Genshin Impact Offline (GIO)

Docker VM for ALL versions (CB1,2,3, 1.0.0 - 3.2.0)

Guide version: 14

Date: 2023.05.06

GIO Guide book for braindead

Guide authors 4
Useful links 4
GIO – Genshin Impact Offline server 4
Prepare your hardware 5
Special toolbox server-requirements.rar 6
Download your favorite game client version 6
Patch your game client version 7
Method 5. GIO Dockers configs for all game versions for “braindead” players 8
Prepare your system 8
Configure Fiddler 11
Configure Docker Desktop 13
Download docker configs 15
Download server bins + data for bins 15
Suitable bins + data list: 16
How to use suitable server data 21
How to use suitable server bins 24
NavMesh and SVO (Sparse Voxel Octrees) 26
How to setup NavMesh 27
Start server bins in Docker Desktop 29
Start server bins + data in Docker Desktop 30
Fix WSL Docker memory (aka white game screen after gates) 34
Start fiddler with special fiddler script 34
Useful info about version.txt in CB (Closed Beta) versions 38
Database access (phpMyAdmin) 38
Database access (adminer) 39
Check available versions in Database 40

GM Cheat commands and MUIP tool 43
MUIP Tool sign key 48
How to use GM commands without MUIP Tool 50
Alternate GM GUI 51
All known GM cheat commands 52
Unknown GM cheat commands 54
How to use IDA/HxD/etc to see other GM commands 59
Adjust events time limit 60
Fix buggy 80001 and 20001 quests for 3.2 server 61
Fix Spiral Abyss for 3.2 server 61
Better Solo Mod for 3.2 server 61
Change banners 61
GM mail 61
Disable telemetry 62
Proto – what is that? 63
Proto dumping 64
KCP/Enet and UDP/TCP 65
Proto shifting on the fly using KCP Proxy 66
Method 2. VM-Installer - run server without Docker in Linux/VMware/Virtualbox/WSL 67
Download installer + script 68
Prepare your virtual machine 68
Method 3. Run already prepared server VM image using VMWare only 80
Method 1. Run server in Linux native/WSL without Docker/Vmware/Virtualbox 93
TCG (aka GCG) game from 3.3 support in 3.2 server 113
Do you really read all pages? 115

Guide authors

Bios#5368, Hotaru#9643, PartyPooper#4658, DiXiao#6611, qawo#0077

And some unknown authors...

Useful links

Our Discord GIO server [Kinky Rooms]:

https://discord.gg/MfxYRNdD9x

Topic:

https://forum.ragezone.com/threads/genshin-impact-leak-docker-vm-for-all-versions-cb1-2-3-1-0-0-3-2-0.1211748/

GIO – Genshin Impact Offline server

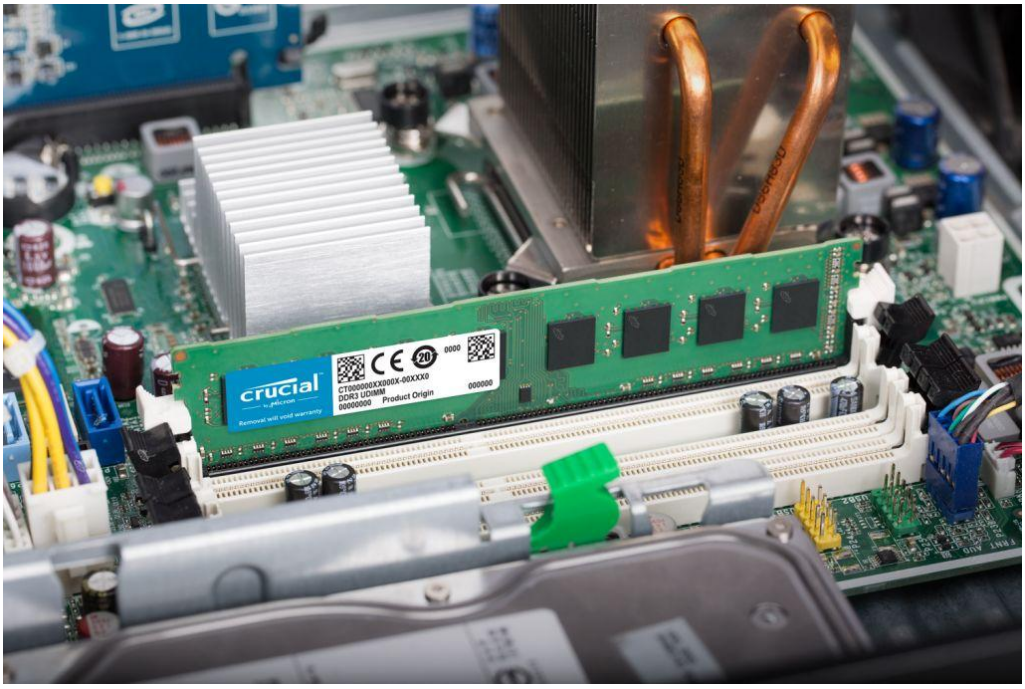
Players wishing to play *Genshin Impact* on private servers have been running into a brick wall, because it's completely prohibited by publishers miHoYo. As *Genshin Impact* operates a gacha system which allows players to pay both real life and in-game currencies on characters, the game must constantly be online to function. This is despite it being a mostly single-player game, which has left many frustrated and wishing for their own private servers to limit the intrusiveness of the online gacha storefront. With the company line telling players that wishes can be bought with Primogems which are free to earn in the game, many players have expressed their frustration at having to grind gameplay for many hours to accrue enough Primogems to spend. Through social media channels this is a contentious issue, with many agreeing with this point and others arguing that the system is a fair model for a free-to-play-game. Either way, the demand for private servers has only increased with time.

Using GIO you can play in offline mode with unlimited resources (primogems, etc.)

Prepare your hardware

- You must have **20+ GB RAM**. It's not a joke. Better do **32 GB RAM** setup, otherwise you can't boot it up. **JUST GIVE UP right now if you haven't enough RAM.**

This is an example of a very bad hardware setup:



This is an example of a very good hardware setup for GIO:

Special toolbox server-requirements.rar

First - download all-in-one useful scripts that can help you in different start methods:

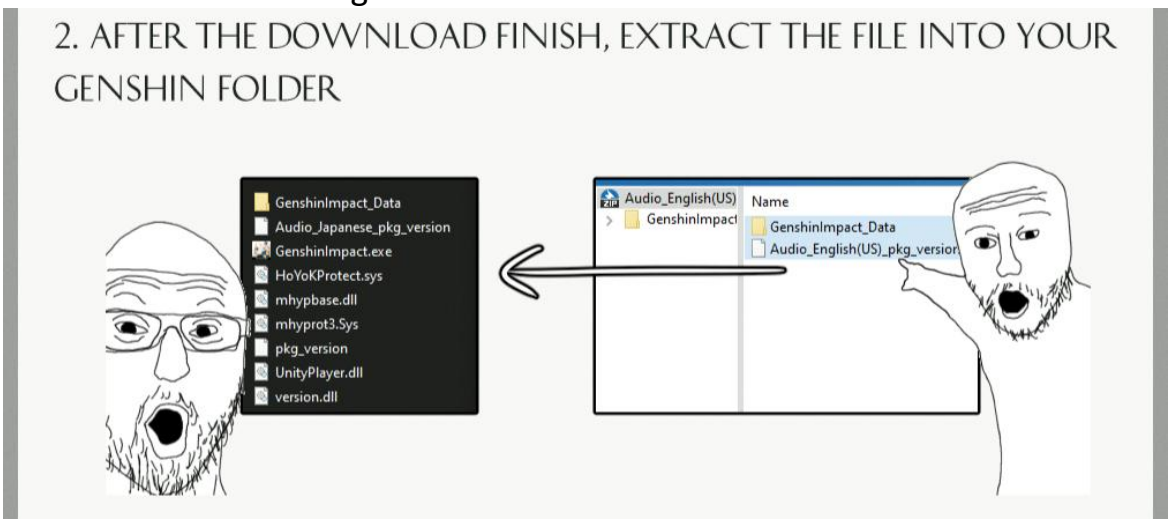
https://anonfiles.com/l602F8odz1/server_requirements_rar

https://upload.biosnod.ru/download/fecdf2778cb6a1c988c13544cb3eb9c5

Download your favorite game client version

All game clients + audio you can download here:
https://file2.yuuki.me/GD1/Project/GenshinImpact/Data/PC
https://github.com/MAnggiarMustofa/GI-Download-Library

Just extract game archive to some folder (without any launcher). Starting at 1.5.0 game version all audio has different archives (not in game archive). Just extract the audio to the extracted game folder like this:



CB = Closed Beta Test (Pre-releases before **1.0.0_rel**)
CB1 = **CB1 1.0.0** (not to be confused with **1.0.0_rel** after **CB3**!)
CB2 = **CB2 0.7.0** (yes, **0.7.0** after **1.0.0** – weird)
CB3 = **CB3 0.9.0**
1.0.0_rel = **1.0.0** release (not to be confused with **CB1 1.0.0** pre-release!)

Other versions of the game are numbered from **1.0.0** to **3.5.0**+
Game version **1.0.0** is **NOT** **CB1 1.0.0**. It's different. After **CB1 1.0.0** next version is **CB2** - have **0.7.0** - it's OK, it's just that the game developers have problems with their heads

Patch your game client version

All patches for all game versions you can download here:
https://anonfiles.com/2bJev3o4za/login_rsa_mhyprot_patches_rar
https://upload.biosnod.ru/download/ac1b9d0fe01cbe2f192ab83333f7ba33

For **CB1** you should use patched **UserAssembly.dll** to skip files checking and your system language should be English only! Otherwise you will see a lot of errors (after gates in game).

For **CB2** and **CB3** you should use patched **mhyprot2.sys** or **mhyprot.sys** (otherwise your game will be closed immediately after run, If one doesnt work - try to use another one)

Also for **CB2** and **CB3** (also for **1.0.0, 1.1.0** and **1.2.0**) you should use ProxyCap + Fiddler both (cuz this versions avoid proxy, yeah, ProxyCap will force this versions to use Fiddler), guide for ProxyCap here (read readme.txt inside) -
https://upload.biosnod.ru/download/ce8375c48bc2aca65e910d675c149a0e
https://anonfiles.com/l0K3vcofz3/ProxyCap_braindead_v2_rar

Other versions - Fiddler only without ProxyCap

For 2.8, 3.0, 3.1 and 3.2+ you should use patched **global-metadata.dat** or **UserAssembly.dll** or **RSAPatch.dll (aka mhybase.dll)**

You shouldn't do any .dll patches for 1.3.0 - 2.7.0, all works without patches for this versions

Patches only for CB1(1.0.0), CB2(0.7.0), CB3(0.9.0) and for 2.8.0 up to 3.2.0+

Method 5. GIO Dockers configs for all game versions for “braindead” players

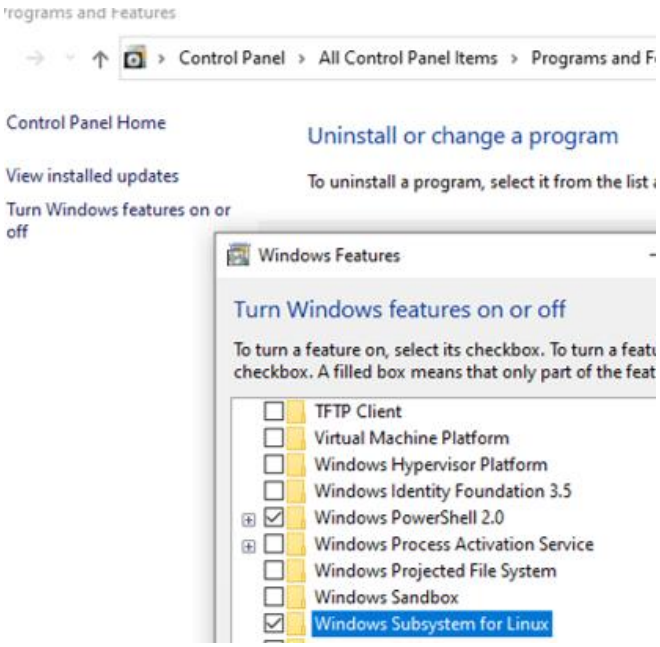
Hello, I am one of the contributors of the Docker configs for the servers =)
I can start every game version, CB1, CB2, CB3 and 1.0.0 up to 3.2.0 and I already tested all of them. All works fine for me, I will help you, just write your troubles in our Discord. I will update this post and provide new Dockers from time to time.

Prepare your system

- For Windows enable WSL (Windows Linux Subsystem) - open Start:

➔ Control Panel
➔ All Control Panel Items
➔ Programs and Features
➔ Turn Windows features on or off

Check Windows Subsystem for Linux, click OK, and restart your computer.

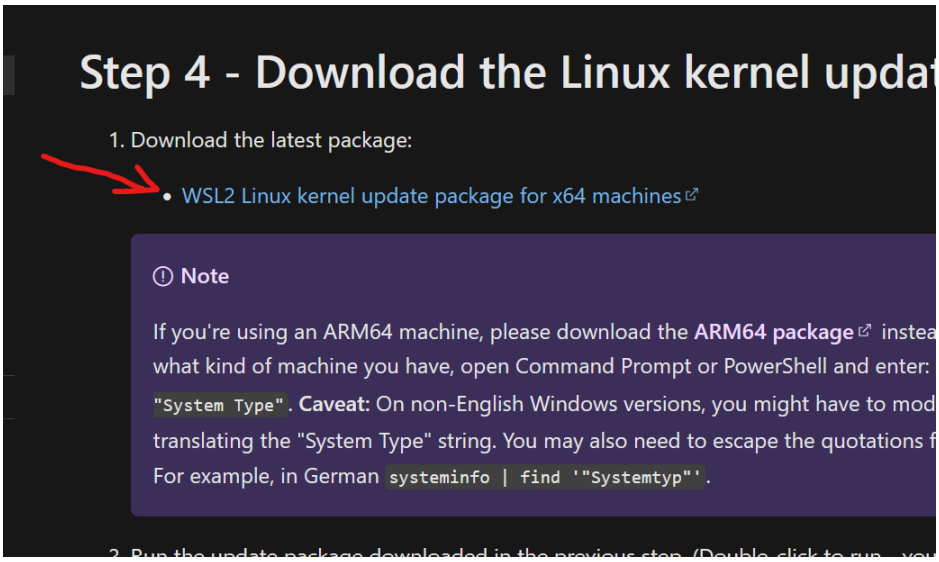


- Install WSL2 update:
https://wslstorestorage.blob.core.windows.net/wslblob/wsl_update_x64.msi
https://anonfiles.com/Y2E8v4o7z2/wsl_update_x64_msi

Or use wsl_update_x64.msi from the “**server-requirements.rar**”

Also, you can learn about WSL here:

https://learn.microsoft.com/en-us/windows/wsl/install-manual



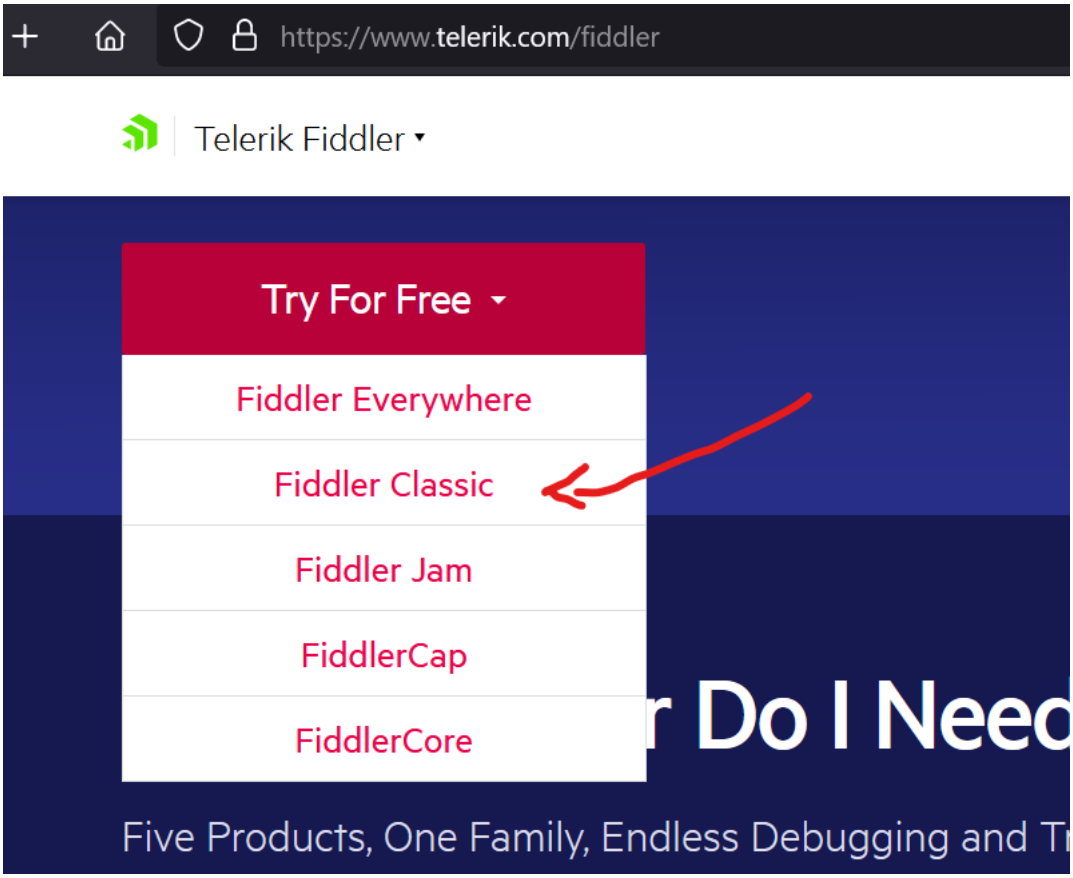
And here: https://learn.microsoft.com/en-us/windows/wsl/install

- Install Docker Desktop for Windows or just Docker for Linux

https://www.docker.com/products/docker-desktop/

- Install Fiddler (Windows only, for Linux search another proxy)
https://www.telerik.com/fiddler

https://anonfiles.com/96aao6oez1/FiddlerSetup_exe

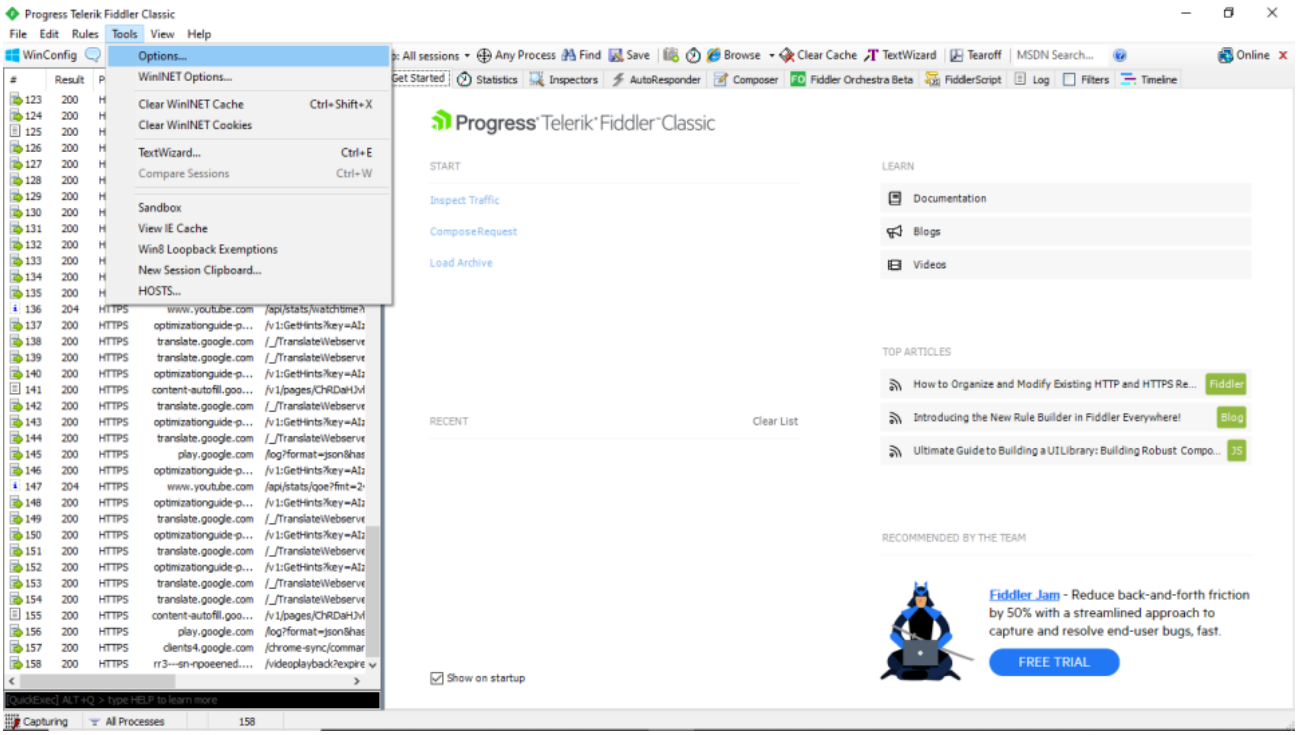


Or use **FiddlerSetup.exe** from the "**server-requirements.rar**"

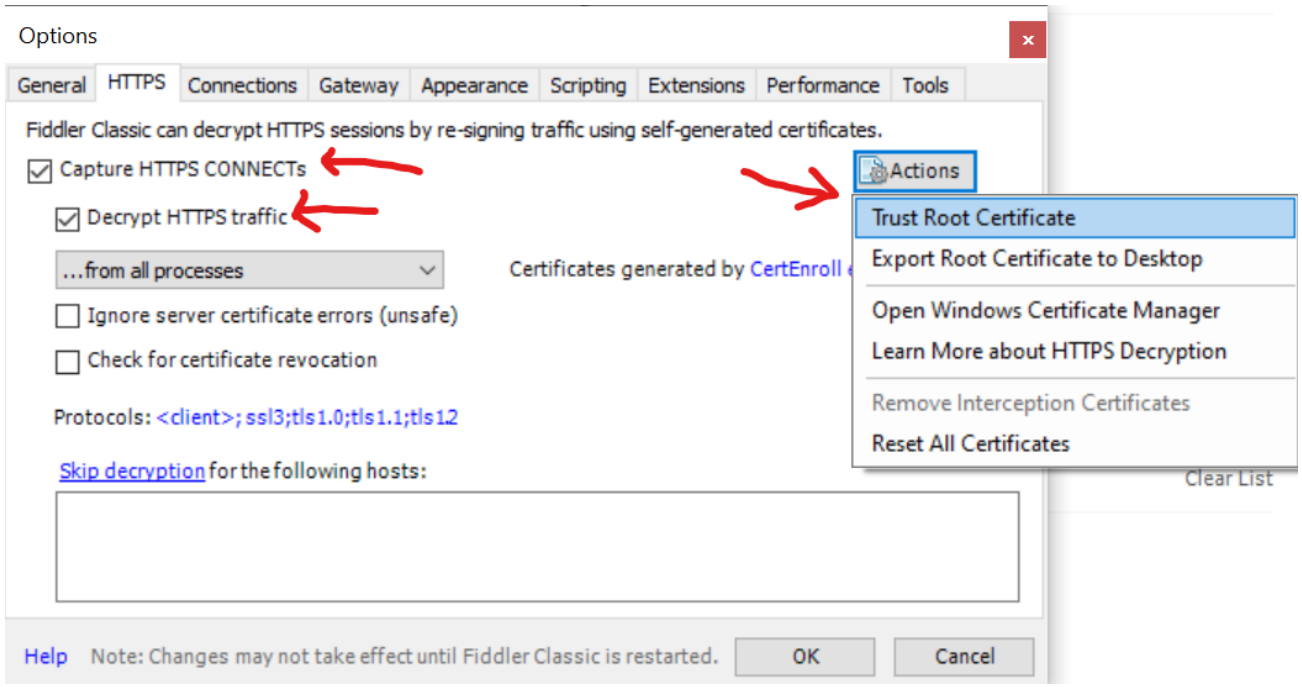
Configure Fiddler

- Enable HTTPS decrypt in Fiddler:

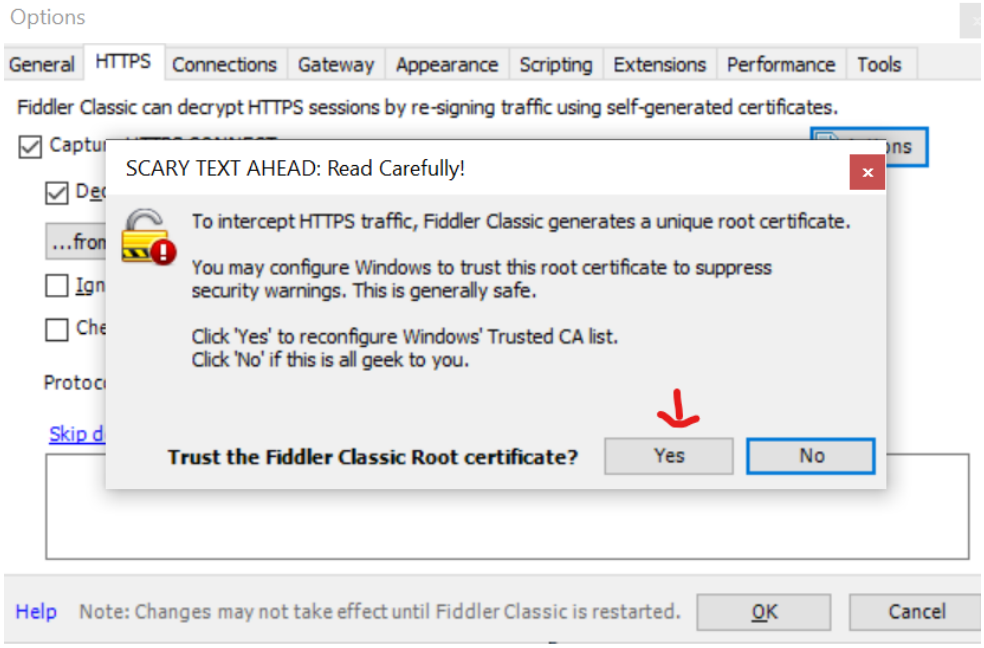
1. Click **Tools > Options > HTTPS**.

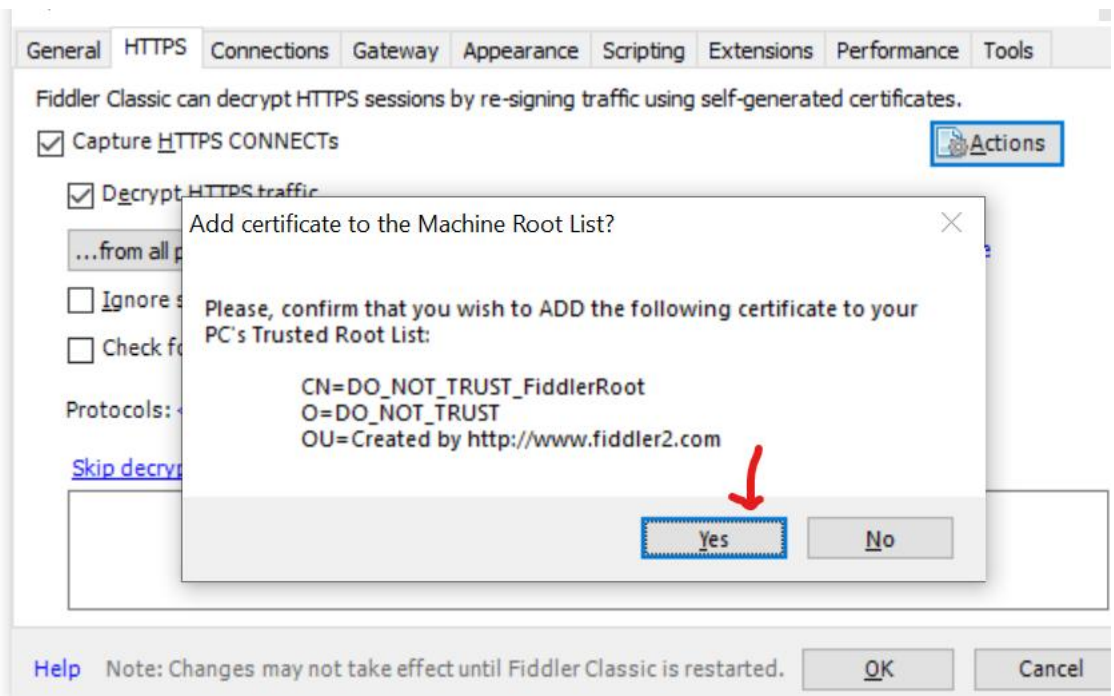


2. Click the **Decrypt HTTPS Traffic** box



3. Install Fiddler certificate (this window will appear when you click "Decrypt HTTPS traffic". **Don't reject it!**



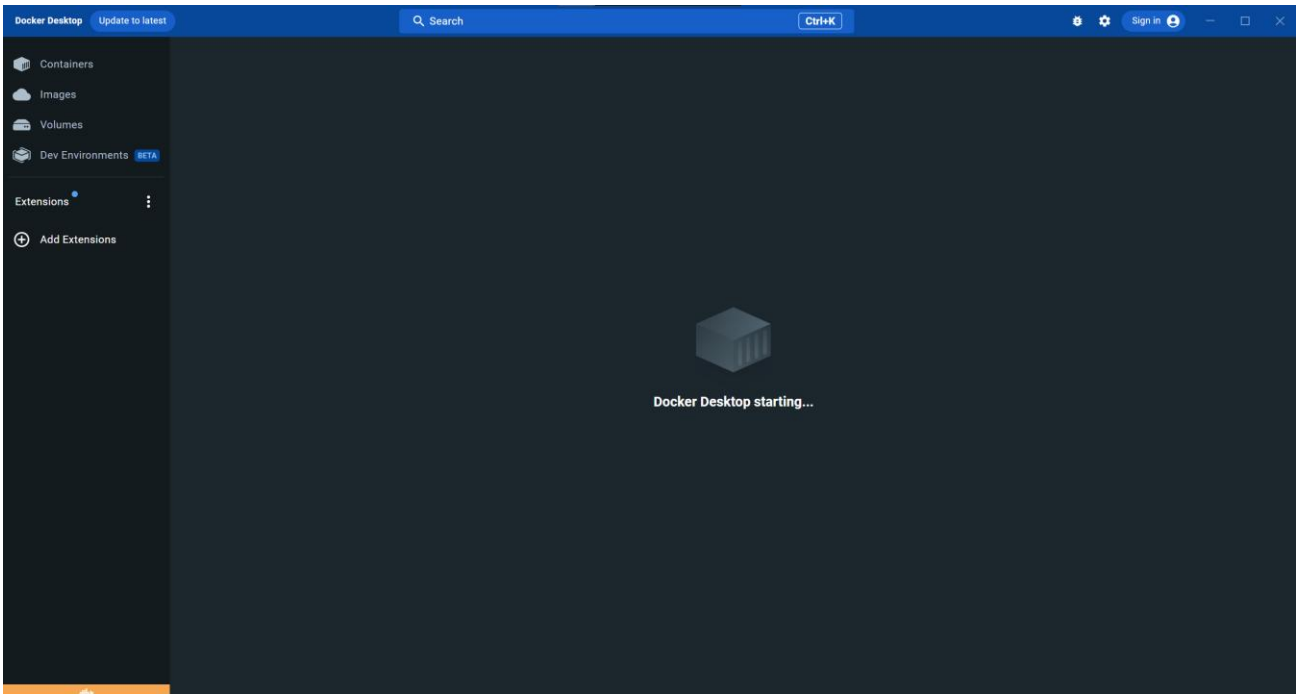


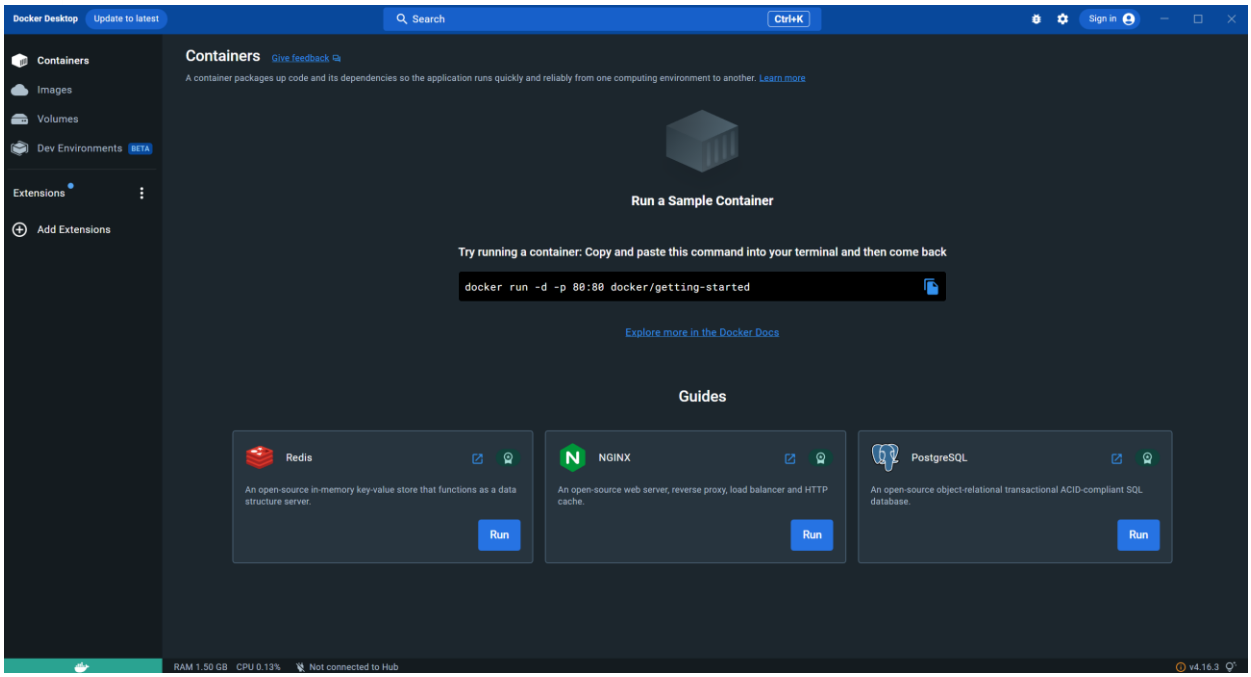
4. Close Fiddler (**CLOSE!** Otherwise, Docker can't install some packages)

Fiddler

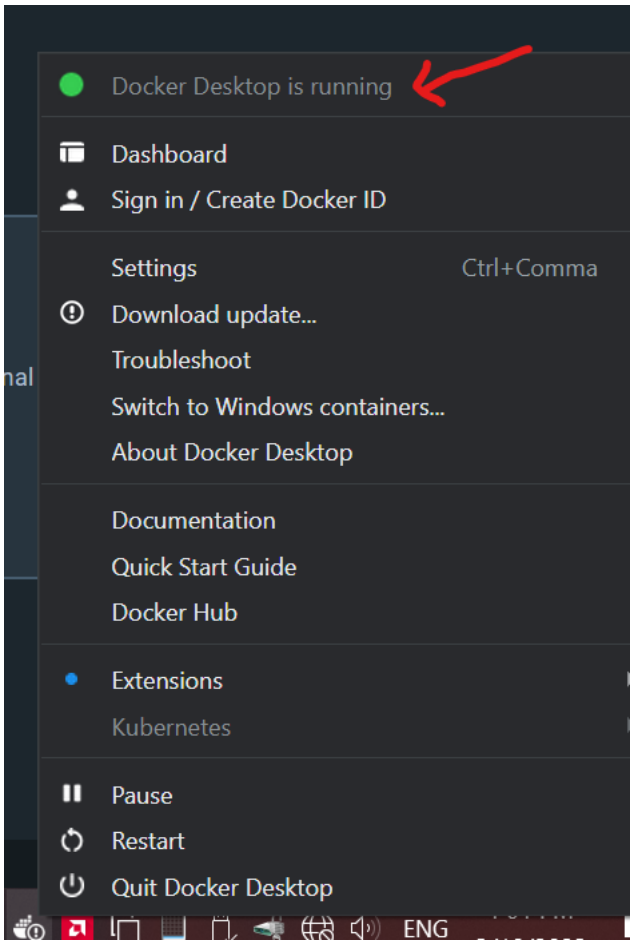
Configure Docker Desktop

- Run Docker Desktop





And make sure it started!



Download docker configs

You can download ALL tested Docker configs (named "Braindead v 13") here (multiple mirrors):
https://github.com/BiosNod/GIO-docker-braindead
https://anonfiles.com/ze89g4p5z7/server_docker_configs_rar
https://upload.biosnod.ru/download/c73da6c3372579fa897168d08de44bde

Download server bins + data for bins

All server bins + data you can download here:
Server data:
https://anonfiles.com/Ta76O2Wfy5/server-1.0-4.0-data_rar
Server bins:
https://anonfiles.com/E6K1O3Wdyb/server-1.0-bins_rar
https://anonfiles.com/N0M2NcW1ya/server-1.1-bins_rar
https://anonfiles.com/t4QaN6W5y2/server-1.2-bins_rar
https://anonfiles.com/D6wbNaW1y3/server-1.3-bins_rar
https://anonfiles.com/Od86NeW9y4/server-1.4-bins_rar
https://anonfiles.com/DbX9V6Wdya/server-1.5-bins_rar
https://anonfiles.com/hffeW6W5y0/server-1.6-bins_rar
https://anonfiles.com/sdw1P3Wby6/server-2.0-bins_rar
https://anonfiles.com/c43fPfW9y2/server-2.1-bins_rar
https://anonfiles.com/26SfOcW0y2/server-2.2-bins_rar
https://anonfiles.com/L6xcU2W2ya/server-2.3-bins_rar
https://anonfiles.com/u469X2Way5/server-2.4-bins_rar
https://anonfiles.com/B3F4Y2W6y2/server-2.5-bins_rar
https://anonfiles.com/Fb59XbW2y2/server-2.6-bins_rar
https://anonfiles.com/B80cY0Wby9/server-2.7-bins_rar
https://anonfiles.com/K2q1afX5y1/server-2.8-bins_rar
https://anonfiles.com/U8n3g7kazb/server_3_0_bins_rar
https://anonfiles.com/fcD4X8Wby6/server-3.1-bins_rar
https://anonfiles.com/F8j4gbX6y1/server-3.1-bins_rar
https://anonfiles.com/sbkbg4Xay3/server-3.2-bins_rar
https://anonfiles.com/R1g8gbX9yf/server-3.3-bins_rar
https://anonfiles.com/G1T3f4Xay0/server-3.4-bins_rar
https://anonfiles.com/L9j8g0X3y1/server-cb1-bins_rar
https://anonfiles.com/H4l6i7X1yc/server-cb2-bins_rar
https://anonfiles.com/C4l5icX1y9/server-cb3-bins_rar
https://anonfiles.com/9fk2h3X7yd/server-ce0.6-bins_rar

https://anonfiles.com/Hak9hdX6ye/server-ce0.8-bins_rar
https://anonfiles.com/lbl2h9X7ye/server-ce0.9-bins_rar
https://anonfiles.com/AfNanfX4y3/server-weird-abc-bins_rar
https://anonfiles.com/k7lcn6Xdy0/server-weird-def-bins_rar
https://anonfiles.com/T0U1i6Xbyf/server-weird-ghi-bins_rar
Server configs (useless, you can skip it):
https://anonfiles.com/Y57cMdWfy3/server-1.0-3.2-configs_rar

Suitable bins + data list:

Please use suitable bins + data - for example: "live" docker and "live" bins and "live" data! It's very important! **Read below how to use the suitable versions in the next chapter!**

Game: 3.2 release

Docker: 3.2_dev

Bins: server-3.2-bins.rar -> hk4e_3.2_beta

Data: server-1.0-4.0-data.rar -> welink_3.2_qa.7z

Game: 3.1 release

Docker: 3.1_dev

Bins: server-3.1-bins.rar -> hk4e_3.1_dev-hk4e_output.1148189

Data: server-1.0-4.0-data.rar -> 3.1_dev-output_10745594-server-data.tgz OR welink_3.1_qa.7z

Game: 3.0 release

Docker: use 2.8.50 beta client instead of 3.0 client

Bins: use 2.8.50 beta client instead of 3.0 client

Data: use 2.8.50 beta client instead of 3.0 client

Game: 2.8.50 beta (NOT 3.0 release because different Proto ids)

Docker: 3.0_dev

Bins: server-3.0-bins.rar -> hk4e_3.0_dev-hk4e_output.1123652

Data: server-1.0-4.0-data.rar -> 3.0_dev-output_9247705-server-data.tgz

Game: 2.8 release

Docker: 2.8_live

Bins: server-2.8-bins.rar -> hk4e_2.8_live-hk4e_output.1133088

Data: server-1.0-4.0-data.rar -> 2.8_live-output_9464149-server-data.tgz

Game: 2.7 release

Docker: 2.7_live

Bins: server-2.7-bins.rar -> hk4e_2.7_live-hk4e_output.1112358

Data: server-1.0-4.0-data.rar -> 2.7_live-output_8227893-server-data.tgz

Game: 2.6 release

Docker: 2.6_live

Bins: server-2.6-bins.rar -> hk4e_2.6_live-hk4e_output.1062793

Data: server-1.0-4.0-data.rar -> 2.6_live-output_6845002-server-data.tgz

Game: 2.5 release

Docker: 2.5_live

Bins: hk4e_2.5_live-hk4e_output.1038836

Data: 2.5_live-output_5999657-server-data.tgz

Game: 2.4 release

Docker: 2.4_live

Bins: hk4e_2.4_live-hk4e_output.1024374

Data: 2.4_live-output_5736476-server-data.tgz

Game: 2.3 release

Docker: 2.3_live

Bins: hk4e_2.3_live-hk4e_output.1006080

Data: 2.3_live-output_5220062-server-data.tgz

Game: 2.2 release

Docker: 2.2_live

Bins: hk4e_2.2_live-hk4e_output.985491

Data: 2.2_live-output_4715326-server-data.tgz

Game: 2.1 release

Docker: 2.1_live

Bins: hk4e_2.1_live-hk4e_output.975292

Data: 2.1_live-output_4398912-server-data.tgz

Game: 2.0 release

Docker: 2.0_live

Bins: hk4e_2.0_live-hk4e_output.964367

Data: 2.0_live-output_4066593-server-data.tgz

Game: 1.6 release

Docker: 1.6_live

Bins: hk4e_1.6_live-hk4e_output.951601

Data: 1.6_live-output_3432262-server-data.tgz

Game: 1.5 release

Docker: 1.5_live

Bins: hk4e_1.5_live-hk4e_output.931242

Data: 1.5_live-output_2957756-server-data.tgz

Game: 1.4 release

Docker: 1.4_live

Bins: hk4e_1.4_live-hk4e_output.922252

Data: 1.4_live-output_2563827-server-data.tgz

Game: 1.3 release

Docker: 1.3_live

Bins: hk4e_1.3_live-hk4e_output.888632

Data: 1.3_live-output_2185906-server-data.tgz

Game: 1.2 release

Docker: 1.2_live

Bins: hk4e_1.2_live-hk4e_output.868686

Data: 1.2_live-output_1816310-server-data.tgz

Game: 1.1 release

Docker: 1.1_live

Bins: hk4e_1.1_live-hk4e_output.825765

Data: 1.1_live-output_1624669-server-data.tgz

Game: 1.0 release

Docker: 1.0_live

Bins: hk4e_1.0_live-hk4e_output.804435

Data: 1.0_live-output_1358691-server-data.tgz

???: 1.0_live-output_1358691-operation.zip

Game: CB3-0.9.0

Docker: cb3update_live

Bins: server-cb3-bins.rar -> hk4e_cb3_update_live-hk4e_output.732427

Data: cb3_update_live-output_873274-server-data.tgz

Game: CB2-0.7.0

Docker: cb2plus_live

Bins: server-cb2-bins.rar -> hk4e_cb2plus_live

Data: cb2plus_live-output_521763-server-data.tgz

Game: CB1-1.0.0

Docker: cb01_live

Bins: server-cb1-bins.rar -> hk4e_cb01_live

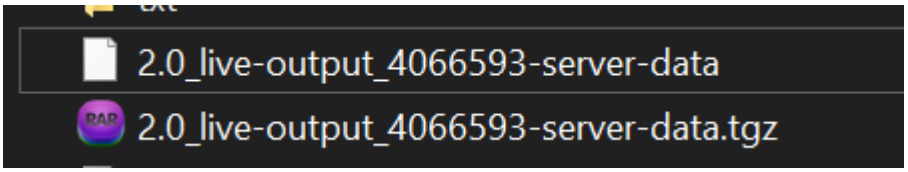
Data: cb1_live-output_141484-server-data.tgz OR cb1_live_os-output_141484-server-data.tgz

How to use suitable server data

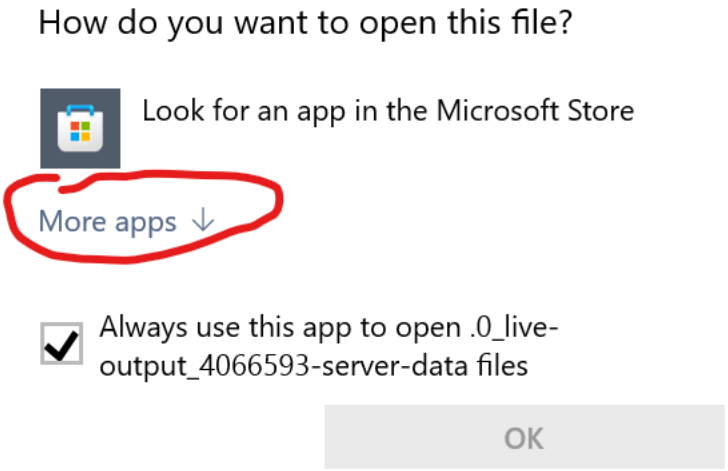
- Unpack any docker version from the **server-docker-configs-braindead.rar** archive to some folder.

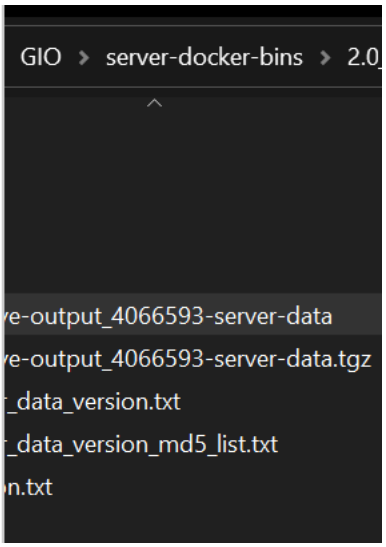
- Unpack data archive from the **server-1.0-4.0-data.rar** archive to **ANY** place (not important). For example, I want to use server for 2.0 game version.

In this case I should unpack **2.0_live-output_4066593-server-data.tgz** archive from the **server-1.0-4.0-data.rar** archive to **ANY** place. And then I should see what is inside. I see **2.0_live-output_4066593-server-data** file without any extension inside **2.0_live-output_4066593-server-data.tgz** archive. Just unpack it again nearby:

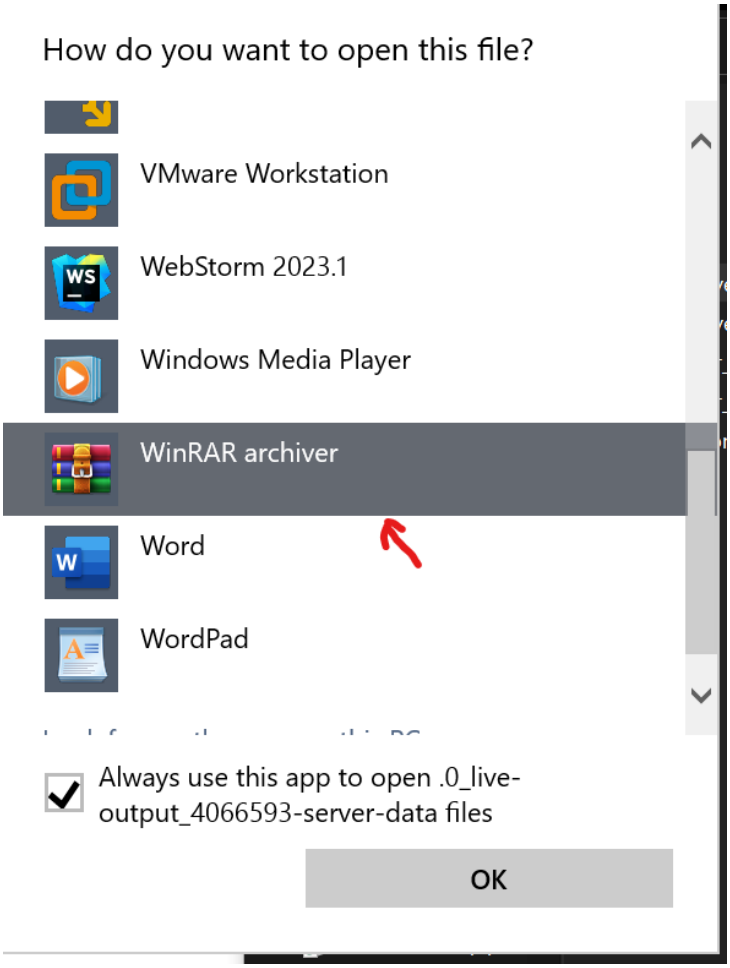


And open **2.0_live-output_4066593-server-data** without extension as archive again:

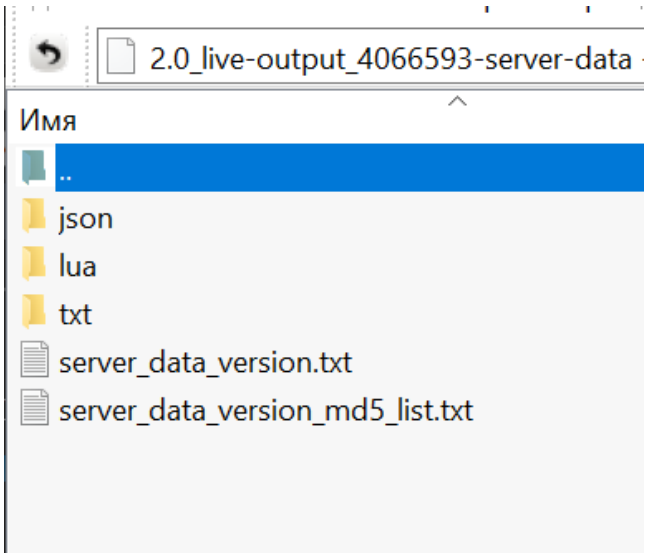




Select your favorite archive tool:



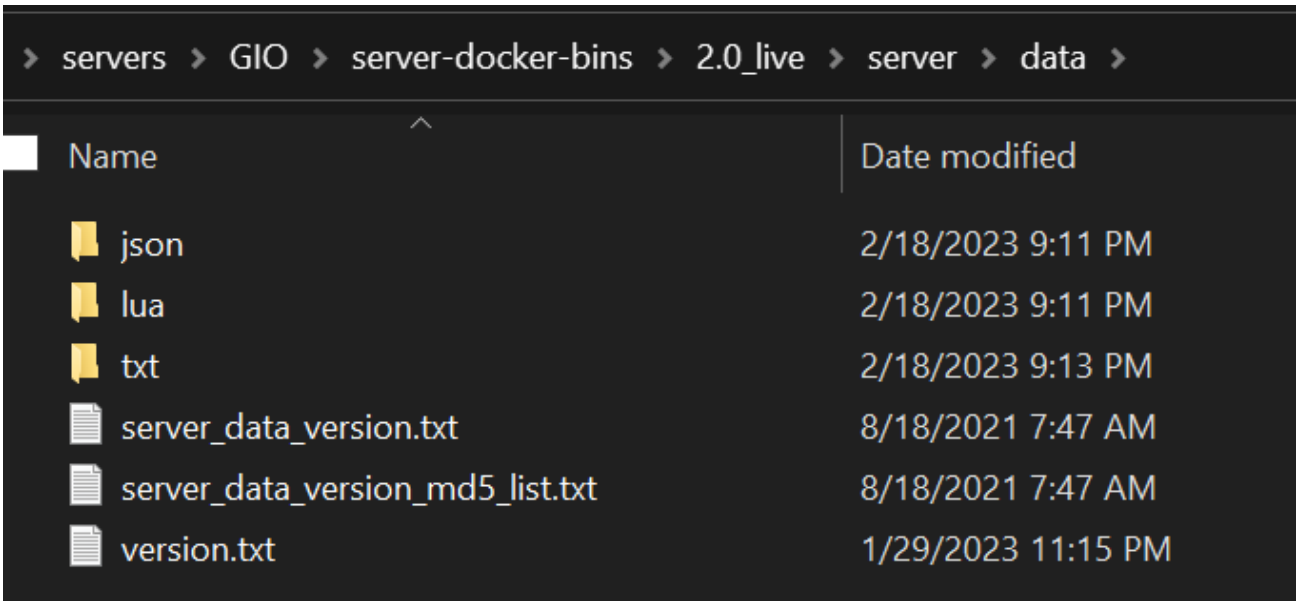
And you will see a normal data structure:



NOTE! In some data archives json + lua + txt already in normal state without second non-extension archive! Just unpack it "as is" in this case.

Next step is unpacking all **json + lua + txt** to your previously unpacked docker config folder from the **server-docker-configs-braindead-v12.rar**

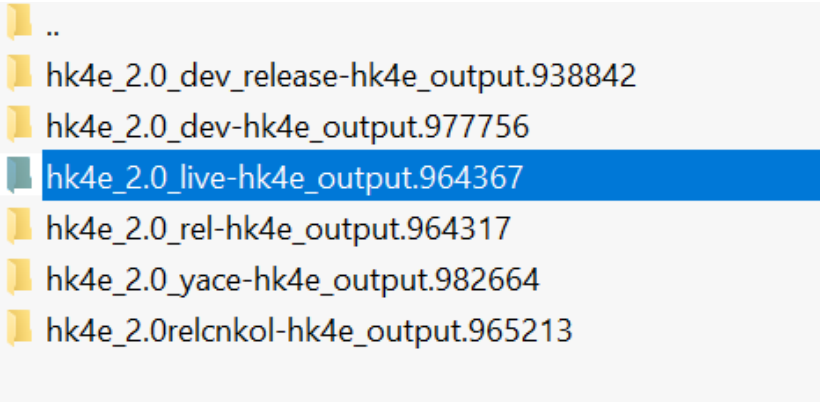
For example – **2.0_live**. Go to **2.0_live/server/data** folder and place it here:



All done with data

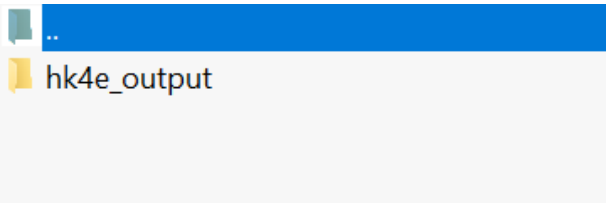
How to use suitable server bins

If I want to run game version 2.0 – I should go to **server-2.0-bins.rar** archive and I will see inside**:**

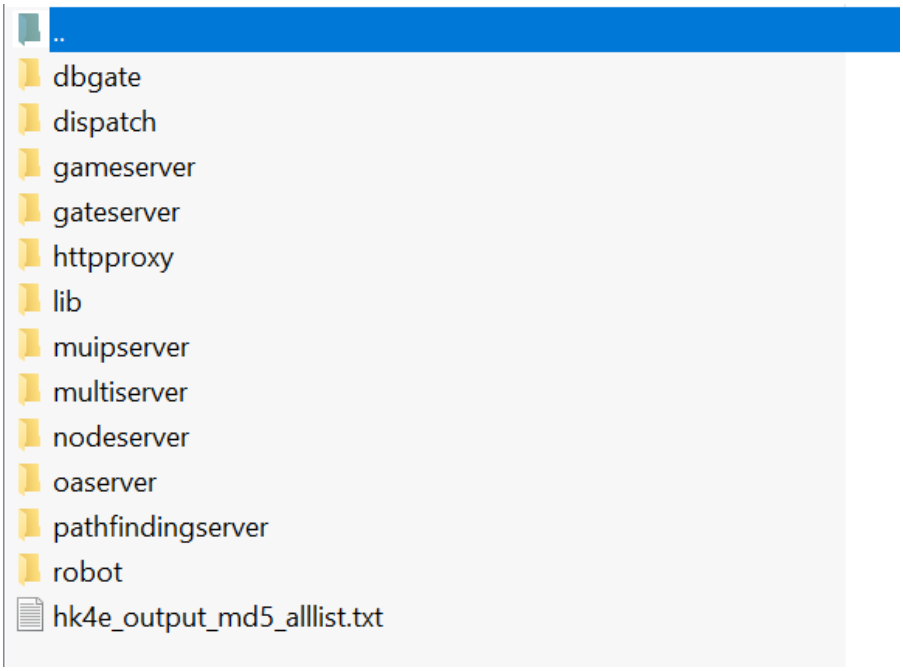


According to **Suitable bins + data list** chapter I should choose this version (see screen above).

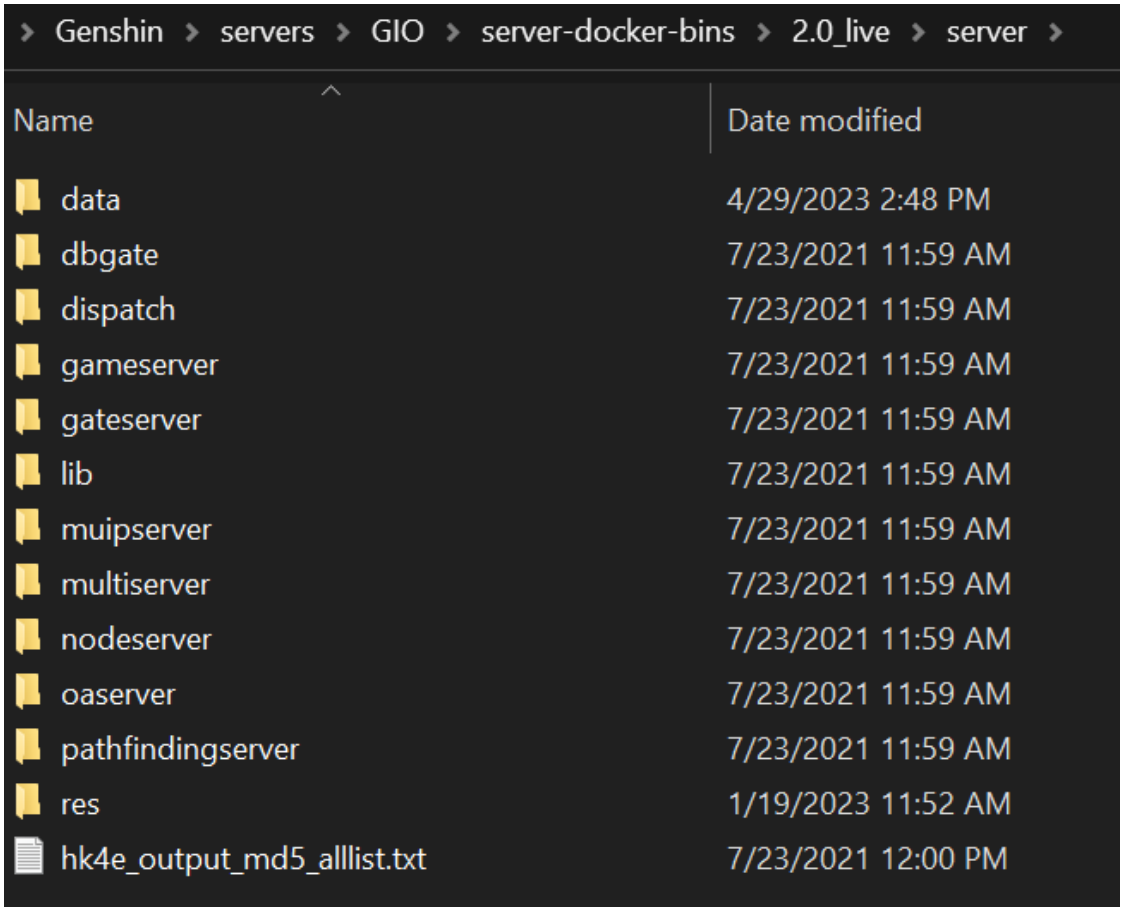
Go deeper:



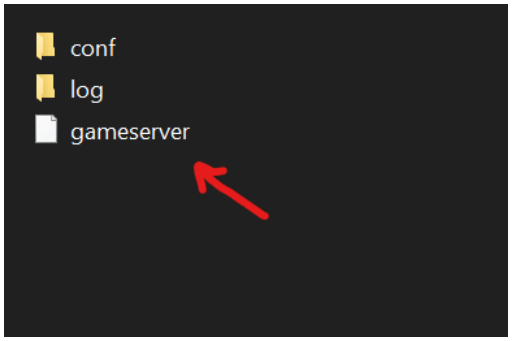
And deeper:



And unpack all of this (except **robot** and **httpproxy**) to **2.0_live/server** folder:



After unpacking, please check **2.0_live/sever/gameserver** folder, it should be like this:



All done with server bins

NavMesh and SVO (Sparse Voxel Octrees)

Sorry but we haven't **NavMesh** and **SVO** for all versions. That's why we can't use **pathfindingserver** and **tothemoonserver**. Because of that some **AI** in game will be more stupid than original official servers. **Hotaru** said: it tracks obstacles, changes in height on map, etc. without them **AI** (be it monsters or **NPCs**) are a little dumber for example during Jean story quest she goes straight through the mond gate and gets stuck on a cart because **AI** isnt aware it's an obstacle there she then teleports away it doesn't stop you from playing it just dumb **NPC** behavior.

So. You can play fine without **NavMesh** and **SVO** and without **pathfindingserver** and **tothemoonserver.**

Also see (if you are very skilled):

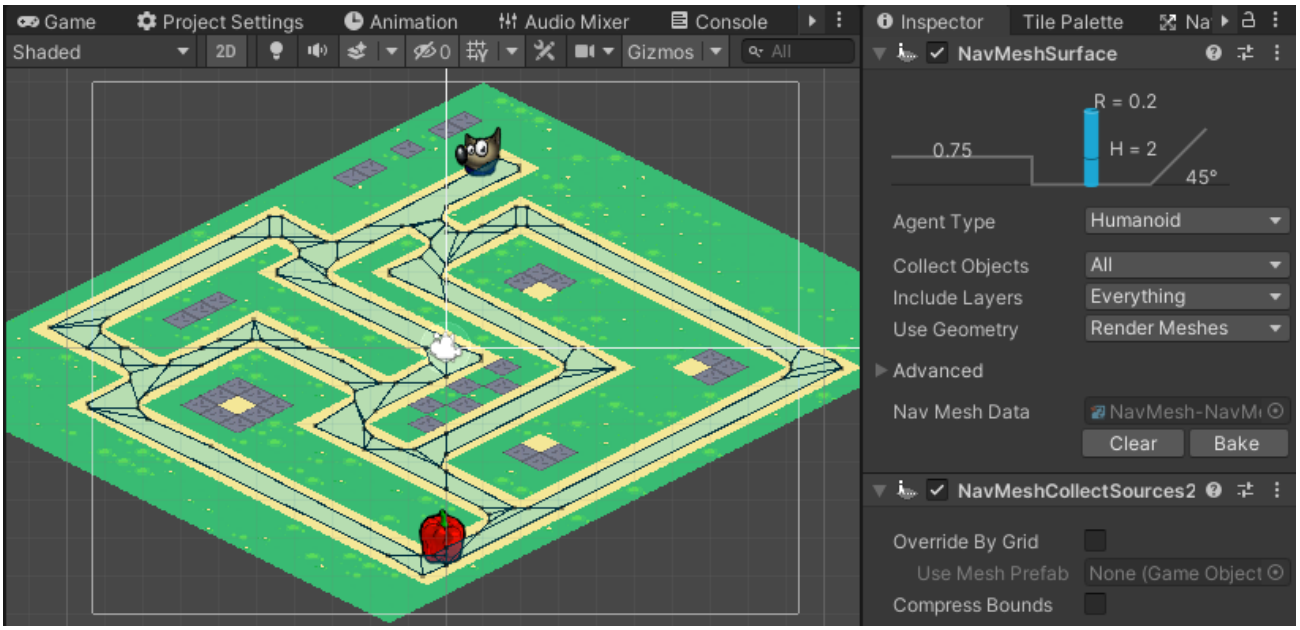
https://github.com/lazytiger/unityai

https://docs.unity3d.com/Manual/nav-BuildingNavMesh.html

https://github.com/BudgetToaster/unity-sparse-voxel-octrees

https://learn.unity.com/tutorial/unity-navmesh#

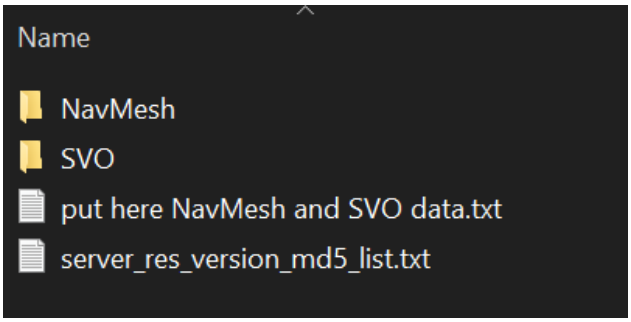
https://library.vuforia.com/develop-area-targets/using-unitys-navmesh-navigation-area-targets



How to setup NavMesh

Good news. We have only one NavMesh – for 2.6 version (but you can use it in other versions I guess). Download it here - https://anonfiles.com/p46ci7p9z0/2_6_apple_output_6145677_server_res_tgz

Put it to **server/res** folder:



Delete **profiles: - donotstart** for **pathfindingserver** in **docker-compose.yml.tmpl** (and in **docker-compose.yml**):

```
pathfindingserver:
  image: debian:11
  restart: on-failure
  networks:
    default:
      ipv4_address: 172.10.3.8
  expose:
    - 21101/udp
  profiles:
    - donotstart
  depends_on:
    - mysql
    - redis
  volumes:
    - ./server:/opt/hk4e
  working_dir: /opt/hk4e/pathfindingserver
  environment:
    <<: *mihoyo-variables
  command: ./pathfindingserver -i 1003.8.8.1
  stop_grace_period: 180s
  deploy:
    resources:
      limits:
        cpus: '%CPU_USAGE_LIMIT%'
      reservations:
        cpus: '%CPU_USAGE_LIMIT%'
```

Start pathfindigserver container and wait for some hours until fully loaded into the server in RAM (it also increases the RAM requirements **+7GB**):

docker compose up -d -f pathfindingserver

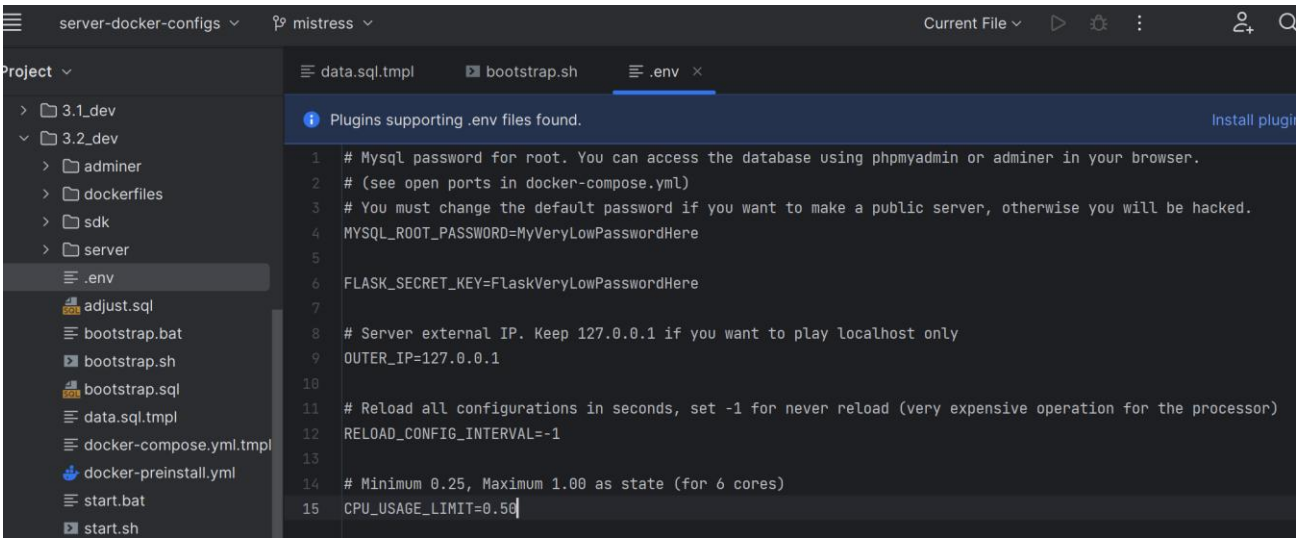
See logs:

docker compose logs -f pathfindingserver

As you see – mobs choose the way around the mountain:

Start server bins in Docker Desktop

- Check **.env** file to adjust your IP/CPU usage/etc.

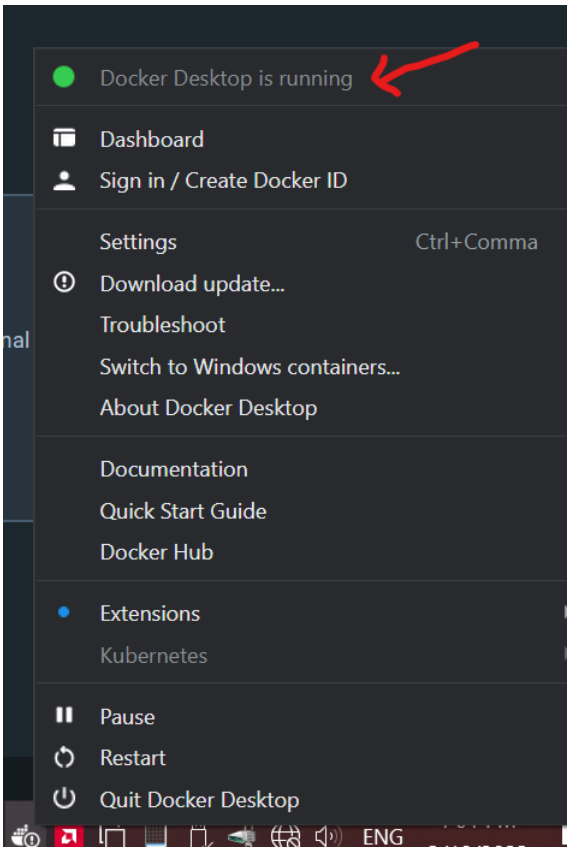


If you ARE NOT programmer – don’t change anything please.

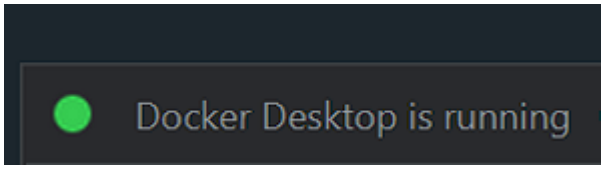
- **Turn off Fiddler. Important!** Otherwise, Docker can't download libs to run servers.

Start server bins + data in Docker Desktop

- Start Docker Desktop and make sure it running



- If you don't see this:

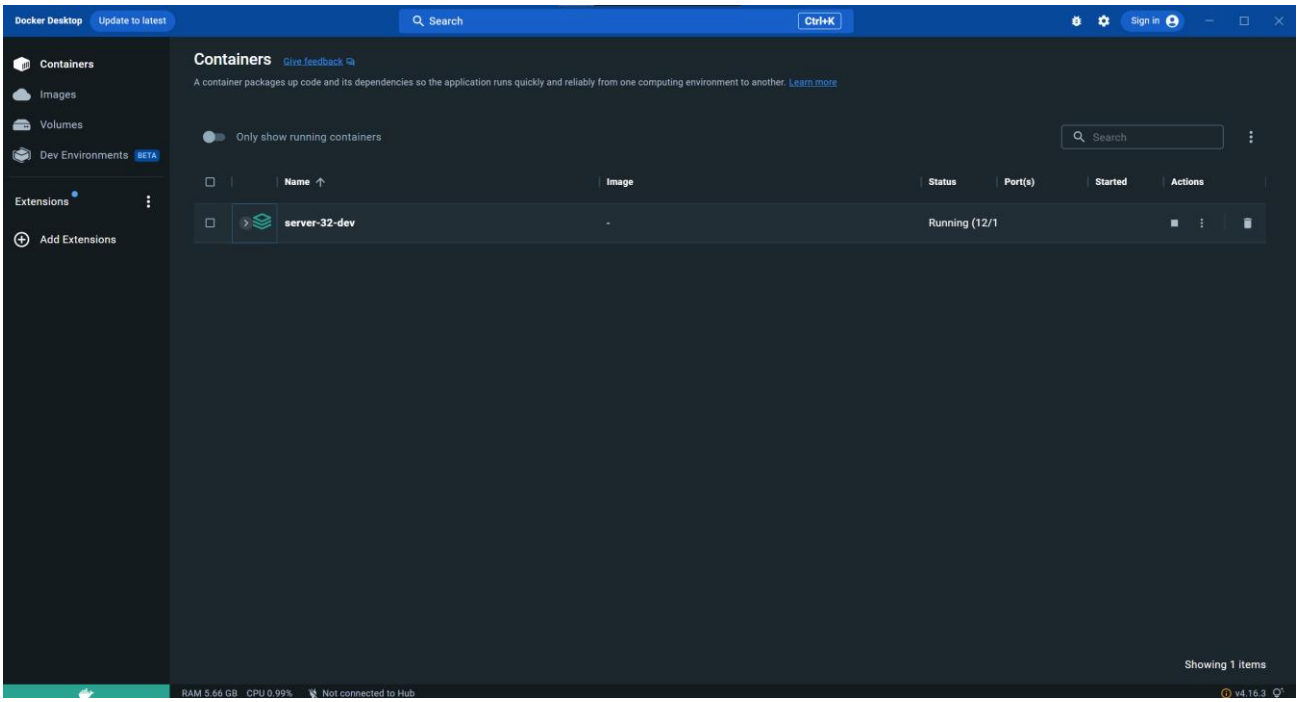


  Don't do the next steps. Better use Google/ChatGPT to solve your problem with Docker Desktop.

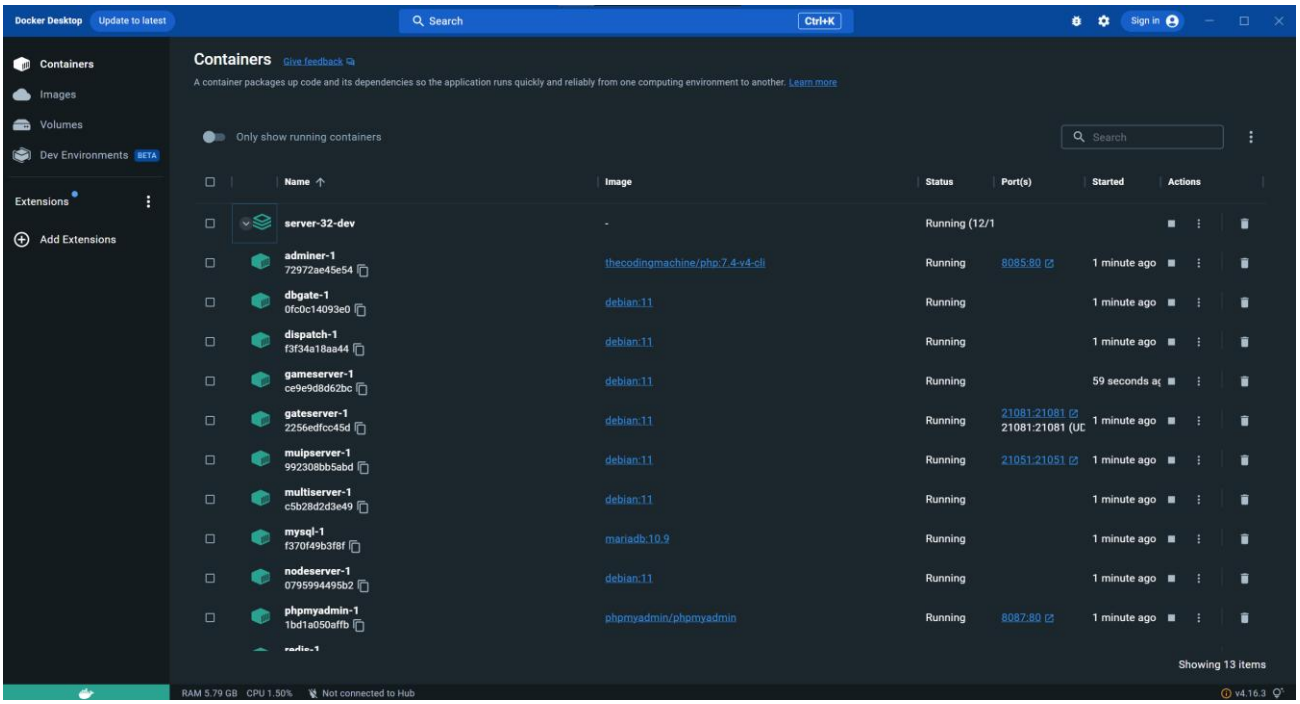
- Run bootstrap script: Windows - run **bootstrap.bat,** Linux - run **bootstrap.sh** (Also, for Linux use **chmod +x bootstrap.sh** to allow run, also better do: **sudo chmod -R +x server_folder**)

Run **bootstrap** only once! Run only **stop/start .bat or .sh** for the next time instead of **bootstrap**. Please don't run "**docker compose up -d**" or "**docker-compose up -d**" manually cuz each Dockers configs have a very uneasy startup procedure, use **.bat** and **.sh** scripts only.

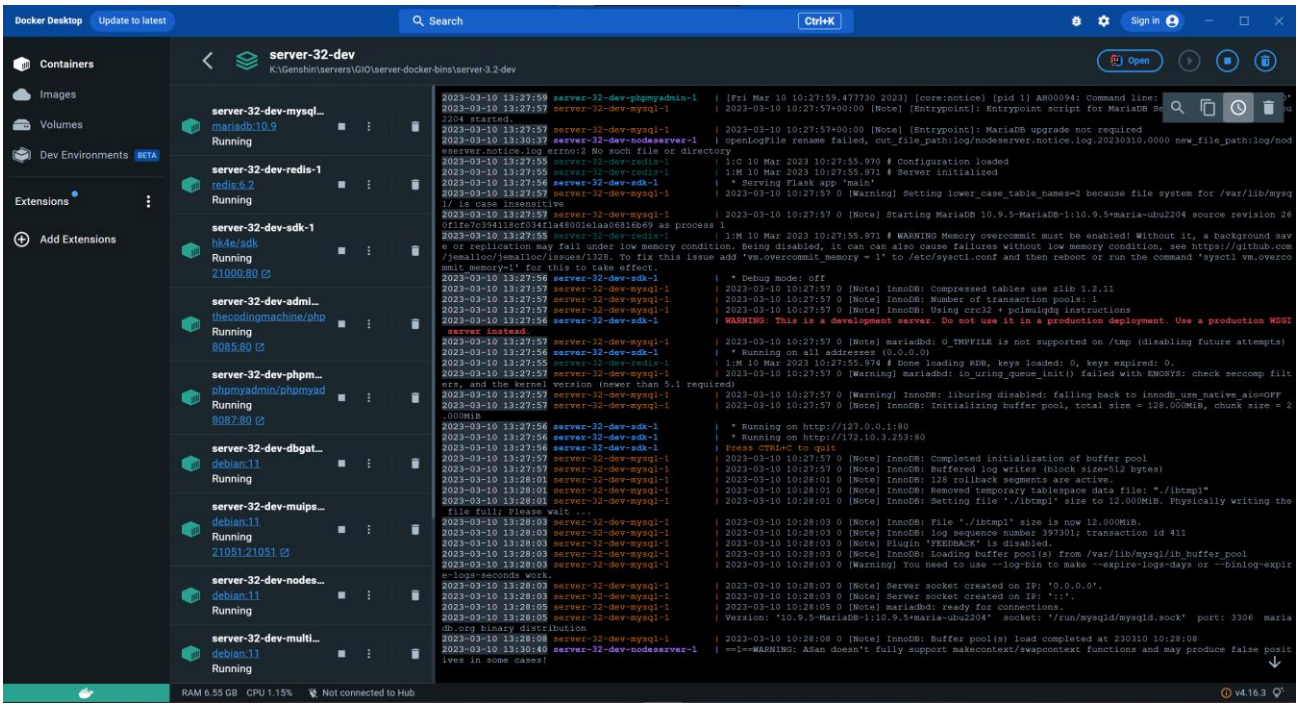
See Docker dashboard (or use "**docker compose ps**" command in Linux terminal)



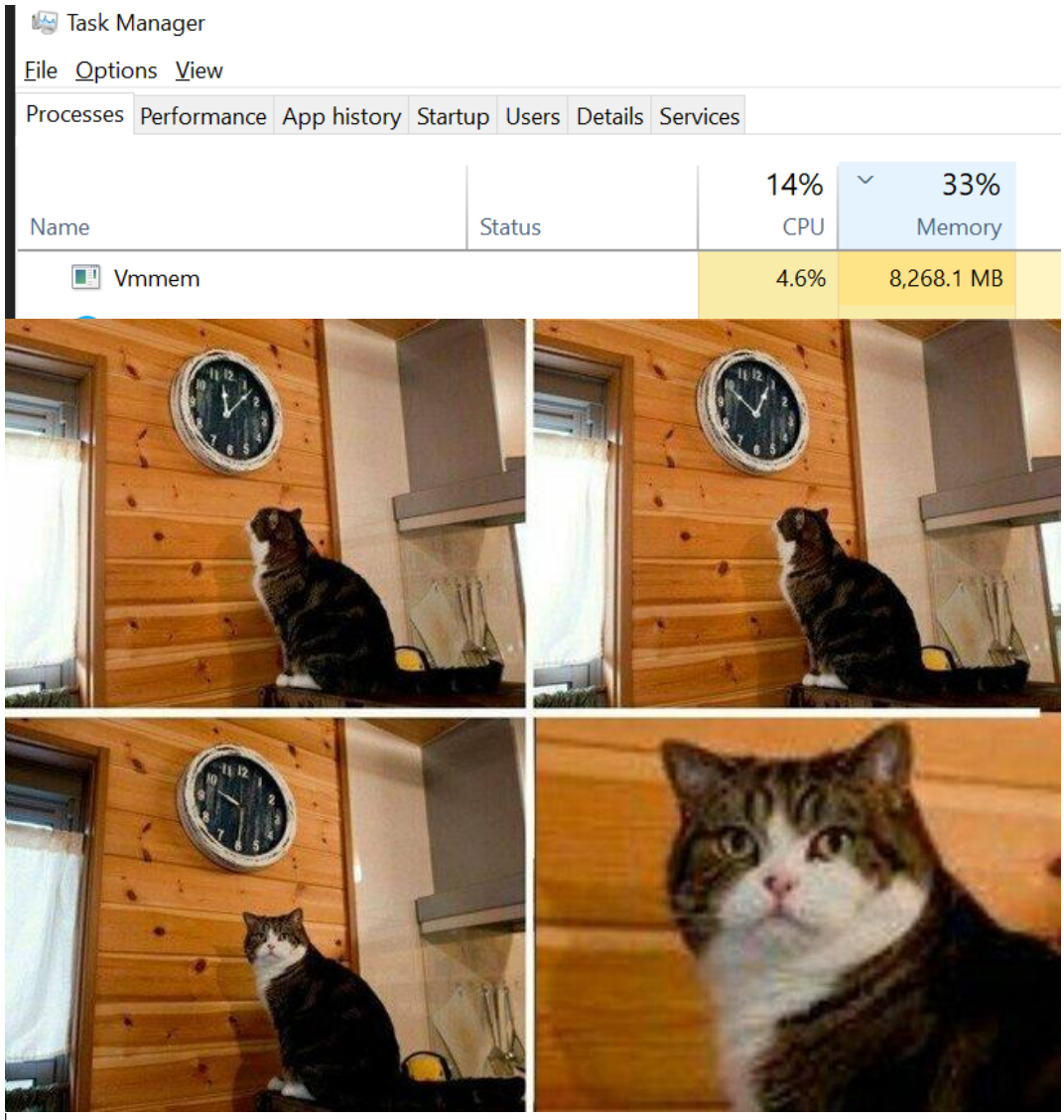
- Click arrow down button to open the containers list:

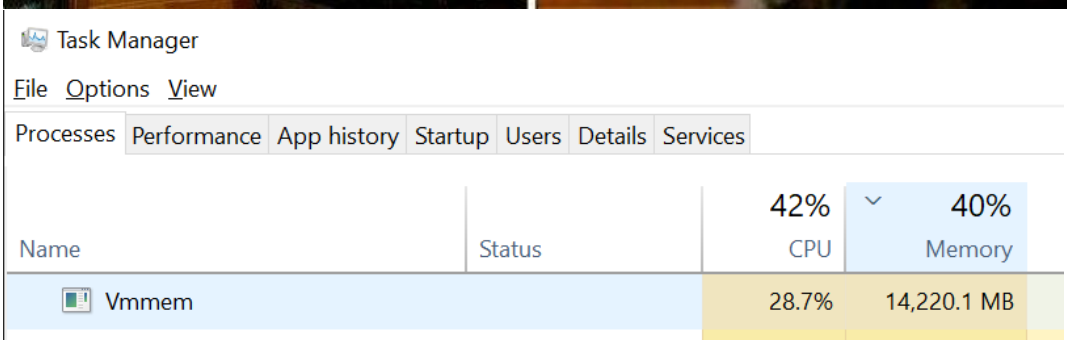


Also, you can click on the main compose configs title to see main logs from the all containers:



- Wait for 10-15 minutes when **Vmmem (Virtual memory)** process will be 10GB+ RAM (in some cases you should wait even for 30 minutes :D)





Fix WSL Docker memory (aka white game screen after gates)

Docker uses 50% of host memory by default. You should create a custom wsl configuration and allocate enough memory to run all services.

Go to **C:\Users\%USERNAME%\.**

Create file **C:\Users\%USERNAME%\.wslconfig** and fill it with content below (colored).

```
[wsl2]
memory=18GB
```

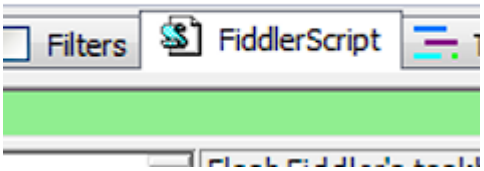
After that, restart Docker Desktop, wait and restart your game client also.

Start fiddler with special fiddler script

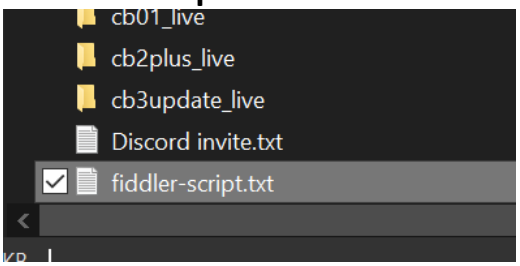
- Run Fiddler (**ONLY** after your docker containers already running and have a green color in Docker Desktop)

- Read **README.md** (open as text) in some docker folder for some game version and read it

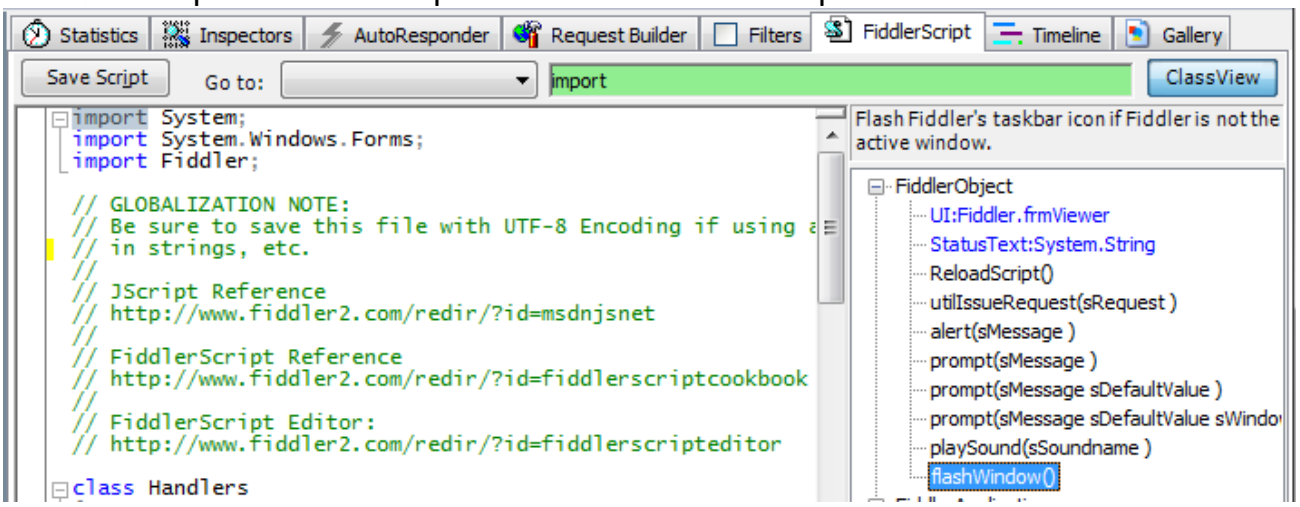
- Go to "**Fiddler script**" tab in Fiddler



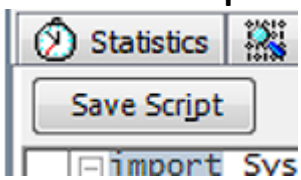
- Copy script from the **fiddler-script.txt** from the **docker-braindead.rar** archive



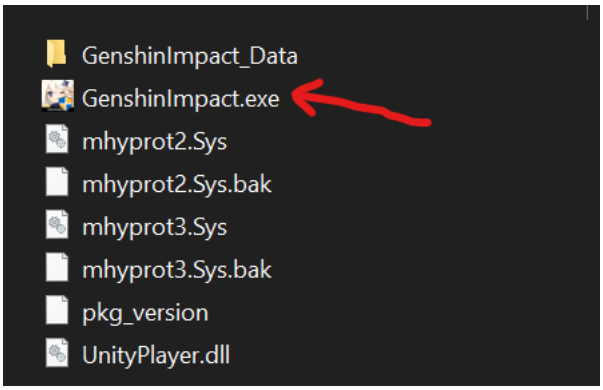
- Paste and replace entire script in the Fiddler to the copied:



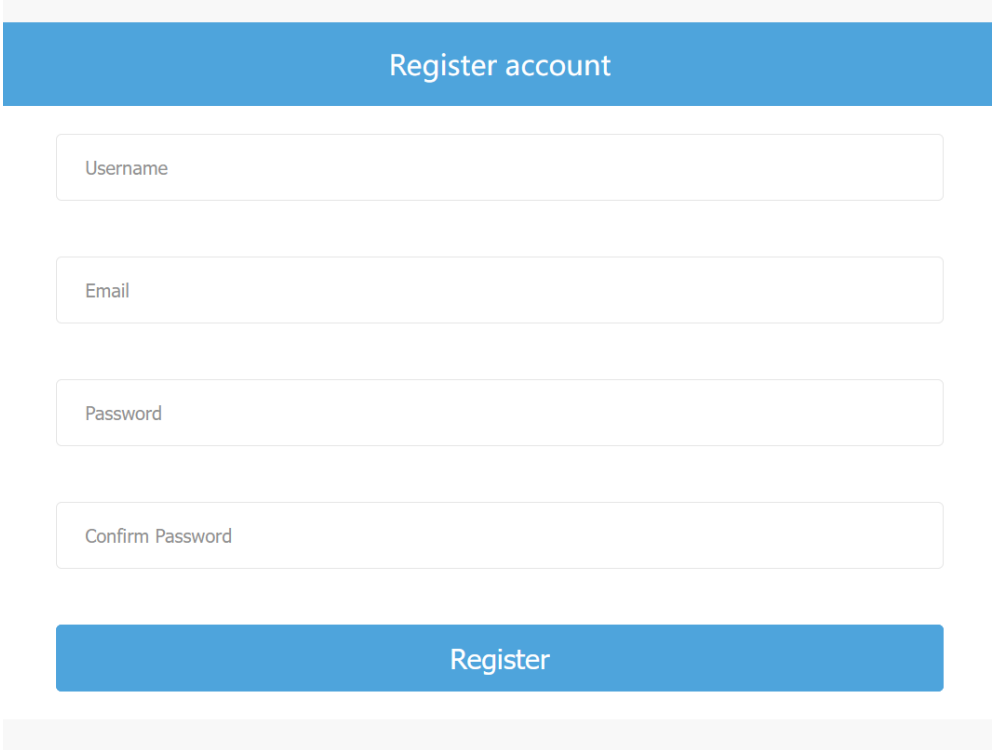
- click "**Save Script**" button (on the top):



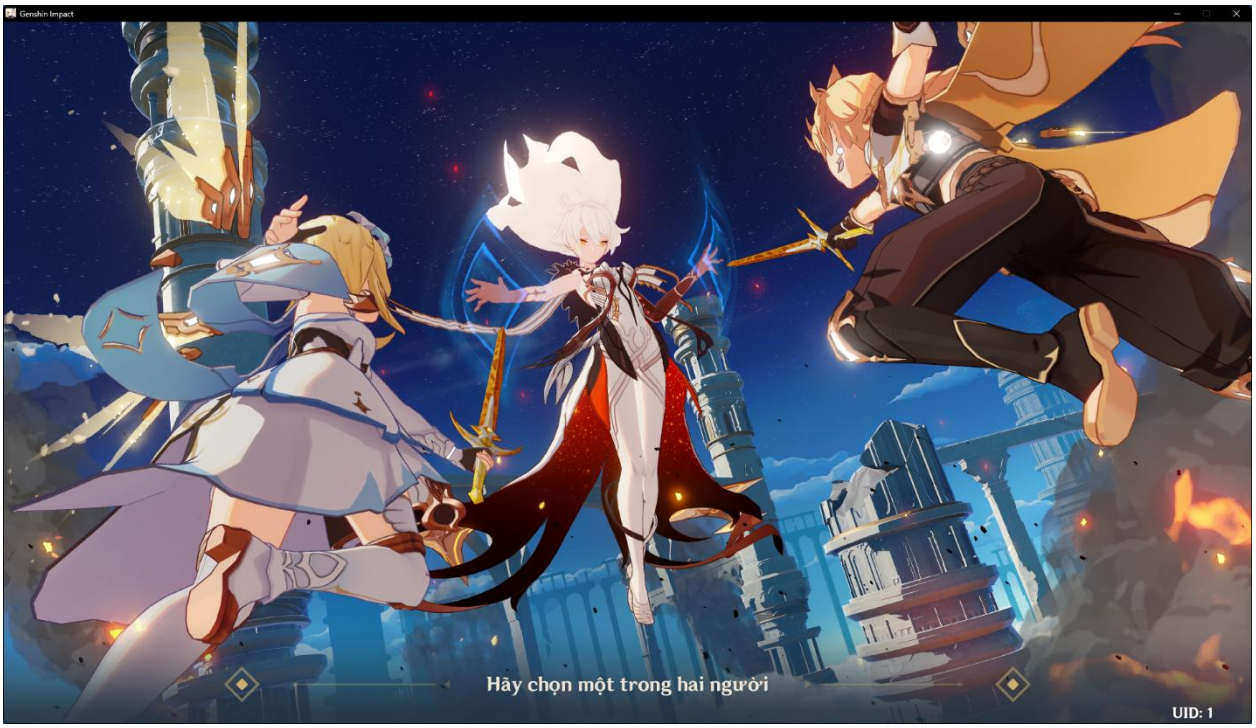
- Run your favorite game client (I hope you already patched it)



- Register account in game by click on "**Register**" button (For **CB1/CB2/CB3** if no reg page in game - just go to the http://127.0.0.1:21000/account/register in your browser and register account!)



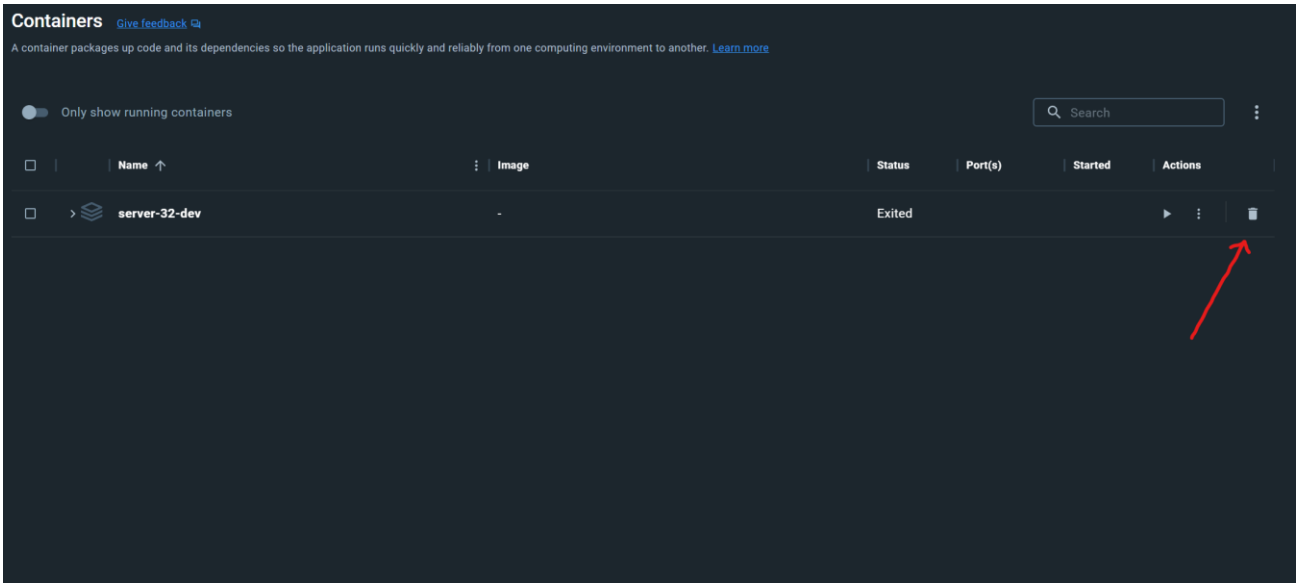
- Play game



- Use **stop.bat/sh** script to stop servers instead of **Docker desktop stop button**.

```
[+] Running 0/12
 - Container server-32-dev-gameserver-1   Killing
 - Container server-32-dev-multiserver-1  Killing
 - Container server-32-dev-mysql-1        Killing
 - Container server-32-dev-nodeserver-1   Killing
 - Container server-32-dev-sdk-1          Killing
 - Container server-32-dev-redis-1        Killing
 - Container server-32-dev-adminer-1      Killing
 - Container server-32-dev-dispatch-1     Killing
 - Container server-32-dev-dbgate-1       Killing
 - Container server-32-dev-muipserver-1   Killing
 - Container server-32-dev-gateserver-1   Killing
 - Container server-32-dev-phpmyadmin-1   Killing
```

- If you want to add and play another version - use **stop.bat/stop.sh** and **Delete** all docker containers from the Docker dashboard first (don't worry, your game progress inside 2 folders - **mysql** and **redis (don't delete it),** just run **start.bat/start.sh** again if you already have this folders even if you already deleted server in Docker dashboard, all be fine).



Useful info about version.txt in CB (Closed Beta) versions

CB1 server bins support only numeric value in data/version.txt like **1** instead of **{}** and will fail with **{}** and **CB2+** support **{}** instead of numeric and will fail with **1** instead of **{}**, please check your **data/version.txt**

Database access (phpMyAdmin)

You can access your database (stored in **mysql** folder) using **phpmyadmin**:

http://127.0.0.1:8087/

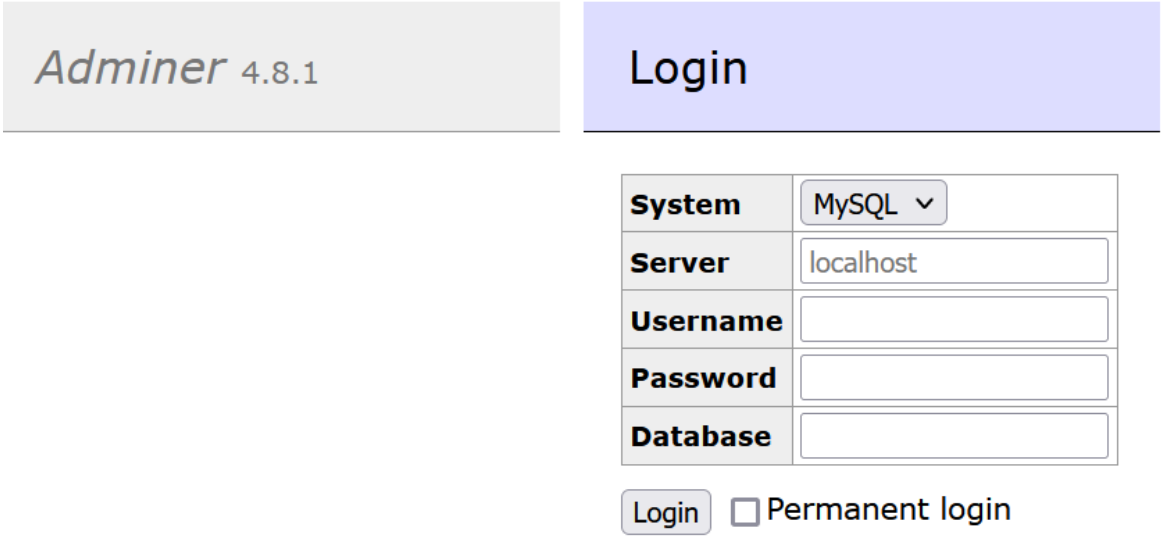
Use **root** as username and password from **.env** file

And go to **HK4E**

Database access (adminer)

Or using **adminer (same as phpMyAdmin)**:

http://127.0.0.1:8085/



Check available versions in Database

You have 2 different ways:

1. In case if you haven't started yet **bootstrap.bat/bootstrap.sh** go to file **data.sql.tmpl (**also in file **data.sql** if exists**)** and add new versions in **t_bind_config**, for example (green - added):

```
INSERT INTO `t_bind_config` (`id`, `client_version`, `region_name`, `channel_i
(1, 'OSRELAndroid0.9.9',  'dev_docker',  '1'),
(2, 'OSRELWin0.9.9',    'dev_docker',   '1'),
(3, 'OSRELiOS0.9.9',  'dev_docker',  '1'),
(4, 'CNRELAndroid0.9.9',  'dev_docker',  '1'),
(5, 'CNRELWin0.9.9',  'dev_docker',  '1'),
(6, 'CNRELiOS0.9.9',  'dev_docker',  '1'),
(7, 'OSRELAndroid0.9.3',  'dev_docker',  '1'),
(8, 'OSRELWin0.9.3',    'dev_docker',   '1'),
(9, 'OSRELiOS0.9.3',  'dev_docker',  '1'),
(10, 'CNRELAndroid0.9.3',  'dev_docker',  '1'),
(11, 'CNRELWin0.9.3',  'dev_docker',  '1'),
(12, 'CNRELiOS0.9.3',  'dev_docker',  '1'),
(13, 'CNINWin0.9.3',  'dev_docker',  '1');
```

Make sure that symbol “;” only one after all rows and all rows must have “,” separator.

```
.3',  'dev_docker',  '1'),
.3',  'dev_docker',  '1'),
3',  'dev_docker',  '1');
```

Also, you should add new versions in **t_client_config:**

```
INSERT INTO `t_client_config` (`id`, `version`, `name`, `stop_server_config_str
(1, 'OSRELAndroid0.9.9',    '', '', '', '{\"sdkenv\":\"2\",\"checkdevice\":fals
(2, 'OSRELWin0.9.9',  '', '', '', '{\"sdkenv\":\"2\",\"checkdevice\":false,\"sl
(3, 'OSRELiOS0.9.9',  '', '', '', '{\"sdkenv\":\"2\",\"checkdevice\":false,\"s
(4, 'CNRELAndroid0.9.9',  '', '', '', '{\"sdkenv\":\"0\",\"checkdevice\":false
(5, 'CNRELWin0.9.9',  '', '', '', '{\"sdkenv\":\"0\",\"checkdevice\":false,\"sl
(6, 'CNRELiOS0.9.9',  '', '', '', '{\"sdkenv\":\"0\",\"checkdevice\":false,\"sl
(7, 'OSRELAndroid0.9.3',    '', '', '', '{\"sdkenv\":\"2\",\"checkdevice\":fals
(8, 'OSRELWin0.9.3',  '', '', '', '{\"sdkenv\":\"2\",\"checkdevice\":false,\"sl
(9, 'OSRELiOS0.9.3',  '', '', '', '{\"sdkenv\":\"2\",\"checkdevice\":false,\"sl
(10, 'CNRELAndroid0.9.3',  '', '', '', '{\"sdkenv\":\"0\",\"checkdevice\":falsc
(11, 'CNRELWin0.9.3',  '', '', '', '{\"sdkenv\":\"0\",\"checkdevice\":false,\"s
(12, 'CNRELiOS0.9.3',  '', '', '', '{\"sdkenv\":\"0\",\"checkdevice\":false,\"s
(13, 'CNINWin0.9.3',  '', '', '', '{\"sdkenv\":\"2\",\"checkdevice\":false,\"sl
```

Make sure that version ids same in both tables

```
(5, 'CNRELWin0.9.9',  'dev_docker',  '1'),
(6, 'CNRELiOS0.9.9',  'dev_docker',  '1'),
(7, 'OSRELAndroid0.9.3',  'dev_docker',  '1'),
(8, 'OSRELWin0.9.3',    'dev_docker',   '1'),
(9, 'OSRELiOS0.9.3',  'dev_docker',  '1'),
(10, 'CNRELAndroid0.9.3',  'dev_docker',  '1'),
(11, 'CNRELWin0.9.3',  'dev_docker',  '1'),
(12, 'CNRELiOS0.9.3',  'dev_docker',  '1'),
(13, 'CNINWin0.9.3',  'dev_docker',  '1');


CREATE TABLE `t_account_type_config` (
  `id` int(10) unsigned NOT NULL AUTO_INCREMENT,

(3, 'OSRELiOS0.9.9',  '', '', '', '{\"sdkenv\":\"2\",\"checkdevice\
(4, 'CNRELAndroid0.9.9',  '', '', '', '{\"sdkenv\":\"0\",\"checkdev
(5, 'CNRELWin0.9.9',  '', '', '', '{\"sdkenv\":\"0\",\"checkdevice\
(6, 'CNRELiOS0.9.9',  '', '', '', '{\"sdkenv\":\"0\",\"checkdevice\
(7, 'OSRELAndroid0.9.3',    '', '', '', '{\"sdkenv\":\"2\",\"checkd
(8, 'OSRELWin0.9.3',  '', '', '', '{\"sdkenv\":\"2\",\"checkdevice\
(9, 'OSRELiOS0.9.3',  '', '', '', '{\"sdkenv\":\"2\",\"checkdevice\
(10, 'CNRELAndroid0.9.3',  '', '', '', '{\"sdkenv\":\"0\",\"checkde
```

2. In case if you already started **bootstrap.bat/bootstrap.sh** and already played the game – you should edit already filled Database (cuz this **.sql** file has already been used to fill the database). Use **phpMyAdmin** or **adminer** to access DB and search tables **t_bind_config** and **t_client_config** in hk4e databases. Then add new rows with the new versions or just change existing rows:

3. Also in case if you already started **bootstrap.bat/bootstrap.sh** and **NEVER** played at game, you can delete **.bootstrap.lock** file and start **bootstrap.bat/bootstrap.sh** – this action will **ERASE** all your current database and refill data using .**sql** file (your game progress will be reset)

GM Cheat commands and MUIP tool

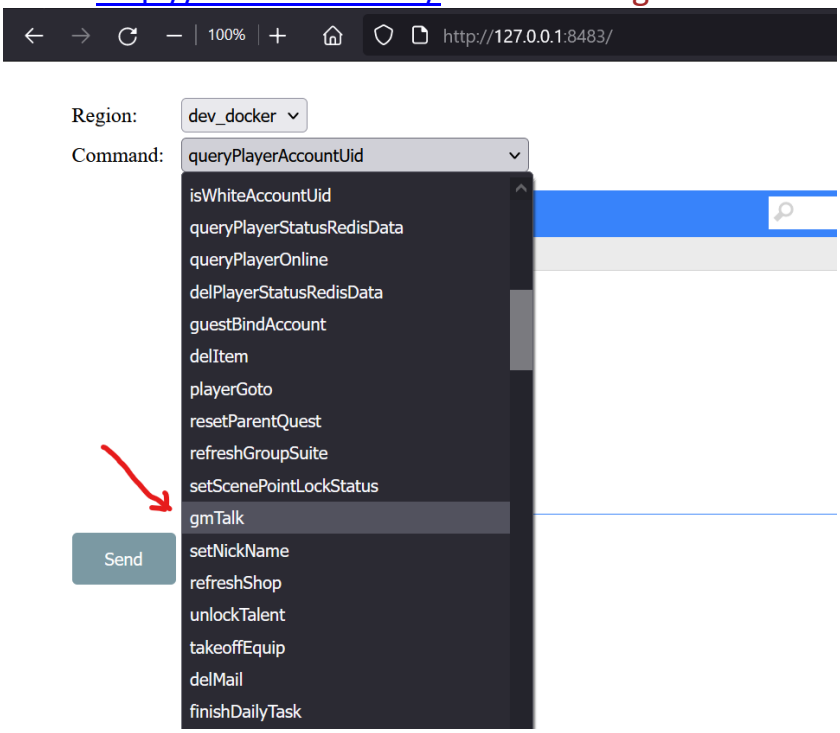
Don't forget about open-source MUIP tool to send GM cheat commands (by **Hotaru**):
https://upload.biosnod.ru/download/f2e1c9a805821d44d2cb717b3a6091c5
https://anonfiles.com/DcT3v4odzc/MuipTool_master_1_rar

muipserver - special server for GM commands.
MUIP tool just send commands to him. Before you should install python 3.10 to use it and run start.bat/start.sh after. Or use Docker file if you want.

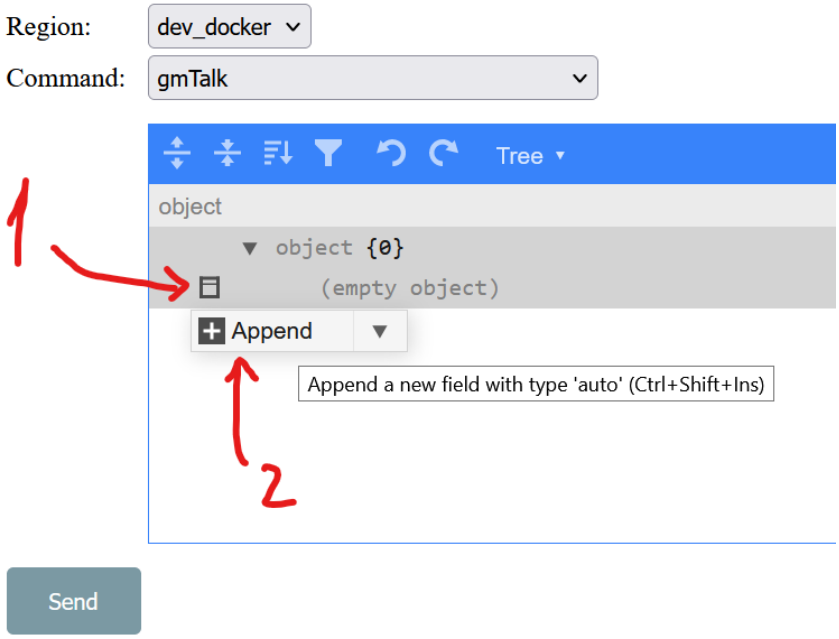
After run start.bat:

```
>> Starting Flask server instance...
 * Serving Flask app 'main'
 * Debug mode: off
WARNING: This is a development server. Do not use it in a
 * Running on http://127.0.0.1:8483
Press CTRL+C to quit
```

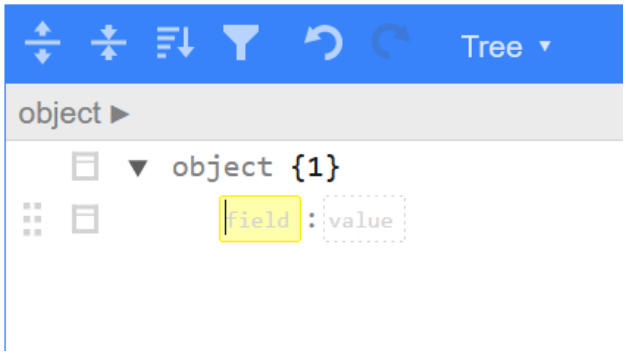
Go to http://127.0.0.1:8483/ and choose gmTalk:



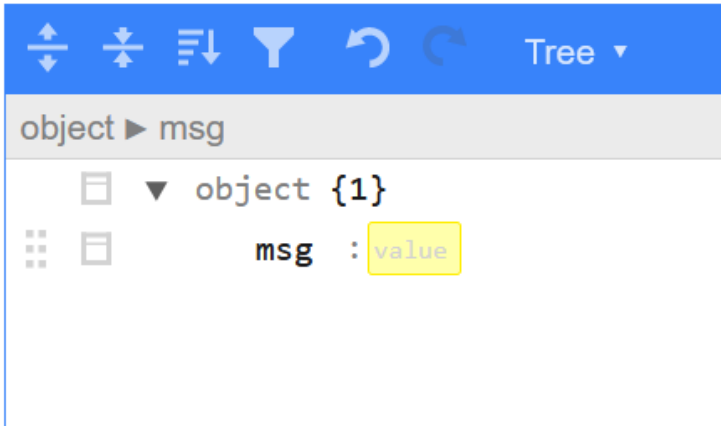
Add auto type by click on “Append”



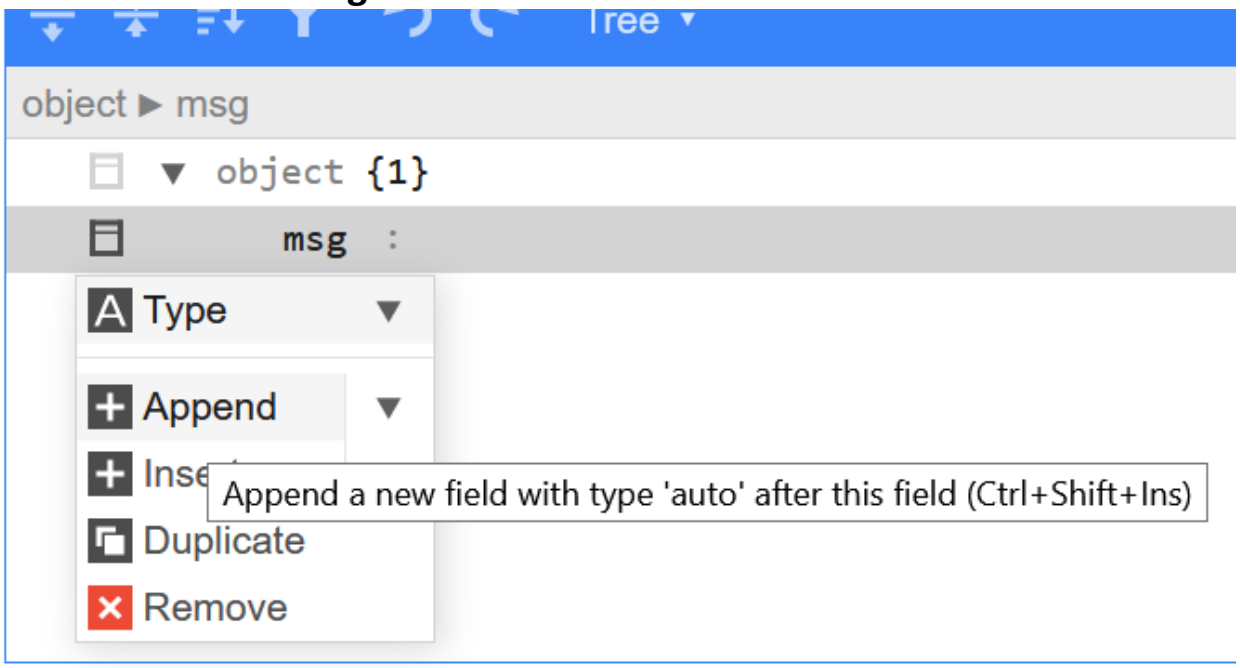
You will see this:



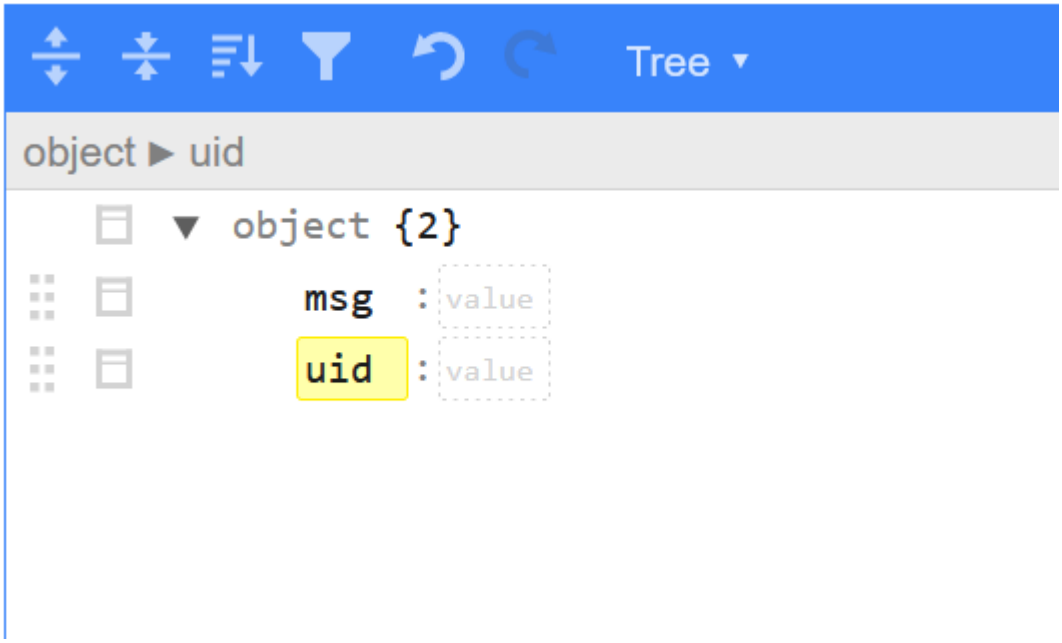
Change field to msg:



Add new field after **msg** field:

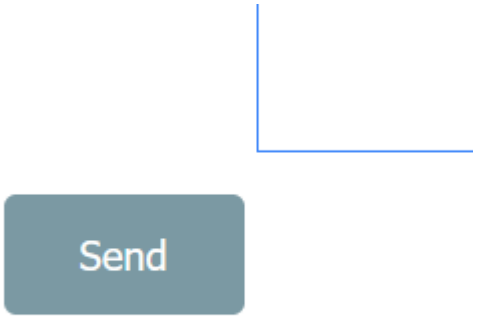


Result:



Let's try to execute some command.
uid- your game user id
msg - GM command
For example, command "**item add 201 10000**" should add 10 000 Primogems:

And press "**send**" button (make sure server bins running before it)



Answer:

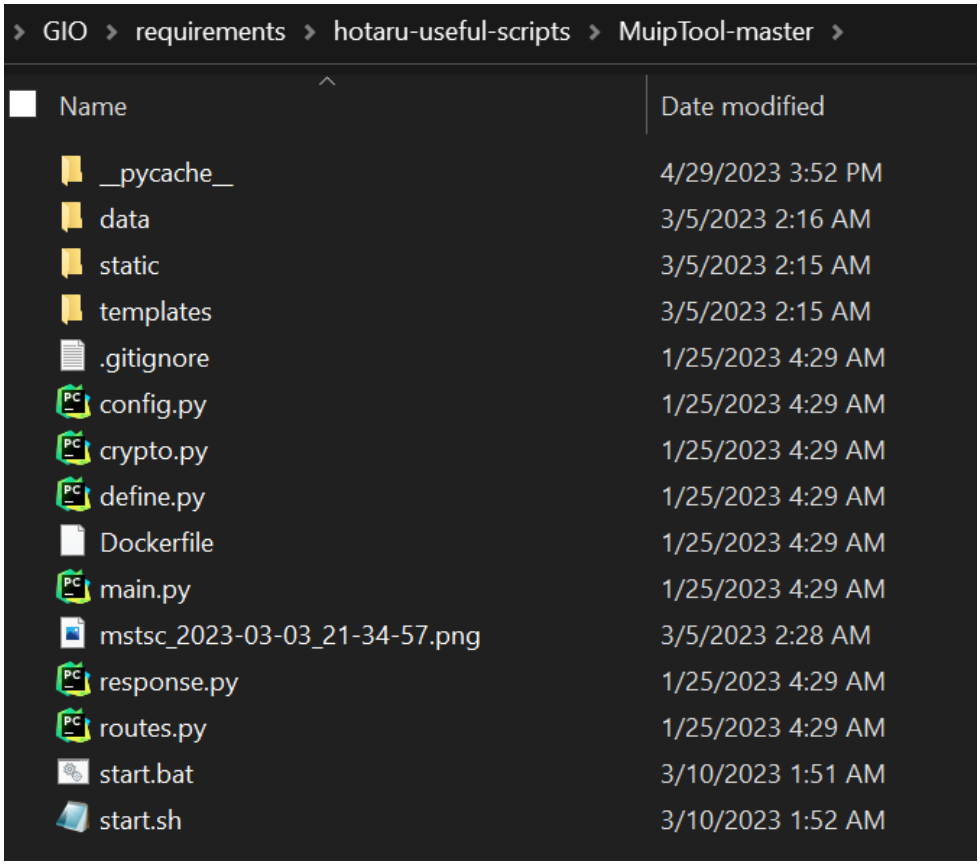

```
View ▾
Select a node...
▼ object {4}
  ▶ data {2}
    msg  : succ
    retcode : 0
    ticket : bnGOCEascAzYRnkTEdLDXwKwZYBkYsSW
```


PLAYER MUST BE ONLINE TO EXECUTE GM COMMAND!

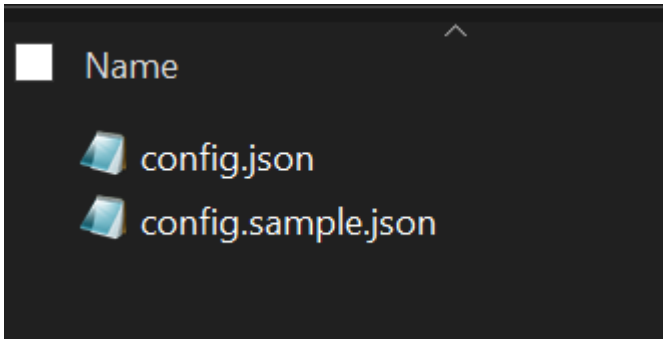
MUIP Tool sign key

Make sure you using same key in MUIP Tool and .xml config (in muipserver), also you can change it:

- Open MUIP Tool folder:



- Go to data:



- Open config.json

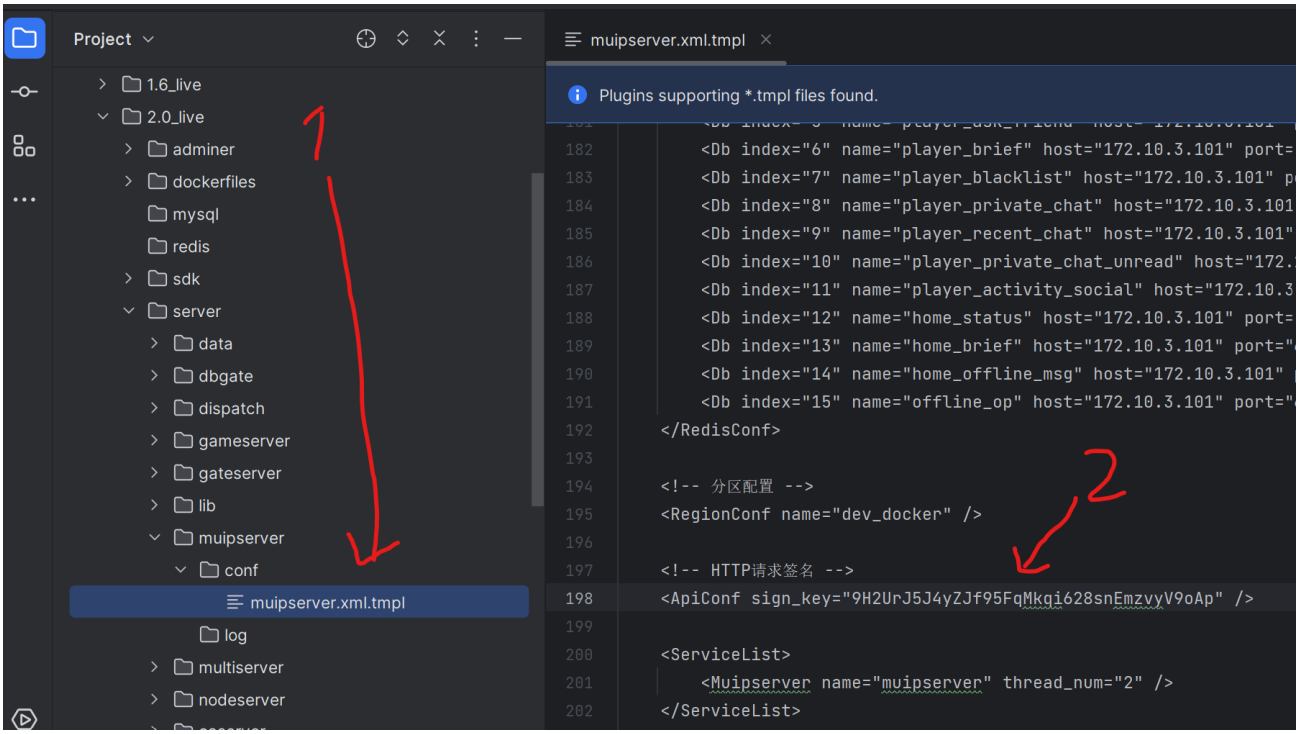
config.json - Notepad

File Edit Format View Help

```
{
  "app": {
    "listen": "127.0.0.1",
    "port": 8483,
    "debug": false
  },
  "muip": {
    "dev_docker": {
      "host": "http://127.0.0.1:21051",
      "sign": "9H2UrJ5J4yZJf95FqMkqi628snEmzvyV9oAp",
      "command_map": {
        "1001": "queryPlayerAccountUid",
        "1002": "queryPlayerUidByAccountUid",
        "1004": "queryPlayerBinInfo",
        "1005": "sendMail",
        "1006": "queryRedisMailInfo",
        "1007": "queryPlayerPostion",
        "1009": "queryCombatForce",
        "1011": "queryRegions",
        "1012": "queryPlayerWorldBinInfo",
        "1013": "queryPlayerBlockBinInfo"
```

1. **host** – real muipserver address from docker container
2. **sign** – key in muipserver.xml

- Open **2.0_live/server/muipserver.xml.tmpl** (and **muipserver.xml** if exists, change both configs) and change **sign_key**



How to use GM commands without MUIP Tool

For example, go to **2.0_live/server/muipserver.xml.tmpl** and **2.0_live/server/muipserver.xml (only if exists)** and change:

<ApiConf sign_key="9H2UrJ5J4yZJf95FqMkqi628snEmzvyV9oAp" />
To:
<ApiConf sign_key="" />

```
193
194     <!-- 分区配置 -->
195     <RegionConf name="dev_docker" />
196
197     <!-- HTTP请求签名 -->
198     <ApiConf sign_key="" />
199
200     <ServiceList>
```

Restart muipserver container and then you can use it without sign key (unsafe if you running a public server):

In my case:

SERVER_IP – 127.0.0.1

MUIP_PORT - 21051

region - dev_docker

UID – 1

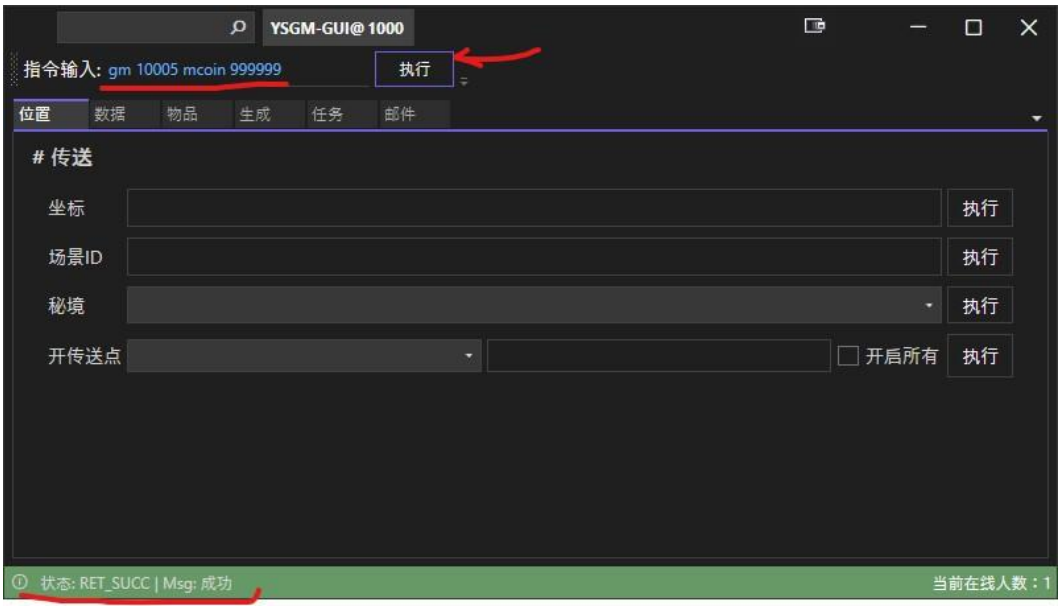
COMMAND - item add 201 10000

Use direct link to execute GM command (without sign_key):

http://SERVER_IP:MUIP_PORT/api?region=dev_docker&ticket=GM&cmd=1116&uid=UID&msg=COMMAND

Alternate GM GUI

Some people use this China GM GUI (not recommended because no source code, I will not provide a link to it)



Change in config**.json** inside GM_GUI:

{"MUIP_HOST":"http://<ServerIP>:10106/api","MUIP_TARGET_REGION":"dev_docker"}

Don't forget to replace <ServerIP> to your Docker IP

Use direct link to execute GM command without sign_key (change your UID and COMMAND):
http://<ServerIP>:10106/api?region=dev_docker&ticket=GM&cmd=1116&uid=UID&msg=COMMAND

You should change only this params to yours:
ServerIP
UID
COMMAND

All known GM cheat commands

Note:
Put these commands to **msg** field in **MuipTool**
<id> mean numeric value like 5 without < and > cgaracters
(on|off) mean on OR off without “(“, “|”, “)” characters

Commands:
quest add|accept|finish <id?> -- Add/Accept/Finish Quest
clear [pquest]
finish [father(v2)] <id>
finishv2 <id>
fail [father] <id>
random <clear|template> [id (if template)]
cancel [father] <id>
set_global <mainQuestId>? <val>
set_timevar <mainQuestId> <val>
state <id> <state>
point <sceneid> <all|id> -- unlock waypoints
avatar add <id> -- Add avatar
avatar add 10000002 -- Add avatar Ayaka
avatar change <id> -- needs already owned avatar
wudi global (avatar|monster) (on|off) -- Infinite hp your avatar or monster
break <level> -- promotelevel
mail addconfig <id> [params]
item add <id>
kill self
kill monster <id|all>
monster <num|id> [id] [level]
mcoin 999999 -- Crystal
talent unlock all -- Talent
point 3 all -- unlock all teleports
energy infinite on / off -- Unlimited ele brust
stamina infinite on / off -- Unlimited stamina
item add id qty -- Send item
quest accept 30904 -- Unlock Multiplayer
quest accept 35801 -- Unlock Wish
quest accept 35603 -- Unlock Fly

player level 1 -- adventure level
player level 60 max -- adventure level
jump 1009 -- Teleports to scene_id, 1009=AngelShare, not all id_scene work
equip add 15505 1 4 -- Item add
monster 20010101 0 1 -- Monster spawn <id> <count> <level>
kill monster 20010101 -- Monster kill
kill monster all -- Kill all Monster
equip add 13501 90 6 (item_id=13501 level=90 protomte_level=6) -- Add Weapons
item add 223 10000 -- Item add, Intertwined Fate x10000
item add 224 10000 -- Item add, Asquaint Fate x10000
item add 102 555 -- Add Item, Adventure experience x555
item add 103 555 -- Add Item, Masterless Stardust x555
item add 104 555 -- Add Item, Masterless Starglitter x555
item add 201 10000 -- Add Item, Primogems x10000
item add 202 555 -- Add Item, Mora x555
item add 203 555 -- Add Item, Genesis Crystall x555
item add all -- Add all item + including all characters with level 1
item clear 102 555 -- Remove item
goto 1 1 1 -- x,y,z - in gc use /pos to see location
dungeon <scenedId> -- Teleports to dungeon
mcoin 10000 -- Add Genesis Crystal
scoin 10000 -- Add ???
hcoin 10000 -- Add Primogems
home_coin 10000 -- add Realm currency (used to exchange for items in the Realm Depot through Tubby and the Traveling Depot through Chubby)
submcoin 100 -- Sub Genesis Crystal
subhcoin 100 -- Sub Primogems
subscoin 100 -- Sub ???
subhome_coin -- Sub Realm currency

Unknown GM cheat commands

=== climate ===
cold
lethal
unlethal
nolethal
sbuff

=== weather ===
sun
hot
heat
cloud
rain
thuner
storm
lightning
snow
freeze
frost
ice
mist
fog
desert

angle <axis_y> <axis_z> <angle> -- ???
authority --
host ??? -- change host
buff ??? --
addteam ??? --
delteam ??? --
pause ??? --
move ??? --
log ??? --
login ??? --
login reward clear ??? --
region ??? --
openstate ??? --
reconnect ??? --

event ??? --
finish parent quest ??? --
platform ??? --
expedition ??? --
data ??? --
hadd ??? --
callback ??? --
takereward ??? --
start ??? --
resetall ??? --
dungeonteam ??? --
ready ??? --
clearwtr ??? --
mail ??? --
addconfig ??? --
mail adconfig ??? --
getall ??? --
update ??? --
makedefine ??? --
dataversion ??? --
fetter ??? --
fin
conceal
addexp
clearall
birth
player_birth
add_all_fetters
open_all_fetters
prop
dailytask
dailytask add <taskid> -- add task
dailytask finish <taskid> -- del task
addsure
dailytask addsure <taskid>
addpossible
dailytask addpossible <task_id> <next_days>
dailytask setvar <task_id> <index> <value>
resetvar

set_cycle_refresh
dailytask set_cycle_refresh <task_id> <value>
setweight
dailytask setweight <task_id> <weight>
removeweight
dailytask removeweight (all or task_id)
randtask
randtask refresh <rand_task_id> <revise_level>
trycreate
randtask trycreate <rand_task_id>
clear_cd
world
levellimit
world level <num>
clearcd
adjustlevel
directlevel
robot_get_data
combat_force
allow
mp allow <uid>
giving
gadget_giving
active
take_back
get_list
simulate_next_day
simulate_get_log
simulate_big_world_chest
simulate_dungeon ???
simulate_finish_one_dailytask
simulate_finish_one_randtask
simulate_expedition
simulate_kill_monster
simulate_add_player_time ???
test_coverity
simulate_interrupt_next_transfer
get_gameserver_appid
get_thread_index

investigation
quest_daily_notify
nearentity
print nearentity
lockteam
activity
clear_banner
get_all_meet_cond
cond
setcond
setscore
setscorelimit
all_watcher_takeable
select_avatar
push_tips
record
set_state
get_num
michiae
open_stage
close_stage
sealamp
popularity
sealamp popularity <value>
reset_watcher
active_watcher
add_watcher_progress
watcher_pushtips
take_activity_watcher_reward
test_trigger_watcher
mp_kill_monster_num
mp_kill_monster_num
tower
mid_ntf
schedule
set_to_newest
windseed
has_elemforce_wind
add_signal

battlepass
recharge
mpplay
gadgetplay
yace
salesman
forge_point
show_avatar_born_time
area_explore
aster
effigy
reset_reward
scene_tag <tagid>
reputation
hunting
group_reload_press
widget
flight
gallery
monster_yace

How to use IDA/HxD/etc to see other GM commands

Also, you can find hidden command by **HXD (**HEX editor here - https://mh-nexus.de/en/downloads.php?product=HxD20**)**, open binary **gameserver (~1GB+)**, and search for known command words like "HOME_COIN". You will see other GM commands around.

```
MCOIN
3 48 4 14 add_mcoin:5693 64 24 11 reason:5706 128 24 18 action_reason:5713
addMcoin
checkAddMcoin fail. add_mcoin:
SUBMCOIN
2 48 4 14 sub_mcoin:5725 64 8 11 reason:5736
subMcoin
subMcoin fails
 subMcoinNegative:
HCOIN
3 48 4 14 add_hcoin:5755 64 16 15 item_param:5762 96 24 11 reason:5763
addHcoin
checkAddItem fail. add_hcoin:
SUBHCOIN
1 32 4 14 sub_hcoin:5784
subHcoin
subHcoinNegative fails
 subHcoinNegative:
SCOIN
3 48 4 14 add_scoin:5808 64 16 15 item_param:5815 96 24 11 reason:5816
addScoin
checkAddHcoin fail. add_scoin:
SUBSCOIN
1 32 4 14 sub_scoin:5837
subScoin
subScoinNegative fails:
 subScoinNegative:
HOME_COIN
3 48 4 18 add_home_coin:5861 64 16 15 item_param:5868 96 24 11 reason:5869
addHomeCoin
checkAddHcoin fail. add_home_coin:
SUBHOME_COIN
1 32 4 18 sub_home_coin:5890
subHomeCoin
subHomeCoinNegative fails:
 subHomeCoinNegative:
```

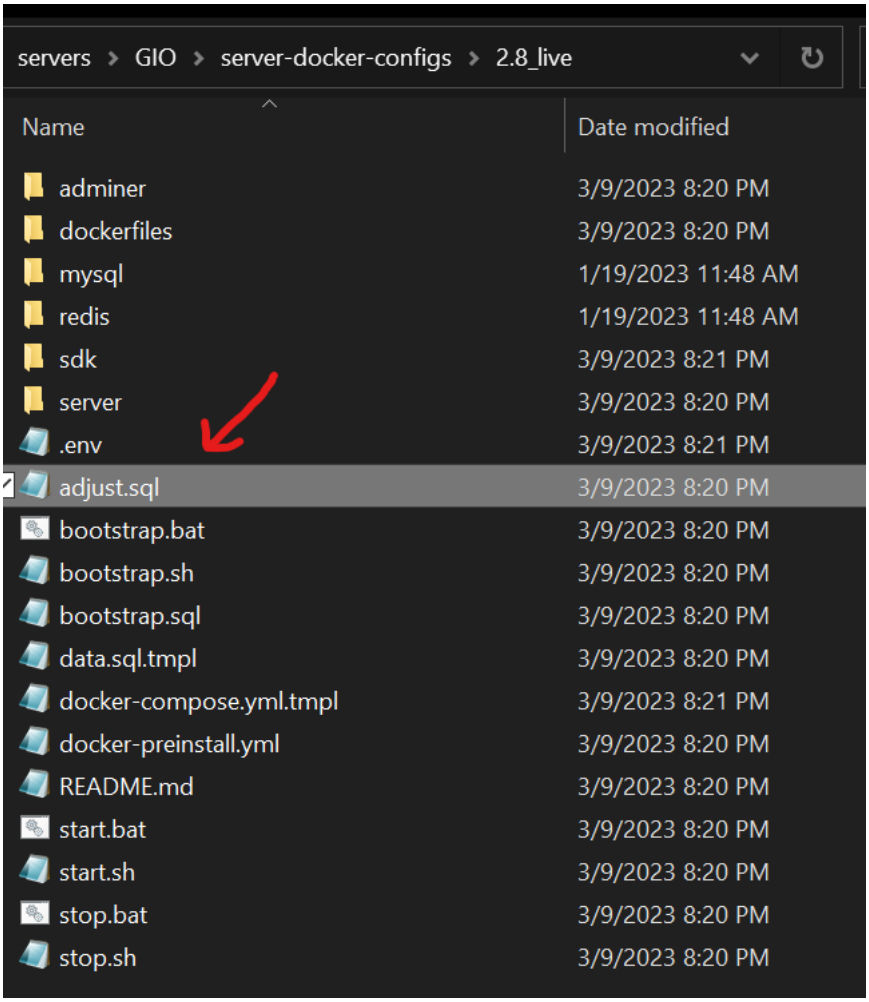
COMMAND

PARAMS

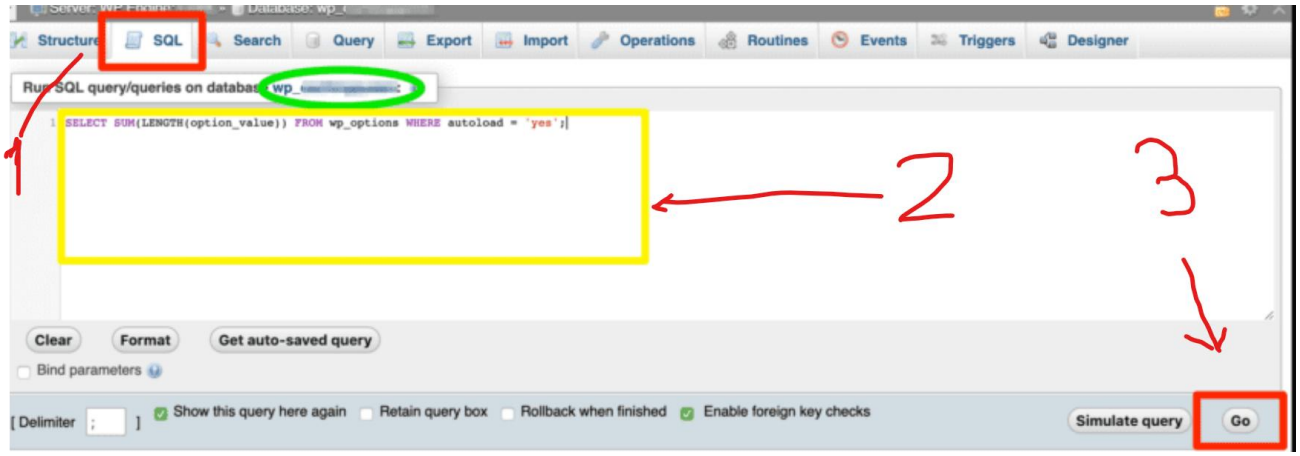
But better to use IDA PRO instead of HXD for RE

Adjust events time limit

If events in game disappear after some time – go to your docker folder and search this file:



Copy entire text from this file, go to **phpMyAdmin** or **adminer,** go to SQL tab, paste and execute. Restart docker containers (maybe).



Fix buggy 80001 and 20001 quests for 3.2 server

Replace **QuestData_Exported.txt** from **data** folder to another from the "**server-requirements**"

Fix Spiral Abyss for 3.2 server

Replace **TowerScheduleData.txt** from data folder to another from the "**server-requirements**"

Better Solo Mod for 3.2 server

Better Solo Mod – very useful mod for alone-gaming. Read readme.txt inside **Better Solo Mod** folder from the "**server-requirements**"

Change banners

If anyone is interested you can add/remove banners in **t_gacha_schedule_config** table in DataBase (use phpmyadmin), you can get all the IDs necessary from banners.json from grasscutter for ex. for Raiden GachaShowPanelA052 you can use the existing banners as example how to fill the data. If you add wrong ID's dbgate/gateserver might crash so watch out for that or just remove the banners making them crash.

GM mail

1. Use "GM Mail Web" from the requirements OR just web site https://memetrolls.net/miniprojects/mailparser/ to generate output.

3. Generate output (like blablbablbablablbabla)

4. Command: mail <output>

- Put this command to **MUIPTool:**

Uid 1000

mail blablablablablablablablablablablablablabla

- Or put to **GM_GUI:**

gm 1000 mail blablablablablablablablablablablablablabla

Where 1000 - your uid and "blabla...." - output from the GM Mail generator

Disable telemetry

Go to the server bins folder and do:

```
find . -type f \( -name "*.xml" -o -name "*.js" -o -name "*.json" -o -name "*.css" \)
-exec sed -i "s/log.trackingio.com/127.0.0.1/g" {} +

find . -type f \( -name "*.xml" -o -name "*.js" -o -name "*.json" -o -name "*.css" \)
-exec sed -i "s/s2s.adjust.com/127.0.0.1/g" {} +

find . -type f \( -name "*.xml" -o -name "*.js" -o -name "*.json" -o -name "*.css" \)
-exec sed -i "s/sandbox-sdk.mihoyo.com/127.0.0.1/g" {} +

find . -type f \( -name "*.xml" -o -name "*.js" -o -name "*.json" -o -name "*.css" \)
-exec sed -i "s/devapi-takumi.mihoyo.com/127.0.0.1/g" {} +

find . -type f \( -name "*.xml" -o -name "*.js" -o -name "*.json" -o -name "*.css" \)
-exec sed -i "s/uploadstatic-test.mihoyo.com/127.0.0.1/g" {} +

find . -type f \( -name "*.xml" -o -name "*.js" -o -name "*.json" -o -name "*.css" \)
-exec sed -i "s/uploadstatic-sandbox.mihoyo.com/127.0.0.1/g" {} +

find . -type f \( -name "*.xml" -o -name "*.js" -o -name "*.json" -o -name "*.css" \)
-exec sed -i "s/uploadstatic.mihoyo.com/127.0.0.1/g" {} +

find . -type f \( -name "*.xml" -o -name "*.js" -o -name "*.json" -o -name "*.css" \)
-exec sed -i "s/uploadstatic-sea.mihoyo.com/127.0.0.1/g" {} +

find . -type f \( -name "*.xml" -o -name "*.js" -o -name "*.json" -o -name "*.css" \)
-exec sed -i "s/api-os-takumi.mihoyo.com/127.0.0.1/g" {} +

find . -type f \( -name "*.xml" -o -name "*.js" -o -name "*.json" -o -name "*.css" \)
-exec sed -i "s/api-takumi.mihoyo.com/127.0.0.1/g" {} +

find . -type f \( -name "*.xml" -o -name "*.js" -o -name "*.json" -o -name "*.css" \)
-exec sed -i "s/preapi-takumi.mihoyo.com/127.0.0.1/g" {} +

find . -type f \( -name "*.xml" -o -name "*.js" -o -name "*.json" -o -name "*.css" \)
-exec sed -i "s/devapi-takumi.mihoyo.com/127.0.0.1/g" {} +

find . -type f \( -name "*.xml" -o -name "*.js" -o -name "*.json" -o -name "*.css" \)
-exec sed -i "s/webstatic.mihoyo.com/127.0.0.1/g" {} +

find . -type f \( -name "*.xml" -o -name "*.js" -o -name "*.json" -o -name "*.css" \)
-exec sed -i "s/op.mihoyo.com/127.0.0.1/g" {} +
```

Proto – what is that?

What is the protobufs? - https://protobuf.dev/ In simple words this is structured data allow the game communicate with the server. Please also learn about serialization/deserialization! Some Proto has an ID plus there are fields inside, fields have ID's too, also the Proto fields have values, some Proto can be nested in other proto

For example, we can compare 3.2 proto with 3.3 proto for **AvatarCardChangeReq**

AvatarCardChangeReq.proto v3.2.0

```
syntax = "proto3";

option java_package = "org.sorapointa.proto";

// CmdId: 688
// EnetChannelId: 0
// EnetIsReliable: true
// IsAllowClient: true
message AvatarCardChangeReq {
  uint32 item_id = 6;
  uint64 avatar_guid = 14;
  uint32 count = 7;
}
```

AvatarCardChangeReq.proto v3.3.0

```
syntax = "proto3";

option java_package = "org.sorapointa.proto";

message AvatarCardChangeReq {
  // enum CmdId {
  //   option allow_alias = true;
  //   NONE = 0;
  //   CMD_ID = 694;
  //   ENET_CHANNEL_ID = 0;
  //   ENET_IS_RELIABLE = 1;
  //   IS_ALLOW_CLIENT = 1;
  // }

  uint32 count = 11;
  uint32 item_id = 4;
  uint64 avatar_guid = 14;
}
```

Omg! Proto CmdId different (688 and 694), also field ids different (count, item_id, avatar_guid). That’s why you must use compatible Proto for each version! Otherwise, you game won't load and will stop on a white screen (after opening the gates). Each server bins already compiled with proto (inside).

Because of the different proto ids and fields ids you can’t play game version 3.3 with server bins for 3.2.

Proto dumping

Already dumped proto:

https://github.com/Sorapointa/Sorapointa-Protos

https://gitlab.com/YuukiPS/GC-Proto

https://github.com/NickTheHuy/3.3.0_proto.git

https://github.com/NickTheHuy/3.4_proto.git

https://github.com/NickTheHuy/3.5_protos.git

https://git.crepe.moe/crepe-inc/crepe-protos.git

https://github.com/CloudyPS/protos.git

Don't use this proto because proto for some versions not fully restored (cuz it obfuscated)! Dump it by yourself from server bins!

Protodumpers:

https://github.com/marin-m/pbtk

https://github.com/partypooperarchive/ProtobufDecoder

https://github.com/lilmayofuksu/sus-scripts

https://github.com/partypooperarchive/ObfProtoDecoder

https://github.com/nitrog0d/ProtoDumper

You can use **pbtk** to dump full proto from the server bins all CB and 1.0 up to 3.4 (after 3.2 srv bins proto shuffled by devs cuz leak and 3.3 game server have different proto than 3.3 game release). Just run pbtk gui and select gameserver bins (NOT game, IN server bins. You can dump it from the game BUT I doubt – it is very obfuscated, you can’t restore all fields properly but in servers bins – you can).

KCP/Enet and UDP/TCP

What is the **KCP**? - https://ims.improbable.io/insights/kcp-a-new-low-latency-secure-network-stack

Also, you should learn about **TCP/UDP** before!!! Just download books or see youtube guides. Game clients use **KCP** protocol to send protobuf data (see **proto**). You can use **KCP** Proxy to see Proto data:

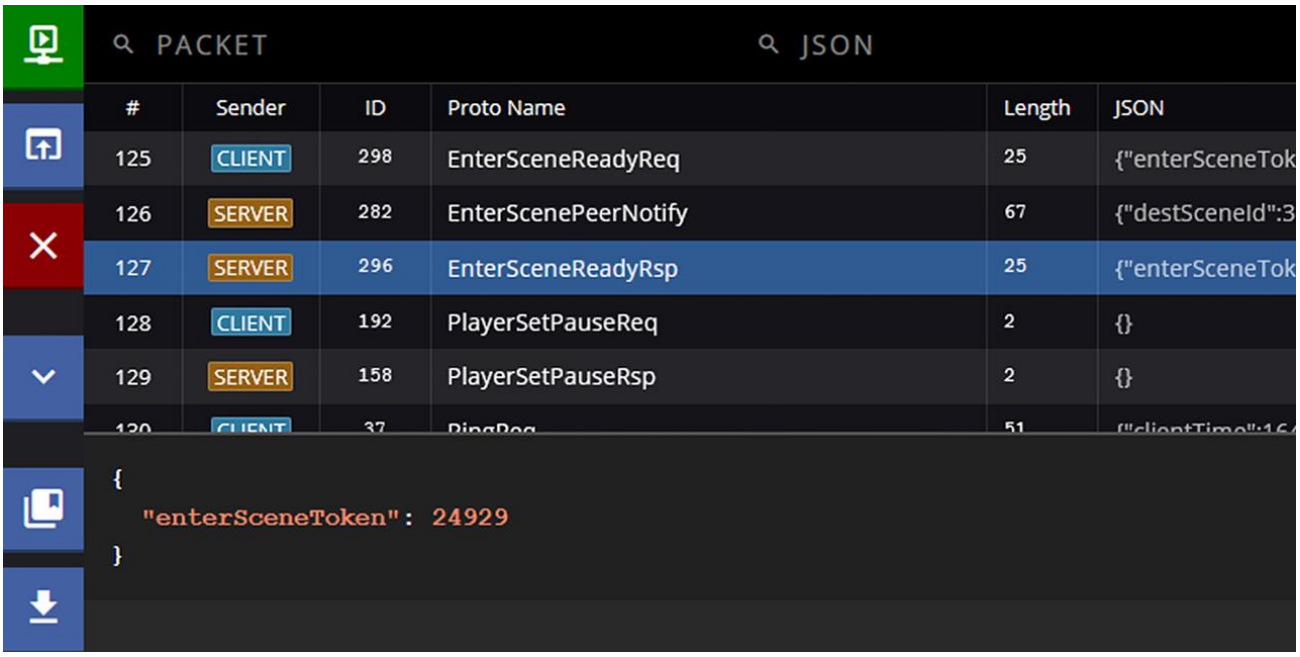
https://github.com/Crepe-Inc/Iridium

https://github.com/GrownNed/gidra

https://github.com/MoonlightPS/Iridium-gidra

Some very old game clients use **Enet** instead of **KCP**! Also, you should be able to dump proto (see **proto dumpers**). And you should be able to restore names/cmdid/fields ids. Also you can play game, record session, decode every proto and see values in proto fields, use brain to figure out how you should do it in your **Private Server,** also you can use **WireShark** to record your game packets to .pcap

Example of the **Iridium** proxy proto visualizer:



Proto shifting on the fly using KCP Proxy

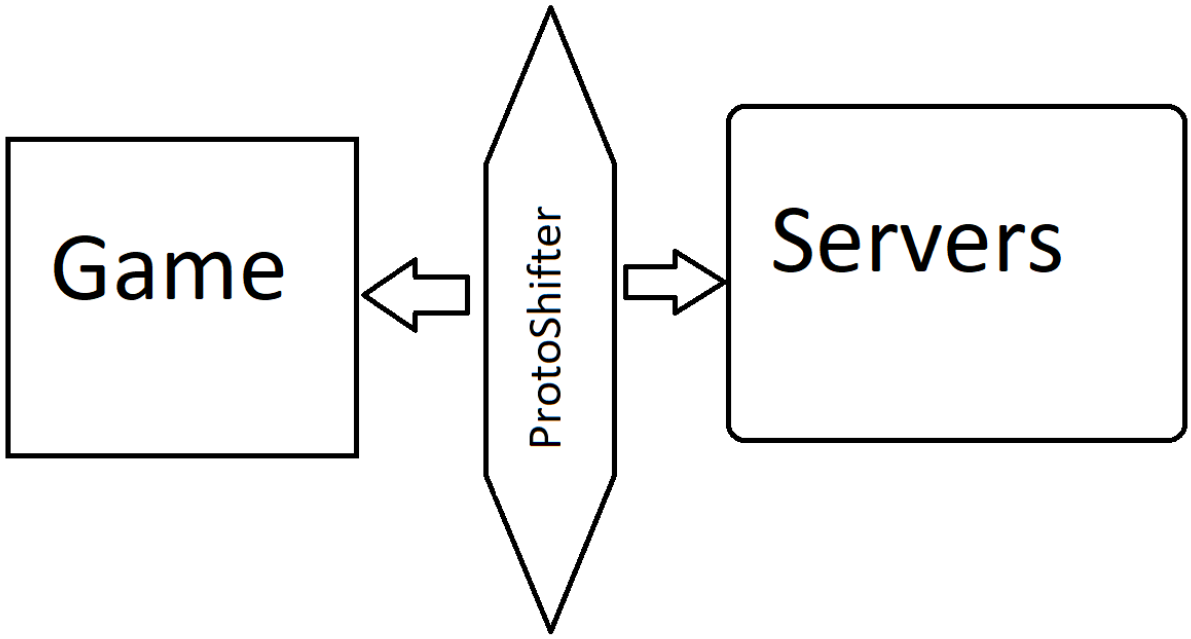
As I said before - Because of the different proto ids and fields ids in the different server bins you can't play game version 3.3 with server bins for 3.2 natively. But we can write a some KCP Proxy to shift all proto ids "on the fly" and convert new Proto ids to old proto ids (or vice versa). This proxy should be in middle between **Game** and **Server** and should decode. Change and encode **protobuf data** via **KCP protocol**. This allows you to play 3.3 game with 3.2 server.

See these repositories of ProtoShifters (choose one):

https://github.com/YuFanXing/ProtoShift

https://github.com/Jx2f/ViaGenshin

(Or use ViaGenshin from the "**server-requirements.rar**")



ProtoShifter can change proto ids on the fly in realtime

Method 2. VM-Installer - run server without Docker in Linux/VMware/Virtualbox/WSL

Download installer + script

Download "server-3.2-installer.rar" and "server-3.2-installer.sh":

https://anonfiles.com/Pd59ndo8z9/server_3_2_installer_rar

https://anonfiles.com/wc5en6o2z4/server_3_2_installer_sh

This installer only for 3.2 version. Server bins for 3.2 and data already in installer archive. Please don't use this method with installer because the new method with Docker better and support all versions.

If links aren't available – please check new guide in our Discord - https://discord.gg/MfxYRNdD9x

Prepare your virtual machine

1. Install VMWare / VirtualBox / WSL (one of these of your choice)
   For example, links for VMWare PRO:
   Download for Win - https://www.vmware.com/go/getworkstation-win
   Download for Lin - https://www.vmware.com/go/getworkstation-linux
   Keys for 17x version:
   MC60H-DWHDS-H80U9-6V85M-8280D
   Keys for 16x version:
   ZF3R0-FHED2-M80TY-8QYGC-NPKYF
   YF390-0HF8P-M81RQ-2DXQE-M2UT6
   ZF71R-DMX85-08DQY-8YMNC-PPHV8

2. If you using WSL - skip this step. For VBox/VMware create a new VM with 30+ GB of HDD space and 18 Gb or RAM (24 Gb recommended).

3. Install **Ubuntu 20.04 Desktop** or **Server edition** (NOT **Ubuntu 22.04**, it's incompatible). Desktop – with **Graphical User Interface**, server – without it (only black terminal)!
   You need ISO of Ubuntu 20.04.5 server OR desktop, which you can get here:

   https://www.releases.ubuntu.com/focal/ubuntu-20.04.5-live-server-amd64.iso

   https://www.releases.ubuntu.com/focal/

Ubuntu 20.04.5 LTS (Focal Fossa)

Select an image

Ubuntu is distributed on three types of images described below.

Desktop image

The desktop image allows you to try Ubuntu without changing your computer at all, and at your option to install it permanently later. This type of image is what most people will want to use. You will need at least 1024MiB of RAM to install from this image.

64-bit PC (AMD64) desktop image

Choose this if you have a computer based on the AMD64 or EM64T architecture (e.g., Athlon64, Opteron, EM64T Xeon, Core 2). Choose this if you are at all unsure.

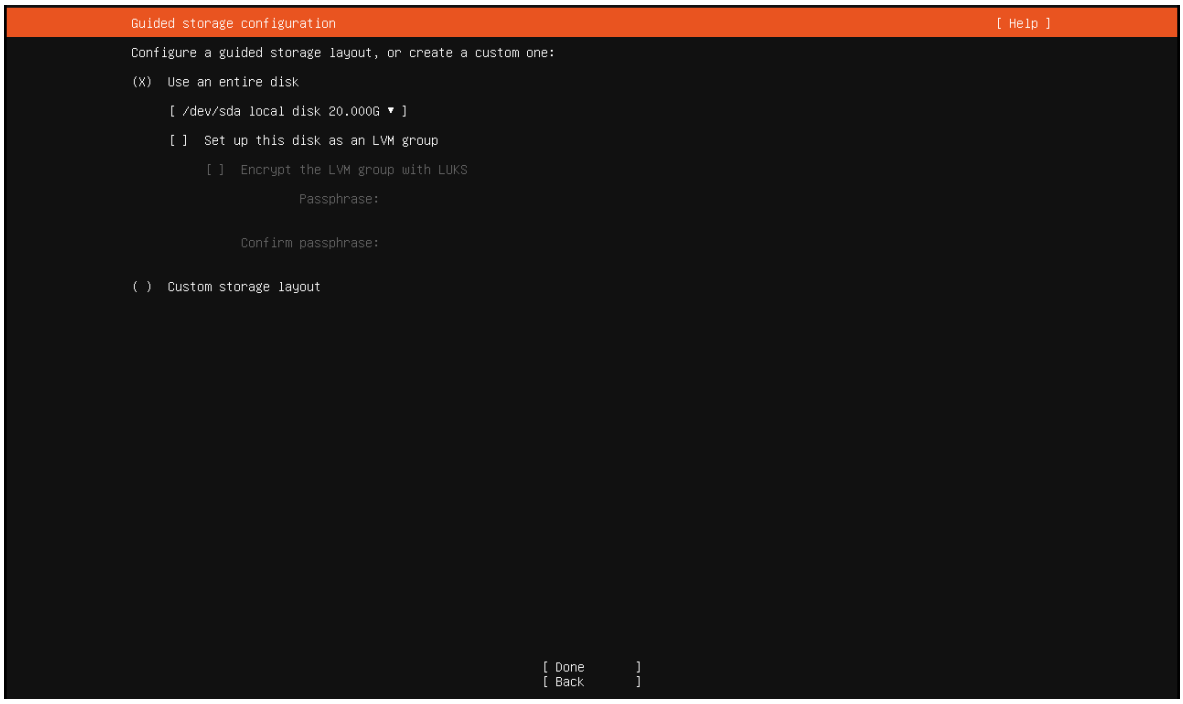
Server install image

The server install image allows you to install Ubuntu permanently on a computer for use as a server. It will not

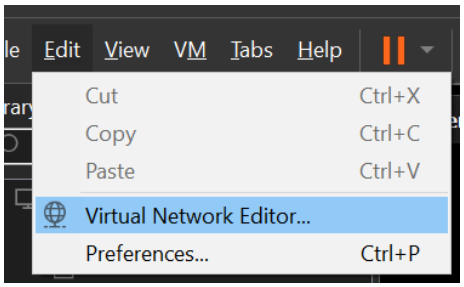
64-bit PC (AMD64) server install image

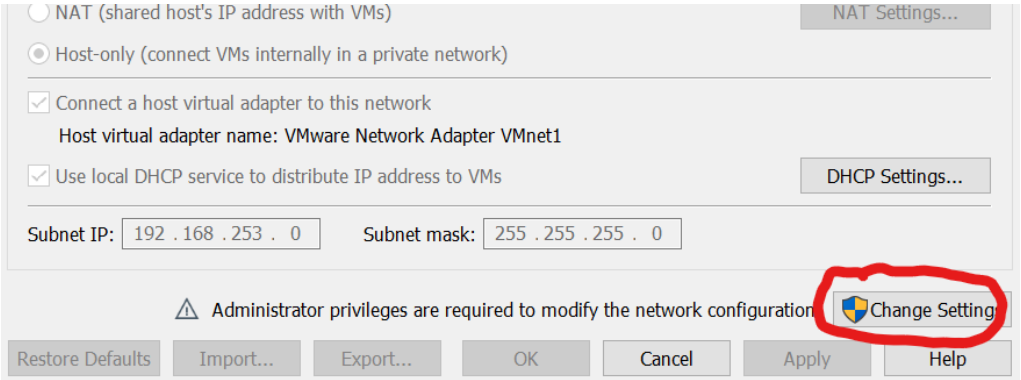
Choose this if you have a computer based on the AMD64 or EM64T architecture (e.g., Athlon64, Opteron, EM64T Xeon, Core 2). Choose this if you are at all unsure.

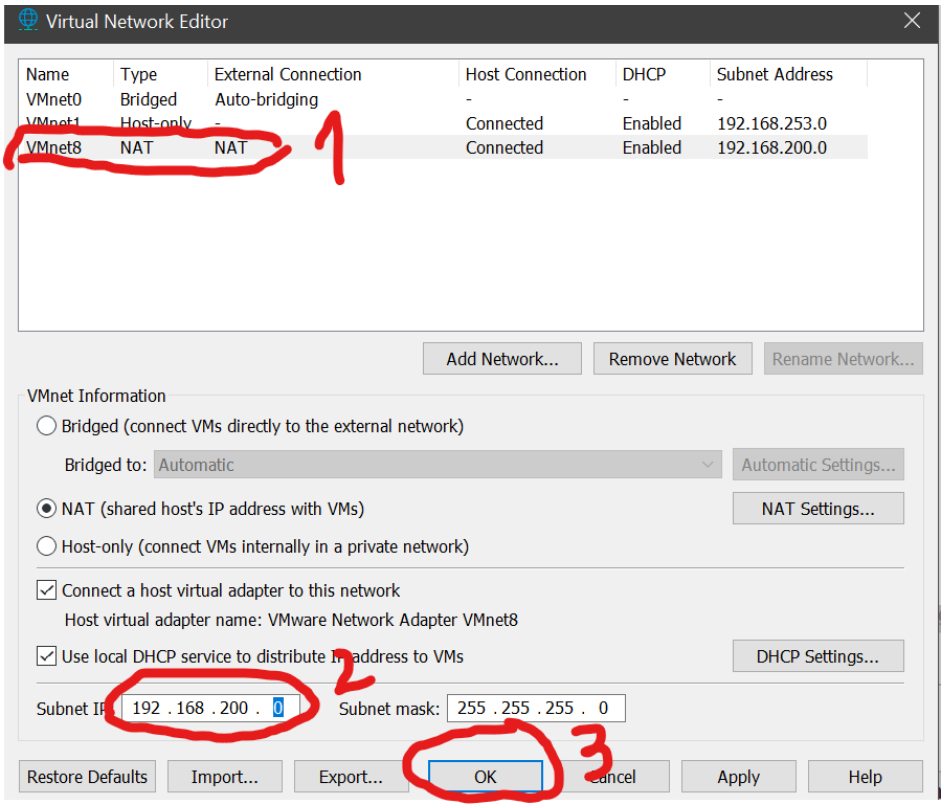
Installation for Desktop is very easy, but if you installing in server mode - very important: when setting up the disk partitioning while installing the ubuntu server, you have to uncheck the "Set up this disk as an LVM group" like on this screenshot:



4. If you using **WSL** - skip this step. Open Vmware/VBox networks settings/configuration:
   In vmware Edit -> Virtual Network Editor -> (on Bottom clock Change Settings with shield) -> select YES -> click on VMnet with NAT -> in Subnet IP set: 192.168.200.0 and mask 255.255.255.0







5. If you using WSL - skip this step. If you using Desktop Ubuntu 20.04 - install **VMware/VBox Guest additions** if it not installed yet (search in google how to install it).

6. Set static ip in Ubuntu in VMware/Vbox (I dunno how to do it in WSL).

For Desktop edition:

sudo nano /etc/netplan/01-network-manager-all.yaml

You should do:

```
# Let NetworkManager manage all devices on this system
network:
  version: 2
  renderer: NetworkManager
  ethernets:
    ens33:
      dhcp4: no
      addresses: [192.168.200.130/24]
      # you can delete this line to disable WAN internet
      # instead of using iptables
      gateway4: 192.168.200.2
      nameservers:
        addresses: [1.1.1.1,8.8.8.8]
```

For Server edition:

sudo nano /etc/netplan/00-installer-config.yaml

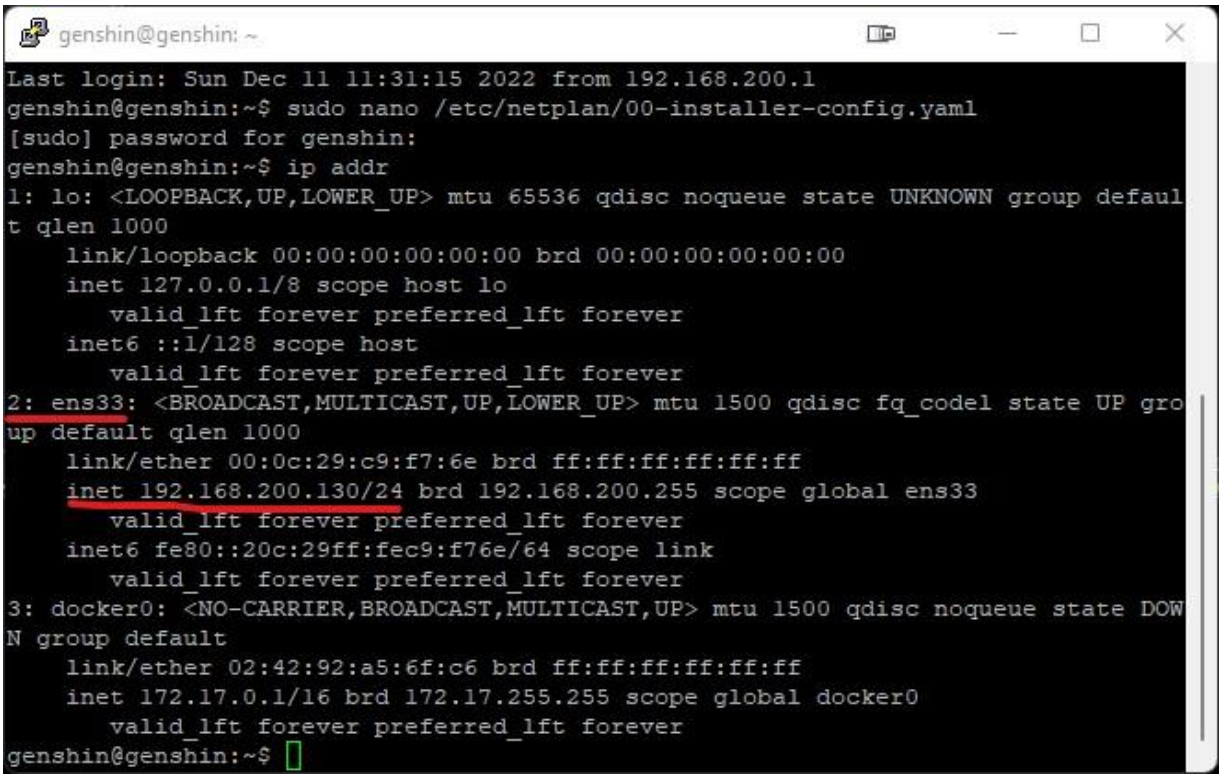
You should do:

```
# This is the network config written by 'subiquity'
network:
  version: 2
```

```
ethernets:
  ens33:
    dhcp4: no
    addresses: [192.168.200.130/24]
    # you can delete this line to disable WAN internet
    # instead of using iptables
    gateway4: 192.168.200.2
    nameservers:
      addresses: [1.1.1.1,8.8.8.8]
```

7. Apply the IP changes:
   sudo netplan --debug apply

8. Enter this command to check your server ip again:
   ip addr
   It should show 192.168.200.130 now. Restart the VM and check IP again.



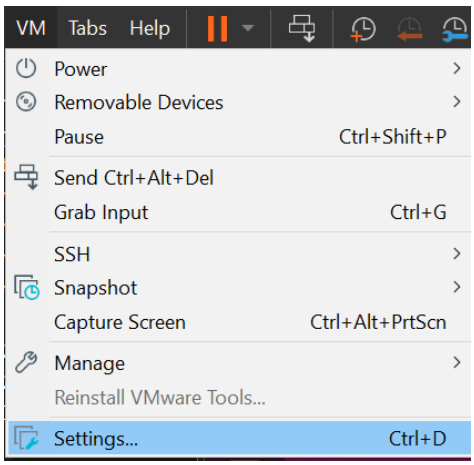
9. Check internet using:
   ping google.com
   wget google.com

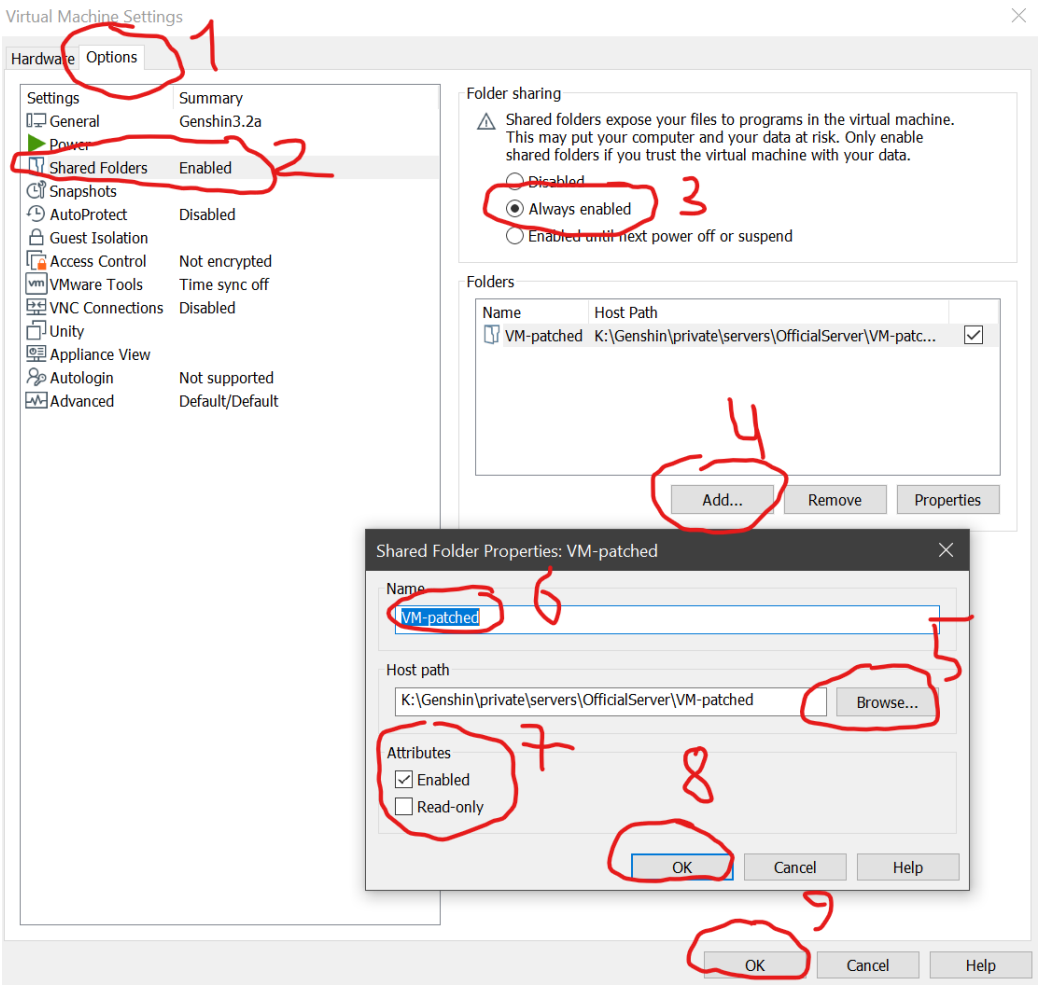
   It should work, you should have internet access (we will disable it later)

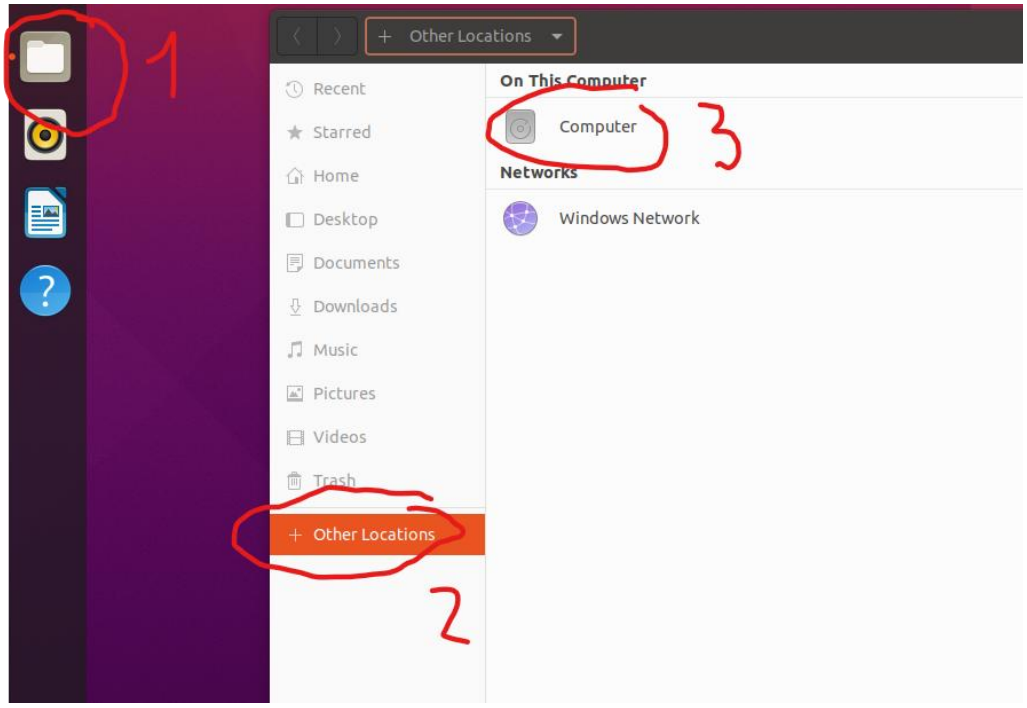
10. Go to shared folder:

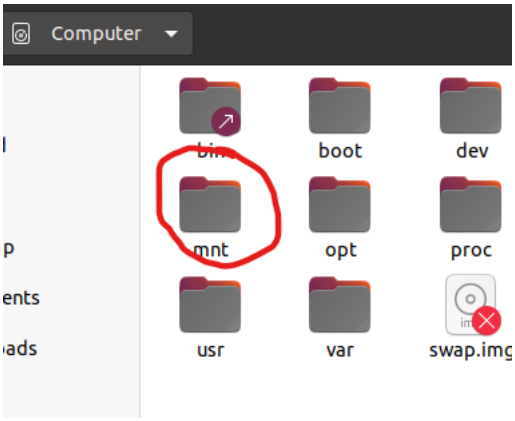
    If you using WSL - you can see all disks using "cd /mnt/ && ls -la". And for example, you can go to "cd /mnt/d/folder1/folder2".

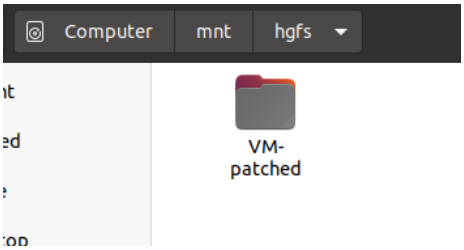
    If you using VMware/VBox with installed guest additions - create a shared folder (from VM settings) pointing to the desired location of the server files. Assign this folder to the VM. In vmware you can go to /mnt/hgfs to see mounted folders. In vbox it is a network drive. See below guide for VMware:




Virtual Machine Settings
Hardware
Options
1
Settings
Summary
General
Genshin3.2a
Power
Shared Folders
Enabled
2
Snapshots
AutoProtect
Disabled
Guest Isolation
Access Control
Not encrypted
VMware Tools
Time sync off
VNC Connections
Disabled
Unity
Appliance View
Autologin
Not supported
Advanced
Default/Default
Folder sharing
Shared folders expose your files to programs in the virtual machine. This may put your computer and your data at risk. Only enable shared folders if you trust the virtual machine with your data.
Disabled
Always enabled
3
Enabled until next power off or suspend
Folders
Name
Host Path
VM-patched
K:\Genshin\private\servers\OfficialServer\VM-patc...
4
Add...
Remove
Properties
Shared Folder Properties: VM-patched
Name
6
VM-patched
5
Host path
K:\Genshin\private\servers\OfficialServer\VM-patched
Browse...
Attributes
7
Enabled
Read-only
8
OK
Cancel
Help
9
OK
Cancel
Help


1
Other Locations
Recent
Starred
Home
Desktop
Documents
Downloads
Music
Pictures
Videos
Trash
Other Locations
2
On This Computer
Computer
3
Networks
Windows Network





VM-patched – just my folder name with all installer scripts. You can use another folder name

11. Allow **HGFS auto mount** on **open-vm-tools** in VMWare

    It appears that when you install open-vm-tools on Ubuntu for VMware Workstation the HGFS mount is never created. To get your shared folders to show up we need to perform a few steps.

    We need to create the /mnt/hgfs folder.

    sudo mkdir /mnt/hgfs

    To mount your shares temporarily run the following command.

    sudo mount -t fuse.vmhgfs-fuse .host:/ /mnt/hgfs -o allow_other

To persist the mount you will need to edit the `/etc/fstab` file. Add the following line.

.host:/ /mnt/hgfs fuse.vmhgfs-fuse allow_other 0 0

Reboot the host to verify persistence of the mount. If it doesn't work, try to add another line:

vmhgfs-fuse /mnt/hgfs fuse defaults,allow_other 0 0

12. Copy **server-3.2-installer.rar** and **server-3.2-installer.sh** to **/root** or **/home/userName** (don't use it "in place" on your NTFS folder instead of EXT4 Linux - otherwise you will get **VERY** poor performance)

13. Run:

sudo su
cd ~
chmod +x server-3.2-installer.sh && ./server-3.2-installer.sh

Enter access ip: **192.168.200.130**
Enter local ip: **192.168.200.130**

14. If you using VmWare/VBOX - Before start servers you'd better disable WAN access (keep local only) because this servers can leak telemetry to the other real original servers:

For Desktop edition:

sudo nano /etc/netplan/01-network-manager-all.yaml

For Server edition:

s**udo nano /etc/netplan/00-installer-config.yaml**

Comment line (add #) with "gateway4: 192.168.200.2":

 # gateway4: 192.168.200.2

Save using “ctrl + x” -> “Y” -> Enter

15. Apply the IP changes:

    sudo netplan --debug apply

16. Try to ping google:

    ping 8.8.8.8

    You shouldn't recieve any bytes!

    You should get message like “ping: connect: network unreachable | error | etc”

    If you can ping google – network isn’t disabled, try to recheck network settings file…

    If you using WSL OR still can ping google - so... In this case - use script from "**requirements/wan disable.sh**"

17. If you need to use internet inside VM again: "requirements/wan enable.sh". ATTENTION! After "wan enable.sh" you should restart all docker containers:

    sudo docker kill $(docker ps -q) && sudo docker start $(docker ps -a -q)

18. Read how to start from installer when it done

    If you wish to start only important servers (set ~16GB RAM, your system should be 20GB+ RAM):

    ./cli start

Following server has been disable in **cli** file:

 # cd ../oaserver

 # chmod +x oaserver

 # nohup ./oaserver -i 9001.9.1.1 &

 # cd ../pathfindingserver

 # chmod +x pathfindingserver

 # nohup ./pathfindingserver -i 9001.8.1.1 &

 # cd ../multiserver

 # chmod +x multiserver

 # nohup ./multiserver -i 9001.7.1.1 &

 # cd ../tothemoonserver

 # chmod +x tothemoonserver

 # nohup ./tothemoonserver -i 9001.10.1.1 &

If you wish to start ALL servers (set ~24GB RAM to VM, your system should be 30GB+ RAM):

./cli-default start

And to stop all servers run:

./cli stop_all

To show server status run:

./cli status

p.s. also you can ./cli stop but it seems stucks, use CTRL + C to terminate and **./cli stop_all**

Also you can run ALL servers (even multiserver for the PVP and tothemoonserver for the Spiral Abyss):

./cli-default start

ATTENTION!

./cli require 16 GB RAM and **./cli-default** requires 23-25 GB RAM

19. Apply Fiddler script from "**requirements/fiddlerScript.txt**"

20. Run Fiddler

21. Run game version 3.2 (NOT 3.3), don't forget to use patched UserAssembly.dll, replace original .dll in game folder in: "GenshinImpact_Data/Native/")

22. Play with full features

23. Read all **GM** commands in previous chapters

24. Also, you can go to the Database admin panel:

    Address: **http://192.168.200.130/phpmyadmin**

    http! NOT https!

    Use login: **root** and pass: **f2c340a9-bf06-4345-9654-00b074b92fe8**

    OR login: **work** and pass: **GenshinImpactOffline2022**

    If it doesn’t work, then install it inside the VM like this:

docker run -d --net=host --restart always -e PMA_HOST="127.0.0.1" -e PMA_ARBITRARY="1" --name phpmyadmin phpmyadmin/phpMyAdmin

Method 3. Run already prepared server VM image using VMWare only

-) Install VmWare PRO

Download for Win - https://www.vmware.com/go/getworkstation-win

Download for Lin - https://www.vmware.com/go/getworkstation-linux

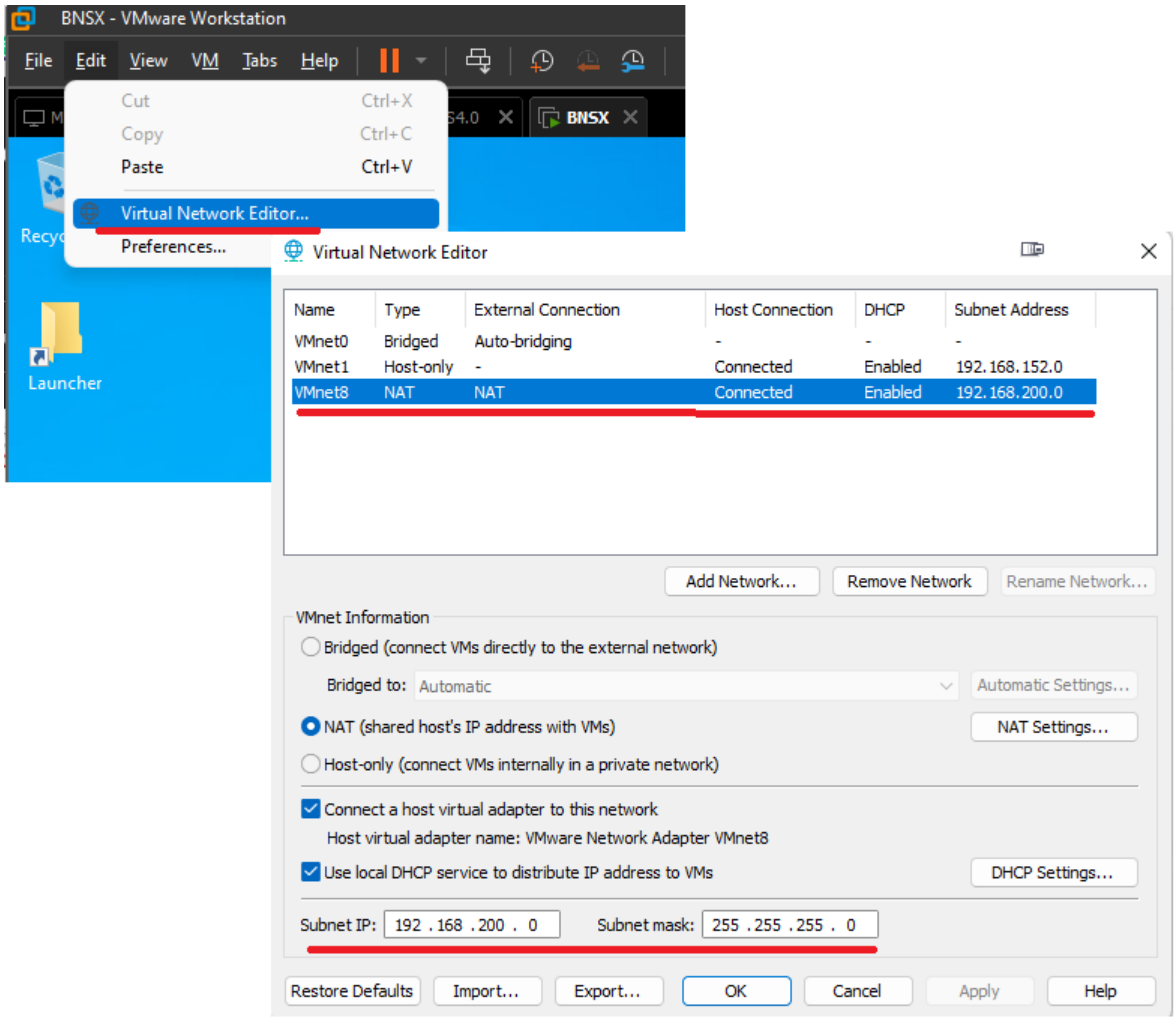
-) Keys for 17x version:

MC60H-DWHDS-H80U9-6V85M-8280D

Keys for 16x version:

ZF3R0-FHED2-M80TY-8QYGC-NPKYF
YF390-0HF8P-M81RQ-2DXQE-M2UT6
ZF71R-DMX85-08DQY-8YMNC-PPHV8

-) Configure Internet



-) Download and unpack **UbuntuServer-20.04-Genshin-3.2-x64.7z** (7 GB total)**:**

https://anonfiles.com/224fEcofz1/UbuntuServer_20_04_Genshin_3_2_x64_7z

OR **UbuntuDesktop-20.04-Genshin-3.2-x64.7z** (10 GB total)**:**

https://anonfiles.com/m7k6Cfo3zb/UbuntuDesktop_20_04_Genshin_3_2_x64_7z

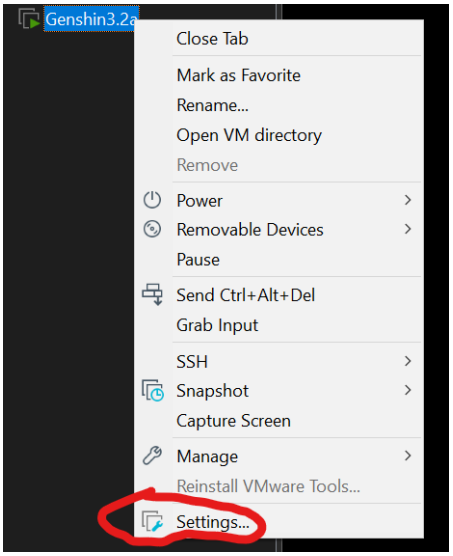
Also, you can try to search in I2P or TOR links if the links above unavailable:

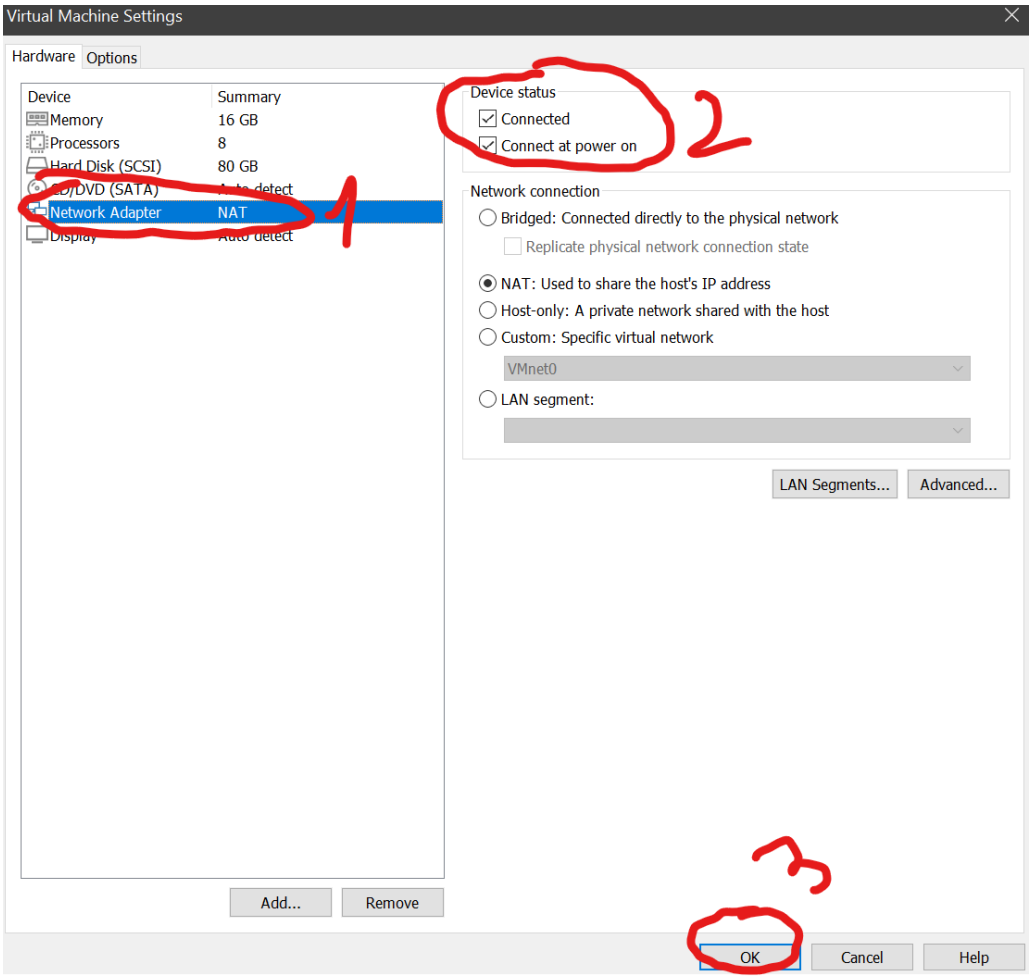
I2P Site URL (recommended, stable connection):
http://ufqsp7hhjc6hvbcqpiumocavnqu7ufpuvcfuynwew63dvoujqa3a.b32.i2p/

TOR ONION Site URL (NOT recommended, unstable connection):
http://b4ssb6i52vqyybsttuhax2wsp22td4rei6kmk2vt3nzj2hj5v54y5nyd.onion/

-) Open file Genshin3.2.vmx, virtual machine should appear in your VMWare dashboard

-) Configure VM using settings:





-) Start VM and login using login and password:

Login: genshin

Password: Genshin2023@@

-) You can connect to VM using: WSL:

ssh genshin@192.168.200.130

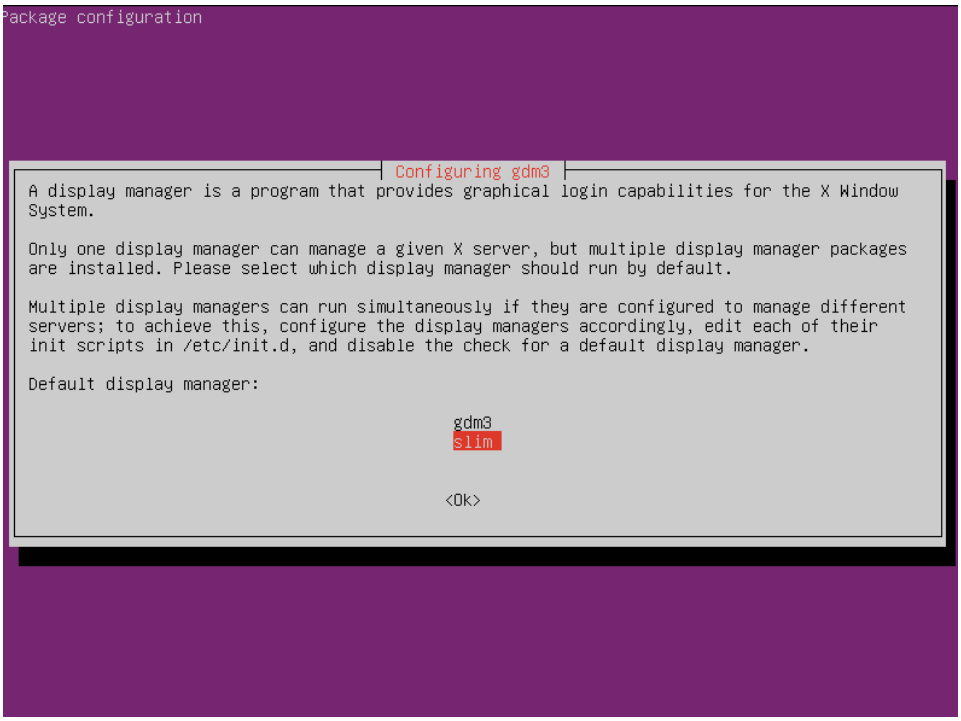
and write password OR using GUI Putty

-) Also, you can setup Desktop GUI in **Server edition** (if you using Desktop edition – just **SKIP** this steps):

sudo su

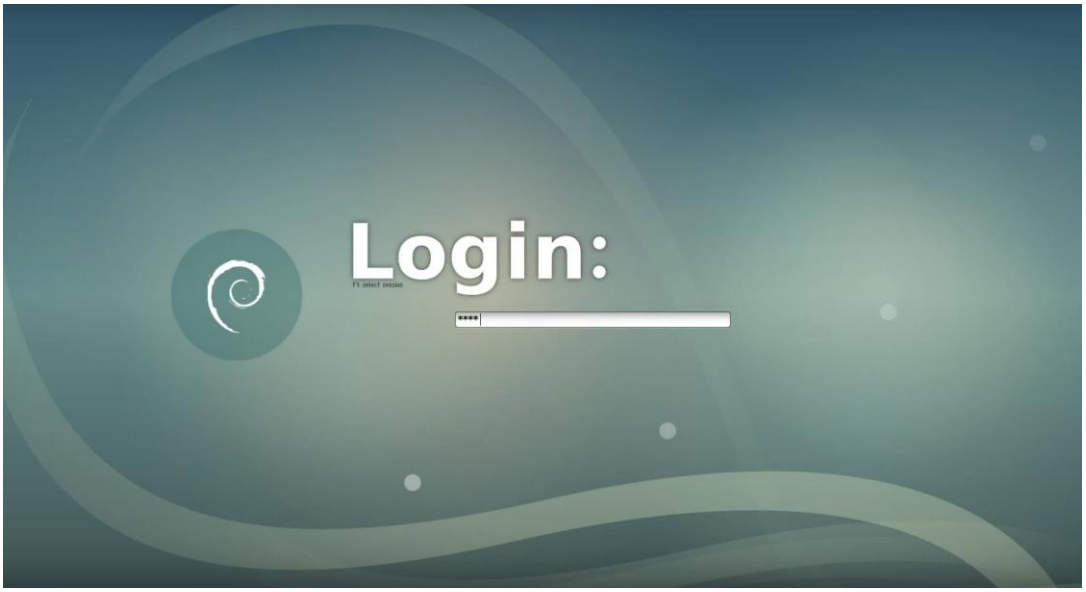
apt update && apt upgrade -y

apt install -y slim wget net-tools ubuntu-desktop

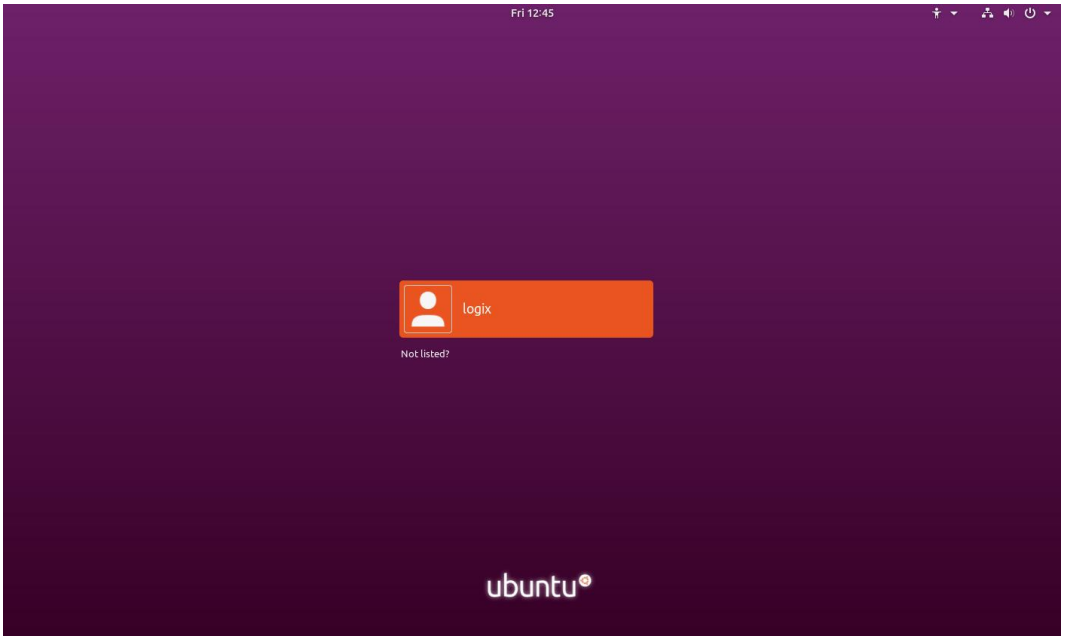


You can choose gdm3 OR slim.

Slim:



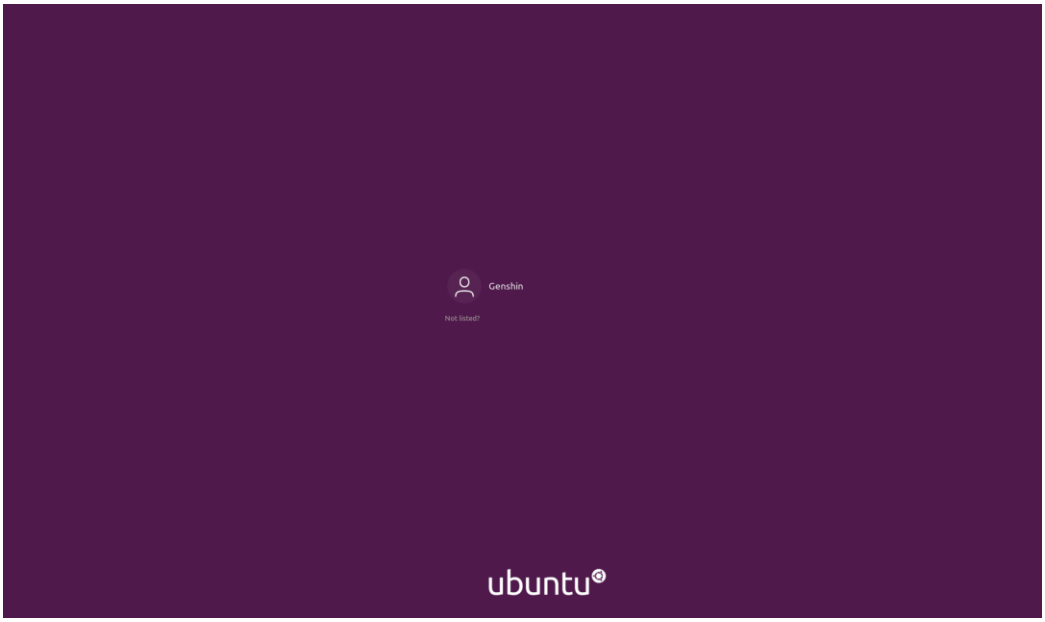
Gdm3:



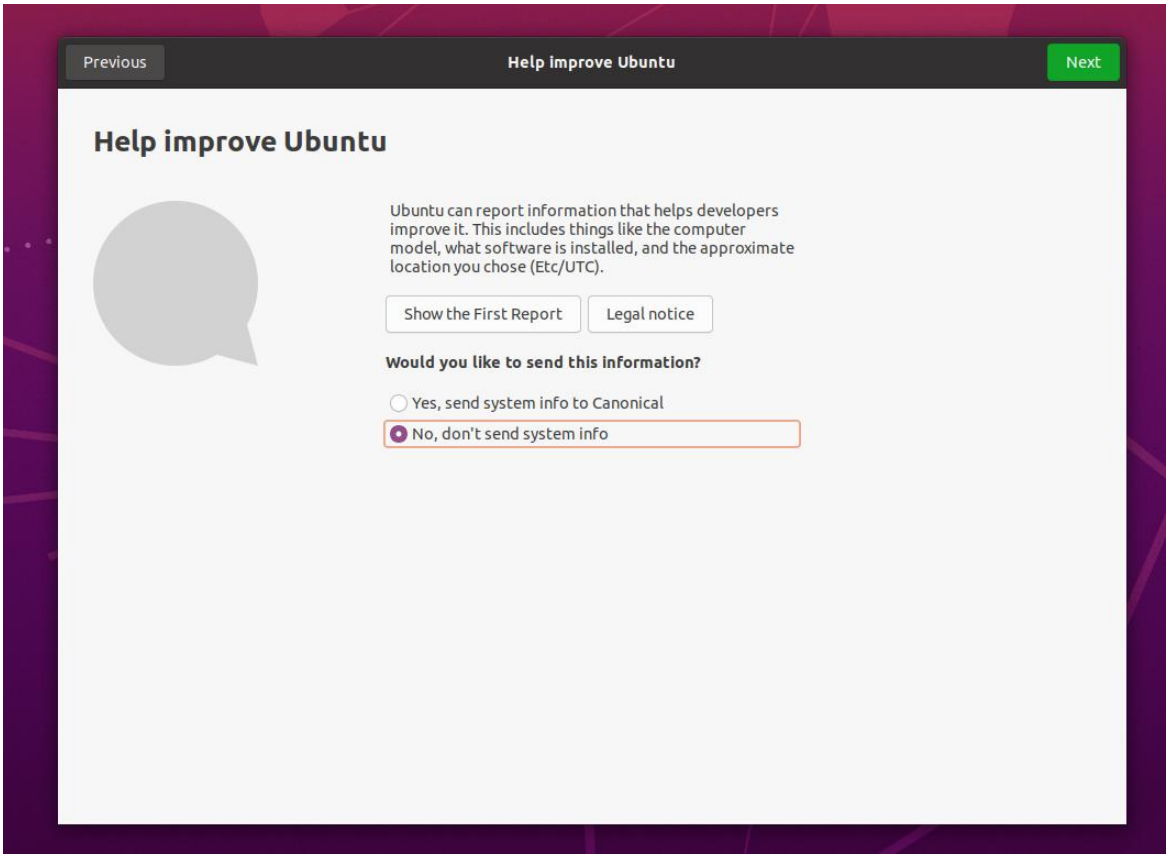
After GUI installation do:

reboot

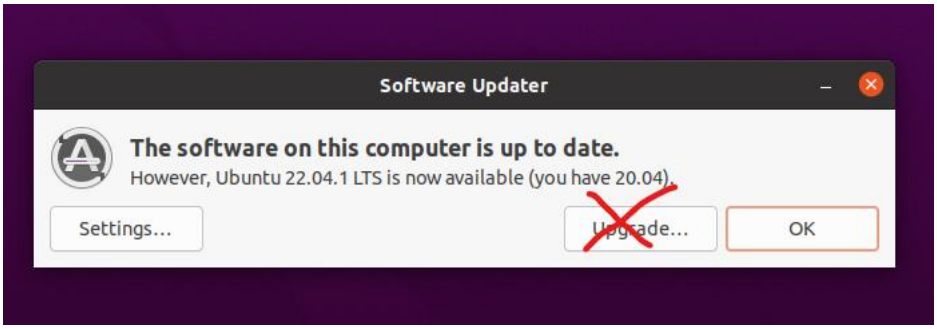
Graphical interface:

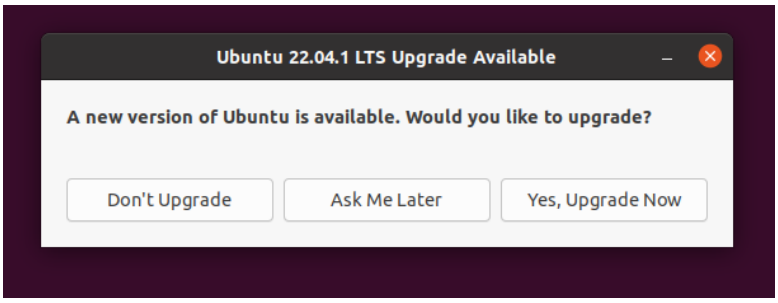


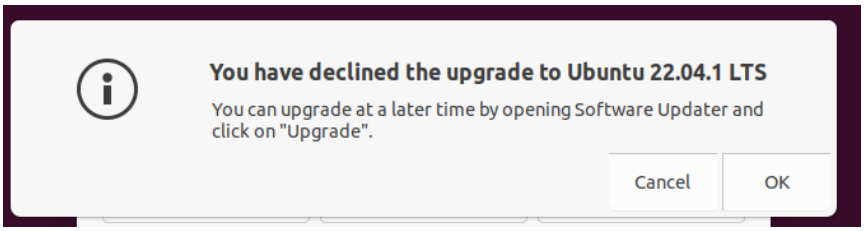
Next -> Next -> Done



Don’t upgrade to 22.04. DON’T.







-) Start game server for 3.2 version:

sudo su

cd /root/genshin/srv

chmod +x ./cli

chmod +x ./cli-default

-) Disable VM internet (keep local only):

For Desktop edition:

sudo nano /etc/netplan/01-network-manager-all.yaml

For Server edition:

sudo nano /etc/netplan/00-installer-config.yaml

Comment line (add #) with “gateway4: 192.168.200.2”:

 # gateway4: 192.168.200.2

Save using “ctrl + x” -> “Y” -> Enter

-) Apply the IP changes:

sudo netplan --debug apply

-) Try to ping google:

ping 8.8.8.8

You shouldn't receive any bytes!

You should get message like “ping: connect: network unreachable | error | etc”

If you can ping google – network isn’t disabled, try to recheck network settings file…

-) If you wish to start only important servers (set ~16GB RAM, your system should be 20GB+ RAM):

sudo ./cli start

Following server has been disable in cli file:

 # cd ../oaserver

 # chmod +x oaserver

 # nohup ./oaserver -i 9001.9.1.1 &

 # cd ../pathfindingserver

 # chmod +x pathfindingserver

 # nohup ./pathfindingserver -i 9001.8.1.1 &

 # cd ../multiserver

 # chmod +x multiserver

 # nohup ./multiserver -i 9001.7.1.1 &

 # cd ../tothemoonserver

 # chmod +x tothemoonserver

 # nohup ./tothemoonserver -i 9001.10.1.1 &

-) If you wish to start ALL servers (set ~24GB RAM to VM, your system should be 30GB+ RAM):

./cli-default start

You will see (**ERRORs in DataLoader and Crypto** is normal, it's OK, just dirty trick with Grasscutter SDKserver):

```
End of configuration.
6:57:38,634 |-INFO in ch.qos.logback.classic.joran.JoranConfigurator@65e2dbf3 -
Registering current configuration as safe fallback point

6:57:39 <ERROR:DataLoader> We were unable to locate your default data files.
6:57:39 <ERROR:Crypto> An error occurred while loading keys.
ava.lang.NullPointerException: null
        at java.base/java.util.Objects.requireNonNull(Objects.java:208)
        at emu.grasscutter.utils.Crypto.loadKeys(Crypto.java:43)
        at emu.grasscutter.Grasscutter.main(Grasscutter.java:81)
6:57:39 <INFO:Grasscutter> 正在启动 Grasscutter...
6:57:39 <INFO:Grasscutter> 游戏版本: 3.2.0
6:57:39 <INFO:Grasscutter> Grasscutter 版本: 1.0.0-dev-6cd3c228
6:57:40 <INFO:HttpServer> [Dispatch] Dispatch 服务器启动于 192.168.200.130:288
6:57:40 <INFO:Grasscutter> 加载完成! 输入 "help" 查看命令列表
7:00:25 <INFO:RegionHandler> [Dispatch] Client 192.168.200.1 request: query_re
on_list
7:00:33 <INFO:DefaultAuthenticators$PasswordAuthenticator> [Dispatch] 客户端 1
.168.200.1 已登录, UID 为 10001
7:00:33 <INFO:DispatchHandler> [Dispatch] 客户端 192.168.200.1 正在尝试登录
7:00:33 <INFO:DefaultAuthenticators$SessionKeyAuthenticator> [Dispatch] 客户端
92.168.200.1 成功交换 token
7:00:33 <INFO:DispatchHandler> [Dispatch] 客户端 192.168.200.1 正在尝试登录
ispatch>
```

-) In your GAME Replace the UserAssembly.dll in Genshin\GenshinImpact_Data\Native game folder with the one from "requirements.rar".

-) Choose: Fiddler or Cultivation. I prefer Fiddler.

-) For Fiddler use script "fiddlerScript.txt" from "requirenents.rar" – start Fiddler app, go to FIddlerScript (inside app), replace all script to text from "fiddlerScript.txt" and save, done

-) For Cultivation - https://github.com/Grasscutters/Cultivation/releases/

https://github.com/Grasscutters/Cultivation/releases/download/v1.0.10-alpha/Cultivation_1.0.10_x64_en-US.msi

Launch Cultivation, go to Options.

Set Game Install Path to your Genshin Game folder

Uncheck Automatically Path Metadata

Check Use Internal Proxy

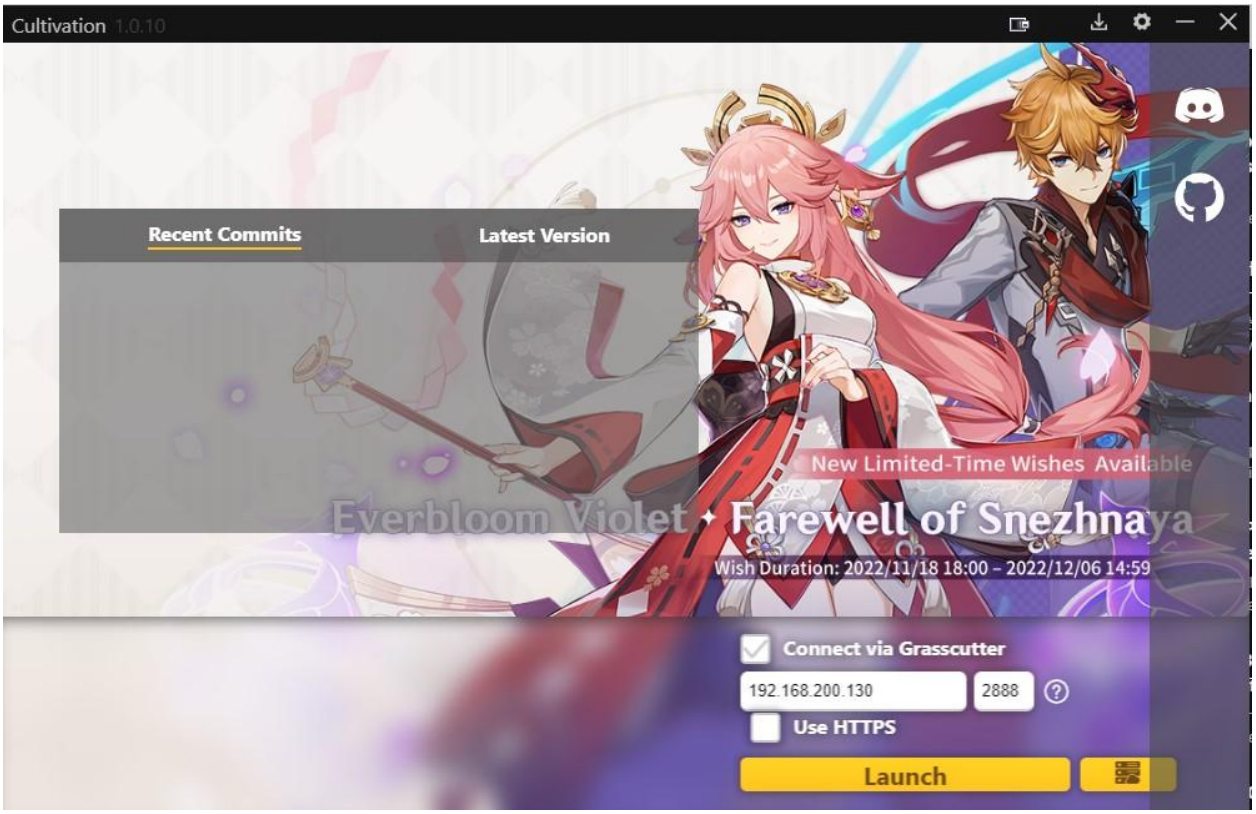
Check Connect via Grasscutter

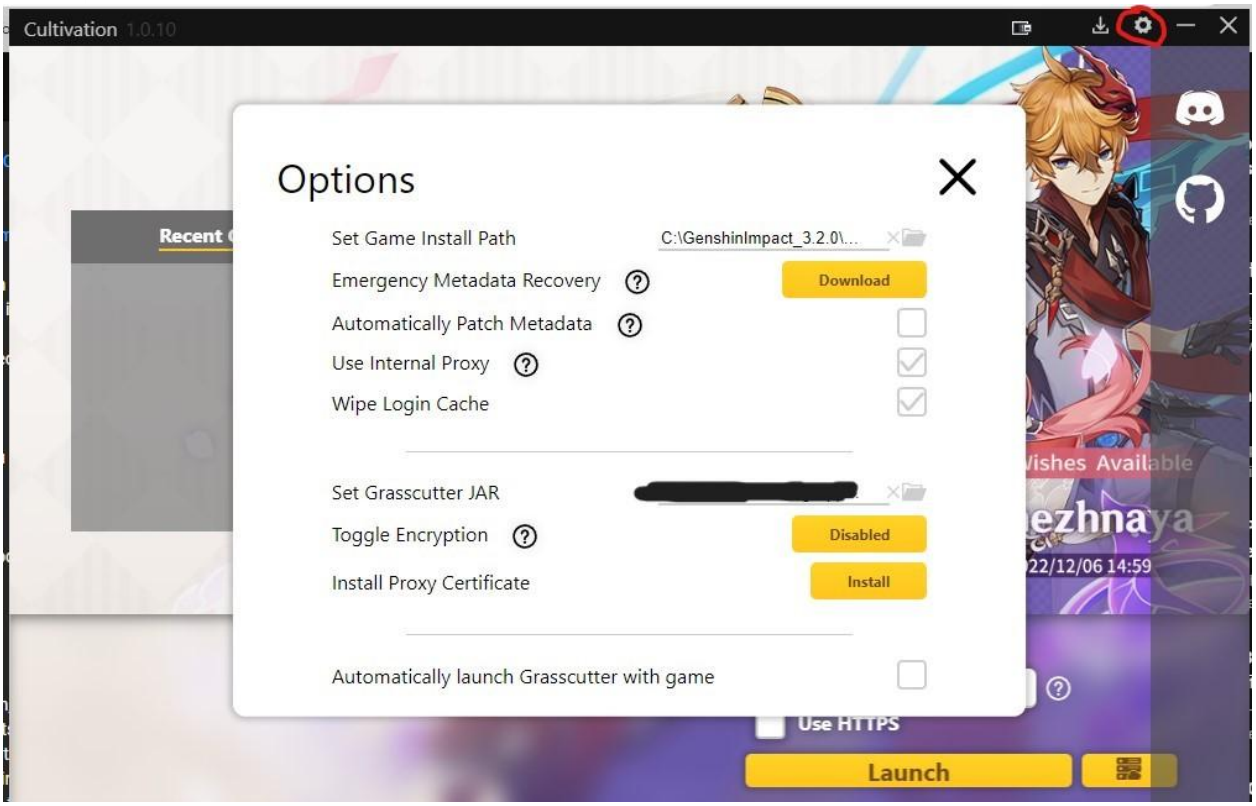
IP: 192.168.200.130

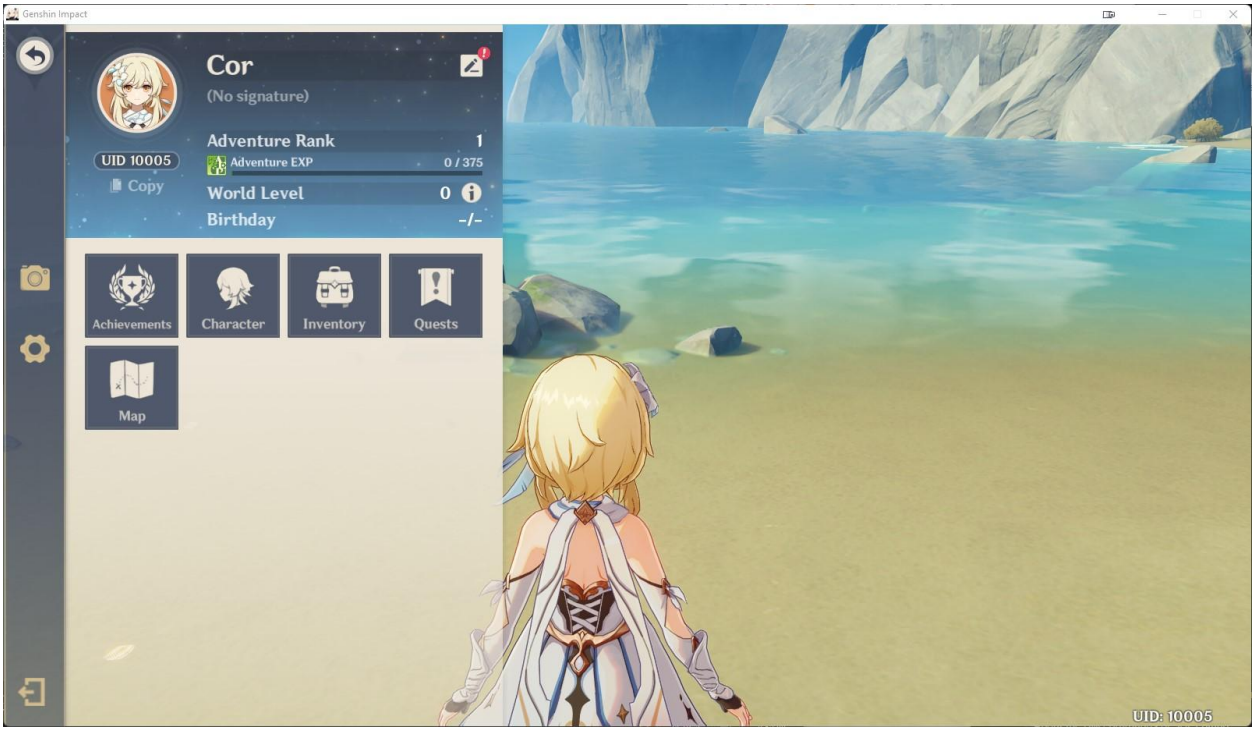
Port: 2888

Uncheck Use HTTPS

Click Launch to play.







PLEASE USE FIDDLER IF YOU CAN'T CONNECT USING CULTIVATION

-) You can write ANY login (just remember), server will create new if doesn't exist. OR you can use existing (not recommended):

username: Test

Password: 123456

-) Probably, you already have phpMyAdmin, go to

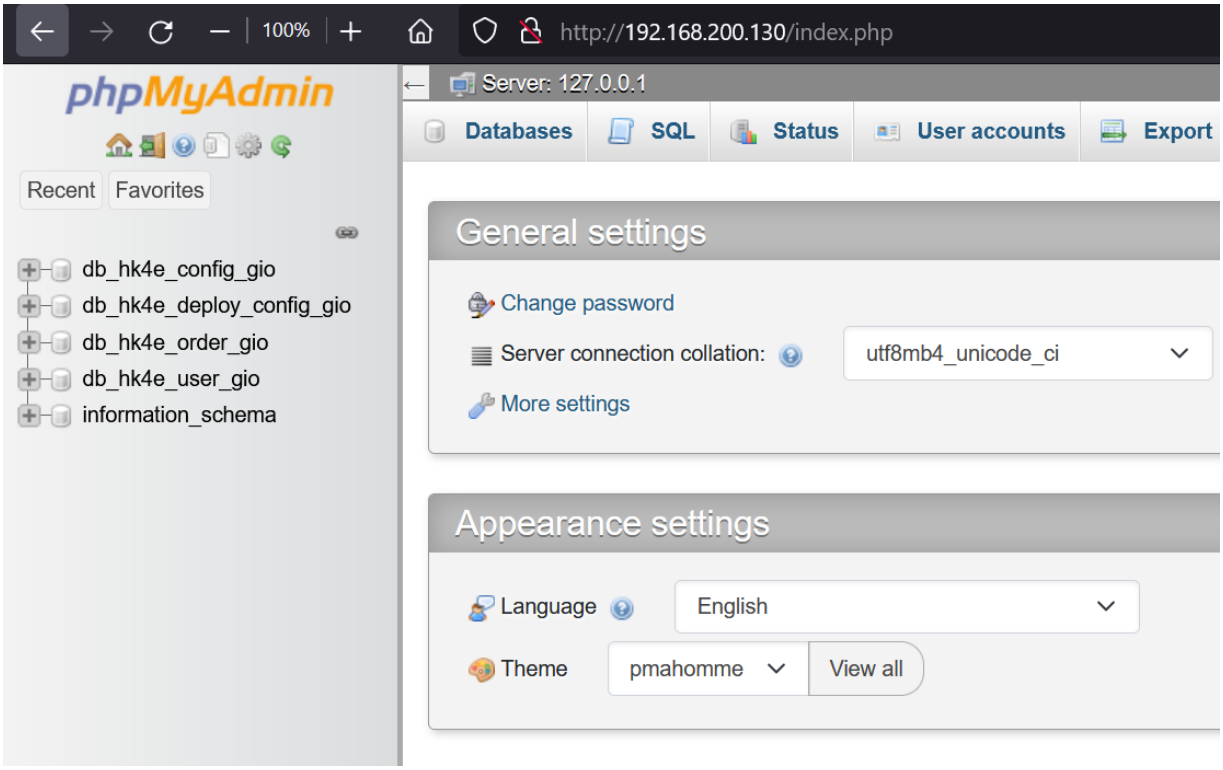
http://192.168.200.130 and check it. http! NOT https!

If it doesn't work, then install it inside the VM like this:

docker run -d --net=host --restart always -e PMA_HOST="127.0.0.1" -e PMA_ARBITRARY="1" --name phpmyadmin phpmyadmin/phpmyadmin

Use login: root and pass: f2c340a9-bf06-4345-9654-00b074b92fe8

OR login: work and pass: GenshinImpactOffline2022


100%
http://192.168.200.130/index.php
phpMyAdmin
Recent
Favorites
db_hk4e_config_gio
db_hk4e_deploy_config_gio
db_hk4e_order_gio
db_hk4e_user_gio
information_schema
Server: 127.0.0.1
Databases
SQL
Status
User accounts
Export
General settings
Change password
Server connection collation:
utf8mb4_unicode_ci
More settings
Appearance settings
Language
English
Theme
pmahomme
View all

Method 1. Run server in Linux native/WSL without Docker/Vmware/Virtualbox

-) Download non-configured server bins and data "as is" (I guess you already downloaded it from the **“Method 5. GIO Dockers”**

Notice: we'll be using a bundled dispatch server, but it can only serve HTTP. You will need to wrap HTTPS traffic from the client by using e.g. mitmproxy OR Fiddler (recommended).

-) Install VMWare / VirtualBox / WSL

-) Install Ubuntu 20.04 Desktop or Server edition (NOT Ubuntu 22.04, it's incompatible)

-) If you using WSL - skip this step. Open Vmware/VBox networks settings/configuration:

in vmware Edit -> Virtual Network Editor -> (on Bottom clock Change Settings with shield) ->

select YES -> click on VMnet with NAT -> in Subnet IP set: 192.168.200.0 and mask 255.255.255.0

-) If you using WSL - skip this step. For VBox/VMware create a new VM with 20 Gb of HDD space and 10 Gb or RAM (16 Gb recommended).

-) If you using WSL - skip this step. If you using Desktop Ubuntu 20.04 - install VMware/VBox Guest additions if it not installed yet (search in google how to install it).

-) Set static ip ONLY in VMware/Vbox (I dunno how to do it in WSL, you should find solution itself):

For Desktop edition:

```
sudo nano /etc/netplan/01-network-manager-all.yaml

# Let NetworkManager manage all devices on this system
network:
  version: 2
  renderer: NetworkManager
  ethernets:
    ens33:
```

```
    dhcp4: no
    addresses: [192.168.200.130/24]
    # you can delete this line to disable WAN internet
    # instead of using iptables
    gateway4: 192.168.200.2
    nameservers:
      addresses: [1.1.1.1,8.8.8.8]
```

For Server edition:

```
sudo nano /etc/netplan/00-installer-config.yaml
```

```
# This is the network config written by 'subiquity'
network:
  version: 2
  ethernets:
    ens33:
      dhcp4: no
      addresses: [192.168.200.130/24]
      # you can delete this line to disable WAN internet
      # instead of using iptables
      gateway4: 192.168.200.2
      nameservers:
        addresses: [1.1.1.1,8.8.8.8]
```

-) Apply the IP changes:

sudo netplan --debug apply

-) Enter this command to check your server ip again:

ip addr

It should show 192.168.200.130 now, and restart the VM. Check ip again. Check internet

using:

ping google.com

wget google.com

It should work, you should have internet access (we will disable it later)

-) Go to shared folder:

If you using WSL - you can see all disks using "cd /mnt/ && ls -la". And for example you can go to "cd /mnt/d/folder1/folder2".

If you using VMware/VBox with installed guest additions - create a shared folder (from VM settings) pointing to the desired location of the server files. Assign this folder to the VM. In vmware you can go to /mnt/hgfs to see mounted folders. In vbox it is a network drive

-) Copy original server bins (for example: server-3.2-3.4-original-bins.rar) to /root or /home/userName (don't use it "in place" on your NTFS folder instead of EXT4 Linux - otherwise you will get VERY poor performance)

-) use command "unrar" or "unzip" to extract original bins, for example:

sudo apt install unrar unzip

unrar server-3.2-bins.rar

-) Create "server" directory, create "tools/op_tools" subdirectories in "server"

-) Move all file and folders from "3.2-sql" (from **server-1.0-3.2-configs.rar** archive) into "server/tools/op_tools" subdirectory.

-) Move all files and folders from "welink_3.2_qa.7z" (from **server-1.0-4.0-data.rar**) into "server/data/" subdirectory.

-) Create a new empty "server/res/" directory at the root "server" folder (nearby server/data/, NOT inside server/data/ !!!). If no "res/" folder - Gameserver will fail

-) Copy provided version.txt (from "**server-requirements.rar**") file into the "server/data/" subdirectory too.

-) Create "conf/dev" in server/tools/ and put "dev_gio.yaml" (from "**server-requirements.rar**") into the directory "server/tools/op_tools/conf/dev".

-) Register this config in "server/tools/op_tools/conf/global.xml" by adding a line <Region name="dev_gio" conf_file="dev/dev_gio.yaml" region_id="9001"/>

-) Change line #418 in "tools/op_tools/bin/build_conf.py" from env="test" to env="dev". Comment by adding "#" before line or remove lines 162 and 196.

-) Change line #108 in "tools/op_tools/bin/server_parse.py" from env="test" to env="dev"

-) update_sql.py - in line 51 replace "sh sql_info_merge.sh " to "bash sql_info_merge.sh "

-) Comment line #173 with "dump_yaml(suffix_name, host, port_begin)" in ready_deploy_dev.py. Add new line #150 with an argument auth_plugin='mysql_native_password'

Must be like that:

```
mydb = mysql.connector.connect(
        host=mysqlInfo[0]['host'],
        user=mysqlInfo[0]['user'],
        passwd=mysqlInfo[0]['pwd'],
        auth_plugin='mysql_native_password'
    )
```

-) Add new line #154 with text:

```
mycursor.execute("CREATE DATABASE IF NOT EXISTS db_hk4e_deploy_config_{0}".format(suffix_name))
```

Next steps are preformed inside the VM

-) Set timezone to UTC using:

sudo dpkg-reconfigure tzdata - select "None of this" -> "UTC" -> OK

-) Run:

echo "$SHELL"

You will see /bin/bash if not - set bash as a default system-wide shell by running "dpkg-reconfigure dash"

-) Install **openssh-server** and configure it for passwordless login of your user (with a key file).

apt install ssh

You can also just type the password everytime scripts require it, but that's a ton of typing...

-) Install git

sudo apt install git

-) Install golang:

sudo apt install golang-go golang

If you fail with go --classic install, use another comand:

sudo snap install go --classic

-) Install mysql-server, libmysqlclient-dev, redis and gawk:

sudo apt install mysql-server libmysqlclient-dev redis gawk

-) Stop service

sudo service mysql stop

-) Remove default stored database (DON'T DO THAT IF YOU ALREADY HAVE IMPORTANT MYSQL DATA)

sudo rm -rf /var/lib/mysql

-) Create a new dir

sudo mkdir /var/lib/mysql

-) Change owner to mysql (user and group)

sudo chown -R mysql:mysql /var/lib/mysql

sudo chmod -R 750 /var/lib/mysql

-) Create MySQL initial data WITHOUT root password (we will set it later)

sudo mysqld --initialize-insecure

-) Start mysql servise:

sudo service mysql start

- Check mysql service:

sudo service mysql status

- Connect to DB:

mysql -u root

- You can set password for the root:

ALTER USER 'root'@'localhost' IDENTIFIED WITH mysql_native_password BY 'YOUR_PASSWORD_HERE';

- Create user "work":

create user 'work'@'localhost' identified by 'GenshinImpactOffline2022';

- Set password "GenshinImpactOffline2022" and auth method "mysql_native_password":

ALTER USER 'work'@'localhost' IDENTIFIED WITH mysql_native_password BY 'GenshinImpactOffline2022';

- Grant all priveleges to this user:

GRANT ALL PRIVILEGES ON *.* TO 'work'@'localhost' WITH GRANT OPTION;

-Flush:

FLUSH PRIVILEGES;

- Type:

Exit;

-) ONLY IF YOU HAVE TROUBLES:

sudo service mysql restart

-) Setup Redis: default password "GenshinImpactOffline2022". Increase databases number to 20.

sudo nano /etc/redis/redis.conf

We change in the file, looking for a column:

####### SECURITY ######

Use CTRL + W to open search, insert "# requirepass foobared" (ctrl + v or right click) after it insert "requirepass GenshinImpactOffline2022" without "#"

Like this:

requirepass foobared

requirepass GenshinImpactOffline2022

Search and replace:

databases 16 -> databases 20

Press CTRL + X, Press (Y) and Enter to save file

-) Restart Redis "/etc/init.d/redis-server restart"

-) Install python2.7 and python2.7-dev

sudo apt install -y python2-minimal python2.7-dev

ONLY if you can't call python2 via terminal - make a symbolic link /usr/bin/python2 and try again

-) Install python-pip and using pip2 install requests, Jinja2, enum and pyyaml packages.

Run only for Ubuntu 20.04:

cd ~ && wget https://bootstrap.pypa.io/pip/2.7/get-pip.py

python2 get-pip.py

(if you can't wget **get-pip.py –** try to use **get-pip.py** from the "**server-requirements.rar**")

Run only for Ubuntu 22.04:

sudo apt install python-pip

Run for both Ubuntu20.04/22.04:

pip2 install requests Jinja2 enum pyyaml

-) Download and unpack a very old version of **mysql-connector-python** package:

wget https://downloads.mysql.com/archives/get/p/29/file/mysql-connector-python-2.0.5.tar.gz

tar -xf mysql-connector-python-2.0.5.tar.gz

rm mysql-connector-python-2.0.5.tar.gz

OR if this file deleted use "**mysql-connector-python-2.0.5**" from the "**server-requirements.rar**"

-) Install this package:

cd mysql-connector-python-2.0.5

sudo python2 setup.py install

-) Clone "schemalex" into "tools/" directory. (After cloning you will get: tools/schemalex):

cd tools

git clone https://github.com/schemalex/schemalex

OR if this rep deleted use folder "**schemalex**" from the "**server-requirements.rar**"

-) Rename it to "tools/schemalex_tool" (because multiple server sources uses "./../../schemalex_tool/bin/schemalex")

mv schemalex/ schemalex_tool/

-) Go to:

cd schemalex_tool/

-) Install make:

sudo apt install make

-) Build it:

make build

-) Go to:

cd /artifacts/v0.1.1/schemalex_linux_amd64

-) Show by:

ls

You will see 3 bins: schemadiff, schemalex, schemalint

-) Move this 3 compiled binaries into "schemalex/bin" subfolder, create if necessary.

mkdir -p ../../../bin

mv schemadiff ../../../bin/

mv schemalex ../../../bin/

mv schemalint ../../../bin/

-) Start all services:

sudo service mysql start

sudo service redis-server start

sudo service apache2 start

(if you using WSL you need to start it every time manually!)

-) Install phpmyadmin:

sudo apt install phpmyadmin

(Mark using space button "configure apache2" and press tab to move to "OK" and enter)

You may see error, but its ok:

* Starting Apache httpd web server apache2 [Fri Dec 23 03:39:41.181001 2022] [core:warn] [pid 23148] (92)Protocol not available: AH00076: Failed to enable APR_TCP_DEFER_ACCEPT

-) Go to the DataBase admin panel:

http://192.168.200.15/phpmyadmin

Where 192.168.200.15 is YOUR_VM_IP (use "**ip addr**" OR "**sudo apt install -y net-tools && ifconfig**" to check IP)

-) In phpmyadmin select Import and select "db_hk4e_all.sql" from "requirements.rar/" and GO/OK/Import. After that you should see 4 new databases (on the left side, hk4e). Check it!

ONLY if you can't do it using phpmyadmin - do it via terminal:

sudo mysql -u work -p < db_hk4e_all.sql

Enter password: GenshinImpactOffline2022

PLEASE don't do that if you already did it using phpmyadmin

-) SKIP THIS STEP. SKIP. It's not a step, just info. How we got "db_hk4e_all.sql"? Do:

python2 ready_deploy_dev.py dev_gio

python2 update_sql.py dev_gio

sudo mysql db_hk4e_deploy_config_gio < tools/op_tools/sql_temp/0.1/hk4e_db_deploy_config.sql

Insert custom prepared data to:

db_hk4e_deploy_config_gio.t_area_config

db_hk4e_deploy_config_gio.t_bind_config

db_hk4e_deploy_config_gio.t_channel_id_config

db_hk4e_deploy_config_gio.t_client_channel_id_config

db_hk4e_deploy_config_gio.t_client_config

db_hk4e_deploy_config_gio.t_client_region_config

db_hk4e_deploy_config_gio.t_rsakey_config

db_hk4e_deploy_config_gio.t_sdk_config

db_hk4e_user_gio.t_player_uid

t_region_config

And modify client version, UID, RSA keys and all URLs.

DON'T DO THIS STEP. It's just info how "**db_hk4e_all.sql**" generated, you already HAVE it in "**server-requirements.rar**"

-) change in op_tools/conf/dev/dev_gio.yaml:

in "server_list" change: "host" and "inner_ip" and "outer_ip" to YOUR_VM_IP (use "ifconfig" to show networks)

-) Cd to:

cd ../../../../op_tools/bin/

-) Run:

bash deploy_dev dev_gio config

You will see a lot of "skip" if you are using WSL/Native linux or ssh if you are running it under remote machine

-) Run:

bash deploy_dev dev_gio reload

You will see:

[172.21.88.25] kill -s SIG.....

Now you should have configs and data ready.

-) Now you have a server structure and database structure with all the reqiured tables + data.

-) Copy all folders from "srv-bins/3.2_beta/hk4e_output/" to the "server/" directory.(nearby tools and data folders, a lot of folders already exists in "server/" but its ok to override them).

-) If you are using WSL - skip this step. Only for Ubuntu20.04 Native or VMware or VBox do:

cd server/lib/ && echo $(pwd) > /etc/ld.so.conf.d/genshin.conf && ldconfig

To register .so libs in the system (.so - like .dll for Linux)

-) install screen:

sudo apt install screen

-) If you using VMWare/VBOX - Before start you can disable WAN access (because this servers leak telemetry to the other real original servers) by removing

this line: "gateway4: 192.168.200.2" from your network settings

(see above) and restart network:

sudo netplan --debug apply

Try:

ping google.com

You shouldn't recieve any bytes!

If you using WSL OR still can ping google in VMWare/VBox - so... In this case - use script from "requirements/wan disable.sh"

If you need to use internet inside VM again: "requirements/wan enable.sh".

-) Run all screens:

sudo screen -dmS gateserver && sleep 1 && sudo screen -r gateserver -X stuff 'cd gateserver && ./gateserver -i 9001.1.1.1\n'

sudo screen -dmS gameserver && sleep 1 && sudo screen -r gameserver -X stuff 'cd gameserver && ./gameserver -i 9001.2.1.1\n'

sudo screen -dmS nodeserver && sleep 1 && sudo screen -r nodeserver -X stuff 'cd nodeserver && ./nodeserver -i 9001.3.1.1\n'

sudo screen -dmS dbgate && sleep 1 && sudo screen -r dbgate -X stuff 'cd dbgate && ./dbgate -i 9001.4.1.1\n'

sudo screen -dmS dispatch && sleep 1 && sudo screen -r dispatch -X stuff 'cd dispatch && ./dispatch -i 9001.5.1.1\n'

You may see: ==1055==WARNING: ASan doesn't fully support makecontext/swapcontext functions and may produce false positives in some cases!

It's OK

p.s. Also you can use "killall screen" to kill all screens and stop all servers if you want to stop

-) Show all screens (check all!):

sudo screen -list

-) [skip this step] Also you can attach to screen and see logs:

screen -r [name], for example: screen -r gateserver

To detach screen and keep running in background: Ctrl + A, Ctrl + D (press sequentially)

-) Wait for the server start and ready until 10-15 minutes if you are using HDD (it's not a joke).

You can open Task manager and look for nodeserver, gateserver, dbgate, dispatch, gameserver RAM and dick usage.

Also, you can install htop (task manager):

sudo apt install -y htop

And go to htop:

htop

Press F10 to exit

-) Go to SDK server emulator from "server-requirements.rar/SDK Servers/sdkserver.rar" and change in config.json 172.21.88.206 to YOUR VM IP (use "ifconfig" to show networks) and save:

```
"dispatch": {
    "regions": [
      {
        "Name": "os_usa",
        "Title": "Grasscutter",
        "type": "DEV_PUBLIC",
        "DispatchUrl": "http://172.21.88.206:20001/query_cur_region"
      }
    ],
```

-) Start SDKServer using "start sdkserver.bat"

-) Install FiddLer, open Fiddler, go to FiddlerScript and put data from "fiddlerScript.txt" (in "server-requirements.rar/Fiddler scripts") and save script, don't close Fiddler

-) Start game 3.2 (NOT 3.3), now you should be able to connect & play. Enjoy!

-) If you want to see open ports:

sudo lsof -i -P -n | grep LISTEN

-) If your vm ip changed, do again:

1. Change manually in op_tools/conf/dev/dev_gio.yaml:

in "server_list": "host" and "inner_ip" and "outer_ip" to YOUR_VM_IP (use "ifconfig" to show networks)

2. Rerun config:

cd ../../../../op_tools/bin/

bash deploy_dev dev_gio config

bash deploy_dev dev_gio reload

3. Run services:

sudo service mysql start

sudo service redis-server start

sudo service apache2 start

4. Rerun screens:

killall screen

sudo screen -dmS gateserver && sleep 1 && sudo screen -r gateserver -X stuff 'cd gateserver && ./gateserver -i 9001.1.1.1\n'

sudo screen -dmS gameserver && sleep 1 && sudo screen -r gameserver -X stuff 'cd gameserver && ./gameserver -i 9001.2.1.1\n'

sudo screen -dmS nodeserver && sleep 1 && sudo screen -r nodeserver -X stuff 'cd nodeserver && ./nodeserver -i 9001.3.1.1\n'

sudo screen -dmS dbgate && sleep 1 && sudo screen -r dbgate -X stuff 'cd dbgate && ./dbgate -i 9001.4.1.1\n'

sudo screen -dmS dispatch && sleep 1 && sudo screen -r dispatch -X stuff 'cd dispatch && ./dispatch -i 9001.5.1.1\n'

-) Read all **GM** commands in previous chapters

TCG (aka GCG) game from 3.3 support in 3.2 server

1) You should use protoshift (or viagenshin) to make 3.3 GIO support! You should play in 3.3 game client with 3.2 server (but with proto shifting feature 3.3 -> 3.2 in server side). Your game data in **Proto**s (3.3) should be converted on the fly to another Proto version (3.2).

2) Change this shit in template (you should know about this template from the **Method 1**):

conf/dev/dev_common.yaml:

line 814 (multiserver.services) add:

```
- name: 'gcgservice'
  type: 'GCGservice'
  thread_num: '1'
  stat_logs:
    - id: 1
      stat_log_level: 'GCG_STAT'
```

```
    thread_name: 'gcg_stat_log'
    stat_log_soft_limit: '20000'
    stat_log_limit: '50000'
    max_wait_time_us: '1000000'
    max_busy_time_us: '1000000'
  - id: 2
    stat_log_level: 'GCG_AI_STAT'
    thread_name: 'gcg_ai_stat_log'
    stat_log_soft_limit: '20000'
    stat_log_limit: '50000'
    max_wait_time_us: '1000000'
    max_busy_time_us: '1000000'
```

line 840 (multiserver.servers) add:

```
  - 'gcgservice'
```

3) conf_temp/template_multiserver.xml OR genshin/src/multiserver/conf:

line 21 (Root.LogConf.LogList):

```
<GCGStatFile path="log/gcg.stat.log" level="GCG_STAT GCG_AI_STAT" />
```

4) Reconfigure server configs (you should know how to do that from **Method 1**):

cd /op_tools/bin/

bash deploy_dev dev_gio config

bash deploy_dev dev_gio reload

Do you really read all pages?

If you really read all of it and understand all – you are skilled **GIO** player now, congrats!

Let’s play old events and CBT (closed beta tests)