# Building a VM with virt-manager

Virt-manager can manage a number of hypervisors, including qemu/KVM that I'll be covering.
Virt-manager can be run as a normal user but I've never done that. Bite the bullet and **sudo /bin/bash** (*supply your password*) or plain **su -** *(supply root password)*.

## Required Packages

You will need these top-level packages:-

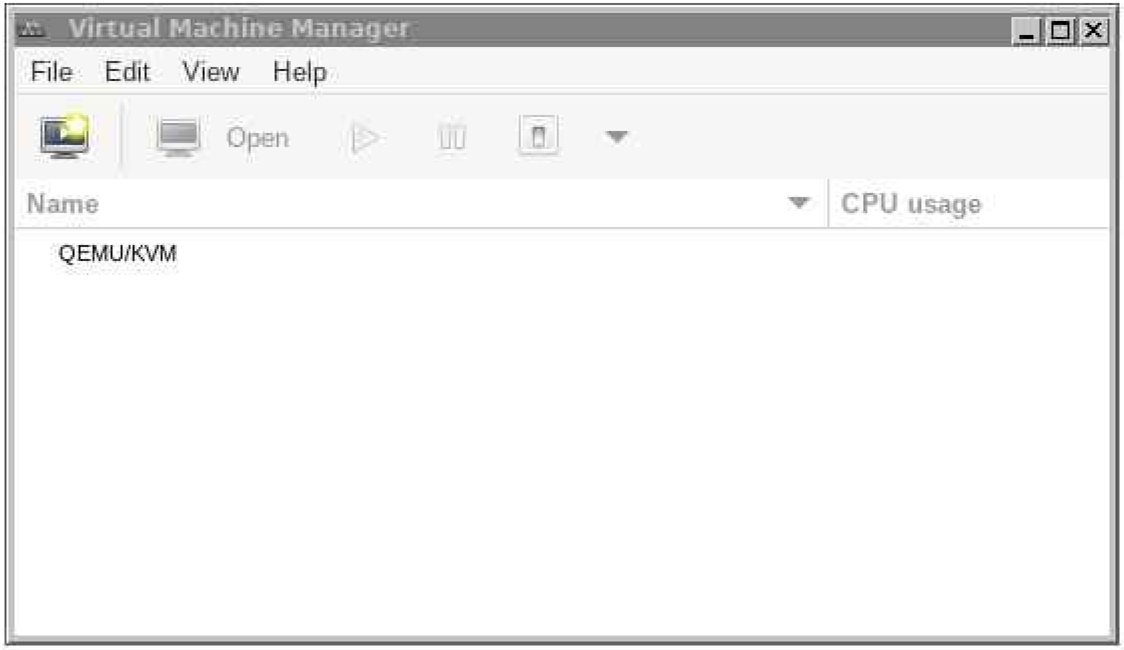
- virt-manager
- qemu
- virt-viewer (optional)
- KVM kernel modules (should load automatically if you have them)

## Start virt-manager

Before starting virt-manager, need to start the libvirt daemon **libvirtd**. (libvirt is a dependency of virt-manager):-
**/etc/rc.d/rc.libvirt start**
Also start the **default** network if necessary:-
**ip link show | grep -E -q virbr0 || virsh -d 0 net-start default**
After that in a root window do:-
**virt-manager >/dev/null 2>&1&**
(redirection may be tricky with sudo)

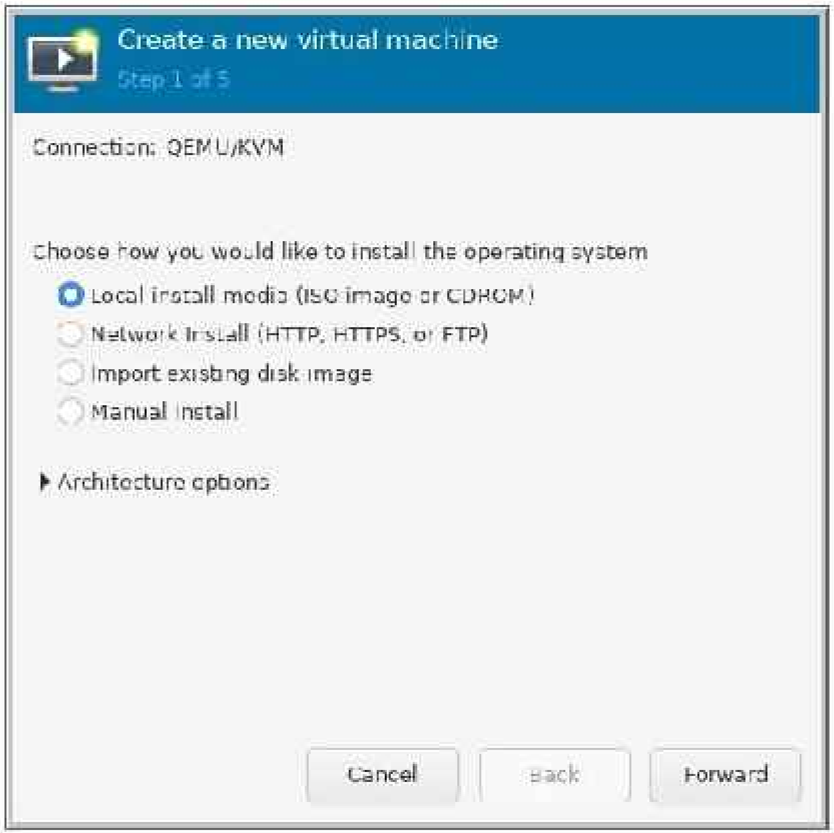
# Initial virt-manager screen

This appears:-



# *File, New Virtual Machine* gives

I shrunk this to fit
But I'll only show
Relevant screen
Portions in the
Following slides



Create a new virtual machine
Step 1 of 5

Connection: QEMU/KVM

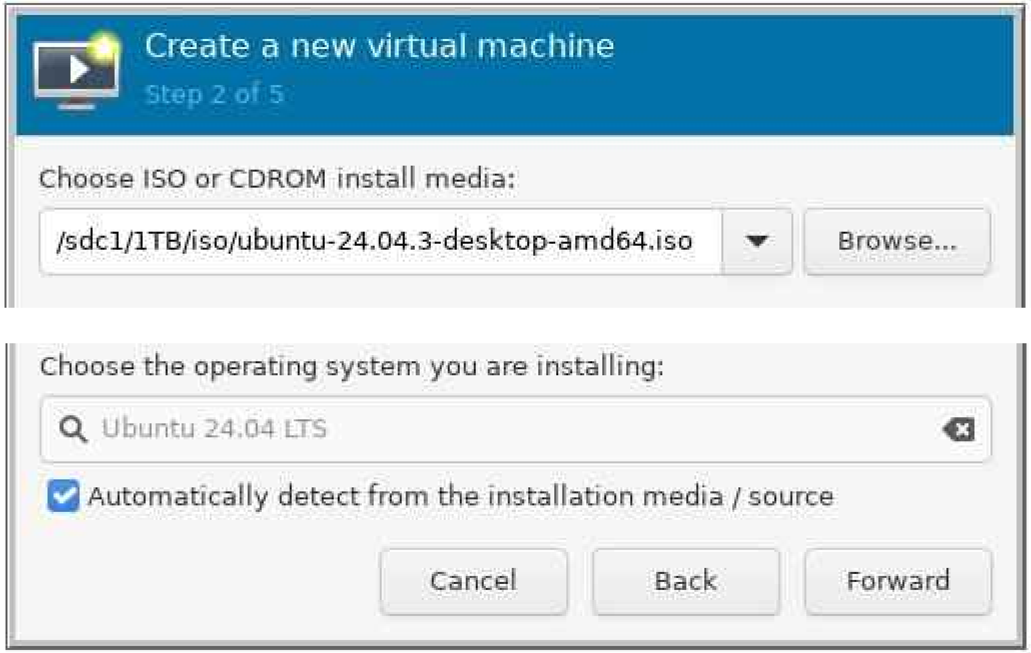
Choose how you would like to install the operating system

- [✓ Local install media (ISO image or CDROM)]
- Network Install (HTTP, HTTPS, or FTP)
- Import existing disk image
- Manual Install

# Step 2: choose input

In the chooser (not shown), you might want to use *Browse Local*.

If **libosinfo.so** recognises the OS it will fill it in for you.



Create a new virtual machine
Step 2 of 5

Choose ISO or CDROM install media:

/sdc1/1TB/iso/ubuntu-24.04.3-desktop-amd64.iso

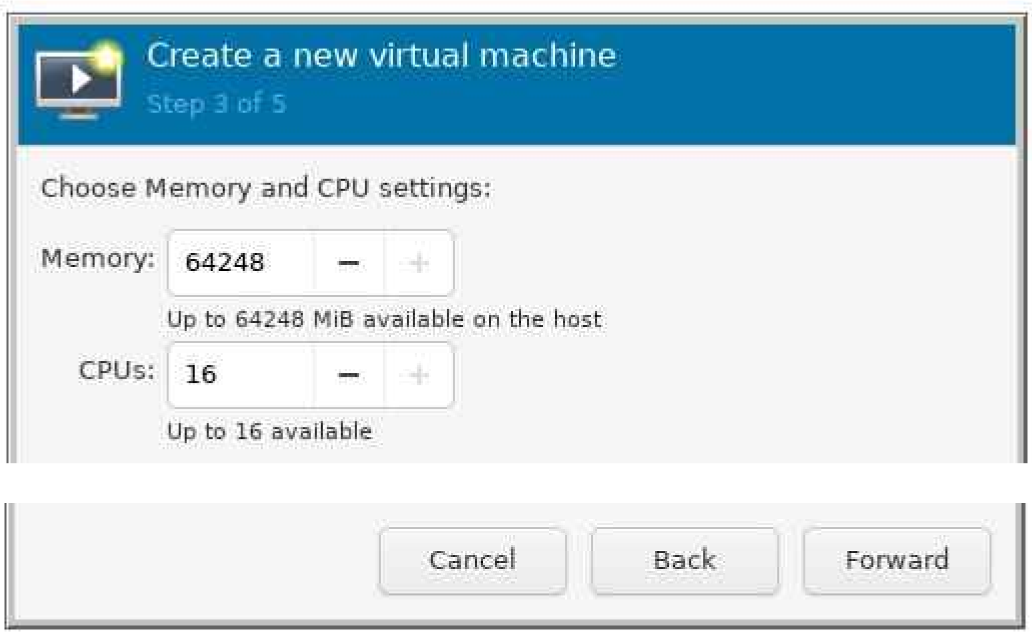
Choose the operating system you are installing:

Ubuntu 24.04 LTS

[✓] Automatically detect from the installation media / source

## Step 3: Memory and CPU(s)

I always choose maximum available CPU & memory.



Choose Memory and CPU settings:

Memory: 64248

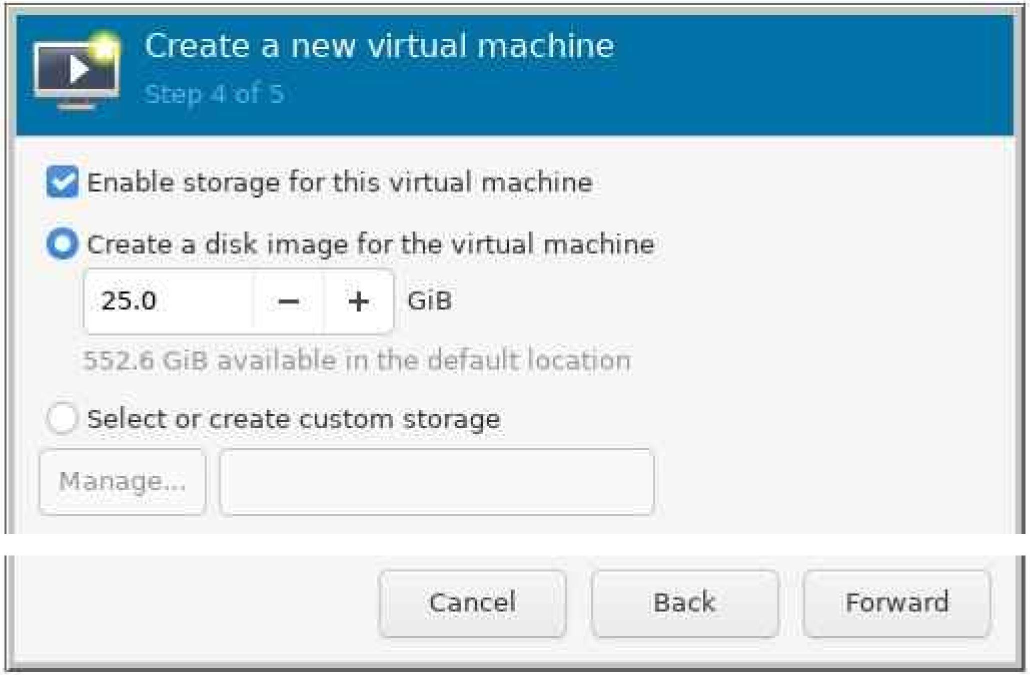
Up to 64248 MiB available on the host

CPUs: 16

Up to 16 available

# Step 4: storage

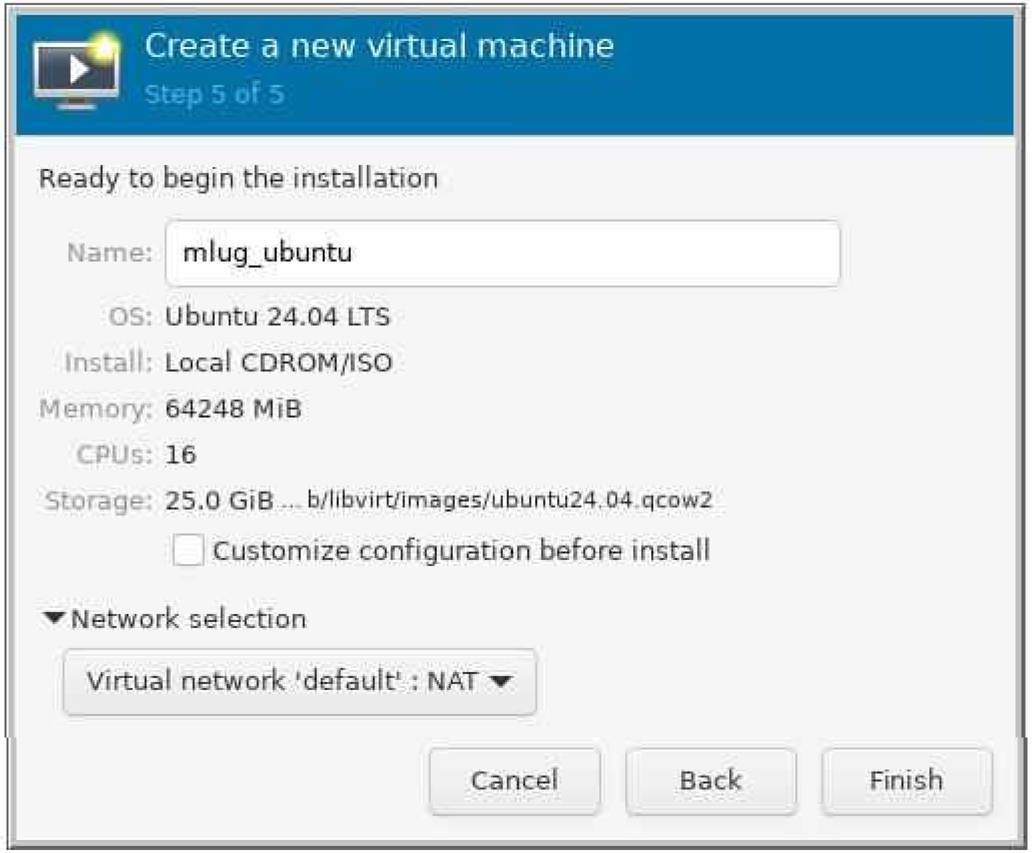
25.0 is the size recommended for this recognised OS. My other ubuntu VM uses 17Gib.


Create a new virtual machine
Step 4 of 5

[✓] Enable storage for this virtual machine
(●) Create a disk image for the virtual machine
25.0 GiB
552.6 GiB available in the default location
( ) Select or create custom storage

## Step 5: Name the system and start install

There was a default name (not shown).
*Network selection* starts off collapsed.


Create a new virtual machine
Step 5 of 5

Ready to begin the installation

Name: mlug_ubuntu
OS: Ubuntu 24.04 LTS
Install: Local CDROM/ISO
Memory: 64248 MiB
CPUs: 16
Storage: 25.0 GiB ... b/libvirt/images/ubuntu24.04.qcow2
[ ] Customize configuration before install

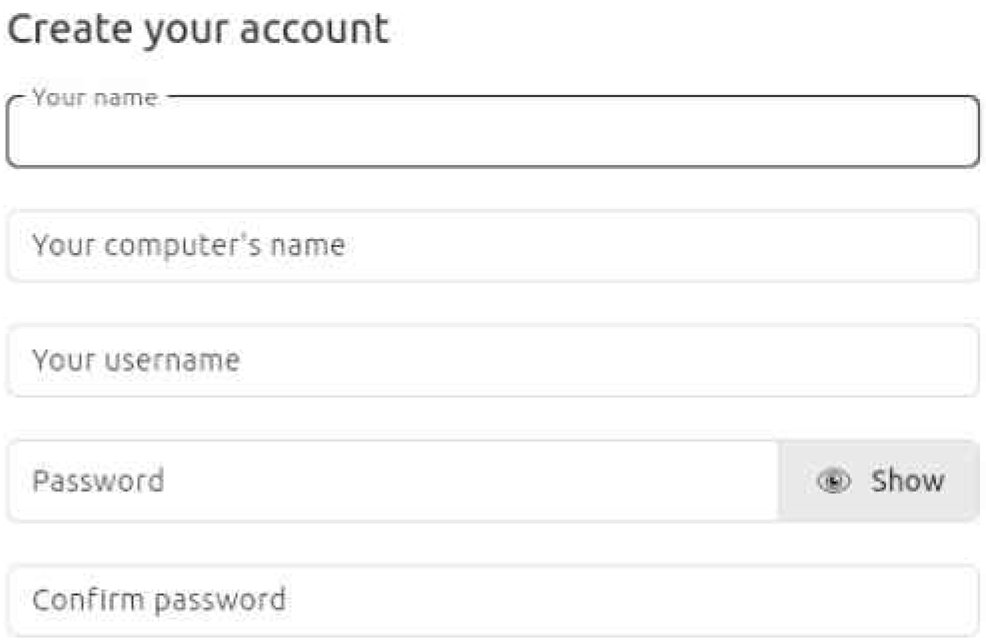
▾ Network selection
Virtual network 'default' : NAT

## Installation starts

- The new system’s console appears in a new window. It showed a GRUB menu but accepted the default before I could get a screenshot.
- Ubuntu starts a GUI and the installer starts.
- Installer asks Language ( default English), Accessibility options, keyboard layout (defaule English (US)), network connection (default wired) and so on until…
- *Create your account* screen appears (next slide).

# Create your account 1/4

This is the empty form.



Create your account

Your name

Your computer's name

Your username

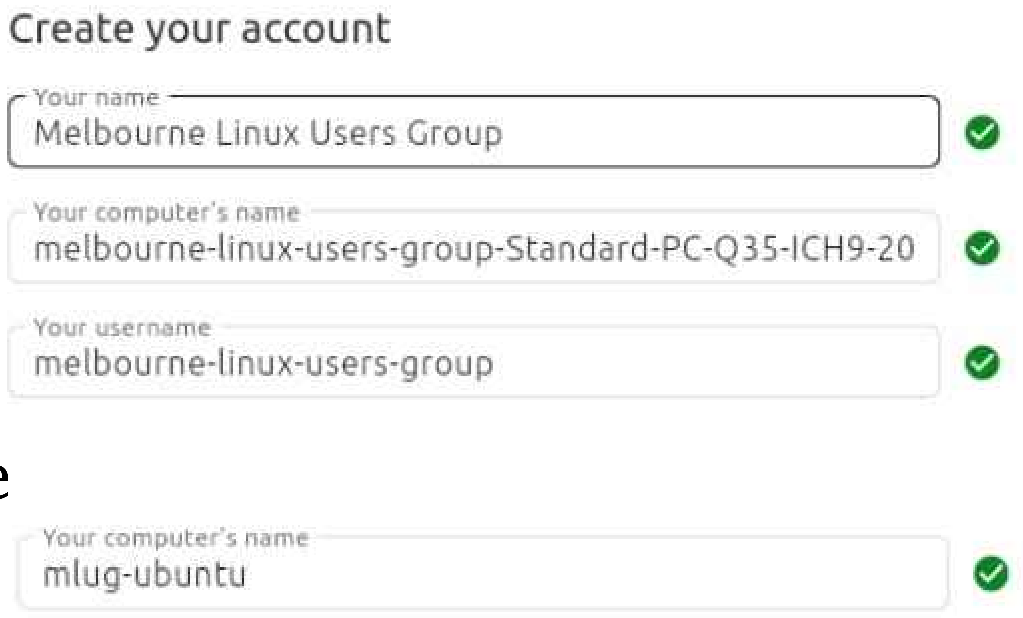
Password

Show

Confirm password

## Create your account 2/4

When you enter your name, the installer guesses the next 2 Fields.
I prefer to use the Computer name from Step 5 (except in this case the underscore was rejected).



Create your account

Your name: Melbourne Linux Users Group ✓

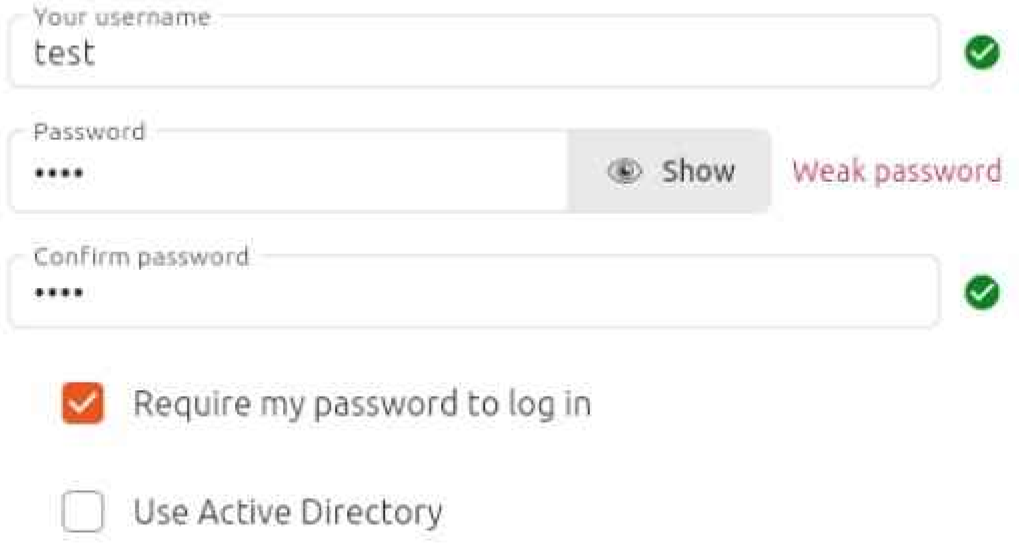
Your computer's name: melbourne-linux-users-group-Standard-PC-Q35-ICH9-20 ✓

Your username: melbourne-linux-users-group ✓

Your computer's name: mlug-ubuntu ✓

# Create your account 3/4

I like to create a test user at this stage. That’s because on all my real & virtual systems I have the same user number 501. Back in the day, default user numbers started there; now they start at 1001. test’s password is test, but I’ll delete test after using it to set the root password.



Your username: test ✓

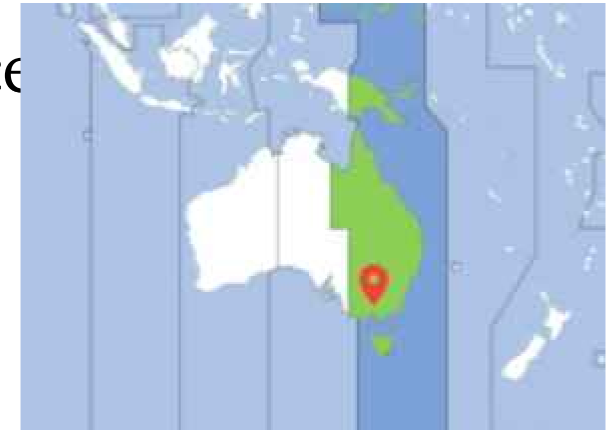
Password: •••• Show Weak password

Confirm password: •••• ✓

☑ Require my password to log in

☐ Use Active Directory

# Create your account 4/4



The timezone selector is rather nice
It’s a map of the whole world with
Timezones (except it doesn’t show
Antarctica (bad luck you guys)).

After timezone selection, there’s a
review screen after which full installation starts.

Other distributions will install quite differently, of course.

## Gaining access to host mount points

*virtiofs* (described next month) claims to work for guest kernels 5.4 or later, but I’ve had trouble with 5.10. My Knoppix 9.1 DVD falls into this category: kernel 5.10 but virtiofsd gets an error:-

```
root@Microknoppix:/# mount -t virtiofs smallstar /smallstar/
mount: /smallstar: wrong fs type, bad option, bad superblock on /smallstar, missing codepage or helper program, or other error.
```

So do it the old way, using *nfs*.

## Worked Example – Knoppix 9.1 1/2

1. On the guest, determine dhcp-assigned ip address (starts 192).

```
root@Microknoppix:/# ifconfig|grep 192
inet 192.168.122.84  netmask 255.255.255.0  broadcast 192.168.122.255
```

2. On the host, add discovered ip to /etc/hosts.

```
12:32:01$ grep 192 /etc/hosts
192.168.122.84 Microknoppix.local.net Microknoppix
```

3. Add 1 entry to /etc/exports for each local partition.

```
12:48:52$ grep Micro /etc/exports
/ Microknoppix(rw,no_root_squash,sync,wdelay,no_subtree_check)
```

4. On the host, activate the new entries.

```
12:48:34# exportfs -r
```

## Worked Example – Knoppix 9.1 2/2

5. On the guest, create mount points for the host partition(s).

```
root@Microknoppix:/# mkdir smallstar
```

6. On the guest, mount the partition(s) added in step 4.

```
root@Microknoppix:/# mount smallstar:/ smallstar
```

Since the guest ip was from *dhcp* it could change but in practice rarely does.
If the ip does change, you only need to change 1 line in <u>hosts</u> rather than several in <u>exports</u>.