Se genera un contenedor a partir d ela imagen eclipse-mosquitto

docker run -d --name mosquitto-container 1883:1883 -p 9001:9001 eclipse-mosquitto

se accede al contenedor para configurar el servicio

docker exec -it mosquitto\_agosto /bin/sh

para instalar en el contenedor nano y poder editar la configuración del servidor de mosquitto

apk update

apk add nano

se edita el siguiente archivo

nano mosquitto/config/mosquitto.conf

se ubican en el archivo las siguientes líneas

se descomentan y setean los siguientes paramentros

allow\_anonymous true

listener 1883 0.0.0.0

una vez hecho esto se sale del contenedor

y reinicia

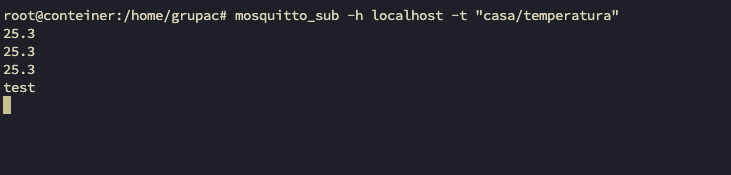
docker restart mosquitto-container

se instala la librería de donde se encuentra para probar de mqtt cliente

mosquitto\_sub -h 192.168.16.76 -t “prueba/robert”

mosquitto\_pub -h localhost -p 1883 -t "prueba/robert" -m "¡Hola MQTT desde Docker!"

y se denerian ver en los nodos subscritos



Se procede a prototipar con node-red

Con el siguiente comando se instala y corre node-red

docker run -d -p 1880:1880 --name mynodered nodered/node-red

desde la url en el explorador puedee entra para gesstionar cualquier tipo de trabejo

CREAR CONTENEDOR PARA MONGO Y GESTIONARLO CON MONGOEXPRESS

<https://github.com/cataniamatt/mongodb-docker>

mongodbcompass

admin

pass

mongodb autfentification

usr: admin

pwd: Grupac123\*

collection: sensor\_mqtt

Url: 192.168.10.175

Port;

database: almacen\_db

user: admin

operation:inserOne

LLMS info

https://jggomezt.medium.com/building-local-ai-applications-integrating-docker-model-runner-genkit-and-langchain-d0dfb4a4dfa7

install docker model run

<https://docs.docker.com/ai/model-runner/>

sudo apt-get update

sudo apt-get install docker-model-plugin

test instalacion

docker model version

docker model run ai/smollm2

<https://hub.docker.com/r/ai/smollm2>

ejemplo

from openai import OpenAI

client = OpenAI(

base\_url="http://localhost:12434/engines/v1", # o según tu endpoint

api\_key="local-key" # cualquier texto, si el servidor local no requiere cl$

)

response = client.completions.create(

# model="ai/gemma3:latest",

model="ai/smollm2:360M-Q4\_K\_M",

prompt="¿Cómo preparo los chilaquiles?",

max\_tokens=100,

)

print(response.choices[0].text)