"Valheim Dedicated Server" Instruction Manual (Steam)

Running a dedicated server is a great way to provide a persistent world in which you and your friends can play anytime. Setting up a dedicated server requires some cursory scripting knowledge and, depending on your configuration, knowing how to set up your router to allow data to be sent to and from the server and its connected clients.

This guide will help you script the server application but due to the multitude of router and modem types in use today, and differing internet connection setups, we can't provide any detailed instructions on how to configure your router. Please refer to your router manual or ISP.

Downloading the Dedicated Server program from Steam

1. In the Steam Library, click the top left dropdown menu and make sure the 'Tools'-checkbox is filled.
2. Scroll down to 'Valheim Dedicated Server' and click 'Install'.

Running the Dedicated Server

Before running the Server, you have to manually enter some relevant information in the 'start_headless_server.bat'-file (Linux: 'start_server.sh').

We recommend that you make a copy of this file and enter the required information in the copy instead of the original, since the original file will be reset each time it is updated in Steam (requiring you to re-enter your information). Just be aware that you won't be able to launch the copy from Steam, since that shortcut only points to the original filename.

⚠️ *PLEASE NOTE: If you're using the Steam backend, the most important thing is to make sure that the Server uses a Port Range that has access to the Internet. This is done by 'Port Forwarding', please refer to your Router manual for details on how to do this. The default Port Range that the Server uses is 2456-2457.*
If you're using the Crossplay backend, the server transmits data via a relay server and therefore does not require Port Forwarding to be accessible from outside of your local network. You can connect to a Crossplay server using the public IP address and port number, a join code or via the server list, however it's not possible to connect using a local IP address or a loopback IP address.

1. Navigate to the installation folder. To find the installation folder, right-click the 'Valheim Dedicated Server'-entry in your Steam Library and choose 'Manage/Browse local files'.
2. Right-click `start_headless_server.bat` (your copy *or* the original) and choose 'Edit' to open the file in Notepad (Linux: open a terminal in the server directory and run the bash script: `"./start_server.sh"`).

3. The following properties are the ones we are interested in, they are all located on line 7 which starts with the text `start valheim_server…`.

`-name "My server"`	Enter the name of your server that will be visible in the Server list.
`-port 2456`	Choose the Port which you want the server to communicate with. Please note that this has to correspond with the Port Forwarding settings on your Router. Valheim uses the specified Port AND specified Port+1. Default Ports are 2456-2457. If you're using the Crossplay backend (enabled using the "-crossplay" argument), you do not need to do Port Forwarding on your Router. The Port number is still used to distinguish between multiple servers on the same public IP address.
`-world "Dedicated"`	A World with the name entered will be created. You may also choose an already existing World by entering its name.
`-password "Secret"`	Set the password.

`-savedir [PATH]`	Overrides the default save path where Worlds and permission-files are stored.
	
	Default path
	WINDOWS:
	`"../%USERPROFILE%/AppData/LocalLow/IronGate/Valheim"`
	LINUX:
	`"~/.config/unity3d/IronGate/Valheim"`

`-public 1`	Set the visibility of your server. 1 is default and will make the server visible in the browser. Set it to 0 to make the server invisible and only joinable via the 'Join IP'-button. Setting public to 0 is a good option if you wish to run a local LAN server, where players can join via the local IP of the server.
`-logFile "d:\log.txt"`	Sets the location to save the log file.
`-saveinterval 1800`	Change how often the world will save in seconds. Default is 30 minutes (1800 seconds).
`-backups 4`	Sets how many automatic backups will be kept. The first is the 'short' backup length, and the rest

	are the 'long' backup length. By default that means one backup that is 2 hours old, and 3 backups that are 12 hours apart.
`-backupshort 7200`	Sets the interval between the first automatic backups. Default is 2 hours (7200 seconds).
`-backuplong 43200`	Sets the interval between the subsequent automatic backups. Default is 12 hours (43200 seconds).
`-crossplay`	Runs the Server on the Crossplay backend (PlayFab), which lets users from any platform join. If you do not include this argument, the Steam backend is used, which means only Steam users can see and join the Server.
`-instanceid "1"`	If you're hosting multiple servers with the same port from the same MAC address, write something unique here for each server to ensure that they get unique PlayFab IDs.
`-preset hard`	Sets world modifier preset. Setting a preset will overwrite any other previous modifiers. Valid values are: ***Normal, Casual, Easy, Hard, Hardcore, Immersive, Hammer.***
`-modifier raids none`	Sets chosen world modifier with value. If combined with a preset should be set after. Valid modifiers and values are: ***Combat***: *veryeasy, easy, hard, veryhard* ***DeathPenalty***: *casual, veryeasy, easy, hard, hardcore* ***Resources***: *muchless, less, more, muchmore, most* ***Raids***: *none, muchless, less, more, muchmore* ***Portals***: *casual, hard, veryhard*
`-setkey nomap`	Sets a world modifier checkbox key. Valid values are: ***nobuildcost, playerevents, passivemobs, nomap***

4. Once you're done setting up (don't forget to save the file) you can start the Server either by launching it directly in Windows or via Steam. If you get a Windows Firewall-popup please fill all checkboxes to allow the Server to communicate with the Internet.

5. When the Server program shows the message 'Game server connected' the Server is up and running, ready to accept connections.

Stopping the Dedicated Server

1. When you wish to stop running the server, **it is important that you close it by pressing CTRL+C in the Command-window**. If you close it by clicking the X in the window frame the Server may keep running in the background, we don't really know!

Administrating the Server and setting permissions

You can edit three separate text-files to set 1) who has admin privileges, 2) who is banned from your server, and 3) who is permitted on your server.

These three files, located in the default save path, are called `adminlist.txt`, `bannedlist.txt`, and `permittedlist.txt`. Add one Platform User ID per line to set desired roles. The Platform User ID can be obtained from the Server log or from within the game using the `F2` panel and follows the format [Platform]_[User ID] (case sensitive).

Please note that adding a person on the permitted list will ban everyone else from the server.

In-Game: Admin console commands (`F5` to bring down console)

`Kick PLAYERNAME`	Kicks a player from the Server.
`Ban PLAYERNAME`	Bans a player from the Server.
`Unban PLAYERNAME`	Unbans a player.
`Banned`	Lists all banned players.

Required packages

Linux Steam client has a large overlap with Valheim dedicated server with respect to library requirement. The following additional packages are required

- libatomic1
- libpulse-dev
- libpulse0

Running via Docker

The dedicated server has minimum requirements of GLIBC_2.29 and GLIBCXX_3.4.26. If your Linux installation uses older versions of those libraries, then one alternative to upgrading to a newer version is to run via Docker.

Installation

Most distributions include a Docker package, installation instructions for several distributions can be found [here](#).

Make sure the docker daemon is running and that the user running the dedicated server is allowed to use docker, typically by being a member of the 'docker' group.

The following command should succeed, make sure associated servers are started if you see this

```
$ docker ps
Cannot connect to the Docker daemon at
unix:///var/run/docker.sock. Is the docker daemon running?
```

and make sure the logged in user is part of the 'docker' group if you see this

```
$ docker ps
Got permission denied while trying to connect to the Docker
daemon socket at unix:///var/run/docker.sock: Get
http://%2Fvar%2Frun%2Fdocker.sock/v1.24/containers/json: dial
unix /var/run/docker.sock: connect: permission denied
```

Using the Docker script

The dedicated server ships with a script named `docker_start_server.sh` location in the same directory as the `start_server.sh` script. Run the `docker_start_server.sh` and pass the server startup-script to run, either the copy you made, or `start_server.sh`

```
$ ./docker_start_server.sh start_server.sh
```

The script will run a boot-strap sequence where it builds the container environment the first time, which takes a few minutes. Consecutive runs will start right away.

Game data is stored in the Docker volume named `DOCKER_DATA_VOLUME` in the `docker_start_server.sh` script. The default is to store the game data in a volume named `valheim_server_data`. If you want to store that information in your user's filesystem, change `valheim_server_data` to the path where the data should be stored.

```
DOCKER_DATA_VOLUME="${HOME}/valheim_storage"
```

will save the data in your home-directory under the folder `valheim_storage`. Make sure to create the directory if it doesn't already exist before executing the script. The worlds, `adminlist.txt`, `bannedlist.txt`, and `permittedlist.txt` are all located inside this directory.

Server is stopped the same way as when running the script directly on the host machine.