Microlinux Enterprise Server 14.1 Installation Guide

The following document provides step-by-step instructions to install and configure the Microlinux Enterprise Server 14.1. It assumes you already know how to install and configure Slackware Linux. Read everything carefully. The bits you skip will come back and bite you.

If you are not familiar with Slackware, head over to the *Slackware Documentation Project* at http://docs.slackware.com and read the following two articles in the section **Getting Started with Slackware** :

- **Slackware installation**
- **Configure your new Slackware system**

MLES installs on top of a carefully selected Slackware base and doesn't ship on its own installation ISO. All you need is a vanilla Slackware installation medium (DVD, CD-Rom, USB stick) and a working internet connection.

Install the base Slackware Linux system

Boot the Slackware installation medium : DVD, first CD or USB disk. Select your keyboard layout, login as root, partition your hard disk and start the Slackware installer.

```
root@slackware:/# setup
```

- **PACKAGE SERIES SELECTION** : unselect the **E**, **KDE**, **KDEI**, **XAP** and **XFCE** package groups. Even though MLES is supposed to be installed in runlevel 3 without a graphical interface, make sure to keep the **T** and **X** package groups, as they contain dependencies for other stuff ;
- **SELECT PROMPTING MODE** : **full** or **terse** ;
- **USE UTF-8 TEXT CONSOLE** : **Yes** ;
- **CONFIRM STARTUP SERVICES TO RUN** : accept the default selection ;
- **SELECT DEFAULT WINDOW MANAGER FOR X** : **xinitrc.twm** ;

Finish the base Slackware installation, exit the installer and reboot. Don't define a user yet. We'll wait until the MLES user profiles are installed.

Download the Microlinux scripts

Microlinux provides a few helper scripts to speed up the installation process. Download the file tree to your `/root` directory using the following command :

```
# cd
# git clone https://github.com/kikinovak/microlinux
```

Configure slackpkg

In its default configuration, the `slackpkg` package manager only works with the official Slackware repositories. We need to install and configure Matteo Rossini's nifty `slackpkg+` plugin to enable third-party repositories. To avoid the hassle, Microlinux provides a custom `slackpkg+` package that comes preconfigured for the Microlinux repositories.

Use the Links browser to navigate through the repositories :

```
# links http://www.microlinux.fr/microlinux/
```

On a 32-bit system, navigate to the `server-14.1-32bit/slackware/ap/` subdirectory. If your system is 64-bit, go to `server-14.1-64bit/slackware64/ap/`. Grab the `slackpkg+` package from the repository (use the **[D]** key in the Links browser to download it) and install it :

```
# installpkg slackpkg+-1.3.2-noarch-5_microlinux.txz
```

Edit `/etc/slackpkg/mirrors` and choose a Slackware mirror according to your geographical location, for example :

```
# /etc/slackpkg/mirrors
...
# GERMANY (DE)
ftp://ftp.fu-berlin.de/unix/linux/slackware/slackware-14.1/
...
```

On a 64-bit system :

```
# /etc/slackpkg/mirrors
...
# GERMANY (DE)
ftp://ftp.fu-berlin.de/unix/linux/slackware/slackware64-14.1/
...
```

Make sure you choose only one single mirror for Slackware stable.[1]

Update GPG keys :

```
# slackpkg update gpg
```

Update information about available packages :

```
# slackpkg update
```

[1] If you're using MLES in France, configure the ftp.fu-berlin.de or the mirror.switch.ch mirror. The admins at OVH can't seem to get their act together.

Trim and upgrade the base installation

The `server-14.1-32bit/tools/` and `server-14.1-64bit/tools` subdirectories each provide a basic `trim.sh` script that takes care of two things :

- install needed base packages ;
- get rid of unneeded base packages.

Trim your Slackware installation :

```
# cd microlinux/server-14.1-32bit/tools/
# ./trim.sh
```

Or :

```
# cd microlinux/server-14.1-64bit/tools/
# ./trim.sh
```

On a side note, the `trim.sh` scripts and respective package selections in the `32bit` and `64bit` subdirectories are symlinked and thus identical.

Now upgrade the base Slackware packages :

```
# slackpkg upgrade-all
```

Install the MLES package collection

Installing the full set of MLES packages is as simple as this :

```
# slackpkg install microlinux-server
```

Now we’ve completed the initial installation, there’s still a few things left to do before we can start using our new system. The following explanations provide some quick help for the most crucial configuration steps. Again, you might want to check out the *Slackware Documentation Project* at http://docs.slackware.com for more in-depth information.

Set locales

You’ll probably have to adjust your environment variables in `/etc/profile.d/lang.sh`. Default variables are set to `fr_FR.UTF8`, since MLES’ main use is in France :

```
# /etc/profile.d/lang.sh
...
export LANG=fr_FR.utf8
export LC_COLLATE=fr_FR.utf8
```

English-speaking Slackware users will use something like this :

```
# /etc/profile.d/lang.sh
...
export LANG=en_US.utf8
export LC_COLLATE=en_US.utf8
```

Switch to the GENERIC kernel

The `HUGE` kernel installed by the Slackware installer is essentially a kernel which has every hardware driver built in which we might have needed for a successful installation of our computer. It is recommended that we switch to the `GENERIC` kernel. This kernel has virtually no drivers built in and requires an initial ramdisk.

The first step will be to identify which kernel modules are needed at boot time. Our Slackware system is shipping a nifty little script to do just that :

```
# /usr/share/mkinitrd/mkinitrd_command_generator.sh
```

The result will vary depending on your hardware.

```
mkinitrd -c -k 3.10.17-smp -f ext4 -r /dev/sda3 -m mbcache:jbd2:ext4
  -u -o /boot/initrd.gz
```

The highlighted interesting bit here is the `-m mbcache:jbd:ext4` option. It means that my `GENERIC` kernel will require the `mbcache`, `jbd2` and `ext4` modules at boot time.

We can now provide this information to create and edit `/etc/mkinitrd.conf` :

```
# cd /etc
# cp mkinitrd.conf.sample mkinitrd.conf
```

Uncomment all options in the file and edit it according to your configuration. Here’s an example, with the relevant bits highlighted :

```
# /etc/mkinitrd.conf
SOURCE_TREE="/boot/initrd-tree"
CLEAR_TREE="1"
OUTPUT_IMAGE="/boot/initrd.gz"
KERNEL_VERSION="$(uname -r)"
KEYMAP="fr-latin1"
```

```
MODULE_LIST="mbcache:jbd2:ext4"
ROOTDEV="/dev/sda3"
ROOTFS="ext4"
RESUMEDEV="/dev/sda2"
RAID="0"
LVM="0"
UDEV="1"
MODCONF="0"
WAIT="1"
```

The relevant options in a nutshell :

- `CLEAR_TREE` defaults to `0` and goes to `1` ;
- `KEYMAP` defines your console keyboard layout ;
- `MODULE_LIST` contains the list of all kernel modules inserted into the initrd, separated by `:`
- `ROOTDEV` is the root partition ;
- `ROOTFS` is the file system used on the root partition ;
- `RESUMEDEV` is the swap partition ;
- You can safely removed all the `LUKS*` lines if you don't encrypt your partitions.

Now all our options are defined, we can build the initrd with this single command :

```
# mkinitrd -F
```

The last step will be to create a new stanza in our bootloader's configuration, pointing to the `GENERIC` kernel and the initrd. Here's an example :

```
# /etc/lilo.conf
...
image = /boot/vmlinuz-generic-smp-3.10.17-smp
  initrd = /boot/initrd.gz
  root = /dev/sda3
  label = Generic
  read-only
image = /boot/vmlinuz
  root = /dev/sda3
  label = Huge
  read-only
```

On a 64-bit system (or on a non-SMP system) the stanza would look like this :

```
image = /boot/vmlinuz-generic-3.10.17
  initrd = /boot/initrd.gz
  root = /dev/sda3
  label = Generic
  read-only
```

Don't forget to run LILO to install the new bootloader image :

```
# lilo
```

Cross your fingers and reboot. Once you're running the GENERIC kernel, you may eventually want to get rid of the HUGE stanza in /etc/lilo.conf.

Basic maintenance

Here's a few tips & tricks to keep your MLES installation up-to-date.

Check out the various ChangeLog.txt files at the root of each package repository for new additions and/or updates.

If you want to know the state of your system, what's installed, what's not installed or what updates are available, you can do all this in a few commands :

```
# slackpkg update
# slackpkg search microlinux-server
```

Enjoy your shiny new Microlinux Enterprise Server !

– *Niki Kovacs <info@microlinux.fr>*