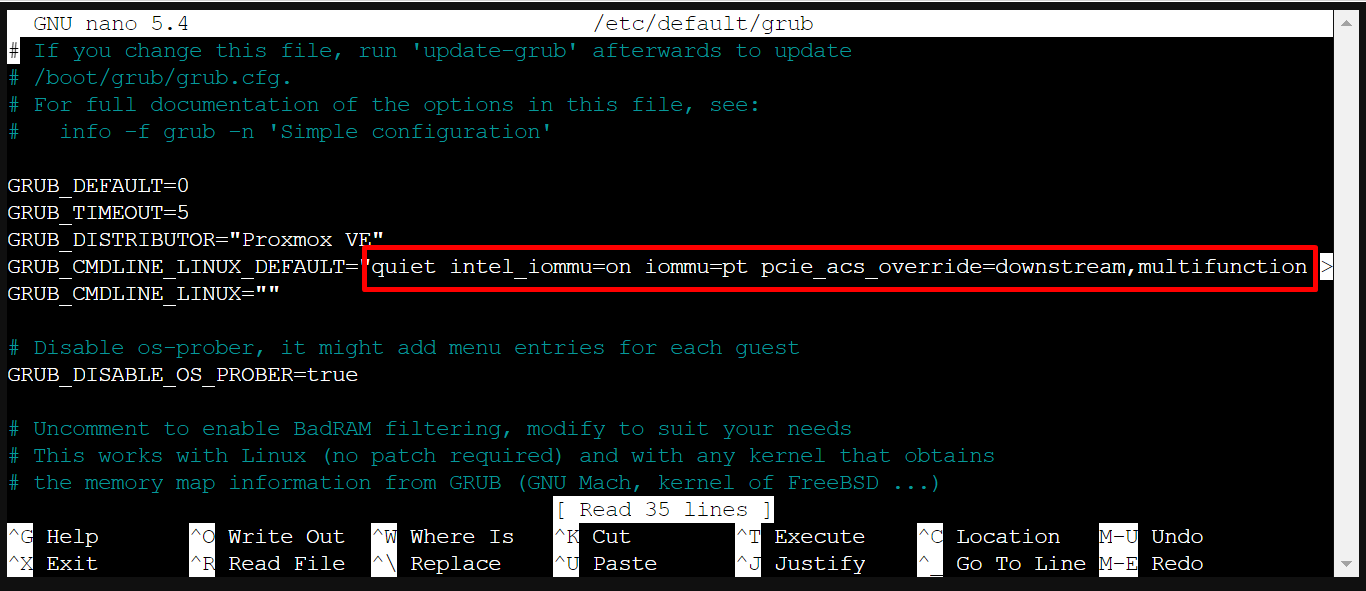
最剛開始設定:

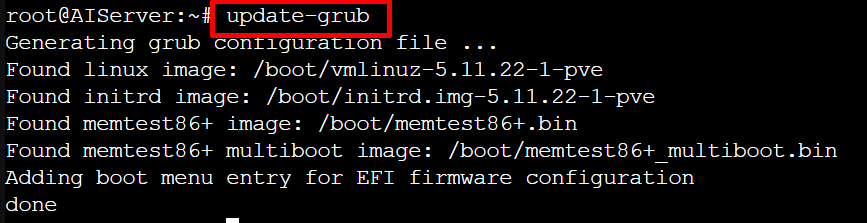
AI Server Terminal:

輸入nano /etc/default/grub

修改

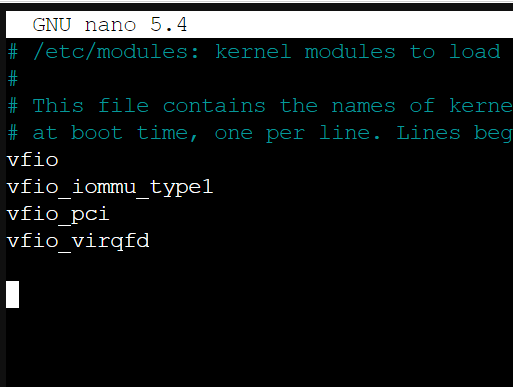


更新



Terminal 輸入 : nano /etc/modules

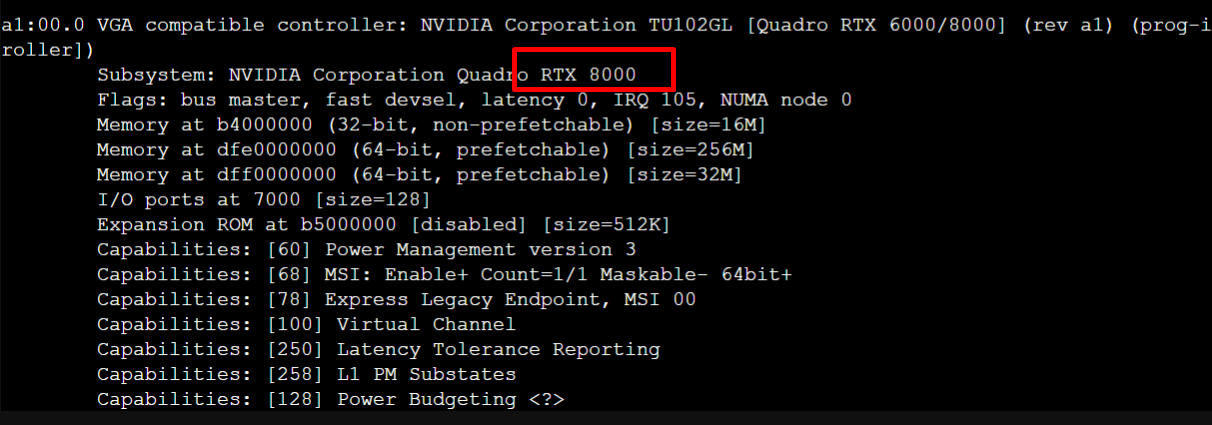
新增

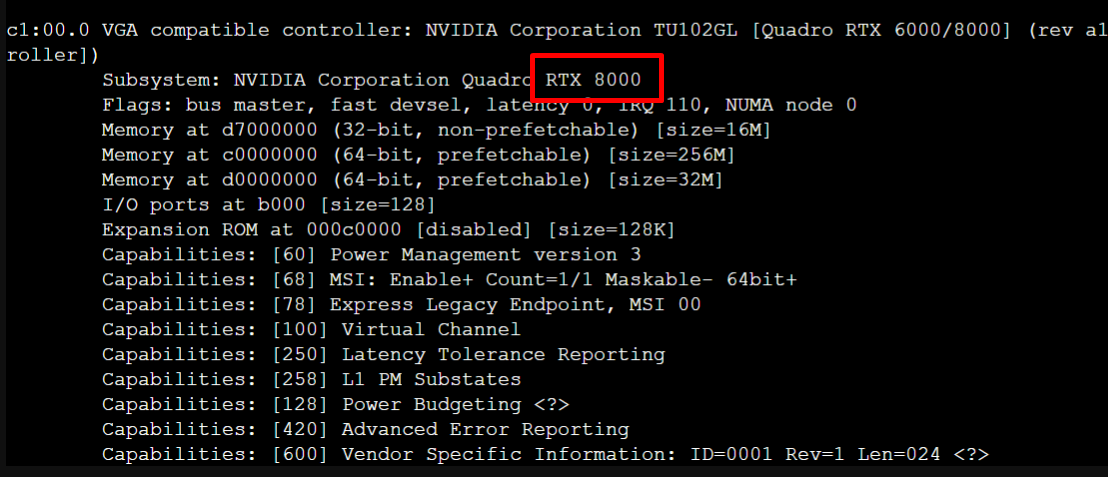


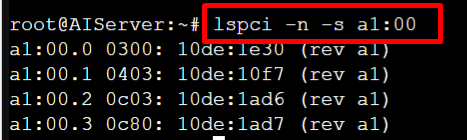
輸入

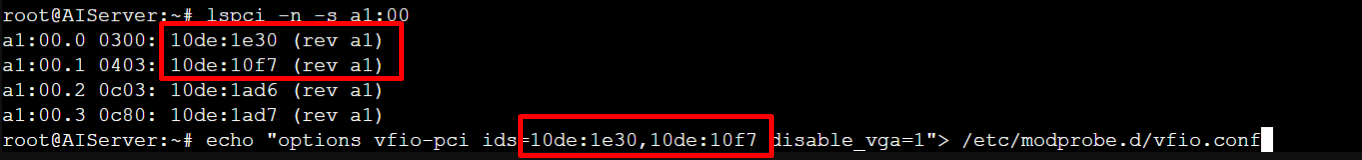
lspci –v

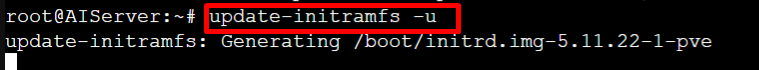
找到:



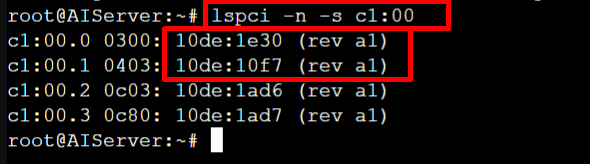








Pin passwd:

Sadmin123

**Windows 10 系統**

**參考網頁:** [**https://www.osslab.com.tw/proxmox-ve-pci-e-pass/**](https://www.osslab.com.tw/proxmox-ve-pci-e-pass/)

**1. 修改 /ETC/DEFAULT/GRUB**

**登入PVE的CLI模式，然後輸入以下指令 (這裡以 nano 文字編輯器為例，您可以使用其他編輯器):**

**nano /etc/default/grub**

**找到以下的字串:**

**GRUB\_CMDLINE\_LINUX\_DEFAULT="quiet"**

**Copy**

**然後改成以下的字串:**

**若是Intel CPU則修改如下:**

**GRUB\_CMDLINE\_LINUX\_DEFAULT="quiet intel\_iommu=on"**

**update-grub**

**2. 修改 /ETC/MODULES**

**接下來，然後輸入以下指令:**

**nano /etc/modules**

**然後輸入以下的字串:**

**vfio**

**vfio\_iommu\_type1**

**vfio\_pci**

**vfio\_virqfd**

**PVE虛擬機下的驅動程式，能夠正確無誤地取得GPU的資源。**

**3. 修改IOMMU岔斷之重新映射表**

**除了上述的修改，記得輸入以下的字串，讓IOMMU與VFIO運作正常**

**echo "options vfio\_iommu\_type1 allow\_unsafe\_interrupts=1" > /etc/modprobe.d/iommu\_unsafe\_interrupts.conf**

**echo "options kvm ignore\_msrs=1" > /etc/modprobe.d/kvm.conf**

**4. 將不必要的驅動程式設成給黑名單**

**由於我們不希望Proxmox主系統去使用到GPU的任何資源，因此要將那些會初始化或動用到GPU的各項驅動程式給”黑名單”化，讓他們不要被載入系統。請輸入以下的指令來禁用這些GPU驅動程式:**

**echo "blacklist radeon" >> /etc/modprobe.d/blacklist.conf**

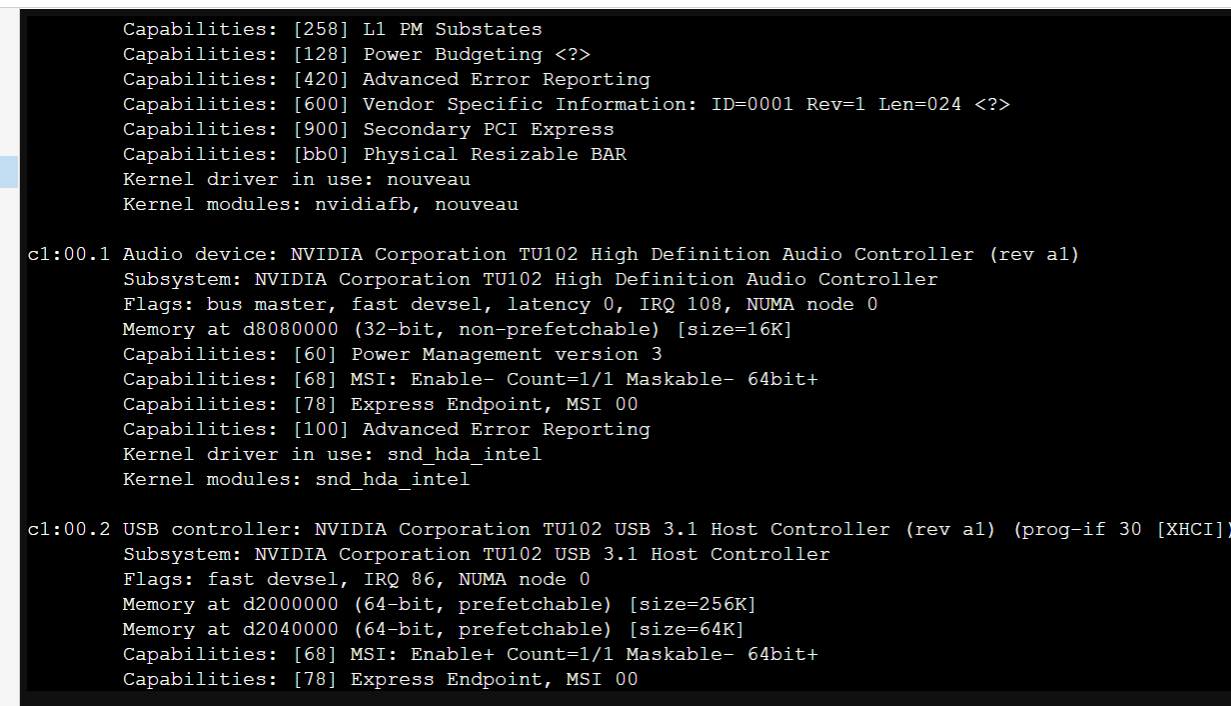
**echo "blacklist nouveau" >> /etc/modprobe.d/blacklist.conf**

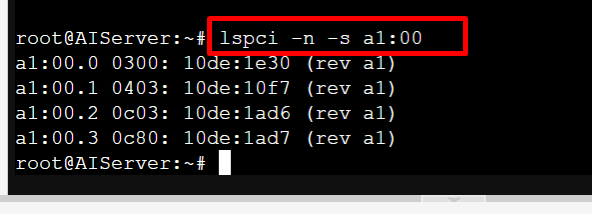
**echo "blacklist nvidia" >> /etc/modprobe.d/blacklist.conf**

**5. 將GPU指派給VFIO**

**這個步驟可說是最重要的步驟之一，必須將GPU的PCI ID指派給VM來使用。請輸入以下指令：**

**lspci –v**





**echo "options vfio-pci ids=10de:le30,10de:10f7 disable\_vga=1"> /etc/modprobe.d/vfio.conf**

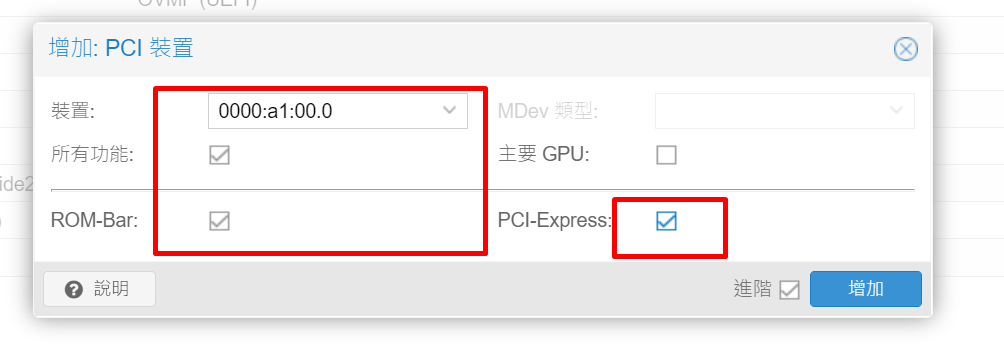
**以上的指令就是告訴VFIO去存取上述的Video controller和Audio controller，讓VM環境底下也能猶如Host端那樣，可以直接存取GPU資源，且讓GPU驅動程式安裝順利。對了，最後再記得輸入以下的指令:**

**update-initramfs -u**

**Copy**

**然後再輸入 restart 或 reboot 來重新開機，這樣您的Proxmox就順利打通PCI直通GPU的任督二脈!**

**\*建立vm**



**進入**

**nano /etc/pve/qemu-server/(vm名稱).conf**

**然後檢查是否以下的參數:**

**machine: q35**

**cpu: host,hidden=1,flags=+pcid**

**args: -cpu 'host,+kvm\_pv\_unhalt,+kvm\_pv\_eoi,hv\_vendor\_id=NV43FIX,kvm=off'**

**Step 3: 安裝作業系統與GPU驅動程式**

**剛剛建立好的VM，現在終於可以開機並安裝Windows 10 (或Ubuntu)了。一般來說，只要上述的步驟都完成之後，即可順利在VM下安裝GPU驅動程式，並執行各式GPU或是GPGPU的應用程式 (包含Direct3D、OpenGL、Vulkan、OpenCL、CUDA、Optix等API)。**

**安裝完作業系統之後，Windows 10使用者一般只要安裝好NVIDIA或AMD的顯示卡驅動程式即可。至於Ubuntu的使用者，則可透過額外驅動程式，來換裝NVIDIA或AMD的驅動程式，必要時才去顯示卡官網下載Linux的Driver。**

**注意:若是使用圖一張顯卡，不能同時開機，一次只能一台開機。**

**\*系統:CentOs7**

安裝前要先加入pci



選擇顯卡，及勾選**所有功能、ROM-Bar及PCI-Express**

安裝Centos7

安裝完成，先設定GPU通透

參考網址:

<https://blog.acesheep.com/index.php/archives/720/>

1. 安裝KVM 套件

* 安裝基礎工具
* 安裝epel-release 源
* 安裝 KVM 套件
* 添加virtio-win 源並安裝 virtio-win

yum install net-tools vim wget kernel-devel

yum update

yum install epel-release htop

yum install virt-manager qemu-img qemu-kvm qemu-kvm-tools libvirt virt-install bridge-utils wget https://fedorapeople.org/groups/virt/virtio-win/virtio-win.repo -O /etc/yum.repos.d/virtio-win.repo

yum install virtio-win

禁用nouveau 驅動:

echo "blacklist radeon" >> /etc/modprobe.d/blacklist.conf

echo "blacklist nouveau" >> /etc/modprobe.d/blacklist.conf

echo "blacklist nvidia" >> /etc/modprobe.d/blacklist.conf

必須雙重確認:

打開vim /usr/lib/modprobe.d/dist-blacklist.conf

#在文件最後加上

blacklist nouveau

options nouveau modeset=0

在 blacklist nvidiafb前面加#

#備份鏡像

mv /boot/initramfs-$(uname -r).img /boot/initramfs-$(uname -r).img.bak

#重建鏡像

dracut -v /boot/initramfs-$(uname -r).img $(uname -r)

#重新開機

reboot

#確認nouveau關閉

lsmod | grep nouveau

#開啟iommu (在quiet後面加上 intel\_iommu=on)

vim /etc/default/grub

GRUB\_CMDLINE\_LINUX\_DEFAULT="... intel\_iommu=on ..."

或

GRUB\_CMDLINE\_LINUX="... intel\_iommu=on ..."

#更新

grub2-mkconfig -o /boot/grub2/grub.cfg

#重新開機

reboot

#驗證

dmesg |grep -e DMAR -e IOMMU

\*若發現dmesg command not found，就輸入yum install util-linux下載

確認lspci | grep NVIDIA

有沒有出現顯卡資訊



\*若出現lspci command not found，下載yum install –y pciutils

\*若是原先安裝文字系統，要安裝圖形介面，

輸入:yum groupintall "X Window System"

輸入:yum groupinstall "GNOME Desktop"

安裝完成可以用startx命令進入圖形介面

Systemctl set-default multi-user.target文字模式

Systemctl set-default graphical.target圖形介面模式

\*Centos8安裝 :  
請輸入dnf group list(確認有哪些可安裝)

接著輸入dnf groupinstall "Server with GUI" -y

圖形介面下:

先至nvidia官方網站下載顯卡對應之驅動程式，存放在Download資料夾底下

進入文字模式安裝

在安裝之前要先安裝gcc

yum install –y gcc

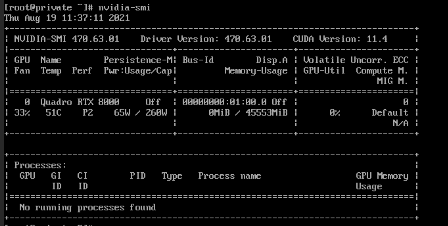
安裝驅動程式

切換到Download資料夾底下

輸入:sh NVIDIA-Linux-x86-64\_470.63.01.run

安裝完成

輸入nvidia-smi



出現顯卡資訊

Centos8遇到問題:



解決方法:

參考: <https://medium.com/@alexanderyegorov_67403/how-to-compile-kernel-module-for-centos8-78287e9d145a>

yum groupinstall “Development Tools”

yum install -y ncurses-devel

yum install -y hmaccalc zlib-devel binutils-devel elfutils-libelf-devel

＊修改節點名稱：

網頁參考<http://www.gienginali.idv.tw/modules/tad_book3/page.php?tbsn=2&tbdsn=1056>

nano /etc/hostname

進去後將名稱改為新名稱

nano /etc/hosts

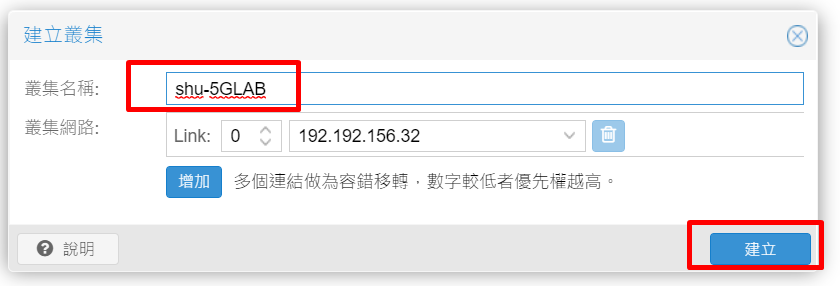
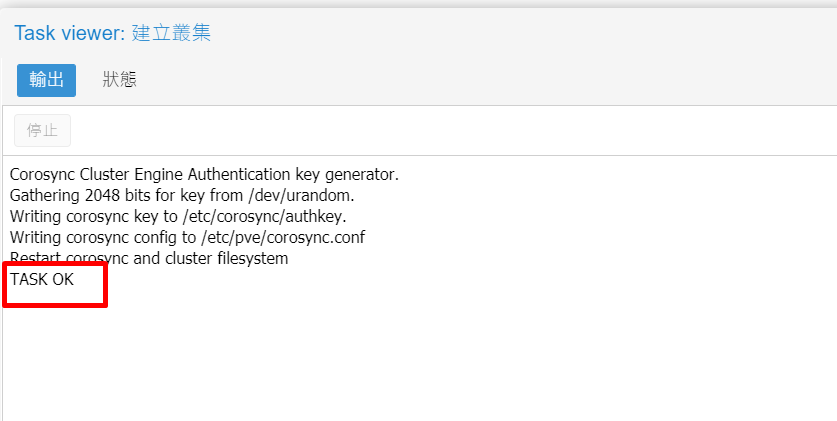
10.0.0.205 新名稱.example.com 新名稱 pvelocalhost

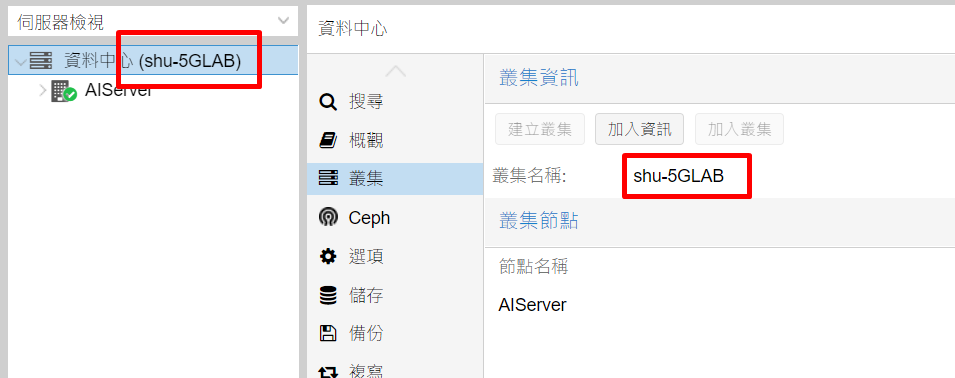
重新開機之後移除舊的名稱資料夾:

rm -rf /etc/pve/nodes/舊名稱/

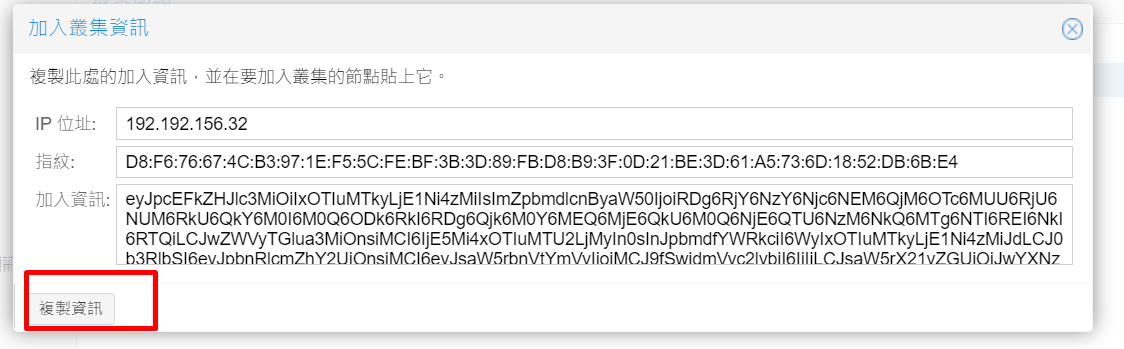
完成之後把備份的設定檔倒回原位置即可正常使用，有簽發SSL的起重新簽發。

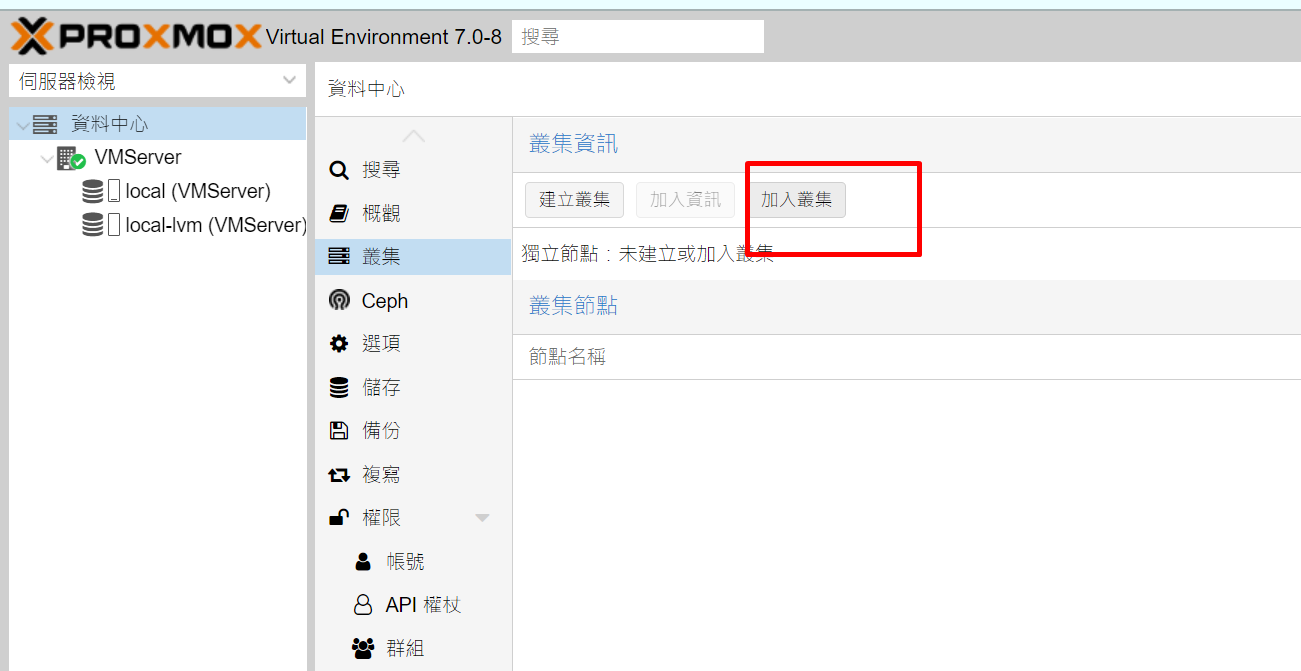
acme.sh --issue --standalone --keypath /etc/pve/local/pveproxy-ssl.key --fullchainpath /etc/pve/local/pveproxy-ssl.pem --

＊建立叢集：

重新整理頁面出現叢集名稱







至另外一台貼上



\*刪除叢集

參考網址:

<https://pvecli.xuan2host.com/proxmox-cluster-delnode-fail-fix/>

**service pve-cluster stop**

**pmxcfs -l**

**rm /etc/pve/corosync.conf**

**rm /etc/corosync/corosync.conf**

**rm /etc/corosync/authkey**

**rm -rf /etc/pve/nodes/其他主機名稱**

**init 6**