NAME

qemu-nbd − QEMU Disk Network Block Device Server

SYNOPSIS

qemu−nbd [*OPTION*]... *filename*

qemu−nbd −L [*OPTION*]...

qemu−nbd −d *dev*

DESCRIPTION

Export a QEMU disk image using the NBD protocol.

Other uses:

- Bind a /dev/nbdX block device to a QEMU server (on Linux).
- As a client to query exports of a remote NBD server.

OPTIONS

filename is a disk image filename, or a set of block driver options if **−−image−opts** is specified.

dev is an NBD device.

−−object type,id=ID,...
Define a new instance of the *type* object class identified by *ID*. See the**qemu(1)** manual page for full details of the properties supported. The common object types that it makes sense to define are the **secret** object, which is used to supply passwords and/or encryption keys, and the **tls−creds** object, which is used to supply TLS credentials for the **qemu−nbd** server or client.

−p, −−port=PORT
TCP port to listen on as a server, or connect to as a client (default **10809**).

−o, −−offset=OFFSET
The offset into the image.

−b, −−bind=IFACE
The interface to bind to as a server, or connect to as a client (default **0.0.0.0**).

−k, −−socket=PATH
Use a unix socket with path *PATH*.

−−image−opts
Treat *filename* as a set of image options, instead of a plain filename. If this flag is specified, the **−f** flag should not be used, instead the **format=** option should be set.

−f, −−format=FMT
Force the use of the block driver for format *FMT* instead of auto−detecting.

−r, −−read−only
Export the disk as read−only.

−A, −−allocation−depth
Expose allocation depth information via the **qemu:allocation−depth** metadata context accessible through NBD_OPT_SET_META_CONTEXT.

−B, −−bitmap=NAME
If *filename* has a qcow2 persistent bitmap *NAME*, expose that bitmap via the **qemu:dirty−bitmap:NAME** metadata context accessible through NBD_OPT_SET_META_CONTEXT.

−s, −−snapshot
Use *filename* as an external snapshot, create a temporary file with **backing_file=***filename*, redirect the write to the temporary one.

−l, −−load−snapshot=SNAPSHOT_PARAM
Load an internal snapshot inside *filename* and export it as an read−only device, SNAPSHOT_PARAM format is **snapshot.id=[ID],snapshot.name=[NAME]** or **[ID_OR_NAME]**

−−cache=CACHE
The cache mode to be used with the file. Valid values are: **none**, **writeback** (the default), **writethrough**, **directsync** and **unsafe**. See the documentation of the emulator's **−drive cache=...** option for more info.

−n, −−nocache
Equivalent to **−−cache=none**.

−−aio=AIO
Set the asynchronous I/O mode between **threads** (the default), **native** (Linux only), and **io_uring** (Linux 5.1+).

−−discard=DISCARD
Control whether **discard** (also known as **trim** or **unmap**) requests are ignored or passed to the filesystem. *DISCARD* is one of **ignore** (or **off**), **unmap** (or **on**). The default is **ignore**.

−−detect−zeroes=DETECT_ZEROES
Control the automatic conversion of plain zero writes by the OS to driver−specific optimized zero write commands. *DETECT_ZER OES* is one of **off**, **on**, or **unmap**. **unmap** converts a zero write to an unmap operation and can only be used if *DISCARD* is set to **unmap**. The default is **off**.

−c, −−connect=DEV
Connect *filename* to NBD device *DEV* (Linux only).

−d, −−disconnect
Disconnect the device *DEV* (Linux only).

−e, −−shared=NUM
Allow up to *NUM* clients to share the device (default **1**), 0 for unlimited. Safe for readers, but for now, consistency is not guaranteed between multiple writers.

−t, −−persistent
Don't exit on the last connection.

−x, −−export−name=NAME
Set the NBD volume export name (default of a zero−length string).

−D, −−description=DESCRIPTION
Set the NBD volume export description, as a human−readable string.

−L, −−list
Connect as a client and list all details about the exports exposed by a remote NBD server. This enables list mode, and is incompatible with options that change behavior related to a specific export (such as *−−export−name*, *−−offset*, ...).

−−tls−creds=ID
Enable mandatory TLS encryption for the server by setting the ID of the TLS credentials object previously created with the −−object option; or provide the credentials needed for connecting as a client in list mode.

−−fork Fork off the server process and exit the parent once the server is running.

−−pid−file=PATH
Store the server's process ID in the given file.

−−tls−authz=ID
Specify the ID of a qauthz object previously created with the *−−object* option. This will be used to authorize connecting users against their x509 distinguished name.

−v, −−verbose
Display extra debugging information.

−h, −−help
Display this help and exit.

−V, −−version
Display version information and exit.

−T, −−trace [[enable=]PATTERN][,events=FILE][,file=FILE]
Specify tracing options.

[enable=]PATTERN
Immediately enable events matching *PATTERN* (either event name or a globbing pattern). This option is only available if QEMU has been compiled with the **simple**, **log** or **ftrace** tracing backend. To specify multiple events or patterns, specify the **−trace** option multiple times.

Use **−trace help** to print a list of names of trace points.

events=FILE
Immediately enable events listed in *FILE*. The file must contain one event name (as listed in the **trace−events−all** file) per line; globbing patterns are accepted too. This option is only available if QEMU has been compiled with the **simple**, **log** or **ftrace** tracing backend.

file=FILE
Log output traces to *FILE*. This option is only available if QEMU has been compiled with the **simple** tracing backend.

EXAMPLES

Start a server listening on port 10809 that exposes only the guest−visible contents of a qcow2 file, with no TLS encryption, and with the default export name (an empty string). The command is one−shot, and will block until the first successful client disconnects:

```
qemu−nbd −f qcow2 file.qcow2
```

Start a long−running server listening with encryption on port 10810, and whitelist clients with a specific X.509 certificate to connect to a 1 megabyte subset of a raw file, using the export name 'subset':

```
qemu−nbd \
  −−object tls−creds−x509,id=tls0,endpoint=server,dir=/path/to/qemutls \
  −−object 'authz−simple,id=auth0,identity=CN=laptop.example.com,,\
            O=Example Org,,L=London,,ST=London,,C=GB' \
  −−tls−creds tls0 −−tls−authz auth0 \
  −t −x subset −p 10810 \
  −−image−opts driver=raw,offset=1M,size=1M,file.driver=file,file.filename=fil
```

Serve a read−only copy of a guest image over a Unix socket with as many as 5 simultaneous readers, with a persistent process forked as a daemon:

```
qemu−nbd −−fork −−persistent −−shared=5 −−socket=/path/to/sock \
  −−read−only −−format=qcow2 file.qcow2
```

Expose the guest−visible contents of a qcow2 file via a block device /dev/nbd0 (and possibly creating /dev/nbd0p1 and friends for partitions found within), then disconnect the device when done. Access to bind **qemu−nbd** to a /dev/nbd device generally requires root privileges, and may also require the execution of **modprobe nbd** to enable the kernel NBD client module. *CAUTION*: Do not use this method to mount filesystems from an untrusted guest image − a malicious guest may have prepared the image to attempt to

trigger kernel bugs in partition probing or file system mounting.

```
qemu-nbd -c /dev/nbd0 -f qcow2 file.qcow2
qemu-nbd -d /dev/nbd0
```

Query a remote server to see details about what export(s) it is serving on port 10809, and authenticating via PSK:

```
qemu-nbd \
  --object tls-creds-psk,id=tls0,dir=/tmp/keys,username=eblake,endpoint=client
  --tls-creds tls0 -L -b remote.example.com
```

SEE ALSO

qemu(1), **qemu-img(1)**

AUTHOR

Anthony Liguori <anthony@codemonkey.ws>

COPYRIGHT