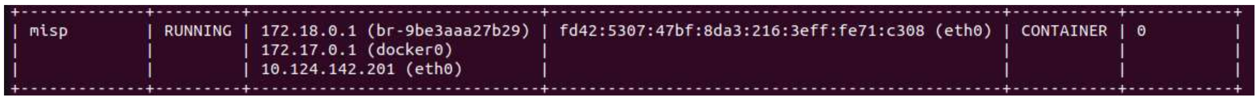
Create LXD container for MISP

```
lokesh@cybercub:~$ lxc launch ubuntu:22.04 misp -c security.nesting=true
Creating misp
Starting misp
```

```
lokesh@cybercub:~$ lxc list
```



```
+------------+---------+------------------------------+----------------------------------------------+-----------+-----------+
| misp       | RUNNING | 172.18.0.1 (br-9be3aaa27b29) | fd42:5307:47bf:8da3:216:3eff:fe71:c308 (eth0) | CONTAINER | 0         |
|            |         | 172.17.0.1 (docker0)         |                                              |           |           |
|            |         | 10.124.142.201 (eth0)        |                                              |           |           |
+------------+---------+------------------------------+----------------------------------------------+-----------+-----------+
```

Access the container

```
lokesh@cybercub:~$ lxc exec misp /bin/bash
root@misp:~# ls
snap
```

Check OS

```
root@misp:~# cat /etc/os-release
PRETTY_NAME="Ubuntu 22.04.5 LTS"
NAME="Ubuntu"
VERSION_ID="22.04"
VERSION="22.04.5 LTS (Jammy Jellyfish)"
VERSION_CODENAME=jammy
ID=ubuntu
ID_LIKE=debian
HOME_URL="https://www.ubuntu.com/"
SUPPORT_URL="https://help.ubuntu.com/"
BUG_REPORT_URL="https://bugs.launchpad.net/ubuntu/"
PRIVACY_POLICY_URL="https://www.ubuntu.com/legal/terms-and-policies/privacy-policy"
UBUNTU_CODENAME=jammy
```

Create a user

```
root@misp:~# sudo useradd lokesh -m -s /bin/bash
root@misp:~# usermod lokesh -aG sudo
root@misp:~# passwd lokesh
New password:
Retype new password:
passwd: password updated successfully
root@misp:~# ls
snap
root@misp:~# su - lokesh
```

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

lokesh@misp:~$
```

Install Docker Engine on Ubuntu LXD Container

Setup Docker's apt repository :

```
lokesh@misp:~$ sudo apt-get update
[sudo] password for lokesh:
Hit:1 http://archive.ubuntu.com/ubuntu focal InRelease
Get:2 http://security.ubuntu.com/ubuntu focal-security InRelease [128 kB]
Get:3 http://archive.ubuntu.com/ubuntu focal-updates InRelease [128 kB]
Get:4 http://security.ubuntu.com/ubuntu focal-security/main amd64 Packages [3238 kB]

lokesh@misp:~$ sudo apt-get install ca-certificates curl -y
Reading package lists... Done
Building dependency tree
Reading state information... Done

lokesh@misp:~$ sudo install -m 0755 -d /etc/apt/keyrings
lokesh@misp:~$ sudo curl -fsSL https://download.docker.com/linux/ubuntu/gpg -o
/etc/apt/keyrings/docker.asc
lokesh@misp:~$ sudo chmod a+r /etc/apt/keyrings/docker.asc
lokesh@misp:~$ echo \
>  "deb [arch=$(dpkg --print-architecture) signed-by=/etc/apt/keyrings/docker.asc]
https://download.docker.com/linux/ubuntu \
> $(. /etc/os-release && echo "$VERSION_CODENAME") stable" | \
> sudo tee /etc/apt/sources.list.d/docker.list > /dev/null
lokesh@misp:~$ echo \
>  "deb [arch=$(dpkg --print-architecture) signed-by=/etc/apt/keyrings/docker.asc]
https://download.docker.com/linux/ubuntu \
> $(. /etc/os-release && echo "$VERSION_CODENAME") stable" | \
> sudo tee /etc/apt/sources.list.d/docker.list > /dev/null
lokesh@misp:~$ sudo apt-get update
Hit:1 http://security.ubuntu.com/ubuntu focal-security InRelease
Hit:2 http://archive.ubuntu.com/ubuntu focal InRelease
```

Install the Docker packages:

```
lokesh@misp:~$ sudo apt-get install docker-ce docker-ce-cli containerd.io docker-buildx-plugin
docker-compose-plugin -y
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  docker-ce-rootless-extras pigz slirp4netns


lokesh@misp:~$ sudo docker run hello-world
Unable to find image 'hello-world:latest' locally
latest: Pulling from library/hello-world
c1ec31eb5944: Pull complete
Digest: sha256:d211f485f2dd1dee407a80973c8f129f00d54604d2c90732e8e320e5038a0348
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/
```

Setting Up of Docker Compose

To download and install Compose standalone, run

```
lokesh@misp:~$ sudo curl -SL
https://github.com/docker/compose/releases/download/v2.29.0/docker-compose-linux-x86_64 -o
/usr/local/bin/docker-compose
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
  0     0    0     0    0     0      0      0 --:--:-- --:--:-- --:--:--     0
100 60.2M  100 60.2M    0     0  3456k      0  0:00:17  0:00:17 --:--:-- 3811k
```

Apply executable permissions to the standalone binary in the target path for the installation.

```
lokesh@misp:~$ sudo chmod +x /usr/local/bin/docker-compose
```

Test and execute compose commands using docker-compose

```
lokesh@misp:~$ sudo ln -s /usr/local/bin/docker-compose /usr/bin/docker-compose
```

Test the Installation

```
lokesh@misp:~$ docker-compose --version
Docker Compose version v2.29.0
```

MISP Set Up

Clone from github

```
lokesh@misp:~$  git clone https://github.com/MISP/misp-docker
Cloning into 'misp-docker'...
remote: Enumerating objects: 2071, done.
remote: Counting objects: 100% (458/458), done.
remote: Compressing objects: 100% (155/155), done.
remote: Total 2071 (delta 400), reused 321 (delta 300), pack-reused 1613 (from 1)
Receiving objects: 100% (2071/2071), 453.42 KiB | 1.17 MiB/s, done.
```

```
Resolving deltas: 100% (1112/1112), done.
```

Get into misp-docker folder

```
lokesh@misp:~$ ls
misp-docker
lokesh@misp:~$ cd misp-docker/
```

Copying into .env

By running this command, you create a .env file (if it doesn't already exist) with the same contents as template.env. .env files are commonly used to store environment variables for applications, such as API keys, database credentials, and other configurations, in a format that can be easily loaded into an application.

```
lokesh@misp:~/misp-docker$ cp template.env .env
```

Edit .env give BASE_URL and Credentails

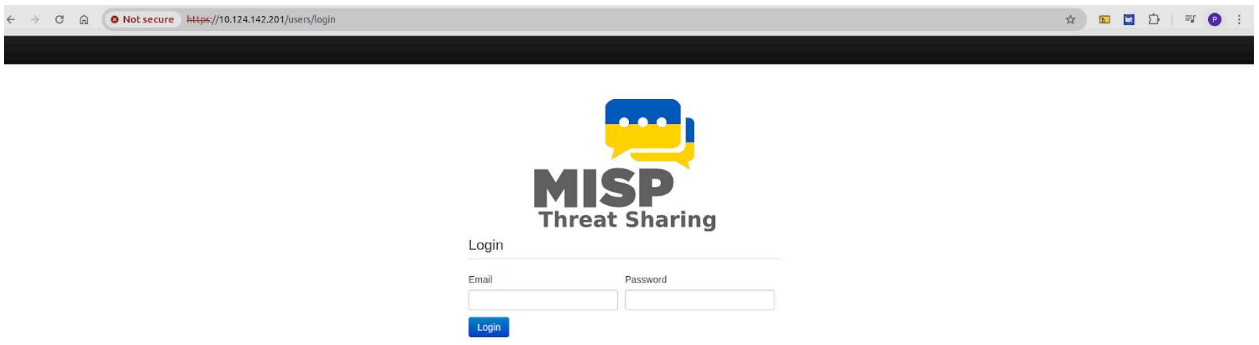
```
lokesh@misp:~/misp-docker$ sudo nano .env
ADMIN_EMAIL="admin@admin.test"
# name of org #1, default to MISP's default (ORGNAME)
ADMIN_ORG=
# defaults to an automatically generated one
ADMIN_KEY=
# defaults to MISP's default (admin)
ADMIN_PASSWORD="admin"
# defaults to 'passphrase'
GPG_PASSPHRASE=
# defaults to 1 (the admin user)
CRON_USER_ID=
# defaults to 'https://localhost'
BASE_URL="https://10.124.142.201"
```

Docker-Compose

This command will pull the images for all the services defined in your docker-compose.yml file.

```
lokesh@misp:~/misp-docker$ sudo docker-compose pull
 ✓ misp-core Pulled
 ✓ misp-modules Pulled
 ✓ mail Pulled
 ✓ redis Pulled
 ✓ db Pulled
```

lokesh@misp:~/misp-docker$ sudo docker-compose up -d
docker-compose up: This command does the work of the docker-compose build and docker-compose run commands. It builds the images if they are not located locally and starts the containers. If images are already built, it will fork the container directly and load as demon by -d



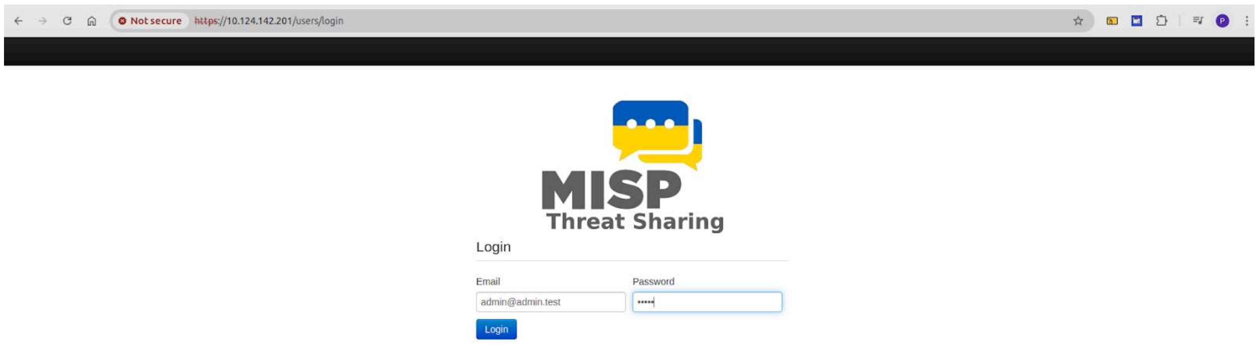
Not secure ~~https~~://10.124.142.201/users/login

Login

Email

Password

Login



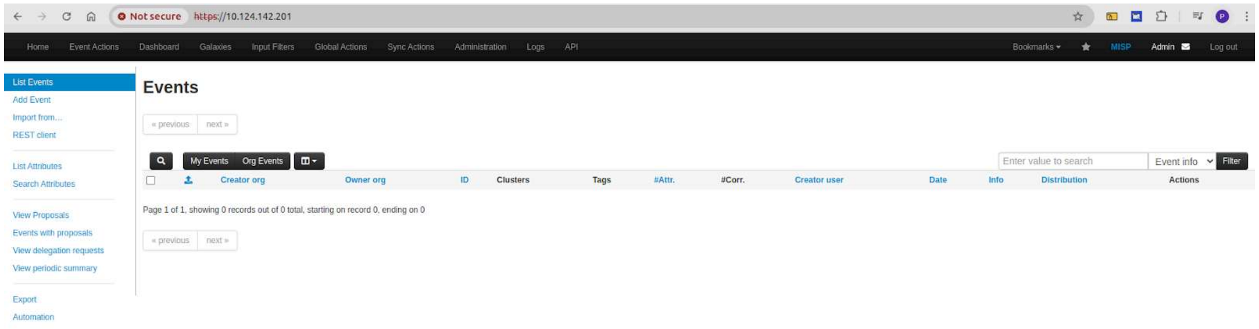
Not secure ~~https~~://10.124.142.201/users/login

Login

Email
admin@admin.test

Password
•••••

Login



Events

		Creator org	Owner org	ID	Clusters	Tags	#Attr.	#Corr.	Creator user	Date	Info	Distribution	Actions

Page 1 of 1, showing 0 records out of 0 total, starting on record 0, ending on 0