GUIDE D'INSTALLATION PAS À PAS

Serveur VPS avec HestiaCP

Guide réutilisable pour tout VPS Ubuntu 24.04

Panel	**HestiaCP 1.10.4**
Système	**Ubuntu 24.04 LTS**
Pile	**Nginx · Apache · PHP-FPM · MariaDB · Exim · Dovecot · Bind**
Commandes vérifiées le	**10 septembre 2026**

Web · DNS · Mail · SSL · Sécurité · Docker · Supervision · Sauvegardes

Fiche du serveur

Remplissez la dernière colonne avant de commencer. Dans tout le guide, remplacez chaque `<VARIABLE>` par votre valeur. Pour obtenir un PDF déjà personnalisé, voir l'annexe « Régénérer ce guide ».

Variable	Signification	Exemple	Votre valeur
`<IP>`	Adresse IPv4 publique du VPS	203.0.113.10	
`<IPV6>`	Adresse IPv6 publique du VPS	2001:db8::10	
`<DOMAINE>`	Domaine principal du serveur	exemple.com	
`<FQDN>`	Nom complet du serveur (hostname)	srv1.exemple.com	
`<HOST>`	Nom court du serveur	srv1	
`<USER>`	Utilisateur Linux sudo (SSH)	jean	
`<ADMIN>`	Administrateur du panel HestiaCP	hadmin	
`<EMAIL>`	E-mail de l'administrateur	admin@exemple.com	
`<SSH_PORT>`	Port SSH personnalisé	50022	
`<PANEL_PORT>`	Port du panel HestiaCP	8083	
`<CLIENT>`	Utilisateur Hestia qui héberge les projets	client1	
`<SITE>`	Domaine d'un projet hébergé	monprojet.com	
`<BACKUP_HOST>`	Serveur de sauvegarde distant (SFTP)	backup.exemple.net	
`<BACKUP_USER>`	Utilisateur SFTP de la sauvegarde	sauvegarde	
`<BACKUP_PORT>`	Port SSH du serveur de sauvegarde	22	

Conventions

- Les commandes se copient telles quelles (pas d'invite `$`). Les lignes grises qui commencent par `#` sont des commentaires.
- Chapitres 1 à 3 : connecté en **root**. À partir du chapitre 4 : avec votre utilisateur `<USER>` et `sudo`.
- Chaque **étape** a une case à cocher : avancez dans l'ordre, une étape à la fois.
- Les encadrés Vérification indiquent le résultat attendu avant de passer à la suite.

Sommaire

1 Avant de commencer 6

Étape 1 · Choisir le système d'exploitation 6
Étape 2 · Commander et réinstaller le VPS 6
Étape 3 · Choisir le nom du serveur 6
Étape 4 · Préparer le DNS chez le registrar (OVH) 7
Étape 5 · Configurer le reverse DNS (PTR) chez l'hébergeur 7
Étape 6 · Se connecter au VPS 7

2 Préparer le système 8

Étape 7 · Mettre le système à jour 8
Étape 8 · Définir le nom du serveur 8
Étape 9 · Régler le fuseau horaire 8
Étape 10 · Supprimer UFW 9
Étape 11 · Vérifier les noms réservés 9
Étape 12 · Éviter le blocage IPv6 pendant l'installation 9
Étape 13 · Vérifier la configuration réseau (Netplan) 9
Étape 14 · Ajouter un fichier d'échange (swap) 10
Étape 15 · Installer les outils de base 10
Étape 16 · Créer votre utilisateur sudo 10
Étape 17 · Installer votre clé SSH 11

3 Installer HestiaCP 12

Étape 18 · Ouvrir une session tmux 12
Étape 19 · Télécharger l'installeur officiel 12
Étape 20 · Lancer l'installation 12
Étape 21 · Redémarrer le serveur 14
Étape 22 · Se connecter au panel 14
Étape 23 · Réparer le certificat du panel (si besoin) 14
Étape 24 · Découvrir l'interface 14

4 Sécuriser le serveur 16

Étape 25 · Se connecter avec votre utilisateur 16
Étape 26 · Rendre les commandes HestiaCP utilisables avec sudo 16
Étape 27 · Activer la double authentification du panel 16
Étape 28 · Fermer l'IPv6 entrante 17
Étape 29 · Changer le port SSH (1/2) : ouvrir le nouveau port 17
Étape 30 · Changer le port SSH (2/2) : fermer le port 22 et durcir 18
Étape 31 · Vérifier le pare-feu et Fail2ban 19
Étape 32 · Cacher phpMyAdmin 19
Étape 33 · Vérifier les mises à jour automatiques 20

5 Créer l'utilisateur d'hébergement 21

Étape 34 · Créer un forfait « projets » 21

Étape 35 · Créer l'utilisateur `<CLIENT>` 21
Étape 36 · Vérifier les serveurs de noms de l'utilisateur 22
Étape 37 · Autoriser votre clé SSH pour `<CLIENT>` 22
Étape 38 · Installer Composer pour `<CLIENT>` 22
Étape 39 · Choisir la version de PHP en ligne de commande 22

6 Héberger son propre DNS 23
Étape 40 · Créer la zone du domaine principal 23
Étape 41 · Tester la zone sur le serveur 23
Étape 42 · Déclarer les glue records chez OVH 24
Étape 43 · Basculer les serveurs DNS chez OVH 24
Étape 44 · Vérifier la délégation 24
Étape 45 · Ajouter un autre domaine (ex. un .ci) 24
Étape 46 · Ajouter un sous-domaine dans le DNS 25
Étape 47 · Activer DNSSEC (facultatif) 25

7 Sites web et certificats SSL 26
Étape 48 · Créer le site 26
Étape 49 · Activer le certificat Let's Encrypt 26
Étape 50 · Choisir la version de PHP du site 26
Étape 51 · Créer un site en sous-domaine 27
Étape 52 · Installer WordPress en un clic (facultatif) 27
Étape 53 · Consulter les journaux d'un site 28

8 Bases de données MariaDB 29
Étape 54 · Créer une base pour un projet 29
Étape 55 · Accéder à phpMyAdmin 29
Étape 56 · Exporter une base 29
Étape 57 · Importer une base existante (migration) 29
Étape 58 · Changer le mot de passe d'une base 30

9 Messagerie 31
Étape 59 · Créer le domaine de messagerie 31
Étape 60 · Activer le certificat SSL de la messagerie 31
Étape 61 · Créer les boîtes mail 31
Étape 62 · Ajouter les enregistrements SRV (domaine principal) 32
Étape 63 · Configurer un logiciel de messagerie 32
Étape 64 · Tester l'envoi et la réception 32
Étape 65 · Utiliser Spamhaus avec une clé gratuite (recommandé) 33
Étape 66 · Envoyer des mails depuis une application (Symfony…) 33
Étape 67 · Configuration automatique pour Thunderbird (facultatif) 33

10 Déployer un projet Symfony 35
Étape 68 · Se connecter en tant que `<CLIENT>` 35
Étape 69 · Créer une clé de déploiement GitHub 35
Étape 70 · Cloner le projet 35

Étape 71 · Configurer l'environnement de production 36
Étape 72 · Installer les dépendances et la base 36
Étape 73 · Pointer le site sur le dossier public/ 37
Étape 74 · Créer le script de déploiement 37
Étape 75 · Ajouter les tâches planifiées 37
Étape 76 · Connaître les limites de PHP-FPM 38

11 Docker et applications en conteneurs 39
Étape 77 · Installer Docker (dépôt officiel) 39
Étape 78 · Configurer Docker : localhost et rotation des journaux 39
Étape 79 · Créer le générateur de modèles de proxy 40
Étape 80 · Publier une application : la méthode 42
Étape 81 · Installer Portainer (gestion de Docker) 42
Étape 82 · Utiliser Docker Compose pour les autres applications 43

12 Supervision 44
Étape 83 · Utiliser les outils intégrés à HestiaCP 44
Étape 84 · Installer Uptime Kuma (disponibilité et alertes) 44
Étape 85 · Installer Netdata (métriques en temps réel) 45
Étape 86 · Recevoir les alertes du serveur 45

13 Sauvegardes 46
Étape 87 · Vérifier la sauvegarde locale 46
Étape 88 · Préparer une clé SSH pour le serveur de sauvegarde 46
Étape 89 · Envoyer les sauvegardes HestiaCP hors du serveur 47
Étape 90 · Sauvegarder la configuration système 47
Étape 91 · Restaurer 48

14 Maintenance et dépannage 49
Étape 92 · Faire les mises à jour mensuelles 49
Étape 93 · Mémo des commandes utiles 49
Étape 94 · Où trouver les journaux 49
Étape 95 · Résoudre les problèmes fréquents 50
Étape 96 · Vérifier que tout est en place 51

A Annexe : adresses et régénération du guide 52
Étape 97 · Récapitulatif des adresses 52
Étape 98 · Générer ce guide pour un autre VPS 52
Étape 99 · Sources officielles 53

CHAPITRE 1

Avant de commencer

Ce guide installe un serveur d'hébergement complet sur un VPS neuf, piloté par **HestiaCP**. Il couvre les sites web, le DNS, la messagerie, le SSL, la sécurité, Docker, la supervision et les sauvegardes. Il remplace ISPConfig pour un serveur unique : HestiaCP est plus léger, son interface est moderne et il est mis à jour régulièrement.

Aspect	ISPConfig	HestiaCP
Interface	Ancienne	Moderne, en français
Poids	Plus lourd	Léger (Nginx + Apache + PHP-FPM)
Multi-serveur	Oui	Non (un serveur = un panel)
Mises à jour	Lentes	Fréquentes, automatiques par défaut
Permissions des sites	Groupes `clientX` / `webX`, ACL	Un utilisateur Linux par client : plus simple

Étape 1 Choisir le système d'exploitation

HestiaCP **1.10.4** accepte Debian 12 et 13, ainsi qu'Ubuntu 22.04, 24.04 et 26.04 LTS, en 64 bits (x86_64 ou ARM64).

Choisissez Ubuntu 24.04 LTS. Elle est supportée par HestiaCP depuis la version 1.9.0 (février 2025) et est donc bien rodée. Le support d'Ubuntu 26.04 n'est arrivé qu'en août 2026 (version 1.10.0).

> **UN SYSTÈME NEUF, OBLIGATOIREMENT**
>
> HestiaCP doit s'installer sur un système **fraîchement installé**. N'installez ni Apache, ni Nginx, ni MariaDB, ni Postfix avant lui. Si le VPS a déjà servi, réinstallez-le depuis le panel de l'hébergeur.

Étape 2 Commander et réinstaller le VPS

Ressources conseillées par HestiaCP : 4 cœurs, 4 Go de RAM, 40 Go SSD. Avec moins de 3 Go de RAM, l'installeur désactive tout seul l'antivirus ClamAV.

1. Commandez le VPS. Exemple : Contabo Cloud VPS 6 avec 6 cœurs, 12 Go de RAM et 200 Go de disque.
2. Choisissez l'image **Ubuntu 24.04**, sans panel ni application préinstallés.
3. Définissez un mot de passe root solide, et ajoutez votre clé SSH publique si l'interface le propose.
4. Notez l'**IPv4** et l'**IPv6** du VPS dans la fiche du serveur, au début de ce guide.

Étape 3 Choisir le nom du serveur

Le serveur a besoin d'un nom complet (FQDN) du type `nom.domaine.tld` : il faut **au moins deux points**. C'est aussi l'adresse du panel.

Nous utilisons `<FQDN>`.

N'UTILISEZ PAS « MAIL. » NI « WWW. » COMME NOM DE SERVEUR

HestiaCP crée lui-même `mail.<domaine>` et `webmail.<domaine>` pour chaque domaine de messagerie, et `www.<domaine>` pour chaque site. Un nom de serveur neutre évite les conflits de certificats. Exemples : `srv1.domaine.com`, `panel.domaine.com`.

Étape 4 Préparer le DNS chez le registrar (OVH)

Au départ, le DNS du domaine reste chez OVH. Il faut juste que le nom du serveur pointe déjà vers le VPS, sinon le certificat SSL du panel ne pourra pas être créé. Le basculement du DNS vers le serveur se fait au chapitre 5.

Dans l'espace client OVH, allez dans **Noms de domaine › `<DOMAINE>` › Zone DNS › Ajouter une entrée** et ajoutez :

Type	Sous-domaine	Cible
A	`<HOST>`	`<IP>`

PAS D'ENREGISTREMENT AAAA (IPV6)

HestiaCP ne sert les sites et les mails qu'en **IPv4**. Si un enregistrement AAAA existe, Let's Encrypt essaie l'IPv6 en priorité et la validation échoue. Supprimez tout AAAA existant pour `<HOST>`, `@` et `www`.

VÉRIFICATION

Depuis votre ordinateur, `nslookup <FQDN>` doit répondre `<IP>`. La propagation prend de quelques minutes à une heure.

Étape 5 Configurer le reverse DNS (PTR) chez l'hébergeur

Gmail et Outlook rejettent ou classent en spam les mails d'un serveur dont l'IP ne « répond » pas au bon nom. Chez Contabo, allez dans **Customer Control Panel › Reverse DNS Management**. Pour l'**IPv4** `<IP>`, indiquez :

```
<FQDN>
```

À SAVOIR

Le PTR doit correspondre exactement au nom du serveur, car c'est le nom qu'Exim annonce en envoyant les mails. Exim n'utilise que l'IPv4 sous HestiaCP ; le PTR de l'IPv6 est facultatif, et vous pouvez y mettre le même nom.

Étape 6 Se connecter au VPS

Les chapitres 2 et 3 se font connecté en **root**. Depuis votre ordinateur :

```
ssh root@<IP>
```

CHAPITRE 2

Préparer le système

On part d'un Ubuntu 24.04 neuf, connecté en **root**. Ce chapitre prépare le terrain sans rien installer qui gênerait HestiaCP.

Étape 7 Mettre le système à jour

```
apt update && apt full-upgrade -y
apt autoremove -y
```

Si un nouveau noyau a été installé, redémarrez puis reconnectez-vous :

```
[ -f /var/run/reboot-required ] && reboot
```

Étape 8 Définir le nom du serveur

```
hostnamectl set-hostname <FQDN>
```

Ajoutez le nom du serveur dans `/etc/hosts` :

```
cat >> /etc/hosts << 'EOF'
<IP> <FQDN> <HOST>
<IPV6> <FQDN> <HOST>
EOF
```

Les images Contabo utilisent **cloud-init**, qui peut remettre l'ancien nom du serveur au redémarrage. Bloquez ce comportement :

```
[ -d /etc/cloud/cloud.cfg.d ] && echo 'preserve_hostname: true' > /etc/cloud/cloud.cfg.d/99-preserve-hostname.cfg
```

VÉRIFICATION

`hostname -f` doit afficher `<FQDN>`.

Étape 9 Régler le fuseau horaire

Adaptez la zone à votre pays. La liste complète s'obtient avec `timedatectl list-timezones`.

```
timedatectl set-timezone Africa/Abidjan
timedatectl
```

Étape 10 Supprimer UFW

HestiaCP gère son propre pare-feu (iptables + Fail2ban). L'installeur considère UFW comme un paquet en conflit : supprimez-le avant.

```
ufw disable
apt purge -y ufw
```

Étape 11 Vérifier les noms réservés

HestiaCP crée un compte Linux pour son administrateur. Le nom `<ADMIN>` ne doit exister **ni comme utilisateur, ni comme groupe**, et doit être différent de `<USER>`.

```
getent passwd <ADMIN>; getent group <ADMIN>
```

VÉRIFICATION

Aucune ligne ne doit s'afficher.

POURQUOI PAS « ADMIN » ?

Sur les images Ubuntu de Contabo, un **groupe `admin` existe déjà**, et l'installeur refuse un nom déjà pris. Choisissez un nom personnel (ex. `hadmin`, `skadmin`) : c'est aussi un nom de moins à deviner pour les robots.

Étape 12 Éviter le blocage IPv6 pendant l'installation

Sur certains VPS, l'installeur se bloque en téléchargeant ses paquets par IPv6. Dites au système de préférer l'IPv4 pour les connexions sortantes (l'IPv6 reste active) :

```
sed -i 's|^#precedence ::ffff:0:0/96  100|precedence ::ffff:0:0/96  100|' /etc/gai.conf
grep "^precedence" /etc/gai.conf
```

VÉRIFICATION

La commande `grep` affiche `precedence ::ffff:0:0/96 100`.

Étape 13 Vérifier la configuration réseau (Netplan)

L'installeur s'attend à trouver la configuration réseau dans `/etc/netplan`.

```
ls /etc/netplan/
ip -6 addr show scope global
```

VÉRIFICATION

Au moins un fichier `.yaml` dans `/etc/netplan/`, et votre IPv6 `<IPV6>` dans la sortie de `ip`.

Étape 14 Ajouter un fichier d'échange (swap)

HestiaCP ne crée un swap que sur les très petits serveurs. Un swap de 4 Go évite qu'un pic de mémoire ne fasse planter MariaDB ou ClamAV.

```
swapon --show
```

Si rien ne s'affiche :

```
fallocate -l 4G /swapfile
chmod 600 /swapfile
mkswap /swapfile
swapon /swapfile
echo '/swapfile none swap sw 0 0' >> /etc/fstab
echo 'vm.swappiness = 10' > /etc/sysctl.d/99-swap.conf
sysctl --system > /dev/null
```

VÉRIFICATION

`free -h` affiche une ligne `Swap:` de 4,0 Gi.

Étape 15 Installer les outils de base

Uniquement des outils sans conflit avec HestiaCP : ni serveur web, ni base de données, ni serveur mail.

```
apt install -y curl wget git micro htop tmux unzip ca-certificates gnupg lsb-release dnsutils
```

Faites de `micro` l'éditeur par défaut :

```
update-alternatives --install /usr/bin/editor editor /usr/bin/micro 60
update-alternatives --set editor /usr/bin/micro
```

Étape 16 Créer votre utilisateur sudo

Vous ne vous connecterez plus en root après le chapitre 4. Créez votre compte d'administration système :

```
adduser <USER>
usermod -aG sudo <USER>
```

Autorisez `sudo` sans mot de passe pour ce compte. C'est pratique, mais cela suppose une connexion par **clé SSH uniquement** (chapitre 4).

```
echo '<USER> ALL=(ALL:ALL) NOPASSWD: ALL' > /etc/sudoers.d/010-<USER>
chmod 440 /etc/sudoers.d/010-<USER>
visudo -c
```

VÉRIFICATION

`visudo -c` répond `parsed OK` pour chaque fichier.

Étape 17 Installer votre clé SSH

Sur votre ordinateur (pas sur le serveur), créez une clé si vous n'en avez pas :

```
ssh-keygen -t ed25519 -C "<USER>@<HOST>"
```

Puis envoyez-la au serveur :

```
ssh-copy-id <USER>@<IP>
```

Sous Windows sans `ssh-copy-id`, copiez le contenu de `C:\Users\<vous>\.ssh\id_ed25519.pub` dans le fichier `/home/<USER>/.ssh/authorized_keys` du serveur.

VÉRIFICATION

Dans un **nouveau** terminal, `ssh <USER>@<IP>` vous connecte sans mot de passe, puis `sudo whoami` répond `root`. Gardez la session root ouverte tant que ce test n'est pas réussi.

CHAPITRE 3

Installer HestiaCP

Toujours connecté en **root**. L'installation dure 15 à 30 minutes.

Étape 18 Ouvrir une session tmux

Si la connexion SSH coupe pendant l'installation, `tmux` la garde en vie sur le serveur.

```
tmux new -s hestia
```

ASTUCE

Si vous êtes déconnecté, reconnectez-vous puis tapez `tmux attach -t hestia` pour retrouver l'installation en cours.

Étape 19 Télécharger l'installeur officiel

```
cd /root
wget https://raw.githubusercontent.com/hestiacp/hestiacp/release/install/hst-install.sh
```

À SAVOIR

En cas d'erreur SSL : `apt-get update && apt-get install -y ca-certificates`, puis relancez le `wget`.
Le site officiel propose aussi un générateur de commande : https://hestiacp.com/install.html.

Étape 20 Lancer l'installation

Chaque option est expliquée juste après la commande. Copiez la commande **en entier** :

```
bash hst-install.sh \
  --hostname <FQDN> \
  --email <EMAIL> \
  --username <ADMIN> \
  --port <PANEL_PORT> \
  --lang fr \
  --apache yes --phpfpm yes --multiphp '8.3,8.4' \
  --vsftpd no --proftpd no \
  --named yes \
  --mysql yes --postgresql no \
  --exim yes --dovecot yes --sieve yes \
  --clamav yes --spamassassin yes \
  --iptables yes --fail2ban yes \
  --quota no --resourcelimit no --webterminal no \
  --api no
```

Option	Choix	Pourquoi
`--hostname`	`<FQDN>`	Nom du serveur et adresse du panel
`--email`	`<EMAIL>`	Reçoit les identifiants et les alertes. Prenez une adresse **externe** (Gmail…) : le mail du serveur n'existe pas encore
`--username`	`<ADMIN>`	Administrateur du panel. Pas `admin` (voir chapitre 2)
`--port`	`<PANEL_PORT>`	Port du panel (8083 par défaut)
`--apache` `--phpfpm`	yes	Nginx en façade + Apache derrière : les `.htaccess` (Symfony, WordPress) fonctionnent
`--multiphp`	8.3, 8.4	Versions de PHP installées ; la plus haute (8.4) devient celle par défaut. D'autres versions peuvent s'ajouter plus tard
`--vsftpd` `--proftpd`	no	Pas de FTP (non chiffré) : on déploie par SSH/SFTP et git
`--named`	yes	Serveur DNS Bind : vos domaines ont leurs propres serveurs de noms
`--mysql`	yes	MariaDB 11.8
`--exim` `--dovecot` `--sieve`	yes	Envoi, réception et filtres de messagerie
`--clamav` `--spamassassin`	yes	Antivirus et antispam (demandent au moins 3 Go de RAM)
`--iptables` `--fail2ban`	yes	Pare-feu et bannissement automatique des attaques
`--resourcelimit`	no	Encore instable d'après les notes de version (1.9.0)
`--webterminal`	no	Pas de terminal dans le navigateur : on utilise SSH
`--api`	no	L'API distante ne sert pas ici ; moins de surface d'attaque

Déroulement :

1. L'installeur affiche le récapitulatif des composants. Répondez `y` pour continuer.
2. Il demande le **mot de passe administrateur**. Tapez un mot de passe long et unique (au moins 16 caractères) et notez-le dans votre gestionnaire de mots de passe. Il s'affiche en clair à l'écran : c'est normal.
3. Si l'installeur signale des paquets en conflit, répondez `y` pour les supprimer. Sur un système neuf préparé au chapitre 2, il ne devrait pas y en avoir.

> **ATTENTION**
> Ne touchez à rien pendant l'installation. À la fin, l'installeur affiche « Congratulations! » avec l'adresse du panel, le nom d'utilisateur et le mot de passe, et envoie le même message à `<EMAIL>`.

Étape 21 Redémarrer le serveur

L'installeur demande d'appuyer sur une touche, puis redémarre. Sinon :

```
reboot
```

Au premier démarrage, une tâche programmée par HestiaCP demande automatiquement un certificat **Let's Encrypt** pour `<FQDN>`. Cela ne marche que si l'enregistrement DNS du chapitre 1 est en place.

> **À SAVOIR**
> Le journal complet de l'installation est dans `/root/hst_install_backups/`, et les journaux Let's Encrypt dans `/var/log/hestia/`.

Étape 22 Se connecter au panel

Ouvrez dans votre navigateur :

```
https://<FQDN>:<PANEL_PORT>
```

Connectez-vous avec `<ADMIN>` et votre mot de passe.

> **VÉRIFICATION**
> Le cadenas du navigateur indique un certificat valide (émis par Let's Encrypt). Si le navigateur signale un certificat non valide, passez à l'étape suivante.

Étape 23 Réparer le certificat du panel (si besoin)

Uniquement si le certificat n'est pas valide. Reconnectez-vous en SSH en root, vérifiez que le nom du serveur pointe bien vers le VPS, puis demandez le certificat :

```
dig +short <FQDN>
/usr/local/hestia/bin/v-change-sys-hostname <FQDN>
/usr/local/hestia/bin/v-add-letsencrypt-host
```

> **VÉRIFICATION**
> `dig` doit répondre `<IP>`, et `v-add-letsencrypt-host` ne doit afficher aucune erreur. Rechargez ensuite le panel dans le navigateur.

Étape 24 Découvrir l'interface

- **Utilisateurs** : les comptes d'hébergement. Chaque utilisateur Hestia est un utilisateur Linux avec son dossier `/home/<utilisateur>`.
- **Web**, **DNS**, **Messagerie**, **Bases de données**, **Tâches cron**, **Sauvegardes** : les ressources de l'utilisateur sélectionné.

- **Roue dentée (Paramètres du serveur)** : services, pare-feu, mises à jour, configuration. Le bouton **Moniteur de tâches** affiche les graphiques de charge.

> **N'HÉBERGEZ RIEN SOUS LE COMPTE ADMINISTRATEUR**
>
> La documentation officielle recommande de créer un utilisateur dédié pour vos sites, domaines et mails (chapitre 5). Le compte `<ADMIN>` sert uniquement à administrer.

CHAPITRE 4

Sécuriser le serveur

À partir de maintenant, on travaille avec votre utilisateur `<USER>` et `sudo`, plus jamais en root direct. HestiaCP a déjà mis en place un pare-feu (iptables) et Fail2ban ; ce chapitre complète ce qu'il ne fait pas.

Étape 25 Se connecter avec votre utilisateur

Depuis votre ordinateur :

```
ssh <USER>@<IP>
```

Étape 26 Rendre les commandes HestiaCP utilisables avec sudo

Les commandes HestiaCP (`v-…`) sont dans `/usr/local/hestia/bin` , mais `sudo` ne cherche pas dans ce dossier. Sans ce réglage, `sudo v-list-users` répond « command not found ».

```
echo 'Defaults secure_path="/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/snap/
bin:/usr/local/hestia/bin"' | sudo tee /etc/sudoers.d/020-hestia-path
sudo chmod 440 /etc/sudoers.d/020-hestia-path
sudo visudo -c
```

VÉRIFICATION

`sudo v-list-users` affiche la liste des utilisateurs, dont `<ADMIN>`.

Étape 27 Activer la double authentification du panel

Dans le panel, connecté en `<ADMIN>` :

1. Cliquez sur votre nom d'utilisateur en haut à droite (**Modifier l'utilisateur**).
2. Cochez **Activer l'authentification à deux facteurs** et enregistrez.
3. Scannez le QR code avec une application TOTP (Aegis, Google Authenticator, 2FAS…).
4. **Notez le code de récupération** dans votre gestionnaire de mots de passe.

ASTUCE

Si vous perdez votre téléphone, vous pouvez désactiver la double authentification en SSH avec `sudo v-delete-user-2fa <ADMIN>` .

Étape 28 Fermer l'IPv6 entrante

Le pare-feu de HestiaCP ne gère **que l'IPv4**. En IPv6, le SSH et le panel (qui écoute aussi sur `[::]:<PANEL_PORT>`) seraient joignables sans pare-feu ni Fail2ban. Comme HestiaCP ne sert rien en IPv6, on bloque tout le trafic IPv6 entrant, sauf les réponses et l'ICMPv6, indispensable au fonctionnement de l'IPv6. Le serveur peut toujours sortir en IPv6.

Créez le script de pare-feu IPv6 :

```
sudo tee /usr/local/sbin/ip6-firewall.sh > /dev/null << 'EOF'
#!/bin/sh
# Pare-feu IPv6 : HestiaCP ne filtre que l'IPv4.
# Bloque tout l'entrant IPv6 sauf loopback, réponses et ICMPv6.
ip6tables -P INPUT ACCEPT
ip6tables -F INPUT
ip6tables -A INPUT -i lo -j ACCEPT
ip6tables -A INPUT -m conntrack --ctstate ESTABLISHED,RELATED -j ACCEPT
ip6tables -A INPUT -p ipv6-icmp -j ACCEPT
ip6tables -P INPUT DROP
EOF
sudo chmod 750 /usr/local/sbin/ip6-firewall.sh
```

Puis le service qui l'applique à chaque démarrage :

```
sudo tee /etc/systemd/system/ip6-firewall.service > /dev/null << 'EOF'
[Unit]
Description=Pare-feu IPv6 (entrant bloque)
DefaultDependencies=no
Wants=network-pre.target
Before=network-pre.target
After=local-fs.target

[Service]
Type=oneshot
RemainAfterExit=yes
ExecStart=/usr/local/sbin/ip6-firewall.sh

[Install]
WantedBy=multi-user.target
EOF
sudo systemctl daemon-reload
sudo systemctl enable --now ip6-firewall.service
```

VÉRIFICATION

`sudo ip6tables -S INPUT` commence par `-P INPUT DROP`. Depuis votre ordinateur, si vous avez l'IPv6, `ssh -6 <USER>@<IPV6>` ne doit **pas** répondre, alors que `ssh <USER>@<IP>` fonctionne toujours.

Étape 29 Changer le port SSH (1/2) : ouvrir le nouveau port

On procède en deux temps pour ne jamais se retrouver bloqué dehors : d'abord SSH écoute sur **les deux ports**, puis on ferme le port 22.

Le fichier commence par `00-` pour passer avant les fichiers de cloud-init : pour un même réglage, OpenSSH garde la première valeur lue.

```
sudo tee /etc/ssh/sshd_config.d/00-securite.conf > /dev/null << 'EOF'
Port 22
Port <SSH_PORT>
EOF
sudo sshd -t && echo "Configuration SSH valide"
```

Ubuntu 24.04 démarre SSH à la demande via `ssh.socket` : il faut recharger systemd pour que le nouveau port soit pris en compte.

```
systemctl is-active ssh.socket
sudo systemctl daemon-reload
sudo systemctl restart ssh.socket
```

À SAVOIR

Si `systemctl is-active ssh.socket` répond `inactive`, votre image utilise le service classique : remplacez les deux dernières lignes par `sudo systemctl restart ssh`.

Mettez à jour le pare-feu HestiaCP. Il détecte tout seul les ports SSH et met à jour sa règle « SSH » et Fail2ban :

```
sudo v-update-firewall
sudo v-list-sys-sshd-port
sudo v-list-firewall | grep SSH
```

VÉRIFICATION

`v-list-sys-sshd-port` affiche `22` et `<SSH_PORT>` ; la règle SSH du pare-feu contient `22,<SSH_PORT>`. Dans un **nouveau terminal**, `ssh -p <SSH_PORT> <USER>@<IP>` fonctionne.

Étape 30 Changer le port SSH (2/2) : fermer le port 22 et durcir

Seulement après avoir réussi la connexion sur le port `<SSH_PORT>`. On ferme le port 22, on interdit la connexion root et on n'accepte plus que les clés SSH :

```
sudo tee /etc/ssh/sshd_config.d/00-securite.conf > /dev/null << 'EOF'
Port <SSH_PORT>
PermitRootLogin no
PasswordAuthentication no
KbdInteractiveAuthentication no
PubkeyAuthentication yes
EOF
sudo sshd -t && sudo systemctl daemon-reload && sudo systemctl restart ssh.socket
sudo v-update-firewall
```

GARDEZ VOTRE SESSION OUVERTE

Ne fermez pas le terminal actuel. Testez dans un nouveau terminal : `ssh -p <SSH_PORT> <USER>@<IP>` doit fonctionner, et `ssh root@<IP> -p <SSH_PORT>` doit être refusé. En cas de problème, la console VNC du panel Contabo permet toujours de se connecter.

VÉRIFICATION

`sudo sshd -T | grep -E "^(port|permitrootlogin|passwordauthentication)"` affiche `port <SSH_PORT>`, `permitrootlogin no` et `passwordauthentication no`.

Sur votre ordinateur, ajoutez un raccourci dans `~/.ssh/config` :

```
Host <HOST>
    HostName <IP>
    Port <SSH_PORT>
    User <USER>
```

Il suffira ensuite de taper `ssh <HOST>`.

Étape 31 Vérifier le pare-feu et Fail2ban

```
sudo v-list-firewall
sudo fail2ban-client status
```

Ports ouverts par HestiaCP : ping, `<PANEL_PORT>` (panel), 80/443 (web), 25/465/587 (SMTP), 143/993 (IMAP), 110/995 (POP3), 53 TCP/UDP (DNS) et `<SSH_PORT>` (SSH).

Prisons Fail2ban actives : `ssh-iptables`, `exim-iptables`, `dovecot-iptables`, `hestia-iptables` (panel), `phpmyadmin-auth` et `recidive`. Une IP bannie 5 fois en une journée est bloquée 10 jours sur tous les ports.

DÉBANNIR UNE IP (LA VÔTRE, PAR EXEMPLE)

`sudo v-list-firewall-ban` affiche les IP bannies. `sudo v-delete-firewall-ban <IP> ALL` débannit une IP partout.

NE RÉACTIVEZ JAMAIS UFW

HestiaCP et UFW se marchent dessus. Toute règle se gère dans le panel (**roue dentée › Pare-feu**) ou avec `v-add-firewall-rule`.

Étape 32 Cacher phpMyAdmin

phpMyAdmin est accessible sur **tous** vos domaines à l'adresse `/phpmyadmin`, que les robots testent en permanence. Remplacez ce chemin par un nom difficile à deviner :

```
sudo v-change-sys-db-alias pma gestion-bdd-7f3k9
```

phpMyAdmin sera alors accessible sur `https://<FQDN>/gestion-bdd-7f3k9/`. Choisissez votre propre suffixe.

Étape 33 Vérifier les mises à jour automatiques

Deux mécanismes complémentaires :

- **unattended-upgrades** (Ubuntu) installe chaque jour les correctifs de sécurité du système ;
- **HestiaCP** se met à jour lui-même chaque nuit (tâche `v-update-sys-hestia-all`).

```
systemctl is-enabled unattended-upgrades
cat /etc/apt/apt.conf.d/20auto-upgrades
sudo v-list-sys-hestia-autoupdate
```

VÉRIFICATION

`enabled` ; les deux lignes `APT::Periodic` valent `"1"` ; l'autoupdate HestiaCP est `Enabled`.

Si `20auto-upgrades` n'existe pas :

```
sudo dpkg-reconfigure -plow unattended-upgrades
```

MISES À JOUR MANUELLES

Pour PHP, Nginx et MariaDB, qui viennent de dépôts externes, lancez de temps en temps `sudo apt update && sudo apt upgrade`. Si apt demande quoi faire d'un fichier de configuration modifié, répondez toujours **garder la version actuelle** (N).

CHAPITRE 5

Créer l'utilisateur d'hébergement

Dans HestiaCP, **chaque utilisateur du panel est un vrai utilisateur Linux**. PHP-FPM, les tâches cron et la connexion SSH de ses sites tournent sous ce même compte.

Tous les fichiers d'un site appartiennent donc à un seul utilisateur. Les problèmes de droits connus sous ISPConfig (`webX:clientY`, ACL, `chmod 777` sur le cache Symfony) disparaissent.

Élément	Emplacement
Fichiers d'un site	`/home/<CLIENT>/web/<domaine>/public_html/`
Fichiers privés (hors web)	`/home/<CLIENT>/web/<domaine>/private/`
Journaux du site	`/home/<CLIENT>/web/<domaine>/logs/`
Boîtes mail	`/home/<CLIENT>/mail/<domaine>/`
Sauvegardes	`/backup/`

Étape 34 Créer un forfait « projets »

Un **forfait** (package) définit les limites et les réglages par défaut d'un utilisateur. Le forfait `default` interdit le SSH et ne garde qu'une sauvegarde : on en crée un plus adapté à partir de lui.

```
sudo cp /usr/local/hestia/data/packages/default.pkg /tmp/projets.pkg
sudo sed -i "s/^SHELL=.*/SHELL='bash'/; s/^BACKUPS=.*/BACKUPS='7'/" /tmp/projets.pkg
sudo v-add-user-package /tmp/projets.pkg projets
sudo rm /tmp/projets.pkg
sudo v-list-user-package projets
```

VÉRIFICATION

Dans la sortie, la ligne `SHELL` indique `bash`, la ligne `BACKUPS` indique `7` et la ligne `NS` indique `ns1.<DOMAINE>,ns2.<DOMAINE>`.

À SAVOIR

`BACKUPS='7'` garde 7 sauvegardes quotidiennes locales. Sur un petit disque, réduisez ce nombre. Le forfait se modifie aussi dans le panel : **roue dentée › Forfaits**.

Étape 35 Créer l'utilisateur `<CLIENT>`

Choisissez un mot de passe fort, que vous notez dans votre gestionnaire de mots de passe. L'espace avant la commande l'empêche d'être enregistré dans l'historique du shell :

```
 sudo v-add-user <CLIENT> 'MOT_DE_PASSE_FORT' <EMAIL> projets
```

ATTENTION

Depuis HestiaCP 1.10, un nom qui existe déjà comme utilisateur Linux est refusé. `<CLIENT>` doit donc être différent de `<USER>` et de `<ADMIN>`.

Étape 36 Vérifier les serveurs de noms de l'utilisateur

Les zones DNS créées par `<CLIENT>` utiliseront ces serveurs de noms :

```
sudo v-change-user-ns <CLIENT> ns1.<DOMAINE> ns2.<DOMAINE>
sudo v-list-user <CLIENT> json | grep -E '"(PACKAGE|SHELL|NS)"'
```

VÉRIFICATION

`"PACKAGE": "projets"`, `"NS": "ns1.<DOMAINE>,ns2.<DOMAINE>"`, `"SHELL": "bash"`.

Étape 37 Autoriser votre clé SSH pour `<CLIENT>`

Pour déployer vos projets, vous vous connecterez directement en `<CLIENT>`. Ajoutez votre clé publique, celle du fichier `~/.ssh/id_ed25519.pub` de votre ordinateur :

```
sudo v-add-user-ssh-key <CLIENT> "$(cat /home/<USER>/.ssh/authorized_keys | head -1)"
```

VÉRIFICATION

Depuis votre ordinateur, `ssh -p <SSH_PORT> <CLIENT>@<IP>` ouvre une session dans `/home/<CLIENT>`.

Étape 38 Installer Composer pour `<CLIENT>`

```
sudo v-add-user-composer <CLIENT>
```

Cette commande installe Composer 2 dans `/home/<CLIENT>/.composer/` et crée l'alias `composer` pour cet utilisateur.

Étape 39 Choisir la version de PHP en ligne de commande

Par défaut, la commande `php` utilise la version la plus récente installée. Pour que `<CLIENT>` utilise PHP 8.4 en SSH :

```
sudo v-change-user-php-cli <CLIENT> 8.4
```

CRON ET SCRIPTS : CHEMIN COMPLET OBLIGATOIRE

Cette commande crée seulement un **alias** dans `~/.bash_aliases`, que les tâches cron et les scripts ne lisent pas. Dans ces cas, écrivez toujours le chemin complet : `/usr/bin/php8.4`.

CHAPITRE 6

Héberger son propre DNS

Le serveur devient le serveur DNS de vos domaines. `<DOMAINE>` a ses propres serveurs de noms, `ns1.<DOMAINE>` et `ns2.<DOMAINE>`, déclarés chez le registrar par des **enregistrements glue**. Vos autres domaines, par exemple en `.ci`, utilisent ensuite ces mêmes serveurs de noms.

L'ORDRE COMPTE

Créez toujours la **zone DNS avant le domaine mail** : HestiaCP n'ajoute la clé DKIM que dans une zone qui existe déjà.

Étape 40 Créer la zone du domaine principal

La zone appartient à `<CLIENT>`, comme le site et les mails de `<DOMAINE>`. Le modèle `child-ns` est prévu pour un domaine qui sert lui-même de serveur de noms : il crée les enregistrements `ns1` et `ns2` qui pointent vers `<IP>`.

```
sudo v-add-dns-domain <CLIENT> <DOMAINE> <IP>
sudo v-change-dns-domain-tpl <CLIENT> <DOMAINE> child-ns
```

DANGER

Changer de modèle **efface les enregistrements ajoutés à la main**. Faites-le uniquement juste après la création, comme ici.

Ajoutez l'enregistrement du nom du serveur, dont dépendent le panel et son certificat :

```
sudo v-add-dns-record <CLIENT> <DOMAINE> <HOST> A <IP>
sudo v-list-dns-records <CLIENT> <DOMAINE>
```

VÉRIFICATION

La liste contient les NS `ns1.<DOMAINE>.` et `ns2.<DOMAINE>.`, les A `@`, `ns1`, `ns2`, `mail` et `<HOST>`, le CNAME `www`, le MX `mail.<DOMAINE>.`, le TXT SPF et le TXT `_dmarc`.

Étape 41 Tester la zone sur le serveur

```
dig @<IP> <DOMAINE> SOA +short
dig @<IP> ns1.<DOMAINE> +short
dig @<IP> <FQDN> +short
```

VÉRIFICATION

La première commande affiche une ligne commençant par `ns1.<DOMAINE>.` ; les deux suivantes affichent `<IP>`.

Étape 42 Déclarer les glue records chez OVH

Dans l'espace client OVH : **Web Cloud › Noms de domaine › `<DOMAINE>` › onglet GLUE › Ajouter**.

Nom d'hôte	Adresse IP
`ns1.<DOMAINE>`	`<IP>`
`ns2.<DOMAINE>`	`<IP>`

Ne mettez que l'IPv4 : l'IPv6 entrante est bloquée par le pare-feu (chapitre 4), donc une glue IPv6 rendrait le DNS injoignable pour une partie d'Internet.

Étape 43 Basculer les serveurs DNS chez OVH

Toujours sur `<DOMAINE>` : **onglet Serveurs DNS › Modifier les serveurs DNS**. Remplacez les serveurs d'OVH par :

```
ns1.<DOMAINE>
ns2.<DOMAINE>
```

À SAVOIR

OVH vérifie que vos serveurs répondent avant d'accepter le changement : c'est pour cela que la zone doit exister avant. La propagation prend de quelques heures à 48 h.

Étape 44 Vérifier la délégation

Depuis le serveur ou votre ordinateur :

```
dig +trace NS <DOMAINE> | tail -n 8
dig NS <DOMAINE> +short
```

VÉRIFICATION

La fin du `+trace` montre `ns1.<DOMAINE>` et `ns2.<DOMAINE>` avec la réponse de `<IP>`. Un contrôle complet est possible sur https://zonemaster.net.

Étape 45 Ajouter un autre domaine (ex. un .ci)

Pour un domaine de projet `<SITE>` enregistré chez un autre registrar, utilisez le modèle par défaut, qui inclut aussi les enregistrements SRV de la messagerie. Ses serveurs de noms seront ceux de l'utilisateur, c'est-à-dire `ns1.<DOMAINE>` et `ns2.<DOMAINE>`.

```
sudo v-add-dns-domain <CLIENT> <SITE> <IP>
sudo v-list-dns-records <CLIENT> <SITE>
```

Chez le registrar de `<SITE>`, remplacez ensuite les serveurs DNS par `ns1.<DOMAINE>` et `ns2.<DOMAINE>`. **Pas de glue** ici : elle ne sert que pour le domaine qui porte les serveurs de noms.

> **À SAVOIR**
> Si `<SITE>` est le même domaine que `<DOMAINE>`, cette étape est déjà faite : passez à la suite.

Étape 46 Ajouter un sous-domaine dans le DNS

HestiaCP **ne crée pas** l'enregistrement DNS quand vous ajoutez un site en sous-domaine. Ajoutez-le vous-même avant de créer le site, puis vérifiez :

```
sudo v-add-dns-record <CLIENT> <DOMAINE> app A <IP>
dig @<IP> app.<DOMAINE> +short
```

Étape 47 Activer DNSSEC (facultatif)

DNSSEC signe la zone pour empêcher la falsification des réponses DNS. HestiaCP ne l'autorise qu'en mode `hestia-zone`, un réglage de cluster sans effet sur un serveur seul.

```
sudo v-change-sys-config-value DNS_CLUSTER_SYSTEM hestia-zone
sudo v-change-dns-domain-dnssec <CLIENT> <DOMAINE> yes
sudo v-list-dnssec-public-key <CLIENT> <DOMAINE> shell
```

La dernière commande affiche l'enregistrement **DS**, avec le key tag, le flag 257, l'algorithme 13 et la clé publique. Chez OVH : `<DOMAINE>` **› onglet Enregistrements DS › Modifier**, puis copiez-y ces valeurs.

> **DÉSACTIVER DNSSEC DANS LE BON ORDRE**
> Pour désactiver DNSSEC plus tard, supprimez **d'abord** le DS chez le registrar et attendez 48 h. Sinon le domaine devient injoignable. Sur un ancien serveur, OVH a déjà refusé un DS avec l'erreur « Error while checking DNSKEYS » : dans ce cas, réessayez plus tard ou contactez le support.

> **VÉRIFICATION**
> https://dnsviz.net affiche une chaîne de confiance verte pour `<DOMAINE>`.

CHAPITRE 7

Sites web et certificats SSL

Chaque site passe par **Nginx** (en façade, pour les fichiers statiques, le SSL et le cache), puis **Apache** (qui lit les `.htaccess`), puis **PHP-FPM**. Le PHP-FPM tourne sous l'utilisateur `<CLIENT>`, avec un pool par domaine.

PRÉREQUIS

Le domaine doit déjà pointer vers le serveur (chapitre 6), sinon Let's Encrypt ne peut pas valider le certificat. Vérifiez avec `dig +short <SITE>` et `dig +short www.<SITE>` : les deux doivent répondre `<IP>`.

Étape 48 Créer le site

```
sudo v-add-web-domain <CLIENT> <SITE>
```

L'alias `www.<SITE>` est ajouté automatiquement. Le site se trouve dans `/home/<CLIENT>/web/<SITE>/public_html/`, avec une page d'accueil HestiaCP par défaut.

Étape 49 Activer le certificat Let's Encrypt

Indiquez explicitement l'alias `www`, sinon le certificat ne couvre que `<SITE>` :

```
sudo v-add-letsencrypt-domain <CLIENT> <SITE> www.<SITE>
```

Forcez ensuite le HTTPS, et activez HSTS pour que les navigateurs refusent ensuite le HTTP :

```
sudo v-add-web-domain-ssl-force <CLIENT> <SITE>
sudo v-add-web-domain-ssl-hsts <CLIENT> <SITE>
```

VÉRIFICATION

`curl -sI http://<SITE> | head -3` renvoie une redirection `301` vers `https://`, et `curl -sI https://www.<SITE> | head -1` renvoie `HTTP/2 200`.

ASTUCE

Le renouvellement est automatique. Dans le panel, la même chose se fait en modifiant le domaine : **Activer le SSL pour ce domaine**, **Utiliser Let's Encrypt pour obtenir un certificat SSL**, **Activer la redirection automatique en HTTPS**.

Étape 50 Choisir la version de PHP du site

Listez les versions disponibles, puis appliquez-en une au site :

```
sudo v-list-sys-php
sudo v-list-web-templates-backend
sudo v-change-web-domain-backend-tpl <CLIENT> <SITE> PHP-8_4
```

Pour installer une version supplémentaire plus tard (ex. 8.5) :

```
sudo v-add-web-php 8.5
```

RÉGLAGES PHP PAR SITE

Pour modifier une valeur PHP d'un seul site (ex. `memory_limit`), créez un fichier `.user.ini` dans le dossier servi par le web (pour Symfony : `public_html/public/`) :

```
memory_limit = 512M
upload_max_filesize = 64M
post_max_size = 64M
```

Il est pris en compte après environ 5 minutes. HestiaCP règle déjà `upload_max_filesize` et `post_max_size` à 100M.

Étape 51 Créer un site en sous-domaine

Exemple avec `app.<DOMAINE>` : d'abord l'enregistrement DNS (chapitre 6), puis le site et son certificat. Pour un sous-domaine, pas d'alias `www` :

```
sudo v-add-dns-record <CLIENT> <DOMAINE> app A <IP>
sudo v-add-web-domain <CLIENT> app.<DOMAINE>
sudo v-add-letsencrypt-domain <CLIENT> app.<DOMAINE>
sudo v-add-web-domain-ssl-force <CLIENT> app.<DOMAINE>
```

Étape 52 Installer WordPress en un clic (facultatif)

HestiaCP intègre un installeur d'applications : WordPress, Laravel, Symfony, Nextcloud, Joomla, Drupal, PrestaShop, entre autres.

1. **Web** › survolez le domaine › **Installer une application**.
2. Choisissez **WordPress**, remplissez le formulaire (titre, compte administrateur WordPress, e-mail, version de PHP) et laissez cochée la création de la base de données.
3. Validez : la base est créée et WordPress est installé dans `public_html`.

ASTUCE

Pour un WordPress, le modèle proxy **caching** de Nginx accélère nettement les pages publiques : `sudo v-change-web-domain-proxy-tpl <CLIENT> <SITE> caching`.

Étape 53 Consulter les journaux d'un site

```
sudo tail -f /var/log/apache2/domains/<SITE>.error.log
sudo tail -f /var/log/nginx/domains/<SITE>.error.log
```

Ces journaux sont aussi visibles dans `/home/<CLIENT>/web/<SITE>/logs/` et dans le panel, via l'icône journal du domaine.

CHAPITRE 8

Bases de données MariaDB

HestiaCP installe **MariaDB 11.8**. Chaque base appartient à un utilisateur Hestia, et son nom comme celui de son utilisateur MariaDB sont **préfixés** par le nom de l'utilisateur Hestia. Le port 3306 reste fermé vers l'extérieur.

Étape 54 Créer une base pour un projet

```
 sudo v-add-database <CLIENT> app app 'MOT_DE_PASSE_BDD'
sudo v-list-databases <CLIENT>
```

VÉRIFICATION

La base `<CLIENT>_app` apparaît, avec l'utilisateur `<CLIENT>_app`. Ce sont ces noms complets qu'il faut utiliser dans l'application.

Le jeu de caractères par défaut est `utf8mb4`. Dans une application, l'hôte est `127.0.0.1` et le port `3306`.

Étape 55 Accéder à phpMyAdmin

Ouvrez `https://<FQDN>/<alias-choisi-au-chapitre-4>/` et connectez-vous avec l'utilisateur de la base (`<CLIENT>_app`) et son mot de passe.

Étape 56 Exporter une base

```
sudo v-dump-database <CLIENT> <CLIENT>_app > ~/<CLIENT>_app.sql
```

Pour une version compressée, utilisez `sudo v-dump-database <CLIENT> <CLIENT>_app file zstd`. La commande affiche le chemin du fichier créé dans `/backup/`, qui est supprimé automatiquement au bout d'une heure.

Étape 57 Importer une base existante (migration)

Depuis l'ancien serveur, exportez la base puis copiez-la sur le nouveau :

```
# Sur l'ancien serveur
mysqldump --single-transaction --routines --triggers ancienne_base > ancienne_base.sql
scp -P <SSH_PORT> ancienne_base.sql <USER>@<IP>:~/
```

Sur le nouveau serveur, importez-la dans la base créée plus haut :

```
mariadb -u <CLIENT>_app -p <CLIENT>_app < ~/ancienne_base.sql
rm ~/ancienne_base.sql
```

ATTENTION

Un export contient souvent des données personnelles. Supprimez les fichiers `.sql` une fois l'import terminé, sur les deux serveurs.

Étape 58 Changer le mot de passe d'une base

```
sudo v-change-database-password <CLIENT> <CLIENT>_app 'NOUVEAU_MOT_DE_PASSE'
```

Pensez à mettre à jour le fichier `.env.local` de l'application ensuite.

CHAPITRE 9

Messagerie

La messagerie repose sur **Exim** (envoi et réception), **Dovecot** (IMAP/POP3), **SpamAssassin** et **ClamAV** (antispam et antivirus), et **Roundcube** (webmail). HestiaCP signe les mails avec **DKIM**. Les enregistrements **SPF** et **DMARC** viennent déjà du modèle DNS.

PRÉREQUIS

- La zone DNS du domaine existe (chapitre 6). Sinon la clé DKIM ne sera pas publiée.
- Le PTR de `<IP>` vaut `<FQDN>` (chapitre 1).

Étape 59 Créer le domaine de messagerie

Antispam, antivirus et DKIM (2048 bits) sont activés par défaut :

```
sudo v-add-mail-domain <CLIENT> <DOMAINE>
sudo v-list-dns-records <CLIENT> <DOMAINE> | grep -E "_domainkey|_dmarc|spf"
```

VÉRIFICATION

Trois enregistrements TXT apparaissent : - `mail._domainkey` : `v=DKIM1; k=rsa; p=…` ; - `_dmarc` : `v=DMARC1; p=quarantine; pct=100` ; - `@` : `v=spf1 a mx ip4:<IP> -all`.

Étape 60 Activer le certificat SSL de la messagerie

Ce certificat couvre `mail.<DOMAINE>` et `webmail.<DOMAINE>`. Exim et Dovecot le présentent automatiquement aux clients mail qui se connectent à `mail.<DOMAINE>`.

```
sudo v-add-letsencrypt-domain <CLIENT> <DOMAINE> '' yes
```

VÉRIFICATION

`https://webmail.<DOMAINE>` affiche la page de connexion Roundcube avec un cadenas valide.

Étape 61 Créer les boîtes mail

```
 sudo v-add-mail-account <CLIENT> <DOMAINE> contact 'MOT_DE_PASSE_MAIL'
 sudo v-add-mail-account <CLIENT> <DOMAINE> noreply 'MOT_DE_PASSE_NOREPLY'
sudo v-list-mail-accounts <CLIENT> <DOMAINE>
```

Ajouter un alias ou une redirection :

```
sudo v-add-mail-account-alias <CLIENT> <DOMAINE> contact info
sudo v-add-mail-account-forward <CLIENT> <DOMAINE> contact adresse@externe.com
```

Étape 62 Ajouter les enregistrements SRV (domaine principal)

Les enregistrements SRV aident les logiciels de messagerie à se configurer seuls. Le modèle `child-ns` du domaine principal n'en contient pas ; les autres domaines, créés avec le modèle `default`, les ont déjà.

```
sudo v-add-dns-record <CLIENT> <DOMAINE> _imaps._tcp SRV '0 993 mail.<DOMAINE>.' 1
sudo v-add-dns-record <CLIENT> <DOMAINE> _submission._tcp SRV '0 587 mail.<DOMAINE>.' 1
sudo v-add-dns-record <CLIENT> <DOMAINE> _submissions._tcp SRV '0 465 mail.<DOMAINE>.' 1
```

Étape 63 Configurer un logiciel de messagerie

Réglage	Valeur
Identifiant	adresse complète, ex. `contact@<DOMAINE>`
Serveur entrant (IMAP)	`mail.<DOMAINE>`, port **993**, SSL/TLS
Serveur sortant (SMTP)	`mail.<DOMAINE>`, port **465** SSL/TLS (ou 587 STARTTLS)
Authentification	mot de passe normal
Webmail	`https://webmail.<DOMAINE>`

Étape 64 Tester l'envoi et la réception

Vérifiez d'abord que le port 25 sortant n'est pas bloqué par l'hébergeur :

```
nc -vz -w 5 aspmx.l.google.com 25
```

VÉRIFICATION

`Connection to aspmx.l.google.com 25 port [tcp/smtp] succeeded!`

Envoyez ensuite un mail depuis le webmail :

1. vers l'adresse fournie par https://www.mail-tester.com : visez **10/10** ;
2. vers une adresse Gmail : dans **Afficher l'original**, SPF, DKIM et DMARC doivent être **PASS**.

Répondez aussi depuis Gmail pour tester la réception.

Étape 65 Utiliser Spamhaus avec une clé gratuite (recommandé)

Spamhaus bloque les requêtes qui passent par les résolveurs DNS publics, ce qui rend son filtre antispam inopérant et peut fausser les résultats. La documentation HestiaCP recommande un compte gratuit **Spamhaus DQS** (https://www.spamhaus.com/free-trial/sign-up-for-a-free-data-query-service-account/), qui fournit une clé.

```
sudo v-list-sys-mail-dnsbl
sudo v-delete-sys-mail-dnsbl zen.spamhaus.org
sudo v-add-sys-mail-dnsbl VOTRE_CLE.zen.dq.spamhaus.net
```

Étape 66 Envoyer des mails depuis une application (Symfony...)

Méthode recommandée : envoi authentifié avec une vraie boîte, par exemple `noreply`. Le certificat correspond au nom utilisé, et les limites d'envoi par compte s'appliquent. Dans `.env.local` :

```
MAILER_DSN=smtp://noreply%40<DOMAINE>:MOT_DE_PASSE_NOREPLY@mail.<DOMAINE>:587
```

À SAVOIR

Dans le DSN, `@` s'écrit `%40`. Tout caractère spécial du mot de passe doit aussi être encodé : `%23` pour `#`, `%2F` pour `/`, `%3A` pour `:`.

Alternative sans mot de passe : Exim accepte les mails envoyés depuis `127.0.0.1` sans authentification. Désactivez le TLS automatique, car le certificat ne correspond pas à `127.0.0.1` :

```
MAILER_DSN=smtp://127.0.0.1:25?auto_tls=false
```

ATTENTION

N'utilisez pas `sendmail://` : les fonctions PHP qu'il utilise (`proc_open`...) sont désactivées dans PHP-FPM par HestiaCP. N'utilisez pas `localhost` non plus : Exim n'écoute qu'en IPv4, alors que `localhost` peut se résoudre en IPv6 (`::1`).

LIMITE D'ENVOI

Par défaut, chaque compte peut envoyer **200 mails par heure**. Pour une application qui envoie beaucoup : `sudo v-change-mail-account-rate-limit <CLIENT> <DOMAINE> noreply 1000`.

Étape 67 Configuration automatique pour Thunderbird (facultatif)

HestiaCP ne gère pas l'autoconfiguration. Thunderbird la cherche sur `autoconfig.<domaine>` : un petit site statique suffit.

```
sudo v-add-dns-record <CLIENT> <DOMAINE> autoconfig A <IP>
sudo v-add-web-domain <CLIENT> autoconfig.<DOMAINE>
sudo v-add-letsencrypt-domain <CLIENT> autoconfig.<DOMAINE>
sudo mkdir -p /home/<CLIENT>/web/autoconfig.<DOMAINE>/public_html/mail
sudo tee /home/<CLIENT>/web/autoconfig.<DOMAINE>/public_html/mail/config-v1.1.xml > /dev/null <<
'EOF'
<?xml version="1.0" encoding="UTF-8"?>
<clientConfig version="1.1">
  <emailProvider id="<DOMAINE>">
    <domain><DOMAINE></domain>
    <displayName><DOMAINE></displayName>
    <incomingServer type="imap">
      <hostname>mail.<DOMAINE></hostname>
      <port>993</port>
      <socketType>SSL</socketType>
      <username>%EMAILADDRESS%</username>
      <authentication>password-cleartext</authentication>
    </incomingServer>
    <outgoingServer type="smtp">
      <hostname>mail.<DOMAINE></hostname>
      <port>465</port>
      <socketType>SSL</socketType>
      <username>%EMAILADDRESS%</username>
      <authentication>password-cleartext</authentication>
    </outgoingServer>
  </emailProvider>
</clientConfig>
EOF
sudo chown -R <CLIENT>:<CLIENT> /home/<CLIENT>/web/autoconfig.<DOMAINE>/public_html/mail
```

VÉRIFICATION

`curl -s https://autoconfig.<DOMAINE>/mail/config-v1.1.xml | head -3` affiche le début du fichier XML.

CHAPITRE 10

Déployer un projet Symfony

Exemple : déployer depuis GitHub un projet Symfony sur `<SITE>`, avec sa base `<CLIENT>_app`.

Avec HestiaCP, PHP-FPM, le cron et SSH tournent **tous** sous `<CLIENT>`. Il n'y a donc plus de cron de permissions sur `var/cache`, plus d'ACL et plus de `chmod 777`.

PRÉREQUIS

- Le site `<SITE>` existe, avec SSL (chapitre 7).
- La base `<CLIENT>_app` existe (chapitre 8).
- La boîte `noreply` existe (chapitre 9).

Étape 68 Se connecter en tant que `<CLIENT>`

Depuis votre ordinateur :

```
ssh -p <SSH_PORT> <CLIENT>@<IP>
```

Toutes les commandes de ce chapitre se lancent **en** `<CLIENT>`, sauf mention contraire.

Étape 69 Créer une clé de déploiement GitHub

```
ssh-keygen -t ed25519 -f ~/.ssh/github_deploy -N "" -C "deploy@<SITE>"
cat >> ~/.ssh/config << 'EOF'
Host github.com
    IdentityFile ~/.ssh/github_deploy
    IdentitiesOnly yes
EOF
chmod 600 ~/.ssh/config
cat ~/.ssh/github_deploy.pub
```

Sur GitHub, dans le dépôt : **Settings › Deploy keys › Add deploy key**. Collez la clé affichée et laissez « Allow write access » **décoché**.

VÉRIFICATION

`ssh -T git@github.com` répond `Hi <compte>/<dépôt>! You've successfully authenticated…` (tapez `yes` à la première connexion).

Étape 70 Cloner le projet

Le dossier `public_html` contient une page d'accueil HestiaCP par défaut : videz-le, puis clonez le projet dedans.

```
cd ~/web/<SITE>/public_html
rm -f index.html robots.txt
# Remplacez COMPTE/PROJET par votre dépôt GitHub
git clone git@github.com:COMPTE/PROJET.git .
ls
```

VÉRIFICATION

Vous voyez `bin/` , `config/` , `public/` , `src/` , `composer.json` …

LE FICHIER .HTACCESS DE SYMFONY

Apache a besoin de `public/.htaccess` pour router les URL vers `index.php` . S'il n'existe pas dans votre dépôt, ajoutez-le **sur votre poste** avec `composer require symfony/apache-pack` , puis commitez-le.

Étape 71 Configurer l'environnement de production

Générez un secret :

```
php -r 'echo bin2hex(random_bytes(16)), PHP_EOL;'
```

Relevez la version exacte de MariaDB :

```
mariadb --version
```

Créez `.env.local` avec vos valeurs. `serverVersion` reprend la version affichée, par exemple `11.8.3-MariaDB` :

```
cat > .env.local << 'EOF'
APP_ENV=prod
APP_DEBUG=0
APP_SECRET=collez_le_secret_ici
DATABASE_URL="mysql://<CLIENT>_app:MOT_DE_PASSE_BDD@127.0.0.1:3306/<CLIENT>_app?
serverVersion=11.8.3-MariaDB&charset=utf8mb4"
MAILER_DSN=smtp://noreply%40<DOMAINE>:MOT_DE_PASSE_NOREPLY@mail.<DOMAINE>:587
EOF
chmod 600 .env.local
```

Étape 72 Installer les dépendances et la base

```
composer install --no-dev --optimize-autoloader
php bin/console doctrine:migrations:migrate --no-interaction
php bin/console cache:clear
```

VÉRIFICATION

`php bin/console about` affiche `Environment: prod` et `Debug: false` .

Étape 73 Pointer le site sur le dossier public/

À faire **avec** `<USER>` (dans une autre session SSH). Le dernier argument `yes` est indispensable : sans lui, PHP n'aurait plus accès à `vendor/`, `var/` ni `config/`, et le site planterait.

```
sudo v-change-web-domain-docroot <CLIENT> <SITE> <SITE> public yes
```

VÉRIFICATION

`curl -sI https://<SITE> | head -1` renvoie `HTTP/2 200`, et la page d'accueil de l'application s'affiche dans le navigateur.

ASTUCE

Dans le panel, c'est **Modifier le domaine › Options avancées › Racine de document personnalisée**, avec **Pointer vers** = `<SITE>` et **Répertoire** = `public`.

Étape 74 Créer le script de déploiement

En `<CLIENT>`, créez `~/deploy-<SITE>.sh`. Il utilise des chemins complets, car les alias (`composer`, `php`) ne fonctionnent pas dans un script.

```
cat > ~/deploy-<SITE>.sh << 'EOF'
#!/bin/bash
set -euo pipefail
PHP=/usr/bin/php8.4
COMPOSER="$PHP $HOME/.composer/composer"
cd "$HOME/web/<SITE>/public_html"

git pull --ff-only
$COMPOSER install --no-dev --optimize-autoloader --no-interaction
$PHP bin/console doctrine:migrations:migrate --no-interaction --allow-no-migration
$PHP bin/console cache:clear
echo "Déploiement terminé : $(git log -1 --format='%h %s')"
EOF
chmod 700 ~/deploy-<SITE>.sh
```

Pour déployer une nouvelle version, depuis votre ordinateur :

```
ssh -p <SSH_PORT> <CLIENT>@<IP> ./deploy-<SITE>.sh
```

À SAVOIR

OPcache vérifie la date des fichiers : le nouveau code est pris en compte en quelques secondes, sans redémarrer PHP-FPM.

Étape 75 Ajouter les tâches planifiées

À faire avec `<USER>`. Les `*` doivent être entre guillemets. Exemple : une commande toutes les 5 minutes, et le worker Messenger, relancé toutes les 5 minutes pendant 4 min 50 s.

```
sudo v-add-cron-job <CLIENT> '*/5' '*' '*' '*' '*' "/usr/bin/php8.4 /home/<CLIENT>/web/<SITE>/
public_html/bin/console app:ma-commande"
sudo v-add-cron-job <CLIENT> '*/5' '*' '*' '*' '*' "/usr/bin/php8.4 /home/<CLIENT>/web/<SITE>/
public_html/bin/console messenger:consume async --time-limit=290 --quiet"
sudo v-list-cron-jobs <CLIENT>
```

Étape 76 Connaître les limites de PHP-FPM

Pour la sécurité, HestiaCP désactive certaines fonctions dans le PHP **web** (PHP-FPM) : `exec`, `system`, `shell_exec`, `passthru`, `proc_open`, `popen`. Tout ce qui lance un processus depuis une page web échouera, par exemple le composant Process ou le transport mail `sendmail://`. Faites ces traitements en console (cron, Messenger), où ces fonctions restent disponibles.

Les chemins accessibles à PHP-FPM (`open_basedir`) se limitent à `public_html/`, `private/`, `~/tmp` et `~/.composer`. Gardez `var/` (cache, logs) **dans** le projet. Pour les fichiers envoyés par les utilisateurs, `public_html/public/uploads` ou `private/` conviennent.

> **JOURNAUX**
> Erreurs Symfony : `~/web/<SITE>/public_html/var/log/prod.log`. Erreurs Apache/PHP : `~/web/<SITE>/logs/`.

CHAPITRE 11

Docker et applications en conteneurs

Docker fait tourner des applications prêtes à l'emploi : Portainer, Uptime Kuma, Gitea, Vaultwarden… Chaque conteneur écoute **uniquement sur 127.0.0.1**, et Nginx de HestiaCP le publie sur un sous-domaine avec un certificat Let's Encrypt.

DOCKER CONTOURNE LE PARE-FEU

Un port publié par Docker (`-p 9000:9000`) passe **avant** le pare-feu de HestiaCP et devient accessible depuis Internet. Publiez toujours sur `127.0.0.1:` ; la configuration ci-dessous l'impose par défaut.

Étape 77 Installer Docker (dépôt officiel)

Ces commandes viennent de la documentation officielle de Docker pour Ubuntu :

```
sudo apt update
sudo apt install -y ca-certificates curl
sudo install -m 0755 -d /etc/apt/keyrings
sudo curl -fsSL https://download.docker.com/linux/ubuntu/gpg -o /etc/apt/keyrings/docker.asc
sudo chmod a+r /etc/apt/keyrings/docker.asc

sudo tee /etc/apt/sources.list.d/docker.sources <<EOF
Types: deb
URIs: https://download.docker.com/linux/ubuntu
Suites: $(. /etc/os-release && echo "${UBUNTU_CODENAME:-$VERSION_CODENAME}")
Components: stable
Architectures: $(dpkg --print-architecture)
Signed-By: /etc/apt/keyrings/docker.asc
EOF

sudo apt update
sudo apt install -y docker-ce docker-ce-cli containerd.io docker-buildx-plugin docker-compose-
plugin
```

VÉRIFICATION

`sudo docker run --rm hello-world` affiche `Hello from Docker!` , et `docker compose version` affiche une version 2.x ou plus.

Étape 78 Configurer Docker : localhost et rotation des journaux

Ce fichier fait publier les ports sur `127.0.0.1` par défaut, et limite la taille des journaux des conteneurs, qui peuvent sinon remplir le disque :

```
sudo tee /etc/docker/daemon.json > /dev/null << 'EOF'
{
  "ip": "127.0.0.1",
  "log-driver": "json-file",
  "log-opts": { "max-size": "10m", "max-file": "3" }
}
EOF
sudo systemctl restart docker
sudo systemctl enable docker containerd
```

Autorisez votre utilisateur à utiliser Docker sans `sudo`, puis déconnectez-vous et reconnectez-vous :

```
sudo usermod -aG docker <USER>
```

ATTENTION

Le groupe `docker` donne des droits équivalents à root. Ne l'accordez qu'à `<USER>`, jamais à `<CLIENT>`.

PARE-FEU HESTIACP ET DOCKER

`v-update-firewall` ne touche pas aux règles de Docker. En revanche, `sudo systemctl restart hestia-iptables` recharge un ancien instantané des règles : lancez `sudo systemctl restart docker` juste après.

Étape 79 Créer le générateur de modèles de proxy

HestiaCP publie un site à l'aide de **modèles Nginx**. Ce petit script crée un modèle `proxy-<port>` qui redirige un domaine vers `127.0.0.1:<port>`. Il gère le HTTPS forcé, HSTS, Let's Encrypt et les WebSockets.

D'abord, la variable Nginx nécessaire aux WebSockets :

```
sudo tee /etc/nginx/conf.d/websocket-map.conf > /dev/null << 'EOF'
map $http_upgrade $connection_upgrade {
    default upgrade;
    ''      close;
}
EOF
sudo nginx -t && sudo systemctl reload nginx
```

Puis le générateur :

```
sudo tee /usr/local/sbin/hestia-proxy-tpl > /dev/null << 'SCRIPT'
#!/bin/bash
# Usage : hestia-proxy-tpl PORT  -> crée le modèle proxy Nginx "proxy-PORT"
set -euo pipefail
PORT="${1:?Usage : hestia-proxy-tpl PORT}"
[[ "$PORT" =~ ^[0-9]{2,5}$ ]] || { echo "Port invalide : $PORT"; exit 1; }
DIR=/usr/local/hestia/data/templates/web/nginx
NAME="proxy-$PORT"

LOCATION='
    location / {
        proxy_pass         http://127.0.0.1:__PORT__;
        proxy_http_version 1.1;
        proxy_set_header   Host              $host;
        proxy_set_header   X-Real-IP         $remote_addr;
        proxy_set_header   X-Forwarded-For   $proxy_add_x_forwarded_for;
        proxy_set_header   X-Forwarded-Proto $scheme;
        proxy_set_header   X-Forwarded-Host  $host;
        proxy_set_header   Upgrade           $http_upgrade;
        proxy_set_header   Connection        $connection_upgrade;
        proxy_read_timeout 3600s;
        proxy_send_timeout 3600s;
        client_max_body_size 512m;
    }

    location /error/ {
        alias %home%/%user%/web/%domain%/document_errors/;
    }
'

cat > "$DIR/$NAME.tpl" << EOF
# Modèle généré par hestia-proxy-tpl : proxy vers 127.0.0.1:__PORT__
server {
    listen      %ip%:%proxy_port%;
    server_name %domain_idn% %alias_idn%;
    access_log  /var/log/%web_system%/domains/%domain%.log combined;
    error_log   /var/log/%web_system%/domains/%domain%.error.log error;

    include %home%/%user%/conf/web/%domain%/nginx.forcessl.conf*;
$LOCATION
    include %home%/%user%/conf/web/%domain%/nginx.conf_*;
}
EOF

cat > "$DIR/$NAME.stpl" << EOF
# Modèle généré par hestia-proxy-tpl : proxy vers 127.0.0.1:__PORT__
server {
    listen      %ip%:%proxy_ssl_port% ssl;
    server_name %domain_idn% %alias_idn%;
    access_log  /var/log/%web_system%/domains/%domain%.log combined;
    error_log   /var/log/%web_system%/domains/%domain%.error.log error;

    ssl_certificate     %ssl_pem%;
    ssl_certificate_key %ssl_key%;

    if (\$anti_replay = 307) { return 307 https://\$host\$request_uri; }
    if (\$anti_replay = 425) { return 425; }
```

```
    include %home%/%user%/conf/web/%domain%/nginx.hsts.conf*;
$LOCATION
    include %home%/%user%/conf/web/%domain%/nginx.ssl.conf_*;
}
EOF

sed -i "s/__PORT__/$PORT/g" "$DIR/$NAME.tpl" "$DIR/$NAME.stpl"
chmod 644 "$DIR/$NAME.tpl" "$DIR/$NAME.stpl"
echo "Modèle $NAME créé."
SCRIPT
sudo chmod 750 /usr/local/sbin/hestia-proxy-tpl
```

> **À SAVOIR**
>
> Les modèles portent des noms personnels (`proxy-9000` …) : les mises à jour de HestiaCP ne les écrasent pas. Ne modifiez jamais les modèles `default` .

Étape 80 Publier une application : la méthode

Pour chaque application, le même schéma en 5 commandes. Exemple : un conteneur sur le port `PORT` , publié sur `appli.<DOMAINE>` .

```
sudo hestia-proxy-tpl PORT
sudo v-add-dns-record <CLIENT> <DOMAINE> appli A <IP>
sudo v-add-web-domain <CLIENT> appli.<DOMAINE>
sudo v-change-web-domain-proxy-tpl <CLIENT> appli.<DOMAINE> proxy-PORT
sudo v-add-letsencrypt-domain <CLIENT> appli.<DOMAINE>
sudo v-add-web-domain-ssl-force <CLIENT> appli.<DOMAINE>
```

Étape 81 Installer Portainer (gestion de Docker)

Portainer est une interface web pour gérer vos conteneurs. On utilise son port HTTP 9000, uniquement sur localhost, derrière le proxy HTTPS :

```
docker volume create portainer_data
docker run -d --name portainer --restart=always \
  -p 127.0.0.1:9000:9000 \
  -v /var/run/docker.sock:/var/run/docker.sock \
  -v portainer_data:/data \
  portainer/portainer-ce:lts
```

Publiez-le sur `portainer.<DOMAINE>` :

```
sudo hestia-proxy-tpl 9000
sudo v-add-dns-record <CLIENT> <DOMAINE> portainer A <IP>
sudo v-add-web-domain <CLIENT> portainer.<DOMAINE>
sudo v-change-web-domain-proxy-tpl <CLIENT> portainer.<DOMAINE> proxy-9000
sudo v-add-letsencrypt-domain <CLIENT> portainer.<DOMAINE>
sudo v-add-web-domain-ssl-force <CLIENT> portainer.<DOMAINE>
```

CRÉEZ LE COMPTE TOUT DE SUITE

Ouvrez `https://portainer.<DOMAINE>` **dans les 5 minutes** et créez le compte administrateur avec un mot de passe fort. Passé ce délai, Portainer se verrouille par sécurité : relancez-le avec `docker restart portainer`.

VÉRIFICATION

`ss -tlnp | grep 9000` montre `127.0.0.1:9000` (et non `0.0.0.0:9000`), et `https://portainer.<DOMAINE>` affiche Portainer avec un cadenas valide.

Étape 82 Utiliser Docker Compose pour les autres applications

Pour une application décrite par un `compose.yaml`, placez-la dans `/opt/<appli>/` et publiez toujours ses ports sur `127.0.0.1` :

```
services:
  appli:
    image: editeur/appli:version
    restart: unless-stopped
    ports:
      - "127.0.0.1:8080:8080"
    volumes:
      - ./data:/data
```

```
cd /opt/appli && docker compose up -d
```

Suivez ensuite la méthode ci-dessus avec le port `8080`.

METTRE À JOUR UN CONTENEUR

Avec Compose : `docker compose pull && docker compose up -d`. Portainer propose aussi un bouton **Recreate › Pull latest image**.

CHAPITRE 12

Supervision

Trois niveaux, du plus simple au plus complet :

1. les outils intégrés à HestiaCP ;
2. **Uptime Kuma**, qui vérifie que vos sites répondent et vous alerte ;
3. **Netdata**, qui affiche en temps réel le processeur, la mémoire, le disque et les services.

Étape 83 Utiliser les outils intégrés à HestiaCP

- **Roue dentée** (page des services) : l'état de chaque service (Nginx, Apache, PHP-FPM, MariaDB, Exim, Dovecot, Bind…), avec redémarrage en un clic.
- **Roue dentée › Moniteur de tâches** : graphiques de charge, mémoire, réseau, web, mail et bases de données.
- **Statistiques** : bande passante et espace disque par utilisateur.

En ligne de commande :

```
sudo v-list-sys-services
sudo v-list-sys-info
df -h / && free -h
```

Étape 84 Installer Uptime Kuma (disponibilité et alertes)

Uptime Kuma teste vos sites, vos certificats SSL, vos ports et vos DNS toutes les minutes. Il vous alerte par e-mail, Telegram, Discord…

```
docker run -d --name uptime-kuma --restart=always \
  -p 127.0.0.1:3001:3001 \
  -v uptime-kuma:/app/data \
  louislam/uptime-kuma:2
```

Publiez-le sur `status.<DOMAINE>` (chapitre 11) :

```
sudo hestia-proxy-tpl 3001
sudo v-add-dns-record <CLIENT> <DOMAINE> status A <IP>
sudo v-add-web-domain <CLIENT> status.<DOMAINE>
sudo v-change-web-domain-proxy-tpl <CLIENT> status.<DOMAINE> proxy-3001
sudo v-add-letsencrypt-domain <CLIENT> status.<DOMAINE>
sudo v-add-web-domain-ssl-force <CLIENT> status.<DOMAINE>
```

Ouvrez `https://status.<DOMAINE>` et créez le compte administrateur. Ajoutez ensuite une sonde **HTTP(s)** pour chaque site, en cochant l'option d'alerte d'expiration du certificat.

Configurez les notifications, par exemple en SMTP avec la boîte `noreply@<DOMAINE>` : serveur `mail.<DOMAINE>`, port 465, sécurité SSL/TLS.

UNE SONDE EXTERNE EN PLUS

Si le serveur tombe, Uptime Kuma tombe avec lui et ne peut plus alerter. Ajoutez une surveillance **externe** gratuite, par exemple UptimeRobot ou Better Stack, au moins sur `<SITE>`.

Étape 85 Installer Netdata (métriques en temps réel)

Netdata détecte seul Nginx, Apache, PHP-FPM, MariaDB, Exim, Docker…

```
wget -O /tmp/netdata-kickstart.sh https://get.netdata.cloud/kickstart.sh
sudo sh /tmp/netdata-kickstart.sh --stable-channel --disable-telemetry --non-interactive
rm /tmp/netdata-kickstart.sh
```

Netdata écoute sur le port 19999, qui reste **fermé** par le pare-feu de HestiaCP (IPv4) et par le pare-feu IPv6 (chapitre 4). Son tableau de bord n'a pas de mot de passe : on y accède par un **tunnel SSH**, sans rien exposer.

Depuis votre ordinateur :

```
ssh -N -L 19999:127.0.0.1:19999 <HOST>
```

Laissez ce terminal ouvert et ouvrez `http://localhost:19999` dans votre navigateur.

VÉRIFICATION

Depuis l'extérieur, `curl -m 5 http://<IP>:19999` doit échouer (délai dépassé), alors que le tableau de bord s'affiche via le tunnel.

À SAVOIR

Netdata propose un compte « Cloud » facultatif : le tableau de bord local fonctionne sans. Il utilise environ 150 à 300 Mo de RAM ; pour l'arrêter, `sudo systemctl disable --now netdata`.

Étape 86 Recevoir les alertes du serveur

HestiaCP envoie ses notifications (mises à jour, sauvegardes, erreurs) à l'adresse de `<ADMIN>`. Vérifiez qu'elle est correcte :

```
sudo v-list-user <ADMIN> | grep EMAIL
```

Pour la changer : `sudo v-change-user-contact <ADMIN> nouvelle@adresse.com`.

CHAPITRE 13

Sauvegardes

HestiaCP sauvegarde chaque utilisateur **tous les jours à 05h10**. La sauvegarde contient ses sites, bases de données, boîtes mail, zones DNS, tâches cron, certificats et son dossier personnel.

Deux compléments sont indispensables :

- une **copie hors du serveur**, car une sauvegarde qui reste sur le serveur disparaît avec lui ;
- la sauvegarde de la **configuration système**, que HestiaCP ne sauvegarde pas : `/etc`, la configuration de HestiaCP, vos modèles, les volumes Docker.

Étape 87 Vérifier la sauvegarde locale

Lancez une sauvegarde manuelle de `<CLIENT>` :

```
sudo v-backup-user <CLIENT> yes
sudo v-list-user-backups <CLIENT>
ls -lh /backup/
```

VÉRIFICATION

Un fichier `<CLIENT>.AAAA-MM-JJ_HH-MM-SS.tar` apparaît dans `/backup/`. Le nombre de sauvegardes conservées est celui du forfait (7 pour `projets`).

À SAVOIR

Pour exclure un dossier volumineux (cache, fichiers temporaires) : **Sauvegardes › Exclusions**, dans le compte de l'utilisateur.

Étape 88 Préparer une clé SSH pour le serveur de sauvegarde

La copie distante se fait en SFTP vers `<BACKUP_HOST>`, avec une clé SSH dédiée, sans mot de passe :

```
sudo ssh-keygen -t ed25519 -f /root/.ssh/hestia-backup -N "" -C "backup@<FQDN>"
sudo cat /root/.ssh/hestia-backup.pub
```

Ajoutez cette clé publique au fichier `~/.ssh/authorized_keys` du compte `<BACKUP_USER>` sur `<BACKUP_HOST>`. Sur un hébergement mutualisé, passez par l'interface (cPanel : **Accès SSH › Gérer les clés SSH › Importer**, puis **Autoriser**).

Testez la connexion et créez le dossier de destination :

```
sudo ssh -i /root/.ssh/hestia-backup -p <BACKUP_PORT> <BACKUP_USER>@<BACKUP_HOST> "mkdir -p hestia-<HOST>/systeme && ls"
```

VÉRIFICATION

Répondez `yes` à la question sur l'empreinte du serveur. La commande affiche le contenu du dossier distant, dont `hestia-<HOST>`, sans demander de mot de passe.

Étape 89 Envoyer les sauvegardes HestiaCP hors du serveur

```
sudo v-add-backup-host sftp <BACKUP_HOST> <BACKUP_USER> /root/.ssh/hestia-backup hestia-<HOST>
<BACKUP_PORT>
sudo v-list-backup-host sftp
```

Les sauvegardes sont maintenant copiées aussi vers `hestia-<HOST>/` sur `<BACKUP_HOST>`, avec la même durée de conservation. Testez tout de suite :

```
sudo v-backup-user <CLIENT> yes
sudo ssh -i /root/.ssh/hestia-backup -p <BACKUP_PORT> <BACKUP_USER>@<BACKUP_HOST> "ls -lh hestia-<HO
ST>"
```

VÉRIFICATION

L'archive `<CLIENT>.….tar` du jour est présente sur le serveur distant, et `<EMAIL>` reçoit un e-mail de confirmation de la sauvegarde.

Étape 90 Sauvegarder la configuration système

Ce script copie chaque nuit vers le serveur distant (qui doit disposer de `rsync`) `/etc`, la configuration et les modèles de HestiaCP, les volumes Docker et `/opt` :

```
sudo tee /usr/local/sbin/backup-systeme.sh > /dev/null << 'EOF'
#!/bin/bash
# Copie de la configuration système vers le serveur de sauvegarde
set -euo pipefail
DEST="<BACKUP_USER>@<BACKUP_HOST>:hestia-<HOST>/systeme/"
SSH="ssh -i /root/.ssh/hestia-backup -p <BACKUP_PORT>"
SOURCES=()
for p in /etc /usr/local/hestia/conf /usr/local/hestia/data /usr/local/sbin \
         /opt /var/lib/docker/volumes /root/.ssh; do
  if [ -e "$p" ]; then SOURCES+=("$p"); fi
done
rsync -aR --delete -e "$SSH" "${SOURCES[@]}" "$DEST"
EOF
sudo chmod 700 /usr/local/sbin/backup-systeme.sh
sudo /usr/local/sbin/backup-systeme.sh && echo "Sauvegarde système OK"
```

Programmez-le chaque nuit à 04h30 (avant la sauvegarde HestiaCP) :

```
echo '30 4 * * * root /usr/local/sbin/backup-systeme.sh > /var/log/backup-systeme.log 2>&1' | sudo
tee /etc/cron.d/backup-systeme
```

ATTENTION

Ces copies contiennent des secrets : clés SSH, mots de passe de bases dans `/usr/local/hestia/conf`, fichiers `.env.local`. Le compte de sauvegarde doit être protégé comme le serveur lui-même.

Étape 91 Restaurer

Lister les sauvegardes d'un utilisateur :

```
sudo v-list-user-backups <CLIENT>
```

Restaurer **un seul** site, une base ou un domaine mail, depuis une archive locale ou distante :

```
sudo v-restore-web-domain <CLIENT> <CLIENT>.AAAA-MM-JJ_HH-MM-SS.tar <SITE>
sudo v-restore-database <CLIENT> <CLIENT>.AAAA-MM-JJ_HH-MM-SS.tar <CLIENT>_app
sudo v-restore-mail-domain <CLIENT> <CLIENT>.AAAA-MM-JJ_HH-MM-SS.tar <DOMAINE>
```

Restaurer **tout** l'utilisateur (sur ce serveur ou sur un nouveau serveur HestiaCP, où l'utilisateur sera recréé) :

```
sudo v-restore-user <CLIENT> <CLIENT>.AAAA-MM-JJ_HH-MM-SS.tar
```

TESTEZ UNE RESTAURATION

Une sauvegarde jamais testée n'est pas une sauvegarde. Une fois par trimestre, restaurez une base dans une base de test, ou le site complet sur un VPS temporaire.

POUR ALLER PLUS LOIN : SAUVEGARDES INCRÉMENTALES

Depuis la version 1.9, HestiaCP sait aussi faire des sauvegardes **incrémentales** avec restic (commande `v-add-backup-host-restic`), vers un stockage compatible rclone (S3, Backblaze B2…). Elles prennent beaucoup moins de place que des archives complètes quotidiennes. Pensez alors à sauvegarder les fichiers `/usr/local/hestia/data/users/*/restic.conf`, sans lesquels la restauration est impossible : le script `backup-systeme.sh` les inclut déjà.

CHAPITRE 14

Maintenance et dépannage

Étape 92 Faire les mises à jour mensuelles

HestiaCP et les correctifs de sécurité d'Ubuntu s'installent seuls. Une fois par mois, mettez aussi à jour le reste (PHP, Nginx, MariaDB, Docker) :

```
sudo apt update && sudo apt upgrade
[ -f /var/run/reboot-required ] && echo "Redémarrage nécessaire"
sudo v-list-sys-hestia-updates
```

ATTENTION

Si apt demande quoi faire d'un fichier de configuration modifié, répondez **N** (garder la version actuelle). HestiaCP gère lui-même ces fichiers.

Mettez à jour les conteneurs de la même façon : `docker compose pull && docker compose up -d` dans chaque dossier `/opt/<appli>`, ou le bouton de mise à jour de Portainer.

Étape 93 Mémo des commandes utiles

Besoin	Commande
État des services	`sudo v-list-sys-services`
Redémarrer un service	`sudo v-restart-service nginx` (ou `apache2`, `php-fpm`, `exim4`, `dovecot`, `mariadb`, `bind9`)
Lister les utilisateurs / sites	`sudo v-list-users` · `sudo v-list-web-domains <CLIENT>`
Reconstruire la config des sites	`sudo v-rebuild-web-domains <CLIENT>`
Reconstruire tout un utilisateur	`sudo v-rebuild-user <CLIENT>`
Pare-feu, IP bannies	`sudo v-list-firewall` · `sudo v-list-firewall-ban`
Débannir une IP	`sudo v-delete-firewall-ban <IP> ALL`
Renouveler un certificat	`sudo v-add-letsencrypt-domain <CLIENT> <domaine> www.<domaine>`
Aide d'une commande	ouvrir le script, ex. `head -12 /usr/local/hestia/bin/v-add-web-domain`

Étape 94 Où trouver les journaux

Journal	Emplacement
Actions HestiaCP	`/var/log/hestia/system.log`, `/var/log/hestia/error.log`

Journal	Emplacement
Connexions au panel	`/var/log/hestia/auth.log`
Let's Encrypt	`/var/log/hestia/LE-<utilisateur>-<domaine>.log`
Sites (Nginx / Apache)	`/var/log/nginx/domains/` , `/var/log/apache2/domains/`
Mail envoyé et reçu	`/var/log/exim4/mainlog`
IMAP / POP3	`/var/log/dovecot.log`
Fail2ban	`/var/log/fail2ban.log`
SSH	`/var/log/auth.log`
Installation	`/root/hst_install_backups/`

Étape 95 Résoudre les problèmes fréquents

Le certificat Let's Encrypt échoue

1. `dig +short <domaine>` doit répondre `<IP>`, et `dig +short AAAA <domaine>` ne doit **rien** répondre (chapitre 1).
2. Le port 80 doit être ouvert (`sudo v-list-firewall | grep WEB`).
3. Lisez le journal `/var/log/hestia/LE-<CLIENT>-<domaine>.log` .
4. Après trop d'échecs, Let's Encrypt bloque le domaine pendant une heure : attendez avant de réessayer.

Erreur 502 ou 503 sur un site

PHP-FPM est arrêté ou saturé :

```
sudo v-restart-service php-fpm
sudo tail -n 50 /var/log/apache2/domains/<SITE>.error.log
```

Erreur 500 sur une application Symfony

Connecté en `<CLIENT>` :

```
tail -n 50 ~/web/<SITE>/public_html/var/log/prod.log
cd ~/web/<SITE>/public_html && php bin/console cache:clear
```

Les mails arrivent en spam

1. Testez sur https://www.mail-tester.com et suivez ses indications.
2. Vérifiez le PTR : `dig +short -x <IP>` doit répondre `<FQDN>` .
3. Vérifiez que l'IP n'est pas sur liste noire : https://check.spamhaus.org.
4. Vérifiez SPF, DKIM et DMARC : `sudo v-list-dns-records <CLIENT> <DOMAINE>` .

Vous êtes bloqué dehors (SSH refusé)

Votre IP a sans doute été bannie par Fail2ban. Connectez-vous par la **console VNC** du panel Contabo, puis :

```
sudo v-list-firewall-ban
sudo v-delete-firewall-ban VOTRE_IP ALL
```

Le disque se remplit

```
df -h /
sudo du -sh /backup /home/* /var/lib/docker /var/log 2>/dev/null | sort -h
docker system prune
```

Réduisez si besoin le nombre de sauvegardes locales dans le forfait (`BACKUPS`), puisqu'une copie existe sur le serveur distant.

Étape 96 Vérifier que tout est en place

✔	Vérification
☐	Panel accessible en HTTPS avec certificat valide et double authentification
☐	SSH uniquement par clé, sur le port `<SSH_PORT>`, root interdit
☐	IPv6 entrante bloquée (`sudo ip6tables -S INPUT`)
☐	UFW absent, pare-feu HestiaCP et Fail2ban actifs
☐	`dig NS <DOMAINE>` renvoie `ns1` et `ns2.<DOMAINE>`
☐	PTR de `<IP>` = `<FQDN>`
☐	Sites en HTTPS forcé, sans enregistrement AAAA
☐	mail-tester.com : 10/10 ; Gmail : SPF, DKIM et DMARC en PASS
☐	Conteneurs Docker publiés uniquement sur 127.0.0.1 (`ss -tlnp`)
☐	Sauvegarde HestiaCP présente sur `<BACKUP_HOST>`
☐	Sauvegarde système (`/var/log/backup-systeme.log` sans erreur)
☐	Uptime Kuma et une surveillance externe actifs
☐	Une restauration de test réussie

ANNEXE

Annexe : adresses et régénération du guide

Étape 97 Récapitulatif des adresses

Service	Adresse
Panel HestiaCP	`https://<FQDN>:<PANEL_PORT>`
SSH administrateur	`ssh -p <SSH_PORT> <USER>@<IP>`
SSH déploiement	`ssh -p <SSH_PORT> <CLIENT>@<IP>`
Webmail	`https://webmail.<DOMAINE>`
Mail (IMAP / SMTP)	`mail.<DOMAINE>` : 993 / 465 ou 587
phpMyAdmin	`https://<FQDN>/<alias-du-chapitre-4>/`
Portainer	`https://portainer.<DOMAINE>`
Uptime Kuma	`https://status.<DOMAINE>`
Netdata	tunnel SSH puis `http://localhost:19999`
Serveurs DNS	`ns1.<DOMAINE>`, `ns2.<DOMAINE>` → `<IP>`

Étape 98 Générer ce guide pour un autre VPS

Ce PDF est produit à partir de fichiers Markdown, qui se trouvent dans le dépôt `vps-docs`, dossier `hestiacp/` :

Élément	Rôle
`chapitres/*.md`	Le contenu, un fichier par chapitre ; `{{…}}` marque une variable
`vps.conf.exemple`	Modèle de fichier de variables
`build/build.py`	Le générateur (Markdown → HTML → PDF avec WeasyPrint)
`build/style.css`, `build/fonts/`	La mise en page et les polices
`build/meta.conf`	Version de HestiaCP et date de vérification affichées en couverture

Installez une fois les outils (Ubuntu ou Debian) :

```
sudo apt install -y python3-venv libpango-1.0-0 libpangoft2-1.0-0
python3 -m venv ~/.venvs/guide
~/.venvs/guide/bin/pip install weasyprint markdown pygments
```

Puis, pour un nouveau VPS :

```
cd vps-docs/hestiacp
cp vps.conf.exemple nouveau-vps.conf
micro nouveau-vps.conf
~/.venvs/guide/bin/python build/build.py --vars nouveau-vps.conf
```

Le fichier `Guide-VPS-HestiaCP-<FQDN>.pdf` est créé, avec toutes les valeurs déjà remplies. Sans `--vars`, vous obtenez la version générique.

AVANT CHAQUE NOUVELLE INSTALLATION

HestiaCP évolue vite. Avant d'installer un nouveau serveur : - consultez les notes de version sur https://github.com/hestiacp/hestiacp/releases ; - vérifiez les options de l'installeur avec `bash hst-install.sh -h` ; - en cas de doute sur une commande `v-…`, lisez son en-tête : `head -12 /usr/local/hestia/bin/<commande>` ; - mettez à jour `build/meta.conf` après vérification.

Étape 99 Sources officielles

- Documentation HestiaCP : https://hestiacp.com/docs/
- Code source et scripts `v-*` : https://github.com/hestiacp/hestiacp
- Forum : https://forum.hestiacp.com
- Installation de Docker : https://docs.docker.com/engine/install/ubuntu/