

ICN – Project

Batch B – Group 17

Minecraft Game Server Hosting

and Administration via
Webpage





The Team

- **Parthvi Manoj**
 - CB.EN.U4AIE21143
- **Sakthi Swaroopan S**
 - CB.EN.U4AIE21159
- **Sanjay Chidambaram**
 - CB.EN.U4AIE21160
- **Taruneshwaran T**
 - CB.EN.U4AIE21170

What is

MINECRAFT



Minecraft

Minecraft is a popular sandbox video game known for its creative and open-ended gameplay. Players are immersed in a virtual world where they can explore, gather resources, build structures, and engage in various activities. The game offers a unique blend of adventure, survival, and limitless possibilities for players to shape and craft their own virtual realms.

Minecraft Multiplayer

- A Minecraft server is a software application that allows multiple players to connect and interact in a shared virtual world.
- It is hosted on a server platform, and players connect to the server using specific IP addresses and ports.
- Server administrators can customize and modify the server using plugins or mods to add new features, adjust gameplay mechanics, and manage server settings.





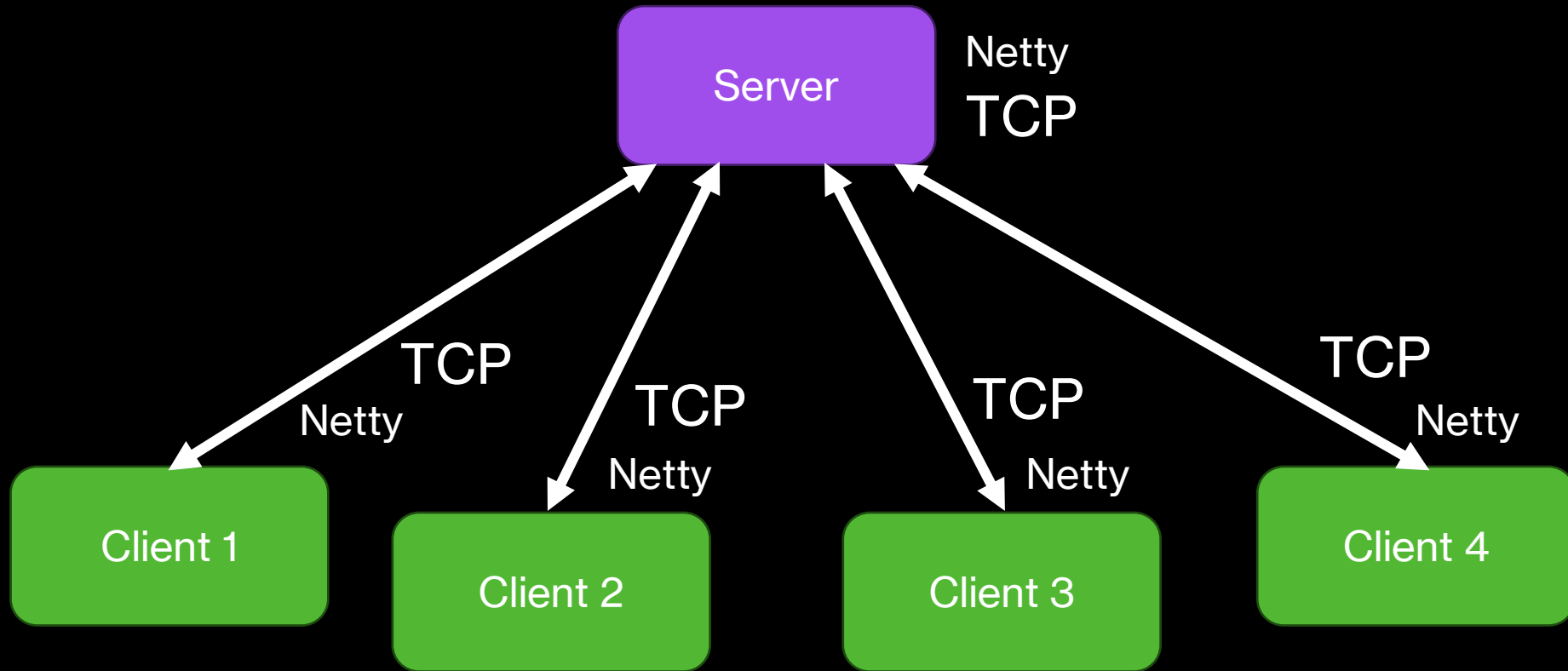
The Networking behind a Minecraft Server

- Minecraft servers use TCP/IP networking protocols.
- Players connect to servers using the Minecraft client software.
- Server administrators can configure and setup server properties in configuration files.
- Server software manages player authentication.
- Server plugins extend server functionality with additional features and commands.
- The server maintains synchronized game state across connected clients.
- Server hosting options include dedicated hardware, VPS, or third-party hosting services like aternos.org

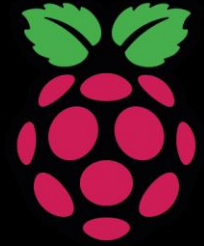
What is Netty?

- Netty is a asynchronous event-driven network application framework
- The way Netty works for Minecraft is that Minecraft Clients and servers have their own "Protocol" for each type of information that they'd like to send
- This is not the same as TCP/IP Protocol, This is more of a format for the packets
- For example, if the Minecraft Client wants to know the server information like number people online, it can send a packet called "info_players", which can be received and decoded by the server to then send a response
- We will be trying to capture some of these packets and analyze them using a Minecraft Client mod from modrinth.com

The Structure of Minecraft Server Communication



Raspberry Pi



- The Raspberry Pi is a series of small, single-board computers (SBCs) developed by the Raspberry Pi Foundation. These compact and affordable computers are designed to promote computer science education and enable users to learn programming and electronics.





Raspberry Pi Minecraft Server

- Raspberry Pi is a low-cost, credit card-sized computer used for hosting Minecraft servers.
- Minecraft server software can be installed and run on a Raspberry Pi.
- Consider the hardware limitations of the Raspberry Pi, which may affect server performance.
- Stable network connectivity is necessary for players to connect to the Raspberry Pi Minecraft server.

Details of our Raspberry Pi Board we used...

- We are using a Raspberry Pi 4, Model B with 4GB of RAM
- We are using a Sandisk 32 GB Micro SD Card with Ubuntu Server OS 22.04

```
prop@prop-pi:~$ neofetch
```

```
  .-/+00ssss00+/-.  
    `:+ssssssssssssss+:`  
  -+ssssssssssssssssyyss+-  
  .ossssssssssssssssdMMNyssso.  
  /ssssssssshdmmNNmmyNMMMMhsssss/  
  +ssssssshmydMMMMMMNdddyssssss+  
  /ssssssshNMMMyhhyyyyhmNMMNhsssss/  
  .sssssssdMMNhssssssssshNMMMdssssss.  
  +sssshhhyNMMNysssssssssyNMMMyssssss+  
  ossyNMMMNyMMhssssssssssshmmhssssssso  
  ossyNMMMNyMMhssssssssssshmmhssssssso  
  +sssshhhyNMMNysssssssssyNMMMyssssss+  
  .sssssssdMMNhssssssssshNMMMdssssss.  
  /ssssssshNMMMyhhyyyyhdNMMNhsssss/  
  +sssssssdmydMMMMMMNdddyssssss+  
  /ssssssssshdmNNNNmyNMMMMhsssss/  
  .ossssssssssssssssdMMNyssso.  
  -+ssssssssssssssssyyss+-  
    `:+ssssssssssssss+:`  
  .-/+00ssss00+/-.
```

```
prop@prop-pi
```


```
-----
```

```
OS: Ubuntu 22.04.2 LTS aarch64  
Host: Raspberry Pi 4 Model B Rev 1.5  
Kernel: 5.15.0-1032-raspi  
Uptime: 22 mins  
Packages: 783 (dpkg), 4 (snap)  
Shell: bash 5.1.16  
Terminal: /dev/pts/0  
CPU: BCM2835 (4) @ 1.800GHz  
Memory: 414MiB / 3790MiB
```



The Initial Idea

- Initially we had installed the Minecraft Server (Vanilla Version 1.19.4) without any server or web ui, that ran purely out of the terminal when we ran the server.jar file using the java command
- After getting the idea to have a webpage where admins could login and manage the server configuration, we tried giving a ssh connection from a webpage to the raspberry pi board directly, which ended up being not possible
- We had developed the following Abstract page for this idea:



Sign Up

It's free and only takes a minute

First Name

Last Name

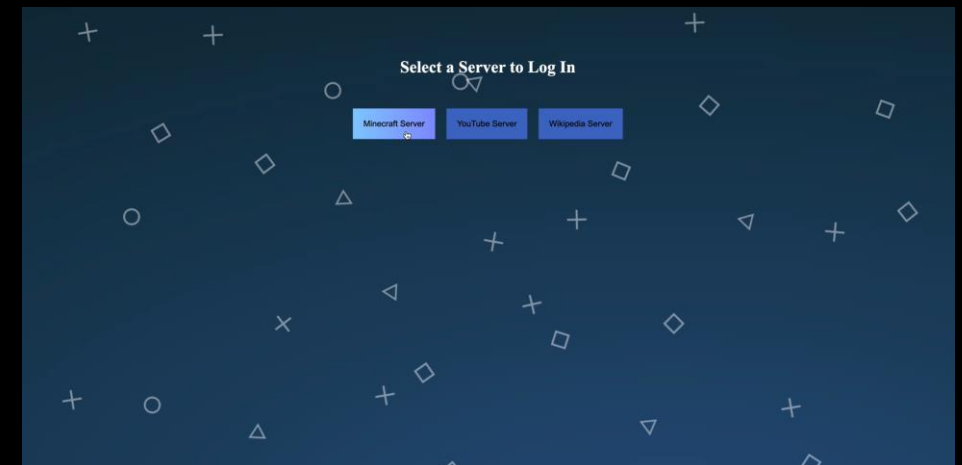
Email

Password

Confirm Password

By clicking the Sign Up button, you agree to our [Terms and Conditions](#) and [Privacy Policy](#)

Already have an account? [Log In](#)



How do we set it up in our Raspberry Pi?

- After a bit of searching, we discovered Pterodactyl Panels
- Pterodactyl is an open-source game server management panel designed to simplify the process of hosting and managing game servers. It provides a user-friendly web interface that allows server administrators to easily deploy, configure, and monitor game servers.
- Originally created specifically for Minecraft servers, Pterodactyl has expanded to support a wide range of popular games, including but not limited to Minecraft, Terraria, ARK: Survival Evolved, Rust, and Counter-Strike: Global Offensive (CS:GO).



Pterodactyl

Layers of Pterodactyl

Pterodactyl

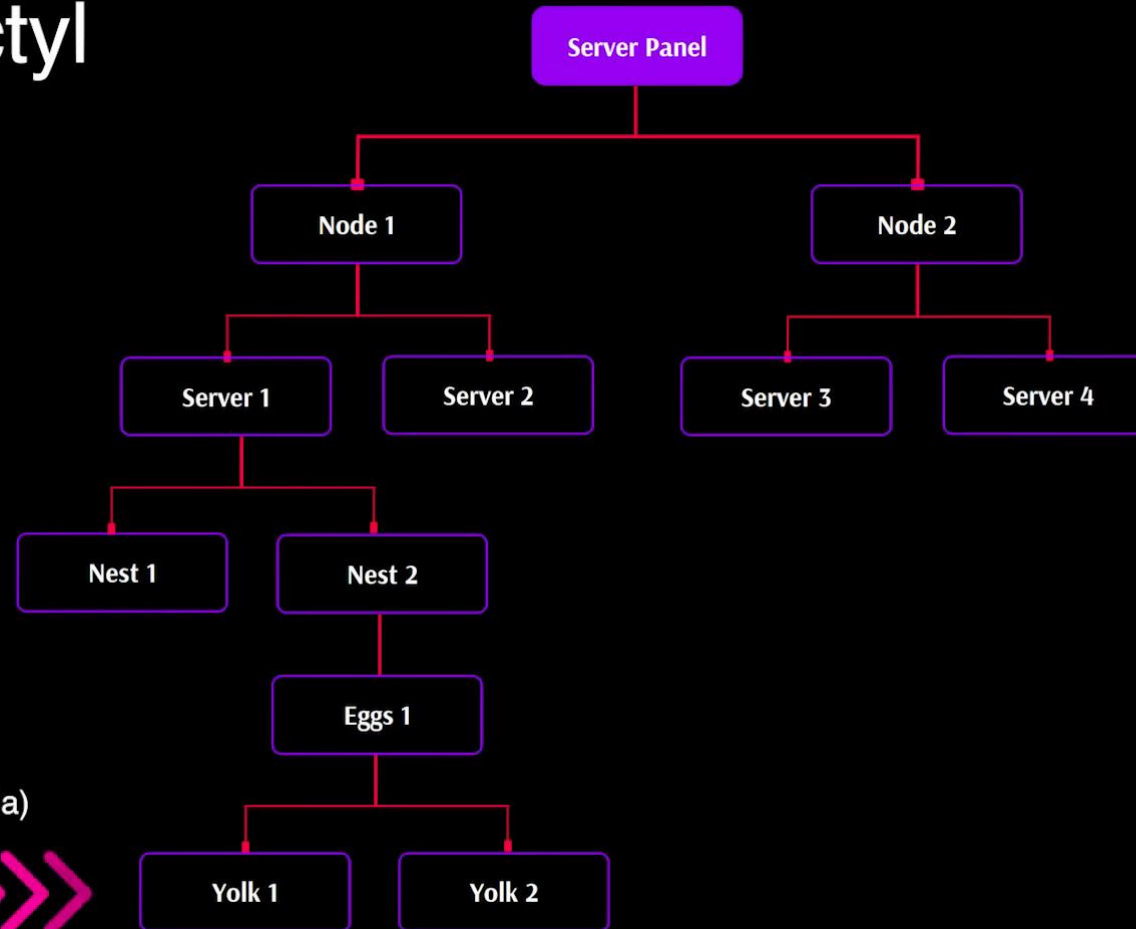
server
(runs wings agent)

games

nests
(Minecraft)

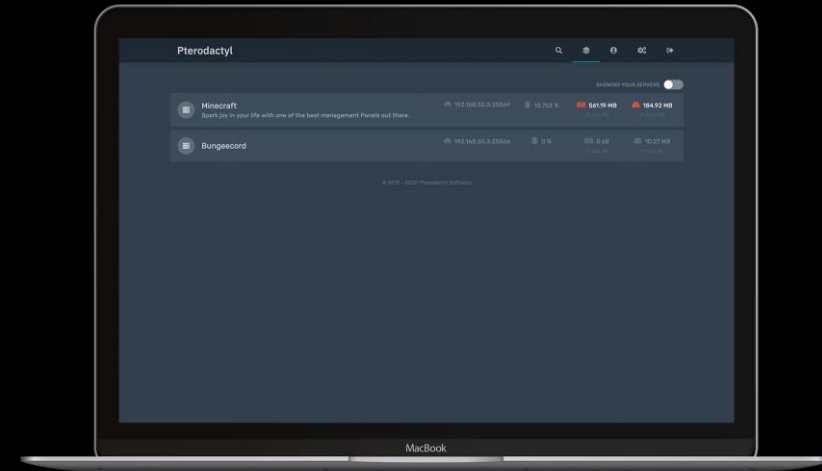
eggs
(Bedrock, Paper, Vanilla)

yolks >>>
(Docker Image)



Uses Of Pterodactyl

- **Server management:** Pterodactyl simplifies the creation, management, and monitoring of game servers through a user-friendly web interface.
- **Multi-game support:** It accommodates various game servers, allowing users to host multiple games on a single platform.
- **Resource allocation:** Pterodactyl enables users to allocate CPU, RAM, and disk space resources to individual game servers.
- We must understand that pterodactyl is just a panel for handling wings.

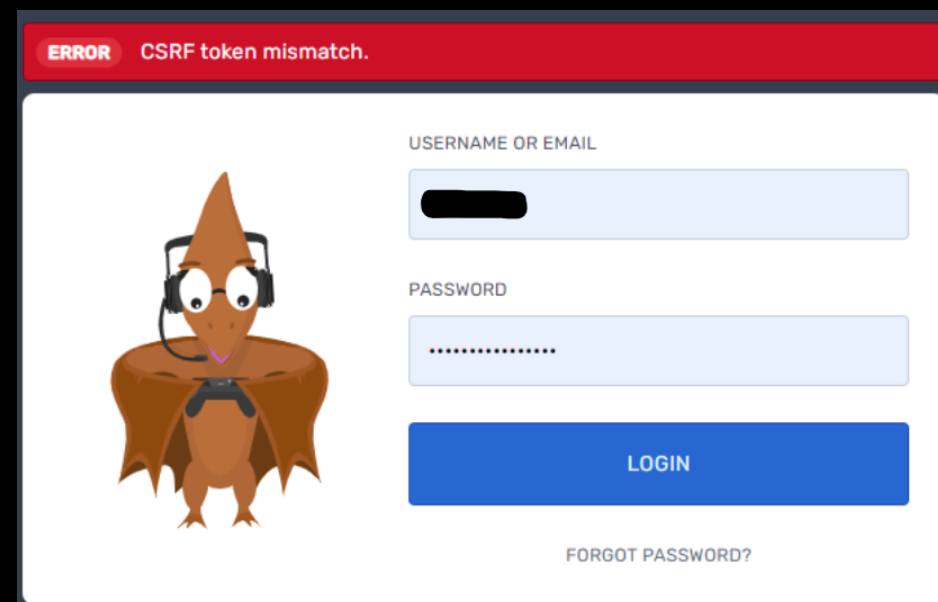


How do we set it up in our Raspberry Pi?

- We will be using the instructions from pterodactyl documentation
- We will be looking into the unique issues and challenges we faced during installation for this presentation
 - ❑ **IP Configuration Issue:** We initially tried to use *duckdns.org* and have a https page on a sub-domain under *duckdns.org* but it interfered with other networks and ended up exposing the router configuration page to the sub-domain instead.
 - We fixed this issue by opting to use http with our local ip addresses instead of *duckdns.org*
 - This reduced the time required for installation
 - If the server was to be set up permanently on a permanent network connection, then the issue could have been resolved by assigning a static ip in the router for the raspberry pi specifically

How do we set it up in our Raspberry Pi?

- ❑ **CSRF Token Mismatch Issue:** After setting up pterodactyl we had an issue while trying to login to the panel saying “CSRF Mismatch”, This issue occurs when the panel is creates a cookie to secure the connection on a unsecure (http) page.
 - This issue was resolved by disabling secure cookie tokens in the pterodactyl environment
 - If it wasn't for the previous issue we could have left the secure cookie tokens enabled



How do we set it up in our Raspberry Pi?

- ❑ **Control Group Error:** This was the hardest issue to solve since there was no clear way of understanding the error. We tried the following fixes which didn't work:
 - Installing Third Party open source Eggs for Minecraft
 - Recreating Nodes with various configuration
 - Recreating Servers multiple times
 - Reinstalling entire OS and Pterodactyl
- Finally we found where the issue was originating from, pterodactyl was trying to use "cgroups v2" using mounts that was meant for "cgroups v1"
- **The Solution** was to add `systemd.unified_cgroup_hierarchy=0` to the boot command line attributes which sets the "cgroups v1" as the first one to be used by processes instead of "cgroups v2"

```
[Pterodactyl Daemon]: finished pulling docker container image
container@pterodactyl:~$ Server marked as offline...
container@pterodactyl:~$ Error Event [ed585a8c-baf3-4972-aa8e-c376bbc19b58]: environment/docker: failed to start container: Error r
esponse from daemon: failed to create task for container: failed to create shim task: OCI runtime create failed: runc create fail
ed: unable to start container process: error during container init: error setting cgroup config for procHooks process: openat2 /s
ys/fs/cgroup/system.slice/docker-9d024106bd3091742901dcff52efec9d82798df8179bdc108cfe9d55f0feae6.scope/io.weight: no such file o
r directory: unknown
[Pterodactyl Daemon]: ----- Detected server process in a crashed state! -----
[Pterodactyl Daemon]: Exit code: 127
```

Setting up Node

Basic Details

Name

node2

Character limits: `a-zA-Z0-9_-` and `[Space]` (min 1, max 100 characters).

Description

sample node for presentation

Location

local

Node Visibility

☒ Public ☐ Private

By setting a node to `private` you will be denying the ability to auto-deploy to this node.

FQDN

Please enter domain name (e.g `node.example.com`) to be used for connecting to the daemon. An IP address may be used *only* if you are not using SSL for this node.

Communicate Over SSL

☐ Use SSL Connection ☒ Use HTTP Connection

In most cases you should select to use a SSL connection. If using an IP Address or you do not wish to use SSL at all, select a HTTP connection.

Behind Proxy

☒ Not Behind Proxy ☐ Behind Proxy

If you are running the daemon behind a proxy such as Cloudflare, select this to have the daemon skip looking for certificates on boot.

Configuration

Daemon Server File Directory

/var/lib/pterodactyl/volumes

Enter the directory where server files should be stored. If you use OVH you should check your partition scheme. You may need to use `/home/daemon-data` to have enough space.

Total Memory

3072

MiB

Memory Over-Allocation

0

%

Enter the total amount of memory available for new servers. If you would like to allow overallocation of memory enter the percentage that you want to allow. To disable checking for overallocation enter `-1` into the field. Entering `0` will prevent creating new servers if it would put the node over the limit.

Total Disk Space

5120

MiB

Disk Over-Allocation

20

%

Enter the total amount of disk space available for new servers. If you would like to allow overallocation of disk space enter the percentage that you want to allow. To disable checking for overallocation enter `-1` into the field. Entering `0` will prevent creating new servers if it would put the node over the limit.

Daemon Port

8080

Daemon SFTP Port

2022

The daemon runs its own SFTP management container and does not use the SSHd process on the main physical server. **Do not use the same port that you have assigned for your physical server's SSH process.** If you will be running the daemon behind CloudFlare® you should set the daemon port to `8443` to allow websocket proxying over SSL.

Create Node

← Using http here

Setting up the Server

Create Server

Add a new server to the panel.

Admin > Servers > Create Server


Core Details

Server Name

sample

Character limits: `a-z A-Z 0-9 _ - .` and `[Space]`.

Server Owner

 Sakthi Swaroopan S (proplay3rplayz@gmail.com)

Email address of the Server Owner.

Server Description

sample

A brief description of this server.

☒ Start Server when Installed

Allocation Management

Node

node1

The node which this server will be deployed to.

Default Allocation

0.0.0.0:22566

The main allocation that will be assigned to this server.

Additional Allocation(s)

Select Additional Allocations

Additional allocations to assign to this server on creation.

Resource Management

CPU Limit

0

%

If you do not want to limit CPU usage, set the value to `0`. To determine a value, take the number of threads and multiply it by 100. For example, on a quad core system without hyperthreading $(4 \times 100 = 400)$ there is `400%` available. To limit a server to using half of a single thread, you would set the value to `50`. To allow a server to use up to two threads, set the value to `200`.

Memory

3072

MiB

The maximum amount of memory allowed for this container. Setting this to `0` will allow unlimited memory in a container.

Disk Space

5120

MiB

This server will not be allowed to boot if it is using more than this amount of space. If a server goes over this limit while running it will be safely stopped and locked until enough space is available. Set to `0` to allow unlimited disk usage.

☐ Enable OOM Killer

Terminates the server if it breaches the memory limits. Enabling OOM killer may cause server processes to exit unexpectedly.

CPU Pinning

Advanced: Enter the specific CPU threads that this process can run on, or leave blank to allow all threads. This can be a single number, or a comma separated list. Example: `0`, `0-1,3`, or `0,1,3,4`.

Swap

0

MiB

Setting this to `0` will disable swap space on this server. Setting to `-1` will allow unlimited swap.

Block IO Weight

500

Advanced: The IO performance of this server relative to other *running* containers on the system. Value should be between `10` and `1000`. Please see [this documentation](#) for more information about it.

Setting up the Server

Nest Configuration

Nest

Minecraft

Select the Nest that this server will be grouped under.

Egg

Fabric

Select the Egg that will define how this server should operate.

☐ Skip Egg Install Script

If the selected Egg has an install script attached to it, the script will run during the install. If you would like to skip this step, check this box.

Docker Configuration

Docker Image

Java 17 (ghcr.io/pterodactyl/yolks.java_17)

Or enter a custom image...

This is the default Docker image that will be used to run this server. Select an image from the dropdown above, or enter a custom image in the text field above.

Startup Configuration

Startup Command

```
java -Xms128M -Xmx{{SERVER_MEMORY}}M -jar {{SERVER_JARFILE}}
```

The following data substitutes are available for the startup command: `{{SERVER_MEMORY}}`, `{{SERVER_IP}}`, and `{{SERVER_PORT}}`. They will be replaced with the allocated memory, server IP, and server port.

Service Variables

Server Jar File

server.jar

The name of the server jarfile to run the server with.

Access in Startup: `{{SERVER_JARFILE}}`

Validation Rules: `required|regex:/^((\w\d_-.)+)(\.-jar)?$/`

Fabric Version

latest

The version of Fabric to install.

Access in Startup: `{{FABRIC_VERSION}}`

Validation Rules: `required|string|between:3,15`

Minecraft Version

latest

The version of Minecraft to install. Use "latest" to install the latest version.

Access in Startup: `{{MC_VERSION}}`

Validation Rules: `required|string|between:3,15`

Fabric Loader Version

latest

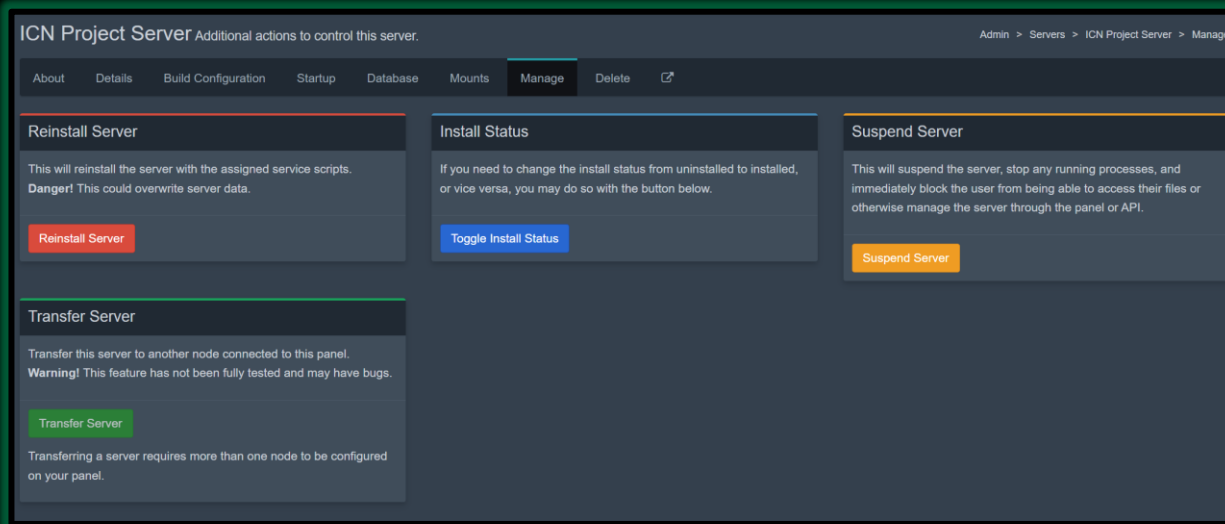
The version of Fabric Loader to install.

Access in Startup: `{{LOADER_VERSION}}`

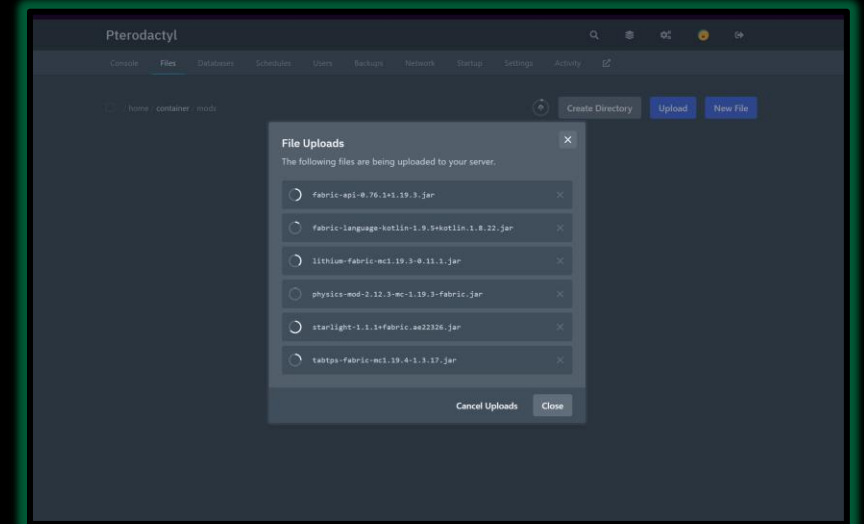
Validation Rules: `required|string|between:3,15`

/ home / container /		Create Directory	Upload	New File
	?		about 5 hours ago	...
	.fabric		about 5 hours ago	...
	libraries		about 5 hours ago	...
	logs		18 minutes ago	...
	mods		2 minutes ago	...
	versions		about 5 hours ago	...
	world		4 minutes ago	...
	banned-ips.json	2 Bytes	15 minutes ago	...
	banned-players.json	2 Bytes	15 minutes ago	...
	eula.txt	9 Bytes	about 1 hour ago	...
	fabric-installer.jar	179.62 KiB	Mar 1st, 2023 2:43PM	...
	fabric-server-launcher.properties	31 Bytes	about 5 hours ago	...
	minecraft-server.jar	44.98 MiB	about 5 hours ago	...
	ops.json	142 Bytes	15 minutes ago	...
	server.jar	713 Bytes	about 5 hours ago	...
	server.properties	1.25 KiB	less than a minute ago	...
	usercache.json	433 Bytes	10 minutes ago	...
	whitelist.json	2 Bytes	about 1 hour ago	...

Managing the Server

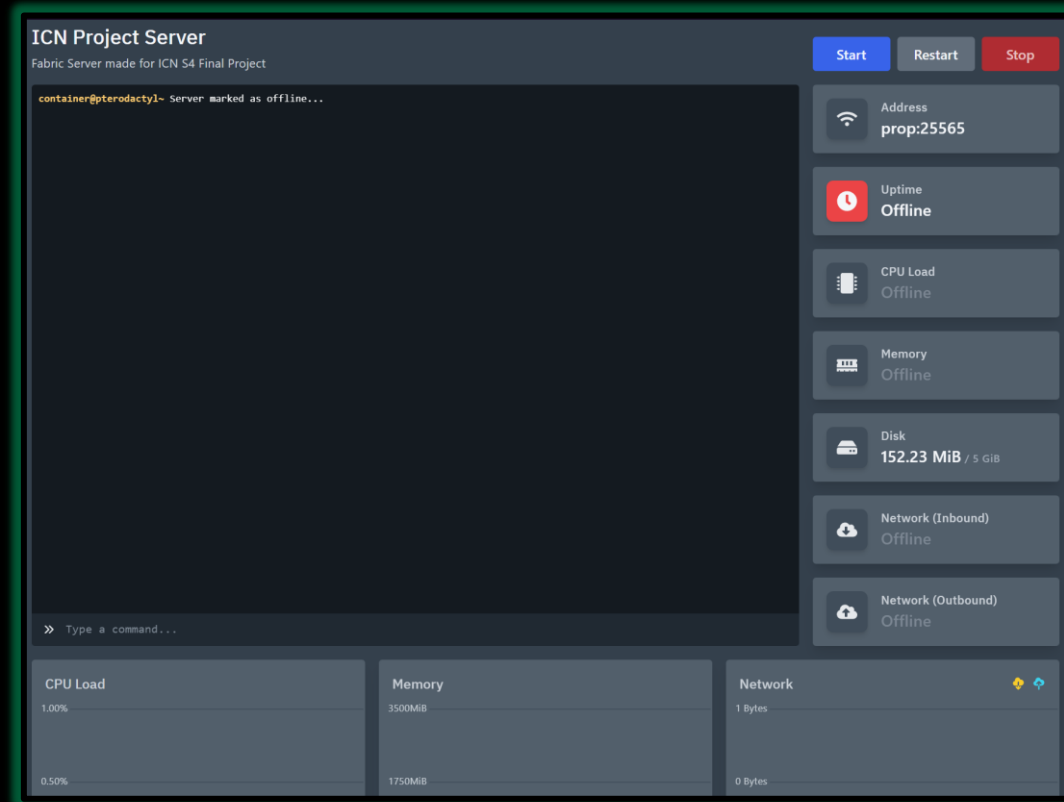


Easy UI for managing Server Status

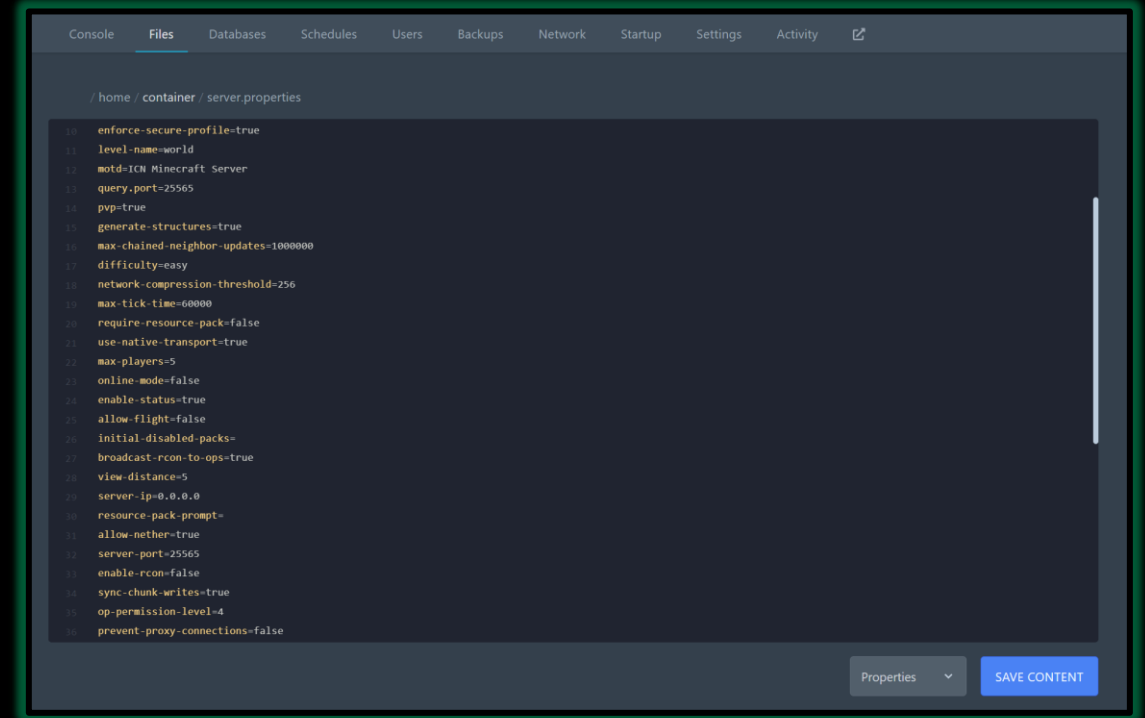


Uploading Mods and Plugins to the Server

Managing the Server



Console Access for the Server
and Server stats available at a glance



Updating Config files of the server directly
on the panel

Scope of Improvement

- Hardware **needs** to be upgraded since a Raspberry Pi has very limited RAM and Processing power to allocate for servers
- Pterodactyl works on both ARM and AMD chips, so using higher end server grade CPUs like Threadripper is supported
- Using optimization mods such as **Ferrit-Core** and **Starlight** improves performance and RAM usage of the server [implemented]
- Using Domain to expose the panel over a Domain so that admins can manage the server by accessing the panel from anywhere (requires static ip and continuous power for the pi)

Demonstration

ICN Project Server

Fabric Server made for ICN S4 Final Project

Start

Restart

Stop

[14:14:00] [Server thread/INFO]: tarunesh38 joined the game

[14:14:02] [Server thread/WARN]: Can't keep up! Is the server overloaded? Running 2040ms or 40 ticks behind

[14:14:09] [Server thread/INFO]: Parthvi[/192.168.150.179:49681] logged in with entity id 256 at (-637.5499369216184, 63.0, 1048.7115034593196)

[14:14:09] [Server thread/INFO]: Parthvi joined the game

[14:15:39] [Server thread/INFO]: tarunesh38 was shot by Skeleton

[14:15:59] [Server thread/INFO]: Parthvi was shot by Skeleton

time set day

[14:16:23] [Server thread/INFO]: Set the time to 1000

weather clear

[14:16:27] [Server thread/INFO]: Set the weather to clear

[14:16:35] [Server thread/INFO]: Parthvi was shot by Skeleton

kill @[type=player]

[14:16:39] [Server thread/INFO]: tarunesh38 was shot by Skeleton

[14:16:39] [Server thread/INFO]: Killed 299 entities

kill @[type=player]

[14:16:41] [Server thread/INFO]: Killed 385 entities

kill @[type=player]

[14:16:42] [Server thread/INFO]: Killed 94 entities

kill @[type=player]

[14:16:44] [Server thread/INFO]: Killed 91 entities

[14:17:25] [Server thread/INFO]: [Not Secure] <tarunesh38> Hello Folks

[14:17:52] [Server thread/INFO]: [Not Secure] <Parthvi> where is your house?

[14:17:59] [Server thread/INFO]: ProPlayerPlayz[/192.168.150.122:50257] logged in with entity id 1534 at (-640.589372973121, 63.0, 1039.5435321397533)

[14:18:00] [Server thread/INFO]: ProPlayerPlayz joined the game

[14:18:25] [Server thread/INFO]: [Not Secure] <ProPlayerPlayz> hey guys :D

[14:18:30] [Server thread/INFO]: [Not Secure] <tarunesh38> Hi Guyss!! Welcome to Minecraft\

[14:18:34] [Server thread/INFO]: [Not Secure] <Parthvi> Helloooooo!!!

[14:20:51] [Server thread/INFO]: [Not Secure] <ProPlayerPlayz> hi guysssss

[14:21:33] [Server thread/INFO]: [ProPlayerPlayz: Set the weather to clear]

[14:21:37] [Server thread/INFO]: [ProPlayerPlayz: Set own game mode to Creative Mode]

[14:22:47] [Server thread/INFO]: [ProPlayerPlayz: Teleported tarunesh38 to ProPlayerPlayz]

[14:23:02] [Server thread/INFO]: [Not Secure] <ProPlayerPlayz> Hello guys

[14:23:04] [Server thread/INFO]: [Not Secure] <tarunesh38> heiii

[14:23:08] [Server thread/INFO]: [Not Secure] <ProPlayerPlayz> where are we building house?

[14:23:09] [Server thread/INFO]: [Not Secure] <Parthvi> Hello!!!

[14:23:16] [Server thread/INFO]: [Not Secure] <tarunesh38> apls

Type a command...

CPU Load

70.00%

35.00%

Memory

3500MiB

1750MiB

Network

48.83 KB

24.41 KB

Address

prop:25565

Uptime

0h 14m 0s

CPU Load

50.28% / ∞

Memory

2.41 GiB / 3 GiB

Disk

161.8 MiB

Network (Inb)

5.81 MiB

Network (Out)

56.77 MiB

tarunesh38

Parthvi

<tarunesh38> heiii

<ProPlayerPlayz> where are we building house?

<Parthvi> Hello!!!



Thank
you