

NAME

ezjail-admin — Administrate ezjail environment

SYNOPSIS

```
ezjail-admin install [-mMpPsS] [-h host] [-r release]  
ezjail-admin create [-bx] [-f flavour] [-r jailroot] [-a archive]  
    [-c jailtype -s imagesize [-C attachargs]] [-z parentzfs]  
    jailname ipaddress[,ipaddress2,...]  
ezjail-admin console [-f] [-e command] jailname  
ezjail-admin list  
ezjail-admin start | stop | restart | cryptostart jailname...  
ezjail-admin config [-r run | norun] [-n newname] [-i attach | detach | fsck]  
    [-z newdataset] [-c newcpuset] [-f newfib] jailname  
ezjail-admin delete [-wf] jailname  
ezjail-admin archive [-Af] [-a archive] [-d archivedir] jailname...  
ezjail-admin restore [-f] [-d archivedir] archive | jailname...  
ezjail-admin update [-s sourcetree | sourceosversion] [-p] -b | -i | -P | -u  
    | -U
```

DESCRIPTION

The **ezjail-admin** utility is used to manage the ezjail environment and all the jails inside the ezjail scope. This man page describes the invocation of **ezjail-admin**. Refer to ezjail(7) in order to get an introduction to the usage of ezjail, as well as usage examples.

The description of some options ends with ‘Variable: “\$ezjail_abcd”’. This means that the default value of the option may be overridden by setting this variable in ezjail.conf(5).

ezjail-admin install

This function sub-command is normally run once in the life of the ezjail environment. It allocates the directory structure used by ezjail and populates the base jail using the minimal distribution set from a FreeBSD FTP server.

The default location for ezjail’s basejail is in /usr/jails, so be sure you have enough space there (a FreeBSD base release without man pages, sources and ports is around 120MB). This location may be modified in ezjail.conf(5).

See also **ezjail-admin update** to install the base jail from source, as well as a method to update the base jail using freebsd-update(8).

The following options are available:

- m Fetch and install man pages (ca. 10MB).
- M Fetch and install man pages, without (re)installing the base jail. May be used to add the man pages to the base jail after the initial installation.
- s Fetch and install sources (ca. 450MB).
- S Fetch and install sources, without (re)installing the base jail.
- p Invoke the portsnap(8) utility to fetch and extract a FreeBSD ports tree from portsnap.FreeBSD.org (ca. 475MB). When a ports tree is added to the base jail, a modified make.conf containing reasonable values to function in the jailed environment is added to the new jail template so all jails created from the new jail template will have a working ports environment. See the appendix *Using Portsnap* in the *FreeBSD Handbook* for details or portsnap(8).
- P Fetch and extract a ports tree, without (re)installing the base jail.
- h *host*
Set the remote host to fetch FreeBSD distribution sets from. If absent the default host ftp.FreeBSD.org is used. Variable: “\$ezjail_ftphost”.

It is possible to install from the `disc1` CDRom, or an extracted `-RELEASE` directory, by specifying the `host` argument as `file://path/to/source`.

-r *release*

Install this release of FreeBSD in the base jail, instead of the version returned by “`uname -r`” on the host system. Note that the FreeBSD FTP servers usually provide only `-RELEASE` versions, not `-STABLE` nor `-CURRENT` versions; you will be prompted for confirmation when trying to install a non `-RELEASE` version. If you want to install a `-CURRENT` version, you may have to compile from source the base jail; see the **ezjail-admin update** sub-command for this.

ezjail-admin create

Create a new jail inside ezjail's scope. It either copies the new jail directory tree template or an ezjail archive directory tree to new jail root directory, `/usr/jails/jailname` by default. Jailname and IP address are mandatory parameters.

When a new jail is created, a corresponding new `/etc/fstab.jailname` file is also created, with a `nullfs(5)` mount giving access to the base jail from the new jail.

The following operands are mandatory:

jailname

The name of the jail. It is customary to use the network name of the jail, such as “`jail1.example.com`” (or maybe simply “`jail1`”), but really any name may be used.

It is an error to have several jails of the same name, note that due to ezjail's internal jailname sanitation, “`sand-box.com`” and “`sand_box_com`” are considered identical. Some names such as “`basejail`” and “`flavours`” are reserved for ezjails internal administrative purposes.

ipaddress[, ipaddress2, ...]

The IP address or addresses of the jail. Since FreeBSD 7.2, it is possible to assign several several IPv4 or IPv6 addresses to a jail, by separating them with commas. Previous versions of FreeBSD allowed only a single IPv4 address per jail.

From FreeBSD 9.0 the ipaddresses may be prefixed with an interface name, followed by the pipe symbol. It will then automatically be configured as an alias on that interface when the jail starts. Else **ezjail-admin** will display a warning if the requested address is not found on any interface, and the jail will probably not start.

It is common to bind jails to loopback addresses, so they provide services visible to other jails only.

The following options are available:

-r *jailroot*

Use this name as the directory name of the new jail. Without this option, it is derived from the jail's name. If this option is given and does not start with a `'/'`, it is interpreted as relative to ezjail's root directory (`/usr/jails` by default). If a specified jailroot path lies outside the ezjail root directory, a soft link is created inside `/usr/jails/` pointing to the location of the newly created jail.

-a *archive*

Restore a jail from an archive created with **ezjail-admin archive**. The archive files are kept in `/usr/jails/ezjail_archives` by default. Use `-` to restore an archive from the standard input.

You will probably need to tidy up things inside an ezjail if you migrate it between different ezjail environments. This may include (but is not limited to) reinstalling ports or packages for different CPUs or library versions. You may also need to copy some libraries from the source host's base

jail.

See also **ezjail-admin restore**, if you only want to revert to an old jail's state from an archive on the same release version.

-x This flag indicates that a jail root directory for that jail already exists. In this case, ezjail will only import the jail to its control directory. Sanity checks are performed.

-f *flavour*

Install the requested *flavour* in the new jail. Refer to ezjail(7) for more details on flavours.

This option may not be used with the **-a** option.

-c **simple** | **bde** | **eli** | **zfs**

Create an image jail of the given type.

simple, **bde** and **eli** image jails are file backed memory discs attached as md(4) devices, so the jail can never grow beyond its allocated size and can even be mounted read only. The jail will be stored in a file named *jailname.img*, unless **-r** *jailroot* is given, in which case the jail is stored in *jailroot.img*.

Both **bde** and **eli** jails use the geom(4) framework to encrypt all data written to the image file using gbde(4) (for **bde**) or geli(8) (for **eli**).

Unless you pass some options to the encryption geom commands using the **-C** parameter, you will be prompted for a passphrase to protect the crypto image. Note that, since starting normal encrypted image jails requires user interaction to enter the passphrase, they will **NOT automatically be started at boot time**. Use **ezjail-admin startcrypto** to manually start all crypto image jails.

A **zfs** jail is backed with a zfs(8) filesystem, whose initial quota is given with the **-s** option. The filesystem by default (see the **-z** option) is created in the "\$ezjail_jailzfs" parent filesystem and compressed using the lzjb method, as set in the "ezjail_zfs_jail_properties" variable, both values configured in ezjail.conf(5).

In each case, the **-s** flag is mandatory when creating a file backed jail (i.e. any image that is not zfs backed). An empty directory (without the *.img* suffix in the case of file-based jails) will be created and used as a mount point when running the jail.

-z *parentzfs*

Normally zfs jails are created in a child of the same zfs, ezjail keeps its working directories in, as configured in the "ezjail_jailzfs" variable set in ezjail.conf(5). Use this option to override this default.

This option implies **-c** *zfs*.

-s *imagesize*

Allocate this size to the jail. Without an unit, the size is in bytes. The valid suffix values are b/B for blocks (i. e. 512 bytes), k/K for kilobytes, m/M for megabytes, and g/G for gigabytes. As a reference point, a newly created jail requires 2 MB.

It is not possible to increase the size of file-based jails after their creation, short of creating a new image jail with a larger size.

-C *imageopt*

Pass this argument to gbde(8) or geli(8) when initialising crypto image jails. The **-P** and **-K** (and **-L** for gbde(4)) options will be translated and passed to the respective attach command when starting the jail. You will have to escape parameters with single ticks to protect them from shell expansion.

- i** Synonym of **-c simple**.
- b** Tell ezjail that starting this jail would block unattended reboots. This may happen when certain services need private SSL keys that require the user to interactively enter a passphrase. The jail is then not automatically started at boot time.

ezjail-admin console

Attach your console to the selected jail. You are logged in as root by default.

The following options are available:

- f** Start the jail if it is not running yet.
- e** *command*
Use *command* instead of the default `"/usr/bin/login -f root"`. `login` command. A one time change to use a different user can be accomplished by using **-e** `"/usr/bin/login -f user"`. Variable: `"$ezjail_default_execute"`.

ezjail-admin list

List all jails inside ezjail's scope. They are sorted by the order they start up, as defined by `rcorder(1)`.

The first column is the status flag consisting of 2 or 3 letters. The first letter is the type of jail:

- D** Directory tree based jail.
- I** File-based jail.
- E** Geli encrypted file-based jail.
- B** Bde encrypted file-based jail.
- Z** ZFS filesystem-based jail.

The second letter is the status of the jail:

- R** The jail is running.
- A** The image of the jail is mounted, but the jail is not running.
- S** The jail is stopped.

If present, the third letter, **N**, means that the jail is not automatically started.

The following columns are the JID (when it is running), the IP addresses, the name and the full path directory name of the jail.

ezjail-admin start | restart | stop | startcrypto [jailname ...]

This is a shortcut to the `rc(8)` **ezjail** script. Refer to `ezjail(7)` section `Starting jails` for details.

Note that, if ezjail is not enabled in `rc.conf(5)` with `"ezjail_enable="YES"`, nothing happens.

Since starting crypto image jails requires interaction with the administrator, they are not run at boot time. Use **startcrypto** to run them all at once.

ezjail-admin config jailname

Manage parameters of specific ezjails. For running jails, most of the configuration changes described below will not be applied until the next time the jail is restarted.

The following options are available:

- r run | norun**
Set the jail to be automatically started or not on boot.
- n newname**
Rename the jail. Unless a custom root directory was given with the **-r** flag when creating the jail, the root directory will be renamed as well. A running jail may not be renamed.

-i attach | detach | fsck

Only valid for stopped image jails. Attaching a jail means making the content of the root of the jail accessible from the host. No other sub-commands will function on an jail while its image is attached. With **fsck**, the image jail is attached, `fsck(8)` is run, then the image jail is detached. You can only fsck image based jails.

-z newdataset

Set the given ZFS dataset to be mounted inside the jail file system when it is started.

-f newfib

Change the FIB of the jail (see `setfib(2)`).

-c newcpuset

Change the CPU affinity set of the jail (see `cpuset(2)`).

ezjail-admin delete *jailname*

Delete a jail. By default, this command only deletes ezjail's control file for the selected jail as well as `/etc/fstab.jailname`. The `/usr/jails/jailname` directory is not deleted.

-f Stop the jail before deleting it.

-w Delete the directory or the file backing the jail.

ezjail-admin archive [*jailname*]

Create a backup of one or all jails. The jail's root directory tree is backed up as a `pax(1)` archive. By default, the jail needs to be stopped.

-A Archive all jails. You must neither specify an archivename nor a jailname in this case.

-a archivename

Use this name for the archive file. If absent, the archive file name is derived from the jail name, with the current date and time appended to the archive's file name. Use `-` to write to stdout.

-d directory

Save the archive in this directory. If this option is not given and "`$ezjail_archivedir`" is not set, the archive is saved in the default directory. Variable: "`$ezjail_archivedir`".

-f Archive the jail even when it is running.

Use **ezjail-admin restore** or **ezjail-admin create** **-a** *archive* to restore an archive.

ezjail-admin restore

Create new ezjails from archived versions. It tries to collect all information necessary to do that without user interaction from the user.

The following operand is mandatory:

archive | *jailname*

Restore this jail. If only the jail name is given, **ezjail-admin** will use the most recent archive file matching the name you specified. To restore an older version, specify the complete archive file name (file name with the date and time of the archive appended to it).

The following options are available:

-d archivedir

Search the archive file in this directory. If this option is not given, the archive is searched in "`$ezjail_archivedir`".

- f** Restore the archive even if running on a host different from where it was archived. By default, **ezjail-admin** will refuse to restore an archive if the archived host system's hostname, its FreeBSD version or CPU architecture do not match the current host.

ezjail-admin update

Updates ezjail's basejail, or in the **-b** or **-i** case, install a FreeBSD world from source to be used as basejail.

Exactly one of the following operand must be specified:

- b** Build a world from source and install it as the (updated) basejail. "make buildworld; make installworld" by default using the sources located at `/usr/src` (but see the **-s** option).
As the old basejail is not deleted, but merely overwritten, this usually leaves all jails in a state where they still find older versions of libraries they were linked against.
- i** As above but only perform a "make installworld", assuming the world has already been built. That is highly likely since it is recommended to update the basejail along with the host system.
- u** Use `freebsd-update(8)` to update the basejail. Note that as `freebsd-update(8)` uses "uname -r" to determine the currently running system, the base jail and the host need to be updated at the same time, without rebooting on the new kernel in the meantime.
- U** Use `freebsd-update(8)` to upgrade the basejail to the hosts operating system version, or a version you may pass `freebsd-update`'s call to "uname -r" via the `UNAME_r` environment variable. Since there currently is no way of inferring the osversion currently installed in the basejail, you need to remember the original osversion and pass it to this script using the **-s** option.
- P** Install only the ports tree, assuming the basejail has already been created. This can be done while jails are running. The `portsnap(8)` utility is invoked to do the actual work.

The following options are available:

- p** Give the new basejail a copy of FreeBSD's ports tree. The `portsnap(8)` utility is invoked to do the actual work.
- s** *sourcedir* | *sourceosversion*
In the **-b** and **-i** case: Use the sources in *sourcedir* instead of `/usr/src`. Variable: `"$ezjail_sourcetree"`.
In the **-U** case: Pass this release tag to `freebsd-update(8)` as the source OS version of the basejail.

See the **install** sub command to install the basejail from binary packages.

If the basejail is managed in its own ZFS filesystem, a snapshot of that filesystem is taken first.

FILES

```
EZJAIL_PREFIX/bin/ezjail-admin
EZJAIL_PREFIX/etc/rc.d/ezjail
EZJAIL_PREFIX/etc/ezjail.conf
EZJAIL_PREFIX/share/examples/ezjail/
EZJAIL_PREFIX/etc/ezjail/*
/usr/etc/fstab.*
```

SEE ALSO

`ezjail(7)`, `ezjail.conf(8)`, `jail(8)`, `devfs(5)`, `fdescfs(5)`, `procfs(5)`, `portsnap(8)`.

AUTHOR

Dirk Engling <erdgeist@erdgeist.org>.

The man page is based on a draft by JoeB <joeb1@alpoweruser.com> and was rewritten by Frederic Perrin <frederic.perrin@resel.fr>.