Alpine setup scripts

Feature descriptions for available Alpine Linux setup scripts (`/sbin/setup-*`).

These scripts can be installed by using `apk` to install the `alpine-conf` package.

If you don't have an Alpine Linux install, you can find and examine the scripts in their git repository (https://github.com/alpinelinux/alpine-conf).

setup-alpine

This is the main Alpine configuration and installation script.

The script interactively walks the user through executing several auxiliary `setup-*` scripts, in the order shown below.

The bracketed options represent example configuration choices, formatted as they may be supplied when manually calling the auxiliary setup scripts, or using a `setup-alpine` "answerfile" (see below).

1. `setup-keymap` [us us]
2. setup-hostname [-n alpine-test]
3. setup-interfaces [-i < interfaces-file]
4. `rc-service networking --quiet start &`
5. if none of the networking interfaces were configured using dhcp, then: `setup-dns` [-d example.com -n "192.168.0.1 [...]"]
6. set the root password
7. if not in quick mode, then: `setup-timezone` [-z UTC | -z America/New_York | -p EST+5]
8. enable the new hostname (`rc-service hostname --quiet restart`)
9. add `networking` and `seedrng` (also referred to as `urandom` in versions prior to OpenRC 0.45) to the **boot** rc level, and `acpid` and `crond` to the **default** rc level, and start the **boot** and **default** rc services
10. extract the fully-qualified domain name and hostname from `/etc/resolv.conf` and `hostname`, and update `/etc/hosts`
11. `setup-proxy` [-q "http://webproxy:8080"], and activate proxy if it was configured
12. `setup-apkrepos` [-r (to select a mirror randomly)]
13. setup-user
14. if not in quick mode, then: `setup-sshd` [-c openssh | dropbear | none]
15. if not in quick mode, then: `setup-ntp` [-c chrony | openntpd | busybox | none]
16. if not in quick mode, then: `DEFAULT_DISK=none` `setup-disk -q` [-m data /dev/sda]

(see Installation#Installation_Overview about the disk modes)

17. if installation mode selected during setup-disk was "data" instead of "sys", then: `setup-lbu` [`/media/sdb1`]
18. if installation mode selected during setup-disk was "data" instead of "sys", then: `setup-apkcache` [`/media/sdb1/cache` | none]

`setup-alpine` itself accepts the following command-line switches:

-h

Shows the up-to-date usage help message.

-a

Create an overlay file: this creates a temporary directory and saves its location in ROOT; however, the script doesn't export this variable so I think this feature isn't currently functional.

-c *answerfile*

Create a new answerfile with default choices. You can edit the file and then invoke `setup-alpine -f` *`answerfile`*.

-f *answerfile*

Use an existing answerfile, which may override some or all of the interactive prompts. You can also specify a HTTP(S) or FTP URL for `setup-alpine` to download (https://gitlab.alpinelinux.org/alpine/alpine-conf/-/merge_requests/22) an answerfile from. Doing so will spin up a temporary networking config if one is not already active.

-q

Run in "quick mode".

setup-hostname

`setup-hostname` [-h] [-n hostname]

Options:

-h *Show help*

-n *Specify hostname*

This script allows quick and easy setup of the system hostname by writing it to `/etc/hostname`. The script prevents you from writing an invalid hostname (such as one that used invalid characters or starts with a '-' or is too long). The script can be invoked manually or is called as part of the `setup-alpine` script.

setup-interfaces

```
setup-interfaces [-i < interfaces-file]
```

Note that the contents of *interfaces-file* has to be supplied as stdin, rather than naming the file as an additional argument. The contents should have the format of `/etc/network/interfaces`, such as:

Contents of `/etc/network/interfaces`

```
auto lo
iface lo inet loopback

auto eth0
iface eth0 inet dhcp
    hostname alpine-test
```

setup-dns

```
setup-dns [-h] [-d domain name] [-n name server]
```

Options:

-h *Show help*

-d *specify search domain name*

-n *name server IP*

The setup-dns script is stored in `/sbin/setup-dns` and allows quick and simple setup of DNS servers (and a DNS search domain if required). Simply running `setup-dns` will allow interactive use of the script, or the options can be specified.

The information fed to this script is written to `/etc/resolv.conf`

Example usage (with 192.168.0.1 being the local router/dns-forwarder):

```
setup-dns -d example.org -n 192.168.0.1
```

Example `/etc/resolv.conf`:

Contents of `/etc/resolv.conf`

```
search example.org
nameserver 192.168.0.1
```

It can be run manually but is also invoked in the `setup-alpine` script unless interfaces are configured for DHCP.

setup-timezone

```
setup-timezone [-z UTC | -z America/New_York | -p EST+5]
```

Can pre-select the timezone using either of these switches:

-z *subfolder of* `/usr/share/zoneinfo`

-p *POSIX TZ format*

To manually change the time-zone, see Setting the timezone.

setup-proxy

```
setup-proxy [-hq] [PROXYURL]
```

Options:

-h *Show help*

-q *Quiet mode* prevents changes from taking effect until after reboot

This script requests the system proxy to use in the form `http://<proxyurl>:<port>` for example: `http://10.0.0.1:8080`

To set no system proxy use `none`. This script exports the following environmental variables:

```
http_proxy=$proxyurl

https_proxy=$proxyurl

ftp_proxy=$proxyurl
```

where `$proxyurl` is the value input. If `none` was chosen then the value it is set to a blank value (and so no proxy is used).

setup-apkrepos

```
setup-apkrepos [-fhr] [REPO...]
```

Setup `apk` repositories.

options:

-c *Enable the community repo*

-f *Detect and add fastest mirror*

-h *Show help*

-r *Add a random mirror and do not prompt*

-1 *Add first mirror on the list (normally a CDN)*

This is run as part of the `setup-alpine` script.

setup-user

> `setup-user` [-h] [-a] [-u] [-f FULLNAME] [-g GROUPS] [-k SSHKEY] [USERNAME]

Options:

-h *Show help*

-a *Create admin user. Add to wheel group and setup doas.*

-u *Unlock the user automatically (eg. create the user non-interactively with an ssh key for login)*

-f *Set the full name for the user*

-g *Comma or space separated list of groups to add user to*

-k *ssh key or URL to ssh key (eg. https://gitlab.alpinelinux.org/user.keys) or* `none` *for no key*

If USERNAME is not specified user will be prompted.

The setup-user script is stored in `/sbin/setup-user` and allows quick and simple setup of user accounts. It can be run manually but is also invoked in the `setup-alpine` script.

setup-sshd

> `setup-sshd` [-h] [-c choice of SSH daemon]

Options:

-h *Show help*

-c *SSH daemon* where SSH daemon can be one of the following:

`openssh` install the `openSSH (https://pkgs.alpinelinux.org/packages?name=openSSH&branch=edge&repo=&arch=x86_64&maintainer=)` daemon

`dropbear` install the `dropbear (https://pkgs.alpinelinux.org/packages?name=dropbear&branch=edge&repo=&arch=x86_64&maintainer=)` daemon

`none` Do not install an SSH daemon

Example usage:

```
setup-sshd -c dropbear
```

The setup-sshd script is stored in `/sbin/setup-sshd` and allows quick and simple setup of either the OpenSSH or Dropbear SSH daemon & client. It can be run manually but is also invoked in

the `setup-alpine` script.

setup-ntp

From Wikipedia (https://en.wikipedia.org/wiki/Network_Time_Protocol):

The **Network Time Protocol (NTP)** is a networking protocol for clock synchronization between computer systems over packet-switched, variable-latency data networks.

```
usage: setup-ntp [-h] [busybox|openntpd|chrony|none]

Setup NTP time synchronization

options:
 -h  Show this help

 User is prompted if no NTP daemon is specified
```

`setup-ntp` script is stored in `/sbin/setup-ntp` and allows quick and simple setup of the NTP client, It can be run manually but is also invoked in the `setup-alpine` script.

setup-disk

> `DEFAULT_DISK=none setup-disk -q` [-m data | sys] [*mountpoint directory* | `/dev/sda` ...]

In "sys" mode, it's an installer, it permanently installs Alpine on the disk, in "data" mode, it provides a larger and persistent `/var` volume.

This script accepts the following command-line switches:

-k *kernel flavor*
-o *apkovl file*
: Restore system from *apkovl file*

-m data | sys
: Don't prompt for installation mode. With **-m data**, the supplied devices are formatted to use as a `/var` volume.

-r
: Use RAID1 with a single disk (degraded mode)

-L
: Create and use volumes in a LVM group

-s *swap size in MB*
: Use 0 to disable swap

-q
: Exit quietly if no disks are found

-v
: Verbose mode

The script also honors the following environment variables:

`BOOT_SIZE`

: Size of the boot partition in MB; defaults to 100. Only used if **-m sys** is specified or interactively selected.

`SWAP_SIZE`

: Size of the swap volume in MB; set to 0 to disable swap. If not specified, will default to twice RAM, up to 4096, but won't be more than 1/3 the size of the smallest disk, and if less than 64 will just be 0. Only used if **-m sys** is specified or interactively selected.

`ROOTFS`

: Filesystem to use for the `/` volume; defaults to ext4. Only used if **-m sys** is specified or interactively selected. Supported filesystems are: ext2 ext3 ext4 btrfs xfs.

`BOOTFS`

: Filesystem to use for the `/boot` volume; defaults to ext4. Only used if **-m sys** is specified or interactively selected. Supported filesystems are: ext2 ext3 ext4 btrfs xfs.

`VARFS`

: Filesystem to use for the `/var` volume; defaults to ext4. Only used if **-m data** is specified or interactively selected. Supported filesystems are: ext2 ext3 ext4 btrfs xfs.

`SYSROOT`

: Mountpoint to use when creating volumes and doing traditional disk install (**-m sys**). Defaults to `/mnt`.

`MBR`

: Path of MBR binary code, defaults to `/usr/share/syslinux/mbr.bin`.

`BOOTLOADER`

: Bootloader to use, defaults to syslinux. Supported bootloaders are: grub syslinux zipl.

`DISKLABEL`

: Disklabel to use, defaults to dos. Supported disklabels are: dos gpt eckd.

Partitioning

If you have complex partitioning needs, that go beyond above alpine-disk options, you can partition, format, and mount your volumes manually, and then just supply the root mountpoint to `setup-disk`. Doing so implicitly behaves as though **-m sys** had also been specified.

See Setting up disks manually for more information.

RAID

`setup-disk` will automatically build a RAID array if you supply the **-r** switch, or if you specify more than one device. The array will always be RAID1 (https://en.m.wikipedia.org/wiki/Standard_RAID_levels#RAID_1) (and --metadata=0.90 (https://raid.wiki.kernel.org/index.php/RAID_superblock_formats#The_version-0.90_Superblock_Format)) for the `/boot` volumes, but will be RAID5 (https://en.m.wikipedia.org/wiki/Standard_RAID_levels#RAID_5) (and --metadata=1.2 (https://raid.wiki.kernel.org/index.php/RAID_superblock_formats#The_version-1_Superblock_Format) for non-boot volumes when 3 or more devices are supplied.

If you instead want to build your RAID array manually, see Setting up a software RAID array. Then format and mount the disks, and supply the root mountpoint to `setup-disk`.

LVM

`setup-disk` will automatically build and use volumes in a LVM group if you supply the **-L** switch. The group and volumes created by the script will have the following names:

- volume group: **vg0**
- swap volume: **lv_swap** (only created when swap size > 0)
- root volume: **lv_root** (only created when **-m sys** is specified or interactively selected)
- var volume: **lv_var** (only created when **-m data** is specified or interactively selected)

The **lv_var** or **lv_root** volumes are created to occupy all remaining space in the volume group.

If you need to change any of these settings, you can use `vgrename`, `lvrename`, `lvreduce` or `lvresize`.

If you instead want to build your LVM system manually, see Setting up Logical Volumes with LVM. Then format and mount the disks, and supply the root mountpoint to `setup-disk`.

setup-lbu

This script will only be invoked for by `setup-alpine` when installing `data` installation types (ramdisk)

It configures where `lbu commit` will store the .apkovl backup. See Alpine local backup for more information.

When started, `setup-lbu` will prompt where to store your data. The options it will prompt for will be taken from the directories found in `/media` (except for `cdrom`). [not sure how these are mounted: are they automatically mounted by setup-lbu? Does the user have to manually mount using another tty?]

setup-apkcache

This script will only be invoked for by `setup-alpine` when installing `data` installation types (ramdisk)

It configures where to save the apk package files. The apkcache is where apk stores downloaded packages, such that the system does not need to download them again on each reboot, and doesn't have to depend on the network. See Local APK cache for a detailed explanation.

You should be able to use a partition that you set up in the previous steps.

setup-bootable

This is a standalone script; it's not invoked by `setup-alpine` but must be run manually.

It allows to create boot media that boots the system running from RAM memory (diskless) like the installation images, but using a writable (i.e. not iso9660) filesystem. So that it can also serve to store local customizations (e.g. apkovl files and cached packages).

First, the script copies files from an ISO image (as file on a CD/DVD/USB etc.) onto a USB-Stick/CompactFlash/SDCard etc., or harddisk partition. And then, it installs the syslinux bootloader to make the device bootable.

However, its current syslinux installation seems to fail on non-FAT32 partitions. So in these cases, you may start over with a FAT32 filesystem, or rather with the desired filesystem and using `setup-bootable` only with the `-u` option, to skip the syslinux install, and then refer to the manual method to fix the problem, or use one of the other bootloader options, instead.

Tip: The Bootloaders page shows different ways to setup booting, and multi-boot menus!

The setup-bootable script accepts the following arguments and command-line switches (you can run `setup-bootable -h` to see a usage message).

```
setup-bootable source [dest]
```

The argument *source* can be a directory or an ISO (will be mounted to `MNT` or `/mnt`) or a URL (will be downloaded with `WGET` or `wget`). The argument *dest* can be a directory mountpoint, or will default to `/media/usb` if not supplied.

-k

Keep alpine_dev in `syslinux.cfg`; otherwise, replace with UUID.

-u
Upgrade mode: keep existing `syslinux.cfg` and don't run `syslinux`

-f
Overwrite `syslinux.cfg` even if **-u** was specified.

-s
Force the running of `syslinux` even if **-u** was specified.

-v
Verbose mode

The script will ensure that *source* and *dest* are available; will copy the contents of *source* to *dest*, ensuring first that there's enough space; and unless **-u** was specified, will make *dest* bootable.

Suppose the target device is `/dev/sdXY`, then this partition can be prepared for booting with

```
# setup-bootable -v /media/<installation-media-device> /dev/sdXY
```

For the manual way to set up boot media see Manually_copying_Alpine_files.

setup-xorg-base

This is a standalone script; it's not invoked by `setup-alpine` but must be run manually.

It configures a graphical environment, installing basic Xorg packages and eudev (replacing mdev), and is also required for Wayland sessions.

The script installs, among other packages, e.g.: `xorg-server xf86-input-libinput xinit eudev`.

Additional packages to install may be supplied as arguments.

```
setup-xorg-base [additional package(s) to install]
```

Video packages (optional)

You may install specific xf86 xorg driver packages for your video card's chipset, as they may support specific features, effects and acceleration modes, and avoid error messages during X initialization.

However, the most basic X features should work fine with just using the default kernel video-modesetting drivers.

Info about the particular video cards that are installed in the computer may be found in the list of PCI devices:

```
# apk add pciutils
$ lspci
```

To see available video driver packages run:

```
$ apk search xf86-video
```

For example,

- For an Sis video chipset install 'xf86-video-sis'.

```
# apk add xf86-video-sis
```

Others:

- For Intel video chipsets see Intel Video.

Tip: In some cases, freezes on suspend/resume stop happening when changing the video port the monitor is connected to.

- For AMD Radeon Video see Radeon Video
- For Alix1D use xf86-video-geode.
- In KVM/QEMU guests see Xorg within KVM/QEMU
- In VirtualBox guests use xf86-video-vboxvideo, and install the VirtualBox guest additions as well. They contain important parts for the driver.
- In VMware guests use xf86-video-vmware
- In Hyper-V guests use xf86-video-fbdev and install the Hyper-V guest services as well.

Input packages

If the **Numlock** settings are not working, or getting **'setleds not found'** errors:

```
# apk add kbd
```

If some input device is not working at all, the available xf86-input drivers can be listed with:

```
$ apk search xf86-input
```

You probably at least want

```
```

```
xf86-input-libinput
```

or

```
xf86-input-evdev
```

libinput is for Wayland with wrapper for Xorg. evdev is Xorg only.

Typical legacy drivers (not packaged. at least as of 2/2022):

```
# apk add xf86-input-mouse xf86-input-keyboard
```

And for touchpad tapping support on many laptops, also:

```
# apk add xf86-input-synaptics
```

Configure xorg-server (optional)

On most systems, xorg should be able to autodetect all devices. However you can still configure xorg-server by hand by launching:

```
# Xorg -configure
```

This will create a `/root/xorg.conf.new` file. You can modify this file to fit your needs.
(When finished modifying and testing the above configuration file, move it to `/etc/X11/xorg.conf` for normal usage.)

Keyboard Layout (optional)

If you use a keyboard layout different than "us", and you are using a window manager or desktop environment that does not support to configure the keyboard layout itself, then you need to

- Enable the "community" repository

and install setxkbmap:

```
# apk add setxkbmap
```

Then try

```
# setxkbmap <%a language layout from /usr/share/X11/xkb/rules/xorg.lst%>
```

In order to make it persistent add this section to `/etc/X11/xorg.conf`:

```
Section "InputClass"
        Identifier      "Keyboard Default"
        MatchIsKeyboard "yes"
        Option          "XkbLayout" "<%a language layout from /usr/share/X11/xkb/ru
EndSection
```

Another way to change the keymap when logging into X is to use `~/.xinitrc`. The following example loads a British keymap, simply add this line to the beginning of the file: `setxkbmap gb &`

If you need to create the `~/.xinitrc` file, you may also want to add a second line like `exec openbox-session` to still start the window manager with `startx` or `xinit`.

setup-desktop

This script is used to set up a desktop environment. This is a replacement for the now-removed `alpine-desktop` package.

Note: Before installing any desktop,

- create a non-privileged user account for security reasons.
- enable the `community` repository. Setup-desktop script automatically does this for you.

Installation using setup-desktop

The Alpine Linux script for setting up a desktop quickly is setup-desktop.

```
# setup-desktop
```

On running the above command, you will be prompted to select a desktop environment.

```
Which desktop environment? ('gnome', 'plasma', 'xfce', 'mate', 'sway' or 'none') [n
```

Once you have chosen a desktop environment, this script installs the chosen desktop along with all the necessary packages, `firefox (https://pkgs.alpinelinux.org/packages?name=firefox&branch=edge&repo=community&arch=x86_64&maintainer=)` browser and adds the necessary services to run on startup. You can reboot when complete and the system will boot into a graphical login screen with the desktop environment. Depending on the desktop chosen, the script also activates the necessary services like dbus, elogind, login manager etc..

To view all the packages that are installed by the script for the chosen desktop you can issue the below command:

```
# cat /sbin/setup-desktop
```

Documentation needed

setup-xen-dom0

setup-mta

Uses ssmtp.

This is a standalone script; it's not invoked by `setup-alpine` but must be run manually.

setup-acf

> `setup-acf` [-ahn] [-e email] [-l address] [PACKAGE...]

Options:

-a *Which acf-* packages to install*

-h *Show help*

-n *Don't create* `/etc/acf/passwd`

-e *email address (for TLS certificate)*

-l *hostname for mini_httpd*

This is a standalone script; it's not invoked by `setup-alpine` but must be run manually.

This script was named `setup-webconf` before Alpine 1.9 beta 4.

See ACF pages for more information.

See also

- Installation
- Post Install