user the document files' pages in a certain arrangement.

Document versions work as views that let users engage with the content of the documents without changing the source files. This implies that changes made to the document version, such as adding, deleting, or rearranging pages, can be made without changing the underlying source file. These are non-destructive changes that happen only within the application.

In addition, much as with document file pages, changes can also be made to document versions' pages. It is crucial to remember that while changes made to a document version page won't impact the source document file, changes made to a document file page will impact all related document version pages.

Document versions are not available for direct download because they are virtual objects. Instead, a downloaded version of the document version is made available to users by exporting it as PDFs. This guarantees that users can see and distribute the version of the document without changing the original files.

### Document Configuration

One of the configurations that can be made which is really crucial is the level of security. Setting several degrees of access for users, such as read-only, editing, and administrative privileges. We can establish version control rules, such as whether people can create new versions, delete existing versions, or identify active versions.

### Document Versioning in real world scenario

**Collaborative Editing of Documents:** Project teams can upload files to the platform, work together in real time, and make as many copies of the document as they require. Any team member can annotate or modify a document without changing the original. Managers and supervisors have the authority to examine and approve documents, offer suggestions, and give their final approval before distributing them to other parties. It is simple for them to keep track of changes and guarantee compliance by comparing various versions.

### Workflow Automation

Workflow automation speeds up document-centric operations by automating repetitive procedures, routing documents for review and approval, and enforcing business standards. Mayan EDMS has strong workflow automation capabilities, allowing enterprises to create unique workflows based on their specific needs.

## Configuration of Workflow Automation

Process automation in Mayan EDMS is configured by establishing process templates, document states, and transition rules between states. Administrators can define workflow phases, assign tasks to users or groups, and set completion dates. They can also use conditioned reasoning and notifications to efficiently manage complex document workflows.

## Workflow Automation in real world scenario

Consider a real-world scenario in which a financial organization processes loan applications. Loan documents can be automatically routed through several phases of approval using workflow automation in Mayan EDMS, from initial submission to final judgment. Managers can track the status of each application, ensure compliance with regulatory standards, and speed up the loan approval process, resulting in increased customer satisfaction and operational agility.

## Color coded tags

Documents can be graphically categorized and arranged using color-coded tags. Possible meanings for each hue include category, rank, and importance.

## Configuration of color coded tags

Users have the option to select set of color codes or configure their own. Red might represent urgent documents, green could represent duties that have been finished, etc.

## Color coded tags in real world scenario

Users can categorize their documents with these color-coded tags according to their state or relevancy. This feature makes documents easier to find quickly and improves their arrangement.

## References

[How To Install Docker Engine on Linux Systems | ComputingForGeeks](#)

[How To Run Mayan EDMS in Docker Containers | ComputingForGeeks](#)

[Document versioning — Mayan EDMS 4.6.1 documentation \(mayan-edms.com\)](#)

[Features — Mayan EDMS 4.6.1 documentation \(mayan-edms.com\)](#)

[Tags — Mayan EDMS 4.6.1 documentation \(mayan-edms.com\)](#)