

Proxmox VE インストール

デフォルトポートは8006

1.DHCPに設定を変更する。

ProxmoxVE(pve)のシェルを開く。

```
nano /etc/network/interfaces
```

固定IPになっているところを調べる（下記は例）

```
iface vmbro inet static
    address 192.168.x.xxx/24
    gateway 192.168.x.1
    bridge-ports enp2s0
    bridge-stp off
    bridge-fd 0
```

固定に変更して上書き(Ctrl + o)終了(Ctrl + x)再起動する

```
iface vmbro inet dhcp
    bridge-ports enp2s0
    bridge-stp off
    bridge-fd 0
```

2.PCIパススルーの設定をする ESXiと異なりBIOS設定してインストールしただけではPCIパススルーが動作しない ProxmoxVE(pve)のシェルを開く。

```
nano /etc/default/grub
```

下記の2行が書かれた箇所を探す。

```
GRUB_CMDLINE_LINUX_DEFAULT="quiet"
GRUB_CMDLINE_LINUX_DEFAULT=""
```

このうち、下の何も記載していない方に追記して、下記のようにする (AMD CPUの場合intel_iommuをamd_iommuに置き換える)

```
GRUB_CMDLINE_LINUX_DEFAULT="quiet"
GRUB_CMDLINE_LINUX_DEFAULT="quiet intel_iommu=on iommu=pt"
```

上書き(Ctrl + o)終了(Ctrl + x)最後に反映のために下記を入力する

```
update-grub
```

最後にrebootして完了となる

<https://www.servethehome.com/how-to-pass-through-pcie-nics-with-proxmox-ve-on-intel-and-amd/>

2.hostnameを変更する。

ProxmoxVE(pve)のシェルを開く。現在の仕様を確認するにはhostnamectl → [Static Hostnameが現在の設定。

```
hostnamectl set-hostname
```

3.ライセンスの状態を修正する。

```
bash -c "$(wget -qL0 -  
https://github.com/tteck/Proxmox/raw/main/misc/post-pve-install.sh)"
```

ProxmoxVE(pve)のシェルを開く。以下をコピーして貼り付ける。

4.内部ネットワークのルーターを作成する。

pveのネットワークを開くと、初期状態でvmbr0がある。作成をクリックしてvmbr1[192.168.100.0/24で作成する。(再起動後に反映)

シェルで下記を実行して[OpenWRTをインストールする。

```
bash -c "$(wget -qL0 -  
https://github.com/tteck/Proxmox/raw/main/vm/openwrt.sh)"
```

インストールはデフォルトでOK

インストール後にシャットダウン[vlanが設定されているほうがLAN側なので[VLAN IDを削除して[vmbr1に変更して起動して完了。

5.Home Assistantのインストール

Home Assistantには種類があって、アドオン入れられるのはHA OSか、 HA Supervisedしかできない。なのでHAのVMを作成する。

```
bash -c "$(wget -qL0 -  
https://github.com/tteck/Proxmox/raw/main/vm/haos-vm.sh)"
```

VMにZigbee Dongle-Pを接続して[Proxmoxのデバイス追加より USB Dongleを設定しておくこと。

6.Home AssistantにZigbee2MQTTを入れる。

参考 : <https://youtu.be/L9IC0Mse0K4?si=gsiOmjTyF3uuAmq9>

A.アドオンストアのMQTTをインストールして、開始する[B.メンバーを追加する。メンバー名:mqtt-user[パスワード:yourmqtt[仮) C.システム - ハードウェア よりすべてのハードウェアを選択する。

Zigbee Dongleのアダプターを探す。自分の環境では/dev/ttyUSB0だった。

D.アドオンストア右上の3点メニューより、下記のリポジトリを追加する。

<https://github.com/zigbee2mqtt/hassio-zigbee2mqtt>

追加後はZigbee2MQTTをインストールする。

E:Z2Mを起動する前に設定タブを開いて、下記の文字列を追加する。B.C.で取得した情報で適宜置き換えること。追記後に保存、サイドバーに表示にチェックを入れてZ2Mを起動する。

```
mqttの項目 :  
server: mqtt://core-mosquitto
```

user: mqtt-user password: yourmqtt

```
serialの項目 :  
port: /dev/ttyUSB0
```

F:Z2Mを起動してSettingよりZigbee Channelを15,20,25のいずれかに変更（もし複数のアクセスポイントがある場合は16,11）して、無線LANとの干渉を抑える。自分は11を選択する。G.同じくSettingよりLast Seenの設定をdisableよりISO_8601_localに変更する。H.同じくSettingよりNetwork keyにGENERATEと入力する。これにより生成されたキーで暗号化されるが、全てのデバイスを再ペアリングする必要がある。I.同じくSettingよりTransmit powerを最大20まで変更できる。初期値は9。実家は広いので20に変更して再起動する。K.この状態で再起動すると、まれに失敗することがある。その場合はFile Kditorを使ってconfig/zigbee2mqtt/coordinator_backup.jsonを削除する。L.Z2MのSettingを再度開き、Availabilityの項目よりAvailability(simple)のチェックボックスを入れ、Submitを押下する。M.ペアリングは、ダッシュボードの上にあるPermit Joinをクリックして、制限時間内にデバイスを探索モードにする。追加後は名前を変更して使用日（電池交換日）を記録する。N:HAの設定のデバイスとサービス - デバイスに追加したデバイスが表示されているのを確認する。H.オーバービューで

□

Dnsmasq

<https://zenn.dev/cran/articles/6c9588e36a0d53>

Proxmoxにてdnsmasqを使ってIPアドレスを節約する

IPアドレスの使用が制限されている環境（会社など）でVMやLXCを設置する場合、VMやLXCのIPアドレスをProxmox内で割り振って、それぞれの必要なポートをProxmoxの別ポートに変換して出力する方法がある。VMやLXC構築前に、下記の準備をしておく。

```
ホストIP 192.168.50.1  
DHCPのスコップ : 192.168.50.50-192.168.50.200
```

A.仮想ネットワークの作成

DHCPを割り振る仮想ネットワークを作成する。今回は名称をvmbr1とする。

```
sudo nano /etc/network/interfaces
```

```
auto vmbr1  
iface vmbr1 inet static  
    address 192.168.50.1
```

```
netmask 255.255.255.0
bridge_ports none
bridge_stp off
bridge_fd 0
post-up echo 1 > /proc/sys/net/ipv4/ip_forward
post-up iptables -t nat -A POSTROUTING -s '192.168.50.0/24' -o enp4s0 -j MASQUERADE
post-down iptables -t nat -D POSTROUTING -s '192.168.50.0/24' -o enp4s0 -j MASQUERADE
```

注記：物理インターフェイスに接続しないのでBridge_portsはnoneに設定しているが、このままでは外部のネットワークに接続ができない。そこで既存のVLANもしくはほかのLinuxブリッジに送受信データをマスカレードする。

```
$sudo ip -f inet a s
```

であらかじめVLANもしくはほかのLinuxブリッジのインターフェイス名を取得しておく。今回の場合はenp4s0だった。

```
$sudo ifup vmbr1 でネットワークを起動する。
$ip address show dev vmbr1と入力してinetの値が192.168.50.1/24となっていれば成功。
sudo systemctl restart networking
$systemctl status networking.service
```

C. /etc/dnsmasq.confをnanoで変更する。

```
#106行目付近
interface=vmbr1

#159行目付近
dhcp-range=192.168.50.50,192.168.50.200,2h

#337行目付近
dhcp-option=vmbr1,3,192.168.50.1
```

#DHCPリースデータベースのパスを記述

```
dhcp-leasefile=/var/lib/misc/dnsmasq.leases
```

設定後は下記で動作確認を行う。

```
$sudo dnsmasq --test
```

dnsmasq: syntax check OK. ←これが返ってくれば、とりまOK

```
sudo systemctl restart dnsmasq
sudo systemctl enable dnsmasq
```

From:

<http://deepsky.jp/wiki/> - うごくといいな

Permanent link:

<http://deepsky.jp/wiki/doku.php?id=memo:pc:pveinstall>

Last update: **2025/10/19 20:34**

