**Install from existing Linux**

**Related articles**

* [**Install from SSH**](https://wiki.archlinux.org/index.php/Install_from_SSH)

This document describes the bootstrapping process required to install Arch Linux from a running Linux host system. After bootstrapping, the installation proceeds as described in the [**Installation guide**](https://wiki.archlinux.org/index.php/Installation_guide).

Installing Arch Linux from a running Linux is useful for:

* remotely installing Arch Linux, e.g. a (virtual) root server
* replacing an existing Linux without a LiveCD (see [**#Replacing the existing system without a LiveCD**](https://wiki.archlinux.org/index.php/Install_from_existing_Linux#Replacing_the_existing_system_without_a_LiveCD))
* creating a new Linux distribution or LiveCD based on Arch Linux
* creating an Arch Linux chroot environment, e.g. for a Docker base container
* [**rootfs-over-NFS for diskless machines**](https://wiki.archlinux.org/index.php/Diskless_network_boot_NFS_root)

The goal of the bootstrapping procedure is to setup an environment from which the scripts from [**arch-install-scripts**](https://www.archlinux.org/packages/?name=arch-install-scripts) (such as pacstrap and arch-chroot) can be run.

If the host system runs Arch Linux, this can be achieved by simply installing [**arch-install-scripts**](https://www.archlinux.org/packages/?name=arch-install-scripts). If the host system runs another Linux distribution, you will first need to set up an Arch Linux-based chroot.

**Note:**This guide requires that the existing host system be able to execute the new target Arch Linux architecture programs. In the case of an x86\_64 host, it is possible to use i686-pacman to build a 32-bit chroot environment. See [**Arch64 Install bundled 32bit system**](https://wiki.archlinux.org/index.php/Arch64_Install_bundled_32bit_system). However it is not so easy to build a 64-bit environment when the host only supports running 32-bit programs.

**Warning:**Please make sure you understand each step before proceeding. It is easy to destroy your system or to lose critical data, and your service provider will likely charge a lot to help you recover.

**Contents**

 [[hide](https://wiki.archlinux.org/index.php/Install_from_existing_Linux)]

* [1 Backup and Preparation](https://wiki.archlinux.org/index.php/Install_from_existing_Linux#Backup_and_Preparation)
* [2 From a host running Arch Linux](https://wiki.archlinux.org/index.php/Install_from_existing_Linux#From_a_host_running_Arch_Linux)
* [3 From a host running another Linux distribution](https://wiki.archlinux.org/index.php/Install_from_existing_Linux#From_a_host_running_another_Linux_distribution)
  + [3.1 Creating the chroot](https://wiki.archlinux.org/index.php/Install_from_existing_Linux#Creating_the_chroot)
    - [3.1.1 Method A: Using the bootstrap image (recommended)](https://wiki.archlinux.org/index.php/Install_from_existing_Linux#Method_A:_Using_the_bootstrap_image_.28recommended.29)
    - [3.1.2 Method B: Using the LiveCD image](https://wiki.archlinux.org/index.php/Install_from_existing_Linux#Method_B:_Using_the_LiveCD_image)
  + [3.2 Using the chroot environment](https://wiki.archlinux.org/index.php/Install_from_existing_Linux#Using_the_chroot_environment)
    - [3.2.1 Initializing pacman keyring](https://wiki.archlinux.org/index.php/Install_from_existing_Linux#Initializing_pacman_keyring)
    - [3.2.2 Selecting a mirror and downloading basic tools](https://wiki.archlinux.org/index.php/Install_from_existing_Linux#Selecting_a_mirror_and_downloading_basic_tools)
  + [3.3 Installation tips](https://wiki.archlinux.org/index.php/Install_from_existing_Linux#Installation_tips)
    - [3.3.1 Debian-based host](https://wiki.archlinux.org/index.php/Install_from_existing_Linux#Debian-based_host)
      * [3.3.1.1 /dev/shm](https://wiki.archlinux.org/index.php/Install_from_existing_Linux#.2Fdev.2Fshm)
      * [3.3.1.2 /dev/pts](https://wiki.archlinux.org/index.php/Install_from_existing_Linux#.2Fdev.2Fpts)
      * [3.3.1.3 lvmetad](https://wiki.archlinux.org/index.php/Install_from_existing_Linux#lvmetad)
    - [3.3.2 Fedora-based host](https://wiki.archlinux.org/index.php/Install_from_existing_Linux#Fedora-based_host)
* [4 Things to check before you reboot](https://wiki.archlinux.org/index.php/Install_from_existing_Linux#Things_to_check_before_you_reboot)
* [5 Replacing the existing system without a LiveCD](https://wiki.archlinux.org/index.php/Install_from_existing_Linux#Replacing_the_existing_system_without_a_LiveCD)
  + [5.1 Set old swap partition as new root partition](https://wiki.archlinux.org/index.php/Install_from_existing_Linux#Set_old_swap_partition_as_new_root_partition)
  + [5.2 Installation](https://wiki.archlinux.org/index.php/Install_from_existing_Linux#Installation)

Backup and Preparation

Backup all your data including mails, webservers, etc. Have all information at your fingertips. Preserve all your server configurations, hostnames, etc.

Here is a list of data you will likely need:

* IP address
* hostname(s), (note: rootserver are mostly also part of the providers domain, check or save your /etc/hosts before you delete)
* DNS server (check /etc/resolv.conf)
* SSH keys (if other people work on your server, they will have to accept new keys otherwise. This includes keys from your Apache, your mail servers, your SSH server and others.)
* Hardware info (network card, etc. Refer to your pre-installed /etc/modules.conf )
* Grub configuration files.

In general, it is a good idea to have a local copy of your original /etc directory on your local hard drive.

From a host running Arch Linux

Install the [**arch-install-scripts**](https://www.archlinux.org/packages/?name=arch-install-scripts) package.

Follow [**Installation guide#Mount the partitions**](https://wiki.archlinux.org/index.php/Installation_guide#Mount_the_partitions). If you already use the /mnt directory for something else, just create another directory such as /mnt/install, and use that instead.

Then follow [**Installation guide#Installation**](https://wiki.archlinux.org/index.php/Installation_guide#Installation). You can skip [**Installation guide#Select the mirrors**](https://wiki.archlinux.org/index.php/Installation_guide#Select_the_mirrors), since the host should already have a correct mirrorlist.

**Tip:**In order to avoid redownloading all the packages, consider following [**Pacman/Tips and tricks#Network shared pacman cache**](https://wiki.archlinux.org/index.php/Pacman/Tips_and_tricks#Network_shared_pacman_cache) or using *pacstrap'*s -c option.

**[Merge-arrows-2.png](https://wiki.archlinux.org/index.php/File:Merge-arrows-2.png)This article or section is a candidate for merging with**[**Moving\_an\_existing\_install\_into\_(or\_out\_of)\_a\_virtual\_machine#Moving\_into\_a\_VM**](https://wiki.archlinux.org/index.php/Moving_an_existing_install_into_(or_out_of)_a_virtual_machine#Moving_into_a_VM)**.[Merge-arrows-2.png](https://wiki.archlinux.org/index.php/File:Merge-arrows-2.png)**

**Notes:** Same approach. (Discuss in [**Talk:Install from existing Linux#**](https://wiki.archlinux.org/index.php/Talk:Install_from_existing_Linux))

**Note:**If you only want to create an exact copy of an existing Arch installation, it is also possible to just copy the filesystem to the new partition. With this method, you will still need to

* Create [**/etc/fstab**](https://wiki.archlinux.org/index.php/Beginners%27_guide#Generate_an_fstab) and edit /etc/hostname
* Delete /etc/machine-id so that a new, unique, one will be regenerated on boot
* Make any other changes appropriate to the installation medium
* Install the bootloader

When copying the filesystem root, use something like cp -ax or rsync -axX. This avoids copying contents of mountpoints (-x), and preserves the [**capabilities**](https://wiki.archlinux.org/index.php/Capabilities) attributes of some system binaries (rsync -X).

From a host running another Linux distribution

There are multiple tools which automate a large part of the steps described in the following subsections. See their respective homepages for detailed instructions.

* [**arch-bootstrap**](https://github.com/tokland/arch-bootstrap) (Bash)
* [**image-bootstrap**](https://github.com/hartwork/image-bootstrap) (Python)
* [**vps2arch**](https://github.com/drizzt/vps2arch) (Bash)
* [**archcx**](https://github.com/m4rienf/ArchCX) (Bash, from Hetzner CX Rescue System)

The manual way is presented in the following subsections. The idea is to run an Arch system inside the host system, with the actual installation being executed from the Arch system. The nested system is contained inside a chroot.

**Creating the chroot**

Two methods to setup and enter the chroot are presented below, from the easiest to the most complicated. Select only one of the two methods. Then, continue at [**#Using the chroot environment**](https://wiki.archlinux.org/index.php/Install_from_existing_Linux#Using_the_chroot_environment).

**Method A: Using the bootstrap image (recommended)**

Download the bootstrap image from a [**mirror**](https://www.archlinux.org/download):

$ curl -O [**https://mirrors.kernel.org/archlinux/iso/2016.04.01/archlinux-bootstrap-2016.04.01-x86\_64.tar.gz**](https://mirrors.kernel.org/archlinux/iso/2016.04.01/archlinux-bootstrap-2016.04.01-x86_64.tar.gz)

Extract the tarball:

# cd /tmp

# tar xzf <path-to-bootstrap-image>/archlinux-bootstrap-2016.04.01-x86\_64.tar.gz

Select a repository server by editing /tmp/root.x86\_64/etc/pacman.d/mirrorlist.

**Note:**If bootstrapping an i686 image from an x86\_64 host system, also edit /tmp/root.i686/etc/pacman.conf and explicitly define Architecture = i686 in order for pacman to pull the proper i686 packages.

Enter the chroot

* If bash 4 or later is installed, and unshare supports the --fork and --pid options:

# /tmp/root.x86\_64/bin/arch-chroot /tmp/root.x86\_64/

* Otherwise, run the following commands:

# mount --bind /tmp/root.x86\_64 /tmp/root.x86\_64

# cd /tmp/root.x86\_64

# cp /etc/resolv.conf etc

# mount -t proc /proc proc

# mount --rbind /sys sys

# mount --rbind /dev dev

# mount --rbind /run run # (assuming /run exists on the system)

# chroot /tmp/root.x86\_64 /bin/bash

**Method B: Using the LiveCD image**

It is possible to mount the root image of the latest Arch Linux installation media and then chroot into it. This method has the advantage of providing a working Arch Linux installation right within the host system without the need to prepare it by installing specific packages.

**Note:**Before proceeding, make sure the latest version of [**squashfs**](http://squashfs.sourceforge.net/) is installed on the host system. Otherwise, errors like the following are to be expected:FATAL ERROR aborting: uncompress\_inode\_table: failed to read block.

* The root image can be found on one of the [**mirrors**](https://www.archlinux.org/download) under either arch/x86\_64/ or arch/i686/, depending on the desired architecture. The squashfs format is not editable, so we unsquash the root image and mount it.
* To unsquash the root image, run

# unsquashfs airootfs.sfs

* Before [**chrooting**](https://wiki.archlinux.org/index.php/Change_root) to it, we need to set up some mount points and copy the resolv.conf for networking.

# mount --bind squashfs-root squashfs-root

# mount -t proc none squashfs-root/proc

# mount -t sysfs none squashfs-root/sys

# mount -o bind /dev squashfs-root/dev

# mount -o bind /dev/pts squashfs-root/dev/pts ## important for pacman (for signature check)

# cp -L /etc/resolv.conf squashfs-root/etc ## this is needed to use networking within the chroot

* Now, everything is prepared to chroot into the newly installed Arch environment

# chroot squashfs-root bash

**Using the chroot environment**

The bootstrap environment is really barebones (no nano, no ping, no cryptsetup, no lvm). Therefore, we need to set up [**pacman**](https://wiki.archlinux.org/index.php/Pacman) in order to download the rest of the base and, if needed, base-devel.

**Initializing pacman keyring**

Before starting the installation, pacman keys need to be setup. Before running the following two commands, read [**pacman-key#Initializing the keyring**](https://wiki.archlinux.org/index.php/Pacman-key#Initializing_the_keyring) to understand the entropy requirements:

# pacman-key --init

# pacman-key --populate archlinux

**Tip:**Installing and running [**haveged**](https://www.archlinux.org/packages/?name=haveged) must be done on the host system, since it is not possible to install packages before initializing pacman keyring and because *systemd* will detect it is running in a chroot and [**ignore activation request**](https://superuser.com/questions/688733/start-a-systemd-service-inside-chroot). If you go with doing ls -Ra / in another console (TTY, terminal, SSH session...), do not be afraid of running it in a loop a few times: five or six runs from the host proved sufficient to generate enough entropy on a remote headless server.

**Selecting a mirror and downloading basic tools**

After [**selecting a mirror**](https://wiki.archlinux.org/index.php/Mirrors#Enabling_a_specific_mirror), [**refresh the package lists**](https://wiki.archlinux.org/index.php/Mirrors#Force_pacman_to_refresh_the_package_lists) and [**install**](https://wiki.archlinux.org/index.php/Install) what you need: [**base**](https://www.archlinux.org/groups/x86_64/base/), [**base-devel**](https://www.archlinux.org/groups/x86_64/base-devel/), [**parted**](https://www.archlinux.org/packages/?name=parted) etc.

**Installation tips**

You can now proceed to [**preparing the storage devices**](https://wiki.archlinux.org/index.php/Beginners%27_guide#Prepare_the_storage_devices) and follow the rest of the [**installation procedure**](https://wiki.archlinux.org/index.php/Installation_guide#Installation).

Some host systems or configurations may require certain extra steps. See the sections below for tips.

**Debian-based host**

**/dev/shm**

On some Debian-based host systems, pacstrap may produce the following error:

# pacstrap /mnt base

==> Creating install root at /mnt

mount: mount point /mnt/dev/shm is a symbolic link to nowhere

==> ERROR: failed to setup API filesystems in new root

This is because in some versions of Debian, /dev/shm points to /run/shm while in the Arch-based chroot, /run/shm does not exist and the link is broken. To correct this error, create a directory /run/shm:

# mkdir /run/shm

**/dev/pts**

While installing archlinux-2015.07.01-x86\_64 from a Debian 7 host, the following error prevented both [**pacstrap**](https://projects.archlinux.org/arch-install-scripts.git/tree/pacstrap.in) and [**arch-chroot**](https://wiki.archlinux.org/index.php/Change_root#Using_arch-chroot) from working:

# pacstrap -i /mnt

mount: mount point /mnt/dev/pts does not exist

==> ERROR: failed to setup chroot /mnt

Apparently, this is because these two scripts use a common function. chroot\_setup()[**[1]**](https://projects.archlinux.org/arch-install-scripts.git/tree/common#n76) relies on newer features of [**util-linux**](https://www.archlinux.org/packages/?name=util-linux), which are incompatible with Debian 7 userland (see [**FS#45737**](https://bugs.archlinux.org/task/45737)).

The solution for *pacstrap* is to manually execute its [**various tasks**](https://projects.archlinux.org/arch-install-scripts.git/tree/pacstrap.in#n77), but use the [**regular procedure**](https://wiki.archlinux.org/index.php/Change_root#Using_chroot) to mount the kernel filesystems on the target directory ("$newroot"):

# newroot=/mnt

# mkdir -m 0755 -p "$newroot"/var/{cache/pacman/pkg,lib/pacman,log} "$newroot"/{dev,run,etc}

# mkdir -m 1777 -p "$newroot"/tmp

# mkdir -m 0555 -p "$newroot"/{sys,proc}

# mount --bind "$newroot" "$newroot"

# mount -t proc /proc "$newroot/proc"

# mount --rbind /sys "$newroot/sys"

# mount --rbind /run "$newroot/run"

# mount --rbind /dev "$newroot/dev"

# pacman -r "$newroot" --cachedir="$newroot/var/cache/pacman/pkg" -Sy base base-devel ... ## add the packages you want

# cp -a /etc/pacman.d/gnupg "$newroot/etc/pacman.d/" ## copy keyring

# cp -a /etc/pacman.d/mirrorlist "$newroot/etc/pacman.d/" ## copy mirrorlist

Instead of using arch-chroot for [**configuring the base system**](https://wiki.archlinux.org/index.php/Beginners%27_guide#Chroot_and_configure_the_base_system), simply use chroot "$newroot".

**lvmetad**

Trying to create [**LVM**](https://wiki.archlinux.org/index.php/LVM) [**logical volumes**](https://wiki.archlinux.org/index.php/LVM#Logical_volumes) from an archlinux-bootstrap-2015.07.01-x86\_64 environment on a Debian 7 host resulted in the following error:

# lvcreate -L 20G lvm -n root

/run/lvm/lvmetad.socket: connect failed: No such file or directory

WARNING: Failed to connect to lvmetad. Falling back to internal scanning.

/dev/lvm/root: not found: device not cleared

Aborting. Failed to wipe start of new LV.

(Physical volume and volume group creation worked despite /run/lvm/lvmetad.socket: connect failed: No such file or directory being displayed.)

This could be easily worked around by creating the logical volumes outside the chroot (from the Debian host). They are then available once chrooted again.

**[Tango-inaccurate.png](https://wiki.archlinux.org/index.php/File:Tango-inaccurate.png)The factual accuracy of this article or section is disputed.[Tango-inaccurate.png](https://wiki.archlinux.org/index.php/File:Tango-inaccurate.png)**

**Reason:** This problem did not arise when installing from a Debian 7 host without lvmetad enabled. The recommended messaround with /etc/lvm/lvm.conf looks rather error prone (2015-07-26). (Discuss in [**Talk:Install from existing Linux#**](https://wiki.archlinux.org/index.php/Talk:Install_from_existing_Linux))

**[Tango-edit-clear.png](https://wiki.archlinux.org/index.php/File:Tango-edit-clear.png)This article or section needs language, wiki syntax or style improvements.[Tango-edit-clear.png](https://wiki.archlinux.org/index.php/File:Tango-edit-clear.png)**

**Reason:** Language and formatting are lacking, links to relevant articles in the wiki as well. (Discuss in [**Talk:Install from existing Linux#**](https://wiki.archlinux.org/index.php/Talk:Install_from_existing_Linux))

Also, if the system you are using has lvm, you might have the following output:

# grub-install --target=i386-pc --recheck /dev/mapper/main-archroot

Installing for i386-pc platform.

/run/lvm/lvmetad.socket: connect failed: No such file or directory

WARNING: Failed to connect to lvmetad. Falling back to internal scanning.

/run/lvm/lvmetad.socket: connect failed: No such file or directory

WARNING: Failed to connect to lvmetad. Falling back to internal scanning.

/run/lvm/lvmetad.socket: connect failed: No such file or directory

WARNING: Failed to connect to lvmetad. Falling back to internal scanning.

/run/lvm/lvmetad.socket: connect failed: No such file or directory

WARNING: Failed to connect to lvmetad. Falling back to internal scanning.

/run/lvm/lvmetad.socket: connect failed: No such file or directory

WARNING: Failed to connect to lvmetad. Falling back to internal scanning.

This is because debian does not use lvmetad by default. You need to edit /etc/lvm/lvm.conf and set use\_lvmetad to 0:

use\_lvmetad = 0

This will trigger later an error on boot in the initrd stage. Therefore, you have to change it back after the grub generation. In a software RAID + LVM, steps would be the following:

* After installing all the system, when you have to do all the initramfs (mkinitcpio) and grub thing.
* Change /etc/mdadm.conf to reflect your RAID config (if any)
* Change HOOKS and MODULES according to lvm and raid requirements: MODULES="dm\_mod" HOOKS="base udev **mdadm\_udev** ... block **lvm2** filesystems ..."
* Generate initrd images with mkinitcpio
* Change /etc/lvm/lvm.conf to put use\_lvmetad = 0
* Generate grub config (grub-mkconfig)
* Change /etc/lvm/lvm.conf to put use\_lvmetad = 1

**Fedora-based host**

On Fedora based hosts and live USBs you may encounter problems when using genfstab to generate your [**fstab**](https://wiki.archlinux.org/index.php/Fstab). Remove duplicate entries and the "seclabel" option where it appears, as this is Fedora-specific and will keep your system from booting normally.

Things to check before you reboot

Before rebooting, chroot into the newly-installed system.

Set a root password so that you can log in with ssh later:

# passwd

Install [**ssh**](https://wiki.archlinux.org/index.php/Ssh) and [**enable**](https://wiki.archlinux.org/index.php/Enable) it to start automatically at boot.

Configure the [**network**](https://wiki.archlinux.org/index.php/Network) connection to start automatically at boot.

Set up a [**boot loader**](https://wiki.archlinux.org/index.php/Boot_loader) and configure it to use the swap partition you appropriated earlier as the root partition. You might want to configure your bootloader to be able to boot into your old system; it is helpful to re-use the server's existing /boot partition in the new system for this purpose.

Replacing the existing system without a LiveCD

Find ~700MB of free space somewhere on the disk, e.g. by partitioning a swap partition. You can disable the swap partition and set up your system there.

**Set old swap partition as new root partition**

Check cfdisk, /proc/swaps or /etc/fstab to find your swap partition. Assuming your hard drive is located on sdaX (X will be a number).

Do the following:

Disable the swap space:

# swapoff /dev/sdaX

Create a filesystem on it

# fdisk /dev/sda

(set /dev/sdaX ID field to "Linux" - Hex 83)

# mke2fs -j /dev/sdaX

Create a directory to mount it in

# mkdir /mnt/newsys

Finally, mount the new directory for installing the intermediate system.

# mount -t ext4 /dev/sdaX /mnt/newsys

**Installation**

If less than 700MB are available, examine the packages in the group base, and select only those required to get a system with internet connection up and running in the temporary partition. This will mean explicitly specifying individual packages to pacstrap, as well as passing it the -c option, to get packages downloaded to the host system to avoid filling up valuable space.

Once the new Arch Linux system is installed, reboot into the newly created system, and [**rsync the entire system**](https://wiki.archlinux.org/index.php/Full_system_backup_with_rsync#With_a_single_command) to the primary partition. Fix the bootloader configuration before rebooting.