Run Proxmox on OCI - FOR FREE

Last updated on Dec 13, 2023

Introduction

This article will walk you through the steps that I took to get Proxmox running inside of an OCI instance. Because you cannot VNC directly into an instance that is booting, this will be a great way to run services that may require some sort of connection other than RDP (Windows Server) or SSH.

In my use case, I was looking to deploy a virtual PBX to a cloud environment so all of our locations did not need to rely on our on premises server. You can imagine the chaos when the phone system goes down.

In this method, we are going to be using a custom Debian image to create a boot image on OCI and create a virtual instance. Then, once that is set up we will be manually installing Proxmox.

Subscribe

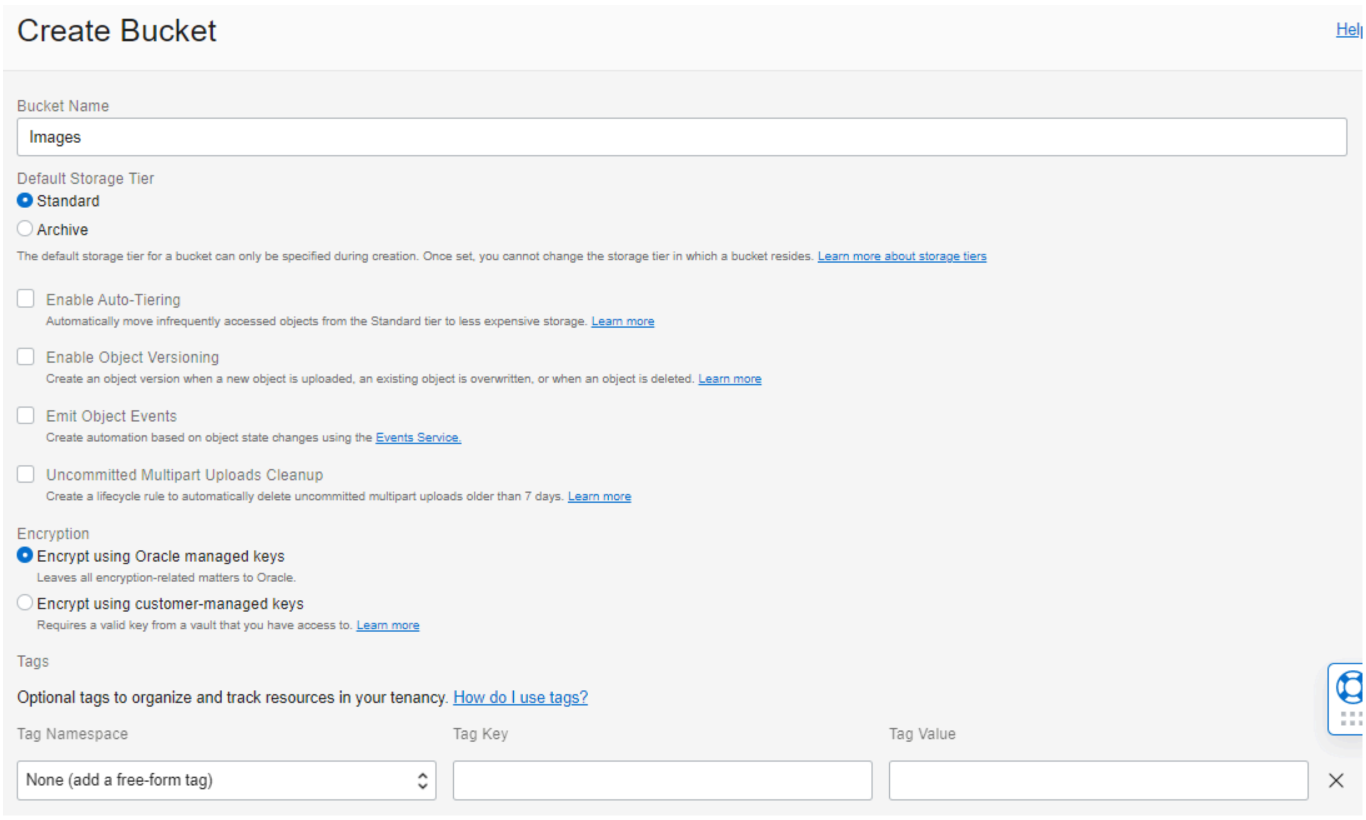
Step 1: Download Required Files

We need to download the Debian image required first. At the time of writing Proxmox is using Debian 11 (bullseye). Lucky for us Debian provides ready-to-deploy cloud images. The image that you will need can be download directly from them here: https://cloud.debian.org/images/cloud/

The version that I used was `debian-11-genericcloud-amd64-20221205-1220.qcow2` . That image can be downloaded directly here: https://cloud.debian.org/images/cloud/bullseye/20221205-1220/debian-11-genericcloud-amd64-20221205-1220.qcow2

Step 2: Create OCI Bucket

Next we will need to start getting our image ready for OCI. Once logged into your OCI Dashboard you will need to navigate to Storage -> Buckets. If you don't already have one, create a new bucket for your custom images by clicking "Create New". These are the options I used:


Create Bucket

Bucket Name
Images

Default Storage Tier
[✓ Standard]
Archive
The default storage tier for a bucket can only be specified during creation. Once set, you cannot change the storage tier in which a bucket resides. Learn more about storage tiers

☐ Enable Auto-Tiering
Automatically move infrequently accessed objects from the Standard tier to less expensive storage. Learn more

☐ Enable Object Versioning
Create an object version when a new object is uploaded, an existing object is overwritten, or when an object is deleted. Learn more

☐ Emit Object Events
Create automation based on object state changes using the Events Service.

☐ Uncommitted Multipart Uploads Cleanup
Create a lifecycle rule to automatically delete uncommitted multipart uploads older than 7 days. Learn more

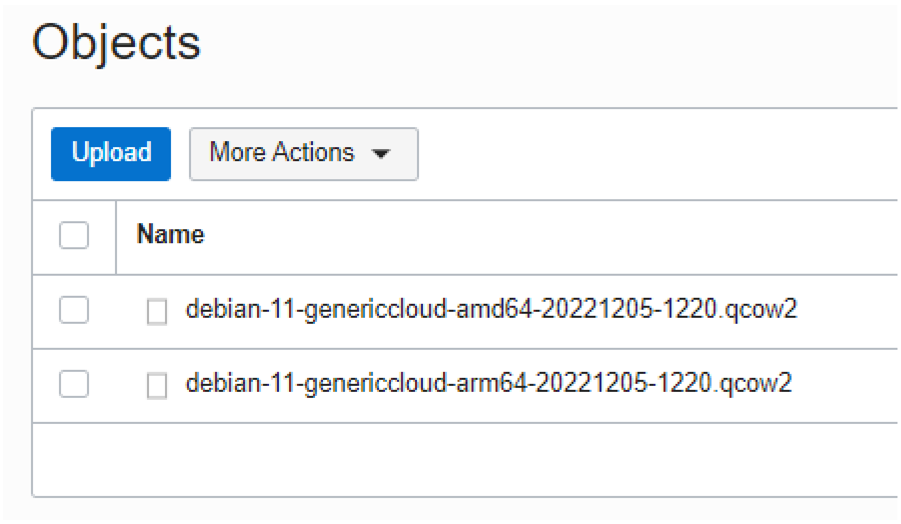
Encryption
[✓ Encrypt using Oracle managed keys]
Leaves all encryption-related matters to Oracle.
Encrypt using customer-managed keys
Requires a valid key from a vault that you have access to. Learn more

Tags
Optional tags to organize and track resources in your tenancy. How do I use tags?

Tag Namespace	Tag Key	Tag Value
None (add a free-form tag)		

Step 3: Upload Debian to Bucket

Now that you have a bucket created, we need to click on it to open it up. Once you are in your bucket, scroll down and choose “Upload” under objects.

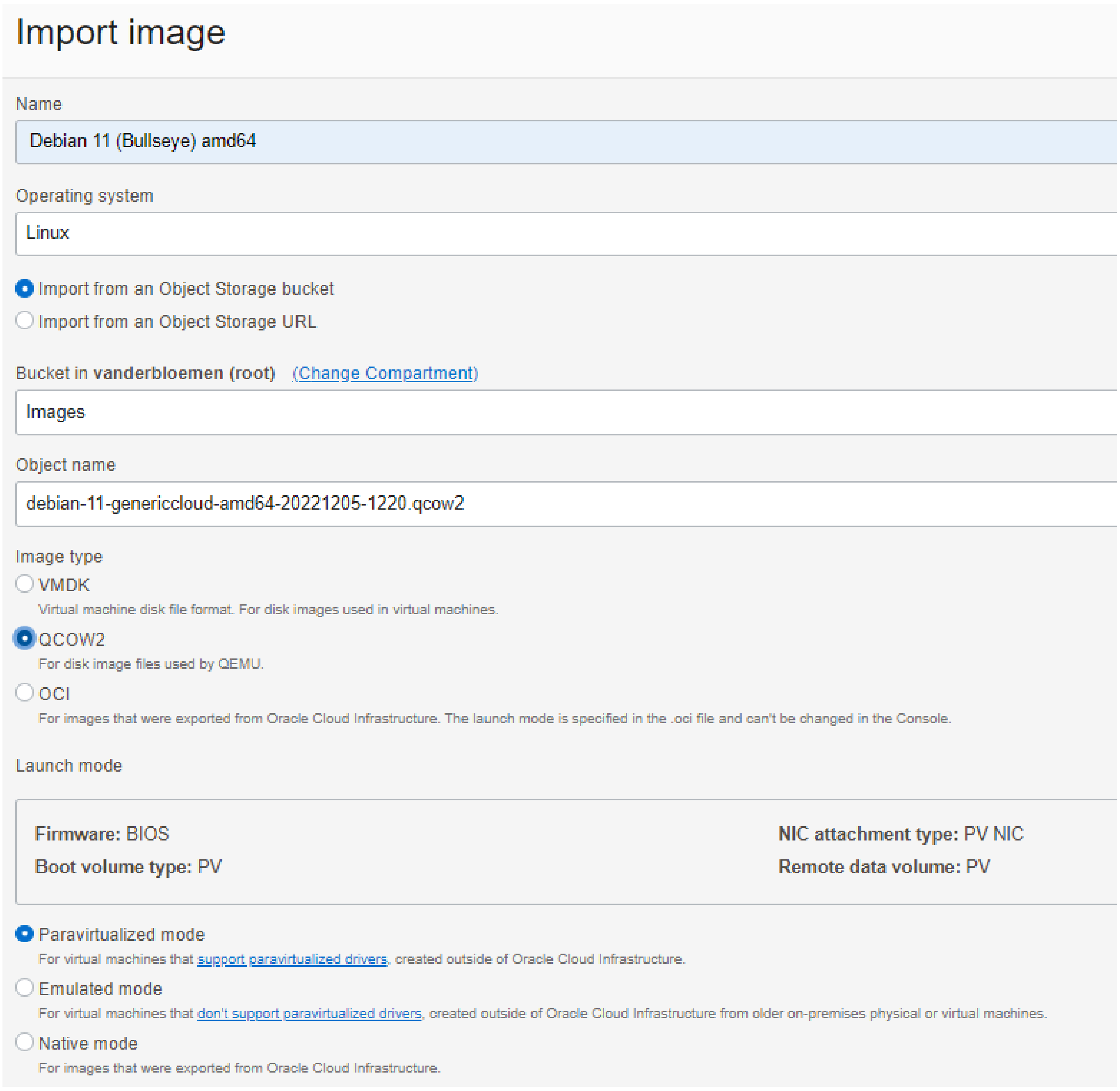


debian-11-genericcloud-amd64-20221205-1220.qcow2
debian-11-genericcloud-arm64-20221205-1220.qcow2

Upload the Debian image from Step 1. You can name it whatever you like.

Step 4: Make Custom Image

With the Debian image uploaded, it is time to convert that to an Oracle Cloud Image. To do this Navigate to Compute -> Custom Images. Once there choose “Import Image”. This is an example of the import settings you should emulate:



Import image

Name: Debian 11 (Bullseye) amd64

Operating system: Linux

[✓ Import from an Object Storage bucket] Import from an Object Storage URL

Bucket in **vanderbloemen (root)** (Change Compartment): Images

Object name: debian-11-genericcloud-amd64-20221205-1220.qcow2

Image type

VMDK
Virtual machine disk file format. For disk images used in virtual machines.

[✓ QCOW2]
For disk image files used by QEMU.

OCI
For images that were exported from Oracle Cloud Infrastructure. The launch mode is specified in the .oci file and can't be changed in the Console.

Launch mode

Firmware: BIOS
Boot volume type: PV
NIC attachment type: PV NIC
Remote data volume: PV

[✓ Paravirtualized mode]
For virtual machines that support paravirtualized drivers, created outside of Oracle Cloud Infrastructure.

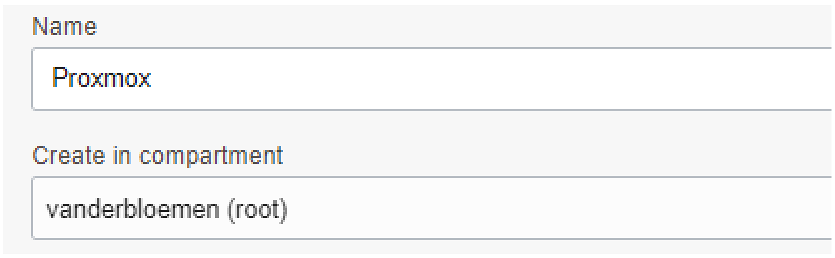
Emulated mode
For virtual machines that don't support paravirtualized drivers, created outside of Oracle Cloud Infrastructure from older on-premises physical or virtual machines.

Native mode
For images that were exported from Oracle Cloud Infrastructure.

This will take a few mins to provision.

Step 5: Deploy Custom Image

Now that our Custom Image has been provisioned, we can now create our custom image. Navigate to Compute -> Instances and choose “Create Instance”. On the “Create compute instance” page start off by creating a name for your Proxmox instance. I called mine “Proxmox”.

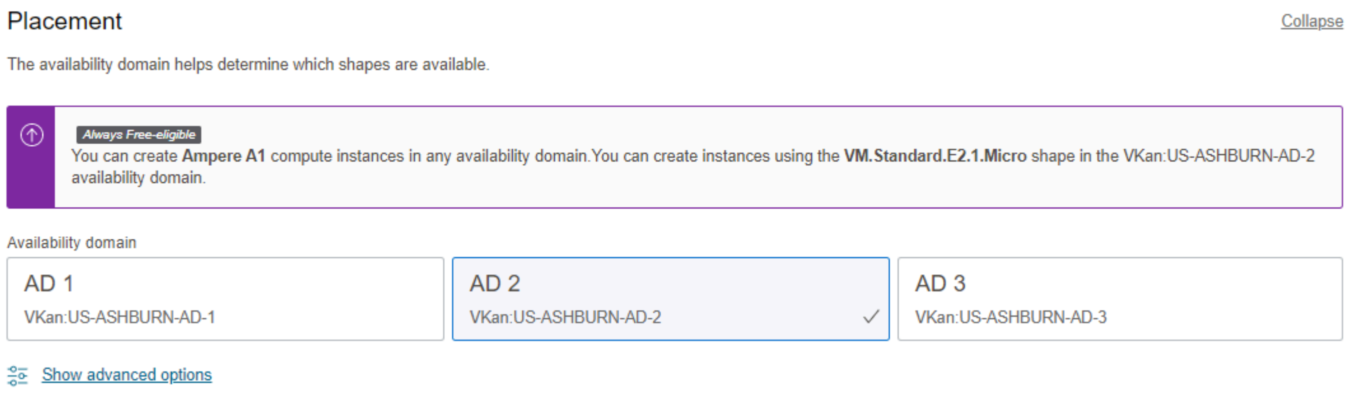

Name

Proxmox

Create in compartment

vanderbloemen (root)

Then choose your Availability Domain


Placement

The availability domain helps determine which shapes are available.

Always Free-eligible

You can create **Ampere A1** compute instances in any availability domain.You can create instances using the **VM.Standard.E2.1.Micro** shape in the VKan:US-ASHBURN-AD-2 availability domain.

Availability domain

AD 1	AD 2 ✓	AD 3
VKan:US-ASHBURN-AD-1	VKan:US-ASHBURN-AD-2	VKan:US-ASHBURN-AD-3

Then When it comes time to choose your Image make sure to choose “Custom Images” and pick the Debian one that we created. Next, choose your shape. You CAN run this on OCI’s “Always Free-eligible” Tier if you’d like.

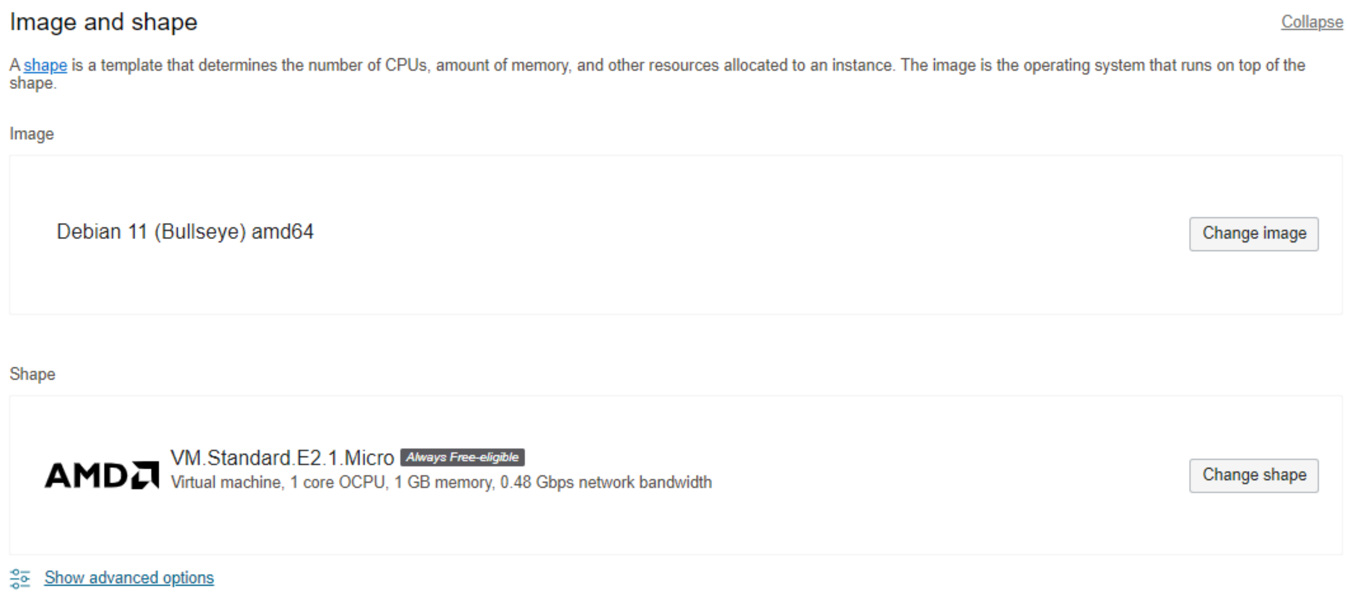


Image and shape

A shape is a template that determines the number of CPUs, amount of memory, and other resources allocated to an instance. The image is the operating system that runs on top of the shape.

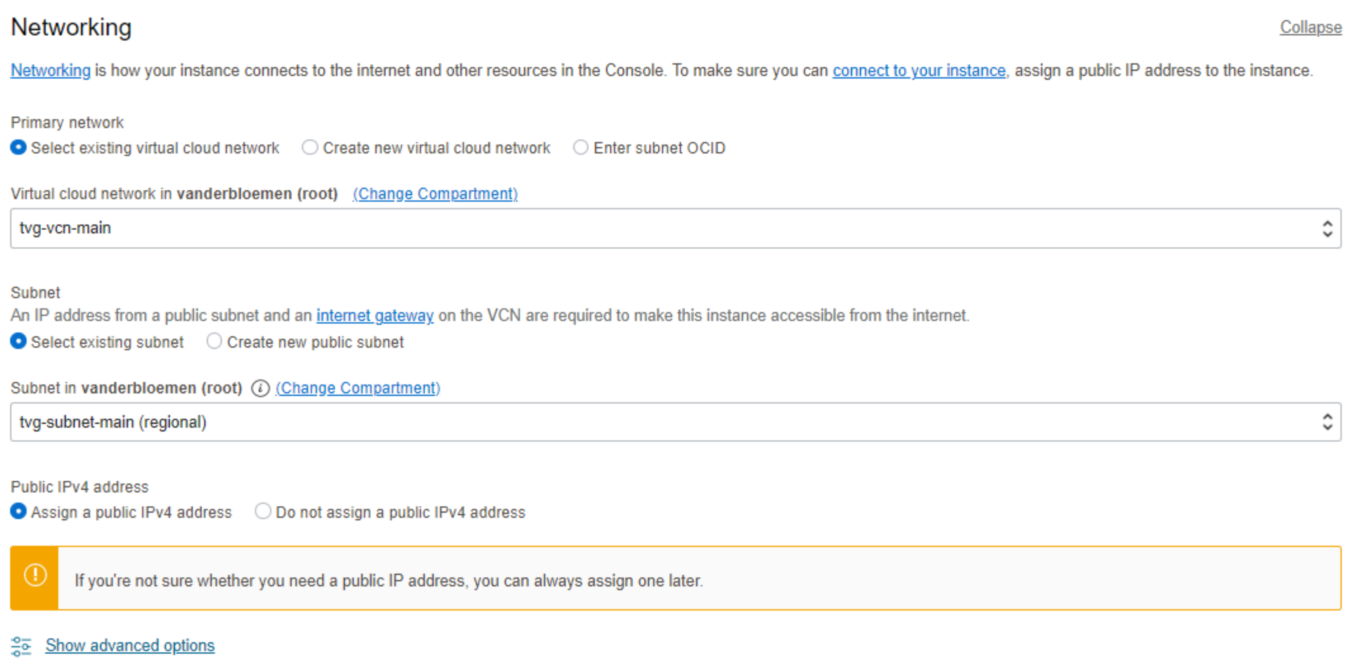
Image

Debian 11 (Bullseye) amd64

Shape

VM.Standard.E2.1.Micro Always Free-eligible
Virtual machine, 1 core OCPU, 1 GB memory, 0.48 Gbps network bandwidth

(Important Note Here) If you would like to run this on Ampere, you can do this by using all the previous steps but using the Debian arm64 Image! Next, select all your networking settings. You can make a separate subnet for this Proxmox machine if you’d like but it’s not necessary.


Networking

Collapse

Networking is how your instance connects to the internet and other resources in the Console. To make sure you can connect to your instance, assign a public IP address to the instance.

Primary network
[✓ Select existing virtual cloud network] Create new virtual cloud network Enter subnet OCID

Virtual cloud network in **vanderbloemen (root)** (Change Compartment)
tvg-vcn-main

Subnet
An IP address from a public subnet and an internet gateway on the VCN are required to make this instance accessible from the internet.
[✓ Select existing subnet] Create new public subnet

Subnet in **vanderbloemen (root)** (Change Compartment)
tvg-subnet-main (regional)

Public IPv4 address
[✓ Assign a public IPv4 address] Do not assign a public IPv4 address

If you're not sure whether you need a public IP address, you can always assign one later.

Show advanced options

Finally, make sure to download your SSH Keys. This is what will allow you to SSH into your machine once it's up and running! Go ahead and Choose "Create" to start up your new Debian image. Wait for OCI to finish provisioning your machine and move on to step 6.

Step 6: Installing Proxmox

Using your instance's public IP address and the key that you downloaded, SSH into your instance. On Windows I use MobaXterm, regular SSH on macOS or Linux is perfectly fine. From here we will run the required commands in order to prep and install Proxmox.

We Will start by making sure the OS is up-to-date

```
sudo apt update && sudo apt upgrade
```

Now we begin to install Proxmox manually.

💡 Most of this process can be found on Proxmox's documentation site here! Make sure to visit that page if you have any issues.

Because this image is set up for a cloud environment, we need to edit the cloud.cfg first. We **WILL NOT** be adjusting the hosts file directly.

`sudo nano /etc/cloud/cloud.cfg`

Make sure to set `preserve_hostname` to true.

```
preserve_hostname: true
```

Then below that line add:

```
hostname: proxmox
```

Now we need to edit the cloud master file found at `/etc/cloud/templates/hosts.debian.tmpl` .

`sudo nano /etc/cloud/templates/hosts.debian.tmpl`

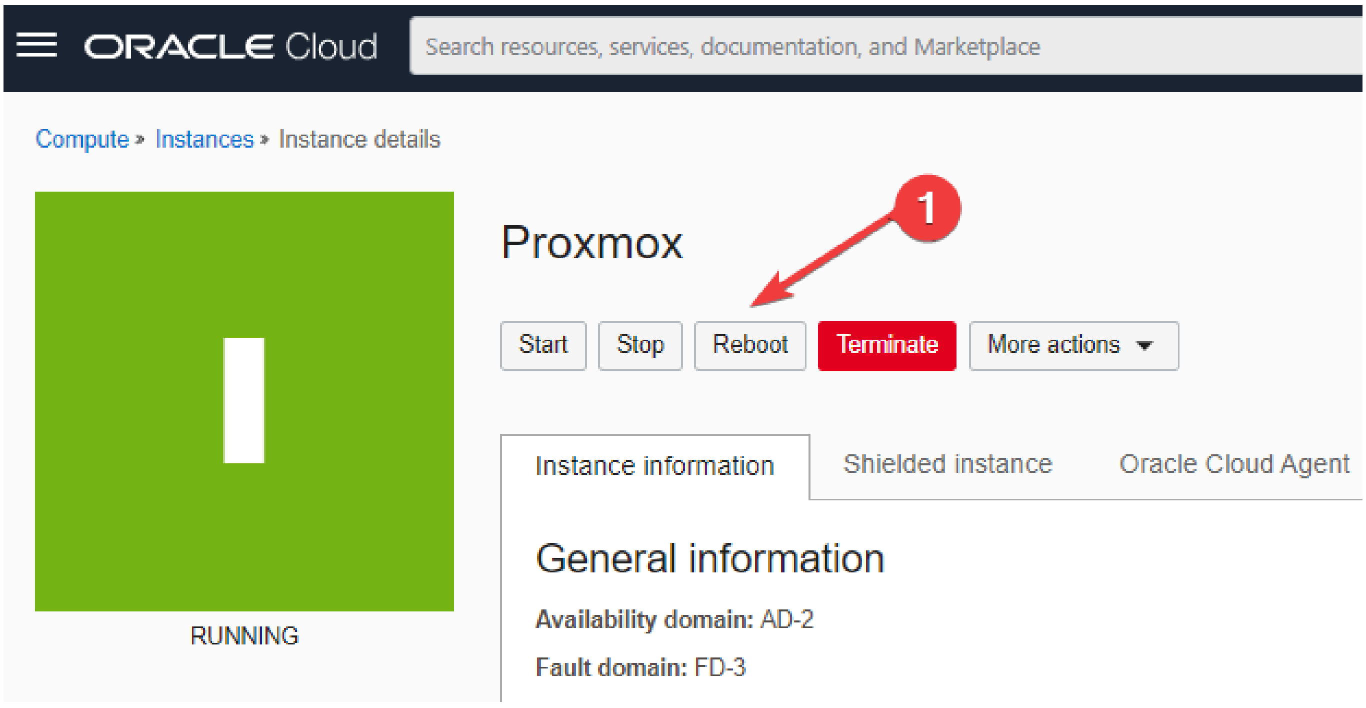
By default, my hosts file looked like this:

```
127.0.1.1
127.0.0.1 localhost
```

I changed it to look like this:

```
129.xx.xxx.xxx proxmox.mydomainname.com proxmox
```

Then once that is finished you will need to reboot your instance from the OCI dashboard.



Once the Reboot has finished, SSH back into your instance and verify the change has been made with this command:

```
hostname --ip-address
```

Your output should look like this:

```
129.xx.xxx.xxx
```

Now we need to add the Proxmox VE repo:

```
sudo nano /etc/apt/sources.list.d/pve-install-repo.list
```

Then add this line:

```
deb [arch=amd64] http://download.proxmox.com/debian/pve bullseye pve-no-subscrip
```

Add the Proxmox VE repository key as sudo:

```
sudo wget https://enterprise.proxmox.com/debian/proxmox-release-bullseye.gpg -O /etc/apt/trusted.gpg.d/proxmox-release-bullseye.gpg
```

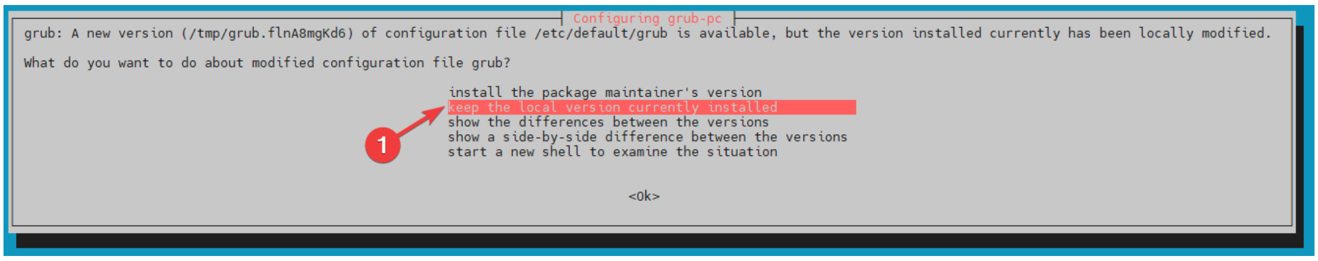
Now update and upgrade your instance:

```
sudo apt update && sudo apt upgrade
```

Next we need to install the Proxmox VE kernel:

```
sudo apt install pve-kernel-5.15
```

> If apt asks if you would like to install the new version of grub, be sure to select "Keep the local version currently installed". We will be manually updating grub later.



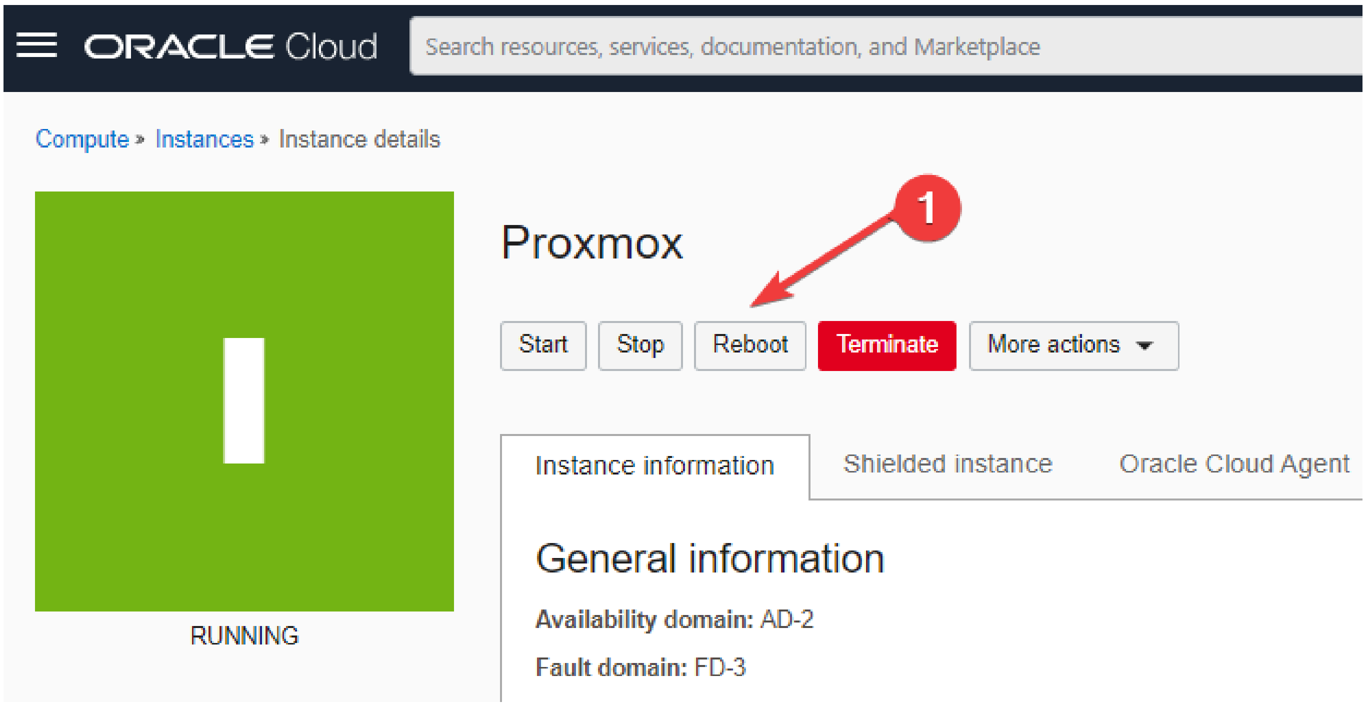
```
Configuring grub-pc
grub: A new version (/tmp/grub.flnA8mgKd6) of configuration file /etc/default/grub is available, but the version installed currently has been locally modified.

What do you want to do about modified configuration file grub?

                        install the package maintainer's version
                        keep the local version currently installed
                        show the differences between the versions
                        show a side-by-side difference between the versions
                        start a new shell to examine the situation

                                        <Ok>
```

Once the Kernel has finished installing make sure to reboot the system using OCI dashboard.



General information

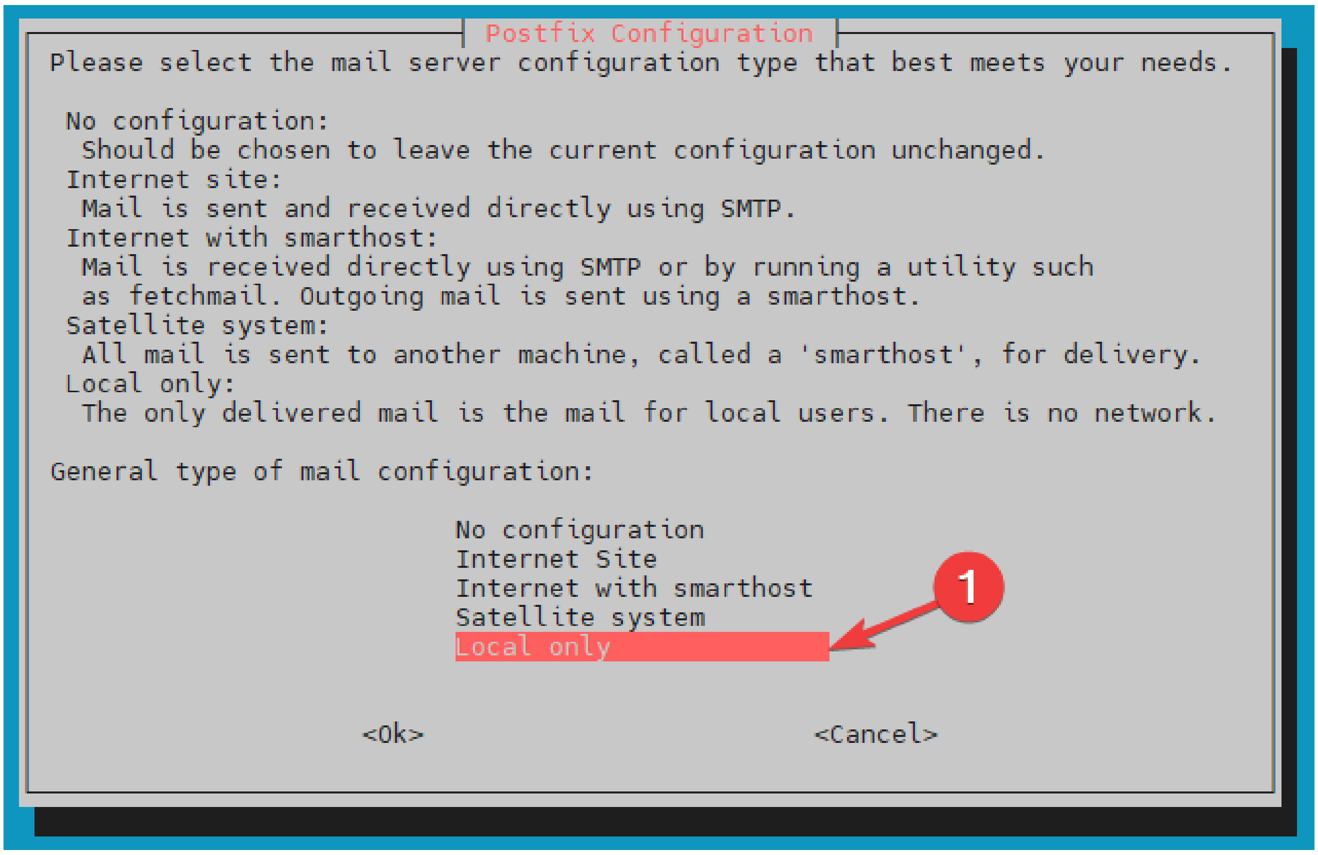
Availability domain: AD-2

Fault domain: FD-3

Once the system has rebooted, SSH back into your instance and you can Install the Proxmox VE packages.

```
sudo apt install proxmox-ve postfix open-iscsi
```

> Choose "Local only" for postfix when prompted.



```
Postfix Configuration
Please select the mail server configuration type that best meets your needs.

 No configuration:
  Should be chosen to leave the current configuration unchanged.
 Internet site:
  Mail is sent and received directly using SMTP.
 Internet with smarthost:
  Mail is received directly using SMTP or by running a utility such
  as fetchmail. Outgoing mail is sent using a smarthost.
 Satellite system:
  All mail is sent to another machine, called a 'smarthost', for delivery.
 Local only:
  The only delivered mail is the mail for local users. There is no network.

General type of mail configuration:

                          No configuration
                          Internet Site
                          Internet with smarthost
                          Satellite system
                          Local only

                    <Ok>                         <Cancel>
```

After all the packages have been installed, it is time to remove the previous Debian Kernel and update Grub:

```
apt remove linux-image-amd64 'linux-image-5.10*'
```

❗ Apt will warn you that removing your kernel is bad. Normally this is true, but the Proxmox packages we installed ship with its own Kernel and we will update Grub to use that one in the next step.

Update Grub manually to use the Kernal installed by Proxmox:

```
sudo update-grub
```

Verify that os-prober package is **NOT** installed:

```
sudo apt remove os-prober
```

Step 7: Configure Proxmox

To enable us to log into the Proxmox Web UI, we will need to set the root user password.

```
sudo passwd root
```

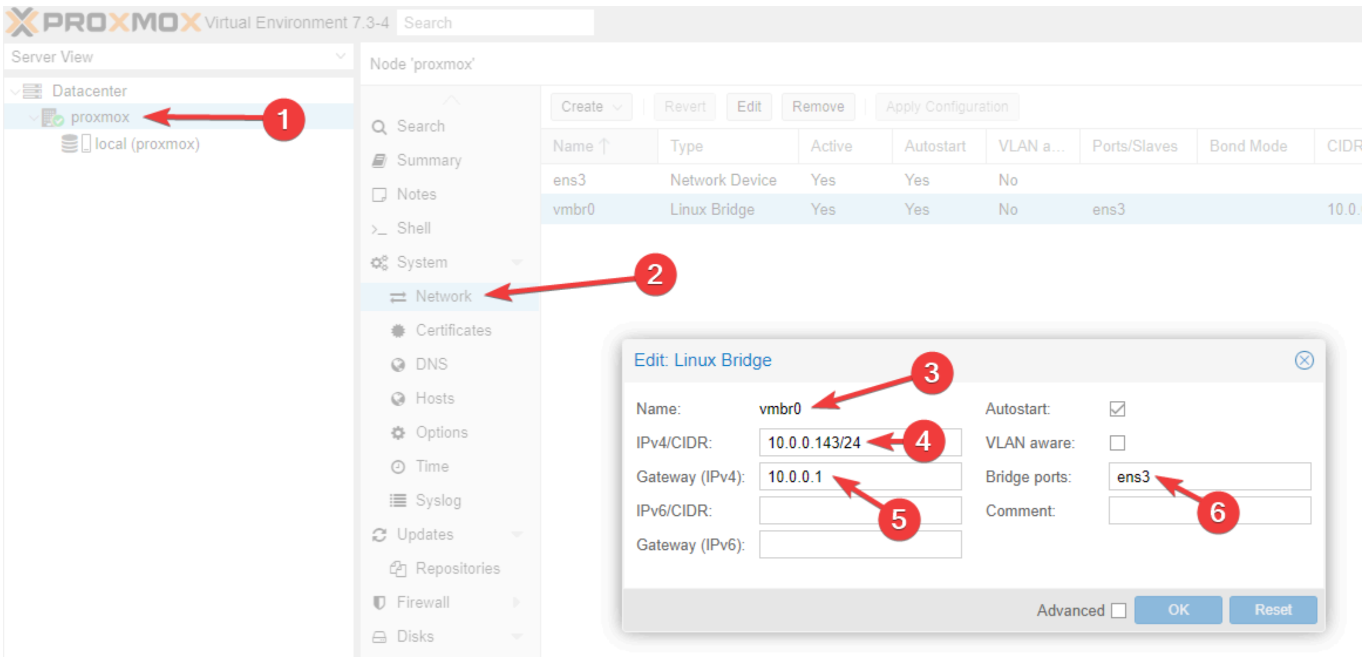
With the root password set, you can now log into your Proxmox Web UI:

```
https://youriphere:8006
```

⚠️ The next time that your Proxmox instance reboots, it will apply some network changes that will break your ability for the machine to connect to the internet. There is a chance that this can break any time there is a Kernel update but it is unlikely. Following are the steps to fix this.

In the Proxmox Web UI, go to your machine's network settings. You will notice it is asking you to create the VM Bridge network since it does not create that by default. We are going to create this network now!

Fill in the sections below to match the Virtual Bridge to you OCI dashboard. Pictured below are my settings:



Name	Type	Active	Autostart	VLAN a...	Ports/Slaves	Bond Mode	CIDR
ens3	Network Device	Yes	Yes	No			
vmbr0	Linux Bridge	Yes	Yes	No	ens3		10.0.

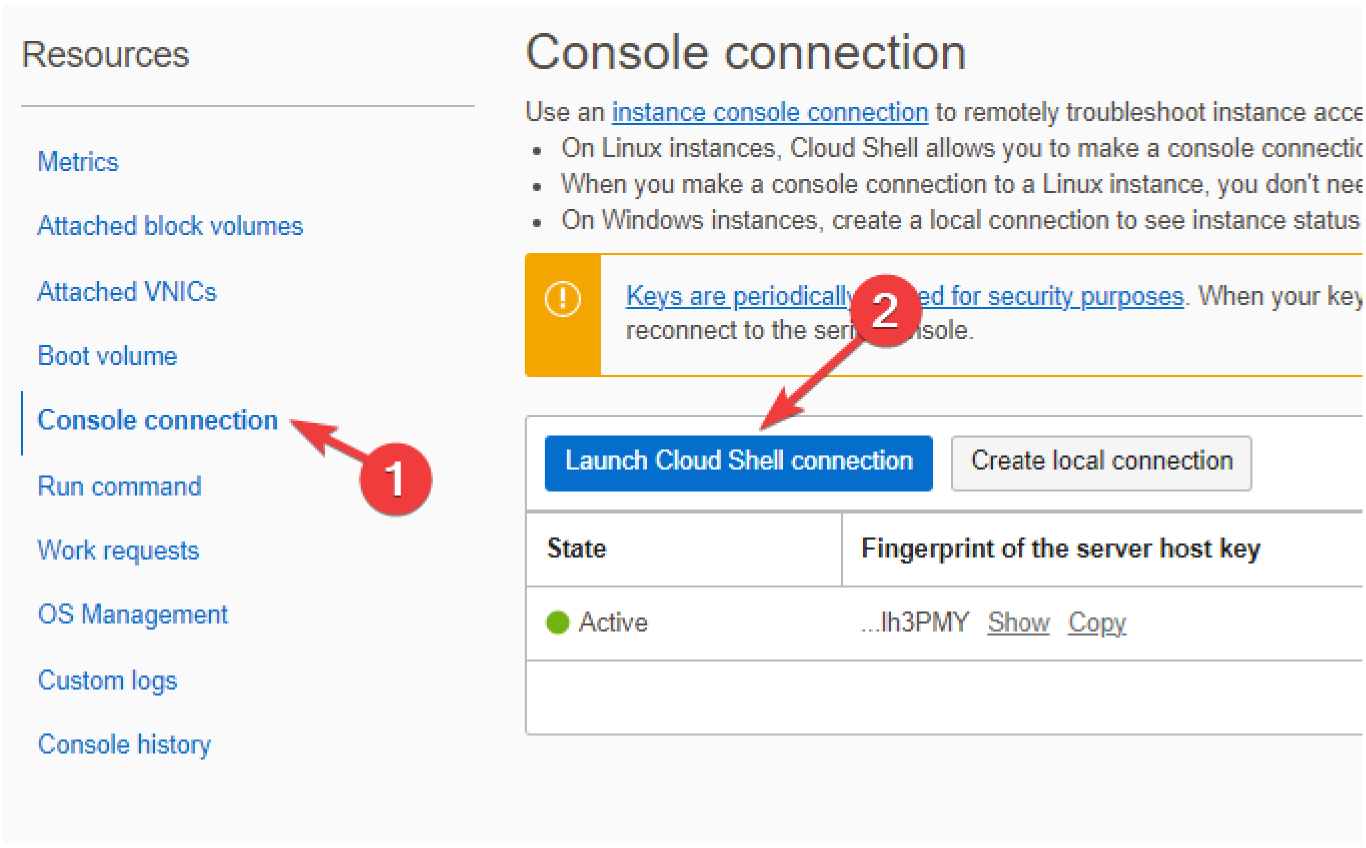
Name:	vmbr0	Autostart:	✓
IPv4/CIDR:	10.0.0.143/24	VLAN aware:	☐
Gateway (IPv4):	10.0.0.1	Bridge ports:	ens3
IPv6/CIDR:		Comment:	
Gateway (IPv6):			

Reboot the instance one more time using the OCI dashboard, **NOT THE PROXMOX WEB UI**, after these changes have been made. Once the reboot has finished, you will not be able to SSH to the instance OR access the WEB UI.

In the OCI Dashboard, select your Instance and open **“Console connection”**

Compute → **Instances** → **Instance details** → **Console connection**

Then click on **“Launch Cloud Shell Connection”**



Console connection

Use an instance console connection to remotely troubleshoot instance acce

- On Linux instances, Cloud Shell allows you to make a console connectic
- When you make a console connection to a Linux instance, you don't nee
- On Windows instances, create a local connection to see instance status

Keys are periodically ed for security purposes. When your key reconnect to the ser sole.

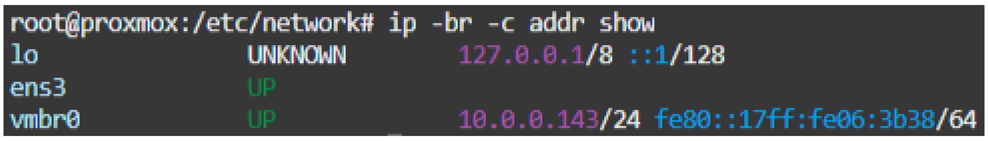
State	Fingerprint of the server host key
● Active	...lh3PMY Show Copy

It will take a few mins for the console to open, but once it does log in using the “root” user and the password you had created in **Step 7**.

Once logged in, get your interface names:

```
ip -br -c addr show
```

Make note of these. For me I had 3: `lo`, `ens3`, and `vmbr0`. You will notice that both `ens3` and `vmbr0` are reporting as down.



```
root@proxmox:/etc/network# ip -br -c addr show
lo               UNKNOWN        127.0.0.1/8 ::1/128
ens3             UP
vmbr0            UP             10.0.0.143/24 fe80::17ff:fe06:3b38/64
```

To fix this we will need to edit `/etc/network/interfaces`:

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

And then adjust your file to match this using your network information:

```
auto lo
iface lo inet loopback

auto ens3

iface ens3 inet manual

auto vmbr0
iface vmbr0 inet static
        address 10.0.0.143/24
        gateway 10.0.0.1
        bridge-ports ens3
        bridge-stp off
        bridge-fd 0

# source-directory /etc/network/interfaces.d
# source-directory /run/network/interfaces.d
```

Once you have saved your changes we can now up the interfaces.

```
ifup ens3
```

```
ifup vmbr0
```

Conclusion

After you have upped your interfaces, your Proxmox install is complete and can now be accessed from the Proxmox Web UI and SSH once again. From here you can make any adjustmets to Proxmox from the web UI. Thanks so much for reading!

Thanks for reading!

Written By: Max Kulik

Sources:

- Install Proxmox VE on Debian 11 Bullseye - https://pve.proxmox.com/wiki/Install_Proxmox_VE_on_Debian_11_Bullseye
- Root Password Reset - https://pve.proxmox.com/wiki/Root_Password_Reset
- Debian Cloud images: https://cloud.debian.org/images/cloud/bullseye/20221205-1220/
- Web UI Unreachable After Adding NVME Drive - https://forum.proxmox.com/threads/webui-unreachable-after-adding-nvme-drive.85500/
- Losing Network Connection after each Proxmox Reboot - https://forum.proxmox.com/threads/losing-network-connection-after-each-proxmox-reboot.108528/

PREVIOUS

Make Transmission WEB-UI Beautiful

NEXT

Install Nginx Proxy Manager on Proxmox LXC & Alpine Linux