



# 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, communiquants entre eux via IP privées.



## Architecture prévue

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

## 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