Steps to install docker:

sudo apt install apt-transport-https ca-certificates curl software-properties-common -y

sudo curl -fsSL https://download.docker.com/linux/ubuntu/gpg -o /etc/apt/keyrings/docker.asc

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

sudo apt update

sudo apt-get install docker-ce docker-ce-cli containerd.io docker-buildx-plugin docker-compose-plugin

To check if docker is installed
sudo docker –version

Docker desktop download

https://desktop.docker.com/linux/main/amd64/docker-desktop-amd64.deb?utm_source=docker&utm_medium=webreferral&utm_campaign=docs-driven-download-linux-amd64&_gl=1*e1j4gm*_ga*MTAwMjk1MjczOS4xNzMyMDg1ODYx*_ga_XJWPQMJYHQ*MTczMjA5OTEwNC4zLjEuMTczMjA5OTIyMS42MC4wLjA.

Download .deb file

sudo dpkg -i <debian filename>

Once Docker is installed next we have to sigh in

Docker Desktop for Linux relies on `pass`

to store credentials in gpg2-encrypted files. Before signing in to Docker Desktop with your Docker ID, you must initialize `pass`. Docker Desktop displays a warning if you've not initialized `pass`.

You can initialize pass by using a gpg key. To generate a gpg key, run:

```
gpg --generate-key
```

The following is an example similar to what you see once you run the previous command:

```
...
GnuPG needs to construct a user ID to identify your key.

Real name: Molly
Email address: molly@example.com
You selected this USER-ID:
   "Molly <molly@example.com>"

Change (N)ame, (E)mail, or (O)kay/(Q)uit? O
...
```

```
pubrsa3072 2022-03-31 [SC] [expires: 2024-03-30]
 <generated gpg-id public key>
uid          Molly <molly@example.com>
subrsa3072  2022-03-31 [E] [expires: 2024-03-30]
```

To initialize `pass`, run the following command using the public key generated from the previous command:

```
 pass init <your_generated_gpg-id_public_key>
```

The following is an example similar to what you see once you run the previous command:

```
mkdir: created directory '/home/molly/.password-store/'
Password store initialized for <generated_gpg-id_public_key>
```

Once you initialize `pass`, you can sign in and pull your private images. When Docker CLI or Docker Desktop use credentials, a user prompt may pop up for the password you set during the gpg key generation.

```
 docker pull molly/privateimage
Using default tag: latest
latest: Pulling from molly/privateimage
3b9cc81c3203: Pull complete
Digest: sha256:3c6b73ce467f04d4897d7a7439782721fd28ec9bf62ea2ad9e81a5fb7fb3ff96
Status: Downloaded newer image for molly/privateimage:latest
docker.io/molly/privateimage:latest
```

Milvus setup we have to do next:

Inside the project where the project is there create docker-compose.yml file
here copy this yml script:

```
version: '3.5'

services:
  milvus-etcd:
    container_name: milvus-etcd
    image: quay.io/coreos/etcd:v3.5.5
    environment:
      - ETCD_ADVERTISE_CLIENT_URLS=http://milvus-etcd:2379
      - ETCD_LISTEN_CLIENT_URLS=http://0.0.0.0:2379
    volumes:
      - ./volumes/etcd:/etcd
    ports:
      - "2379:2379"  # Expose etcd to the host

  milvus-minio:
    container_name: milvus-minio
```

```
    image: minio/minio:RELEASE.2022-03-17T06-34-49Z
    environment:
      MINIO_ACCESS_KEY: minioadmin
      MINIO_SECRET_KEY: minioadmin
    volumes:
      - ./volumes/minio:/minio_data
    command: minio server /minio_data
    ports:
      - "9000:9000"  # Expose MinIO to the host

  milvus-standalone:
    container_name: milvus-standalone
    image: milvusdb/milvus:v2.3.0
    command: ["milvus", "run", "standalone"]
    environment:
      ETCD_ENDPOINTS: milvus-etcd:2379
      MINIO_ADDRESS: milvus-minio:9000
    volumes:
      - ./volumes/milvus:/var/lib/milvus
    ports:
      - "19530:19530"
    depends_on:
      - milvus-etcd
      - milvus-minio

networks:
  default:
    name: milvus-network
```

Once i have saved the file next I have to write the docker compose up -d to start the milvus
To stop the milvus i have to docker compose down