# 📦 Planification du projet Cloud-Like user-management-api-cloud

## 🌟 Objectif

Créer une architecture distribuée locale avec **une VM par service**, simulant un environnement cloud. Le frontend est accessible via navigateur, les services sont isolés, communicants entre eux via IP privées.

## 🔧 Architecture prévue

| VM | Rôle | IP Privée | RAM | CPU | Ports VM | Ports internes | Accès Web | Description |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| vm-nginx | Reverse Proxy Web | 192.168.56.10 | 512 MB | 1 | 80 | - | ✅ via navigateur | Route les appels vers Tomcat |
| vm-tomcat | Serveur d’application | 192.168.56.11 | 2 GB | 2 | - | 8080 | ⛘️ direct | Héberge l’API Spring Boot |
| vm-rabbit | Broker RabbitMQ | 192.168.56.12 | 512 MB | 1 | 15672 | 5672 | ✅ admin UI | Gestion de files et messages |
| vm-cache | Memcached | 192.168.56.13 | 512 MB | 1 | - | 11211 | ⛘️ | Caching rapide des données |
| vm-search | Elasticsearch | 192.168.56.14 | 2 GB | 2 | 9200 | 9200 | ✅ REST API | Moteur d’indexation |
| vm-mysql | Base de données | 192.168.56.15 | 1 GB | 1 | - | 3306 | ⚠️ optionnel (phpMyAdmin) | Données persistantes |
| vm-nfs | (optionnel) NFS Server | 192.168.56.16 | 512 MB | 1 | - | 2049, etc. | ⛘️ | Stockage partagé |

## 🧱 Modules à provisionner

Chaque VM aura son propre script de provisioning (setup-\*.sh) dans le dossier provisioning/.

* setup-nginx.sh : installe Nginx + conf proxy
* setup-tomcat.sh : installe Java + déploie backend
* setup-rabbitmq.sh : installe + configure RabbitMQ
* setup-memcache.sh : installe memcached
* setup-elasticsearch.sh : installe ES
* setup-mysql.sh : installe et configure MySQL
* setup-nfs.sh *(optionnel)* : installe et partage /mnt/nfs

## 🌐 Communication inter-VM

* Tomcat ↔ MySQL (192.168.56.15:3306)
* Tomcat ↔ RabbitMQ (192.168.56.12:5672)
* Tomcat ↔ Memcached (192.168.56.13:11211)
* Tomcat ↔ Elasticsearch (192.168.56.14:9200)
* Nginx ↔ Tomcat (192.168.56.11:8080)

## 🌍 Accès attendu depuis l’hôte

| Service | URL | Statut |
| --- | --- | --- |
| Frontend | http://192.168.56.10 | ✅ |
| RabbitMQ UI | http://192.168.56.12:15672 | ✅ |
| Elastic API | http://192.168.56.14:9200 | ✅ |

## 📋 Variables d’environnement (fichier .env)

A conserver tel quel si possible, avec éventuellement :

* RABBITMQ\_HOST=192.168.56.12
* MYSQL\_HOST=192.168.56.15
* CACHE\_HOST=192.168.56.13
* SEARCH\_HOST=192.168.56.14

## 📁 Arborescence projet recommandée

user-management-api-cloud/  
│  
├── backend/  
├── frontend/  
├── provisioning/  
│ ├── nginx/setup-nginx.sh  
│ ├── tomcat/setup-tomcat.sh  
│ ├── rabbitmq/setup-rabbitmq.sh  
│ ├── memcache/setup-memcache.sh  
│ ├── elasticsearch/setup-elasticsearch.sh  
│ ├── mysql/setup-mysql.sh  
│ └── nfs/setup-nfs.sh  
│  
├── .env  
├── Vagrantfile  
├── PLANIFICATION.md  
└── README.md

## ✅ Étapes suivantes

1. Créer le Vagrantfile multi-VM
2. Écrire les scripts setup-\*.sh
3. Adapter application.properties et .env aux IPs internes
4. Tester l’accès web et la communication inter-VM