

Guia Prático: Node-RED + MQTT (Mosquitto) na Oracle Cloud Always Free

Texto em português; nomes de menus/campos conforme no console em inglês.

Parte 1 — Criar Conta na Oracle Cloud (site em inglês)

- 1) Acesse <https://www.oracle.com/cloud/free> e clique em **Start for Free**.
- 2) **Create your account**: preencha **First Name**, **Last Name**, **Email**, **Country** e defina a **Password**.
- 3) **Email verification**: confirme pelo link recebido.
- 4) **Payment information**: informe **credit card** internacional físico. Cartão virtual geralmente falha. Sem cobranças se permanecer no **Always Free** (pode haver autorização de US\$0–US\$1).
- 5) **Home Region**: selecione (ex.: **Brazil East (São Paulo)** ou **US East (Ashburn)**).
- 6) Se PJ: **Company Name** e **Tax ID** (CNPJ). PF: dados pessoais (CPF).
- 7) **Phone verification** e **Review & Accept** → **Create Account**.
- 8) Acesse o console em <https://cloud.oracle.com>.

Parte 2 — Criar a VM (Always Free)

- 1) Console: **Menu** → **Compute** → **Instances** → **Create Instance**.
- 2) **Basic information**: **Name** = NodeRED-Server; **Availability domain** = AD-1; **Capacity type** = on-demand; **Fault domain** = Let Oracle choose the best fault domain.
- 3) **Image and shape**: **Image** → **Change image** → **Canonical Ubuntu 22.04**; **Security** → **Shielded instance**: default; **Shape** → **Change shape** → **VM.Standard.E2.1.Micro (Always Free-eligible)**.
- 4) **Networking**: **Create new virtual cloud network + Create new public subnet** (CIDR 10.0.0.0/24). **Public IPv4 address**: Yes (em public subnet vem automático). **Use network security groups**: No.
- 5) **SSH keys**: **Generate an SSH key pair** → baixe a **.pem** (privada) e a **public key**.
- 6) **Boot volume**: default ≈46.6 GB (~50 GB), criptografia Oracle-managed. Sem block volumes.
- 7) **Management / Availability configuration**: default. 8) **Create**.

Parte 3 — Abrir portas na Security List (subnet pública)

Caminho: **Menu** → **Networking** → **Virtual Cloud Networks** → sua VCN → **Security Lists** → abra a lista da public subnet.

Adicione **Ingress Rules** (stateful): TCP 22, TCP 1880, TCP 1883 (todas de 0.0.0.0/0).

Garanta **Egress Rule** ampla: All protocols para 0.0.0.0/0.

Se criou nova Security List, associe em: **VCN** → **Subnets** → sua subnet → **Security Lists** → **Add Security List**.

Obs.: ICMP/ping pode não funcionar; teste pelas portas.

Parte 4 — Encontrar o IP público

Menu → **Compute** → **Instances** → NodeRED-Server → **Instance Information** → **Public IP Address**.

Parte 5 — SSH (cliente)

Linux/macOS:

```
chmod 600 /caminho/chave.pem
ssh -i /caminho/chave.pem ubuntu@<PUBLIC_IP>
```

Windows (PuTTY): converta .pem em .ppk (PuTTYgen); no PuTTY, Host: ubuntu@; selecione a .ppk em Connection → SSH → Auth.

Parte 6 — Docker + Compose (na VM)

```
sudo apt update && sudo apt -y install ca-certificates curl gnupg
curl -fsSL https://get.docker.com | sh
sudo usermod -aG docker $USER
logout
# reconecte por SSH
sudo apt -y install docker-compose-plugin
```

Parte 7 — Node-RED + Mosquitto (compose mínimo)

Na VM:

```
mkdir -p ~/stack && cd ~/stack

cat > docker-compose.yml <<'YML'
services:
  mosquitto:
    image: eclipse-mosquitto:2
    restart: unless-stopped
    ports:
      - "1883:1883"
    command: ["mosquitto", "-c", "/mosquitto-noauth.conf"]
    volumes:
      - ./mosquitto-noauth.conf:/mosquitto-noauth.conf:ro

  nodered:
    image: nodered/node-red:latest
    restart: unless-stopped
    ports:
      - "1880:1880"
    environment:
      - TZ=America/Sao_Paulo
      - NODE_OPTIONS=--max-old-space-size=256
    volumes:
      - ./nodered_data:/data
YML

printf "listener 1883\nallow_anonymous true\n" > mosquitto-noauth.conf

docker compose up -d
docker compose ps
```

Parte 8 — Se Node-RED reiniciar (EACCES)

```
cd ~/stack
docker compose down
[ -d nodered_data ] || { rm -f nodered_data; mkdir -p nodered_data; }
sudo chown -R 1000:1000 nodered_data
sudo find nodered_data -type d -exec chmod 775 {} \;
sudo find nodered_data -type f -exec chmod 664 {} \; || true
docker compose up -d
docker compose logs --tail=80 nodered
```

Parte 9 — Testes externos (Windows PowerShell)

```
Test-NetConnection <PUBLIC_IP> -Port 22
Test-NetConnection <PUBLIC_IP> -Port 1880
Test-NetConnection <PUBLIC_IP> -Port 1883
```

Parte 10 — MQTT com senha (após validar)

```
cd ~/stack
mkdir -p mosquitto
printf "listener 1883\nallow_anonymous false\npassword_file /mosquitto/passwd\n" > mosquitto/mosquitto.conf

docker run --rm -it -v "$PWD/mosquitto:/mosquitto" eclipse-mosquitto:2 \
  mosquitto_passwd -c /mosquitto/passwd usuario

sed -i 's|mosquitto-noauth.conf:/mosquitto-noauth.conf:ro|mosquitto:/mosquitto|' docker-compose.yml

docker compose up -d
```

Parte 11 — (Opcional) Swap 1 GB

```
sudo fallocate -l 1G /swapfile
sudo chmod 600 /swapfile
sudo mkswap /swapfile
sudo swapon /swapfile
echo '/swapfile none swap sw 0 0' | sudo tee -a /etc/fstab
```

Validação de Rede (passo adicional recomendado)

Após criar a instância e conectar via SSH, valide a conectividade:

```
ping -4 -c 4 8.8.8.8  
curl -4 ifconfig.me
```

Se responder -> rede OK, prossiga com a instalação.

Se não responder -> a VCN/subnet está com problema de saída (egress).

Nesse caso, recrie a VCN usando o Wizard (VCN with Internet Connectivity) e crie a VM novamente na p