

FICHE PROJET : Sécurisation d'un Serveur et Déploiement d'un chatbot IA

1. Contexte et Objectifs

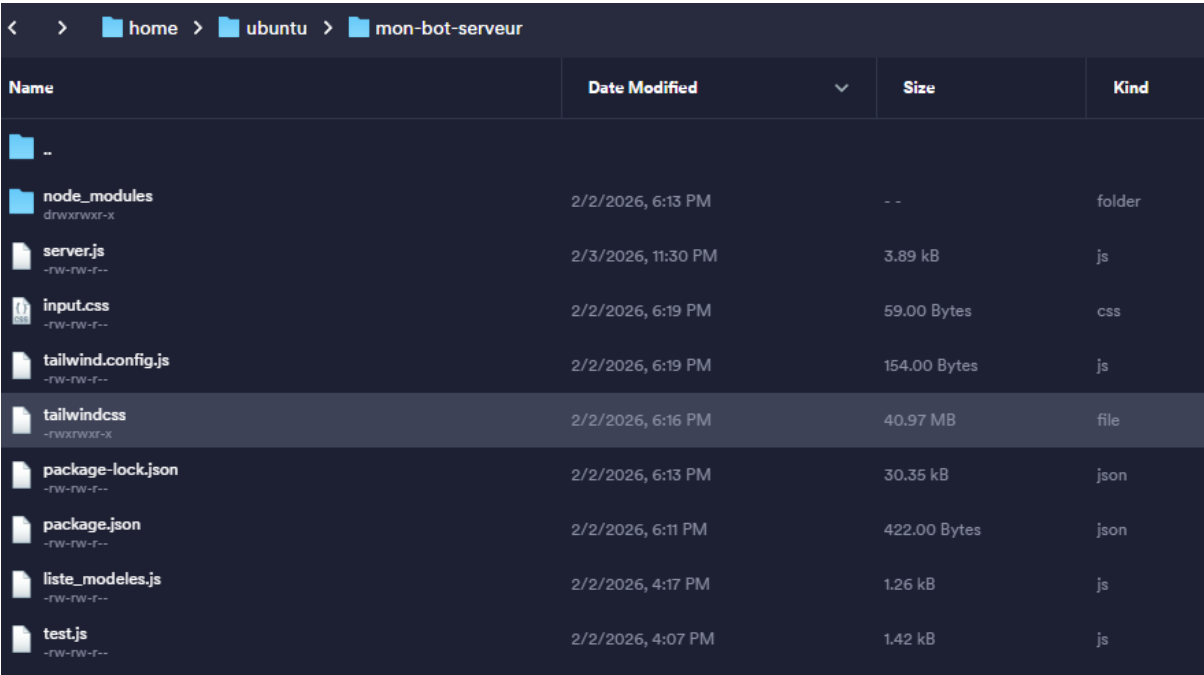
L'objectif de ce projet était de déployer un agent conversationnel (Chatbot IA) capable d'interagir avec les utilisateurs sur une interface web. La problématique principale relevait de la **cybersécurité** : comment intégrer une API externe (Google Gemini) sans exposer les clés d'authentification aux attaques côté client, tout en assurant la haute disponibilité du service sur un serveur distant.

2. Configuration et Administration du Serveur OVH (VPS)

L'infrastructure applicative repose sur un Serveur Privé Virtuel (VPS) (sous distribution Linux Ubuntu).

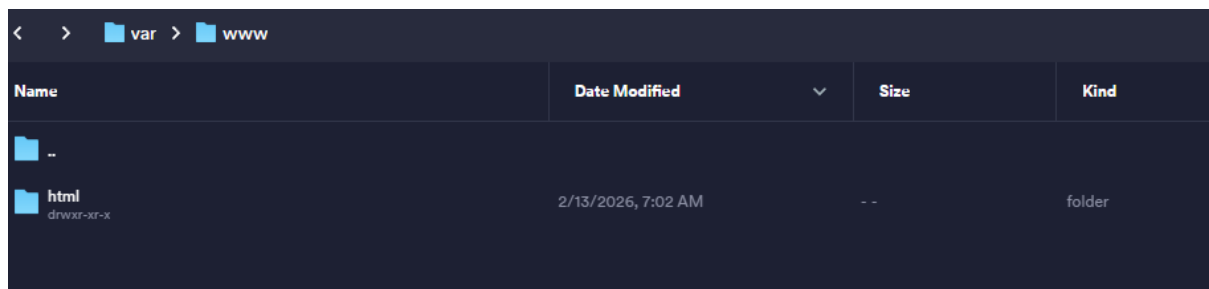
- **Mise en place de l'environnement** : Installation et configuration de l'environnement d'exécution Node.js pour héberger le serveur applicatif (Backend).

Backend :



Name	Date Modified	Size	Kind
..			
node_modules drwxrwxr-x	2/2/2026, 6:13 PM	- -	folder
server.js -rw-rw-r--	2/3/2026, 11:30 PM	3.89 kB	js
input.css -rw-rw-r--	2/2/2026, 6:19 PM	59.00 Bytes	css
tailwind.config.js -rw-rw-r--	2/2/2026, 6:19 PM	154.00 Bytes	js
tailwindcss -rwxrwxr-x	2/2/2026, 6:16 PM	40.97 MB	file
package-lock.json -rw-rw-r--	2/2/2026, 6:13 PM	30.35 kB	json
package.json -rw-rw-r--	2/2/2026, 6:11 PM	422.00 Bytes	json
liste_modeles.js -rw-rw-r--	2/2/2026, 4:17 PM	1.26 kB	js
test.js -rw-rw-r--	2/2/2026, 4:07 PM	1.42 kB	js

Frontend:



- **Haute Disponibilité (Gestionnaire de processus) :** Utilisation de **PM2**. Le service applicatif a été pour s'assurer que le serveur Node.js fonctionne en arrière-plan de manière continue. En cas de crash, PM2 assure le redémarrage automatique du processus (`pm2 restart all`).

```
ubuntu@> $ pm2 restart all
Use --update-env to update environment variables
[PM2] Applying action restartProcessId on app [all](ids: [ 0 ])
[PM2] [bot-backend](0) ✓
```

id	name	namespace	version	mode		uptime	↑	status	cpu	mem	user	
0	bot-backend	default	1.0.0	fork		0s	1	online	0%	26.0mb	ubuntu	

3. Architecture Sécurisée du Chatbot IA

Le chatbot exploite un modèle linguistique externe (LLM). Pour répondre aux exigences de sécurité , une architecture **Client-Serveur (Proxy Backend)** a été développée.

Isolation des identifiants (Gestion des secrets) :

- *Vulnérabilité identifiée :* Une intégration Front-End classique (JavaScript exécuté dans le navigateur) expose la clé d'API au vol via l'inspection du code source.
- *Remédiation :* Déplacement de la logique d'appel API sur le serveur VPS. Stockage sécurisé de la clé d'authentification dans des **variables d'environnement** via un fichier `.env`, cloisonné et inaccessible depuis l'extérieur.

```
GNU nano 7.2
API_KEY=AIzaSyA6bC[REDACTED]
```

- **Mise en place d'un Serveur Relais (Proxy) :** L'interface utilisateur ne communique plus directement avec l'API externe. Elle envoie la requête au serveur, qui y injecte la clé secrète de manière transparente, interroge l'API, puis renvoie la réponse filtrée au client.
- **Prévention des abus réseaux :** Gestion des appels API côté serveur permettant de limiter les requêtes, prévenant ainsi les attaques de type "Denial of Wallet" (épuisement volontaire et malveillant des quotas d'API de l'infrastructure).

home ubuntu mon-bot-serveur				
Name	Date Modified	Size	Kind	
..				
node_modules	2/2/2026, 6:13 PM	-	folder	drwxrwxr-x
server.js	2/3/2026, 11:30 PM	3.89 kB	js	-rw-rw-r--
input.css	2/2/2026, 6:19 PM	59.00 Bytes	css	-rw-rw-r--
tailwind.config.js	2/2/2026, 6:19 PM	154.00 Bytes	js	-rw-rw-r--
tailwindcss	2/2/2026, 6:16 PM	40.97 MB	file	-rwxrwxr-x
package-lock.json	2/2/2026, 6:13 PM	30.35 kB	json	-rw-rw-r--
package.json	2/2/2026, 6:11 PM	422.00 Bytes	json	-rw-rw-r--
liste_modeles.js	2/2/2026, 4:17 PM	1.26 kB	js	-rw-rw-r--
test.js	2/2/2026, 4:07 PM	1.42 kB	js	-rw-rw-r--

4. Compétences SISR Mobilisées

Administrer des systèmes et des réseaux (configuration VPS; Ubuntu, environnement Linux).

Mettre en place une architecture applicative sécurisée (Architecture Proxy/Backend).

Protéger l'accès aux ressources (Variables d'environnement `.env`, masquage de clés API).

Superviser l'infrastructure (Analyse des logs, PM2, requêtes HTTP, tâches automatisées).

Pour tester le bot : <http://213.32.20.241:8080/>

