How to make One hour one life server tutorial

grzegorz2047 (53) ▾ **(/@grzegorz2047)** in **onehouronelife (/trending/onehouronelife)** • last month

---

Hi, in this tutorial I would like to show you how to create a One hour one life server. If you ever wanted to play this game on your private server. After this tutorial, your server should be up and running :)

English version

Requirements

To create server, we will use a linux operating system. In my case I will use ubuntu 16.04 server. We need to get some tools to compile server for our usage.
All you need is in this case linux server. You can get it by buying VPS for few bucks.
Below you can find command which will download all required tools to build executables.
Type this in console and press enter

```
sudo apt-get install git g++ imagemagick xclip libsdl1.2-dev lib
glu1-mesa-dev libgl1-mesa-dev
```

additionally, we will need screen software to make server run in background. You can install it like that:

```
sudo apt-get install screen
```

Once you get all required libraries and tools, we need to download source files which we got when we bought the game.
The url to site with source code should look like following:

```
http://onehouronelife.com/ticketServer/server.php?action=show_do
wnloads&ticket_id=yourgamecode
```

It should look like that:

We need file called `OneLife_Live4_UnixSource.tar.gz`.
We can download it using command wget:

```
sudo wget http://download6.onehouronelife.com/downloads/OneLife_Live4_UnixSource.tar.gz
```

Of course link can change so you need to copy it for yourself. If you downloaded server files, we need to extract them using this command:

```
sudo tar xzf OneLife_Live4_UnixSource.tar.gz
```

Our files are extracted, now we need to proceed with our server files using command:

```
cd OneLife_Live4_UnixSource
```

Now, we need to execute script which will get and compile everything we need like that:

```
sudo ./pullAndBuildLatest
```

Above script will get source code of a game from github repository so it will take a while and also it has to compile everything.
After it's done we have basically a compiled game. It will be needed to create our first server.
Next step is to go to server files using command:

```
cd OneLife/server
```

It means that we go from folder OneLife_Live4_UnixSource to folder OneLife and then folder server.
Now we have to compile our server files using command:

```
sudo ./configure
```

We should see something like that

```
select platform:
  1 --   GNU/Linux
  2 --   MacOSX
  3 --   Win32 using MinGW
  4 --   Raspbian on Raspberry Pi (experimental)
  q --   quit
```

Our choice is probably linux if not, choose as you need.

After that we have to execute:

```
make
```

And almost last step is to link some game files to server files using commands line by line:

```
sudo ln -s ../../OneLifeData7/objects .
sudo ln -s ../../OneLifeData7/transitions .
sudo ln -s ../../OneLifeData7/categories .
```

Now, finally we can run our server like that:

```
sudo ./OneLifeServer
```

If you did everything correctly, you should see something like this:

```
Server starting up

lifeLog starting up
Backup system inited.  Backups on = 1, target hour UTC = 8, current hour UTC = 20, last backup was 422516.460278 hours ago

Press CTRL-8 to shut down server gracefully

Loaded 0 animations from animations folder
Decompressing took 0.009000 seconds
Loaded 608 objects from objects folder
  Auto-generated 321 'used' objects
Decompressing took 0.000000 seconds
Loaded 19 categories from categories folder
Decompressing took 0.000000 seconds
Loaded 844 transitions from transitions folder
Auto-generated 330 transitions based on categories
```

Now, to connect as player, we have to set server ip and enable custom server connection in player configuration.

First, we have to go to our game "one hour one life" files located on our PC or other hardware we got.

Next is to find `settings` folder. And in that folder, we need file called `customServerAddress.ini`. This file we just edit using software like notepad or other text editor. We should see localhost address. We change it to ip of our server. After we save it and close it.

Next file which we need to change is called `useCustomServer.ini`. There should be number `0`. When we change it to `1`, game will connect directly to our server.

If we changed server port, then we have to change it in game file called `customServerPort.ini`.

Last setting we might want to change is when our server wants password. If so, we have to set it in file called `serverPassword.ini`.

If we did all described above, we are ready to go.

Let's run our game on our computer and click login using our credentials.
We should be on the server and our console should look like this:

```
Listening for another connection on port 8005
Got new player logged in
Placing new Eve:  Found an existing camp at (0.000000,0.000000) with 1 pla
Placing new Eve:  trying radius of 102 from camp
New player grzegorz2047@gmail.com connected as player 3
Compressed 5922 bytes down to 502
Starting new web request for http://onehouronelife.com/reviewServer/server
Logging Eve death:   Eve died too young (age=14.118150, min=60.000000), no
Closing connection to player 3 on error (cause: Socket read failed)
0 remaining player(s) alive on server
Starting new web request for http://onehouronelife.com/reviewServer/server
7F6CD539289D45DC156739
Server log_game request rejected by stats server
```

Success, we created our server and it works :)

If you want to change server port, let's get back to server console.
First, let's turn off our server. It's not in background yet, so just use shortcut `ctrl + d` which turns off server.
We should be in path `/OneLife_Live4_UnixSource/OneLife/server`
We have to go to folder called `settings` using command:

```
cd settings
```

Now, let's find files `maxplayers.ini`, `port.ini`, `requireClientPassword.ini` and `clientPassword.ini`.
Names of files are self-explaining, so I will talk only about enabling server password. If you want to enable server password, head to file called `requireClientPassword.ini` set value to `1` instead of `0` and set your password in file `clientPassword.ini`. After those steps, you configured all basic configurations.
Now we want to use screen software to run server process in background. It means that server will not turn off after we close a terminal (console).
Let's head back to folder called `server`. If we are in folder settings, we can go back using command:

```
cd ..
```

Now let's run the server in the background using command:

```
sudo screen -dmS oneliveserver ./OneLifeServer
```

After using this command, our server process is in background mode. To attach to server process, type:

```
screen -r oneliveserver
```

Above command will bring you back to your "One hour one life" server console. To back into background mode, you need to use shortcut key `ctrl + a` and `ctrl + d`. It will detach server process and put it into background mode so it will run still.

I guess that's all you need to know to make "One hour one life" server up and running.

Good luck!

All links used to make this tutorial possible:

https://onehouronelife.gamepedia.com/Advanced_Settings#Custom_Server

https://github.com/jasonrohrer/OneLife/blob/095754746d35d3835097be4341 b0df9ba9aca9e3/documentation/html/compileNotes.php
https://github.com/jasonrohrer/OneLife/blob/fd85518f76bad3ecf19a4540845 6e957ee11ea63/documentation/EditorAndServerBuildNotes.txt
http://onehouronelife.com/ticketServer/server.php? action=show_downloads&ticket_id=yourgamecode

If you like my work contribution to steemit, vote on me here: https://steemit.com/~witnesses (https://steemit.com/~witnesses) with nick `grzegorz2047`

onehouronelife (/trending/onehouronelife) server (/trending/server) tutorial (/trending/tutorial)

gaming (/trending/gaming) linux (/trending/linux)

last month by **grzegorz2047** (53) ▼ (/@grzegorz2047) **Reply** | 💬 0

⌃ **$ 1.81** ▼ 20 votes ▼ (/onehouronelife/@grzegorz2047/how-to-create-server-for-a-game-one-hour-one-life) f 🐦 in

🔗