

Windows Dedicated Minecraft Server

Complete Setup Guide — Beginner Friendly

Engine: PaperMC 1.21.1 | Java: Eclipse Adoptium JDK 21 | OS: Windows (headless, SSH-managed)

No coding experience required — every command is copy-paste ready

■ Every command in this guide is given as its own isolated, single-line code block, designed to be copied and pasted exactly as shown into a PowerShell window. Do not retype commands manually — copy each box precisely, one at a time, to avoid errors. No command box contains more than one command, and no box relies on line-continuation characters.

Component	Spec	Role
RAM	12 GB total	8 GB locked to Minecraft, 3.25 GB for Windows, 0.75 GB buffer
CPU	4 physical cores	Forced into High-Performance mode — no power throttling
Storage	1 TB HDD	World pre-generation (Chunky) prevents slow random reads
GPU	None	Not needed — server runs fully headless with no display
Network	Wi-Fi (see Networking section)	Two options: Tailscale (easy, friends need app) or DuckDNS (open internet, no app needed)

✓ This server runs completely in the background. Once set up, you can close your SSH terminal and disconnect — the server keeps running 24/7 on its own via the Windows Service (NSSM) installed in Phase 3.

Before You Start — How to Use This Guide

- Open PowerShell as Administrator on your server (via SSH or directly).
- Each command below is its own grey box. Copy one box, paste it into PowerShell, press Enter.
- Run one command at a time — wait for it to finish before pasting the next one.
- If you see an error — read what it says. Most errors in this guide have been pre-diagnosed and fixed.
- Never retype commands manually — one wrong character breaks things.

PHASE 1 — DIRECTORY SETUP & ENGINE DOWNLOAD

Creates the folder structure and downloads the Minecraft server engine (PaperMC).

What this does:

- Creates C:\MinecraftServer\ with subfolders for backups and plugins
- Downloads PaperMC 1.21.1 — the optimised server engine this setup is built on

Run each of the following commands separately, one at a time:

```
Write-Host "Phase 1: Building infrastructure workspace..." -ForegroundColor Cyan
```

```
New-Item -ItemType Directory -Force -Path "C:\MinecraftServer",  
"C:\MinecraftServer\backups", "C:\MinecraftServer\plugins"
```

```
Set-Location "C:\MinecraftServer"
```

```
$PaperUri = "https://api.papermc.io/v2/projects/paper/versions/1.21.1/builds/130/downloads/paper-1.21.1-130.jar"
```

```
Invoke-WebRequest -Uri $PaperUri -OutFile "C:\MinecraftServer\paper.jar"
```

```
Write-Host "Phase 1 complete." -ForegroundColor Green
```

✓ Expected output: a green *Phase 1 complete.* message. No errors.

PHASE 2 — EULA ACCEPTANCE & SERVER CONFIGURATION

Accepts the Minecraft licence agreement and writes optimised server settings.

■ The original setup script ran the server interactively here, which caused it to hang waiting for manual EULA input. This version accepts the EULA automatically before launch and uses `--initSettings` to unpack config files without blocking.

What this does:

- Automatically accepts the Minecraft EULA (required to run any server)
- Unpacks the default config files and exits cleanly — no hanging
- Sets view distance to 6, simulation distance to 4, max players to 20
- Enables RCON (the correct way to send commands to the server remotely)

Block 2a — Accept EULA and unpack configs. Run each line separately:

```
Set-Content -Path "C:\MinecraftServer\eula.txt" -Value "eula=true"
```

```
java -Xms1G -Xmx1G -jar "C:\MinecraftServer\paper.jar" --initSettings
```

```
Start-Sleep -Seconds 5
```

```
Write-Host "Configs unpacked." -ForegroundColor Green
```

Block 2b — Apply optimised server settings. Run each command separately, in order:

```
$PropsPath = "C:\MinecraftServer\server.properties"
```

```
(Get-Content $PropsPath) | ForEach-Object { if ($_ -match "^view-distance=") {  
"view-distance=6" } elseif ($_ -match "^simulation-distance=") { "simulation-distance=4"  
} elseif ($_ -match "^max-players=") { "max-players=20" } elseif ($_ -match  
"^spawn-protection=") { "spawn-protection=0" } elseif ($_ -match "^enable-rcon=") {  
"enable-rcon=true" } elseif ($_ -match "^rcon.port=") { "rcon.port=25575" } elseif ($_  
-match "^rcon.password=") { "rcon.password=ChangeThisPassword" } elseif ($_ -match  
"^server-ip=") { "server-ip=" } else { $_ } } | Set-Content $PropsPath
```

```
Write-Host "Server properties written." -ForegroundColor Green
```

■ Remember to change the RCON password from ChangeThisPassword to something real. Instructions are in the Administration section.

PHASE 3 — WINDOWS SERVICE INSTALLATION (NSSM)

Registers the server as a Windows Service so it runs 24/7 without a terminal window open.

✓ This is the most important phase. After this step, the server starts automatically when the machine boots and keeps running even after you close your SSH session.

What this does:

- Downloads NSSM — a tool that wraps the Minecraft server as a proper Windows Service
- Registers the service with Aikar's optimised G1 garbage collection JVM flags
- Sets the service to auto-start on every boot
- Starts the service immediately

■ Original bug fixed: the original script broke `$AppParams` across multiple lines using `+` continuation, which PowerShell misread as arithmetic. Below, the entire argument string is built as a single-line command (using PowerShell's `-join` operator), so it can never be broken across lines by copy/paste.

Block 3a — Download and install NSSM. Run each command separately:

```
[Net.ServicePointManager]::SecurityProtocol = [Net.SecurityProtocolType]::Tls12
```

```
Invoke-WebRequest -Uri  
"https://github.com/kirillgreg/nssm-mirror/raw/master/nssm-2.24.zip" -OutFile  
"$env:TEMP\nssm.zip"
```

```
Expand-Archive -Path "$env:TEMP\nssm.zip" -DestinationPath "$env:TEMP\nssm_extracted"  
-Force
```

```
New-Item -ItemType Directory -Force -Path "C:\nssm"
```

```
Copy-Item "$env:TEMP\nssm_extracted\nssm-2.24\win64\nssm.exe" -Destination  
"C:\nssm\nssm.exe" -Force
```

```
Write-Host "NSSM installed." -ForegroundColor Green
```

Block 3b — Set the JVM arguments. This is a single command (do not split it):

```
$JavaPath = (Get-Command java).Source
```

```
$AppParams = @('-Xms8G', '-Xmx8G', '-XX:+UseG1GC', '-XX:+ParallelRefProcEnabled', '-XX:MaxGC  
PauseMillis=100', '-XX:+UnlockExperimentalVMOptions', '-XX:+DisableExplicitGC', '-XX:+Alway  
sPreTouch', '-XX:G1NewSizePercent=30', '-XX:G1MaxNewSizePercent=40', '-XX:G1HeapRegionSize=  
8M', '-XX:G1ReservePercent=20', '-XX:G1HeapWastePercent=5', '-XX:G1MixedGCCountTarget=4', '-  
XX:InitiatingHeapOccupancyPercent=15', '-XX:G1MixedGCLiveThresholdPercent=90', '-XX:G1RSet  
UpdatingPauseTimePercent=5', '-XX:SurvivorRatio=32', '-XX:+PerfDisableSharedMem', '-XX:MaxT  
enuringThreshold=1', '-jar', '"C:\MinecraftServer\paper.jar"', '--nogui') -join ' '
```

Block 3c — Register the service. Run each command separately, in order:

■ If you see 'The specified service already exists' on the first command — that is fine. Skip it and continue with the rest of the commands below.

```
& "C:\nssm\nssm.exe" install MinecraftServer "$JavaPath" $AppParams
```

```
& "C:\nssm\nssm.exe" set MinecraftServer AppDirectory "C:\MinecraftServer"
```

```
& "C:\nssm\nssm.exe" set MinecraftServer AppStdout "C:\MinecraftServer\logs\service.log"
```

```
& "C:\nssm\nssm.exe" set MinecraftServer AppStderr  
"C:\MinecraftServer\logs\service-error.log"
```

```
& "C:\nssm\nssm.exe" set MinecraftServer Application "$JavaPath"
```

```
& "C:\nssm\nssm.exe" set MinecraftServer AppParameters $AppParams
```

```
sc.exe config MinecraftServer start= auto
```

```
Start-Service -Name "MinecraftServer"
```

```
Start-Sleep -Seconds 15
```

```
Get-Service -Name "MinecraftServer"
```

✓ Expected output: Status = Running. Then confirm Minecraft actually loaded by checking the log with the command below. Look for: *Done (Xs)! For help, type "help"* — this confirms the world loaded successfully.

```
Get-Content "C:\MinecraftServer\logs\latest.log" -Tail 10
```

PHASE 4 — AUTOMATED ROLLING BACKUPS

Sets up a backup script that runs every 48 hours and keeps the last 10 backups automatically.

■ Original bug fixed: the original backup script only saved the 'world' folder. A complete Minecraft world is three folders: world, world_nether, and world_the_end. All three are now included.

Block 4a — Write the backup script. This writes a multi-line script FILE to disk (backup.ps1); it is not itself run line-by-line. Copy this entire box as one paste:

```
$BackupScriptContent = @'
$worlds = @("world", "world_nether", "world_the_end")
$backupRoot = "C:\MinecraftServer\backups"
$date = Get-Date -Format "yyyy-MM-dd_HH-mm"
$backupPath = "$backupRoot\backup_$date"
New-Item -ItemType Directory -Force -Path $backupPath | Out-Null
foreach ($w in $worlds) {
$src = "C:\MinecraftServer\$w"
if (Test-Path $src) { Copy-Item $src "$backupPath\$w" -Recurse -Force }
}
$backups = Get-ChildItem $backupRoot | Where-Object { $_.PSIsContainer } | Sort-Object
LastWriteTime
while ($backups.Count -gt 10) {
Remove-Item $backups[0].FullName -Recurse -Force
$backups = Get-ChildItem $backupRoot | Where-Object { $_.PSIsContainer } | Sort-Object
LastWriteTime
}
}'@
```

```
Set-Content -Path "C:\MinecraftServer\backup.ps1" -Value $BackupScriptContent
```

```
Write-Host "Backup script written." -ForegroundColor Green
```

Block 4b — Schedule it to run every 48 hours. Run each command separately:

```
$BTaskAction = New-ScheduledTaskAction -Execute "PowerShell.exe" -Argument
"-ExecutionPolicy Bypass -File C:\MinecraftServer\backup.ps1"
```

```
$BTaskTrigger = New-ScheduledTaskTrigger -Daily -At 4:00AM -DaysInterval 2
```

```
Register-ScheduledTask -TaskName "MinecraftServer_RollingBackup" -Action $BTaskAction
-Trigger $BTaskTrigger -User "SYSTEM" -RunLevel Highest -Force
```

```
Write-Host "Backup task registered." -ForegroundColor Green
```

✓ Expected output: a task definition block printed to screen, then the green confirmation. Backups will run automatically at 4:00 AM every other day.

PHASE 5 — FIREWALL RULES & POWER SETTINGS

Opens the network ports Minecraft needs and prevents Windows from sleeping.

Block 5a — Open firewall ports. Run each command separately:

```
New-NetFirewallRule -DisplayName "Minecraft Java 25565" -Direction Inbound -Protocol TCP  
-LocalPort 25565 -Action Allow
```

```
New-NetFirewallRule -DisplayName "Minecraft Geyser 19132" -Direction Inbound -Protocol  
UDP -LocalPort 19132 -Action Allow
```

```
New-NetFirewallRule -DisplayName "Minecraft RCON 25575" -Direction Inbound -Protocol TCP  
-LocalPort 25575 -Action Allow
```

```
Set-NetFirewallRule -DisplayName "Minecraft RCON 25575" -RemoteAddress LocalSubnet
```

```
Write-Host "Firewall rules set." -ForegroundColor Green
```

■ Port 25565 (TCP) = Java Edition players. Port 19132 (UDP) = Bedrock/mobile players via Geyser. Port 25575 (TCP) = RCON admin commands — restricted to local network only.

Block 5b — Disable sleep and set high-performance power mode. Run each command separately:

```
powercfg -change -standby-timeout-ac 0
```

```
powercfg -change -hibernate-timeout-ac 0
```

```
powercfg -setactive SCHEME_MIN
```

```
Write-Host "Power settings applied. Server will never sleep." -ForegroundColor Green
```

Plugin Installation

Plugins add features to your server. Run each download command separately, then restart the server once all four are in the plugins folder.

Plugin	What it does
Chunky	Pre-generates the world on your HDD so players never cause lag by exploring new areas
Geyser	Lets Bedrock Edition players (mobile, console, Windows 10) join your Java server
Floodgate	Works with Geyser to handle Bedrock player accounts
EssentialsX	Adds /home, /tp, /warp, and other standard admin commands

Download Chunky. Run each command separately:

```
Invoke-WebRequest -Uri  
"https://cdn.modrinth.com/data/fALzjump/versions/P3y2MXnd/Chunky-Bukkit-1.4.40.jar"  
-OutFile "C:\MinecraftServer\plugins\Chunky.jar"
```

```
Write-Host "Chunky downloaded." -ForegroundColor Green
```

Download Geyser. Run each command separately:

```
Invoke-WebRequest -Uri "https://download.geysermc.org/v2/projects/geyser/versions/latest  
/builds/latest/downloads/spigot" -OutFile "C:\MinecraftServer\plugins\Geyser.jar"
```

```
Write-Host "Geyser downloaded." -ForegroundColor Green
```

Download Floodgate. Run each command separately:

```
Invoke-WebRequest -Uri "https://download.geysermc.org/v2/projects/floodgate/versions/lat  
est/builds/latest/downloads/spigot" -OutFile "C:\MinecraftServer\plugins\Floodgate.jar"
```

```
Write-Host "Floodgate downloaded." -ForegroundColor Green
```

Download EssentialsX. Run each command separately:

```
Invoke-WebRequest -Uri "https://github.com/EssentialsX/Essentials/releases/download/2.20  
.1/EssentialsX-2.20.1.jar" -OutFile "C:\MinecraftServer\plugins\EssentialsX.jar"
```

```
Write-Host "EssentialsX downloaded." -ForegroundColor Green
```

Restart the server to load all plugins. Run each command separately:

```
Restart-Service -Name "MinecraftServer"
```



```
Start-Sleep -Seconds 20
```

```
Get-Content "C:\MinecraftServer\logs\latest.log" -Tail 15
```

✓ Look for lines like *[Geyser-Spigot] Loading ...* and *[Chunky] Loaded* in the log output. If any plugin shows an error, the log will tell you what went wrong.

Administration — Managing Your Running Server

Once the server is running you manage it through RCON — a remote command system built into Minecraft. You never need to keep a terminal window open. Just SSH in, run your command, and disconnect.

Step 1 — Download mcrcon (one-time setup). Run each command separately:

```
Invoke-WebRequest -Uri "https://github.com/Tiiffi/mcrcon/releases/download/v0.7.2/mcrcon-0.7.2-windows-x86-64.zip" -OutFile "$env:TEMP\mcrcon.zip"
```

```
Expand-Archive "$env:TEMP\mcrcon.zip" -DestinationPath "C:\nssm" -Force
```

```
Write-Host "mcrcon ready." -ForegroundColor Green
```

Step 2 — Change your RCON password (important). Run each command separately:

■ Do this before sharing your server with anyone. Replace YOUR_NEW_PASSWORD with something only you know.

```
Stop-Service -Name "MinecraftServer"
```

```
(Get-Content "C:\MinecraftServer\server.properties") -replace  
"rcon.password=ChangeThisPassword", "rcon.password=YOUR_NEW_PASSWORD" | Set-Content  
"C:\MinecraftServer\server.properties"
```

```
Start-Service -Name "MinecraftServer"
```

```
Write-Host "Password updated." -ForegroundColor Green
```

Step 3 — Give yourself operator (admin) status:

■ Original bug fixed: the old method wrote a fake UUID to ops.json which does not work. RCON automatically fetches your real Mojang UUID.

```
& "C:\nssm\mcrcon.exe" -H 127.0.0.1 -P 25575 -p "YOUR_NEW_PASSWORD" "op  
YourMinecraftUsername"
```

Other useful commands — each is its own isolated command:

Add a friend to the whitelist:

```
& "C:\nssm\mcrcon.exe" -H 127.0.0.1 -P 25575 -p "PASSWORD" "whitelist add FriendName"
```

Save the world manually:

```
& "C:\nssm\mcrcon.exe" -H 127.0.0.1 -P 25575 -p "PASSWORD" "save-all"
```

Check if server is running:

```
Get-Service -Name "MinecraftServer"
```

Read the last 10 log lines:

```
Get-Content "C:\MinecraftServer\logs\latest.log" -Tail 10
```

Restart the server:

```
Restart-Service -Name "MinecraftServer"
```

Stop the server:

```
Stop-Service -Name "MinecraftServer"
```

Start the server:

```
Start-Service -Name "MinecraftServer"
```

World Pre-Generation (Chunky)

Because your server uses a mechanical hard drive, new chunk generation causes lag spikes when players explore areas that haven't been visited before. Chunky pre-generates a large area around spawn so the disk work is done before players arrive.

■ Run this once after the server is fully set up. Pre-generating 5000 blocks takes time — the server stays online while it works. Check progress with the log command.

Run each command separately, in order:

```
& "C:\nssm\mcrcon.exe" -H 127.0.0.1 -P 25575 -p "YOUR_NEW_PASSWORD" "chunky spawn"
```

```
& "C:\nssm\mcrcon.exe" -H 127.0.0.1 -P 25575 -p "YOUR_NEW_PASSWORD" "chunky radius 5000"
```

```
& "C:\nssm\mcrcon.exe" -H 127.0.0.1 -P 25575 -p "YOUR_NEW_PASSWORD" "chunky start"
```

Check progress:

```
Get-Content "C:\MinecraftServer\logs\latest.log" -Tail 20
```

Chunky prints percentage progress to the log. You can disconnect and check back later.

Networking — How Friends Connect

This guide provides two networking options. Choose the one that fits your situation. Both work — they just involve different tradeoffs for your friends.

OPTION A — Tailscale (Recommended for privacy — friends need a free app)

Tailscale creates a private encrypted tunnel between your server and each friend's device. No port forwarding required, no public IP exposed, works from anywhere in the world. The tradeoff is that every friend must install the free Tailscale app before they can connect.

What	Value
Tailscale IP	100.94.235.87
Java Edition port	25565
Bedrock port	19132
RCON port	25575 (admin only — never share this)

How friends join via Tailscale:

- Friend downloads and installs Tailscale from tailscale.com — it is free.
- You invite them from your Tailscale admin panel at tailscale.com/admin → Invite users.
- Once they accept and are connected to your Tailscale network: Java Edition → Minecraft → Multiplayer → Add Server → Address: 100.94.235.87. Bedrock/mobile → Add server → Address: 100.94.235.87 → Port: 19132.

OPTION B — DuckDNS (Open internet — friends connect with no extra app)

DuckDNS gives you a free hostname like jbv.duckdns.org that always points to your current public IP. Friends connect directly over the internet — no Tailscale required. You must open your router's ports and run a small update script on your server so the hostname stays current if your IP ever changes.

■ Your public IP will be visible to anyone who resolves your DuckDNS hostname. Since this is friends-only this is acceptable, but never share your DuckDNS address publicly.

Step 1 — Set up DuckDNS (do this from any browser):

- Go to duckdns.org and log in with Google, GitHub, or Reddit.
- In the 'sub domain' box, type your chosen name (e.g. jbv) and click Add domain.
- Your new hostname is jbv.duckdns.org — copy your token from the top of the page.

Step 2 — Port forward on your router:

Open your router admin page (usually 192.168.1.1) and create two port forwarding rules pointing to your server's local IP. Find your local IP with this command:

```
Get-NetIPConfiguration -InterfaceAlias "Wi-Fi"
```

Protocol	External Port	Forward To	Purpose
TCP	25565	Your server local IP	Java Edition players

UDP	19132	Your server local IP	Bedrock/Geyser players
-----	-------	----------------------	------------------------

■ Do NOT forward port 25575 (RCON) — it has no brute-force protection and would give anyone full admin access.

Step 3 — Install the DuckDNS auto-update script on your server

This runs every 5 minutes and keeps your hostname pointing at your current IP. Replace YOUR_TOKEN_HERE and the subdomain "jbv" with your real values before running.

Create the update script. Run each command separately, in order:

```
$DuckToken = "YOUR_TOKEN_HERE"
```

```
$DuckSub = "jbv"
```

```
$DuckScriptContent = @"  
(Invoke-WebRequest -Uri  
"https://www.duckdns.org/update?domains=$DuckSub&token=$DuckToken&ip="  
-UseBasicParsing).Content  
"@
```

```
Set-Content -Path "C:\MinecraftServer\duckdns_update.ps1" -Value $DuckScriptContent
```

```
Write-Host "DuckDNS update script written." -ForegroundColor Green
```

Schedule it to run every 5 minutes. Run each command separately, in order:

```
$DAction = New-ScheduledTaskAction -Execute "PowerShell.exe" -Argument "-ExecutionPolicy  
Bypass -File C:\MinecraftServer\duckdns_update.ps1"
```

```
$DRepeat = New-ScheduledTaskTrigger -RepetitionInterval (New-TimeSpan -Minutes 5) -Once  
-At (Get-Date)
```

```
Register-ScheduledTask -TaskName "DuckDNS_Update" -Action $DAction -Trigger $DRepeat  
-User "SYSTEM" -RunLevel Highest -Force
```

```
Write-Host "DuckDNS updater scheduled." -ForegroundColor Green
```

Test it works right now. Run each command separately, in order (these can reuse the \$DuckToken / \$DuckSub variables set earlier in this session):

```
$DuckToken = "YOUR_TOKEN_HERE"
```

```
$DuckSub = "jbv"
```

```
(Invoke-WebRequest -Uri  
"https://www.duckdns.org/update?domains=$DuckSub&token=$DuckToken&ip="  
-UseBasicParsing).Content
```

✓ Expected output: OK — this means DuckDNS successfully updated. If you see KO, double-check your token and subdomain name.

How friends connect with DuckDNS:

- Java Edition: Minecraft → Multiplayer → Add Server → Address: jbv.duckdns.org
- Bedrock/mobile: Add server → Address: jbv.duckdns.org → Port: 19132
- No app download required — works exactly like any public server.

Check your active network adapters to confirm Wi-Fi is up:

```
Get-NetAdapter | Where-Object { $_.Status -eq "Up" }
```


Networking Troubleshooting — APIPA / 169.254.x.x Error

An IP starting with 169.254 means Windows could not get a valid address from your router (DHCP failure). The server will be unreachable. This is a network problem, not a Minecraft problem.

1. Check that your Wi-Fi adapter is connected and the network is reachable.
2. Run each of the following commands separately:

```
ipconfig /release
```

```
ipconfig /renew
```

3. If DHCP still fails, set a static IP — but check your adapter name first:

```
Get-NetIPConfiguration -InterfaceAlias "Wi-Fi"
```

Then disable DHCP and set a static address. Run each command separately, in order:

```
Set-NetIPInterface -InterfaceAlias "Wi-Fi" -Dhcp Disabled
```

```
New-NetIPAddress -InterfaceAlias "Wi-Fi" -IPAddress "192.168.X.X" -PrefixLength 24  
-DefaultGateway "192.168.X.1"
```

```
Set-DnsClientServerAddress -InterfaceAlias "Wi-Fi" -ServerAddresses "1.1.1.1"
```

■ Replace 192.168.X.X with an address from the Get-NetIPConfiguration output above. Using the wrong subnet will cut your SSH connection immediately.

Setup Completion Checklist

Phase	What was done	Status
Phase 1	Folders created, PaperMC downloaded	Done
Phase 2	EULA accepted, server.properties configured, RCON enabled	Done
Phase 3	NSSM Windows Service installed and running	Done
Phase 4	Backup script registered — runs every 48 h, keeps 10 copies	Done
Phase 5	Firewall ports open, RCON subnet-locked, sleep disabled	Done
Plugins	Chunky, Geyser, Floodgate, EssentialsX installed	Done
RCON	mcrcon downloaded, password changed, operator granted	Done
Chunky	World pre-generation started	Done
Network (A)	Tailscale IP confirmed, friends invited	Choose one
Network (B)	DuckDNS hostname set up, router port-forwarded, updater scheduled	Choose one

✓ Setup is complete. You can close your SSH terminal — the server runs on its own. It will restart automatically if the machine reboots. SSH back in any time to send commands or check the log.

Frequently Asked Questions

Do I need to keep a terminal open?

No. The NSSM Windows Service keeps the server running permanently in the background.

What happens if the machine reboots?

The server starts automatically on boot — no manual action needed.

How do I give myself admin/op?

```
& "C:\nssm\mcrcn.exe" -H 127.0.0.1 -P 25575 -p "PASSWORD" "op YourName"
```

How do friends connect (Tailscale)?

They install Tailscale, you invite them, they connect to 100.94.235.87 port 25565 (Java) or 19132 (Bedrock).

How do friends connect (DuckDNS)?

No app needed. They add jbv.duckdns.org in Minecraft multiplayer, port 25565 Java or 19132 Bedrock.

Why does my DuckDNS name not resolve?

Check the scheduled task ran by viewing the script and running it manually:

```
Get-Content C:\MinecraftServer\duckdns_update.ps1
```

If output is KO, check your token.

How many players can join?

Comfortably 15-20 on vanilla or light plugins. Not suitable for large modpacks like ATM or RLCraft.

Is it lag-optimised?

Yes. G1GC (Aikar's flags), fixed 8 GB heap (no resize stutters), 4-chunk simulation distance, HDD pre-generation via Chunky.

What version of Minecraft?

Java Edition 1.21.1 on PaperMC. Bedrock players can also join via the Geyser plugin.

What plugins are installed?

Chunky (world pre-gen), Geyser (Bedrock crossplay), Floodgate (Bedrock accounts), EssentialsX (admin tools).

How do I add more plugins?

Drop the .jar file into C:\MinecraftServer\plugins\ then run:

```
Restart-Service -Name "MinecraftServer"
```

Where are my backups?

C:\MinecraftServer\backups\ — a new folder is created every 48 hours. The oldest is deleted when count exceeds 10.

What if the server crashes?

Run the following and look for ERROR lines:

```
Get-Content "C:\MinecraftServer\logs\latest.log" -Tail 30
```