

# **Projet Azure Cloud Computing (Examen E4) : Dossier de Conception Détailé**

## **Contexte et Objectifs**

Ce document détaille la conception architecturale et le plan d'action pour le déploiement des ressources Azure, conformément aux exigences de l'examen E4 (Décembre 2025).

### **Objectifs de l'Examen**

Exigence	Service Cible	Configuration	Note (Max)
<b>1. Hébergement Statique</b>	VM Windows Server 2025	Page HTML simple via IIS (Port 80).	4
<b>2. Base de Données</b>	Azure Database for MySQL	Base de données PaaS (apdb) pour l'application.	4
<b>3. Déploiement Applicatif</b>	VM Ubuntu Server	Application conteneurisée (Node.js/React) via Docker (Ports 3000, 5000).	4
<b>Bonus</b>	Azure App Service	Déploiement en PaaS de l'application (en parallèle).	3

## **Architecture Générale**

L'architecture est une solution hybride IaaS/PaaS, intégrant des composants de mise à l'échelle et de résilience, avec un point d'accès unique via un équilibrEUR de charge.

### **1.1 Composants Clés**

- Azure Load Balancer (Standard)** : Point d'entrée unique.
- VNet & Subnet** : Réseau privé pour l'infrastructure IaaS.

- **Deux VMs** : Windows Server (Hébergement Statique) et Ubuntu (Application Docker).
- **Azure Database for MySQL** : Base de données gérée PaaS.
- **Azure Backup** : Solution de continuité d'activité pour les VMs.

## II. Détail des Services Réseau et Sécurité

Service	Rôle Principal	Nom / SKU	Configuration Spécifique
VNet	Réseau Privé Logique	VNET-Project-Azure	Préfixe d'Adresse : 10.10.0.0/16
Subnet	Réseau des serveurs	SUBNET-Project-Azure	Préfixe d'Adresse : 10.10.1.0/24
Azure Load Balancer	Distribution du trafic	LB-Project-Azure (SKU Standard)	<b>Pool Backend</b> : VM Windows et VM Ubuntu. <b>IP Frontend</b> : Publique, Statique.
NSG	Pare-feu de Sous-réseau / NIC	NSG-Project-Azure	<b>Règles Inbound Essentielles (Source : Load Balancer / IP Admin)</b> : Port 22 (SSH), Port 3389 (RDP), Port 80 (HTTP), Port 3000 (Backend), Port 5000 (Frontend).

### III. Détail des Ressources de Calcul (VMs)

Machine Virtuelle	Rôle	Image / SKU	Configuration Applicative
<b>VM-WINDOWS-01</b>	<b>Hébergement Statique (Exigence 1)</b>	Windows Server 2025 Datacenter Gen 2 / Standard_B2s	Installation de IIS (Web Server Role). Déploiement du fichier <code>index.html</code> .
<b>VM-UBUNTU-01</b>	<b>Application Conteneurisée (Exigence 3)</b>	Ubuntu 22.04 LTS / Standard_B2s	Installation de <b>Docker Engine</b> via script Bash. Déploiement de l'image Docker (React/Node.js).

**Note sur le Load Balancer :** Le trafic HTTP/80 sera dirigé vers le service IIS de la VM Windows, tandis que les ports 3000/5000 seront dirigés vers l'application Docker de la VM Ubuntu, le Load Balancer agissant comme un aiguilleur simple dans ce contexte.

## IV. Détail de la Base de Données et de la Résilience

Service	Rôle	Configuration	Justification
<b>Azure Database for MySQL</b>	Base de Données PaaS (Exigence 2)	Nom : apdb . Tier : Flexible Server (Recommandé). Version : MySQL 8.0.	Solution PaaS recommandée pour réduire l'overhead d'administration (patching, maintenance).
<b>Sécurité DB</b>	Connexion sécurisée	Règle de Pare-feu autorisant l'accès depuis le VNet ( 10.10.1.0/24 ).	Restreindre l'accès à la base de données uniquement aux VMs de l'application.
<b>Azure Backup</b>	Sauvegarde des VMs (Résilience)	<b>Recovery Services Vault :</b> Création et configuration de la politique de sauvegarde.	Assure la continuité des opérations et la capacité de restauration complète des VMs IaaS.
<b>Azure App Service</b>	Bonus PaaS	Déploiement de l'application Node.js/React.	Démonstration d'une solution PaaS (sans gestion d'OS) pour l'application.

## V. Outils de Déploiement

Le déploiement sera effectué en utilisant une combinaison d'outils standards pour une approche professionnelle :

- 1. Azure CLI (Interface en ligne de commande) :** Utilisé via des **scripts PowerShell** pour l'automatisation du déploiement de l'infrastructure (VNet, NSG, VMs, LB, DB).
- 2. Scripts Bash :** Utilisés via des extensions de VM ou après SSH pour la configuration spécifique de la VM Ubuntu (installation de Docker).

3. **Commandes RDP/SSH :** Utilisées pour les configurations finales (IIS sur Windows, commande docker run sur Ubuntu).

## VI. Prochaines Étapes

---

1. **Finalisation des Scripts d'Infrastructure :** Intégration complète des commandes Azure CLI pour créer toutes les ressources listées.
2. **Script de Nettoyage :** Création d'un script ( cleanup.sh ou .ps1 ) pour supprimer le groupe de ressources afin d'éviter les frais.
3. **Tests de Validation :** Vérification de l'accès à la page HTML (Port 80) et à l'application Node.js (Ports 3000/5000) via l'IP Publique du Load Balancer.

## Configuration

---

## Création du groupe de ressources - Vnet - NSG - deux

# VM linux

---

```

# =====
# SCRIPT D'AUTOMATISATION DU DÉPLOIEMENT D'INFRASTRUCTURE AZURE (IaaS)
# Version : 2.0 (Double VM Linux - No Windows)
# =====

# Connexion (décommentez si nécessaire)
# Connect-AzAccount

# =====
# 1. DÉFINITION DES VARIABLES GLOBALES
# =====

# --- Configuration de Base ---
$RESOURCE_GROUP = "Project-Azure"
$LOCATION = "norwayeast"

# --- Configuration Réseau ---
$VNET_NAME = "VNET-Project-Azure"
$VNET_PREFIX = "10.10.0.0/16"
$SUBNET_NAME = "SUBNET-Project-Azure"
$SUBNET_PREFIX = "10.10.1.0/24"

# --- Configuration Sécurité (NSG) ---
$NSG_NAME = "NSG-Project-Azure"

# --- Configuration des Machines Virtuelles (VM Linux uniquement) ---
$VM_LINUX_01 = "VM-UBUNTU-01"
$VM_LINUX_02 = "VM-UBUNTU-02"
$VM_SIZE = "Standard_B2s"
$UBUNTU_IMAGE = "Ubuntu2204"

# --- Informations d'Administration ---
$ADMIN_USER = "dspl"
$ADMIN_PASSWORD = "Azure@2023Hello#"

# =====
# 2. CRÉATION DU GROUPE DE RESSOURCES ET DU RÉSEAU
# =====

Write-Host "➡ Démarrage du déploiement dans la région $LOCATION..."
az group create --name $RESOURCE_GROUP --location $LOCATION --output none

Write-Host "Création du VNet et Subnet..."
az network vnet create -g $RESOURCE_GROUP -n $VNET_NAME --address-prefix $VNET_
az network vnet subnet create -g $RESOURCE_GROUP --vnet-name $VNET_NAME --name

# =====
# 3. CRÉATION DU NSG ET RÈGLES (SSH, HTTP, HTTPS, NODE)
# =====

Write-Host "Création du NSG ($NSG_NAME) et des règles Linux..."

```

```

az network nsg create -g $RESOURCE_GROUP -n $NSG_NAME --location $LOCATION --ou

# SSH (22)
az network nsg rule create -g $RESOURCE_GROUP --nsg-name $NSG_NAME --name Allow

# Web (80, 443)
az network nsg rule create -g $RESOURCE_GROUP --nsg-name $NSG_NAME --name Allow
az network nsg rule create -g $RESOURCE_GROUP --nsg-name $NSG_NAME --name Allow

# Node.js App (3000, 5000)
az network nsg rule create -g $RESOURCE_GROUP --nsg-name $NSG_NAME --name Allow
az network nsg rule create -g $RESOURCE_GROUP --nsg-name $NSG_NAME --name Allow

# =====
# 4. CRÉATION DES MACHINES VIRTUELLES LINUX
# =====

$VMS = @($VM_LINUX_01, $VM_LINUX_02)

foreach ($VM_NAME in $VMS) {
    Write-Host "🚀 Déploiement de la machine : $VM_NAME..."
    az vm create -g $RESOURCE_GROUP -n $VM_NAME ` 
        --location $LOCATION ` 
        --image $UBUNTU_IMAGE ` 
        --size $VM_SIZE ` 
        --vnet-name $VNET_NAME --subnet $SUBNET_NAME ` 
        --nsg $NSG_NAME ` 
        --admin-username $ADMIN_USER ` 
        --admin-password $ADMIN_PASSWORD ` 
        --public-ip-sku Standard ` 
        --output none
}

# =====
# 5. RÉSUMÉ DES RESSOURCES
# =====

Write-Host "---"
Write-Host "✅ DÉPLOIEMENT TERMINÉ."
Write-Host "---"

foreach ($VM_NAME in $VMS) {
    $IP = az vm show -g $RESOURCE_GROUP -n $VM_NAME --query "publicIps" -o tsv
    Write-Host "💻 $VM_NAME :"
    Write-Host "    - IP Publique : $IP"
    Write-Host "    - Connexion : ssh $ADMIN_USER@$IP"
}
Write-Host "---"
az vm list -g $RESOURCE_GROUP -o table

```

**Microsoft Azure**

Accueil > Groupes de ressources >

### Project-Azure

Vue d'ensemble

Rechercher

Créer Gérer l'affichage Supprimer le groupe de ressources Actualiser Exporter au format CSV Ouvrez une requête ...

Type Vue JSON

Bases

Ressources Recommandations

Filtrer un champ... Type est égal à tout Emplacement est égal à tout Ajouter un filtre

Nom	Type	Emplacement
Adresse IP publique (2)		
VM-UBUNTU-01PublicIP	Adresse IP publique	Norway East
VM-UBUNTU-02PublicIP	Adresse IP publique	Norway East
Disque (2)		
VM-UBUNTU-01_OsDisk_1_1b67d678bbb148bf80ae33af95e4e528	Disque	Norway East
VM-UBUNTU-02_OsDisk_1_8812695cac4c4116b6a0bd792a54d1a2	Disque	Norway East

Vue d'ensemble

Rechercher

Créer Gérer l'affichage Supprimer le groupe de ressources Actualiser Exporter au format CSV Ouvrez une requête ...

Bases

Ressources Recommandations

Filtrer un champ... Type est égal à tout Emplacement est égal à tout Ajouter un filtre

Groupe de sécurité réseau (1)			
NSG-Project-Azure	Groupe de sécurité réseau	Norway East	
Interface réseau (2)			
VM-UBUNTU-01VMNic	Interface réseau	Norway East	
VM-UBUNTU-02VMNic	Interface réseau	Norway East	
Machine virtuelle (2)			
VM-UBUNTU-01	Machine virtuelle	Norway East	
VM-UBUNTU-02	Machine virtuelle	Norway East	
Réseau virtuel (1)			
VNET-Project-Azure	Réseau virtuel	Norway East	

Ajouter ou supprimer des favoris en appuyant sur Ctrl+Shift+F Envoyer

# Déploiement Azure Database pour MySQL

**Microsoft Azure** | Rechercher dans les ressources, services et documents (G+/) | Copilot | 4CLAZUR.pape.lo@estia... ESTIAM (ESTIAM.COM)

Accueil > MySqlFlexibleServer\_e99d60da788e41ec967bc48c70a5a5ff | Vue d'ensemble

MySQLFlexibleServer\_e99d60da788e41ec967bc48c70a5a5ff | Vue d'ensemble

Rechercher | Supprimer | Annuler | Redéployer | Télécharger | Actualiser

Le déploiement est en cours

Nom du déploiement : MySqlFlexibleServer\_e99d60da788e41ec967bc48c70a5a5ff | Heure de début : 18/12/2025 15:55:58  
Abonnement : Azure for Students | ID de corrélation : 9351d3a5-8be1-45b5-b28a-5bf51c2ba69  
Groupe de ressources : Project-Azure

Détails du déploiement

Ressource	Type	Statut	Détails de l'opération
databaseprojectazure	Serveur flexible Azure Database	Accepted	Détails de l'opération

Étapes suivantes

Aidez-nous à améliorer cette page en appuyant sur <https://portal.azure.com/#home>

**Microsoft Azure** | Rechercher dans les ressources, services et documents (G+/) | Copilot | 4CLAZUR.pape.lo@estia... ESTIAM (ESTIAM.COM)

Accueil > MySqlFlexibleServer\_e99d60da788e41ec967bc48c70a5a5ff | Vue d'ensemble

databaseprojectazure | Serveur flexible Azure Database pour MySQL

Rechercher | Se connecter | Afficher la liste des processus | Supprimer | Réinitialiser le mot de passe | Restaurer | Redémarrer | Mettre à niveau | Arrêter | Actualiser

Vue d'ensemble

Azure Database pour MySQL – Série de webinaires en direct : Découvrez les dernières mises à jour (avec démos) et interagissez directement avec le groupe de produits le 2ème mercredi de chaque mois ! [Abonnez-vous à notre chaîne YouTube](#) aujourd'hui !

Bases

Point de terminaison	Nom de connexion de l'a...	Configuration	Version de MySQL	Zone de disponibilité	Créé le
databaseprojectazure.mysql.database.azure.com	dspl	: Expansible_B1ms_1 vCores_2 Gio de RAM_20 de stockage_360 IQ...	8.0 Mettre à niveau	1	2025-12-18 14:58:28.8894082 UTC

Étiquettes (modifier) : Ajouter des étiquettes

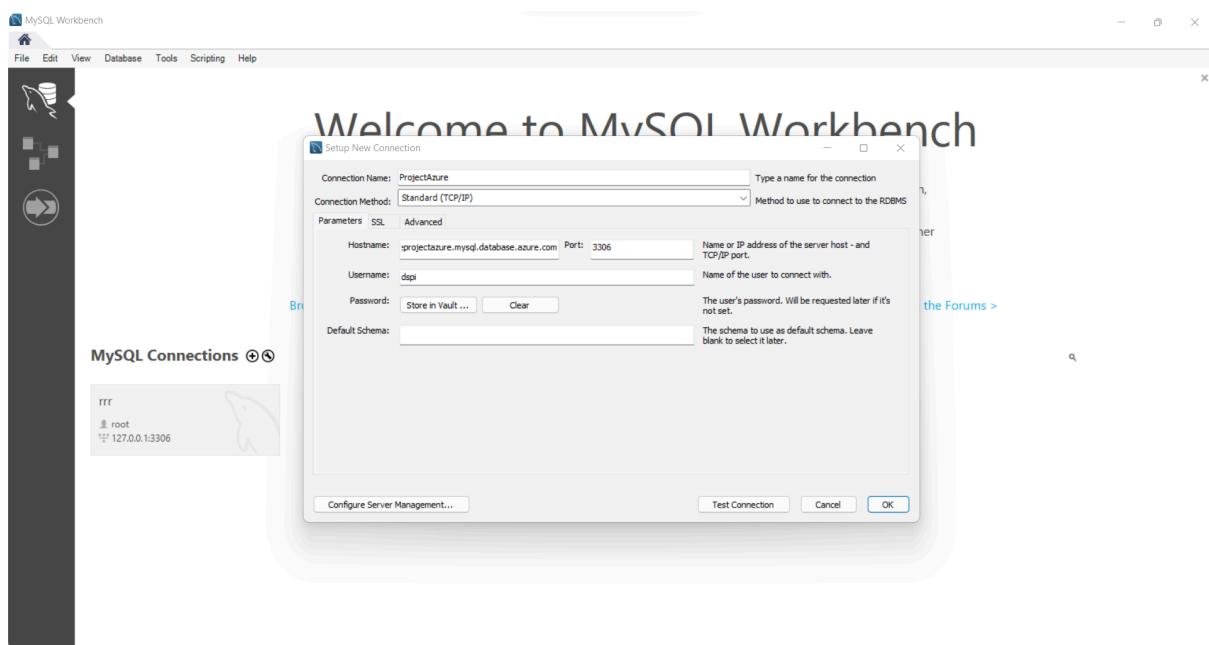
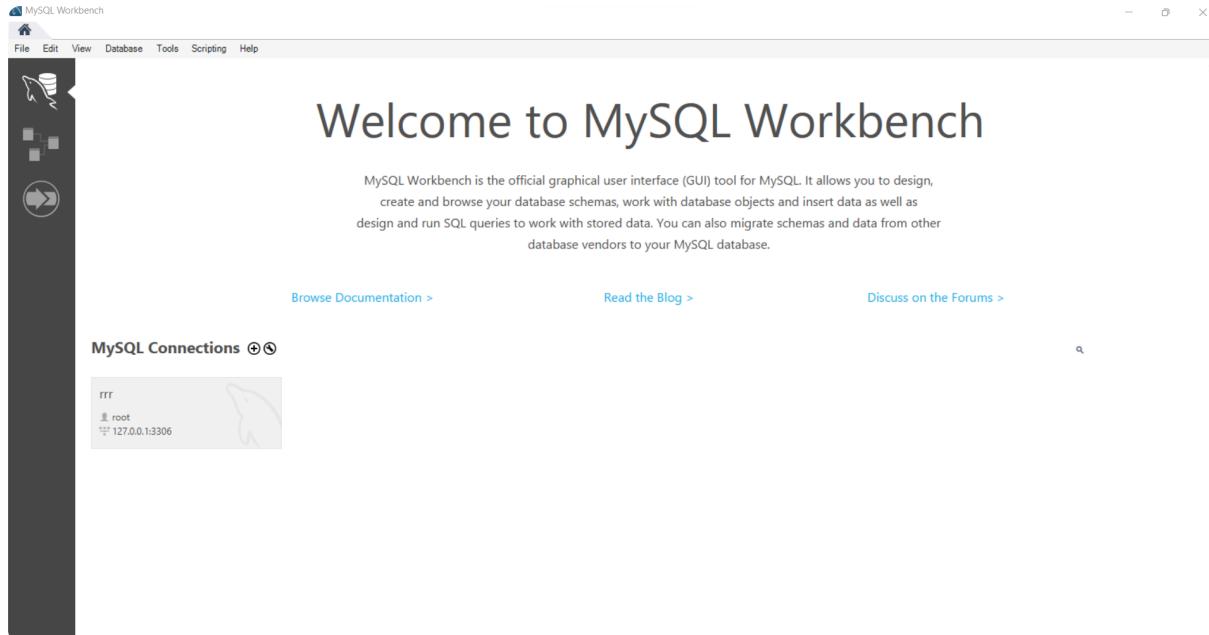
Commencer Propriétés Recommandations Supervision Tutoriels

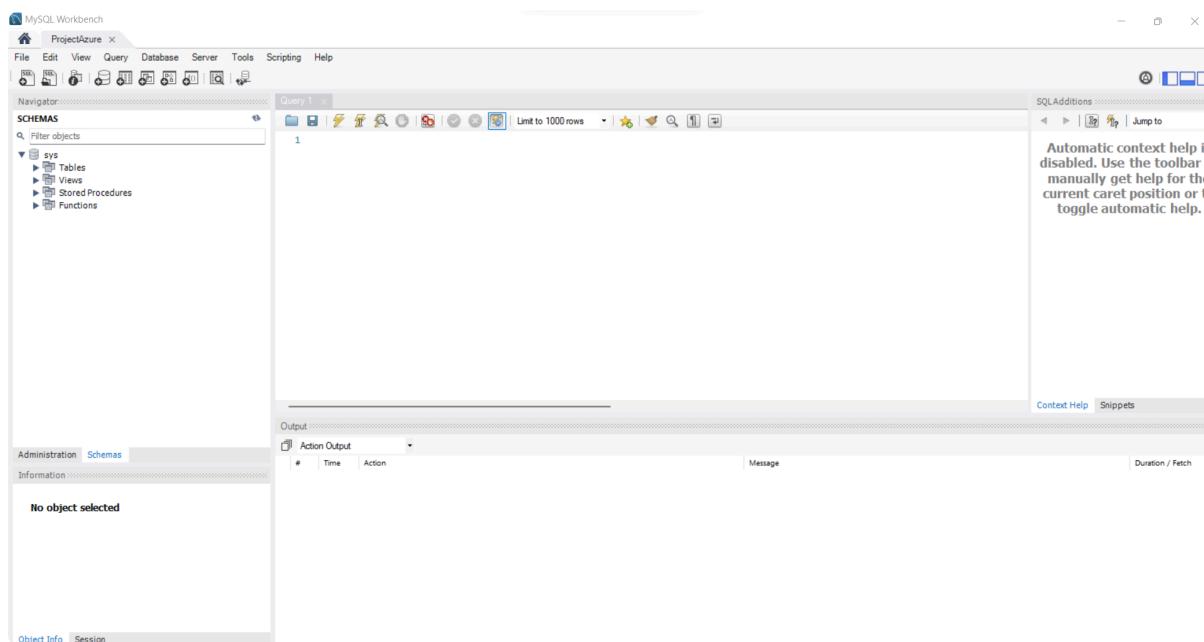
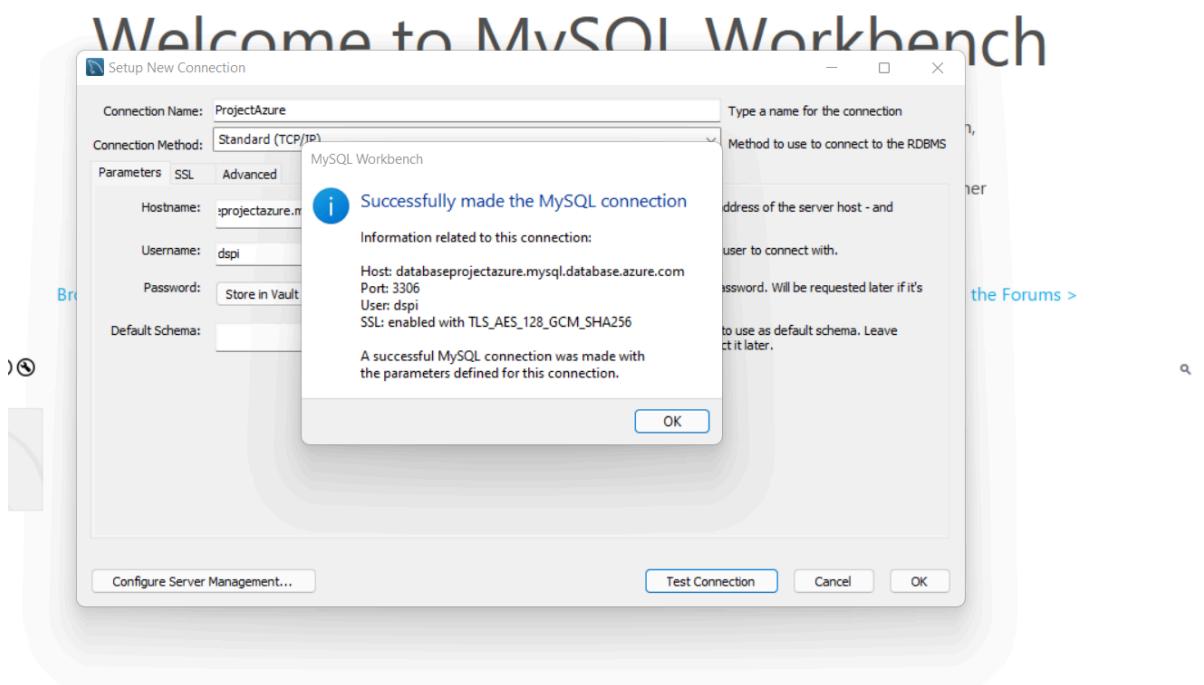
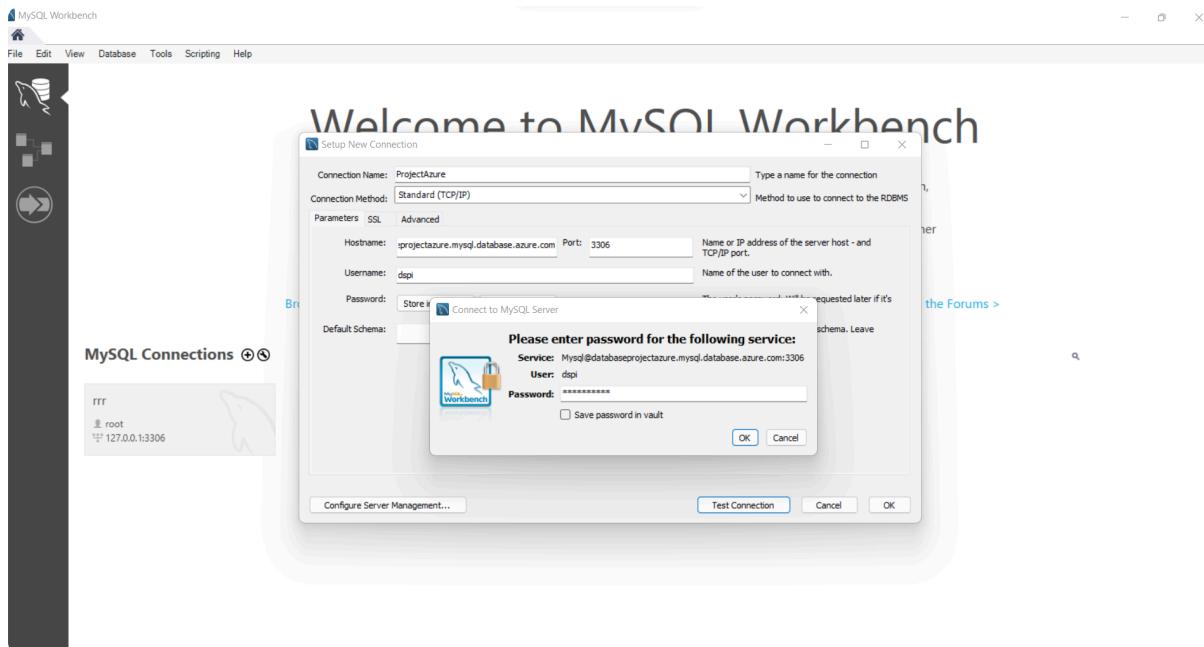
Nous avons préparé une liste de vérification pour vous aider à démarrer

Aidez-nous à améliorer cette page en appuyant sur <https://portal.azure.com/#home>

# Connexion au serveur pour la création de la base de données

- MySQL Workbench





- Création de la base de données et les table

-- 1. Création de la base de données

```
CREATE DATABASE IF NOT EXISTS appdb
CHARACTER SET utf8mb4
COLLATE utf8mb4_unicode_ci;
```

-- 2. Utiliser la base

```
USE appdb;
```

-- 3. Création de la table employees

```
CREATE TABLE IF NOT EXISTS employees (
    id VARCHAR(50) NOT NULL,
    firstName VARCHAR(100) NOT NULL,
    lastName VARCHAR(100) NOT NULL,
    email VARCHAR(255) NOT NULL,
    phone VARCHAR(50) NULL,
    department VARCHAR(100) NOT NULL,
    position VARCHAR(100) NOT NULL,
    status ENUM('active','inactive','remote') NOT NULL DEFAULT 'active',
    hireDate DATE NOT NULL,
    salary DECIMAL(10,2) NOT NULL,
    avatar VARCHAR(255) NULL,
    PRIMARY KEY (id),
    UNIQUE KEY uniq_email (email)
);
```

-- 4. Création de la table contact

```
CREATE TABLE IF NOT EXISTS contact (
    id INT AUTO_INCREMENT PRIMARY KEY,
    name VARCHAR(255) NOT NULL,
    email VARCHAR(255) NOT NULL,
    subject VARCHAR(255) NOT NULL,
    message TEXT NOT NULL,
    created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    INDEX idx_email (email),
    INDEX idx_created_at (created_at)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_unicode_ci;
```

The screenshot shows the MySQL Workbench interface. In the Navigator pane, under SCHEMAS, the 'appdb' schema is selected. In the Query Editor (Query 1), the following SQL script is run:

```

2 CREATE DATABASE IF NOT EXISTS appdb
3   CHARACTER SET utf8mb4
4   COLLATE utf8mb4_unicode_ci;
5
6 -- 2. Utiliser la base
7 USE appdb;
8
9 -- 3. Création de la table employees
10 CREATE TABLE IF NOT EXISTS employees (
11   id VARCHAR(50) NOT NULL,
12   firstName VARCHAR(100) NOT NULL,
13   lastName VARCHAR(100) NOT NULL,
14   email VARCHAR(255) NOT NULL,
15   phone VARCHAR(50) NULL,
16   department VARCHAR(100) NOT NULL,
17   position VARCHAR(100) NOT NULL,
18   status ENUM('active','inactive','remote') NOT NULL DEFAULT 'active',
19   hireDate DATE NOT NULL,
20   salary DECIMAL(10,2) NOT NULL,
21   avatar VARCHAR(255) NULL,
22   PRIMARY KEY (id),
23   UNIQUE KEY uniq_email (email)
24 );

```

The Output pane shows the results of the queries:

#	Time	Action	Message	Duration / Fetch
1	16:13:39	CREATE DATABASE IF NOT EXISTS appdb CHARACTER SET utf8mb4 COLLATE utf8mb4_unicode_ci;	1 row(s) affected	0.063 sec
2	16:13:39	USE appdb;	0 row(s) affected	0.031 sec
3	16:13:39	CREATE TABLE IF NOT EXISTS employees ( id VARCHAR(50) NOT NULL, firstName VARCHAR(100) NOT NULL, lastName VARCHAR(100) NOT NULL, email VARCHAR(255) NOT NULL, phone VARCHAR(50) NULL, department VARCHAR(100) NOT NULL, position VARCHAR(100) NOT NULL, status ENUM('active','inactive','remote') NOT NULL DEFAULT 'active', hireDate DATE NOT NULL, salary DECIMAL(10,2) NOT NULL, avatar VARCHAR(255) NULL, PRIMARY KEY (id), UNIQUE KEY uniq_email (email) );	0 row(s) affected	0.110 sec

The screenshot shows the MySQL Workbench interface. In the Navigator pane, under SCHEMAS, the 'appdb' schema is selected. In the Query Editor (Query 1), the following SQL script is run:

```

1 USE appdb;
2 SELECT * FROM employees;
3
4

```

The Result Grid shows the structure of the 'employees' table:

ID	firstName	lastName	email	phone	department	position	status	hireDate	salary	avatar
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

The Output pane shows the results of the queries:

#	Time	Action	Message	Duration / Fetch
3	16:13:39	CREATE TABLE IF NOT EXISTS employees ( id VARCHAR(50) NOT NULL, firstName VARCHAR(100) NOT NULL, lastName VARCHAR(100) NOT NULL, email VARCHAR(255) NOT NULL, phone VARCHAR(50) NULL, department VARCHAR(100) NOT NULL, position VARCHAR(100) NOT NULL, status ENUM('active','inactive','remote') NOT NULL DEFAULT 'active', hireDate DATE NOT NULL, salary DECIMAL(10,2) NOT NULL, avatar VARCHAR(255) NULL, PRIMARY KEY (id), UNIQUE KEY uniq_email (email) );	0 row(s) affected	0.110 sec
4	16:14:54	USE appdb;	0 row(s) affected	0.031 sec
5	16:14:55	SELECT * FROM employees LIMIT 0, 1000;	0 row(s) returned	0.047 sec / 0.000 sec

- Pare-feu de la base de données

Azure portal screenshot showing network settings for a MySQL server. The 'Mise en réseau' (Network) tab is selected. Under 'Accès public' (Public access), there are two checkboxes: one for allowing public access via Internet using a public IP address, and another for allowing public access via a service in Azure. Below these are two rows of network rules:

Nom de la règle de pare-feu	Adresse IP de début	Adresse IP de fin
ClientIPAddress_2025-12-18_15-56-58	88.138.237.162	88.138.237.162
VM_Ubuntu01	20.251.223.213	20.251.223.213
VM_Ubuntu02	4.235.106.204	4.235.106.204

## Installation Docker

- Créer un fichier bash et lui donner les autorisations



## Dépendances applicatives dans chaque VM

```
sudo apt update && sudo apt upgrade -y
# Installation de Node.js (via NodeSource pour avoir une version récente)
curl -fsSL https://deb.nodesource.com/setup_20.x | sudo -E bash -
sudo apt install -y nodejs nginx git
# Installation globale de PM2
sudo npm install -g pm2
```

## Cloner L'application

### Accès GitHub via SSH

Au niveau de chaque VM :

- 1 Générer une clé SSH

```
ssh-keygen -t ed25519 -C "pape-dev"
```

- 3 Copier la clé publique

```
cat ~/.ssh/id_ed25519.pub
```

- **4** Ajouter la clé sur GitHub : GitHub → Settings → SSH and GPG keys → New SSH key - Colle la clé → Save
- **5** Cloner le repo en SSH

```
git clone git@github.com:pape-dev/dspi-tech-employee-hub.git
```

## Lancement avec PM2

---

```
cd ~/dspl-tech-employee-hub  
pm2 start server/index.js --name "api-backend"  
pm2 save
```

## Installer les dépendances

---

```
npm install
```

## Build du projet

---

```
npm run build
```

## Configuration de Nginx (Reverse Proxy)

---

```
nano /etc/nginx/sites-available/mon_app
```

code à mettre :

```

server {
    listen 80;
    server_name 4.235.106.204; # Votre IP Azure

    # Serveur de fichiers statiques (Frontend)
    location / {
        root /home/dspi/dspi-tech-employee-hub/dist;
        index index.html;
        try_files $uri $uri/ /index.html;
    }

    # Proxy vers le Backend (Express)
    location /api {
        proxy_pass http://127.0.0.1:3000;
        proxy_http_version 1.1;
        proxy_set_header Upgrade $http_upgrade;
        proxy_set_header Connection 'upgrade';
        proxy_set_header Host $host;
        proxy_cache_bypass $http_upgrade;
    }
}

```

## Gestion des permissions

---

```

# Autoriser Nginx à accéder à votre dossier utilisateur
sudo chmod +x /home/dspi

# Donner les droits de lecture sur le projet
sudo chmod -R 755 /home/dspi/dspi-tech-employee-hub

```

## Activation

---

```

sudo ln -s /etc/nginx/sites-available/mon_app /etc/nginx/sites-enabled/
sudo nginx -t
sudo systemctl restart nginx

```

# Quelques commandes utiles

---

```
# Voir si le backend est "online"
pm2 status
```

```
# Redémarrer le backend
pm2 restart api-backend
```

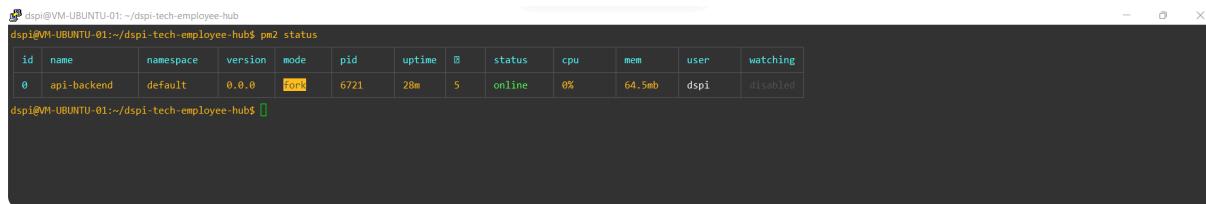
```
# Voir les logs du backend (erreurs de code ou base de données)
pm2 logs api-backend
```

```
# Sauvegarder pour le redémarrage automatique de la VM
pm2 save
```

## Statut du pm2

---

- VM 1



```
dspi@VM-UBUNTU-01:~/dspl-tech-employee-hub$ pm2 status
          id  name  namespace  version  mode  pid  uptime  status  cpu  mem  user  watching
          0  api-backend  default  0.0.0  fork  6721  28m  online  0%  64.5mb  dspl  disabled
dspi@VM-UBUNTU-01:~/dspl-tech-employee-hub$
```

- VM 2



```
dspi@VM-UBUNTU-02:~/dspl-tech-employee-hub$ pm2 status
          id  name  namespace  version  mode  pid  uptime  status  cpu  mem  user  watching
          0  api-backend  default  0.0.0  fork  5855  9m  online  0%  61.8mb  dspl  disabled
dspi@VM-UBUNTU-02:~/dspl-tech-employee-hub$
```

# Se connecter à l'application

- VM 1 : 20.251.223.213



ID	Employé	Département	Poste	Statut	Date d'embauche	Salaire	Contact
EMP237828	Keelie Gray	Marketing	UX Designer Senior	Actif	17 oct. 1992	34 €	✉️ ☎️
EMP210845	Nasim Hoover	Commercial	Marketing Manager	Actif	30 janv. 2009	54 €	✉️ ☎️
EMP787915	Magee McKay	Infrastructure	Backend Developer	Actif	14 fevr. 2018	60 €	✉️ ☎️
EMP650725	Ira Munoz	Design	Product Designer	Actif	9 fevr. 2014	10 €	✉️ ☎️

- VM 2 : 4.235.106.204

The screenshot shows the homepage of the DSPI-TECH application. At the top, there's a header bar with the DSPI-TECH logo and navigation links like Accueil, Salariés, Nouveau, and Contact. Below the header, a banner reads "Plateforme RH Nouvelle Génération" and features the title "Bienvenue chez DSPI-TECH". A subtitle below it says, "Gérez efficacement vos ressources humaines avec notre plateforme intuitive et performante. L'excellence technologique au service de vos équipes." At the bottom of the banner are two buttons: "Voir les salariés →" and "Ajouter un employé".

The screenshot shows the "Gestion des Salariés" (Employee Management) page. The header is identical to the homepage. The main area displays a summary of employee status: 5 Total, 5 Actifs, 0 Remote, and 0 Inactifs. Below this is a search bar with placeholder "Rechercher un employé..." and dropdown filters for "Tous les..." and "Tous les statuts". A large table lists three employees: Keelie Gray (UX Designer Senior), Nasim Hoover (Marketing Manager), and Magee McKay (Backend Developer). Each row includes a small profile picture, the employee's name, department, position, status (green badge indicating active), hire date, salary (34€, 54€, or 60€), and contact icons.

ID	Employé	Département	Poste	Statut	Date d'embauche	Salaire	Contact
EMP237828	KG Keelie Gray	Marketing	UX Designer Senior	<span>Actif</span>	17 oct. 1992	34 €	<a href="#">✉</a> <a href="#">📞</a>
EMP210845	NH Nasim Hoover	Commercial	Marketing Manager	<span>Actif</span>	30 janv. 2009	54 €	<a href="#">✉</a> <a href="#">📞</a>
EMP787915	MM Magee McKay	Infrastructure	Backend Developer	<span>Actif</span>	14 févr. 2018	60 €	<a href="#">✉</a> <a href="#">📞</a>

# Vérifier les insertions de la base de données

The screenshot shows the MySQL Workbench interface with the following details:

- Navigator:** Shows the schema `appdb` containing the `employees` table.
- Query Editor:** Displays the SQL query:
 

```
1 • USE appdb;
2 • SELECT * FROM employees;
```
- Result Grid:** Shows the data from the `employees` table, including columns: id, firstName, lastName, email, phone, department, position, status, hireDate, salary, and avatar.
- Action Output:** Shows the history of actions taken during the session:
 

#	Time	Action	Message	Duration / Fetch
9	18:11:26	SELECT * FROM employees LIMIT 0, 1000	5 row(s) returned	0.031 sec / 0.000 sec
10	18:11:33	USE appdb	0 row(s) affected	0.031 sec
11	18:11:33	SELECT * FROM employees LIMIT 0, 1000	5 row(s) returned	0.031 sec / 0.000 sec

# Configuration du load balancer

[Accueil > Équilibrage de charge et distribution de contenu | Vue d'ensemble >](#)

### Créer un équilibrEUR de charge

Validation réussie

De base Configuration d'adresse IP frontale Pools principaux Règles de trafic entrant Règles de trafic sortant Étiquettes Vérifier + créer

**De base**

Abonnement	Azure for Students
Groupe de ressources	Project-Azure
Nom	LoadBlancerAzure
Région	Norway East
RÉFÉRENCE (SKU)	Standard
Niveau	Régional
Type	Public

**Configuration d'adresse IP frontale**

Nom de la configuration IP frontale	Load1
Adresse IP de configuration IP frontale	À créer

**Pools principaux**

[Créer](#) < Précédent Suivant > Télécharger un modèle pour automation Fournir des commentaires

---

[Accueil > CreateLoadBalancerBladeV2-20251218182211 | Vue d'ensemble >](#)

### LoadBlancerAzure

ÉquilibrEUR de charge

Vue d'ensemble Journal d'activité Contrôle d'accès (IAM) Étiquettes Diagnostiquer et résoudre les problèmes Visualiseur de ressources Paramètres Supervision Automatisation Aide

Vérifier les performances de mon Load Balancer Nombre d'alertes critiques pour cet équilibrEUR de charge Analyser les alertes pour cet équilibrEUR de charge

[Rechercher](#) Déplacez Supprimer Actualiser Envoyer des commentaires

[Bases](#)

Groupe de ressources (...) :	Project-Azure	Pool de back-ends :	Pool1 (2 machines virtuelles)
Emplacement (déplacer) :	Norway East	Règle d'équilibrage de c... :	Rule1 (Tcp/80)
Abonnement (déplacer) :	Azure for Students	Sonde d'intégrité :	SOND1 (tcp :80)
ID d'abonnement :	0dc0a6ca-643a-469f-bb07-a45737b65645	Règles NAT de trafic entr... :	-
SKU :	Standard	Règles de trafic sortant :	-
Niveau :	Regional	Adresse IP front-end :	4.220.27.171 (IP_Load)
Étiquettes (modifier) :	Ajouter des étiquettes		

[Vue JSON](#)

**Configurer une haute disponibilité et une scalabilité pour vos applications**

Créez des applications scalables et à haut niveau de disponibilité en quelques minutes en utilisant l'équilibrage de charge intégré pour les services cloud et les machines virtuelles. Azure Load Balancer prend en charge les protocoles TCP/UDP et ceux utilisés pour les applications de messagerie vocale et vidéo en temps réel. [Découvrir plus d'informations](#)

 **Équilibrer les adresses IPv4 et IPv6**  
Les points de terminaison natifs bi-piles permettent de répondre aux exigences réglementaires et de gérer rapidement le nombre d'appareils en mode mobile et IoT. [Découvrir plus d'informations](#)

 **Créer des applications à haut niveau de fiabilité**  
Load Balancer améliore la disponibilité des applications en routant le trafic vers les nœuds sains. [Découvrir plus d'informations](#)

 **Sécurisez vos réseaux**  
Contrôlez le trafic réseau et protégez les réseaux privés à l'aide d'une traduction d'adresses réseau (NAT) intégrée. [Découvrir plus d'informations](#)

Ajouter ou supprimer des favoris en appuyant sur <https://portal.azure.com/?Cert+Shift+F>

## Au niveau du code > VM 1 & VM 2 Mettre à jour la configuration du nginx

```
sudo nano /etc/nginx/sites-available/mon_app
```

Les IP "Server\_Name" ont été remplacés par "\_" :

```

dspi@VM-UBUNTU-02:~/dspi-tech-employee-hub$ cat /etc/nginx/sites-available/mon_app
server {
    listen 80;
    server_name _; # Votre IP Azure
    # Serveur de fichiers statiques (Frontend)
    location / {
        root /home/dspi/dspi-tech-employee-hub/dist;
        index index.html;
        try_files $uri $uri/ /index.html;
    }
    # Proxy vers le Backend (Express)
    location /api {
        proxy_pass http://127.0.0.1:3000;
        proxy_http_version 1.1;
        proxy_set_header Upgrade $http_upgrade;
        proxy_set_header Connection 'upgrade';
        proxy_set_header Host $host;
        proxy_cache_bypass $http_upgrade;
    }
}
dspi@VM-UBUNTU-02:~/dspi-tech-employee-hub$ 

dspi@VM-UBUNTU-01:~/dspi-tech-employee-hub/src/pages$ cat /etc/nginx/sites-available/mon_app
server {
    listen 80;
    server_name _; # Votre IP Azure
    # 1. Le Frontend React (Fichiers statiques)
    location / {
        # Vérifiez bien si votre dossier est 'dist' ou 'build'
        root /home/dspi/dspi-tech-employee-hub/dist;
        index index.html;
        try_files $uri $uri/ /index.html;
    }
    # 2. Le Backend Express (Proxy vers le port 3000)
    location /api {
        proxy_pass http://localhost:3000; # Port de votre backend Express
        proxy_http_version 1.1;
        proxy_set_header Upgrade $http_upgrade;
        proxy_set_header Connection 'upgrade';
        proxy_set_header Host $host;
        proxy_cache_bypass $http_upgrade;
    }
}
dspi@VM-UBUNTU-01:~/dspi-tech-employee-hub/src/pages$ 

```

`npm run build`

`sudo systemctl restart nginx`

`pm2 restart api-backend`

# Se connecter avec l'IP du Load Balancer

Microsoft Azure

Accueil > LoadBlancerAzure

LoadBlancerAzure | Configuration d'adresse IP frontale

Équilibrage de charge

Rechercher

Ajouter Actualiser

Tapez pour commencer à filtrer...

Affichage de tous les éléments

Nom	Adresse IP	Nombre de règles
Load1	4.220.27.171 (IP_Load)	1

Configuration d'adresse IP frontale

- Pools principaux
- Sondes d'intégrité
- Règles d'équilibrage de charge
- Règles NAT de trafic entrant
- Règles de trafic sortant

Ajouter ou supprimer des favoris en appuyant sur Ctrl+Shift+F

Envoyer des commentaires

Le load Balancer affiche L'App de la VM 1 :

Non sécurisé 4.220.27.171

Important Applications Réseaux-sociaux Cours Stage Azure Importés depuis Fir... Cisco IOU License G... LabHub GitHub - ishare2-or... Cours et TP Réseaux

DSPI-TECH

VM-UBUNTU-01 Azure

Bienvenue chez DSPI-TECH

Gérez efficacement vos ressources humaines avec notre plateforme intuitive et performante. L'excellence technologique au service de vos équipes.

Voir les salariés → Ajouter un employé

On actualise la page il affiche l'App de la VM 2 :

ID	Employé	Département	Poste	Statut	Date d'embauche	Salaire	Contact
EMP237828	Keelle Gray	Marketing	UX Designer Senior	Actif	17 oct. 1992	34 €	
EMP210845	Nasim Hoover	Commercial	Marketing Manager	Actif	30 janv. 2009	54 €	

## ✓ Conclusion

Ce projet valide les compétences suivantes :

- Provisionnement IaaS via scripts automatisés.
- Gestion de services PaaS (MySQL Managé).
- Conteneurisation et Reverse Proxy (Docker / Nginx).
- Haute Disponibilité (Standard Load Balancer).
- Déployer & héberger des applications
- Sécurisation (Groupes de sécurité et SSH).

# Application Web : DSPI-TECH Employee Hub

---

Application web complète de gestion des employés et des contacts pour DSPI-TECH. Cette application permet de gérer les informations des collaborateurs, d'ajouter de nouveaux employés, de consulter les données et d'exporter les informations au format CSV.

## Table des matières

---

- [Description](#)
- [Fonctionnalités](#)
- [Technologies utilisées](#)
- [Prérequis](#)
- [Installation](#)
- [Configuration](#)
- [Structure du projet](#)
- [API Endpoints](#)
- [Base de données](#)
- [Utilisation](#)
- [Scripts disponibles](#)
- [Déploiement](#)
- [Contribution](#)

## Description

---

DSPI-TECH Employee Hub est une application full-stack moderne permettant de :

- Visualiser et gérer les informations des employés
- Ajouter de nouveaux collaborateurs
- Gérer les contacts via un formulaire
- Exporter les données en CSV
- Filtrer et trier les données des employés

## Fonctionnalités

---

### Page d'accueil (Index)

- Vue d'ensemble avec statistiques des employés
- Présentation des fonctionnalités principales
- Navigation vers les différentes sections

### Gestion des salariés (Salaries)

- Affichage de tous les employés dans un tableau interactif
- Recherche par nom, email ou poste
- Filtrage par département et statut
- Tri par nom, département, poste, date d'embauche ou salaire
- Statistiques en temps réel (Total, Actifs, Remote, Inactifs)
- Export CSV des employés filtrés

### Ajout d'employé (Nouveau)

- Formulaire complet pour ajouter un nouvel employé
- Validation des champs obligatoires
- Génération automatique d'ID unique
- Gestion des départements et postes prédéfinis

### Contact

- Formulaire de contact pour les visiteurs
- Enregistrement des messages en base de données
- Export CSV des contacts
- Informations de contact de l'entreprise

## Technologies utilisées

---

### Frontend

- **React 18.3.1** - Bibliothèque UI
- **TypeScript 5.8.3** - Typage statique
- **Vite 7.3.0** - Build tool et dev server

- **React Router DOM 6.30.1** - Routage
- **Tailwind CSS 3.4.17** - Framework CSS
- **shadcn/ui** - Composants UI basés sur Radix UI
- **Lucide React** - Icônes
- **TanStack Query** - Gestion des données serveur
- **React Hook Form** - Gestion des formulaires
- **Zod** - Validation de schémas

## Backend

- **Node.js** - Runtime JavaScript
- **Express 4.22.1** - Framework web
- **MySQL2 3.16.0** - Driver MySQL
- **CORS 2.8.5** - Gestion CORS
- **dotenv 16.6.1** - Variables d'environnement

## Outils de développement

- **ESLint** - Linter
- **TypeScript ESLint** - Linter TypeScript
- **Concurrently** - Exécution parallèle de scripts

## Prérequis

---

Avant de commencer, assurez-vous d'avoir installé :

- **Node.js** (version 18 ou supérieure) - [Télécharger Node.js](https://nodejs.org/) (<https://nodejs.org/>).
- **npm** (inclus avec Node.js) ou **yarn**
- **MySQL** (version 8.0 ou supérieure) - [Télécharger MySQL](https://dev.mysql.com/downloads/mysql/) (<https://dev.mysql.com/downloads/mysql/>).
- **Git** - [Télécharger Git](https://git-scm.com/) (<https://git-scm.com/>).

# Installation

## 1. Cloner le repository

```
git clone <URL_DU_REPOSITORY>
cd dspi-tech-employee-hub
```

## 2. Installer les dépendances

```
npm install
```

## 3. Configurer la base de données

Créez une base de données MySQL et exécutez le script SQL :

```
mysql -u root -p < Docs_Config/bd.sql
```

Ou connectez-vous à MySQL et exécutez le contenu du fichier `Docs_Config/bd.sql`.

## 4. Configurer les variables d'environnement

Créez un fichier `.env` à la racine du projet :

```
# Configuration de la base de données
DB_HOST="Server database for MySQL"
DB_PORT=3306
DB_NAME=appdb
DB_USER=votre_utilisateur
DB_PASSWORD=votre_mot_de_passe

# Configuration du serveur API
PORT=3000

# Configuration du frontend (optionnel)
VITE_API_URL=http://localhost:3000
```

## 5. Démarrer l'application

### Option 1 : Démarrer frontend et backend ensemble

```
npm run dev:full
```

## Option 2 : Démarrer séparément

Terminal 1 - Frontend :

```
npm run dev
```

Terminal 2 - Backend :

```
npm run server
```

L'application sera accessible sur :

- **Frontend** : <http://localhost:8080> (<http://localhost:8080>)
- **Backend API** : <http://localhost:3000> (<http://localhost:3000>)

## Configuration

---

### Variables d'environnement

#### Backend ( .env )

Variable	Description	Exemple
DB_HOST	Adresse du serveur MySQL	localhost
DB_PORT	Port MySQL	3306
DB_NAME	Nom de la base de données	appdb
DB_USER	Utilisateur MySQL	root
DB_PASSWORD	Mot de passe MySQL	password
PORT	Port du serveur API	3000

#### Frontend ( .env ou .env.local )

Variable	Description	Exemple
VITE_API_URL	URL de l'API backend	<a href="http://localhost:3000">http://localhost:3000</a>

## Configuration Vite

Le fichier `vite.config.ts` configure :

- Port du serveur de développement : **8080**
- Alias `@` pour le dossier `src`
- Plugin React avec SWC pour une compilation rapide

# Structure du projet

---

```
dspi-tech-employee-hub/
├── Docs_Config/           # Documentation et scripts
│   ├── bd.sql              # Script de création de la base de données
│   ├── Architecture.png    # Diagramme d'architecture
│   ├── Deploy_VM.ps1        # Script de déploiement PowerShell
│   └── docker.sh            # Script Docker
├── public/                 # Fichiers statiques
│   ├── favicon.ico
│   └── robots.txt
├── server/                 # Backend Express
│   ├── db.js                # Configuration de la connexion MySQL
│   └── index.js              # Serveur API Express
└── src/                    # Code source frontend
    ├── components/          # Composants React
    │   ├── Layout.tsx         # Layout principal
    │   ├── NavLink.tsx        # Composant de navigation
    │   └── ui/                 # Composants shadcn/ui
    ├── data/                  # Données statiques
    │   └── employees.ts       # Types et données d'exemple
    ├── hooks/                 # Hooks React personnalisés
    │   ├── use-mobile.tsx
    │   └── use-toast.ts
    ├── lib/                   # Utilitaires
    │   └── utils.ts            # Fonctions utilitaires
    ├── pages/                 # Pages de l'application
    │   ├── Index.tsx           # Page d'accueil
    │   ├── Salaries.tsx         # Gestion des salariés
    │   ├── Nouveau.tsx          # Ajout d'employé
    │   ├── Contact.tsx          # Formulaire de contact
    │   └── NotFound.tsx         # Page 404
    ├── App.tsx                # Composant racine
    ├── App.css                # Styles globaux
    ├── index.css              # Styles Tailwind
    ├── main.tsx                # Point d'entrée
    └── vite-env.d.ts           # Types Vite
    .env                         # Variables d'environnement (à créer)
    package.json                # Dépendances et scripts
    tsconfig.json               # Configuration TypeScript
    vite.config.ts              # Configuration Vite
    tailwind.config.ts          # Configuration Tailwind
    README.md                   # Ce fichier
```

# 🔌 API Endpoints

## Health Check

```
GET /api/health
```

Vérifie l'état du serveur.

**Réponse :**

```
{  
    "status": "ok"  
}
```

## Employés

### Récupérer tous les employés

```
GET /api/employees
```

**Réponse :**

```
[  
    {  
        "id": "EMP123456",  
        "firstName": "Jean",  
        "lastName": "Dupont",  
        "email": "jean.dupont@dspi-tech.com",  
        "phone": "+33 6 12 34 56 78",  
        "department": "IT",  
        "position": "Développeur",  
        "status": "active",  
        "hireDate": "2024-01-15",  
        "salary": 50000,  
        "avatar": null  
    }  
]
```

### Créer un nouvel employé

```
POST /api/employees
```

**Body :**

```
{
    "id": "EMP123456",
    "firstName": "Jean",
    "lastName": "Dupont",
    "email": "jean.dupont@dspi-tech.com",
    "phone": "+33 6 12 34 56 78",
    "department": "IT",
    "position": "Développeur",
    "status": "active",
    "hireDate": "2024-01-15",
    "salary": 50000,
    "avatar": null
}
```

**Réponse :**

```
{
    "message": "Employé créé",
    "id": 1
}
```

**Contacts****Récupérer tous les contacts**

GET /api/contact

**Réponse :**

```
[
    {
        "id": 1,
        "name": "John Doe",
        "email": "john@example.com",
        "subject": "Question",
        "message": "Bonjour...",
        "created_at": "2024-01-15T10:30:00.000Z"
    }
]
```

**Créer un nouveau contact**

POST /api/contact

**Body :**

```
{
  "name": "John Doe",
  "email": "john@example.com",
  "subject": "Question",
  "message": "Bonjour, j'aimerais..."
}
```

**Réponse :**

```
{
  "message": "Contact créé",
  "id": 1
}
```

 **Base de données****Structure****Table employees**

Colonne	Type	Description
id	VARCHAR(50)	Identifiant unique (PK)
firstName	VARCHAR(100)	Prénom
lastName	VARCHAR(100)	Nom
email	VARCHAR(255)	Email (UNIQUE)
phone	VARCHAR(50)	Téléphone (nullable)
department	VARCHAR(100)	Département
position	VARCHAR(100)	Poste
status	ENUM	Statut : 'active', 'inactive', 'remote'
hireDate	DATE	Date d'embauche
salary	DECIMAL(10,2)	Salaire annuel
avatar	VARCHAR(255)	URL avatar (nullable)

## Table contact

Colonne	Type	Description
id	INT	Identifiant auto-incrémenté (PK)
name	VARCHAR(255)	Nom complet
email	VARCHAR(255)	Email
subject	VARCHAR(255)	Sujet
message	TEXT	Message
created_at	TIMESTAMP	Date de création (auto)

## Script SQL

Le script de création de la base de données se trouve dans `Docs_Config/bd.sql`.

Pour créer la base de données :

```
mysql -u root -p < Docs_Config/bd.sql
```



## Utilisation

---

### Navigation

L'application propose 4 pages principales :

1. **Accueil** ( / ) - Vue d'ensemble et statistiques
2. **Salariés** ( /salaries ) - Liste et gestion des employés
3. **Nouveau** ( /nouveau ) - Formulaire d'ajout d'employé
4. **Contact** ( /contact ) - Formulaire de contact

## Fonctionnalités principales

### Gestion des salariés

- Utilisez la barre de recherche pour filtrer par nom, email ou poste
- Sélectionnez un département dans le filtre déroulant
- Filtrez par statut (Actif, Remote, Inactif)
- Cliquez sur les en-têtes de colonnes pour trier

- Cliquez sur "Exporter" pour télécharger un CSV

## Ajout d'employé

- Remplissez tous les champs obligatoires (marqués d'un \*)
- Sélectionnez un département et un poste dans les listes déroulantes
- L'ID est généré automatiquement
- Le statut est défini par défaut sur "active"

## Contact

- Remplissez le formulaire de contact
- Les messages sont enregistrés en base de données
- Utilisez le bouton "Exporter" pour télécharger tous les contacts en CSV

## Scripts disponibles

Script	Description
<code>npm run dev</code>	Démarre le serveur de développement frontend (port 8080)
<code>npm run server</code>	Démarre le serveur API backend (port 3000)
<code>npm run dev:full</code>	Démarre frontend et backend simultanément
<code>npm run build</code>	Compile l'application pour la production
<code>npm run build:dev</code>	Compile en mode développement
<code>npm run preview</code>	Prévisualise la build de production
<code>npm run lint</code>	Exécute ESLint pour vérifier le code

## Déploiement

### Build de production

```
npm run build
```

Les fichiers compilés seront dans le dossier `dist/`.

## Déploiement du backend

Le serveur Express peut être déployé sur :

- **Heroku**
- **Railway**
- **Render**
- **VPS** (avec PM2)
- **Azure App Service**

## Variables d'environnement en production

Assurez-vous de configurer toutes les variables d'environnement nécessaires sur votre plateforme de déploiement.

## Exemple avec PM2

```
# Installer PM2
npm install -g pm2

# Démarrer le serveur
pm2 start server/index.js --name "dspi-api"

# Sauvegarder la configuration
pm2 save
```

## Contribution

---

1. Fork le projet
2. Créez une branche pour votre fonctionnalité (`git checkout -b feature/AmazingFeature`)
3. Committez vos changements (`git commit -m 'Add some AmazingFeature'`)
4. Push vers la branche (`git push origin feature/AmazingFeature`)
5. Ouvrez une Pull Request

## Notes

---

- L'application utilise MySQL pour la persistance des données
- Le frontend communique avec l'API via des requêtes HTTP

- Les exports CSV incluent un BOM UTF-8 pour une compatibilité optimale avec Excel
- Les filtres et tris sont appliqués côté client pour une meilleure performance



## Licence

---

Ce projet est privé et propriétaire de DSPI-TECH.



## Auteurs

---

- **DSPI-TECH** - Développement initial



## Support

---

Pour toute question ou problème, contactez l'équipe DSPI-TECH.