

Half-Life

Half-Life is a proprietary FPS game.

Dedicated Server

This article will guide you through the installation and configuration of a Half-Life Dedicated Server (HLDS). [Half-Life](#) was [Valve Software's](#) first game. Even though it was released in 1998, it is still very popular among on-line gamers.

Running a Half-Life server is fun for you and your friends. The Half-Life game is resource-friendly and fast -even on not-too-new hardware. Running a Half-Life server allows you to choose the list of maps which will be rotated on your server or other forms of customization (like password-protecting the server so that *you* determine who can join the games).

The Half-Life Dedicated Server has been available for Linux ever since the game was released. However the *game* itself has only recently been ported to Linux. It has become available as the first commercial game for the native Linux port of the Steam client in early 2013. Together with the free-to-play Team Fortress 2 you have two very good reasons to start playing games in the Steam content platform.


Installing the Half-Life server and keeping it updated, requires that you use the Steam updater.

The Steam updater and the game's server binaries are currently all 32-bit, so you must run 32-bit Slackware or a 64-bit multilib setup in order to install and run it.

Deciding on a Server host

The Steam program as well as the game servers that run under Steam, are 32-bit programs. It makes most sense to run the server on a 32-bit Slackware host.

If your server is 64-bit Slackware, and you do not want to install multilib, then you have two options:

1. setup a 32-bit Slackware in a LXC container. Slackware supports LXC (an  [OS-level light virtualization](#) solution) out of the box.
For an example of how to do this see [Ponce's blog](#).
2. setup a 32-bit Slackware in a Virtual Machine (using QEMU or VirtualBox for instance).

The virtual 32-bit Slackware machine does not have to be a full installation. You can set up a minimal Slackware that uses less than 500 MB of disk space. See [this article](#) for more information about a “minimal Slackware” installation.

Preparing your Server

As the “*root*” user, you first create a dedicated user account which the HLDS server will run as. Running a game server as root is a bad idea. Always prepare for the worst and give your services the lowest privilege level that it needs to function.

```
# groupadd steam
# useradd -g steam -m -d /home/steam -c "Steam server account" -s /bin/bash
steam
```

Then set a password for the new “steam” account:

```
# passwd steam
```

As of 14 feb 2013, steam updates to the Half-Life Dedicated Server must be done with the [steamcmd](#) program instead of the old [hldsupdate](#) which no longer runs on a modern distro like Slackware 14.

This requires the use of a Steam user account. Either use your own gaming account (but make sure you are logged out when you run steamcmd) but this is really not recommended. Rather, you should create a separate [Steam account](#) for your server - the accounts are free.

In the example below, I will use the directory “~/server” as the HLDS “working area”, where the SteamCMD utility will download the server binaries and game files, and where we will keep the scripts that operate the server.

The idea is to keep your own downloads and the Steam downloads separate, this avoids issues later on.

You will now continue as the user “steam”:

```
# su - steam
```

- Create a logging directory, we will be using that later:

```
$ mkdir ~/server/logs
```

- Download and extract the archive containing “steamcmd” binary.

```
$ mkdir downloads
$ cd downloads
$ wget http://media.steampowered.com/client/steamcmd_linux.tar.gz
```

We are going to extract this tarball into a separate subdirectory inside our *server* working area:

```
$ mkdir -p ~/server/steamcmd
$ tar -C ~/server/steamcmd -xvf steamcmd_linux.tar.gz
```

Do not extract the contents of the updater tool to the regular Steam Client folder!

- Launch SteamCMD for the first time:

```
$ cd ~/server/steamcmd
$ ./steamcmd.sh
```

The program will automatically update and enter you in to a **Steam>** prompt. Type “help” for more information.

- To download the Half-Life game server files, you login to your Steam server account:

```
Steam> login <username>
```

- Enter your password. Then check your e-mail for your Steam Guard access code and enter it (you will only need to do this once per computer).
You should see a message stating that you have successfully logged in with your account.
- Set your server install directory. In this article, we will use a subdirectory “*hlds_1*” in the server's “working” directory, not a subdirectory of the directory where `steamcmd` is located - we need to enter a full path:

```
Steam> force_install_dir /home/steam/server/hlds_1/
```

- Install or update the HLDS game server files. Every Steam application has its own “*app_id*” or “Steam Application ID” which will be one of the parameters. You can find the *AppId* for the game(s) you want to download here:
https://developer.valvesoftware.com/wiki/Steam_Application_IDs .

```
Steam> app_update <AppID>
```

For Half-Life Dedicated Server, the AppID is “90”. We want to download, install *and* validate HLDS:

```
Steam> app_update 90 validate
```

- Once the download and update is finished, type “quit” to properly log off of the Steam servers.

```
Steam> quit
```

This will have populated the “`~steam/server/hlds_1`” directory (it is of course an arbitrary name, pick any name you like) with the Half-Life Dedicated Server binaries (`hlds*`) as well as full game content for Half-Life in its sub-directory “`valve`”. A total of around 410 MB.

Starting the Game Server

The *SteamCMD* utility will have installed the `hlds` linux binaries in the subdirectory you specified with the “`force_install_dir`” parameter, in our example case, that directory is “`~steam/server/hlds_1`”. The Half-Life game files will have been downloaded to the “`valve`” subdirectory below that (other games such as Counter Strike all get their own separate subdirectory).

You can start the Half-Life game server directly using the following commands:

```
$ cd ~/server/hlds_1
$ ./hlds_run -game valve +maxplayers 4 +map crossfire +exec server.cfg
```

If your server is multi-homed (i.e. it has multiple IP addresses) or you want to run the server on a non-standard port, then you may be required to add commandline parameters like “`+ip 192.168.1.1 +port 27015`”.

Check out the file “`server.cfg`” in the “`hlds_1/valve`” directory, it determines how your HLDS behaves. Some important “`sv_`” variables which you definitely should configure are:

```
// Show my contact email:
sv_contact "alien@slackware.com"

// We report ourselves as an European server:
sv_region 3

// default server name. Change to "Bob's Server", etc.
// hostname "Half-Life"
hostname "Alien BOB's server"

// Allow clients to download maps
sv_allowdownload 1

// Allow clients to upload decals
sv_allowupload 1

// LAN only play
//sv_lan 1

// Use this if clients need a password to join:
//sv_password "sesame"
```

Because this is a game server, you do not want to be forced to keep a terminal opened all the time while the server is running. Therefore it is much more convenient to run the server in a “screen” session, using a start script like this one which also configures your screen session to log all output to a file:

```
$ cat <<EOT > /home/steam/server/start_hlds.sh
#!/bin/bash
cd server/hlds_l
screen -A -mdLS hlds_valve -c /home/steam/.screenrc \
  ./hlds_run -console -game valve -pingboost 2 -pidfile hlds.pid \
  +sys_ticrate 1000 +sv_maxupdaterate 100 +ip 192.168.1.1 +port 27015
+maxplayers 4 +map crossfire +cpu 1
EOT
$ chmod +x /home/steam/server/start_hlds.sh
$ cat << EOT > /home/steam/.screenrc
log on
logfile /home/steam/server/logs/hlds_screen.log
EOT
```

Now you can run this start script and it will start the Half-Life Dedicated server in a screen session. If you want your Slackware server to start the HLDS automatically on boot, you can add lines like this to your `/etc/rc.d/rc.local` file:

```
# Start the Half-Life Dedicated Server:
if [ -x /home/steam/server/start_hlds.sh ]; then
  echo "Starting the Half-Life Dedicated Server:
~steam/server/start_hlds.sh"
  su - steam -c "server/start_hlds.sh"
```

```
fi
```

Automatically update the game server files

It is good practice to let the Steam components update themselves. You can use a cron job to run an update once a day.

Create a wrapper script which updates the game(s) automatically.
Create a new file “~/server/steamwrapper.sh” as follows:

```
$ cat <<EOT > ~/server/steamwrapper.sh
#!/bin/sh
/home/steam/server/steamcmd/steamcmd.sh +runscript
/home/steam/server/hlds_steamcmd_update.txt
EOT
$ chmod 755 ~/server/steamwrapper.sh
```

The *runscript* which is used there, contains the commands which are passed to SteamCMD. The file looks like this:

```
// Update hlds (valve directory, app_id 90)
login my_user my_pass
force_install_dir /home/steam/server/hlds_l
app_update 90 validate
exit
```

And then configure the steam user's cron table to call the script once a day. Run this command to start the crontab editor:

```
crontab -e -u steam
```

and then add this line which will schedule the update at 03:55 every day:

```
55 3 * * * /home/steam/server/steamwrapper.sh 2>&1 | tee -a
/home/steam/server/logs/cron.log
```

Enjoy the game!

Custom server maps for small servers

If you enjoy close-combat games with your friends, some relatively small (less than 8 players) and enjoyable maps can be found here: <http://alienbase.nl/half-life/>.

Make sure that the friends who join your server, have downloaded your server's custom map pack before they join. This will prevent long in-game loading times. Downloading a map “in-game” may take as long as the actual cycle time of the map!

Custom maps go into the server's (and client's) “maps” subdirectory of the “valve” game directory.

Sources

- Originally written by [Eric Hameleers](#)

[howtos](#), [games](#), [server](#), [steam](#)

From:

<https://docs.slackware.com/> - **SlackDocs**

Permanent link:

<https://docs.slackware.com/howtos:games:halflife>

Last update: **2019/02/21 11:38 (UTC)**

