exploration

- Hostinger Cloud Hosting vs VPS for Docker Workloads

Hostinger Cloud Hosting vs VPS for Docker Workloads

Managed Cloud Hosting and **VPS (Virtual Private Server)** are quite different in terms of control and suitability for Docker:

- **Root Access and Software Installation:** Hostinger's VPS plans provide full **root access**, meaning you can install Docker and any software you need. In contrast, Hostinger's managed Cloud Hosting does **not** offer root access

  hostinger.com

  . This means on Cloud plans you cannot install custom server-wide software like Docker. In effect, Cloud Hosting is similar to an enhanced shared environment – you are limited to the provided stack (usually aimed at PHP/WordPress) with Hostinger's hPanel interface and cannot run arbitrary background services or daemons. For Docker containers, **VPS is the only viable choice** (since Docker requires root privileges to install and manage)

  hostinger.com

  .

- **Resource Allocation:** A Hostinger Cloud plan bundles a set amount of vCPU cores, RAM, and NVMe storage with a user-friendly managed setup. For example, **Cloud Startup** includes 2 CPU cores, 3 GB RAM, 100 GB NVMe storage

  hostinger.com

  hostinger.com

  . Higher Cloud tiers (Professional, Enterprise) increase these resources (up to 6 CPU / 12 GB on Enterprise

  hostinger.com

  hostinger.com

). Cloud plans also come with features like PHP worker limits (100–300) geared toward web hosting. By contrast, **VPS plans** dedicate the full claimed resources to you with no management layer overhead. Hostinger's "Docker VPS" plans (KVM-based) offer configurations like: **KVM 2** – 2 vCPU, 8 GB RAM, 100 GB NVMe SSD, 8 TB bandwidth

  hostinger.com

  ; **KVM 4** – 4 vCPU, 16 GB RAM, 200 GB NVMe, 16 TB bandwidth

hostinger.com
, etc. These resources are reserved entirely for your containers and are **“guaranteed”** to your VM
hostinger.com
. In practice, Cloud and VPS of similar specs should handle similar traffic, but VPS gives you more direct control over resource usage (and no neighbor interference beyond hypervisor limits).

- **Scalability:** Cloud Hosting is designed to easily handle typical website traffic spikes – for example, more PHP workers and built-in caching to absorb surges in pageviews
hostinger.com

hostinger.com
. It’s aimed at high-traffic sites without needing user management of the server. However, **Cloud plans cannot be scaled beyond the set tiers** – if you outgrow Enterprise, there is no larger Cloud tier (you’d consider VPS or dedicated). **VPS plans** can be scaled by upgrading to a larger plan (Hostinger allows you to upgrade the VPS plan via their panel, which usually requires a reboot but keeps your data). While scaling a VPS isn’t “instant” auto-scaling, it’s fairly straightforward to move up to more vCPU/RAM when needed. Notably, Hostinger’s VPS plans also advertise “instant scalability,” meaning you can upgrade your VPS resource allotment with minimal downtime when your projects grow
hostinger.com
.

- **Ease of Use vs Flexibility:** Cloud Hosting is fully managed – Hostinger handles the system setup, updates, security, etc., and you get a simpler hPanel interface for deploying websites. This is great if you *don’t* need custom apps. VPS, on the other hand, is self-managed (you are the admin). It’s suited for **tech-savvy users** who need custom setups
hostinger.com

hostinger.com
. Since you specifically need Docker and multiple custom services, you fall into the latter category by necessity – the managed environment’s simplicity would be a limitation for your use case.

Bottom Line: To run Docker containers on Hostinger, choose a **VPS plan**. Hostinger’s Cloud Hosting is not designed for running Docker or custom containerized apps (it lacks root access and Docker support)

hostinger.com
. The VPS gives you full control to install Docker Compose, run containers, and configure your stack as needed.

Recommended Plan: Based on the requirement of multiple containers and ~1000+ daily requests (which is roughly 30k+ monthly, a moderate load), at minimum consider the **Hostinger VPS KVM 2 plan** (2 vCPU, 8 GB RAM)

hostinger.com

. This “Most Popular” tier provides a healthy amount of RAM for running several containers concurrently and dedicated CPU cores for handling requests. 8 GB RAM is plenty for services like an n8n workflow automation container (which can use ~1-2 GB depending on workflows), a MySQL database, and a couple of Node/PHP apps, with headroom for caching or additional smaller services. The 2 vCPUs mean the server can multitask these containers without one blocking all CPU time.

If your workloads grow in complexity or number (for example, many more than 1000 requests/day or additional heavy services), you might upgrade to **KVM 4 (4 vCPU, 16 GB)**

hostinger.com

to ensure ample CPU for peak loads. But for starting out, KVM 2 strikes a good balance between cost and capability for a multi-container environment. Even **KVM 1 (1 vCPU, 4 GB)** could run a few containers (Docker itself is lightweight), but 4 GB RAM and a single core may become a bottleneck under concurrent usage or if any container is memory-hungry. So KVM 2 or higher is recommended for reliability.

Hostinger’s VPS plans come with a variety of one-click OS templates, including an **“Ubuntu with Docker”** template to get you started quickly

hostinger.com

hostinger.com

. Once your VPS is running, you’ll have full freedom to set up your Dockerized projects as needed.

Control Panels for Managing Multiple Dockerized Apps

Running a VPS gives you a blank slate – you can manage everything via the command line, but that can be complex with multiple apps/containers. This is where **server control panels** come in. There are two broad categories of panels to consider:

- **Container-focused DevOps Panels** – These are modern platforms that **natively use Docker** to deploy and isolate applications. They provide a UI to manage Docker containers, images, networks, and often support features like one-click app deployments, CI/CD, environment variables management, etc. Examples: **Coolify**, **Dokploy** (sometimes spelled Dokplay), **Cloudron**, **Easypanel**. These are great for running Node.js apps, databases, and more in containers with minimal sysadmin work. They essentially act as a lightweight PaaS on your VPS.
- **Traditional Hosting Control Panels** – These manage the server in a more classical way (installing web server, database, PHP, mail server, etc. on the OS). They are designed for hosting websites and may not inherently use Docker (though some support Docker in a limited way). Examples: **cPanel**, **Plesk**, **DirectAdmin**, **CyberPanel**, **HestiaCP**, **Webuzo**, **FASTPANEL**, **TinyCP**, **Webmin**, etc. These can run PHP/MySQL out of the box and some can proxy to Node.js apps or allow custom setups, but they are not primarily built for orchestrating Docker containers (with the notable exception of Plesk, which has a Docker extension).

Below we evaluate each panel you listed, focusing on their **Docker compatibility, support for Node.js (n8n) and other stacks, ease of use, resource footprint, and suitability for production**.

Container-Focused Panels (Docker-Native)

These panels are specifically designed to deploy and manage applications in Docker containers, which aligns perfectly with your use case. They abstract away a lot of Docker's complexity behind a GUI while still giving you the benefits of containerization (isolation of projects, easy deployments, etc.).

- **Coolify –** *Open-source Docker PaaS for apps and databases.* Coolify is a popular choice for self-hosting web apps in a Docker environment. It provides a friendly web UI to deploy applications from source (via Git) or from Docker images/compose files. It supports any tech stack (Node.js, Python, PHP, static sites, etc.) and even has one-click integrations for databases. A big advantage is its **ease of use** – Coolify's dashboard is very organized and beginner-friendly

  hostinger.com

  hostinger.com
  . It also comes with useful features out of the box: for example, **automatic SSL certificates** (Let's Encrypt) for your apps and a "no vendor lock-in" approach (you can stop using it and your containers remain usable)

  hostinger.com

hostinger.com

. Coolify's resource usage is relatively light; in testing, an idle Coolify installation used around 9% CPU and 41% of 8 GB RAM

hostinger.com

(meaning ~3.3 GB RAM, mainly because it itself runs in Docker). It requires at least **2 CPU cores, 2 GB RAM** to run smoothly

hostinger.com

. **Pros:** Easy setup and UI, good built-in security (auto SSL), supports deploying most applications and databases, active development. **Cons:** Designed for a single server (no built-in multi-server clustering), so scaling out beyond one VPS is not as seamless as some others. Also, while free and open-source, enterprise support is only via a paid managed service (optional)

hostinger.com

hostinger.com

. Overall, Coolify is an excellent choice to run multiple apps (including an n8n Docker container) easily on one Hostinger VPS. It's free to self-host (paid plan only if you want Coolify's team to host/manage it for you)

hostinger.com

.

- **Dokploy –** *Open-source Docker management panel focused on fast deployments.* Dokploy is very similar in goal to Coolify – it's essentially a lightweight self-hosted PaaS – but with some differences. It integrates deeply with **Docker Compose and Docker Swarm**, allowing you to manage multiple containers and even multiple nodes (servers) from one dashboard

  hostinger.com

  hostinger.com

  . Dokploy includes a centralized container management CLI built into its UI and uses **Traefik** as a built-in reverse proxy/load balancer

  hostinger.com

  . This gives it an edge for **microservices or multi-server setups** – it can orchestrate a cluster of Docker nodes and distribute apps across them, which is powerful if your needs grow beyond one VPS. It supports deploying any app via Docker, and you can use pre-built Compose templates (they mention one-click deploying of things like Plausible Analytics, etc.)

  hostinger.com

  . **Pros:** Supports horizontal scaling (cluster mode with Swarm), built-in Traefik proxy for routing and load-balancing, fine-grained user permission controls in the panel

  hostinger.com

hostinger.com
. **Cons:** Dokploy's UI, while minimalistic, is a bit less beginner-friendly (it lacks some of the out-of-box conveniences – for example, it does *not* auto-issue SSL certificates by default, requiring a manual step)

hostinger.com
. It also consumes similar resources to Coolify (needs ~2 GB RAM minimum). Dokploy is completely free (no paid tier at all)

hostinger.com

hostinger.com
. It can be a great choice if you anticipate possibly expanding to a multi-server cluster or want to use Docker Swarm. For a single-server use, Dokploy and Coolify are comparable, with Coolify offering a slightly more polished experience for beginners, and Dokploy offering more **advanced scalability** (multi-node) features

hostinger.com
.

- **Cloudron –** *App platform for self-hosting, using Docker under the hood.* Cloudron takes a slightly different approach: it provides an "app store" of popular self-hosted applications (such as n8n, Nextcloud, WordPress, GitLab, etc.) that you can install with one click. Each app is deployed in a Docker container managed by Cloudron

  hostinger.com

  hostinger.com
  . It handles updates for those apps, manages backups, DNS, user access control, and generally aims to make running your own server as easy as a SaaS platform. For your needs, Cloudron could be very convenient **if all the apps you want are available in its library** – notably, **n8n is available on Cloudron** and is installed in a secure, production-ready way with just a few clicks

  cloudron.io
  . Cloudron's interface is **extremely easy to use** – suitable even for non-technical users – and it takes care of things like automatic **encrypted backups to cloud storage** and domain DNS configuration for you

  hostinger.com

  hostinger.com
  . **Pros:** Easiest learning curve (app store model), good for managing multiple projects with different stacks because it provides a unified user management and permission system

  hostinger.com
  , and the apps are pre-configured for production (secure defaults). Also supports email server if needed and other services integrated. **Cons:** Less flexible for custom applications

– you are somewhat limited to either the apps they provide or you must package a custom app in Cloudron's format. If you have a custom Node/Python app not in the store, deploying it on Cloudron requires creating a custom Cloudron app (which means writing a Dockerfile and Cloudron-specific manifest). This is doable but not as straightforward as using a generic Docker panel. Also, Cloudron is **not fully free**: it has a **free tier (limits you to 2 apps)**, and to run unlimited apps you need a paid subscription (~$15/month)

hostinger.com

. Given you plan multiple containers, you'd likely need the paid plan for Cloudron or be limited to two containers. Cloudron is an excellent choice if you prefer a very polished solution and are okay with its licensing model (especially if you want hassle-free maintenance of things like n8n, which Cloudron will update for you automatically). For maximum flexibility and no cost, an open-source panel like Coolify/Dokploy might be preferable.

- **Easypanel –** *Developer-friendly Docker control panel.* Easypanel is another open-source project that serves as a web GUI for deploying apps in Docker containers. It is conceptually similar to CapRover or Dokku (other Docker PaaS tools), and in fact uses **Cloud Native Buildpacks** (the same technology as Heroku buildpacks) to deploy apps without you writing a Dockerfile

  hostinger.com

  . Easypanel emphasizes simplicity and zero-downtime deployments

  hostinger.com

  hostinger.com

  . You can either pull a Docker image or have it build your code via buildpacks. It also provides one-click templates for common apps like WordPress, etc., and an **in-browser SSH terminal** so you can run commands in the container or host easily

  hostinger.com

  hostinger.com

  . In terms of stack support, it can run anything that can be containerized; notably it supports MySQL, PostgreSQL, MongoDB, Redis, etc., as easily as any other app (similar to Coolify). **Pros:** Simple interface, **Docker integration is core** (it literally "has Docker integration to run applications as Docker images and supports Heroku Buildpacks" for many languages

  hostinger.com

  ), and it supports multiple databases and services out of the box

  hostinger.com

  hostinger.com

  . **Cons:** Easypanel's open-source version is fully functional but the project offers paid plans for "unlimited projects" – the free version might restrict the number of applications or projects you can deploy (for example, free tier might limit how many containers you

can run/manage, whereas paid unlocks unlimited)

hostinger.com

. According to Hostinger's info, paid plans range ~$9.9–23.9/month for Easypanel's hosted version

hostinger.com

, but if you self-host the open source version, those limits may or may not apply. It's worth checking the latest details on Easypanel's licensing. In general, Easypanel is a solid option if you want a CapRover-like experience with a nice UI. It's relatively new on the scene, so community support might be smaller than Coolify's.

Summary (Docker-Native Panels): All the above (Coolify, Dokploy, Cloudron, Easypanel) are capable of orchestrating multiple Docker containers and are suitable for Node.js (n8n) as well as MySQL/PHP/Python stacks. They make deploying multiple isolated projects much easier by providing a unified interface. **Coolify** and **Easypanel** prioritize ease-of-use and are great if you want a quick, free solution. **Dokploy** is excellent if you need multi-server scaling or are comfortable with a slightly more hands-on approach. **Cloudron** is the most user-friendly for predefined apps like n8n, but consider the app limit and pricing. All of these run on your VPS (Hostinger even provides one-click installation templates for many of them, e.g. "Coolify VPS hosting" templates

hostinger.com

hostinger.com

, Dokploy templates, etc.).

Traditional Hosting Panels (Non-Docker-Centric)

Now, let's consider the more traditional control panels. These typically install services directly on the VPS (LAMP/LEMP stack) and provide a web GUI for managing domains, databases, email, etc. They may not be built around Docker, but you could still run Docker containers on the same server in parallel – just without integration into the panel UI (except Plesk). Depending on your needs, you might choose to use such a panel for certain projects (e.g. hosting a PHP website) and run Docker containers separately for other apps. We'll look at their capabilities with respect to multi-project management, and whether they can accommodate Node.js or Docker in any way:

- **cPanel/WHM –** *Industry-leading web hosting panel (Linux only).* cPanel is very feature-rich for website hosting – it handles Apache or LiteSpeed web server, PHP, MySQL/MariaDB, email accounts, DNS, etc., all through an easy UI. It's battle-tested for

production hosting. **Pros:** Extremely familiar interface (if you've used shared hosting, you know cPanel), robust management of multiple websites and domains, and a large ecosystem of plugins. **Cons:** It is **not designed for Docker or container management** – you cannot deploy Docker containers via cPanel's interface. All your apps would typically be running on the host OS (for example, PHP sites on Apache). Running Node.js with cPanel is possible using additional modules (cPanel offers an "Application Manager" with Phusion Passenger to run Node/Python apps, but it may require CloudLinux or specific configuration). In practice, cPanel is best for PHP/HTML sites and maybe basic Node apps, but not something like managing an n8n container. Another con is **resource usage** – cPanel/WHM is relatively heavy, consuming a chunk of RAM/CPU for its services. Also, **cPanel is a paid license** (usually around $15–$20+ per month depending on license type)

hostinger.com

. Because cPanel doesn't align well with Docker, you'd only choose it if you intended to abandon the Docker approach for some services and run them traditionally. In your case, since Docker is a core requirement, cPanel would add little value (you'd likely end up managing Docker via CLI anyway). It's an excellent panel for traditional web hosting, but a mismatch for multi-container orchestration.

- **Plesk –** *Another top commercial panel, available on Linux and Windows.* Plesk is comparable to cPanel in scope (websites, DNS, mail, databases, etc.), but it has a more modern UI and a key advantage: **built-in Docker support**. Plesk has an extension called "Docker" that allows you to search and run Docker images from the Plesk interface

  hostinger.com

  . This means you could, for example, deploy a Dockerized application through Plesk and manage its container (start/stop) in the panel. This is quite useful if you want the convenience of a hosting panel but occasionally need to spin up a container for something. Plesk also has a **Node.js extension** which makes hosting Node apps (without Docker) easier – it can manage NPM packages, start the app with PM2, etc., all from the GUI. **Pros:** Versatile and user-friendly, with broad technology support – you can host PHP sites, static sites, Node.js apps, and even run some Docker containers side by side

  hostinger.com

  . Good security and WordPress management features (if you need them), and it supports multi-user scenarios nicely. **Cons:** Like cPanel, it is **commercial software** – licenses range roughly from ~$14 up to $60/month depending on edition

  hostinger.com

  . It also consumes significant resources for the panel itself (though generally Plesk is considered a bit lighter than cPanel). Plesk's Docker integration is more for convenience than high-volume container orchestration – it doesn't replace something like Coolify or Kubernetes. For instance, Plesk can run a Dockerized database or app for you, but it won't provide the advanced deployment pipelines that Docker-specific panels do. In summary, if you needed a traditional hosting environment for say a WordPress site **and** also wanted to deploy a few containers, Plesk could handle both in one interface. It's a solid option for a mixed workload. Just factor in the license cost and overhead.

- **DirectAdmin –** *Lightweight commercial panel.* DirectAdmin is another paid panel but generally cheaper and more lightweight. It provides the basics of web hosting management (websites, DNS, MySQL, email) without as much polish as cPanel/Plesk, but also with less bloat. **Pros:** Lower cost – licenses can be as low as a few dollars per month (it's known as a budget alternative)

  hostinger.com

  . It's also relatively light on resources and supports multiple levels (Admin/Reseller/User) for client management. **Cons:** No native support for Docker or Node.js out of the box – it's mainly LAMP stack oriented. The interface, while improved in recent versions, is not as modern or intuitive as Plesk's. DirectAdmin could be suitable if you have a lot of small PHP sites to host and want to conserve RAM, but want something more user-friendly than free panels. In your Docker-centric scenario, DirectAdmin doesn't offer integration – you'd be running containers manually alongside it. Thus, like cPanel, it might only be useful if you have some portion of your projects better served by traditional hosting.

- **CyberPanel –** *Free hosting panel powered by OpenLiteSpeed.* CyberPanel is an interesting option because it's **completely free and open source**. It's built around the OpenLiteSpeed web server (which is a high-performance HTTP server, especially good for PHP sites with LSCache). CyberPanel's feature set targets WordPress and PHP developers – it has one-click WordPress installs, LSCache integration, and a decent UI. **Pros:** Free license, fairly easy to install (Hostinger even offers "CyberPanel VPS" images

  hostinger.com

  ), and it's lightweight since OpenLiteSpeed is efficient. It also includes a mail server, DNS, FTP, and backup features. **Cons:** It doesn't natively manage Docker containers. You could install Docker on the same server, but CyberPanel won't be aware of those containers. Also, CyberPanel is somewhat niche – if you aren't specifically needing LiteSpeed, you might not benefit from its main advantage. It's great for hosting multiple PHP projects with low overhead (for example, many people use it to replace cPanel for WordPress hosting). For Node.js or Python, CyberPanel isn't ideal; you'd have to configure those manually (OpenLiteSpeed can proxy to a running Node.js app, but you set that up yourself). Overall, use CyberPanel if you want a free cPanel alternative for web hosting, but it won't help with n8n or Docker management (aside from co-existing on the server).

- **HestiaCP –** *Free, open-source control panel (fork of VestaCP).* HestiaCP provides the usual hosting features: it can host multiple websites (NGINX or Apache web server), manage databases, emails, DNS, and so on. It's relatively user-friendly for a free panel and has a modernized interface compared to its predecessor VestaCP

  hostinger.com

  . **Pros:** Completely free, supports multiple domains and users, and even has a one-click app installer for some common web apps. It's known for being **lightweight** and not consuming too many resources – good for smaller VPSs. **Cons:** No built-in Docker or container features. It's primarily for LAMP/LEMP stacks; Node.js support would require manual setup (you could configure a proxy to a Node app, but Hestia won't manage the

Node process for you). Hestia is great for simple web hosting needs, and because it's free, it's cost-effective. In the context of Docker: you could still run Docker containers on the server, but you'd manage them via command line or another tool, separately from Hestia. If your projects include some traditional websites that you want a panel for, HestiaCP is a nice free choice. But it doesn't add value for the Dockerized parts of your stack (they'll be invisible to it).

- **Webuzo –** *Single-server web hosting panel by Softaculous.* Webuzo can be thought of as Softaculous's version of cPanel but streamlined. It's often used by individuals or resellers to manage a single server with multiple apps. Webuzo supports Apache/NGINX, multiple PHP versions, MySQL, etc., and has the Softaculous one-click installer integrated (so you can install apps like WordPress, Joomla easily)

  hostinger.com

  hostinger.com
  . **Pros:** It offers a *comprehensive application stack* out of the box (Apache, NGINX, multiple database engines)

  hostinger.com
  , so it's ready for a variety of web applications. It also has an import tool to migrate data from other panels, which is convenient

  hostinger.com
  . **Cons:** Webuzo is *not free* for multi-user use – it has paid plans (around $2.5 to $25/month depending on version)

  hostinger.com
  . There may be a limited free version, but typically you'd purchase a license for full functionality. It doesn't have any special Docker integration. It's mainly useful if you want the Softaculous app catalog and a GUI to manage websites on your VPS. For your scenario, unless you specifically need Softaculous or prefer its interface, Webuzo doesn't offer benefits for Docker or custom Node/Python apps.

- **FASTPANEL –** *Free (closed-source) hosting panel.* FASTPANEL is provided free of charge (you just sign up for a license on their site)

  hostinger.com
  . It's aimed at being simple and fast, suitable for developers or startups who need to host sites. It manages NGINX or Apache, MySQL, email, etc., through a clean interface. **Pros:** Free and fairly lightweight. It has some nice features like a **web-based SSH terminal ("shell in a box")** for convenience

  hostinger.com
  , and the ability to set user roles per website (helpful if you host sites for clients)

  hostinger.com
  . FASTPANEL also includes handy tools like backup scheduling (it supports periodic backups to local or Dropbox)

hostinger.com

hostinger.com

and an integrated file manager and database manager. **Cons:** As with others, no direct Docker support. Also, since it's not open-source, you rely on the company for updates (but it has been stable so far). FASTPANEL is a good free option if you primarily host websites on the VPS and want a nice UI without cost. But if your goal is managing Docker containers, FASTPANEL itself won't help with that (you'd still be manually handling Docker). It could coexist – for example, you might host your PHP sites via FASTPANEL and separately run n8n in a Docker container – but there's no integration between the two.

- **TinyCP –** *Minimalist control panel.* TinyCP is a lesser-known, lightweight panel for Linux. It covers basic tasks: managing web servers (Apache or Nginx), databases, Samba shares, FTP, etc., through a very simple interface. **Pros:** Very small footprint – TinyCP is known to use extremely low RAM/CPU, which is ideal for tiny VPS instances or if you want virtually no overhead. It's also free. **Cons:** The project is not as active or widely adopted as others, meaning documentation and community help are limited. Feature-wise, it's more limited – for instance, it may not have one-click installers or advanced features that other panels offer. **Docker support:** There is no built-in support; in fact, there have been user requests to add Docker, but it's not implemented yet (as of the last updates)

  tinycp.com

  . If your main goal is to save resources, TinyCP might appeal, but given you have an 8 GB VPS, saving a few hundred MB by using TinyCP over something like Hestia is probably not critical. TinyCP could manage simple LAMP sites, but you'd still be on your own for Docker container orchestration.

- **Webmin (with/without Virtualmin) –** *Webmin is a general systems management GUI.* Unlike others, Webmin is not solely focused on web hosting – it's a tool to manage all aspects of a Unix system via web browser (editing config files, managing users, services, etc.). For web hosting, people often add the **Virtualmin** module on top of Webmin, which provides a more cPanel-like experience for managing multiple virtual hosts (domains), mail, databases under Webmin's framework. **Pros:** Webmin/Virtualmin is **free and open-source**. It is very powerful – you can configure almost any service (Apache, Nginx, Postfix, MySQL, firewall, etc.) through it. It's also quite lightweight given its breadth. **Cons:** The interface can be a bit dated and overwhelming, especially Webmin's default look. There is a learning curve to figure out how to do typical hosting tasks since it's not as streamlined as cPanel for those specific tasks. In terms of Docker, Webmin doesn't manage Docker containers by default (though Webmin's community has modules for many things – it's possible someone made a Docker plugin, but it's not mainstream). You can always just use Webmin's built-in *Terminal* or *Command* module to run Docker commands, but that's not much different than SSH. Webmin is suitable if you want to manually administer the server through a GUI rather than command line – it gives you toggles and forms for most config files. If you feel comfortable with Linux administration already, Webmin might be unnecessary. For a production environment, Webmin/Virtualmin is quite stable (many

hosting providers use Virtualmin for low-cost offerings). But again, it doesn't inherently simplify Docker usage. Use it if you want a free all-in-one server admin GUI, not specifically for app deployment.

- **Kusanagi –** *Special mention: Kusanagi isn't a full control panel like others, but a performance-optimized stack.* Hostinger offers a **KUSANAGI 9** VPS template

  support.hostinger.com

  , which is essentially an optimized virtual machine image for running PHP CMS like WordPress, Drupal, Joomla at very high speed. It's developed by a Japanese company (Prime Strategy) and focuses on performance and security (it can switch between Nginx/Apache, has caching, etc.). Kusanagi is primarily operated via command-line "kusanagi" CLI commands rather than a web GUI

  support.hostinger.com

  support.hostinger.com

  . It's very useful if your goal is maximum WordPress speed – Kusanagi VMs are known to be extremely fast. However, **it is not designed for Docker or multi-language environments**. It doesn't provide a general interface to manage Node or Python apps (though you could install them manually). If you were simply hosting a few PHP CMS sites with top performance, Kusanagi is great. In your case, where Node.js (n8n) and containers are needed, Kusanagi would not be the right fit (it's too specialized). It also lacks a web control panel for general management – it's more of a configured environment. So unless your projects revolve solely around PHP CMS and you want to forego Docker for those, you wouldn't use Kusanagi for this scenario.

Summary of Panel Options

In conclusion, if your priority is to **manage multiple Dockerized applications easily**, the **Docker-native panels (Coolify, Dokploy, Cloudron, Easypanel)** stand out as the best fits. They provide exactly what you need: a way to deploy and control containers (for Node.js, databases, etc.) across multiple projects from a single interface. Among these, **Coolify** offers a great balance of ease and features (free SSL, simple UI) for a single-server setup, whereas **Dokploy** excels if you anticipate clustering or just prefer open-source with no frills. **Cloudron** is ideal if you want an app-store experience and don't mind the subscription for more than 2 apps – it could simplify running n8n and similar apps drastically (essentially one-click install and automated updates

cloudron.io

). **Easypanel** is also a strong contender if you like the idea of using buildpacks and a Heroku-like flow for your apps.

The traditional panels like cPanel/Plesk/DirectAdmin are proven for hosting, but they will treat your VPS more like a traditional hosting server rather than a container host. If you have a mix of needs (for example, you want to host a marketing WordPress site and also run separate microservices in Docker), you might even use *both* a traditional panel and Docker side-by-side. Plesk, with its Docker extension, is uniquely positioned to do a bit of both – but it comes at a monetary cost. For a purely Docker-centric workflow, most users skip cPanel/Plesk on a VPS and use tools like Portainer, Docker Compose, or one of the above OSS panels instead.

Below is a comparison table summarizing the key attributes of each panel discussed, to help you quickly assess which might suit your needs.

Comparison Table of Control Panels

Control Panel	Docker Support	Ease of Use	Multi-Project Management	Node.js (n8n) Compatibility	Free or Paid
Coolify	**Full –** Built on Docker (deploys apps in containers). Great for running any Dockerized app out of the box.	**High –** User-friendly GUI, good for beginners hostinger.com hostinger.com .	**Yes –** Designed to host multiple apps/databases on one server. Single-server focus (no clustering).	**Yes –** Can deploy Node.js apps easily (via Docker images or Git). Suitable for n8n (community uses it for n8n).	**Free OSS** (self-host) or $4/mo managed plan hostinger.com .
Dokploy	**Full –** Uses Docker Compose/Swarm; native Docker integration (CLI & Traefik) hostinger.com . Ideal for container orchestration.	**Medium –** Clean UI but more dev-focused; requires some Docker knowledge (no auto SSL by default) hostinger.com .	**Yes –** Multi-app and **multi-server** capable (supports clustering with Swarm) hostinger.com . Great for microservices.	**Yes –** Any Dockerized app. Easily runs Node.js services; suits n8n via Docker Compose templates.	**Free OSS** (no paid tier) hostinger.com hostinger.com .
Cloudron	**Full –** Containerizes all apps internally (one-click app store deployments in Docker) hostinger.com .	**Very High –** Extremely easy; non-technical friendly app-store interface.	**Yes –** Manage many apps on one server; plus user roles & groups across apps hostinger.com . (No multi-server unless migrating entire server).	**Yes –** Provides official n8n app cloudron.io and many others. Custom apps possible but less flexible than pure Docker panels.	**Freemium** – Free for 2 apps; paid ~$15/mo for unlimited apps hostinger.com .

Control Panel	Docker Support	Ease of Use	Multi-Project Management	Node.js (n8n) Compatibility	Free or Paid
Easypanel	**Full –** Built on Docker (uses buildpacks & images) hostinger.com . Zero-downtime deploys.	**High –** Modern UI geared to developers; easy one-click app deploys and web terminal hostinger.com .	**Yes –** Can host multiple applications and databases. (Free version may limit project count; paid unlocks unlimited) hostinger.com .	**Yes –** Supports Node.js, Python, PHP, etc. via Docker images or buildpacks. Running n8n is supported (either by image or Dockerfile).	**Free** (open-source, with project limits) **or Paid** ($9.9–$23.9/mo for pro) hostinger.com .
Plesk	**Partial –** Offers a Docker extension for running containers from the GUI hostinger.com . Not a full orchestrator, but can manage individual containers.	**High –** Polished interface, easy for admins and clients.	**Yes –** Supports many domains, sites, and users on one server. Good for hosting multiple websites.	**Yes (via extension)** – Plesk has a Node.js extension for running Node apps. n8n could be run as a Node app or via Docker in Plesk, but requires manual setup.	**Paid** – Commercial license (≈$14–$40+/mo depending on edition) hostinger.com .
cPanel/WHM	**None natively –** Not designed for Docker. (Containers would need to be managed outside cPanel).	**High –** Very familiar UI for website hosting tasks.	**Yes –** Manages multiple accounts, domains, databases easily (industry standard for shared hosting).	**Limited** – No built-in Node support except via third-party modules. Not ideal for Node apps like n8n (requires custom configuration).	**Paid** – Commercial license (≈$15–$60/mo, scaling with accounts) hostinger.com .
DirectAdmin	**None –** No Docker integration. Traditional hosting only.	**Medium –** Decent interface (not as slick as Plesk). Relatively straightforward.	**Yes –** Can host multiple websites and user accounts; suitable for reselling or multiple projects.	**Limited** – Focuses on PHP/Apache. No native Node.js features. (Can run Node apps manually, but panel won't manage them.)	**Paid** – Lower cost than cPanel/Plesk (licenses ~$5–$29/mo) hostinger.com .
CyberPanel	**None –** No container support; built around OpenLiteSpeed web server.	**Medium –** Web UI is fairly user-friendly, especially for WordPress users (LSCache integration).	**Yes –** Can host many websites; has multi-user (admin/reseller) capabilities.	**Limited** – Optimized for PHP (WordPress). No built-in Node support. Would need manual config for Node or Python apps.	**Free** (Open-source). Optional enterprise add-ons (LiteSpeed Enterprise webserver license) are paid but not required.

Control Panel	Docker Support	Ease of Use	Multi-Project Management	Node.js (n8n) Compatibility	Free or Paid
HestiaCP	**None –** No Docker features. Traditional LAMP/LEMP stack management.	**Medium/High –** Simple, clean interface; easy for basic hosting tasks (one-click installs, etc.).	**Yes –** Supports multiple domains and users on one server. Suitable for various small projects/sites.	**Limited** – Geared toward PHP and static sites. No native support for running a Node app persistently.	**Free** (Open-source).
Webuzo	**None –** (No container support; installs services on OS).	**Medium –** User-friendly, especially with Softaculous installer for apps.	**Yes –** Multi-domain and multi-user support (good for hosting many sites) hostinger.com .	**Limited** – Primarily for PHP/MySQL applications. Would require manual tweaks to host a Node app.	**Paid** – License required (plans ~$2.5–$25/mo) hostinger.com . (Might have a basic free tier with limited features.)
FASTPANEL	**None –** No native Docker support.	**Medium –** Easy to navigate; not cluttered. Has nice extras like web-based SSH and backups.	**Yes –** Supports multiple websites and user roles per site hostinger.com . Good for managing client sites on one server.	**Limited** – Focus on standard web stacks. No built-in Node management, so not straightforward for n8n (would be manual).	**Free** (requires free registration; proprietary but no cost).
TinyCP	**None –** No Docker integration (as of now).	**Medium –** Interface is very minimalistic. Easy for basic tasks but not much documentation.	**Yes –** Can host multiple sites/domains on one server (no fancy multi-user controls, but capable of several projects).	**Limited** – No special support for Node.js. It's mainly for simple web/mail/DB management.	**Free** (Open-source).
Webmin/Virtualmin	**None –** No out-of-box Docker UI integration.	**Low/Medium –** Powerful but complex; Webmin is more sysadmin-oriented. Virtualmin theme improves hosting usability.	**Yes –** Very robust multi-site, multi-user management (Virtualmin can manage many domains, reseller accounts, etc., even across multiple servers) hostinger.com hostinger.com .	**Limited** – Can configure almost anything, but you must set up Node apps manually. Not specifically tailored for Node workflows.	**Free** (Open-source).

*(Above “Node.js (n8n) Compatibility” refers to how easily the panel can deploy or manage a Node.js application like n8n. In all cases, you **can** run n8n on the VPS – the difference is whether the panel helps manage it or you’d do it manually.)*

Recommendations and Conclusion

For your scenario – running multiple services in Docker containers – a **Hostinger VPS** is the clear choice (over Cloud Hosting) due to the need for root access and Docker support

hostinger.com

. We recommend starting with the **KVM 2 VPS plan** (2 vCPU, 8 GB RAM) which offers a strong foundation for multi-container workloads

hostinger.com

. Keep an eye on CPU/RAM usage as you deploy your projects; if you approach the limits, Hostinger allows easy upgrading to a larger VPS (e.g., 4 vCPU, 16 GB)

hostinger.com

.

When it comes to managing your applications, leveraging a **Docker-centric control panel** will greatly simplify operations. If you prefer a no-cost, open-source solution and don’t mind a bit of initial setup, **Coolify** is an excellent starting point for a single-server deployment – you’ll get a slick interface to deploy Node.js services like n8n and also spin up MySQL/Redis/Python apps in containers as needed. It will handle SSL, monitoring, and backups in a convenient way

hostinger.com

hostinger.com

. On the other hand, if you value premium support or an app-store model, **Cloudron** might be worth the subscription for the convenience (literally a few clicks to have n8n, databases, etc. running). For maximum flexibility and no vendor lock-in, open-source options like **Dokploy** (if you plan to scale out or use Docker Swarm) or **Easypanel** are also great.

Traditional panels like cPanel/Plesk are best if you also need to host standard websites or email on the same server and want those familiar management tools. If you go that route, **Plesk** would be preferable over cPanel because it at least can interface with Docker containers

hostinger.com

. You could, for example, use Plesk to host a PHP site and simultaneously run an n8n Docker container via Plesk's Docker extension. Just be mindful of resource usage and the complexity of running two "systems" on one server (the hosting panel plus Docker). Many users in your position opt to skip cPanel/Plesk on a VPS and instead use lightweight tools (like those Docker panels or even just Docker Compose + Portainer) to avoid the overhead and cost.

Actionable next steps:

- Deploy your Hostinger VPS and ensure Docker (and Docker Compose) is installed – using Hostinger's Docker template VM can jump-start this

  hostinger.com

  .

- Choose a control panel based on the above comparisons. If uncertain, you could start by trying an open-source panel (since it's free to test). For instance, install Coolify or Portainer (Portainer is another popular Docker management UI) to get an immediate web interface for your containers.
- Gradually deploy your services (database containers, n8n, etc.) and monitor usage. Make sure to configure proper reverse proxies (the Docker-focused panels usually handle this via Traefik or Nginx proxies automatically).
- If you find you miss traditional hosting features (like an integrated mail server or easy file manager), you can still install something like HestiaCP or CyberPanel on the VPS, but note that they may conflict with Docker on ports (e.g., both might try to use port 80). It's usually best to commit to either a Docker-centric architecture or a traditional one to avoid complexity.
- Keep security in mind: whichever panel you use, regularly update it and your containers. Hostinger VPS plans come with weekly backups (depending on plan) and you can set up your own backup routines (Cloudron and Coolify both support automated backups to S3, for example)

  hostinger.com

  .

By selecting a capable VPS and the right control panel, you'll have a powerful yet manageable environment for your multiple projects. You'll be able to run isolated services for each project in Docker containers while ensuring the host has enough resources to handle ~1000 daily requests (which should be well within the capacity of the recommended VPS). As your projects grow, you can scale vertically (upgrade VPS) or even horizontally (if using a multi-node tool like Dokploy, or by adding additional VPS for specific services behind a reverse proxy). This setup gives you a lot of flexibility and reliability for production use.

Overall, **Hostinger's VPS with a Docker-friendly control panel** will meet your requirements and provide a future-proof infrastructure for your Node.js, PHP, Python, and MySQL based applications. Good luck with your deployment!

Sources:

- Hostinger – *VPS Docker Hosting Plans (KVM)*
  hostinger.com

  hostinger.com
- Hostinger – *Managed Cloud Hosting vs VPS (root access)*
  hostinger.com
- Hostinger – *Plesk Features (Docker integration)*
  hostinger.com
- Hostinger – *CloudPanel (supports Node.js, Python templates)*
  hostinger.com
- Hostinger – *Cloudron description (containerized apps, backups, user management)*
  hostinger.com

  hostinger.com
- Hostinger – *Cloudron Pricing (free tier vs paid)*
  hostinger.com
- Hostinger – *Dokploy features (Traefik, multi-node, DB support)*
  hostinger.com

  hostinger.com
- Hostinger – *Coolify vs Dokploy comparison (performance, security, pricing)*
  hostinger.com

  hostinger.com

  hostinger.com
- Hostinger – *FASTPANEL features (backups, web SSH)*
  hostinger.com

hostinger.com

- Hostinger Tutorials – *What is Cloud Hosting (explanation of root access on VPS vs Cloud)*
  hostinger.com