



# Survial Engine Online

Installing Cloud Server



## Cloud Server Installation

Follow these steps to create a server and host the lobby server and/or the game server.

In this example, we are going to use **Digital Ocean**, but the server application can be hosted on any cloud provider of your choice.

If you do not have an account already, you can use this link to create an account, and the first \$100 will be free of charge. This should be enough for a few months of free dev server.

<https://m.do.co/c/ca46a35fd4dd>

## Prerequisites

You will need to download these two applications in order to access your server. One is a SFTP file manager and the other is a SSH command line tool. These tools are for Windows, but there are plenty of SFTP and SSH alternatives on Mac/Linux.

WinSCP: <https://winscp.net/eng/index.php>

Putty: <https://www.putty.org/>

When installing Putty, make sure to also install PuttyGen (it is to generate ssh keys).

## Create a Droplet

After you created a Digital Ocean account and entered your credit card information, and that your account is ready to use, you are ready to create a Droplet.

- 1) In Droplets section, click on Create -> Droplet
- 2) Select the OS: I use Ubuntu 22.04
- 3) Choose the specs of your server  
I recommend 2+ CPU.  
(For dev the \$20 server should be more than enough, but if you have a lot of players, you will probably need to upgrade that later.)
- 4) Choose any region (closest to you will have better performances).
- 5) Select a SSH key (see next section to generate a new one)
- 6) Select a hostname for your server (can be anything).
- 7) Click Create Droplet

Create Droplets

Choose an image ⓘ

Distributions Marketplace Snapshots Backups Custom Images

Ubuntu 22.04 x64 Select version Select version Select version Select version Select version

Choose a plan ⓘ Help me choose ⓘ

SHARED CPU Basic General Purpose CPU-Optimized Memory-Optimized Storage-Optimized

DEDICATED CPU

Basic virtual machines with a mix of memory and compute resources. Best for small projects that can handle variable levels of CPU performance, like blogs, web apps and dev/test environments.

CPU options: Regular with SSD Premium Intel with NVMe SSD (new) Premium AMD with NVMe SSD (new)

\$7/mo \$0.002/hour	\$14/mo \$0.021/hour	\$21/mo \$0.031/hour	\$28/mo \$0.042/hour	\$56/mo \$0.083/hour	\$112/mo \$0.167/hour
1 GB / 1 Intel CPU 25 GB NVMe SSDs 1000 GB transfer	2 GB / 1 Intel CPU 50 GB NVMe SSDs 2 TB transfer	2 GB / 2 Intel CPUs 80 GB NVMe SSDs 3 TB transfer	4 GB / 2 Intel CPUs 80 GB NVMe SSDs 4 TB transfer	8 GB / 4 Intel CPUs 160 GB NVMe SSDs 5 TB transfer	16 GB / 8 Intel CPUs 320 GB NVMe SSDs 6 TB transfer

Once your droplet is ready, the IP of your server will be displayed. You will need that IP later when creating the unity build.



## Generate SSH key with PuttyGen

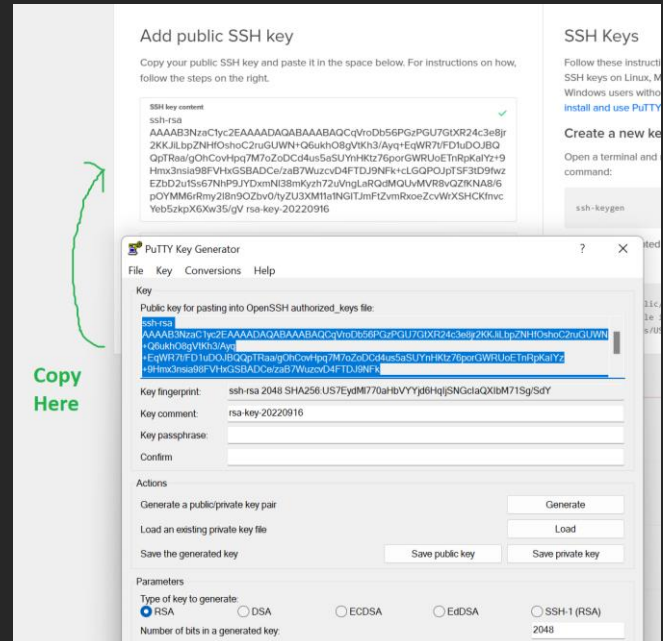
To generate a key, open PuttyGen and click on generate.

Once done, save the private key file on your computer.

Then copy the public key displayed on screen to Digital Ocean.

On Digital Ocean, Add a name to your key and then click on Add SSH key.

Make sure to store your private key file in a secure place, if you lose the key you won't be able to access your droplet.



## Create the Server Build in Unity

There are 2 builds you need to create: ServerLobby and ServerGame.

The lobby is an independent app that allows players to find each other, once a game is started by the lobby, players will be redirected to the game server. The game server will be launched once for every game created. While the lobby server never has more than 1 instance running. You can create your builds from File->Build Settings.

### Game Server

When building the game server. Make sure to add the **ServerGame** scene as the first scene. You also need to include **all the game scenes** you want players to be able to play in. Since our Digital Ocean server is on linux, select **Linux** as the platform on x86\_64, and check the box Server Build.

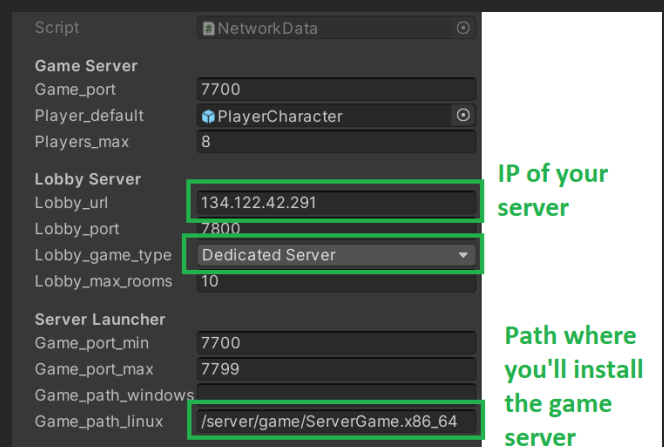
### Lobby Server

Before building the lobby, you will need to go inside Resources folder and select NetworkData file.

There you need to set 3 things: lobby\_url, lobby\_game\_type, and game\_path\_linux.

(The path is absolute, unless it starts with .. or ./ then it will be relative to Application.dataPath)

When building the lobby server. You only need to include 1 scene: **ServerLobby**. Since our Digital Ocean server is on linux, select **Linux** as the platform on x86\_64, and check the box Server Build.





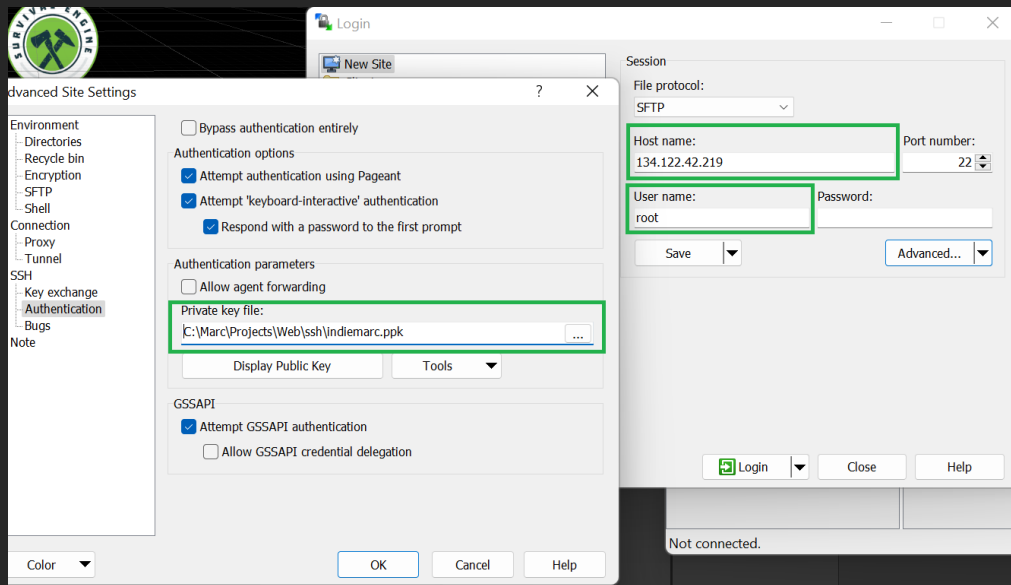
## Connect to your server with WinSCP

Open WinSCP and create a new connection.

Enter the IP of your server as Host, and enter Root as the username.

The password is blank since we use a SSH key instead.

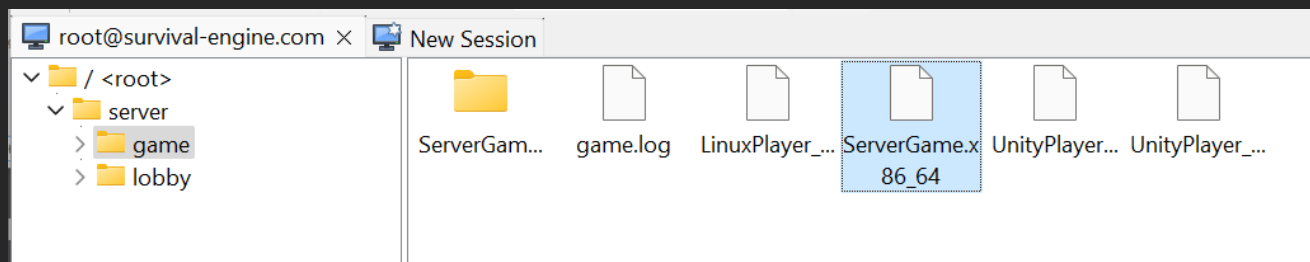
Click on Advanced, then SSH/Authentication, then select your private key file there.



This will give you access to the server files, now you can upload your builds into 2 separate folders, in my demo I created these folders:

/server/game

/server/lobby



To access the **command line**, from WinSCP, click on Commands->Open in Putty.

Once you gain access you can navigate to the server folder with

**cd /server**

Now we need to give the server permission to execute our build, to do that, we can run

**chmod -R 755 game**

**chmod -R 755 lobby**



## Test the Game Server

Run the game server to test it:

```
cd /server/game
```

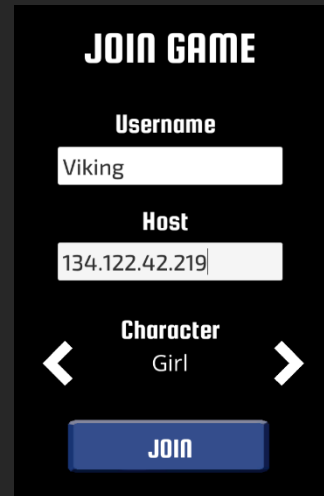
```
./ServerGame.x86_64
```

Now you should be able to connect to it from your client build (or from unity). Launch the game in Unity from the regular menu, click on join, and use the server IP as host.

If you manage to connect, this means your game server is working!

If you can't connect or receive an error about the scene not found, make sure that the Blank scene is enabled in the Build Settings (this is the default scene the game server runs on, you can change that default scene in ServerGame scene).

When the game server is launched by the lobby, the scene will be passed as parameter instead of running the Blank scene.



## Test the Lobby Server

```
cd /server/lobby
```

```
./LobbyGame.x86_64
```

In Unity, in Resources/NetworkData, make sure the lobby\_url is set correctly, since this is the IP your client will try to connect to.

Now from Unity, launch the LobbyMenu scene and try connect. If you reach the list of games, this means you can connect to your lobby. You can now try to create a game and start it.

**Important:** Your lobby server relies on being able to launch the game server application for each game room created, so make sure your game server is properly setup before you run the lobby in Dedicated Server mode. If you run the lobby in Peer-2-peer or in Relay mode, you don't need to upload and install a game server since the lobby will make players connect directly to each other.



## Run the lobby as a Service

While running directly the game and lobby is a good way to test (since you can see the live debug console). This method isn't persistent and your app will stop as soon as you leave putty. To create a persistent server you need to create a service.

In Unity, go to SurvivalEngine/Install folder, there should be a lobby.service file there. Upload that file to your server at this path:

**/etc/systemd/system**

Open the file you just uploaded (double click in WinSCP) then make sure the path to the lobby server is set correctly.

To start/restart/stop the service, you can use these commands from putty ("lobby" is the name of your service file).

**systemctl start lobby**  
**systemctl restart lobby**  
**systemctl stop lobby**

You can also run this command to check if the service is running (and use ctrl-c when you want to quit top)

**top**

Now that your service is running, you can try to connect to it from the client build or from Unity. In the same way you connected to it before.

## Firewall setup

It's good to block the port you don't use on your server, from the command line, you can run

**ufw allow ssh**  
**ufw allow 7700:7800/udp**  
**ufw allow 7700:7800/tcp**  
**ufw enable**

And to check the status of your firewall:

**ufw status**

Be sure to always run allow ssh before enabling the firewall, otherwise you will lose your putty connection and won't be able to connect anymore.