Slackware ARM project web site | Forum | Slackware ARM development documentation | Slackware ARM installation guides

Installing Slackware: Paravirtualization on Apple Silicon within UTM hypervisor

Platform	AArch64/ARM64
Hardware Model	macOS / Apple Silicon
Document Version	1.00, Apr 2025
Author	Stuart Winter <mozes@slackware>

Video Tutorial

This tutorial is also available in video form.

Other Virtualization Platforms

If you'd prefer to use VMWare Fusion, follow the Installation Guide.

Supporting the Slackware ARM Project

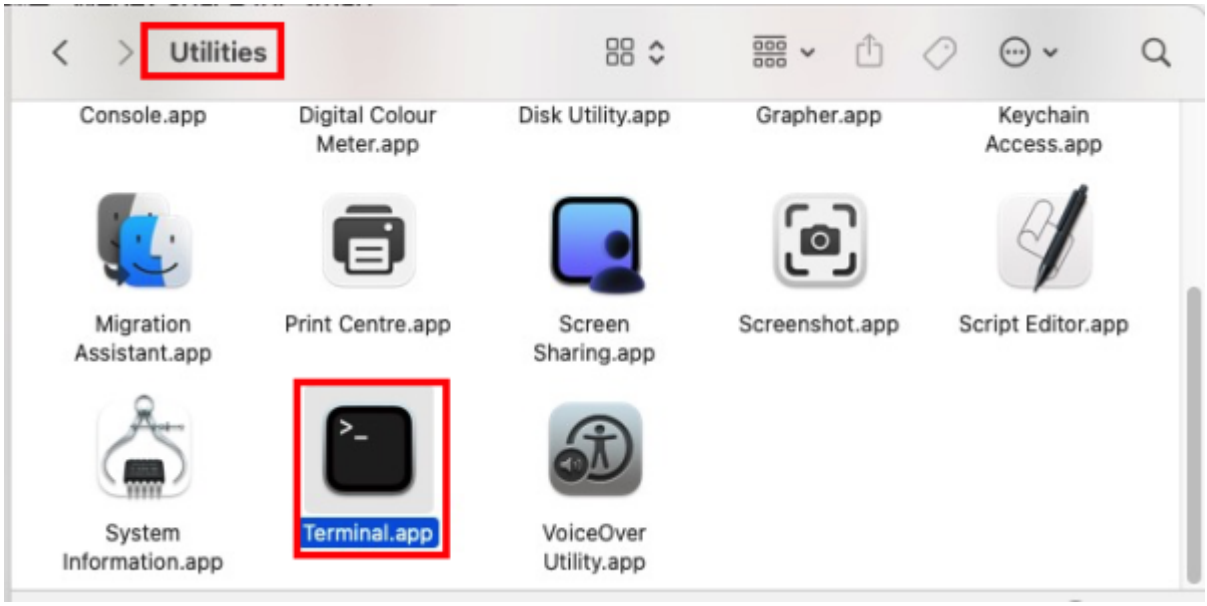
If you like what we're doing here, please consider becoming a patron.

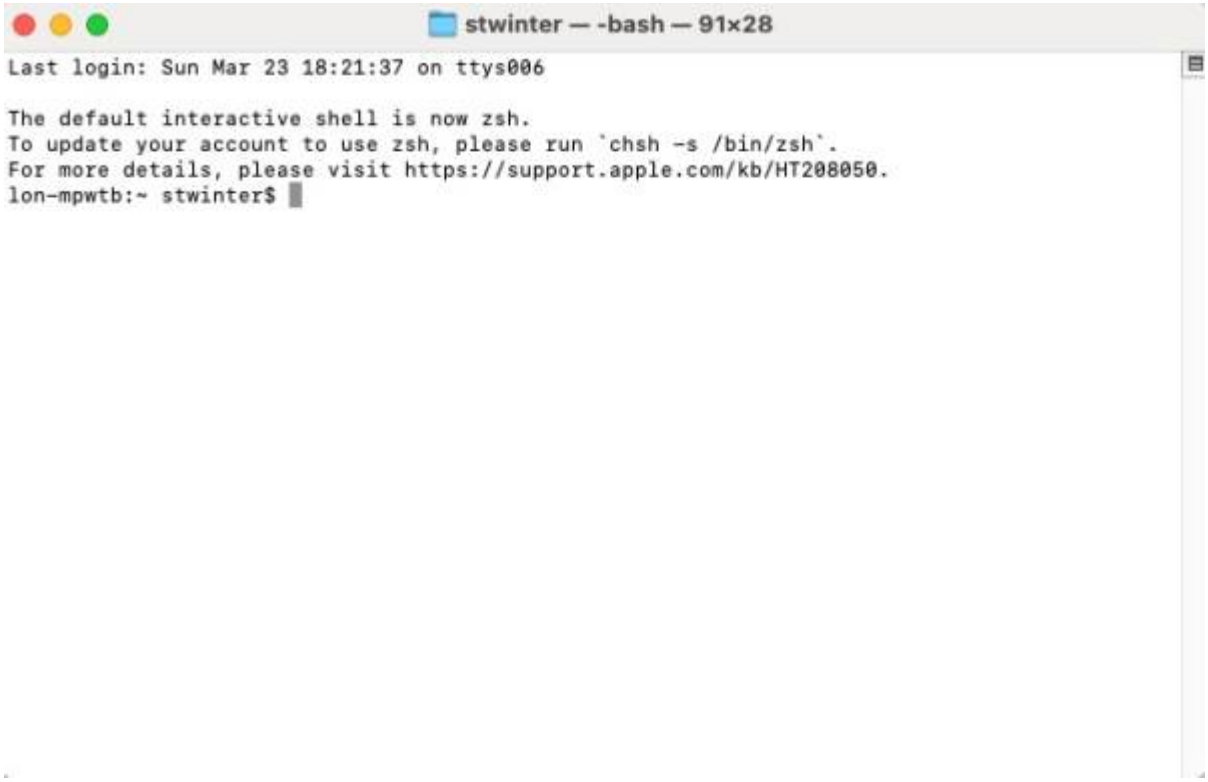
Download the Slackware Linux AArch64 Installer ISO image

The '$' prefixes in the commands indicates the shell prompt - it's not to be typed/copied

Open a shell/terminal on the Mac:

- The Terminal application can be found within `Applications` → `Utilities`





Prepare a directory to hold the downloaded assets

We'll download the Slackware Linux distribution into a directory named 'slackware'.

```
$ cd ## this returns to the root of your home directory
$ mkdir slackware
$ cd slackware
```

Set the version of Slackware AArch64 to download

At the time of writing, the only version available is 'current'.

```
$ SLKVER=current
```

Set the Internet media distribution server

If you are using a mirror server rather than the master Slackware ARM server, set it here. The format is: <hostname>::<rsync module name>

```
$ SLKSRV=ftp.arm.slackware.com::slackwarearm
```

Download the Slackware Linux Installer

```
$ rsync -PavL $SLKSRV/platform/aarch64/bootware/installer-
aio/slackwareaarch64-${SLKVER}/aarch64_generic.iso.asc slk-
aarch64_generic.iso.asc
$ rsync -PavL $SLKSRV/platform/aarch64/bootware/installer-
aio/slackwareaarch64-${SLKVER}/aarch64_generic.iso.md5 slk-
aarch64_generic.iso.md5
$ rsync -PavL $SLKSRV/platform/aarch64/bootware/installer-
aio/slackwareaarch64-${SLKVER}/aarch64_generic.iso slk-aarch64_generic.iso
```

The Slackware Installer images are approximately 5 GBytes in size.

Verify the Slackware Installer image

Since GPG is not provided on MacOS by default, the easiest way to verify the images is using the MD5 hash.

Hash method

These MD5 sums are examples - the images you download will have different hashes

```
$ openssl md5 -r slk-aarch64_generic.iso ; cat slk-aarch64_generic.iso.md5
380d7a8a06dc575e2d4e3140725f7a12 *slk-aarch64_generic.iso
380d7a8a06dc575e2d4e3140725f7a12
```

Manually verify that the MD5 sums match.

GPG method

GnuGPG is not installed in MacOS by default

If you have GPG installed, verify the digital signature of the Slackware Installer.

Installing the Slackware ARM GPG key

The Slackware ARM GPG key can be used to verify your downloads.

If you are able to verify the GPG signatures, you can download and import the GPG key like this:

```
$ curl -sSL https://www.slackware.com/infra/keys/arm/GPG-KEY | gpg --import
-
```

```
$ gpg --verify slk-aarch64_generic.iso.asc
```

As the images are large, verification may take a minute or two.

The output will be similar to this. You are looking for 'Good signature from Slackware ARM...'

```
gpg: assuming signed data in `slk-aarch64_generic.iso'
gpg: Signature made Mon 24 Mar 2025 06:07:44 PM GMT
gpg:                using RSA key F7ABB8691623FC33
gpg: Good signature from "Slackware ARM (Slackware ARM Linux Project)
<mozes@slackware.com>" [unknown]
gpg: WARNING: This key is not certified with a trusted signature!
gpg:          There is no indication that the signature belongs to the
owner.
Primary key fingerprint: 36D3 7609 2F12 9B6B 3D59  A517 F7AB B869 1623 FC33
```

If you see 'BAD signature' you should re-download as it may have become corrupt. If this doesn't help, drop a note to the Slackware ARM forum

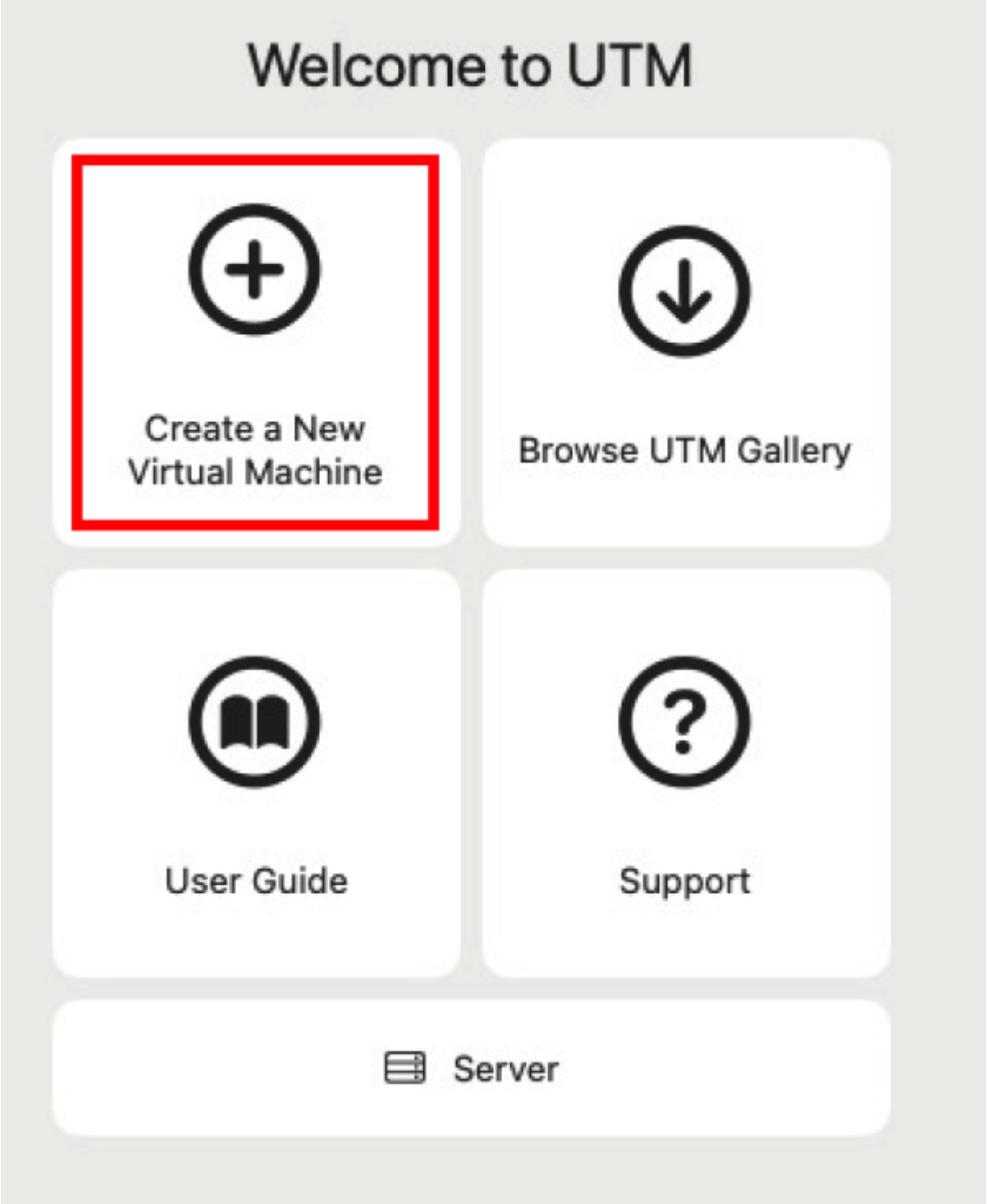
Download and Install UTM

If you do not already have UTM installed, Download UTM for MacOS and install it.

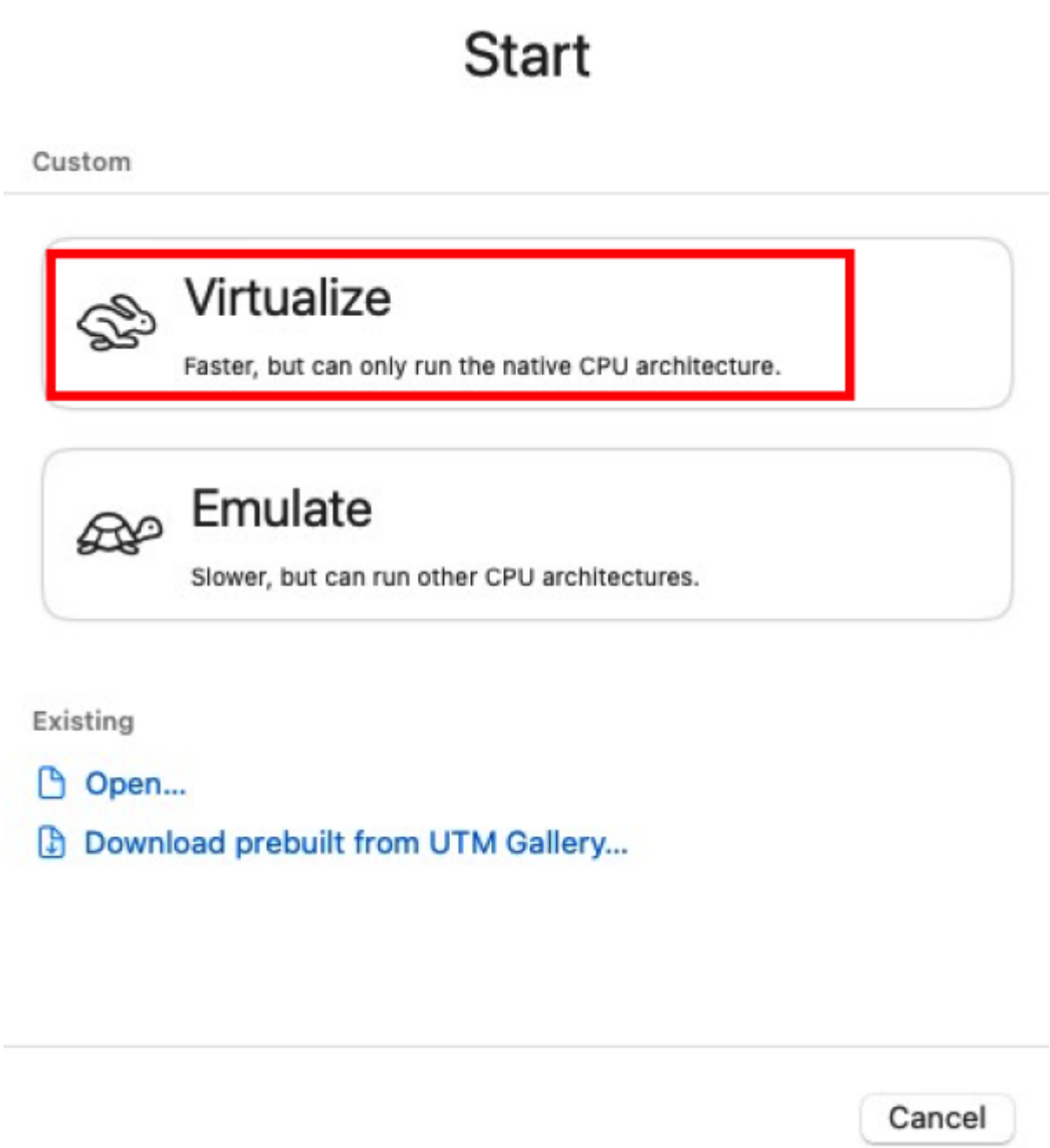
Version of UTM tested: **4.6.4 (107)**

Creating the Virtual Machine within UTM

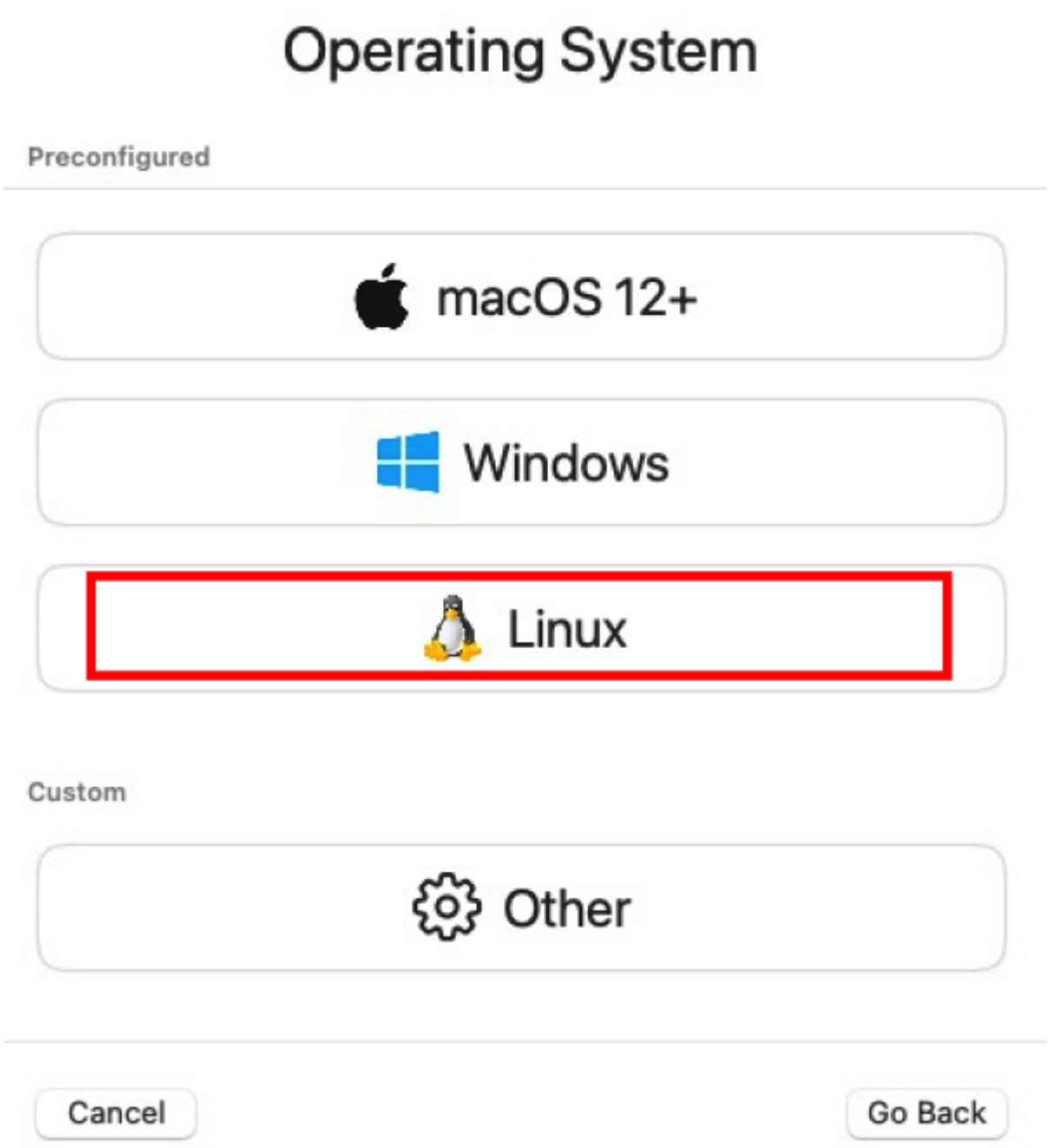
- Click `Create a new Virtual Machine`



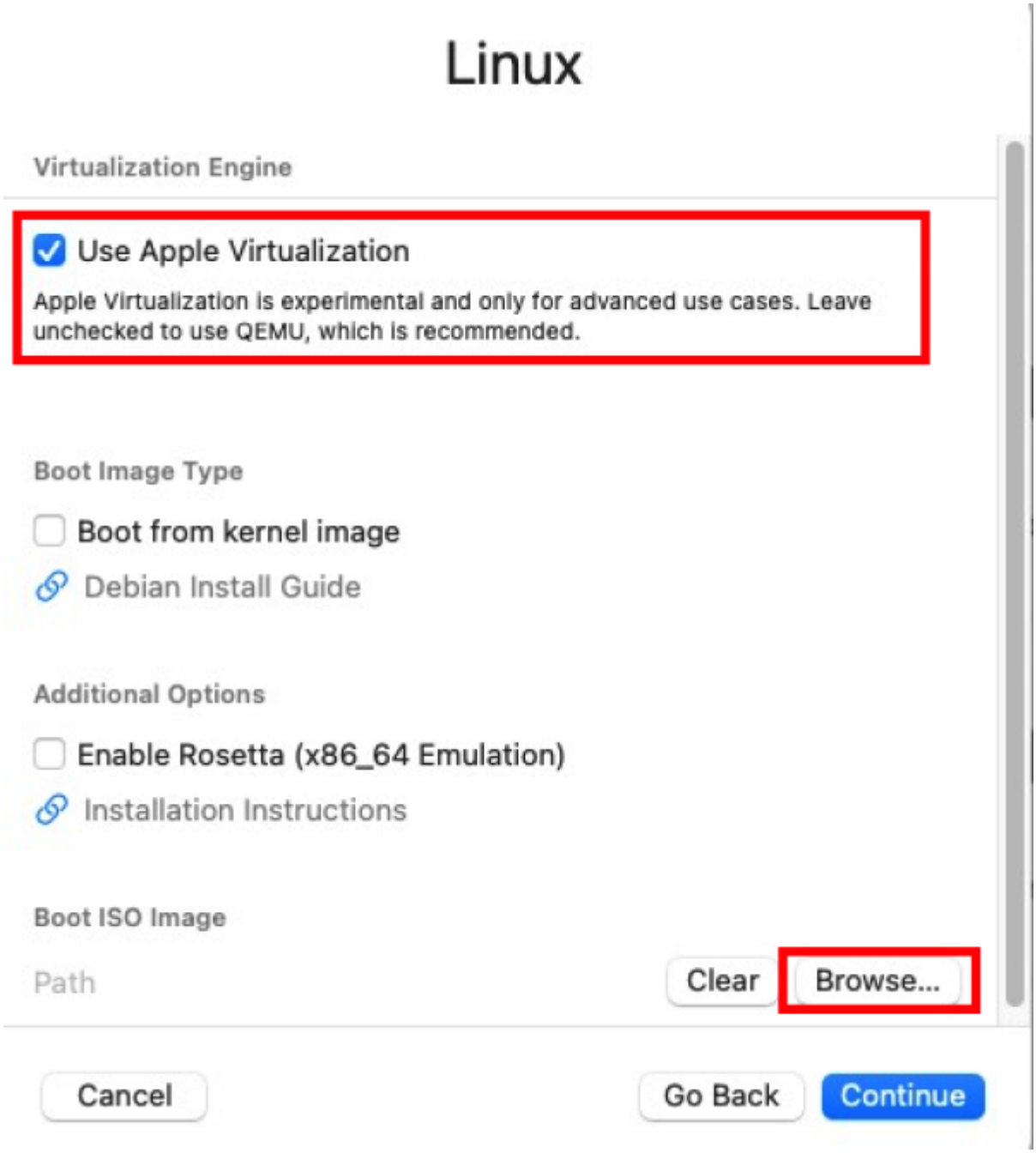
- Click `Virtualize`


Start
Custom
Virtualize
Faster, but can only run the native CPU architecture.
Emulate
Slower, but can run other CPU architectures.
Existing
Open...
Download prebuilt from UTM Gallery...
Cancel

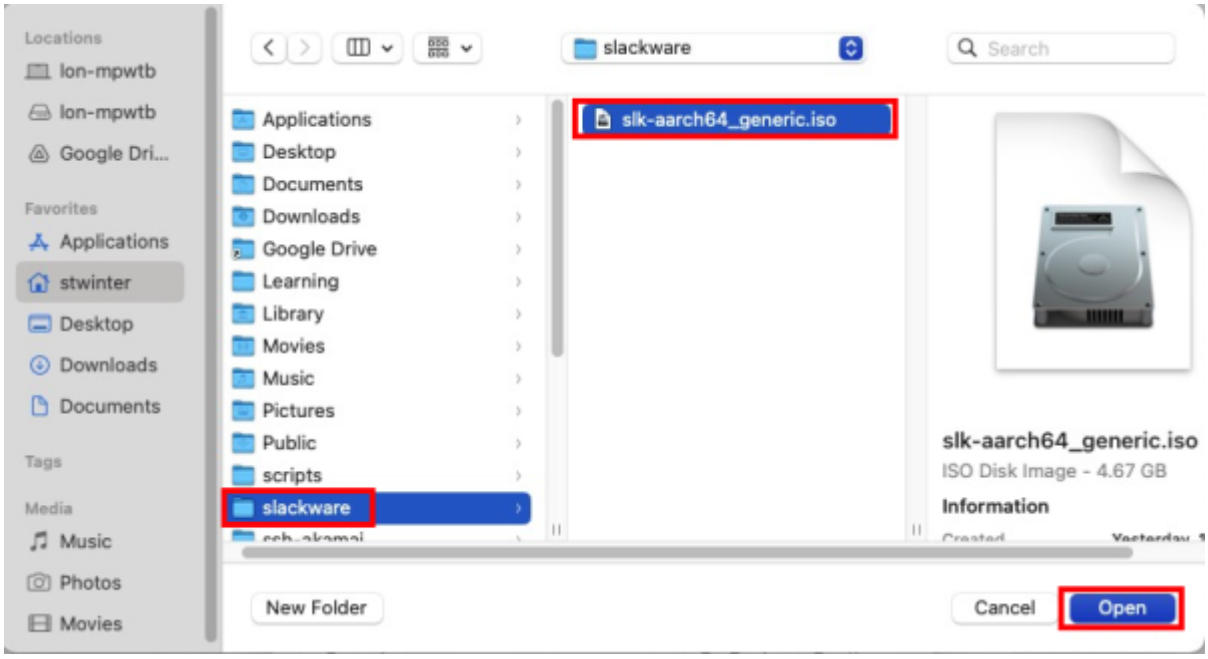
- Click `Linux`



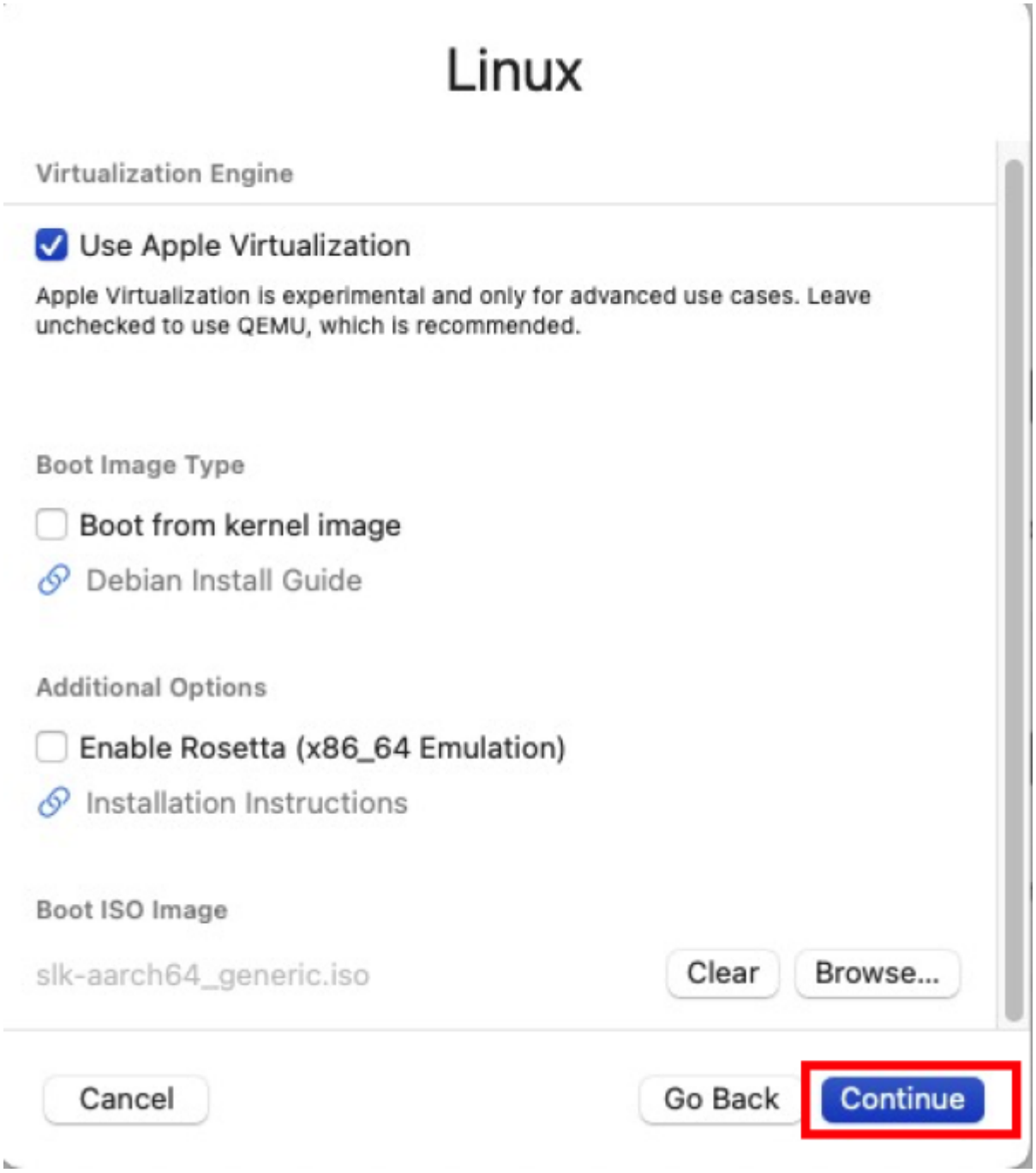
- Tick `Use Apple Virtualization`



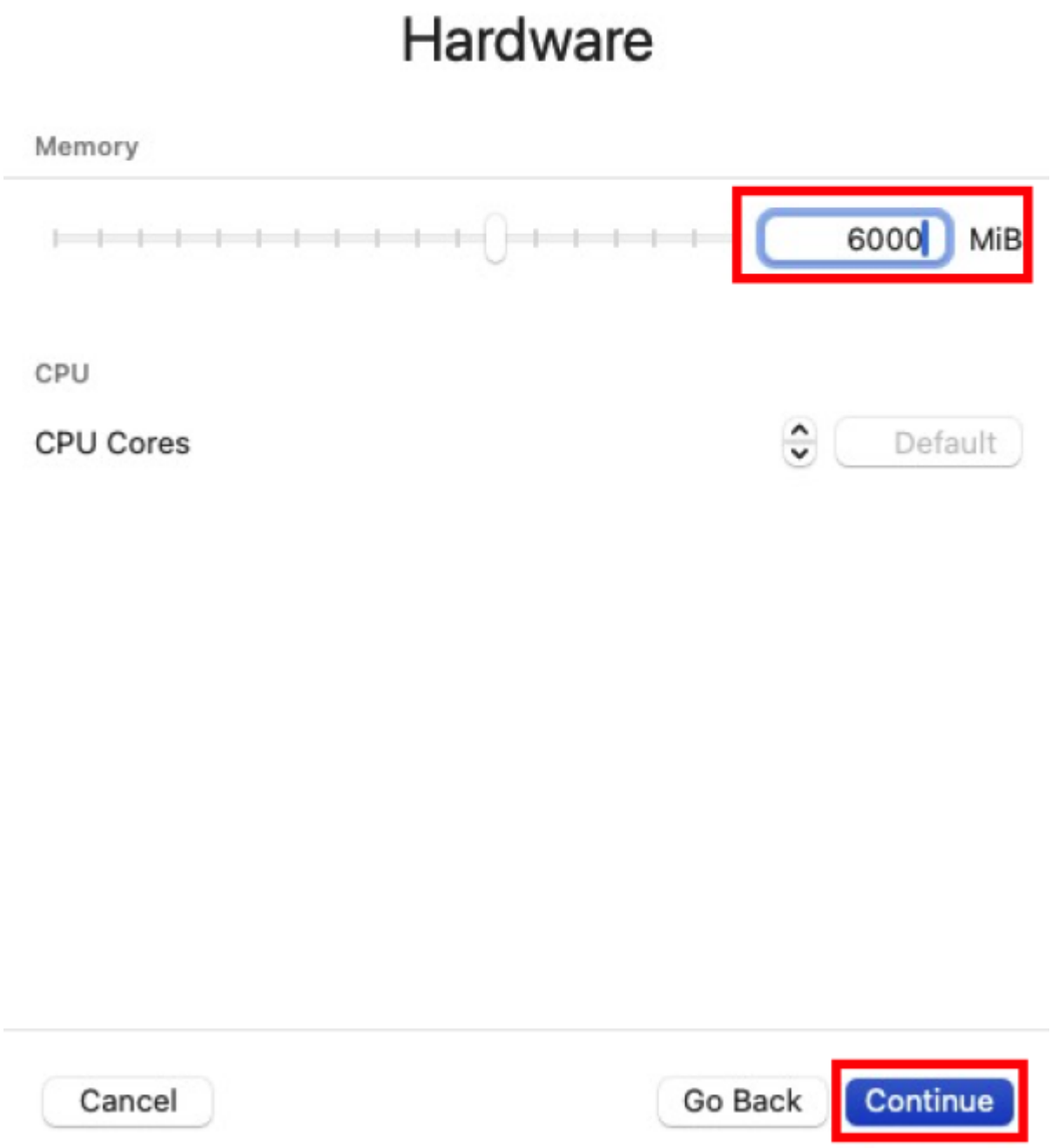
- Select the Slackware ISO image that you downloaded



- Click `Continue`

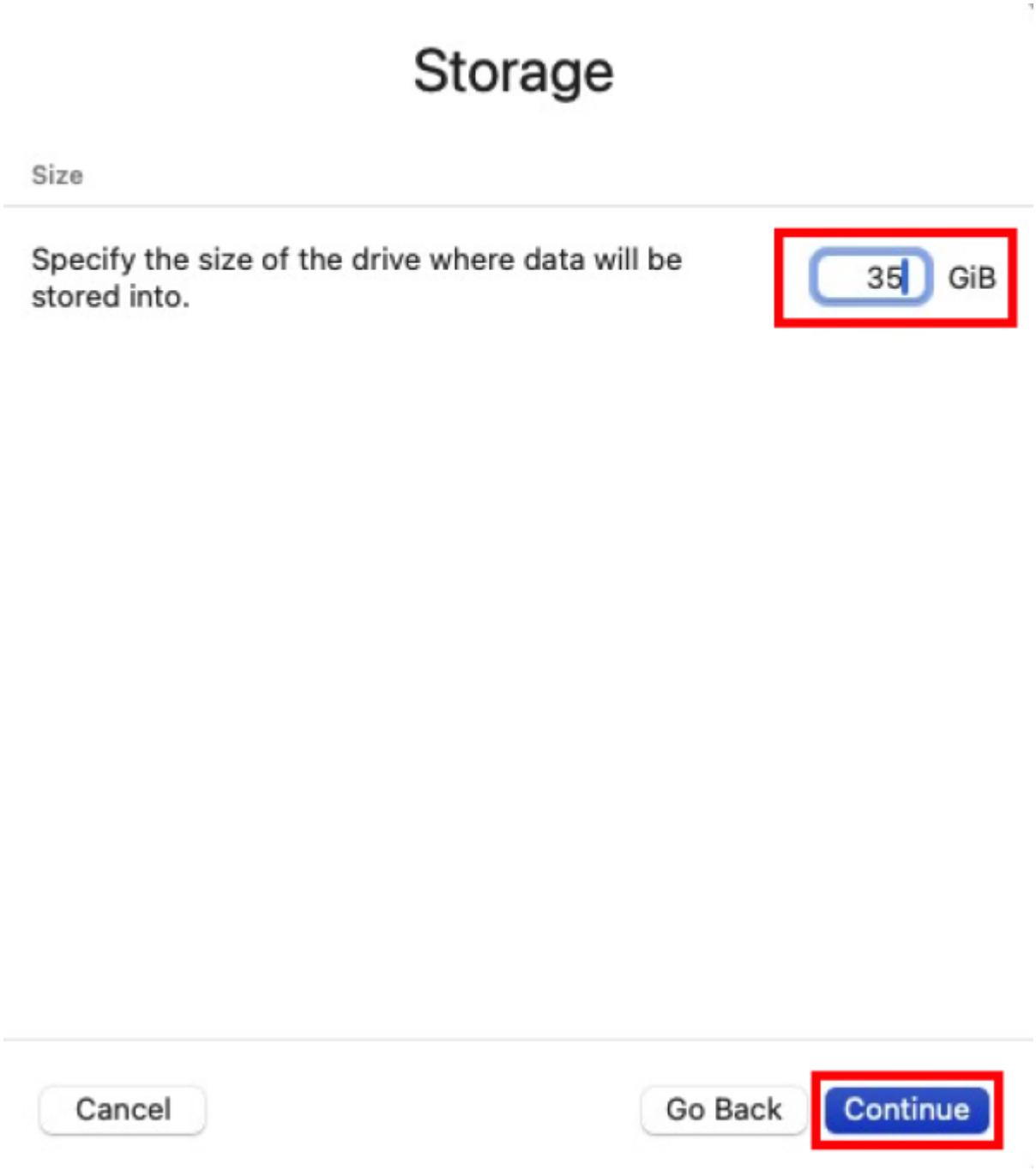


- Adjust the amount of RAM allocated to 6GB. The installation will hang when only 4GB of RAM is allocated (despite working fine in VMware Fusion and on bare metal, where the installer can even run with as little as 1GB of RAM).

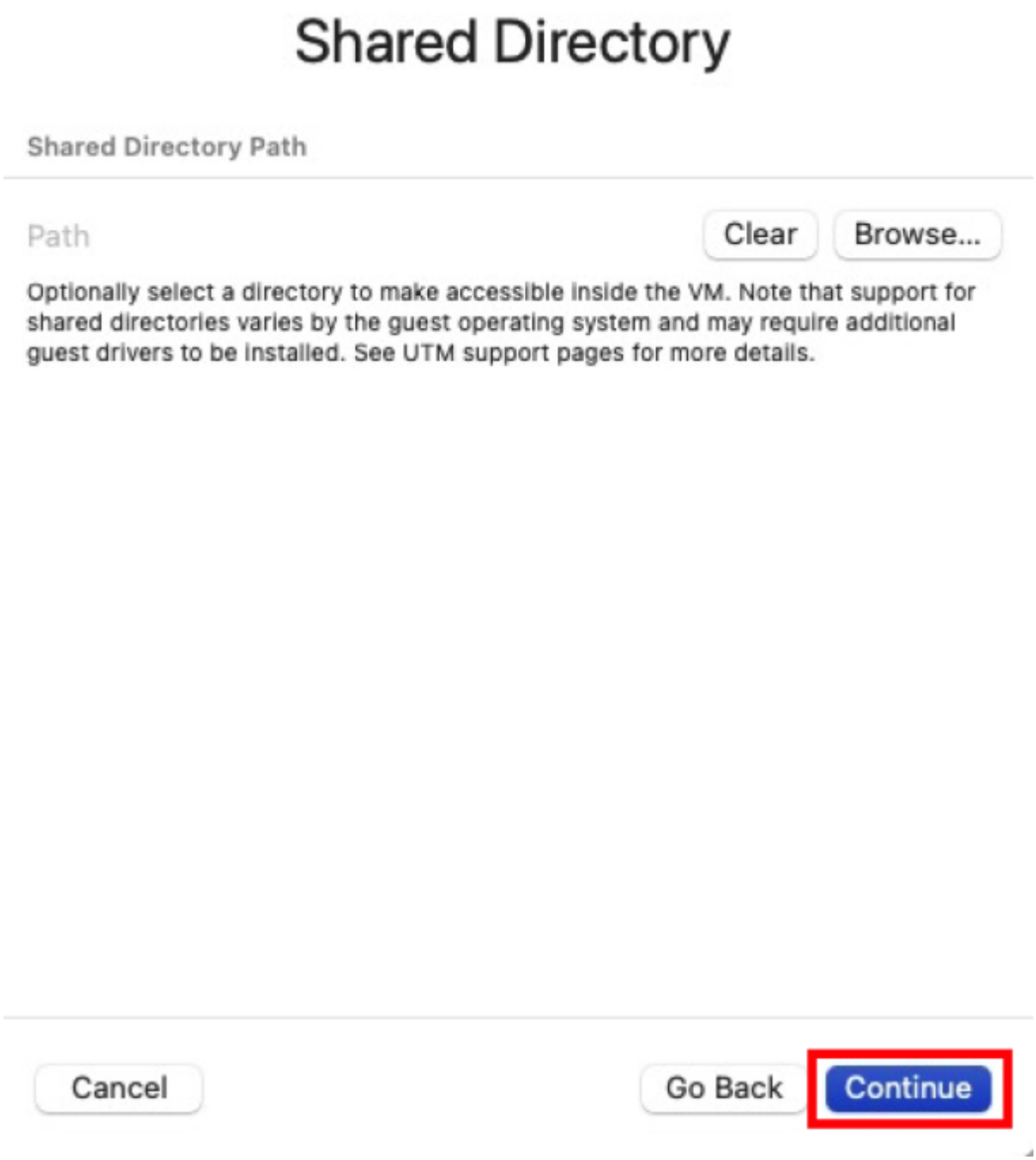


- The default allocation for storage is 64G, but 35GB is sufficient unless you plan on housing large amounts of data.

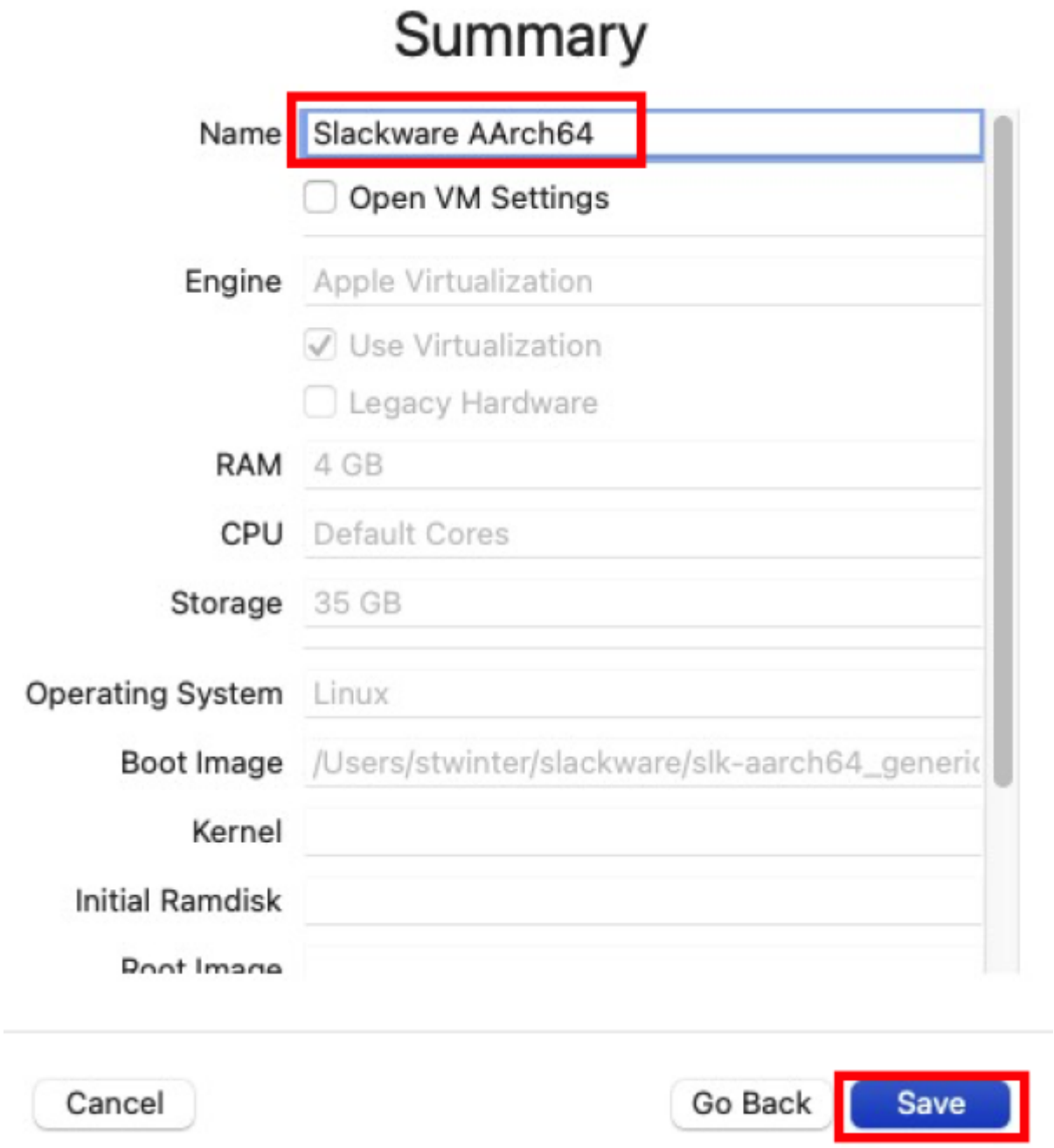

A full installation of Slackware requires approximately 25GB



- You may configure some shared directories if you wish. This is optional. In this guide we will not configure any.



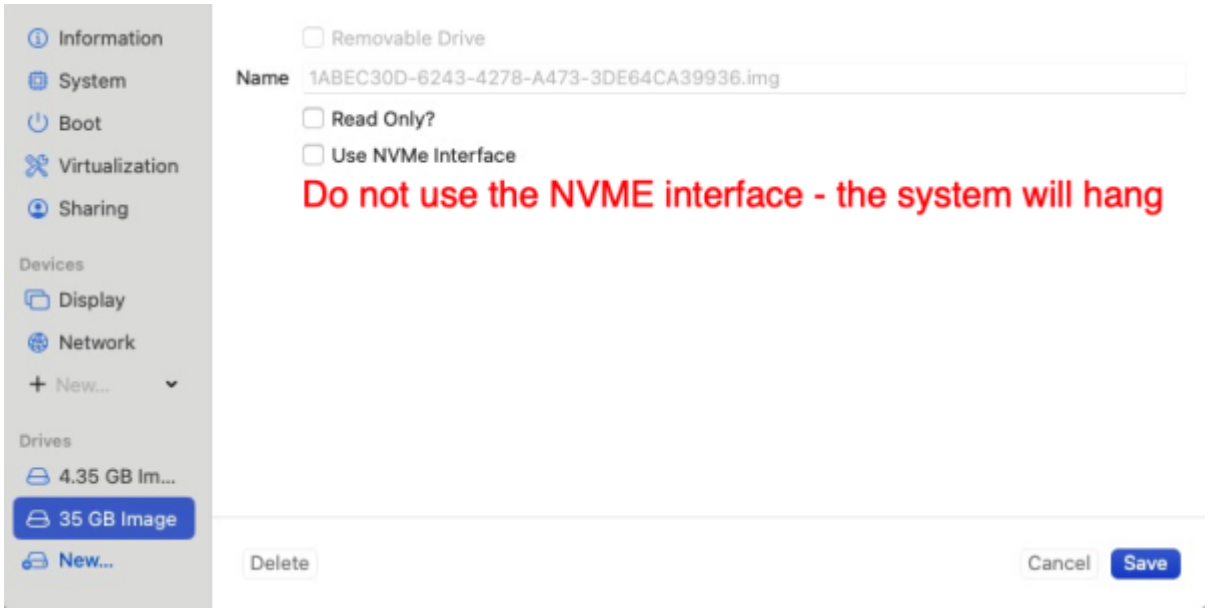
- Pick a name for your VM and click `Continue`



UTM VM Settings

There are some other settings for UTM that you may wish to configure at this point. However:

Do not select 'Use NVME' interface' for the hard disk because the system will hang under load



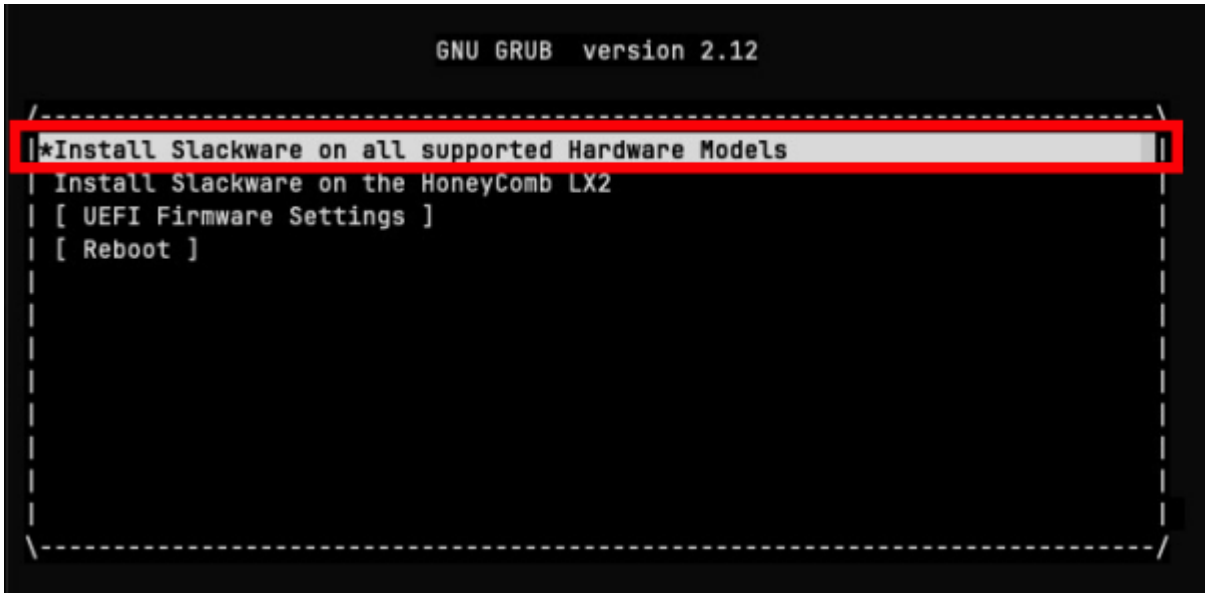
Booting the Slackware Installer

The system will boot into a menu.

You may press `e` to edit any of the boot loader settings if you wish - which may be useful if using the Installer as a rescue environment

The `UEFI Firmware Settings` option is unsupported in UTM

Press ENTER on the menu item `Install Slackware on all supported Hardware Models`

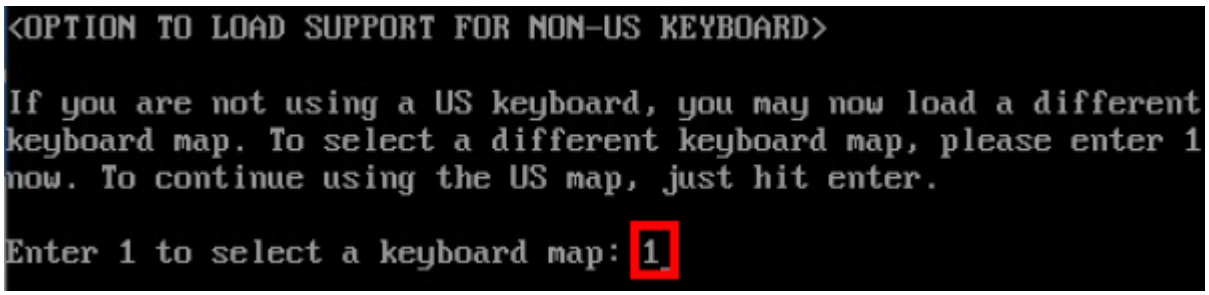


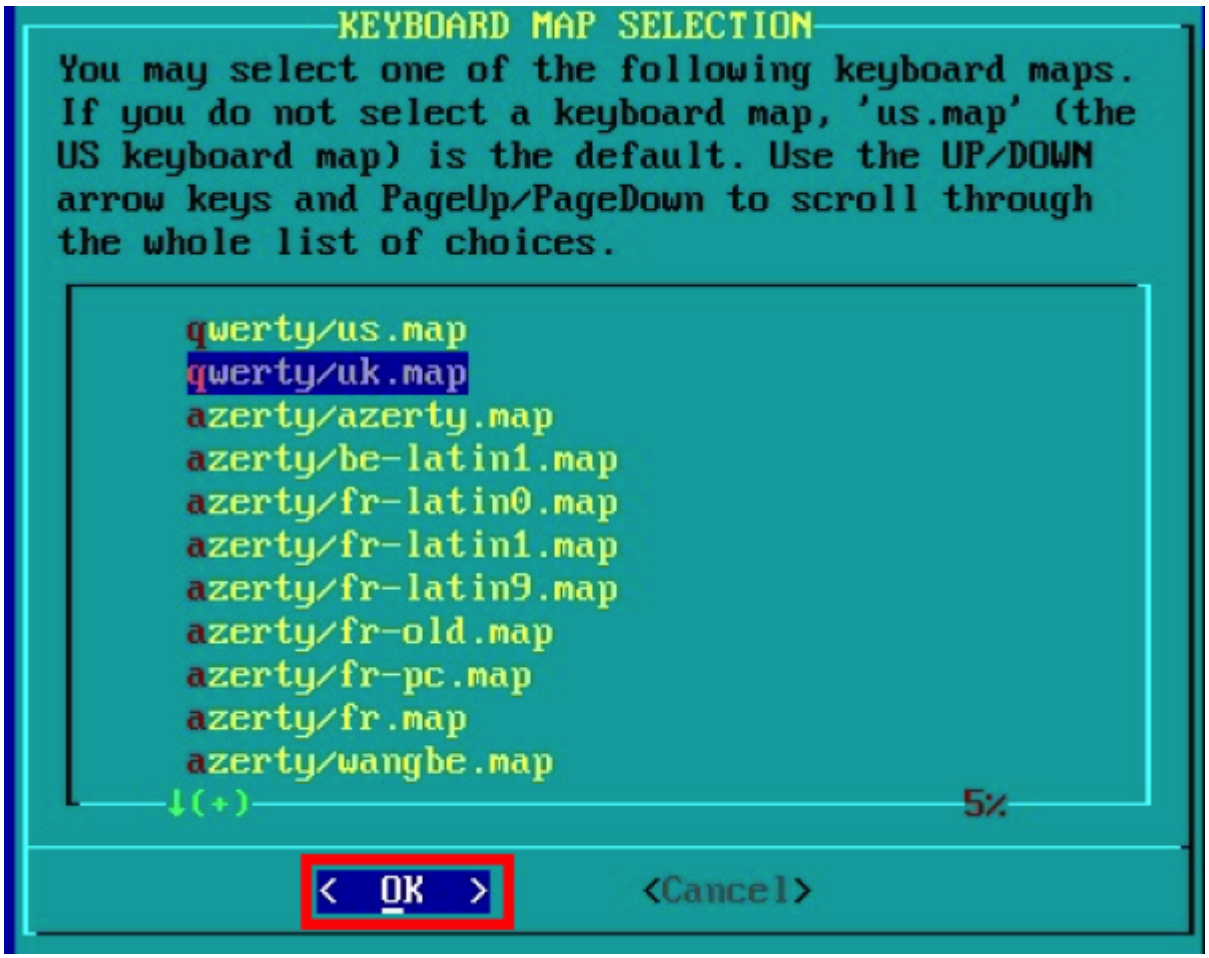
The system will boot within a couple of seconds, and will attempt to acquire an IP address via DHCP. If no DHCP server is available, this will take a little longer to time out.

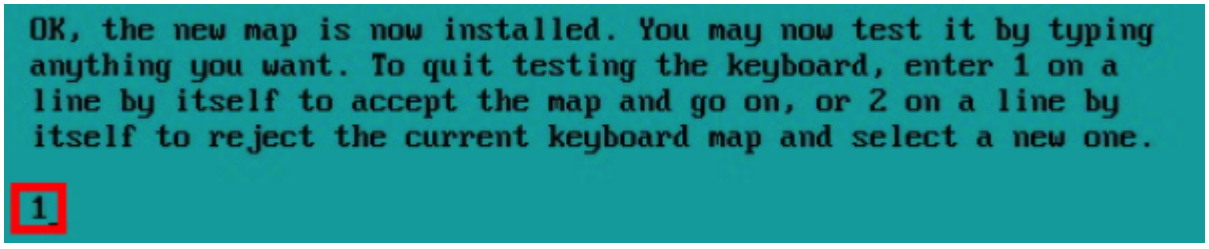
Networking is not required to install Slackware - the packages are contained within the ISO

Select a Key Map

If you are not using a US key map, select the appropriate key map for you:

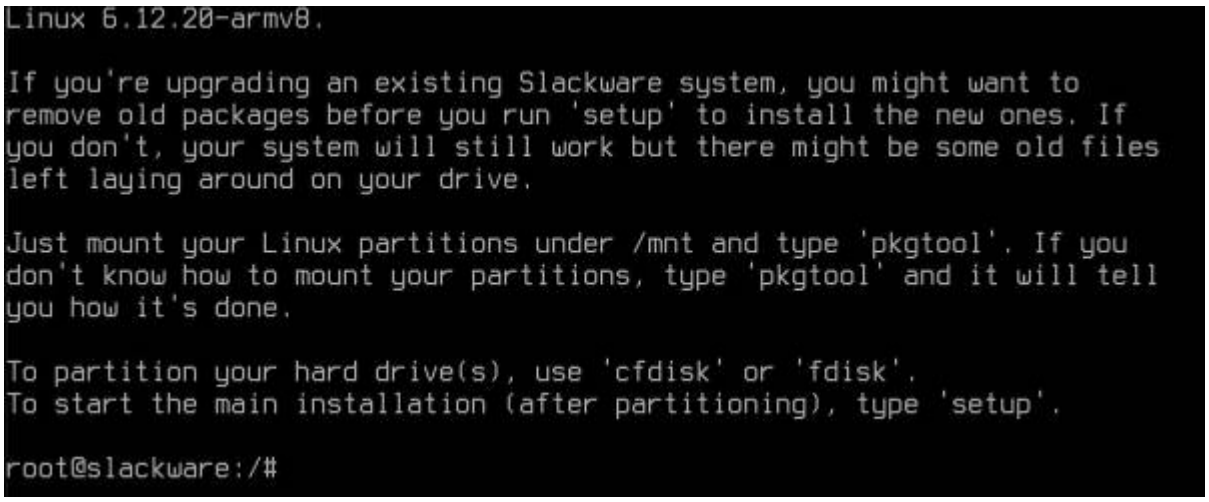






Beginning the Setup

You will be presented with a shell prompt:



Creating the Partition Table

You need to create the partition table for the OS and your data to live on.

Open the fdisk Tool

In this guide, we'll create a simple partition scheme. You're free to customise the layout as needed, but make sure to set the disk label to GPT and create a 100MB EFI partition as the first partition.

```
fdisk /dev/vda
```

Create a GPT Disk Label

As we're using an EFI system, we'll create a 100MB EFI partition first

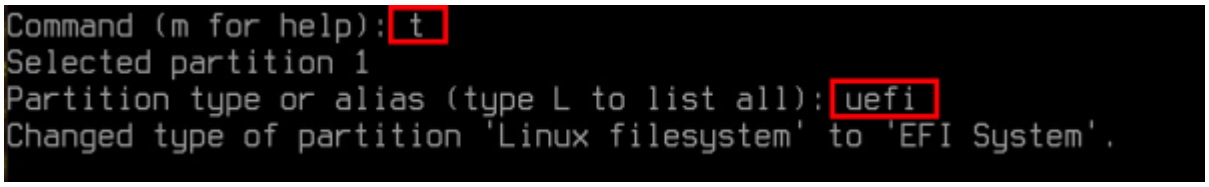
Type g and press ENTER:

Create a 100MB UEFI Partition

- Type n and press ENTER
- Press ENTER for both the *"partition number"* and *"First sector"*
- For the *"Last sector"* enter +100M

Set the partition type to `uefi`:

- Type `t` and press ENTER
- Type `uefi` and press ENTER



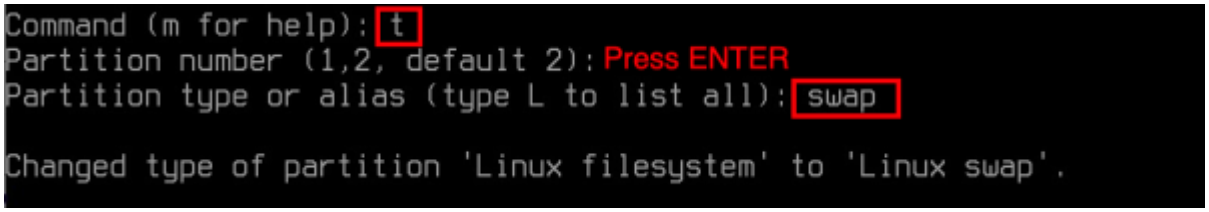
Create the Swap Partition

For the 2nd partition we'll create a 4GB Swap partition.

- Type n and press ENTER
- Press ENTER for the *"partition number"*
- Press ENTER for the *"First sector"*
- For the *"Last sector"* enter +4G

Set the partition *type* to Swap:

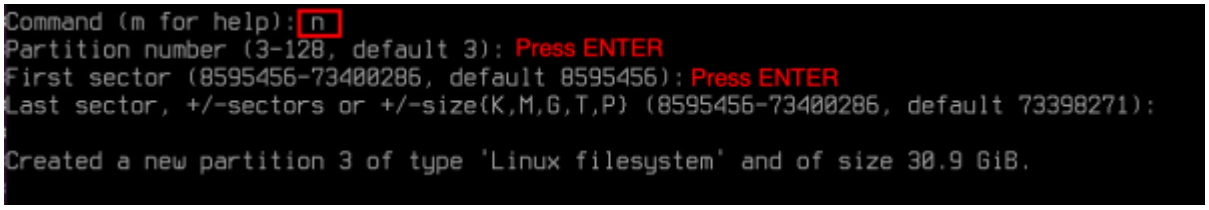
- Type `t` and press ENTER
- Press ENTER for the *"partition number"*
- Type swap and press ENTER



Create the root Partition

This partition will house the OS and your data and will fill the remainder of the allocated storage:

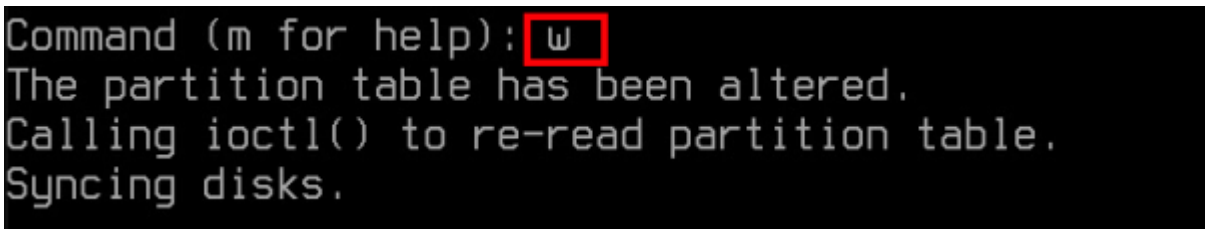
- Type n and press ENTER
- Press ENTER for the *"partition number"*
- Press ENTER for the *"First sector"*
- Press ENTER for the *"Last sector"*



Save the Partition Layout

The partition layout must now be saved.

- Type w and press ENTER



Begin the Installation

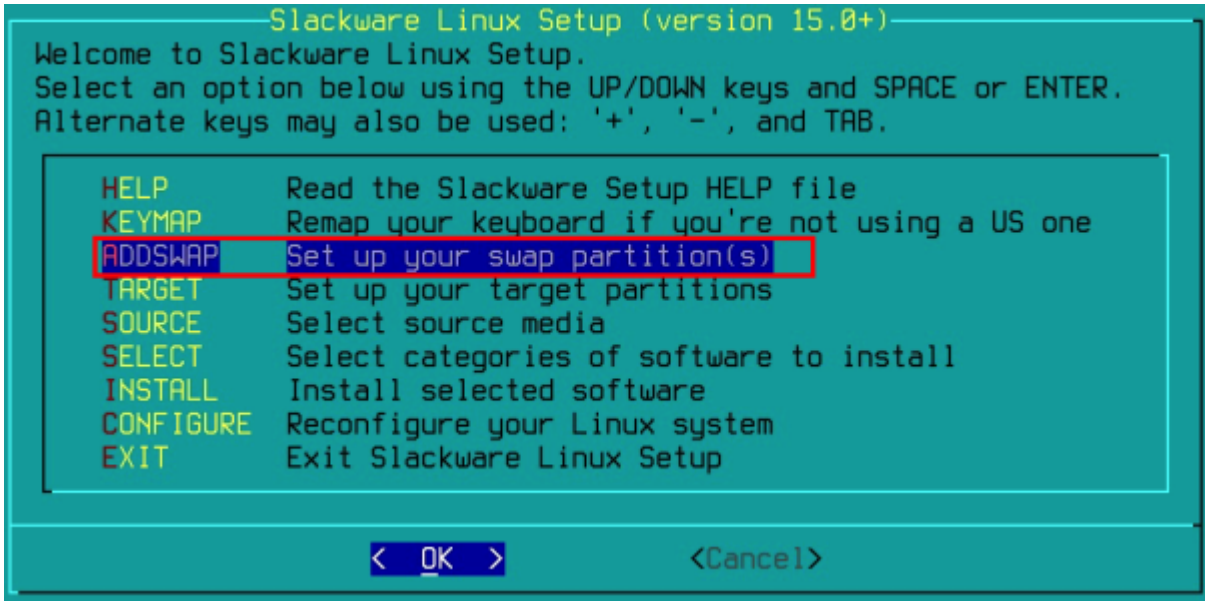
You may now begin the Slackware installation process.

At the shell, type the command `setup` and press ENTER

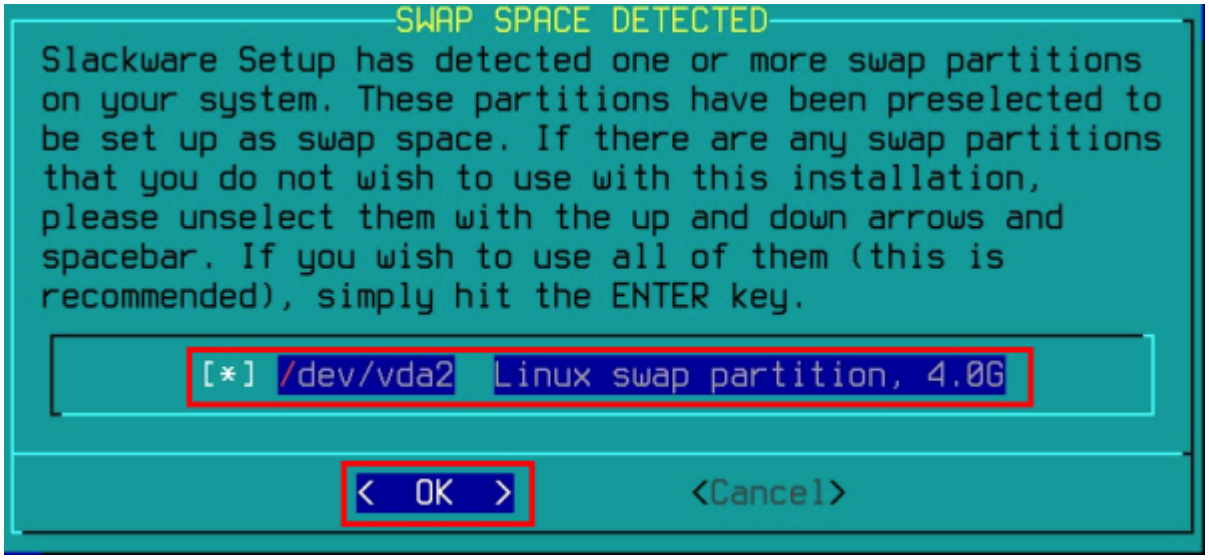
Installation Essentials

We'll now perform the initial setup steps:

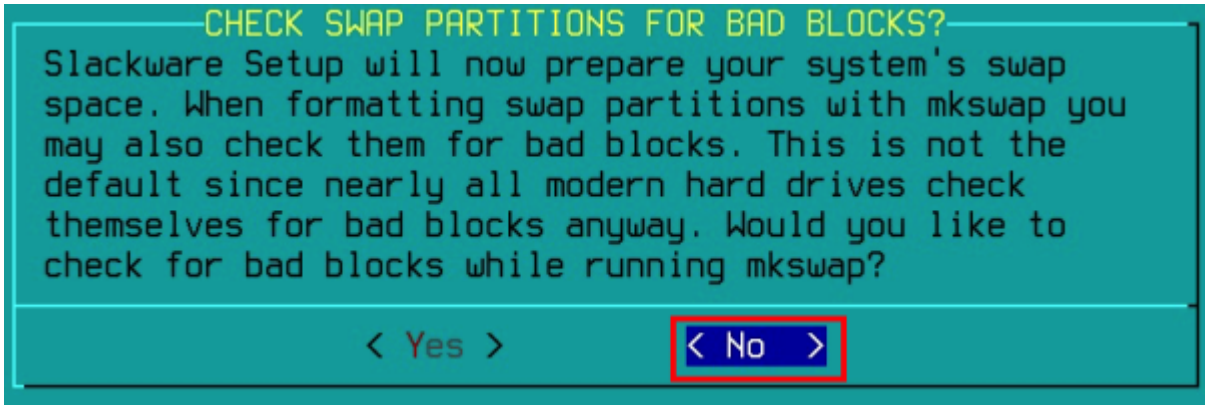
- Scroll down to ADDSWAP and press ENTER

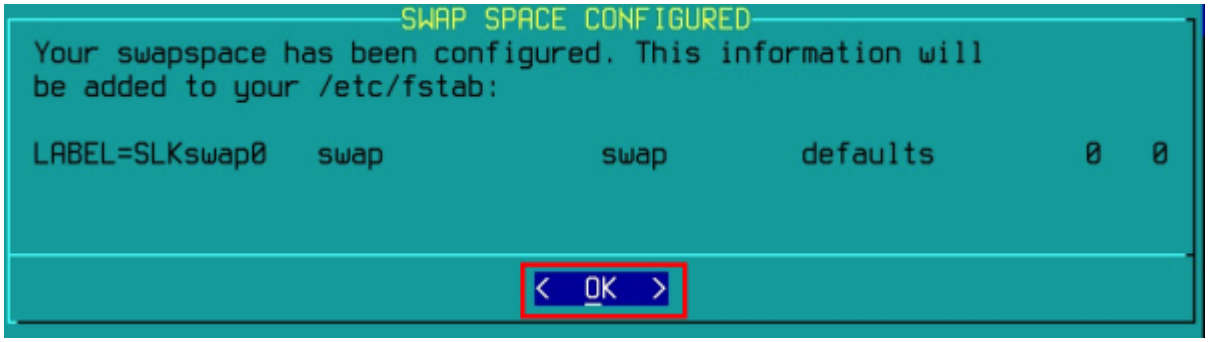


- Press ENTER to accept the default offered

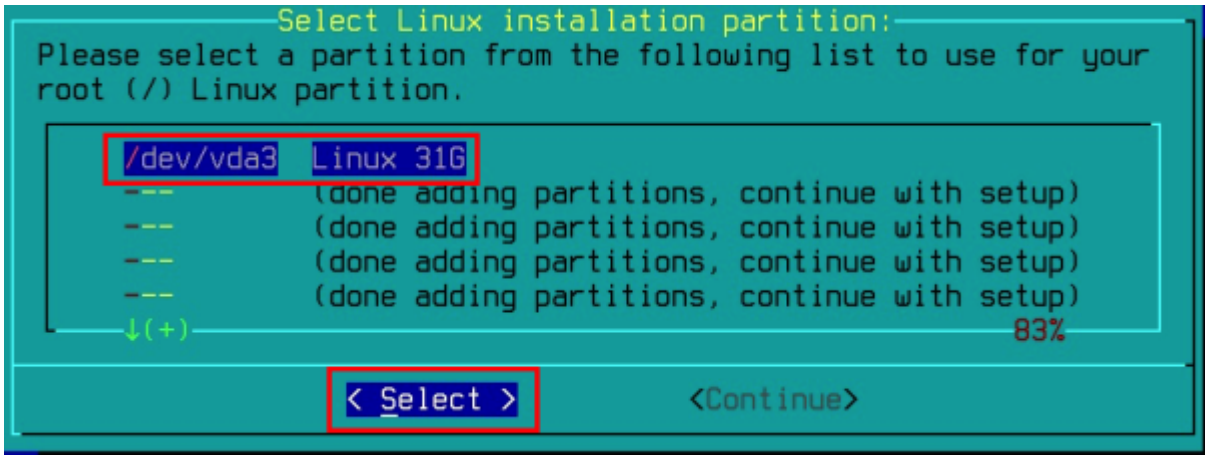


- Choose No to scanning for bad blocks

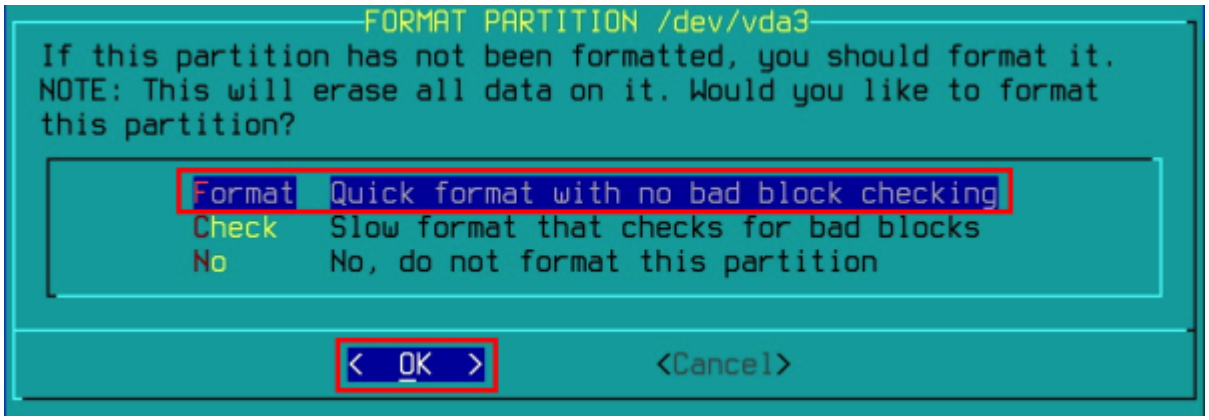




- Press ENTER to accept the default partition suggested for use as the root partition.

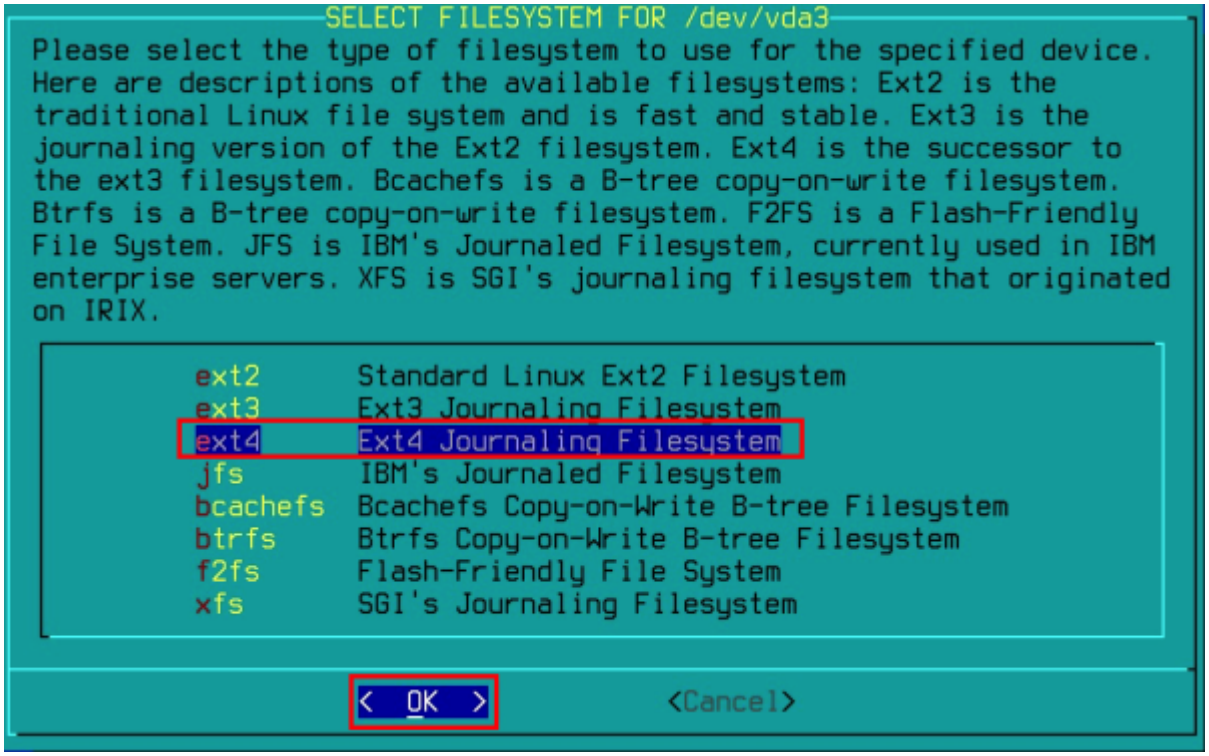


- Press ENTER to format the partition

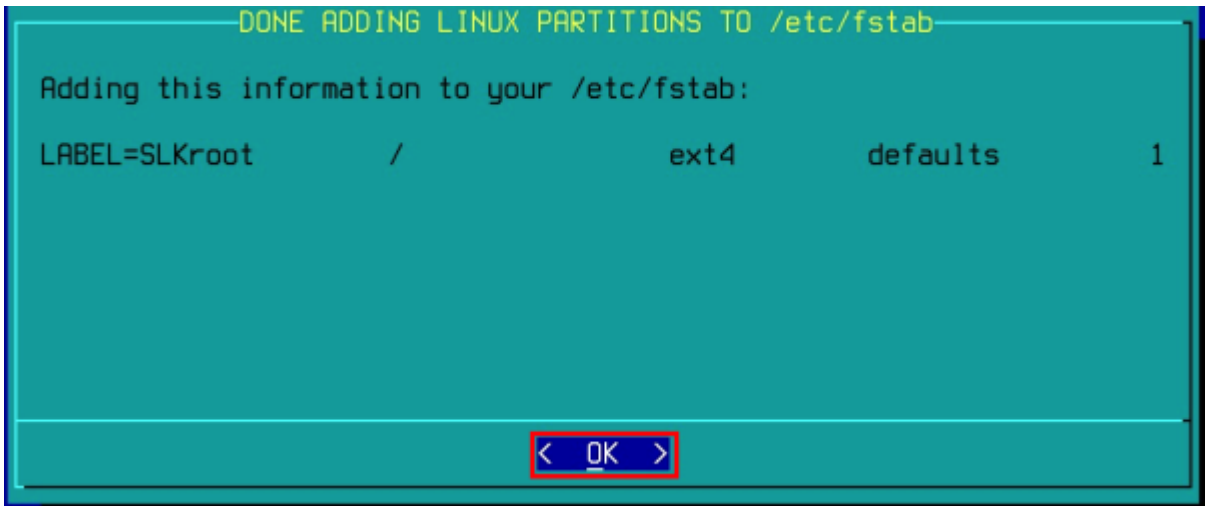


- Press ENTER to pick the suggested file system type:

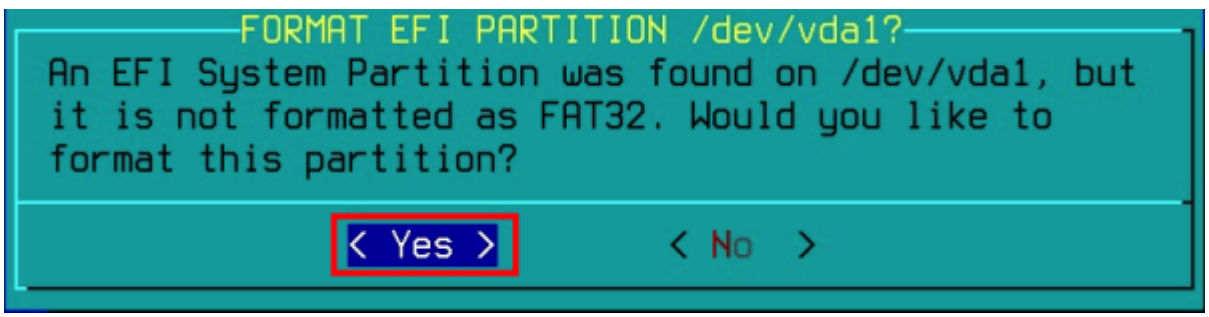
You can pick a different file system type if you wish



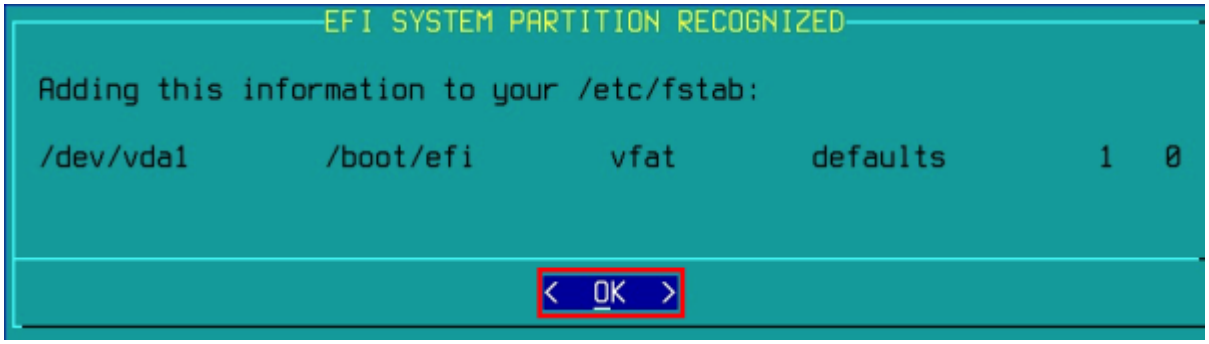
- Press ENTER to close the information dialog box:



- Press ENTER to format the EFI partition:



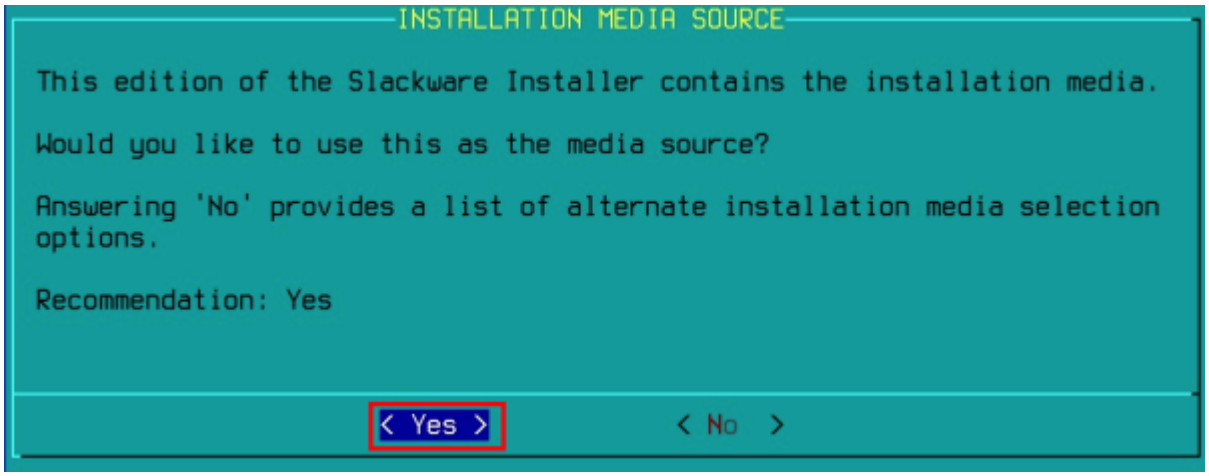
- Press ENTER to close the information dialog box:



Selecting the Installation Media

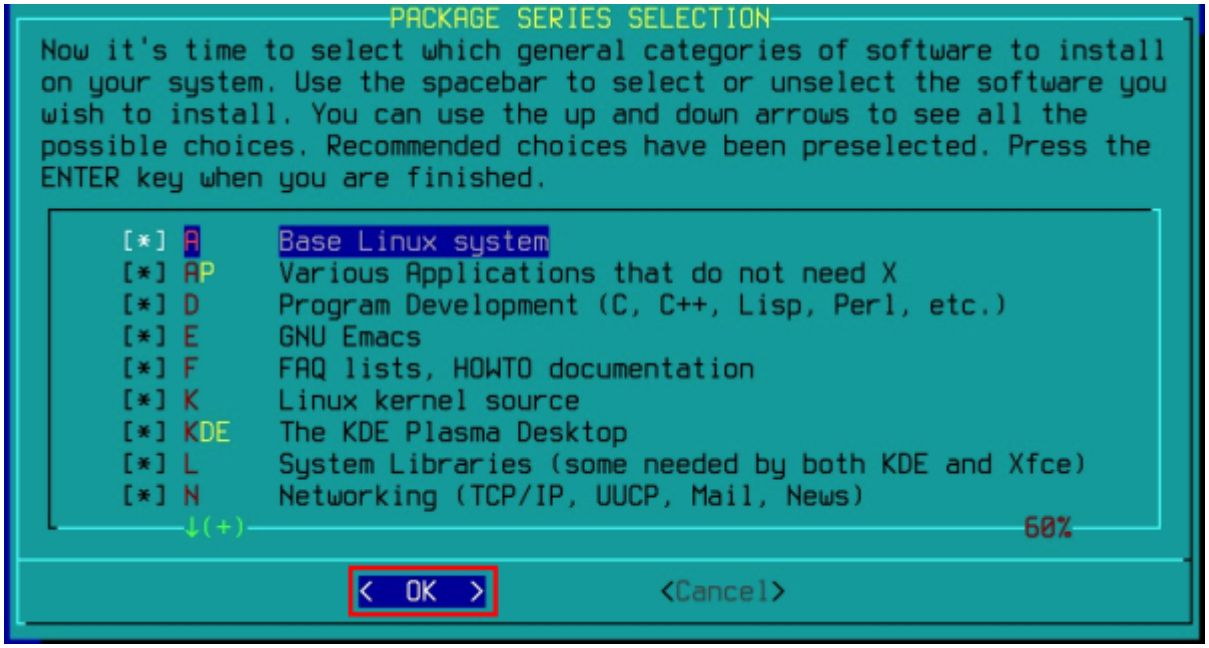
This edition of the Slackware AArch64 Installer contains the full set of Slackware packages.

Whilst other media installation (locally mounted directory, NFS, HTTP and others) can be selected by choosing No, the recommendation is to accept the defaults, installing the packet set bundled with the Slackware Installer.

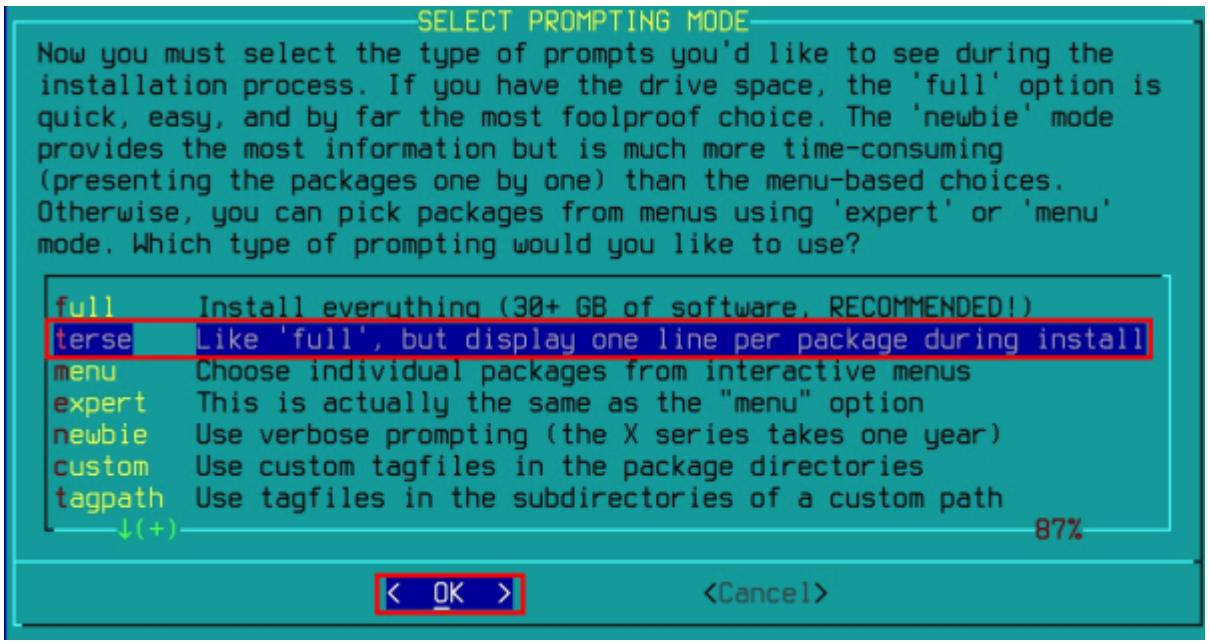


Selecting the Packages

Slackware packages are organised into loose groups called “series.” If you're an experienced Slackware user, you can customize which series to install. However, to keep things simple and avoid potential issues with dependencies (as Slackware does not manage them automatically), it's recommended to install all packages—at least for the initial setup.



- Press ENTER to proceed



- Press ENTER to accept the suggested prompting mode

Installation begins

The packages will now install. This takes approximately 10 minutes.

```
dcron-4.5-aarch64-13: Dillon's Cron daemon ..................................... [ 200K]
devs-2.3.1-aarch64-2: system device files ...................................... [ 5.0M]
dialog-1.3_20250116-aarch64-1: display dialog boxes from shell scripts ......... [ 570K]
dosfstools-4.2-aarch64-2: tools for working with FAT filesystems ............... [ 310K]
dracut-106-aarch64-1: creates an initrd for the kernel ......................... [ 2.7M]
e2fsprogs-1.47.2-aarch64-1: ext2/3/4 filesystems utilities ..................... [ 7.2M]
ed-1.21-aarch64-1: text editor ................................................. [ 150K]
efibootmgr-18-aarch64-1: tool to modify UEFI boot entries ...................... [ 190K]
efivar-20201015_cff88dd-aarch64-1: library and utilities to handle UEFI ........ [ 430K]
elogind-255.17-aarch64-1: logind extracted from systemd ........................ [ 5.4M]
elvis-2.2_0-aarch64-7: ex/vi text editor ....................................... [ 3.0M]
etc-15.1-aarch64-14: system configuration files ................................ [ 140K]
eudev-3.2.14-aarch64-2: dynamic device directory system ........................ [ 9.5M]
exfatprogs-1.2.8-aarch64-1: exFAT filesystem utilities ......................... [ 460K]
f2fs-tools-1.16.0-aarch64-2: Flash-Friendly File System ........................ [1010K]
file-5.46-aarch64-2: a utility to determine file type .......................... [ 2.9M]
findutils-4.10.0-aarch64-1: utilities to locate files ......................... [ 2.1M]
floppy-5.6-aarch64-1: floppy disk utilities .................................... [ 630K]
gawk-5.3.1-aarch64-1: pattern scanning and processing language ................. [ 4.3M]
gettext-0.24-aarch64-1: internationalization framework ......................... [ 2.0M]
```

In-built Entertainment

If you're short of things to occupy yourself whilst the packages install, you can play the bundled `bricktick` game!

- Hold down the `alt` key and press the `RIGHT arrow` key.

```
Please press Enter to activate this console. _
```

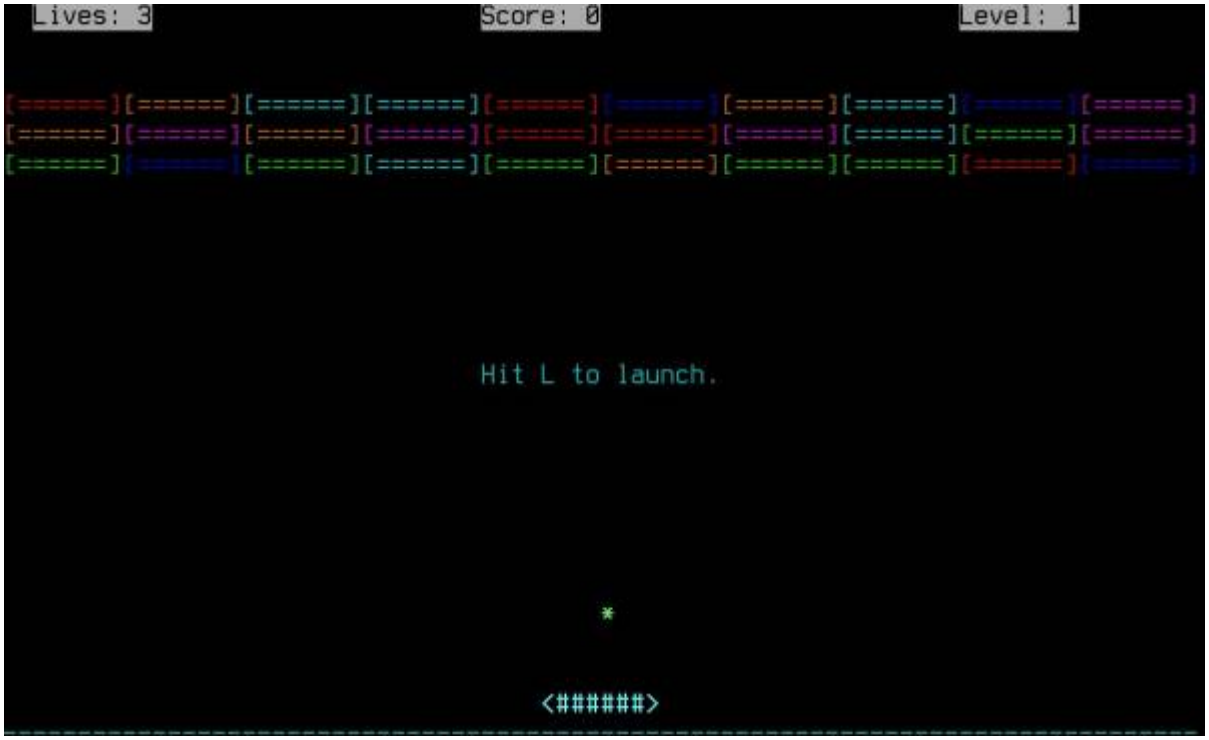
- Press ENTER

```
root@slackware:/# bricktick
```

- Type `bricktick` and press ENTER

Read the on-screen instructions and begin playing!

```
Welcome to Bricktick v0.9!
Charms:
% is +1,000 score.
@ is +1 lives,
# is 10 second slow ball.
* is 3 second nuke mode.
Ingame, press 's' to save a game and 'o' to load it.
Press space to start a game, 'o' to load one, or ESC to exit._
```



When you've had enough, and would like to check on the status of the installation:

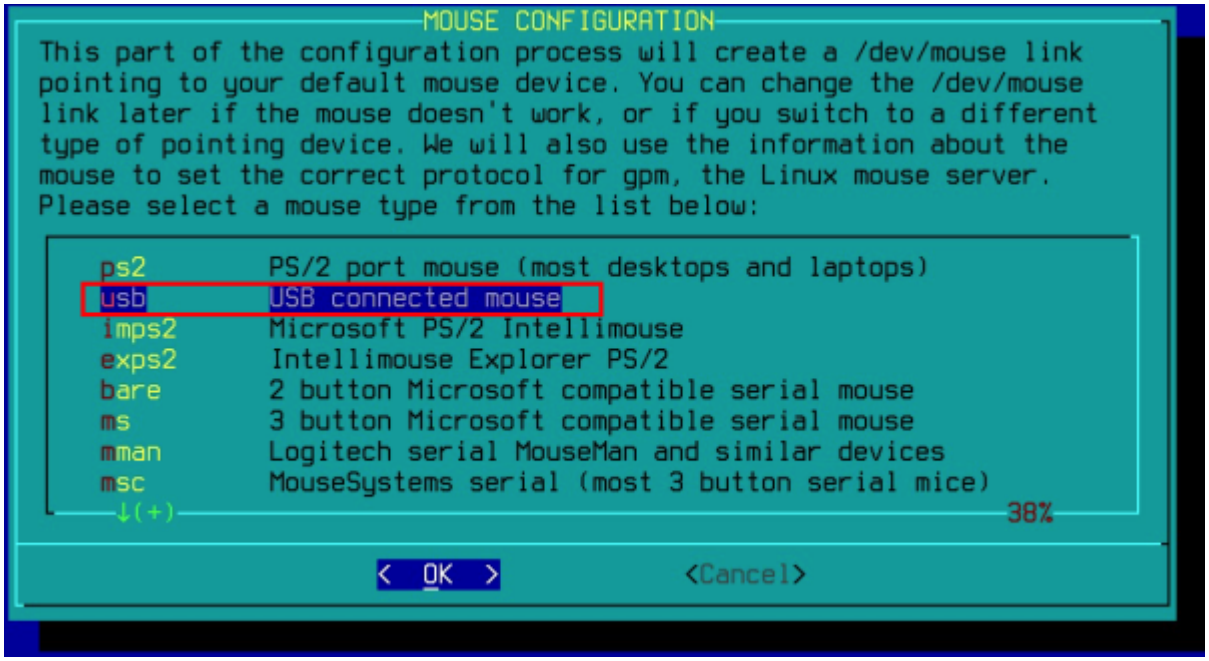
- Hold down the `alt` key and press the `LEFT arrow` key.

Post Installation Configuration

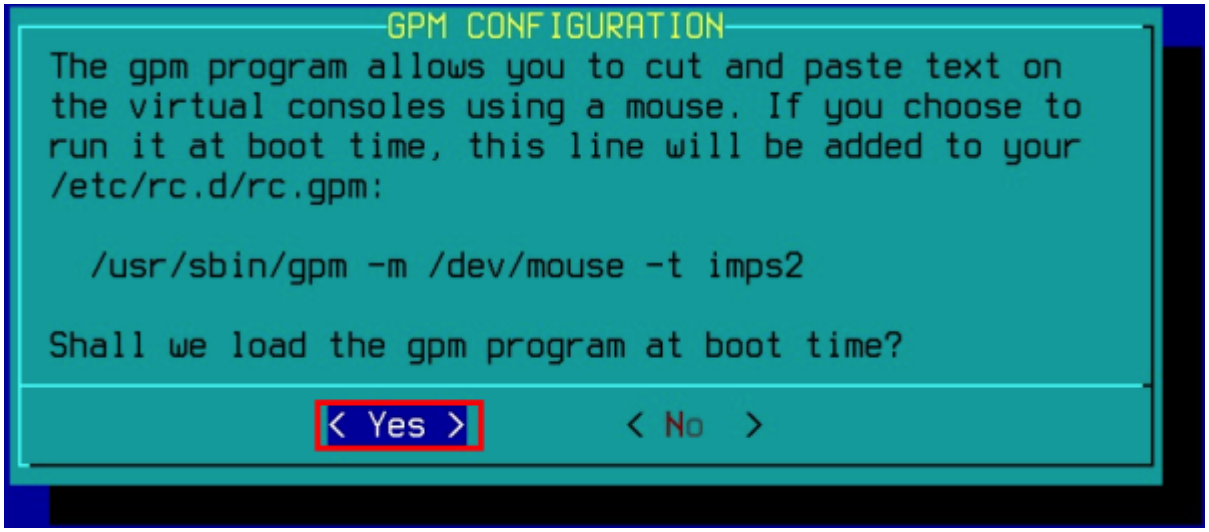
Once the packages have completed installing, there are some post configuration actions:

Mouse Configuration

- Select USB mouse and press ENTER

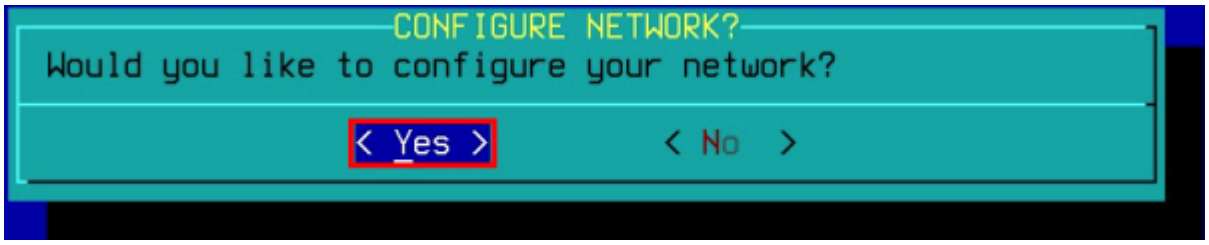


- Choose Yes to load the mouse driver for the Linux virtual console

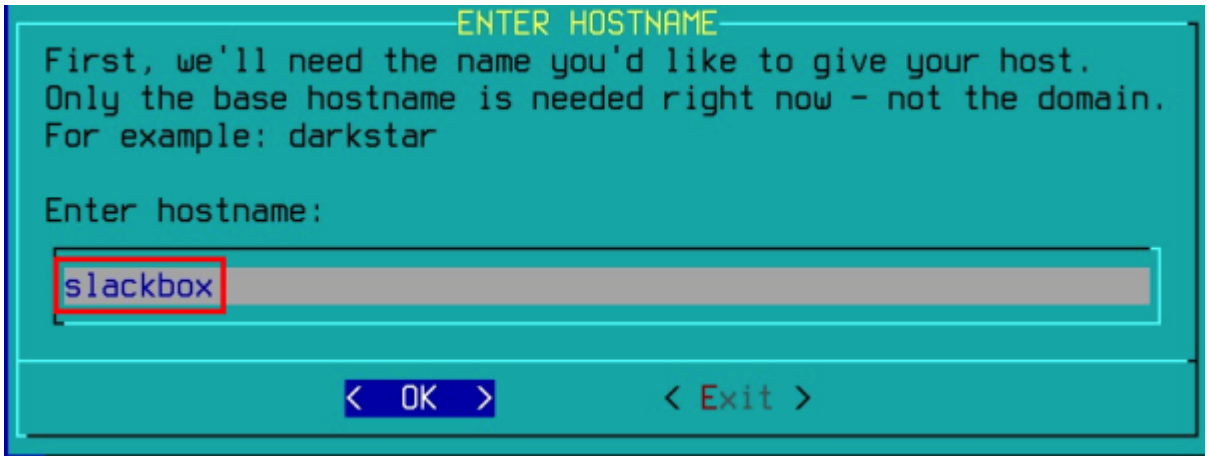


Network Configuration

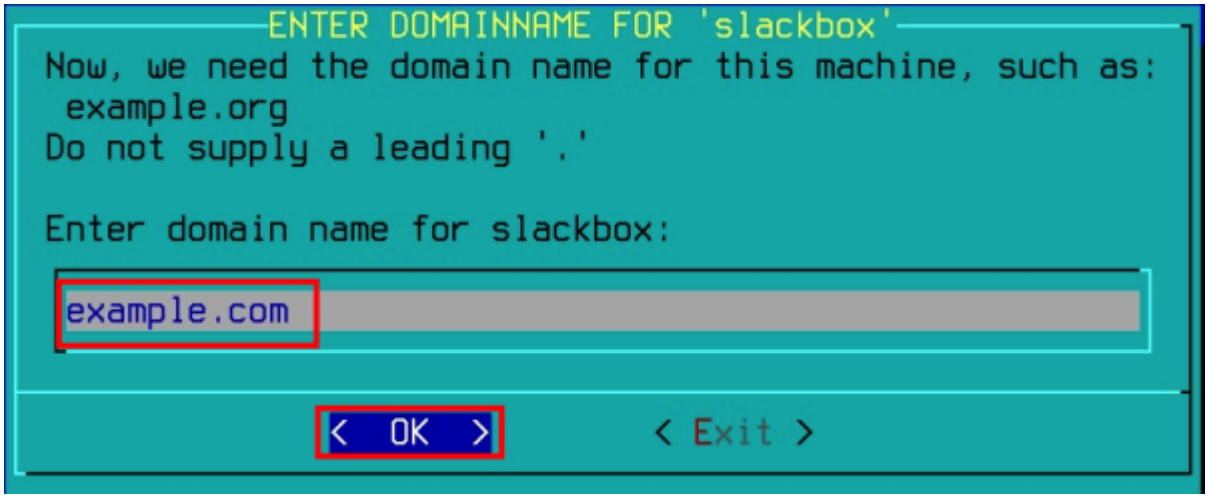
- Press ENTER to configure the network



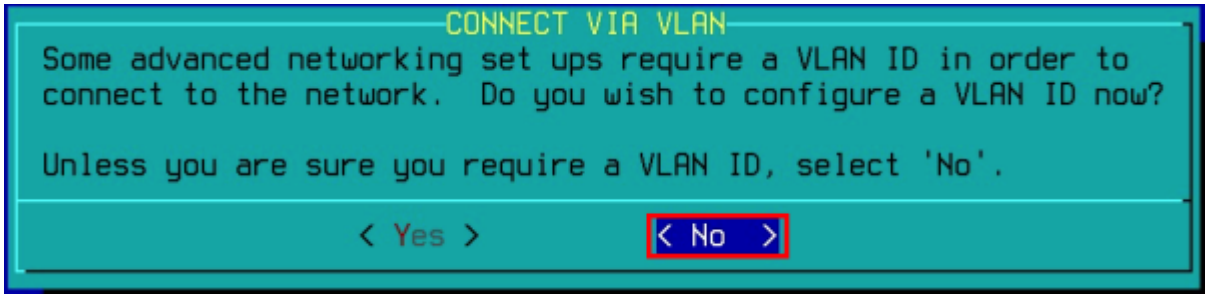
- Enter a host name for this machine



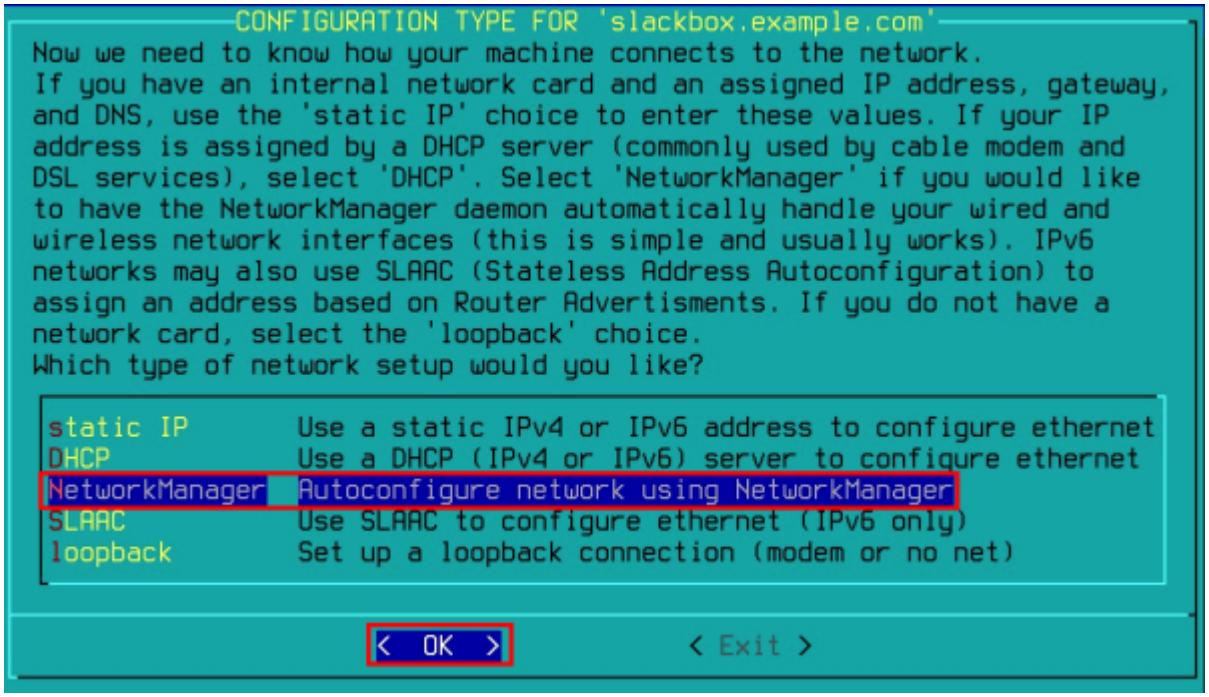
- Enter a domain name for this machine. If you don't have a local domain, pick `example.com`



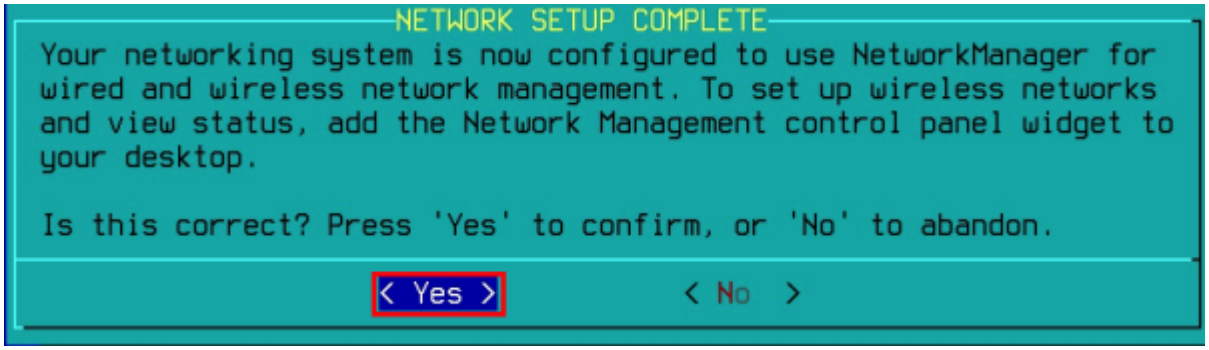
- Choose No in response to configuring a VLAN ID:

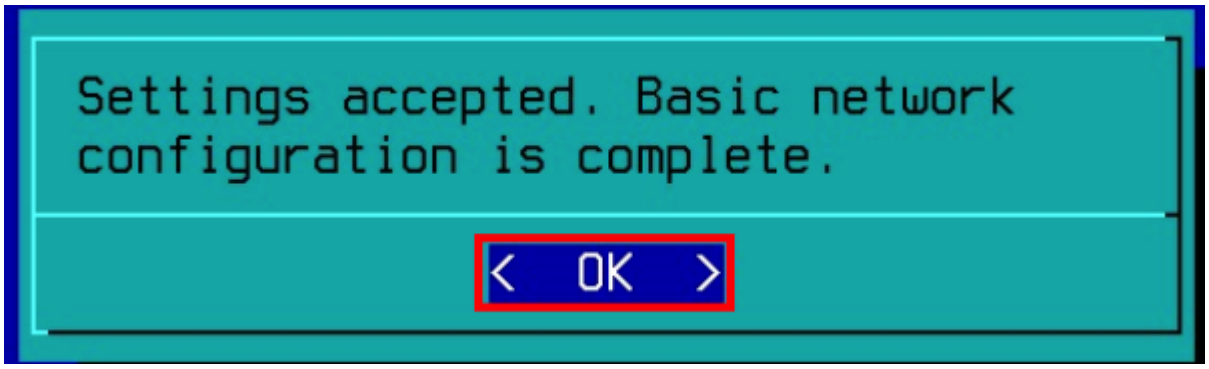


- Pick the default suggestions to configure the network using `Network Manager`



- Press ENTER on both dialog boxes to proceed



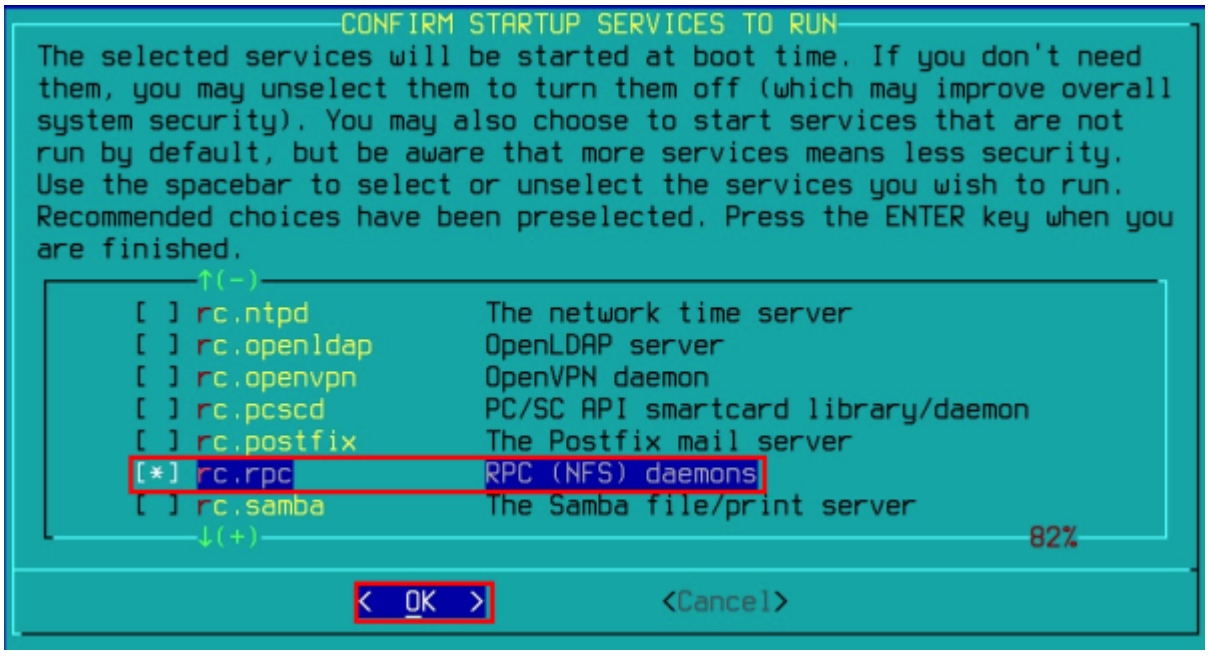


Automatically starting Services

Slackware uses the System-V init system, which uses a collection of `rc` (`run control`) scripts to control which daemons (services) start at boot time.

You can use the UP and DOWN arrow keys to move through the menu and press SPACE bar to select or de-select any services. In this example, we're selecting the `rc.rpc` service as we want to mount NFS shares.


You can run this script again within the OS

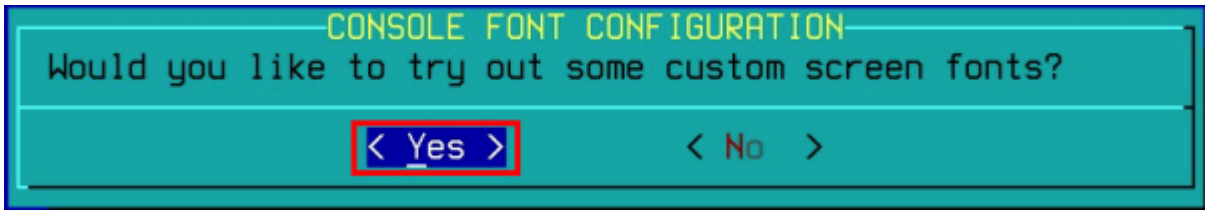


Virtual Console Font Configuration

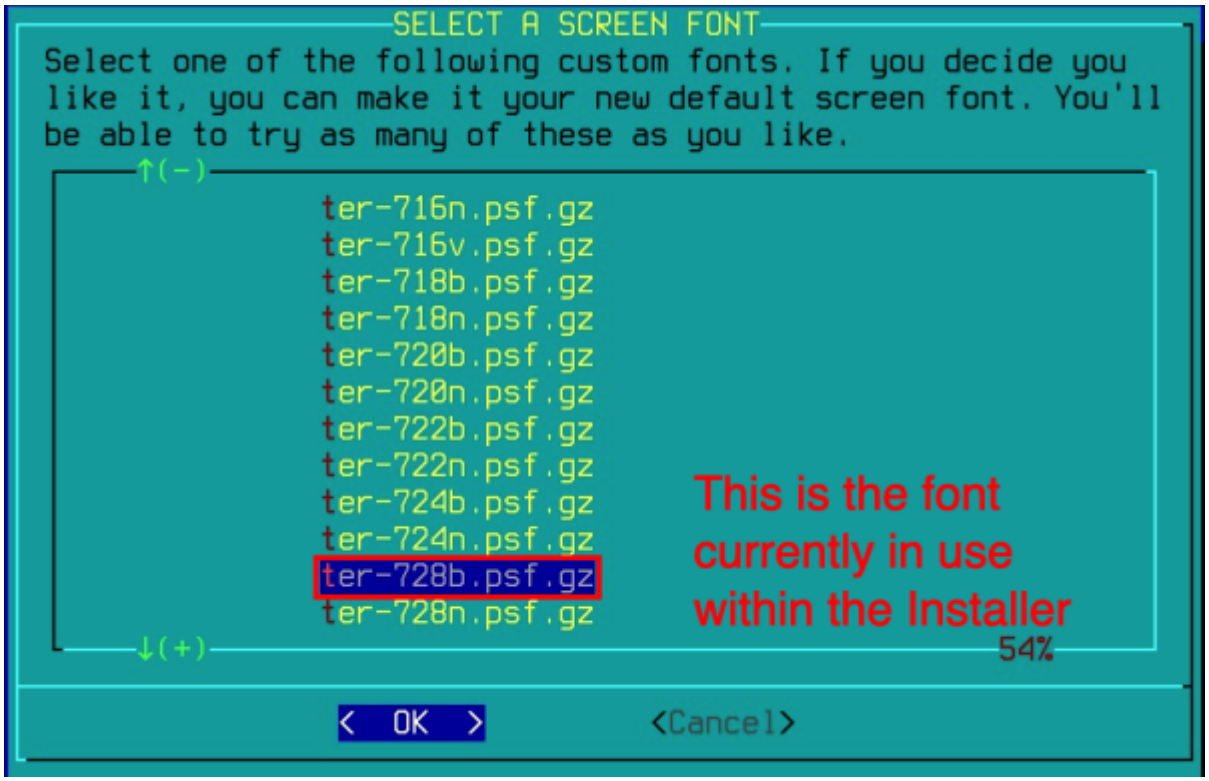
While optional, selecting a console font is recommended—otherwise, the default text may appear quite small and hard to read.

It's generally best to choose the font currently used by the installer, though you're free to select any font you prefer.

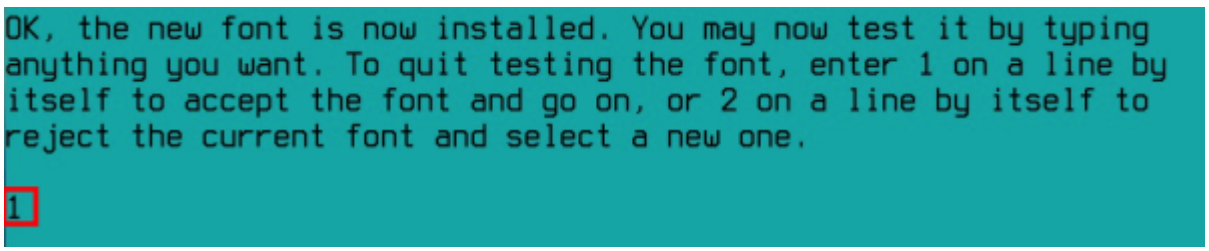
- Use the LEFT arrow key to move the selection to `Yes` and press ENTER



- Press `t` to jump to the fonts named 't', and the `DOWN` arrow to locate the font `ter-728b`

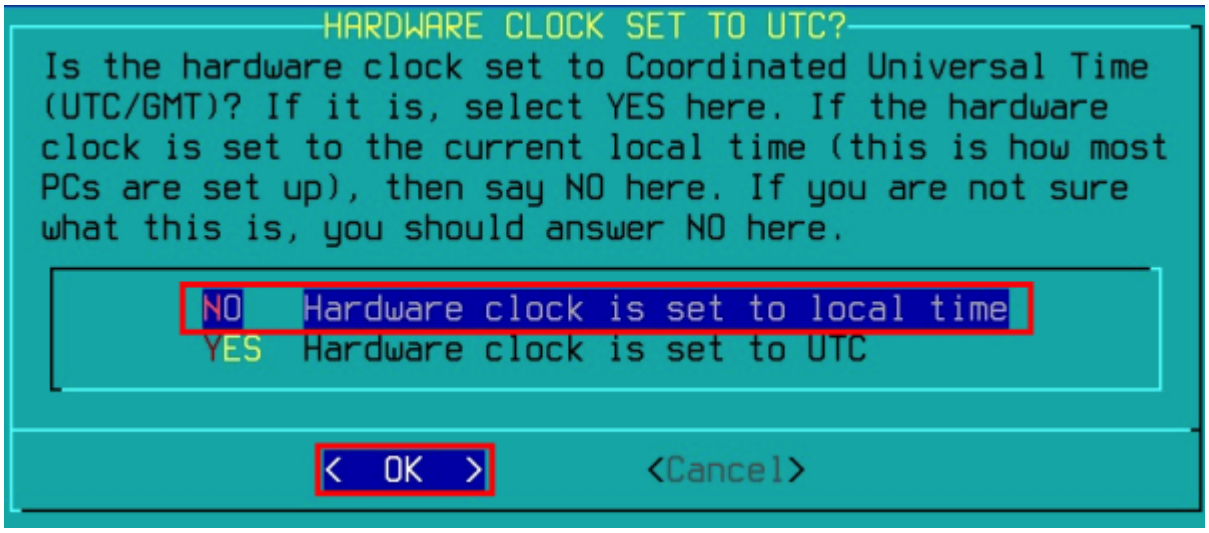


If you like this font, type 1 and press ENTER. If you'd prefer to use a different font, type 2 and press ENTER to return to the font selection menu.



Time Zone Configuration

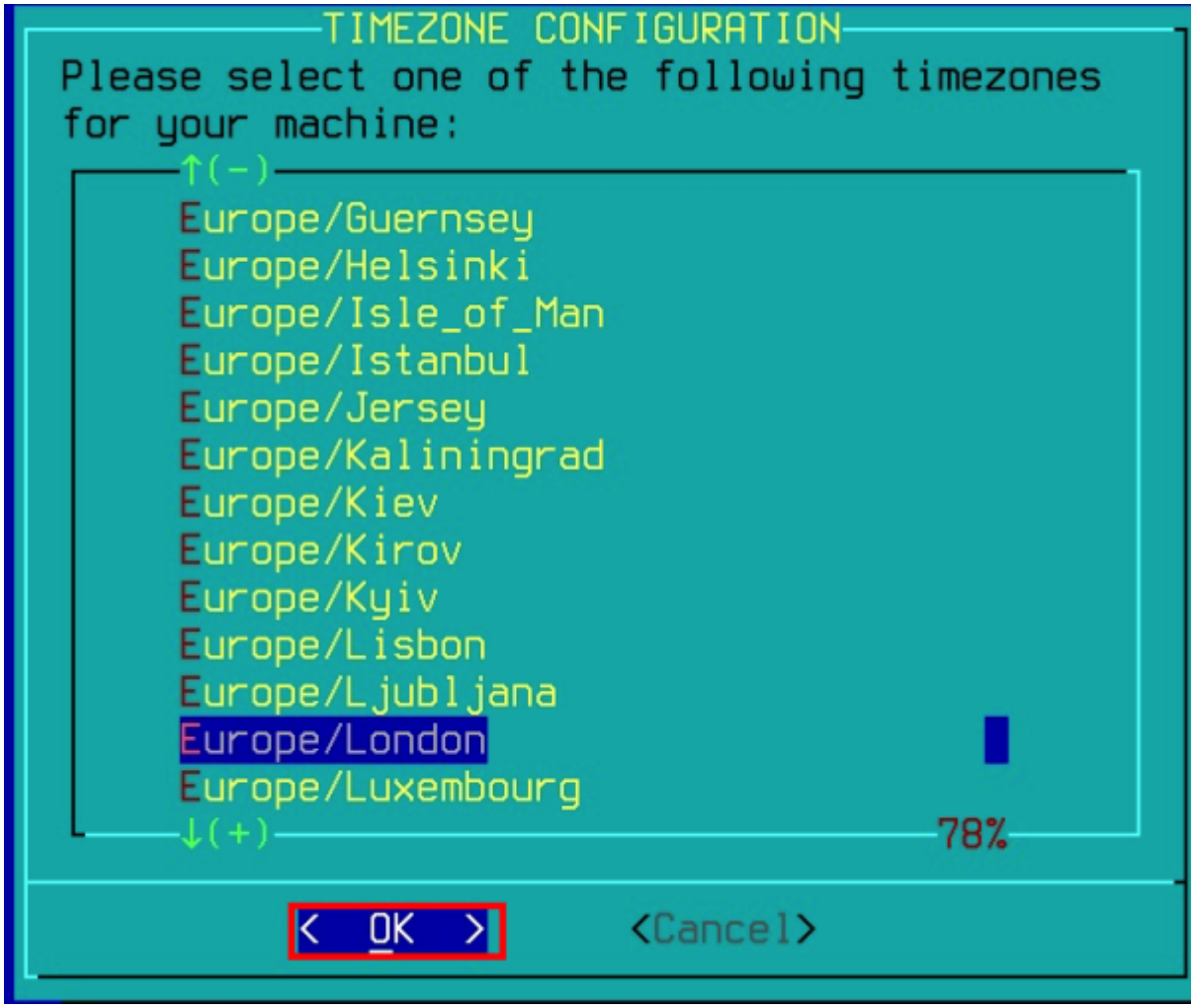
If you'd like to select a specific time zone (rather than UTC), select NO and press ENTER



Select the appropriate time zone for your locale:

You can jump sections within the menu by pressing the letter that corresponds to your region. For example, to select Europe/London, I can press e then use the DOWN arrow keys to locate the final entry

- Use the arrow keys to select your appropriate time zone
- Press ENTER to select the time zone

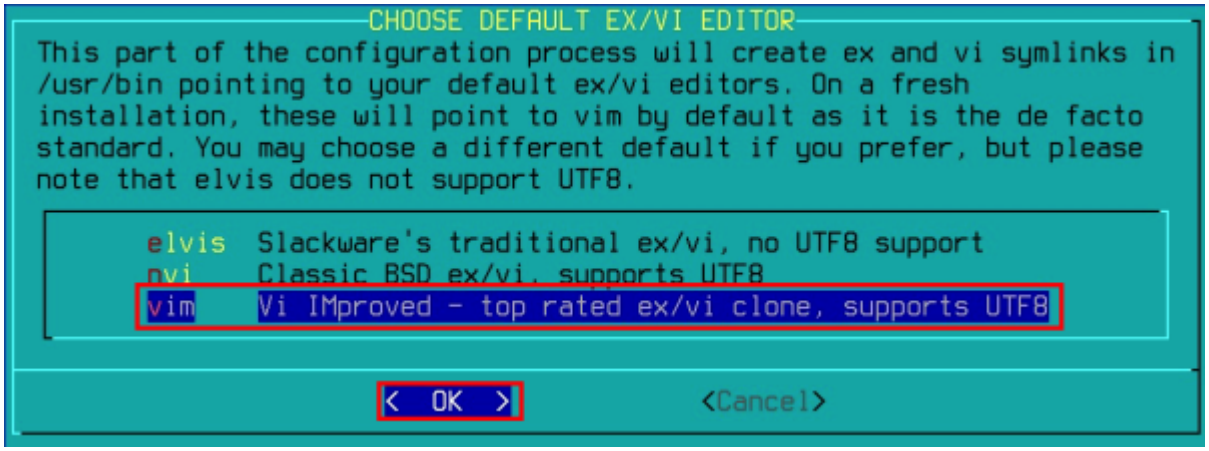


Configure the Default "vi" Editor

Select the implementation of the `vi` editor that you prefer. This is the default that will be used to edit `crontabs` and so forth.

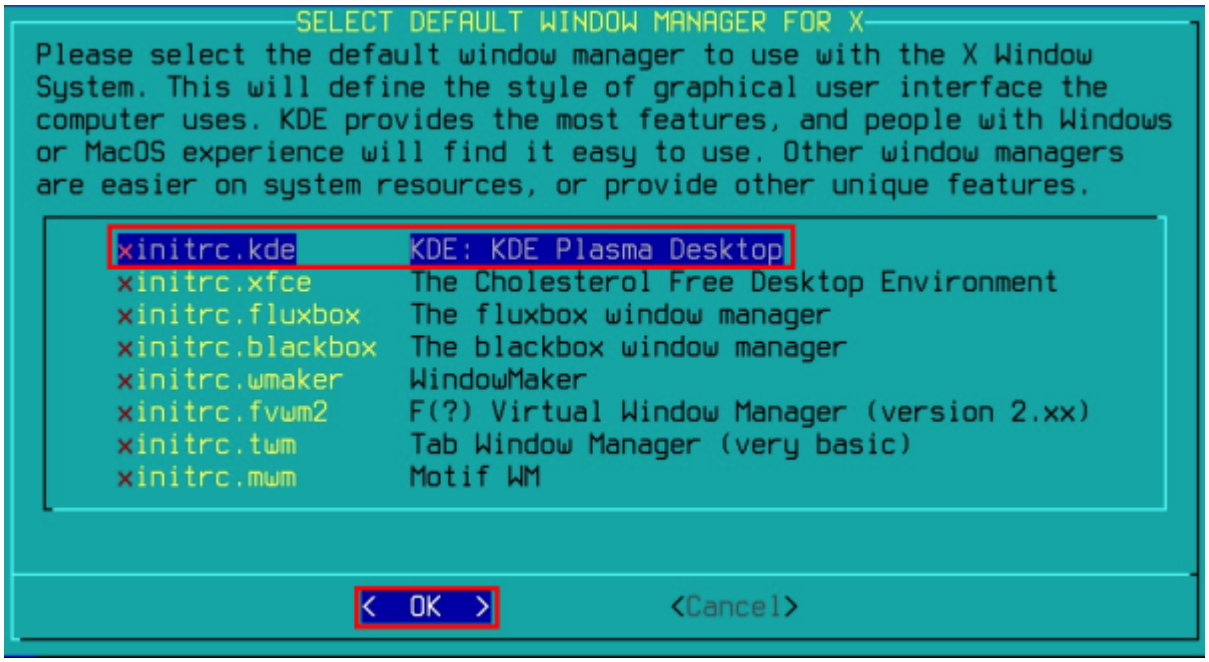
You can override the default editor by setting the environment variable `$EDITOR` within your shell profile

- Unless you have a preference, press ENTER to choose the default of `vim`



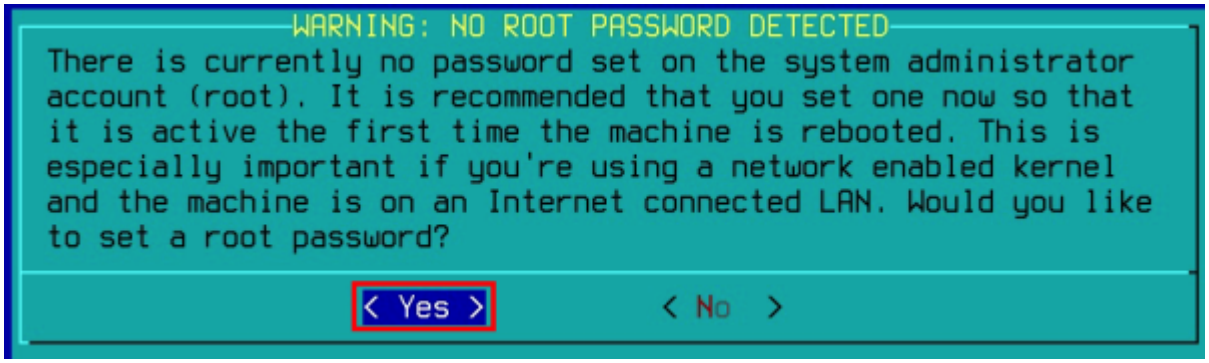
Configure the Window Manager for X

Select the window manager of your choice. KDE and Xfce are both solid options when running under paravirtualization, but you may prefer one of the classic lightweight window managers if you're looking for a more minimal setup.

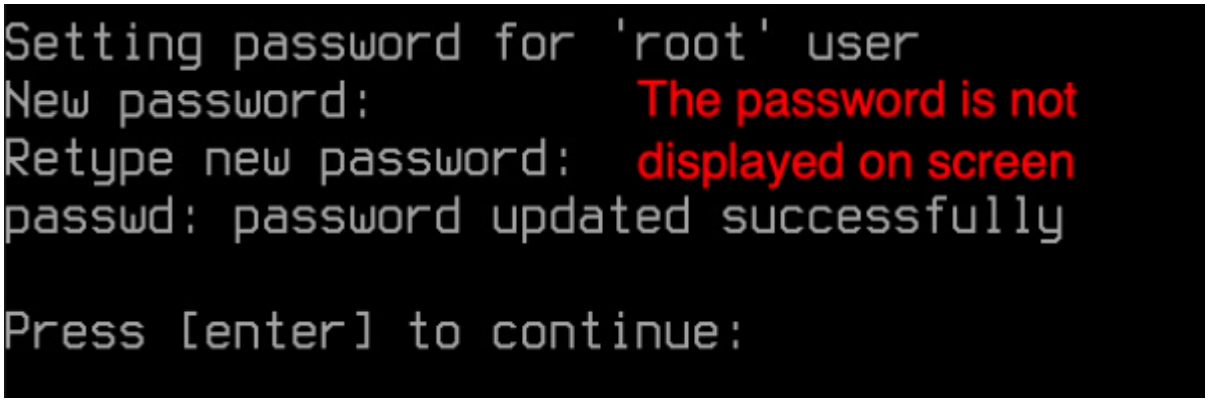


Set the root Password

- Press ENTER to set the root password

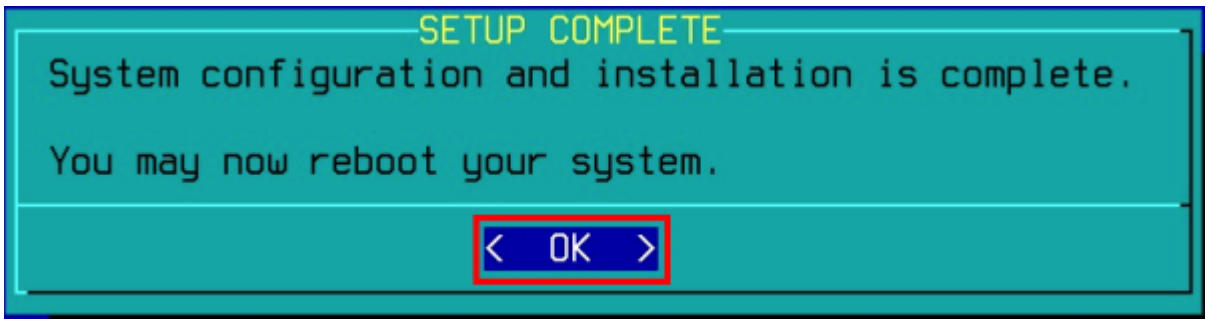


- Type the password in once, press ENTER
- Re-enter the password to confirm it, and press ENTER

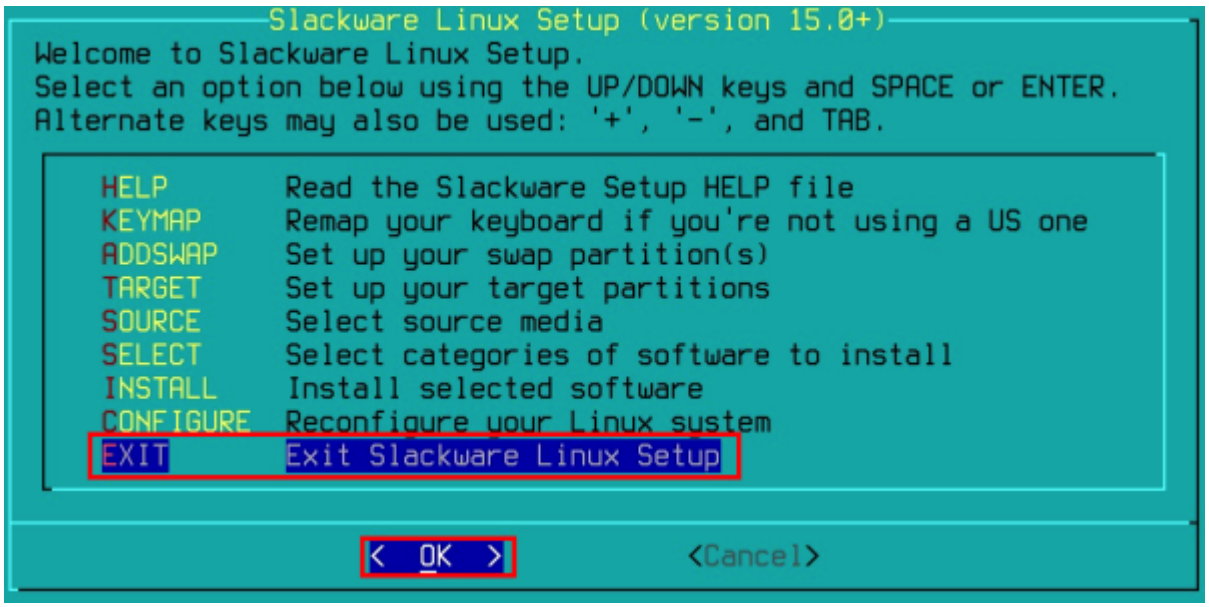


Complete the Setup

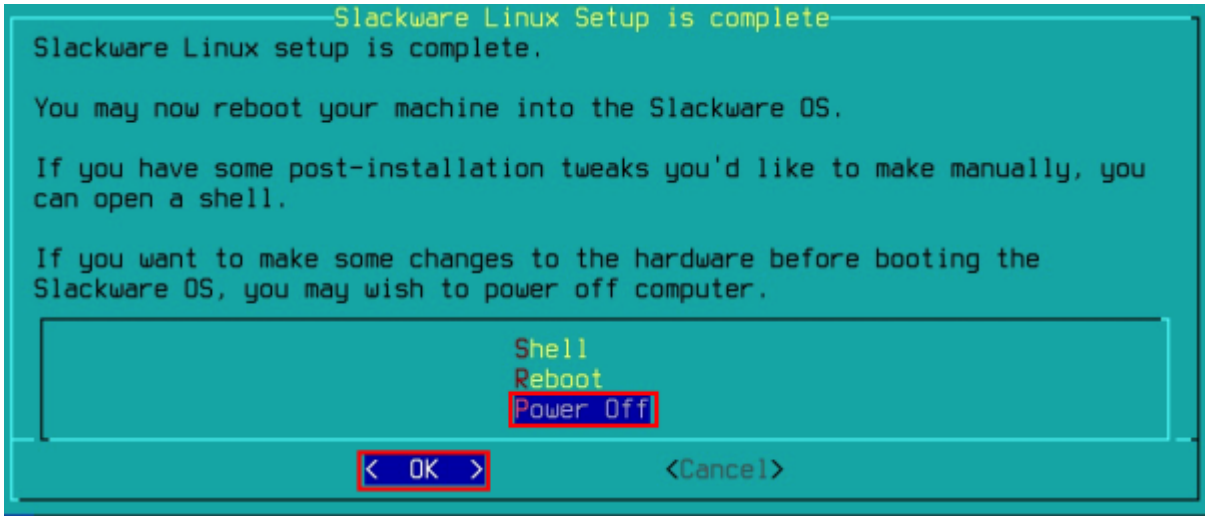
- Press ENTER to proceed



- Select EXIT from the menu
- Press ENTER



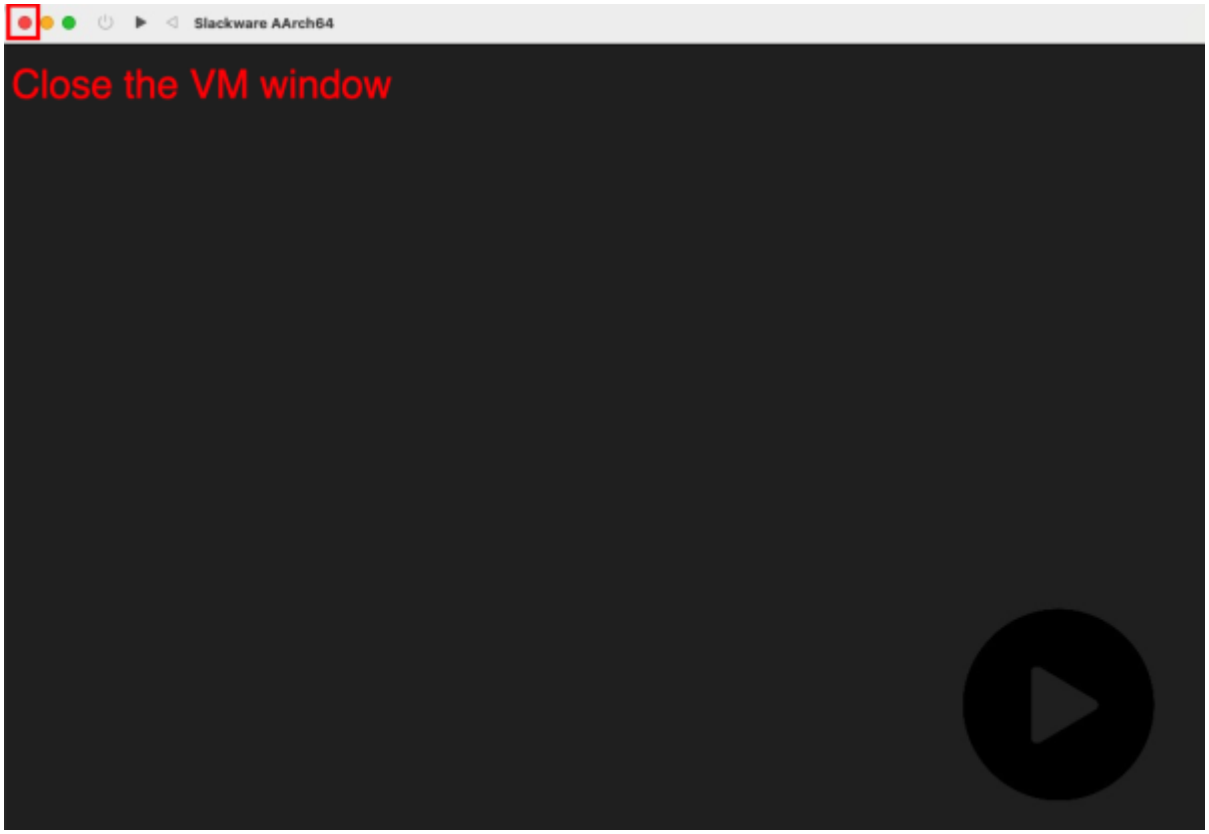
- Select `Power Off`
- Press ENTER



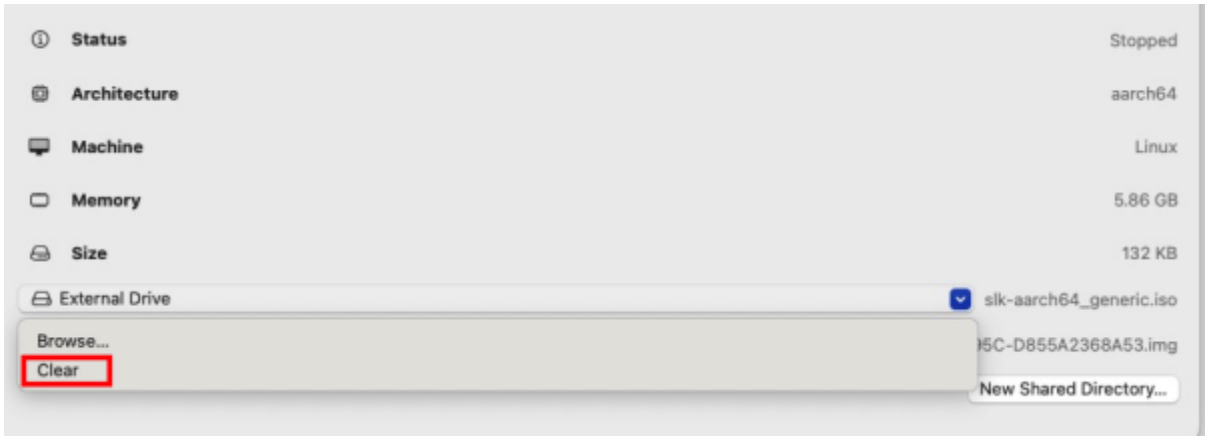
Configure UTM to boot the Operating System

Now that the Slackware OS has been installed, there are a couple of final changes to make within UTM:

- Close the VM window:

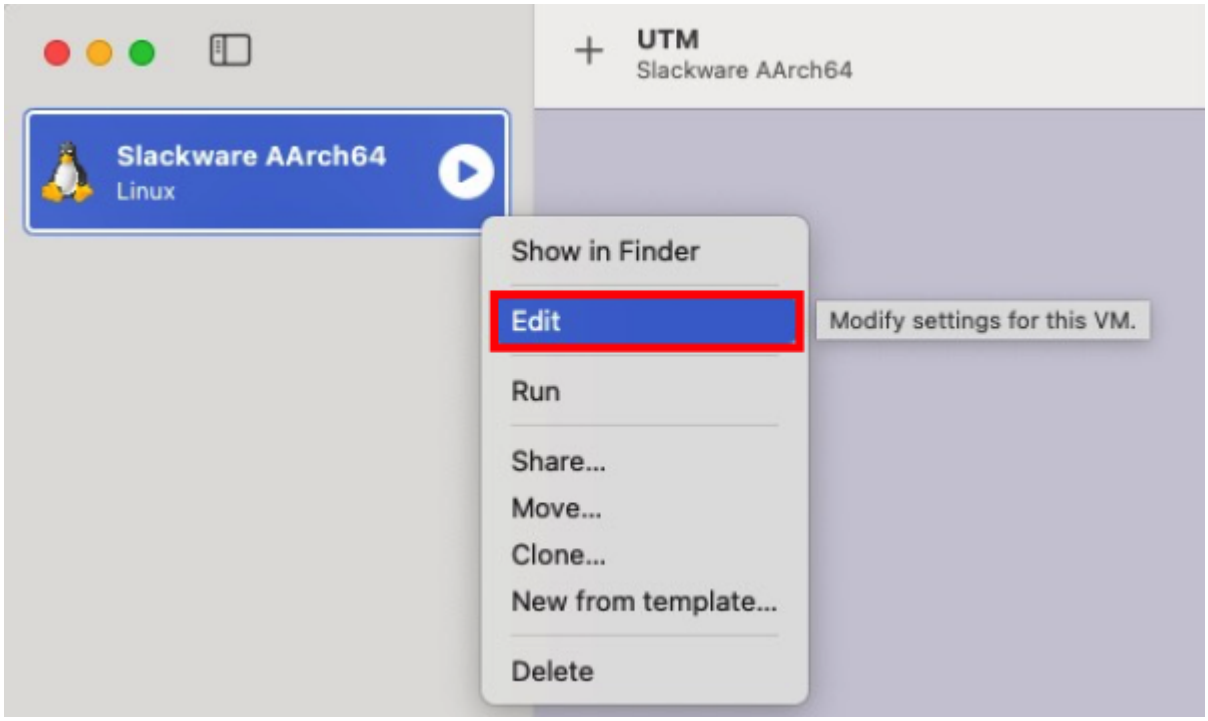


- Clear the Slackware Installer ISO so that the OS will boot:

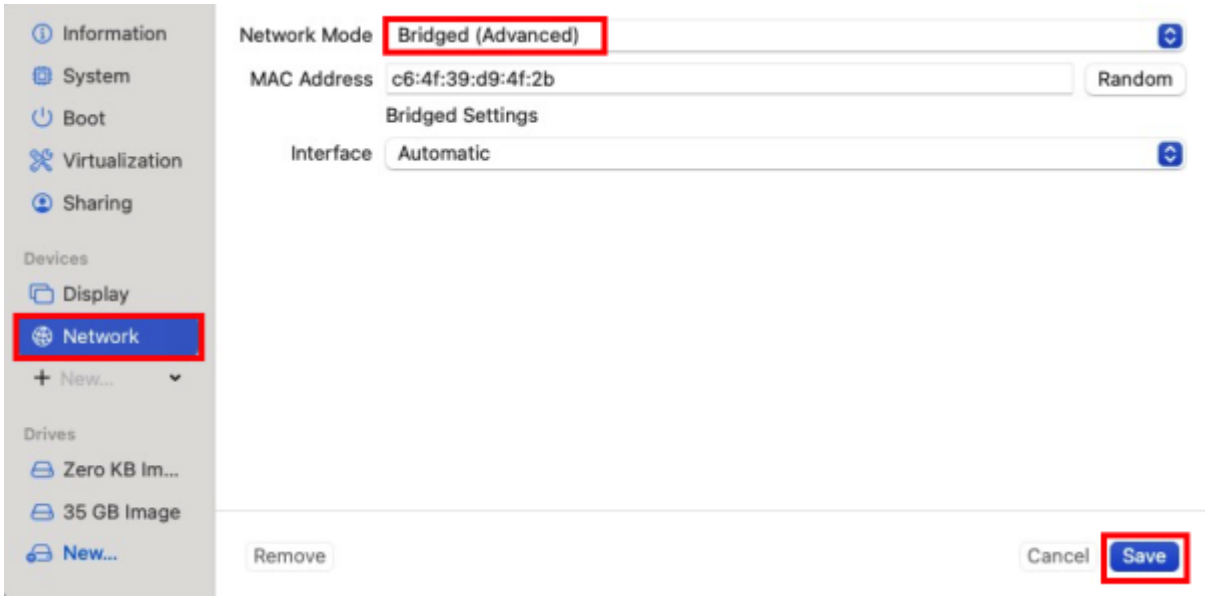


We will use `Bridged` mode for the VM's networking, otherwise the DNS doesn't work. You may use the default of `Shared Network` if you want to figure out how to make it work.

- Right-click on the VM and select `Edit`



- Click on the `Network` settings
- Choose `Bridged (Advanced)` as the `Network Mode`
- Click `Save`



Boot Slackware

You can now boot the Slackware Operating System.

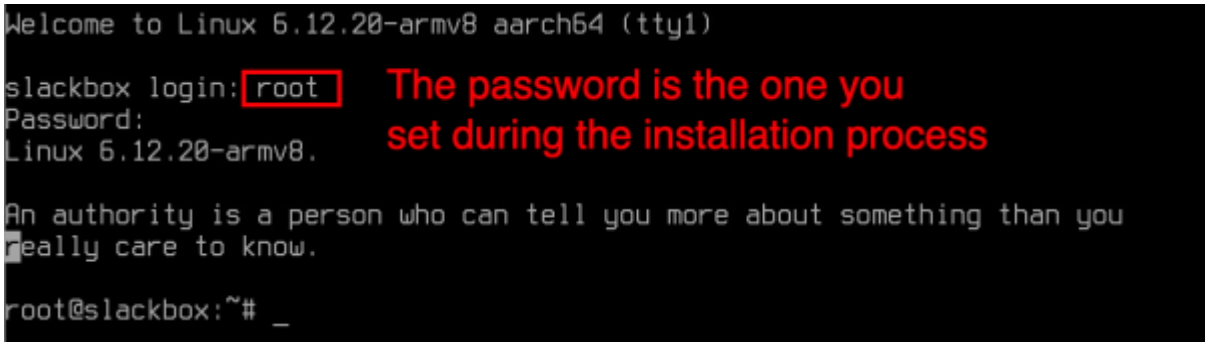
- Click on the `Play` icon

- Press ENTER on the `Slackware` menu option

The `UEFI Firmware Settings` option is unsupported in UTM

Login as root

You can now login as `root` and perform some post installation tasks.



Post Installation Configuration

There are a few post-installation configuration tasks to complete.

Add a plebeian user

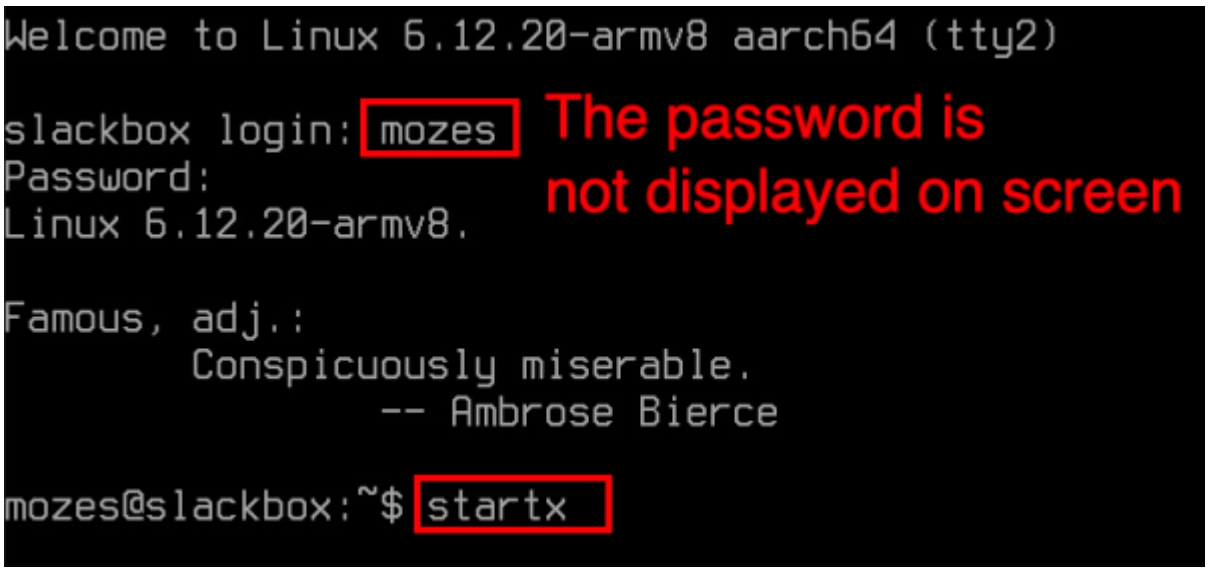
You should add a plebeian (non-root) user using the `adduser` tool.

This is documented here.

Starting X

Once you have added your plebeian user, you can login to another virtual console:

- Hold down the ALT key and press the RIGHT arrow key



- Login with your plebeian user
- Type `startx` and press ENTER

Managing Slackware

Setting up UTM shared directories

If you've created any shared directories (such as UTM sharing directories on the host macOS computer, made available within the Slackware guest), you can set them up now:

As the `root` user, within Slackware:

Make a mount point

```
$ mkdir /mnt/utm
```

Add the mount point to /etc/fstab

```
$ echo "share   /mnt/utm        virtiofs        rw,nofail       0       0"
>> /etc/fstab
```

Mount it

```
$ mount /mnt/utm
```

Verify

You can see the`slackware` and `Desktop` directories shared from the macOS host:

```
root@apple-utm:~# ls /mnt/utm/
```

```
Desktop/  slackware/
```

Keeping the Slackware OS up to date

One of the preferred tools to keep your system up to date is slackpkg.

Upgrading the Kernel: In Slackware AArch64, you simply upgrade the Kernel packages and reboot - no manual steps are required

Loading Additional Linux Kernel Modules within the OS Proper

Often Kernel modules for discovered hardware will be automatically loaded, but occasionally you will need to manually configure the loading of some modules.

```
/etc/rc.d/rc.modules.local
```

This file is a shell script that is run as one of the last steps before the OS has fully booted. You can enter `modprobe` commands here to load the specific modules you require.

Configuration files within the directory `/lib/modprobe.d/` can be used to configure the parameters of the modules. Existing files within that directory serve as reference examples should you need them.

Loading Additional Linux Kernel Modules early in the boot sequence

There are a number of peripherals that may require Kernel modules loading early on in the boot sequence. An example of this would be RTCs (Real Time Clocks) or storage controllers that are required to access the file systems on which the OS lives.

Usually you won't need to load modules early in the boot sequence. See the previous section about loading modules from within the OS Proper.

To load Kernel modules during the early boot sequence, read:

```
/boot/local/README.txt
```

As root, the easiest way to begin is by renaming the example script:

```
mv /boot/local/load_kernel_modules.post.sample
```

```
/boot/local/load_kernel_modules.post
```

Then add the appropriate module loading commands to: `/boot/local/load_kernel_modules.post` You can also add shell code here to initialise a peripheral - writing something to the peripheral's Kernel interface, for example.

Starting Services at Boot Time

During the installation, you configured which services to start at boot time.

You can start or disable other services either manually or by using `pkgtool`

Manually change the permissions

As `root`:

```
root@slackware:# cd /etc/rc.d
```

To enable CUPS:

```
root@slackware:/etc/rc.d# chmod +x rc.cups
```

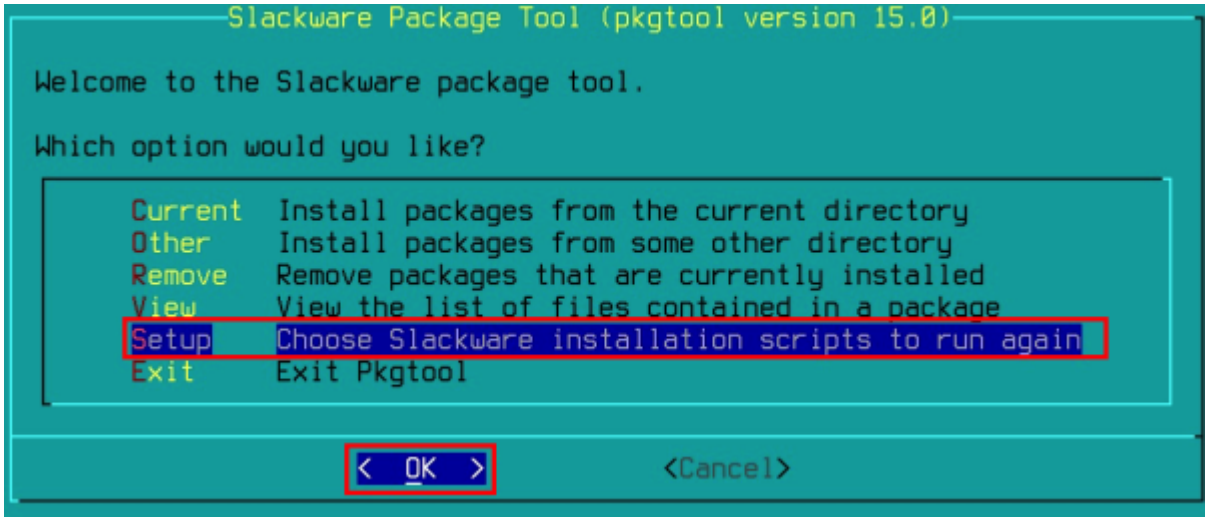
To disable CUPS:

```
root@slackware:/etc/rc.d# chmod -x rc.cups
```

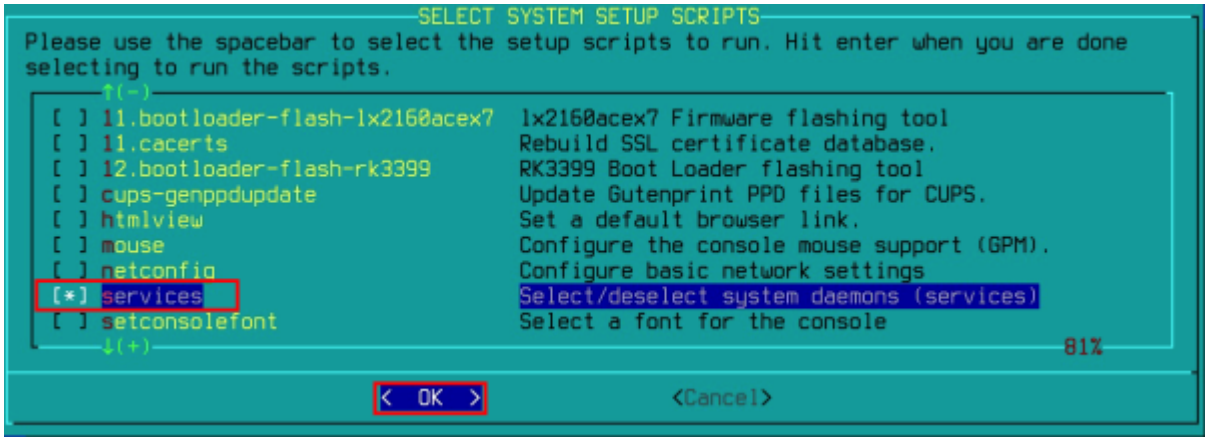
Use ''pkgtool''

```
root@slackware:# pkgtool
```

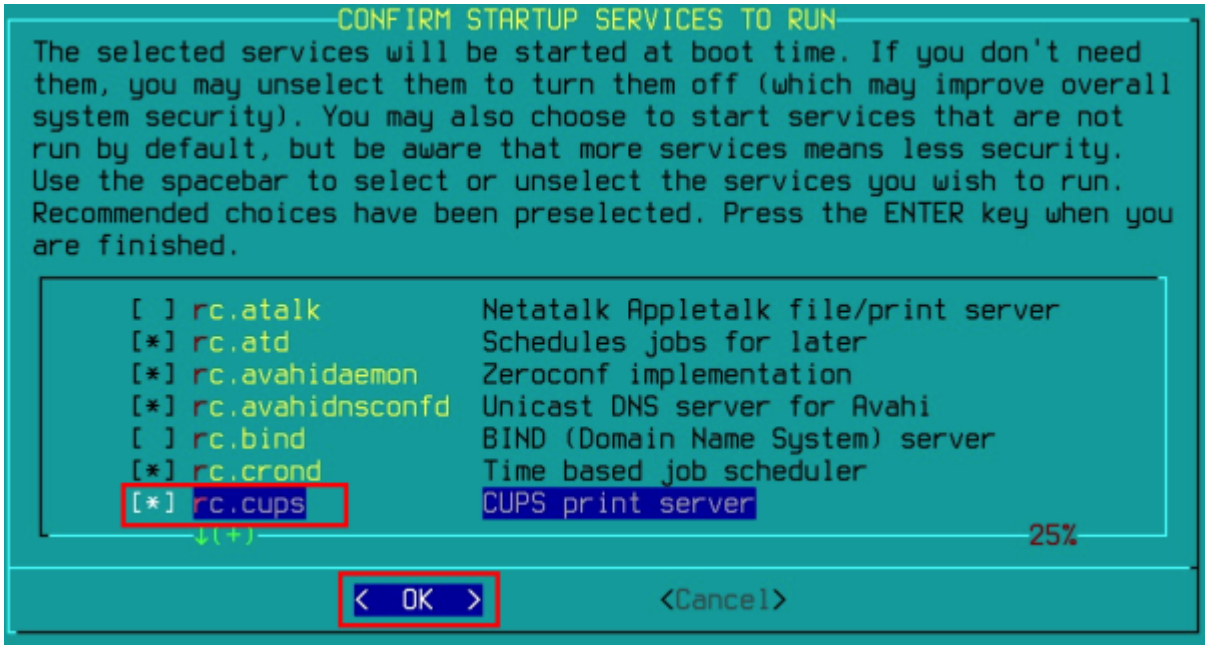
- Select `Setup` from the menu and press ENTER



- Select `services` and press ENTER



- Choose the service you want and press ENTER



Starting a Graphical Login Manager by Default

If you prefer to use a graphical login manager, you can configure the default runlevel as 4:

```
su -
sed -i 's?id:3:?id:4:?g' /etc/inittab
reboot
```

Customising the Slackware Linux Kernel

If you'd like to customise the Linux Kernel, the easiest way is to follow the HOWTO guide and use the Slackware AArch64 Kernel build script to create new packages.

Reducing Boot Time

This section isn't relevant for virtualized platforms, as the OS boots extremely quickly. However, it's included here for completeness, as it appears in the guides for other hardware models.

Slackware AArch64 ships with a generic OS InitRD (Operating System Initial RAM Disk - the environment that prepares the machine to boot the Operating System Proper), so as to support a wide range of Hardware Models.

However, this isn't the optimal setup once the Slackware OS has been installed because the generic OS InitRD typically exceeds 40MB, which in some cases can add a couple of seconds to the boot time whilst it's loaded from the SD card.

The `os-initrd-mgr` (Operating System Initial RAM Disk Manager) tool has an option to synchronize the OS InitRD's Kernel modules with *only* those presently loaded within the Operating System.

To do this:

```
$ su -c 'os-initrd-mgr --sync-loaded-kmods' -  # note the final -
```

This option isn't the default, but you can make it so by following the instructions within `/etc/os-initrd-mgr.conf.sample`

This way when you upgrade the Kernel packages in the order described above, it'll automatically synchronize the modules.

os-initrd-mgr has a safety check to only proceed when the running Kernel and incoming Kernel are at the same major version and patch level.

For example, when running Linux 5.17.1, upgrading to 5.17.2 will work; but an upgrade of Linux 5.17.1 → 5.18.1 will require a reboot then to run os-initrd-mgr again to re-sync.

If at any point you want to revert to the generic OS InitRD, simply reinstall the `a/kernel` package (and unset the setting if you configured it in `/etc/os-initrd-mgr.conf`).

Installing extra Software

Slackware comes with a good base of software applications, but there are plenty more available in the Open Source Ecosystem.

The best way to add new software is to use the build scripts from SlackBuilds.org.

Useful Guides and Resources

- Slackware ARM Support Forum

Known Issues and Feature Gaps

- No currently known issues.

More details on the development Road Map here.