

Préconisations techniques

1. TABLE DES MATIÈRES

1. TABLE DES MATIERES	1
2. PREAMBULE	2
2.1. OBJECTIFS DU DOCUMENT	2
2.2. VERSIONS DU DOCUMENT	2
2.3. DOCUMENTS DE REFERENCE	2
3. LE PROJET	3
3.1. OBJECTIF DU PROJET	3
4. NOS PRECONISATIONS	4
4.1. PREAMBULE.....	2
4.2. ARCHITECTURE LARAVEL + NUXT.JS.....	2
4.2.1. BRIQUES DE L'ARCHITECTURE COTE BACKEND (LARAVEL).....	2
4.2.1.1. CONFIGURATIONS SPECIFIQUES	3
4.2.2. BRIQUES DE L'ARCHITECTURE COTE FRONTEND (NUXT.JS)	3
4.2.2.1. CONFIGURATIONS SPECIFIQUES	3
4.3. ARCHITECTURE CMS.....	6
4.3.1. BRIQUES DE L'ARCHITECTURE.....	6
4.3.1.1. CONFIGURATIONS SPECIFIQUES	7
4.4. CONFIGURATIONS	7
4.4.1. MODULES PHP	7
4.4.2. SSL.	8
4.4.3. CONFIGURATIONS DIVERSES.....	8
4.5. ENVOI DE MAILS (SMTP).....	9

PRÉAMBULE

Nous aurons besoin de 2 espaces :

- Pré-production
- Production

Nous aurons besoin d'accéder aux différents serveurs via des accès SSH / SFTP afin de pouvoir déployer les fichiers source à chaque déploiement et lancer des commandes en lien avec ces déploiements (migrations BDD, purge de cache, etc.) ou ponctuelles (consultation des logs, redémarrage d'une instance PM2, etc.). Nous avons besoin d'un accès par clef SSH uniquement, voici la clef publique à autoriser :

```
ecdsa-sha2-nistp521
AAAAE2VjZHNhLXNoYTItbmlzdHA1MjEAAAABmlzdHA1MjEAAACFBAAayePqUu77mGer3ICUB2FpHuk+Q+DZ
2YRAwoPnD6hZcYSfRuM4uZuMXPhmfKbBO15F+c4jHDANYPMEbSgSrb4+QF/v93GTA3Bl/zPmll1H+yxg5Dbt
QelrDn6sSgceh+iYDrjrvYLnrexVMr5TfWfGEbEy1CIT+n1qJNK7hjH4UWZg== deploy.ext@gingerminds.fr
```

D'autres clefs publiques pourront être transmises séparément en fonction des besoins

1.1. ARCHITECTURE LARAVEL + NUXT.JS

1.1.1. Briques de l'architecture côté Backend (Laravel)

Brique technique	Solution logicielle	Version	Commentaires
OS	Distribution Linux type Debian		
Serveur Web	NGINX ou Apache	Dernière version	
Serveur d'applications	PHP	>=8.1	
Outils d'installation	Composer	2+	
Outils	Node	16.x	Seulement si pas de déploiement automatisé
Gestion de Cache en RAM	Redis	7+	
Système de Base de Données	MariaDB ou MySQL	MariaDB 10.3+ MySQL 5.7+	Créer une base de données et nous fournir les accès. Prévoir à minima 2 Go de stockage sur disque.
Serveur Mail			Autoriser l'envoi de mails depuis la plateforme ou fournir des accès SMTP

			pour envoi de mails transactionnels
Outils d'administration	PhpMyAdmin ou Adminer	Dernière version	Activation de filtrage d'accès possible (IP par ex.) et/ou basic auth

1.1.2. Configurations spécifiques

NGinx/Apache

- Autoriser l'utilisation du htaccess
- Autoriser le mod_rewrite
- Modifier la racine du serveur Web (DocumentRoot) vers /current/public
- Autoriser toutes les méthodes HTTP dont OPTIONS

Redis

- Maxmemory_policy : allkeys-lfu (<https://redis.io/docs/reference/eviction/>)

Systeme

- Autoriser les méthodes suivantes : proc_open, proc_close, curl_exec, exec

DNS/Apache

- Ajout de 2 sous-domaines "api" et "media" qui sont des alias pointant vers le vhost. L'idée sera d'utiliser ces alias respectivement pour les échanges avec l'API et pour la récupération des medias (visuels, brochures, etc.) et ainsi éviter d'afficher en clair le sous-domaine du back-office.

1.1.3. Briques de l'architecture côté Frontend (Nuxt.js)

Brique technique	Solution logicielle	Version	Commentaires
OS	Distribution Linux type Debian		
Serveur Web	Apache	2+	
Serveur d'applications	Node	18.x	

1.1.4. Configurations spécifiques

NGinx/Apache

- Autoriser l'utilisation du htaccess
- Autoriser le mod_rewrite
- Modifier la racine du serveur Web (DocumentRoot) vers /current/public

> Si vous utilisez Apache, il faudra configurer un reverse proxy comme suit (le port 6001 peut être adapté) :

```
ProxyPass « / » http://127.0.0.1:3000/
ProxyPassReverse « / » http://127.0.0.1:3000/
```

> Si vous utilisez NGINX qui est utilisé il faut suivre les recommandations de la doc suivante pour configurer le reverse proxy :

<https://nuxtjs.org/deployments/nginx/>

Dans les 2 cas Il faudra nous fournir le port sur lequel devra écouter l'application NodeJS (dans l'exemple ci-dessus "3000") afin de le paramétrer dans le fichier "ecosystem.config.js" par nos soins.

Node

- PM2 installé globalement avec :
 - L'option « watcher » activé
 - Mécanisme de redémarrage automatique du service (en cas de redémarrage du serveur par exemple) :
- <https://pm2.keymetrics.io/docs/usage/startup/>