



05 de octubre del 2025

***Despliegue de bróker MQTT  
con menor tiempo de latencia  
en la transmisión de datos en  
arquitecturas IoT: Caso de  
estudio monitoreo de ruido***

Manual de configuración de  
máquinas virtuales



Thais Nicole Cartuche Peralta  
UNIVERSIDAD NACIONAL DE LOJA

El presente manual detalla el proceso de instalación, configuración y puesta en marcha de tres brokers MQTT (EMQX, VerneMQ y Mosquitto) desplegados en máquinas virtuales Ubuntu Server 22.04 dentro de Microsoft Azure. Cada broker fue instalado en una máquina virtual independiente con IP pública y configurado para permitir conectividad externa mediante los puertos MQTT estándar.

### Creación de las Máquinas Virtuales en Azure

Cada máquina virtual fue creada a través del portal de Azure, como se muestra en la figura 1. Se puede observar la lista de instancias desplegadas todas operando bajo Linux en la región "West US 2", con sus respectivos nombres, direcciones IP y configuraciones activas.

<input type="checkbox"/>	Nombre ↑		Suscripción	Grupo de recu...	Ubicación	Estado	Sistema operat...	Cambiar el ta...	Dirección IP p...	Discos
<input type="checkbox"/>	 brokerEmqx	...	Azure for Stude...	Brokers_publicos	West US 2	En ejecución	Linux	Standard_B1s	20.64.244.7	1
<input type="checkbox"/>	 brokerMosquitto	...	Azure for Stude...	Brokers_publicos	West US 2	En ejecución	Linux	Standard_B1s	74.179.58.252	1
<input type="checkbox"/>	 brokerVerneMQ	...	Azure for Stude...	Brokers_publicos	West US 2	En ejecución	Linux	Standard_B1s	20.64.248.245	1

Figura 1. Instancias de máquinas virtuales

En la tabla 1 se pueden apreciar las características generales de las máquinas virtuales:

Tabla 1. Características generales de las máquinas virtuales

Parámetro	Valor
Imagen	Ubuntu Server 22.04 LTS
Tamaño	Standard_B1s (1 vCPU, 1 GB RAM)
Autenticación	SSH Key
Red	Red virtual única con subred predeterminada
IP pública	Estática
Puertos abiertos	22 (SSH), 1883 (MQTT), 8080 (Panel Web si aplica)

### Habilitación de puerto 1883

En cada una de las instancias nos dirigimos a configuración de red y creamos una regla de puerto de entrada en la cual habilitamos el puerto 1883 tal y como se muestra en la figura 2 este puerto es el por defecto de las conexiones MQTT sin cifrado.



Prioridad ↑	Nombre		Puerto	Protocolo	Origen	Destino	Acción
✓	Reglas de puerto de entrada (5)						
102	MQTT		1883	TCP	Cualquiera	Cualquiera	Allow
300	SSH		22	TCP	Cualquiera	Cualquiera	Allow

Figura 2. Habilitación del puerto 1883

### Instalación y configuración de EMQX (20.64.244.7)

1. Descargar el “.deb” de EMQX (Ultima versión estable)

```
wget https://www.emqx.com/en/downloads/broker/5.8.0/emqx-5.8.0-ubuntu22.04-amd64.deb
```

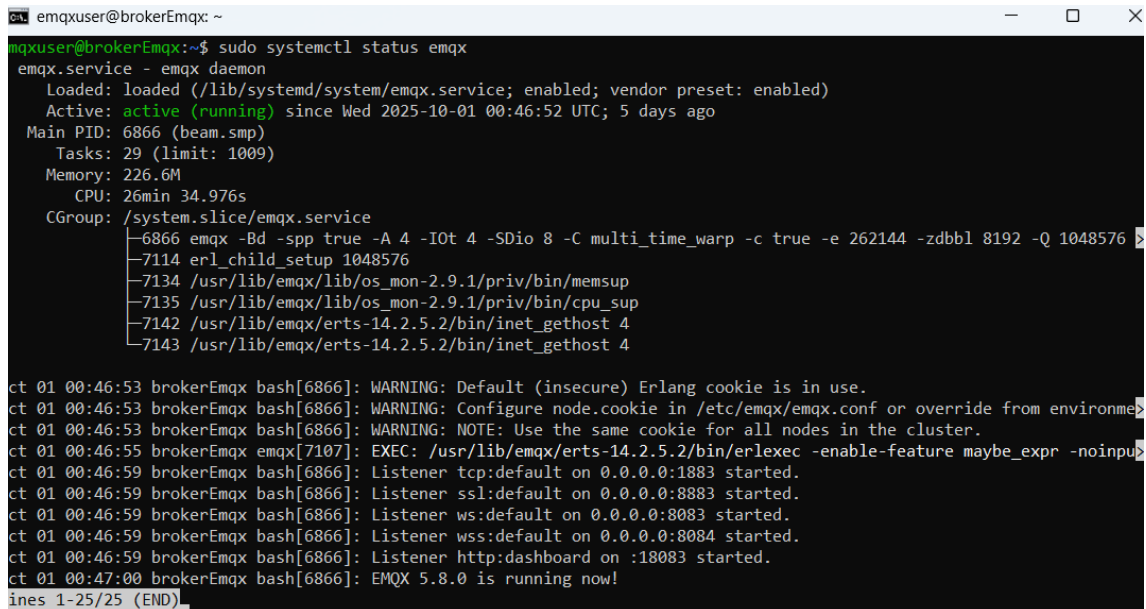
2. Instalar el paquete

```
sudo dpkg -i emqx-5.8.0-ubuntu22.04-amd64.deb
```

3. Iniciar y habilitar EMQX

```
sudo systemctl start emqx
sudo systemctl enable emqx
sudo systemctl status emqx
```

NOTA: EMQX escucha en 0.0.0.0 por lo que ya responde directamente en la IP publica y únicamente es necesario tocar el emqx.conf si se requiere restringir a localhost o a una red interna.



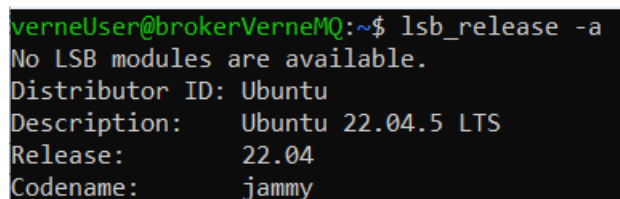
```
emqxuser@brokerEmqx: ~  
emqxuser@brokerEmqx:~$ sudo systemctl status emqx  
emqx.service - emqx daemon  
Loaded: loaded (/lib/systemd/system/emqx.service; enabled; vendor preset: enabled)  
Active: active (running) since Wed 2025-10-01 00:46:52 UTC; 5 days ago  
Main PID: 6866 (beam.smp)  
Tasks: 29 (limit: 1009)  
Memory: 226.6M  
CPU: 26min 34.976s  
CGroup: /system.slice/emqx.service  
├─6866 emqx -Bd -spp true -A 4 -IOT 4 -SDio 8 -C multi_time_warp -c true -e 262144 -zdbbl 8192 -Q 1048576  
├─7114 erl_child_setup 1048576  
├─7134 /usr/lib/emqx/lib/os_mon-2.9.1/priv/bin/memsup  
├─7135 /usr/lib/emqx/lib/os_mon-2.9.1/priv/bin/cpu_sup  
├─7142 /usr/lib/emqx/erts-14.2.5.2/bin/inet_gethost 4  
└─7143 /usr/lib/emqx/erts-14.2.5.2/bin/inet_gethost 4  
  
ct 01 00:46:53 brokerEmqx bash[6866]: WARNING: Default (insecure) Erlang cookie is in use.  
ct 01 00:46:53 brokerEmqx bash[6866]: WARNING: Configure node.cookie in /etc/emqx/emqx.conf or override from environme  
ct 01 00:46:53 brokerEmqx bash[6866]: WARNING: NOTE: Use the same cookie for all nodes in the cluster.  
ct 01 00:46:55 brokerEmqx emqx[7107]: EXEC: /usr/lib/emqx/erts-14.2.5.2/bin/erlexec -enable-feature maybe_expr -noinput  
ct 01 00:46:59 brokerEmqx bash[6866]: Listener tcp:default on 0.0.0.0:1883 started.  
ct 01 00:46:59 brokerEmqx bash[6866]: Listener ssl:default on 0.0.0.0:8883 started.  
ct 01 00:46:59 brokerEmqx bash[6866]: Listener ws:default on 0.0.0.0:8083 started.  
ct 01 00:46:59 brokerEmqx bash[6866]: Listener wss:default on 0.0.0.0:8084 started.  
ct 01 00:46:59 brokerEmqx bash[6866]: Listener http:dashboard on :18083 started.  
ct 01 00:47:00 brokerEmqx bash[6866]: EMQX 5.8.0 is running now!  
lines 1-25/25 (END)
```

Figura 3. Estado del broker EMQX

## Instalación y configuración de VerneMQ (20.64.248.245)

1. Revisamos la distribución de Linux que estamos ejecutando con el siguiente comando, la salida del comando se observa en la figura 4.

```
lsb_release -a
```



```
verneUser@brokerVerneMQ:~$ lsb_release -a  
No LSB modules are available.  
Distributor ID: Ubuntu  
Description:    Ubuntu 22.04.5 LTS  
Release:        22.04  
Codename:       jammy
```

Figura 4. Distribución de linux

2. Descargamos el paquete para Jammy Jellyfish

```
wget
```

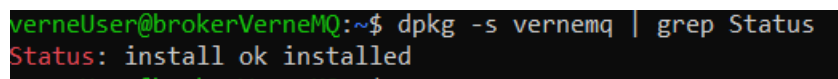
```
https://github.com/vernemq/vernemq/releases/download/2.1.1  
/vernemq-2.1.1.jammy.x86_64.deb
```

3. Instalamos el paquete

```
sudo dpkg -i vernemq-2.1.1.jammy.x86_64.deb
```

4. Verificamos que se haya instalado de manera correcta. En la figura 5 se observa la salida del comando que indica que el bróker está instalado correctamente.

```
dpkg -s vernemq | grep Status
```



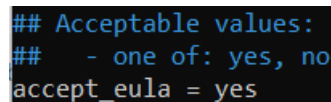
```
verneUser@brokerVerneMQ:~$ dpkg -s vernemq | grep Status  
Status: install ok installed
```

*Figura 5. Verificación de instalación*

5. Nos dirigimos al archivo de configuración en el cual se van a realizar las siguientes configuraciones para el correcto funcionamiento del bróker

```
sudo nano /etc/vernemq/vernemq.conf
```

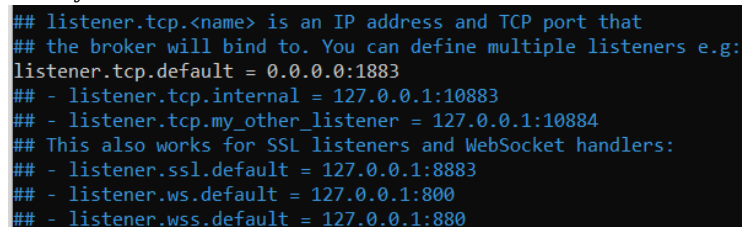
- Aceptar licencia EULA (Acuerdo de Licencia de Usuario Final) como se muestra en la figura 6. Esta es un contrato legal entre el propietario del software y el usuario final que establece las condiciones de uso, los derechos y las restricciones sobre cómo se puede utilizar un programa.



```
## Acceptable values:  
## - one of: yes, no  
accept_eula = yes
```

*Figura 6. Licencia aceptada*

- Des comentamos o escribimos la línea que muestra en la figura 7. Esta línea configura el puerto de escucha principal de VerneMQ, indicando al broker que acepte conexiones MQTT a través del puerto 1883 desde cualquier dirección de red, permitiendo que clientes locales y remotos se conecten al servidor MQTT.



```
## listener.tcp.<name> is an IP address and TCP port that  
## the broker will bind to. You can define multiple listeners e.g:  
listener.tcp.default = 0.0.0.0:1883  
## - listener.tcp.internal = 127.0.0.1:10883  
## - listener.tcp.my_other_listener = 127.0.0.1:10884  
## This also works for SSL listeners and WebSocket handlers:  
## - listener.ssl.default = 127.0.0.1:8883  
## - listener.ws.default = 127.0.0.1:800  
## - listener.wss.default = 127.0.0.1:880
```

*Figura 7. Configuración del puerto*

- Y por último como no está en un entorno de producción modificamos la siguiente línea como se muestra en la figura 8 que se encuentra en “off” por “on” para permitir las conexiones sin usuario/contraseña.

```
## Allow anonymous users to connect, default is 'off'. !!NOTE!!
## Enabling this completely disables authentication of the clients and
## should only be used for testing/development purposes or in case
## clients are authenticated by some other means.
##
## Default: off
##
## Acceptable values:
##   - on or off
allow_anonymous = on
```

Figura 8. Permitir conexiones sin autenticación

- Guardamos el archivo de configuración y reiniciamos el servicio con los siguientes comandos

```
sudo systemctl restart verneq
sudo systemctl enable verneq
```

- Verificamos el estado del broker como se muestra en la figura 9 con el siguiente comando.

```
sudo systemctl status verneq
```

```
verneUser@brokerVerneMQ:~$ sudo systemctl status verneq
● verneq.service - VerneMQ Server
   Loaded: loaded (/lib/systemd/system/verneq.service; enabled; vendor preset: enabled)
   Active: active (running) since Mon 2025-10-06 21:49:55 UTC; 3s ago
     Process: 18756 ExecStartPre=/usr/lib/verneq/bin/verneq chkconfig (code=exited, status=0/SUCCESS)
    Main PID: 18905 (run_eri)
      Tasks: 173 (limit: 1009)
     Memory: 117.1M
        CPU: 9.056s
    CGroup: /system.slice/verneq.service
            └─18836 /usr/lib/verneq/erts-15.2.6/bin/epmd -daemon
              └─18905 /usr/lib/verneq/erts-15.2.6/bin/run_eri /tmp/verneq// /var/log/verneq "exec /usr/lib/verneq/bin/
                └─18907 /usr/lib/verneq/erts-15.2.6/bin/beam.smp -P 256000 -e 256000 -Q 262144 -A 64 -K true -W w -- -root
                  └─19124 erl_child_setup 1048576
                    └─19174 sh -s disksetup
                      └─19176 /usr/lib/verneq/lib/os_mon-2.10.1/priv/bin/memsup
                        └─19177 /usr/lib/verneq/lib/os_mon-2.10.1/priv/bin/cpu_sup

Oct 06 21:49:46 brokerVerneMQ systemd[1]: Starting VerneMQ Server...
Oct 06 21:49:49 brokerVerneMQ verneq[18756]: config is OK
Oct 06 21:49:49 brokerVerneMQ verneq[18756]: -config /var/lib/verneq/generated.configs/app.2025.10.06.21.49.47.config
Oct 06 21:49:49 brokerVerneMQ systemd[1]: verneq.service: Found left-over process 18836 (epmd) in control group while s
Oct 06 21:49:49 brokerVerneMQ systemd[1]: This usually indicates unclean termination of a previous run, or service impl
Oct 06 21:49:49 brokerVerneMQ verneq[18837]: vmq_cluster_node_sup
Oct 06 21:49:52 brokerVerneMQ verneq[19053]: Starting up
Oct 06 21:49:55 brokerVerneMQ systemd[1]: Started VerneMQ Server.
lines 1-25/25 (END)
```

Figura 9. Estado del broker VerneMQ

## Instalación y configuración de mosquitto (74.179.58.252)

- Instalación directa desde el repositorio oficial y habilitación del bróker con los siguientes comandos

```
sudo apt install -y mosquitto mosquitto-clients
sudo systemctl enable mosquitto
sudo systemctl start mosquitto
```

- Configuramos los puertos y la conexión sin autenticación desde el archivo de configuración entrando con el siguiente comando y escribiendo las líneas que se muestran en la figura 10

```
sudo nano /etc/mosquitto/mosquitto.conf
```

```
#Escuchar desde cualquier dirección en el puerto estandar
listener 1883 0.0.0.0
#Permitir acceso anónimo
allow_anonymous true_
```

*Figura 10. Configuración del bróker*

3. Finalmente reiniciamos para que se apliquen las configuraciones  
`sudo systemctl restart mosquitto`

#### Anexos

- Archivos de configuración de las máquinas virtuales:  
[https://drive.google.com/drive/folders/10dj4TBArvQHDI3vcr\\_YfNEF\\_fY0HxkpK?usp=sharing](https://drive.google.com/drive/folders/10dj4TBArvQHDI3vcr_YfNEF_fY0HxkpK?usp=sharing)