# NetMaker Self-Hosted

Installing Netmaker with an existing Caddy reverse proxy is a common setup for users who already have a centralized proxy managing their services. The key is to perform a manual installation of Netmaker using Docker Compose, but without the Caddy service that is included in the standard setup scripts.

Here is a step-by-step guide to install Netmaker on your Proxmox LXC at `192.168.1.39` and configure it to work with your existing Caddy instance at `192.168.1.15`.

## Step 1: Prepare the Netmaker LXC Container

First, you need to prepare the new LXC container where Netmaker will be installed.

1. **Access your Netmaker LXC:** SSH into your Proxmox LXC container at `192.168.1.39`.
2. **Install Docker and Docker Compose:** Netmaker runs in Docker containers. If you don't have them installed, run the following commands:

```
# Install Docker
apt-get update
apt-get install -y docker.io

# Install Docker Compose
apt-get install -y docker-compose
```

## Step 2: Set Up Netmaker Installation Files

Instead of using the `nm-quick.sh` script, you will manually create the `docker-compose.yml` and environment files. This allows you to remove the Caddy service.

1. **Create a directory for Netmaker:**

```
mkdir /netmaker
cd /netmaker
```

2. **Create the `docker-compose.yml` file:** Create a file named `docker-compose.yml` and paste the following configuration. This is a modified version of the standard Netmaker `docker-compose` file, with the `caddy` service removed.

```
version: "3.4"
services:
```

```
netmaker:
  container_name: netmaker
  image: gravitl/netmaker:latest
  env_file: ./netmaker.env
  restart: always
  volumes:
    - dnsconfig:/root/config/dnsconfig
    - sqldata:/root/data
  ports:
    - "8081:8081" # Expose API port
  cap_add:
    - NET_ADMIN
    - NET_RAW
    - SYS_MODULE

netmaker-ui:
  container_name: netmaker-ui
  image: gravitl/netmaker-ui:latest
  env_file: ./netmaker.env
  environment:
    # This should point to your Caddy reverse proxy address for the API
    BACKEND_URL: "https://api-net.yourdomain.com"
  restart: always
  ports:
    - "8082:80" # Expose UI port

coredns:
  container_name: coredns
  image: coredns/coredns
  command: -conf /root/dnsconfig/Corefile
  env_file: ./netmaker.env
  restart: always
  volumes:
    - dnsconfig:/root/dnsconfig

mq:
  container_name: mq
  image: eclipse-mosquitto:2.0.15-openssl
  env_file: ./netmaker.env
  restart: unless-stopped
```

```
    command: [ "/mosquitto/config/wait.sh" ]
    volumes:
      - ./mosquitto.conf:/mosquitto/config/mosquitto.conf
      - ./wait.sh:/mosquitto/config/wait.sh
      - mosquitto_logs:/mosquitto/log
      - mosquitto_data:/mosquitto/data
    ports:
      - "8883:8883" # Expose MQTT websocket port


volumes:
  sqldata: {}
  dnsconfig: {}
  mosquitto_logs: {}
  mosquitto_data: {}
```

3. **Create supporting configuration files:** You'll need `mosquitto.conf` and `wait.sh` in the same directory.
   - **`mosquitto.conf`:**

```
wget https://raw.githubusercontent.com/gravitl/netmaker/master/docker/mosquitto.conf
```

   - **`wait.sh`** :

```
wget https://raw.githubusercontent.com/gravitl/netmaker/master/docker/wait.sh
chmod +x wait.sh
```

## Step 3: Configure the Netmaker Environment

Create a `netmaker.env` file to configure your Netmaker server. This is where you'll define your domain, IP address, and other critical settings.

1. **Create the `netmaker.env` file:**

```
touch netmaker.env
```

2. **Edit the file** and add the following, replacing the placeholder values with your own:

```
# == Base Configuration ==
# Your API domain for Netmaker (e.g., api-net.yourdomain.com)
NM_DOMAIN=api-net.yourdomain.com


# Your public IP address
```

```
SERVER_HOST=api-net.yourdomain.com

# A strong, secret key for the API
MASTER_KEY=your_very_secret_master_key

# Your email for SSL certificates (handled by your main Caddy, but good practice)
NM_EMAIL=your.email@example.com

# == Database ==
DATABASE=sqlite

# == Broker Configuration ==
# Credentials for the MQTT broker
MQ_USERNAME=netmaker
MQ_PASSWORD=your_secure_mq_password

# == Netmaker API Port ==
API_PORT=8081

# == Telemetry (Optional) ==
TELEMETRY=off
```

**Important:** In the `docker-compose.yml` file, remember to replace `https://api.yourdomain.com` in the `netmaker-ui` service environment with the actual API URL you will configure in Caddy.

## Step 4: Launch Netmaker

Now you can start the Netmaker services.

1. **Start Docker Compose:** From your `/root/netmaker` directory, run:

```
docker-compose up -d
```

2. **Verify services are running:**

```
docker ps
```

You should see `netmaker`, `netmaker-ui`, `mq`, and `coredns` containers running.

## Step 5: Configure Your Existing Caddy Proxy

On your Caddy server at `192.168.1.15`, you need to add reverse proxy rules to direct traffic to the new Netmaker services on `192.168.1.39`.

Edit your `Caddyfile` and add the following blocks. Replace `netmaker.yourdomain.com` with the domain you set in your `.env` file.

```
# Netmaker Dashboard UI
dashboard-net.yourdomain.com {
    reverse_proxy 192.168.1.39:8082
}

# Netmaker API
api-net.yourdomain.com {
    reverse_proxy 192.168.1.39:8081
}

# Netmaker MQTT Broker (for websockets)
broker-net.yourdomain.com {
    reverse_proxy / 192.168.1.39:8883 {
        header_up Host {http.reverse_proxy.upstream.hostport}
        header_up X-Forwarded-Host {host}
    }
}
```

After saving the `Caddyfile`, reload Caddy to apply the changes:

```
sudo systemctl reload caddy
```

## Step 6: Finalize Port Forwarding and DNS

1. **DNS Records:** Ensure you have the following `A` records pointing your domain to your Caddy server's public IP address:
   - `dashboard.netmaker.yourdomain.com`
   - `api.netmaker.yourdomain.com`
   - `broker.netmaker.yourdomain.com`
2. **Router/Firewall Port Forwarding:** You must forward the necessary UDP ports for WireGuard traffic to your **Netmaker LXC** at `192.168.1.39`. The default port is `51821`.[^2]
   - Forward UDP port `51821` to `192.168.1.39`.

Your Netmaker installation should now be accessible through your existing Caddy reverse proxy. You can access the dashboard at `https://dashboard-net.yourdomain.com`.

**Ins0mniA**

---

Revision #5
Created 2025-09-10 19:29:03 EEST by Green
Updated 2025-09-10 20:51:14 EEST by Green