



# Designing a Minecraft Bedrock Server Hosting Website (Aternos-inspired)

To build a *Minecraft Bedrock Edition* server hosting site like Aternos, we must gather extensive information on Minecraft servers, hosting best practices, and user features. Minecraft is a massively popular sandbox game (over 350 million copies sold <sup>1</sup>) and Bedrock is its cross-platform edition (published by Xbox Game Studios) that runs on consoles, mobile and Windows <sup>1</sup>. Aternos' free hosting service offers a good model: "Minecraft servers. Free. Forever." with *your own personal server to play on with friends all day and night* <sup>2</sup>. Key principles from Aternos include free and unlimited play, easy one-click setup, DDoS protection, backups, and extensibility with plugins or add-ons <sup>2</sup> <sup>3</sup>. Our site must replicate and improve these features for Bedrock players.

## Core Hosting Features and Customization

- **Free and Unlimited Servers:** As with Aternos, the service should be free to use (at least initially) and allow large numbers of players. Aternos emphasizes "unlimited slots – play with as many friends as you want" <sup>4</sup>. We aim to support up to 1,000 concurrent players per server (or more) and handle hundreds of simultaneous server creations. The architecture must auto-scale (e.g. via cloud instances or containers) to avoid lag or downtime.
- **Server Software Options:** Offer the official *Bedrock Dedicated Server (BDS)* and a popular plugin framework like *PocketMine-MP*. Aternos' Bedrock section lists "Bedrock (the official server software with full survival support)" and "Pocketmine (plugins)" <sup>5</sup>. We should provide options for pure vanilla Bedrock worlds and modded add-ons. Bedrock supports "add-ons and behavior packs", so include easy upload of custom worlds, resource packs or add-ons <sup>6</sup>.
- **Custom Worlds and Modding:** Allow users to upload pre-built worlds or install community "add-ons & datapacks" for Bedrock <sup>6</sup>. We should support popular Bedrock extensions (e.g. via Nukkit or other platforms) if possible. The site should let users browse and add new content easily, similar to Aternos' drag-and-drop for add-ons <sup>6</sup>.
- **Server Management Console:** Implement a real-time web console and controls so users can start, stop or configure servers without delay. Aternos touts a "Real-time console" for easy management <sup>7</sup>. Permissions should be granular: e.g., server owner vs friends can have rights to *Start/Stop*, use the console, change settings, and manage players/files. The Aternos Access system shows all these permissions (start, console, plugins, world files, backups, etc.) <sup>8</sup>. Our site UI should mimic this, letting owners share access with friends (as moderators or members) who can do tasks like adding plugins or viewing logs <sup>9</sup> <sup>8</sup>.

## User Authentication & Access Control

- **Login and Subscription Gate:** Require users to sign in with a phone number (international support) or Gmail account. This can be implemented with modern auth (e.g. Firebase Auth or OAuth2 Google Sign-In). The user should verify their phone by SMS (global coverage) and/or use Google OAuth. After login, check YouTube subscription: only users subscribed to your channel (ID

UCPyiY8Heam2aPRP\_IA82irw ) are allowed to create servers. This would involve using the YouTube Data API to verify subscription status during account creation.

- **Roles and Permissions:** On the website, define roles like *Owner*, *Moderator*, *Member*, *Visitor*. The server owner (creator) has full control, while moderators or members (friends granted access) get limited controls. For example, Aternos lets you grant friends access with specific permissions (start/stop server, manage plugins/worlds, etc.) <sup>8</sup>. We should similarly allow the owner to add friends by username and assign them rights. Visitors (people browsing) should have no server control. All user actions should be logged or auditable.

## Server Creation Workflow

- **One-Click Setup:** The site should enable server creation in under a minute. After login and channel verification, the user clicks “Create Server”. We then spin up a new Bedrock server instance (Docker container or VM) pre-configured. Using pre-downloaded Bedrock Dedicated Server binaries and selected plugins (or a PocketMine instance) can speed deployment. A direct “Play” button should connect the user’s game client to the server. For example, show the server IP/port and a “Join” link; on mobile this could use the Minecraft URI scheme to launch the game and connect. This meets the user’s goal: “no waiting, 1 minute server that opens directly in Minecraft.”
- **Mobile & Desktop UI:** The website must be fully responsive. Aternos supports mobile and desktop players; ours should too. Use a modern frontend framework (React/Vue) to render forms and status pages that work on phones and PCs. Provide a language switcher for international users (English, Hindi, Spanish, etc.). Aternos itself supports 50+ languages in its UI <sup>10</sup>, so we should allow users to select their language on the site.

## Scalability, Performance, and Reliability

- **High Capacity:** The backend must handle many concurrent users (target at least 1000 simultaneous server creations or players). Use scalable infrastructure (cloud servers with load balancers). Aternos reports “1 Million players every day” <sup>11</sup>, so we should design for similar scale. Implement queueing if needed (but aim for immediate provisioning).
- **DDoS Protection:** Integrate network-level protection (e.g. Cloudflare Spectrum or a DDoS mitigation service) so that every Bedrock server is “fully DDoS protected” as Aternos promises <sup>12</sup>. This will keep gameplay smooth and uninterrupted.
- **Automatic Backups:** As on Aternos, offer free automatic backups of world data. Let users link a cloud drive (e.g. Google Drive) for storage, or keep server snapshots on our storage. This ensures users can restore worlds if needed <sup>13</sup>.

## User Experience and Smooth Operation

- **Easy Modding and Updates:** Provide simple controls to switch server software versions or add mods. Aternos lets users pick from versions and modpacks <sup>14</sup>. For Bedrock, support switching between vanilla and modded servers (e.g., official BDS vs. PocketMine) <sup>5</sup>. Ensure changes (like switching software) trigger a quick reinstall or reload.
- **Direct Minecraft Launch:** After server creation, prominently display connection details (IP address and port). For convenience, include a QR code or “Open in Minecraft” button. On Bedrock, users can add the server under the “Servers” tab with provided details. The goal is minimal friction so friends can join without manually entering info.

- **No Lag or Bugs:** Use monitoring and performance tuning to keep servers smooth. Since this is a heavy-coding project, emphasize rigorous testing: simulate many players, optimize tick rates, and test on both desktop and mobile clients. Use logs and alerts to catch issues early.

## Future Monetization and Support

- **Monetization:** Initially free, but plan for monetization after launch. Possible models: optional premium accounts (faster queue, extra RAM), non-intrusive ads on the site, or donations. For now, ensure the login system (phone/Gmail) is ready to tie into payment or subscription modules later.
- **Community and Support:** Provide a help center and maybe a Discord for users. Include terms of service noting the Bedrock license. Since this is not officially affiliated with Mojang/Microsoft, clarify that it uses *Mojang's official server software* (per license) <sup>15</sup>.

**In summary,** the website prompt should specify a free, Aternos-like Minecraft Bedrock hosting platform. It must allow users (upon YouTube subscription verification and login) to instantly create Bedrock servers (official or PocketMine) with unlimited players <sup>4</sup> <sup>5</sup>. Key features include multilingual UI, easy one-click server creation (join within 1 minute), roles (owner/moderator/member) with granular permissions <sup>9</sup> <sup>8</sup>, DDoS protection, automatic backups <sup>12</sup> <sup>13</sup>, and mobile/desktop support. All design and coding should aim for smooth, lag-free gameplay and a bug-free experience, leveraging cloud infrastructure for scalability. This thorough plan, drawn from Aternos' example and Minecraft Bedrock server requirements, should guide the website's creation.

**Sources:** Aternos official features and help pages <sup>2</sup> <sup>3</sup> <sup>5</sup> <sup>8</sup>, Minecraft Wikipedia <sup>1</sup>, and Bedrock server docs <sup>15</sup>.

---

<sup>1</sup> Minecraft - Wikipedia  
<https://en.wikipedia.org/wiki/Minecraft>

<sup>2</sup> <sup>3</sup> <sup>4</sup> <sup>5</sup> <sup>6</sup> <sup>7</sup> <sup>9</sup> <sup>10</sup> <sup>11</sup> <sup>12</sup> <sup>13</sup> <sup>14</sup> Aternos | Minecraft servers. Free. Forever.  
[https://aternos.org/en/----](https://aternos.org/en/)

<sup>8</sup> Share access – Aternos | Help Center  
<https://supportaternos.org/hc/en-us/articles/360026950952-Share-access>

<sup>15</sup> Bedrock Server Software | Bedrock Wiki  
<https://wiki.bedrock.dev/servers/server-software>