

# Documentation Technique : Déploiement de mon API NodeJS Pour mon NoSQL.

---

## Introduction

Cette documentation détaille les étapes nécessaires pour déployer une API Node.js sur un serveur hébergé chez O2Switch en utilisant l'interface cPanel.

## Prérequis

Avant de commencer le processus de déploiement, assurez vous d'avoir les éléments suivants :

1. Accès au cPanel de votre compte chez O2Switch.
2. Une API Node.js prête à être déployée, avec tous ses fichiers et dépendances.

## Étapes du déploiement

### 1. Accéder à cPanel

Connectez vous à votre compte O2Switch et accédez à l'interface cPanel.

### 2. Créer un sous-domaine (optionnel)

Si vous souhaitez déployer votre API sur un sous-domaine spécifique, créez-le via l'option "Sous-domaines" dans cPanel.

### 3. Accéder à l'outil "Gestionnaire de fichiers"

Dans cPanel, cherchez l'option "Gestionnaire de fichiers" et ouvrez le.



## 4. Télécharger les fichiers de votre API

Téléversez les fichiers de votre API Node.js dans le répertoire racine de votre serveur, ou dans le répertoire correspondant au sous-domaine que vous avez créé.

**Chargement de fichiers**

Sélectionnez le fichier à charger sur « /home/nizu7869/arcadia.thrpub.fr ».

Taille maximale de fichier autorisée pour le chargement : ∞

☐ Écraser les fichiers existants

Déposez les fichiers ici pour démarrer le chargement

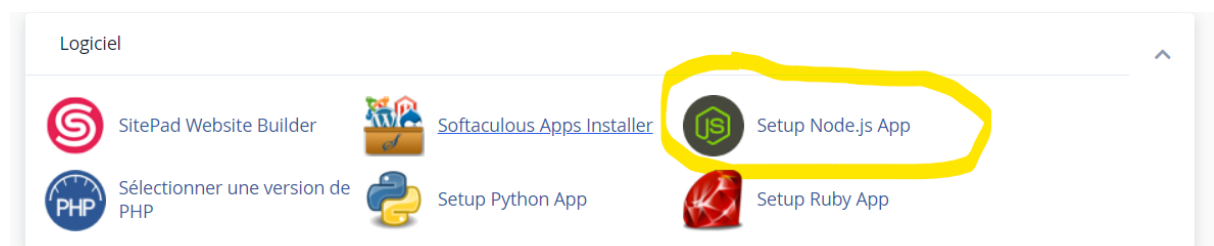
ou

[Sélectionner un fichier](#)

[Revenir à « /home/nizu7869/arcadia.thrpub.fr »](#)

## 5. Installer les dépendances

Si votre API utilise des dépendances Node.js, assurez-vous d'installer celles-ci depuis l'app NodeJS du Cpanel.





Node.js

WEB APPLICATIONS

[+ CREATE APPLICATION](#)

App URI	App Root Directory	Mode	Status	Actions
<a href="#">arcadia.thrpub.fr/</a>	/home/nizu7869/arcadia.thrpub.fr	development	● started (v19.9.0)	

WEB APPLICATIONS [ARCADIA.THRPUB.FR/](#)

DESTROY

Node.js version 19.9.0

Application mode Development

Adds value for NODE\_ENV variable

Application root

It is a physical address to your application on a server that corresponds with its URI. Upload your application files here.

Application URL  [OPEN](#)

It is an HTTP/HTTPS link to your application

Application startup file

Passenger log file

You can define the path along with the filename (e.g. /home/nizu7869/logs/passenger.log)

Detected configuration files

package.json

Run NPM Install

Run JS script

Edit

## 6. Configurer les paramètres de démarrage

Si nécessaire, créez un fichier de configuration pour votre API (par exemple, un fichier `.env` pour les variables d'environnement) et assurez-vous qu'il est correctement configuré pour votre environnement de production.

Ici on l'intègre en bas de l'app.

Environment variables

[+ ADD VARIABLE](#)

Name	Value	Actions
MONGODB_URL	mongodb+srv://THRPUb:Azerty11%40@arcadia.s6c3pnq.mongodb.net/animalDB	
PORT	4000	

## 7. Autoriser l'accès au IP du serveur sur MongoDB ATLAS

MongoDB Atlas est une plateforme de base de données cloud entièrement gérée qui permet aux développeurs de déployer, de gérer et d'évoluer facilement des clusters MongoDB sur des infrastructures cloud, sans se soucier de la complexité opérationnelle.

Ici j'ai donné l'accès a toutes les IP pour ne pas être embêter.

THRSOCIETE'S ORG - 2024-05-06 > PROJECT 0

## Network Access

IP Access List

Peering

Private Endpoint

+ ADD IP ADDRESS

You will only be able to connect to your cluster from the following list of IP Addresses:

IP Address	Comment	Status	Actions
0.0.0.0/0 (includes your current IP address)		<div><div></div>Active</div>	<div><div>EDIT</div><div>DELETE</div></div>

secondes pour me dire qu'il n'arrive pas à se connecter à cause d'un soucis d'adresse IP :

```
Serveur en cours d'exécution sur le port 4000
Erreur de connexion à MongoDB MongooseServerSelectionError: Could not connect to any servers in your MongoDB Atlas cluster. One common reason is that you're trying to access the database from an IP that isn't whitelisted. Make sure your current IP address is on your Atlas cluster's IP whitelist: https://www.mongodb.com/docs/atlas/security-whitelist/
    at handleConnectionErrors (/home/nizu7869/nodeenv/arcadia.thrpub.fr/19/lib/node_modules/mongoose/lib/connection.js:875:11)
    at NativeConnection.openUri (/home/nizu7869/nodeenv/arcadia.thrpub.fr/19/lib/node_modules/mongoose/lib/connection.js:826:11) {
  reason: TopologyDescription {
    type: 'ReplicaSetNoPrimary',
    servers: Map(3) {
      'ac-oey8uza-shard-00-02.s6c3pnq.mongodb.net:27017' => [ServerDescription],
      'ac-oey8uza-shard-00-00.s6c3pnq.mongodb.net:27017' => [ServerDescription],
      'ac-oey8uza-shard-00-01.s6c3pnq.mongodb.net:27017' => [ServerDescription]
    },
    stale: false,
    compatible: true,
    heartbeatFrequencyMS: 10000,
    localThresholdMS: 15,
    setName: 'atlas-jnlrnd-shard-0',
    maxElectionId: null,
    maxSetVersion: null,
    commonWireVersion: 0,
    logicalSessionTimeoutMinutes: null
  },
  code: undefined
}
```

J'ai bien sur vérifier que ma DB sur MongoDB Atlas accepte bien toutes les IP (Voir image si dessus).

J'ai essayer de nombreuses possibilités mais rien ne fonctionne. Si cela me serai arriver pour un vrais client, je lui aurait proposé d'implémenter un compteur de clic dans une base de donnée SQL classique comme pour le reste de mon api Symfony.

## Conclusion

Voici les étapes, pour déployer avec succès mon API Node.js sur un serveur hébergé chez O2Switch via l'interface cPanel.