## Free Hosting Setup Using a Hybrid-Cloud Architecture (Step-by-Step)

It **is feasible** to host the jaegis-RAVERSE MCP server (a Python application) fully for free and keep it always online. This provided text document outlines a complete 100% free (for personal, non-commercial use) hybrid-cloud architecture.

This approach uses four free services together to run the Python application, provide a persistent database, keep the server from "sleeping," and give it a permanent public URL.

## 1. Set Up the Free "Always-On" Database (Aiven)

The server needs a persistent database and cache that don't sleep. Aiven's free tiers are perfect for this.

1. **Sign up** for a free Aiven account.
2. **Create Postgres DB:** On your Aiven dashboard, select **Create service**. Choose **Aiven for PostgreSQL** and select the **Free plan**. Name it (e.g., raverse-pg-db) and create it.
3. **Create Valkey (Redis) Cache:** Create another service. This time, choose **Aiven for Valkey** and select the **Free plan**. Name it (e.g., raverse-valkey-cache).
4. **Save Connection URIs:** For *both* running services, go to the **Connection information** tab, copy the **Service URI**, and save it in a secure note. You will need these "passwords" in the next step.

## 2. Deploy the Free Backend Server (Render)

This step hosts the main Python server code.

1. **Sign up** for a free Render account and connect your GitHub account.
2. **Fork the Repository:** On GitHub, create your own "fork" of the jaegis-raverse-mcp-server repository.
3. **Create Web Service:** On Render, click **New +** > **Web Service** and select your newly forked repository.
4. **Configure Settings:**
    - **Instance Type:** Choose **Free**.
    - **Runtime:** Select **Python 3**.
    - **Build Command:** pip install -r requirements.txt
    - **Start Command:** gunicorn main:app --workers 4 --worker-class uvicorn.workers.UvicornWorker --bind 0.0.0.0:$PORT
5. **Add Environment Variables:** Scroll down to the "Advanced" section. Add your two database keys from Aiven:
    - **Key:** DATABASE_URL | **Value:** [Paste your Aiven PostgreSQL Service URI]
    - **Key:** CACHE_URL | **Value:** [Paste your Aiven Valkey Service URI]
6. **Deploy:** Click **Create Web Service**. After it deploys, copy your new .onrender.com URL (e.g., https://raverse-mcp-server.onrender.com).

## 3. Make the Backend "Always-On" (UptimeRobot)

Render's free services sleep after 15 minutes. This step prevents that, making your server "always-on."

1. **Sign up** for a free UptimeRobot account.
2. **Create Monitor:** Click **+ Add New Monitor**.
   - **Monitor Type:** Select **HTTP(s)**.
   - **Friendly Name:** Give it a name (e.g., RAVERSE Keep-Alive).
   - **URL (or IP):** Paste your .onrender.com URL from Render.
   - **Monitoring Interval:** Set this to **5 minutes**.
3. **Save Monitor.** UptimeRobot will now ping your app every 5 minutes, which is faster than Render's 15-minute idle timer, keeping it permanently awake.

## 4. Create the Free Public URL (Cloudflare Workers)

This final step gives you a clean, permanent public URL that points to your Render app.

1. **Sign up** for a free Cloudflare account.
2. **Install Wrangler:** Install the Cloudflare command-line tool by running: npm install --global wrangler.
3. **Log In:** Run wrangler login to connect it to your account.
4. **Create Project:** Create a new folder, cd into it, and create two files:
   **index.js** (This is the proxy code. **Replace the BACKEND_URL** with your URL from Render.)

   ```
   const BACKEND_URL = "[https://raverse-mcp-server.onrender.com](https://raverse-mcp-server.onrender.com)"; // <-- PASTE YOUR RENDER URL HERE

   export default {
     async fetch(request, env, ctx) {
       const url = new URL(request.url);
       url.hostname = new URL(BACKEND_URL).hostname;
       const forwardedRequest = new Request(url, request);
       forwardedRequest.headers.set("X-Forwarded-By", "Cloudflare-Worker");
       return await fetch(forwardedRequest);
     },
   };
   ```

   **wrangler.toml** (This is the configuration file.)

   ```
   name = "raverse-mcp-proxy"
   main = "index.js"
   compatibility_date = "2025-10-29"
   ```

5. **Deploy:** From your terminal in that folder, run wrangler deploy. This will give you the final, permanent .workers.dev URL for your server.

## Summary Table

This architecture gives you a complete, always-on server for free.

| Component | Hosting Service | Cost | Purpose |
|---|---|---|---|
| **Database** (Postgres) | Aiven Free Tier | Free | Persistent vector data storage (always-on). |
| **Cache** (Valkey/Redis) | Aiven Free Tier | Free | Caching and high-speed data (always-on). |
| **Application** (Python) | Render Free Tier | Free | Runs the main jaegis-RAVERSE server code. |
| **Persistence** | UptimeRobot Free Tier | Free | Pings Render every 5 mins to prevent sleeping. |
| **Proxy / Public URL** | Cloudflare Workers | Free | Provides the final, permanent public-facing URL. |

## Important Notes

- This hybrid method provides a **completely free, always-on MCP server setup** for **personal, non-commercial use** (due to UptimeRobot's terms).
- This architecture runs the **full Python server**, not a "basic" one.
- This entire setup runs in the cloud, so it doesn't require your own hardware.