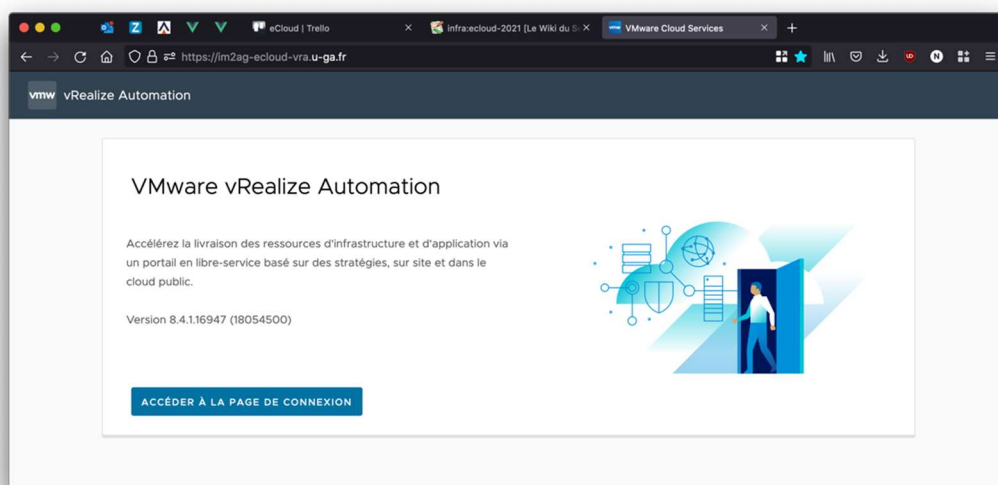


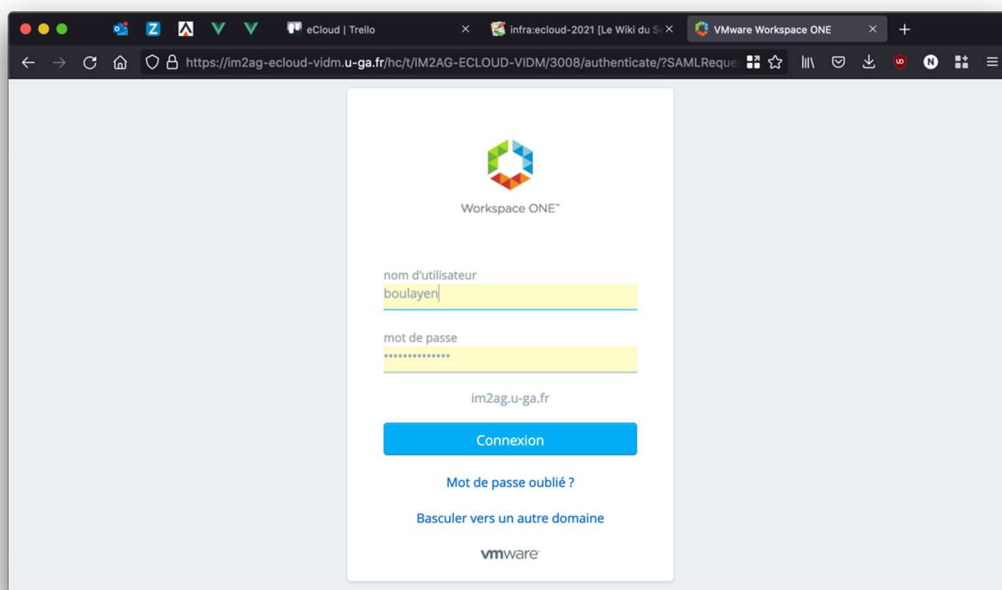
Instanciation de projet sur la plateforme eCloud

1- Accès à la plateforme

eCloud est accessible depuis l'URL <https://im2ag-ecloud-vra.u-ga.fr>



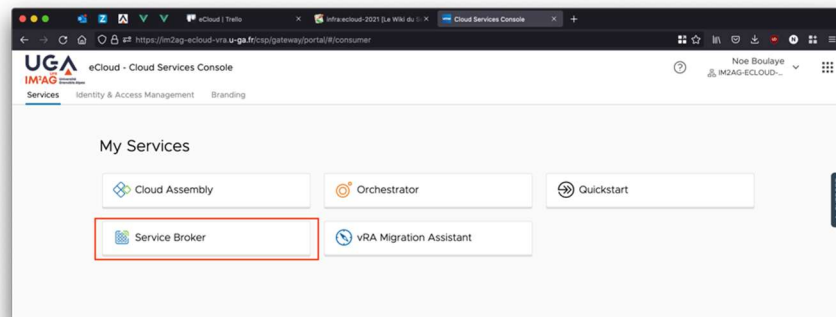
Il faut accéder à la page de connexion



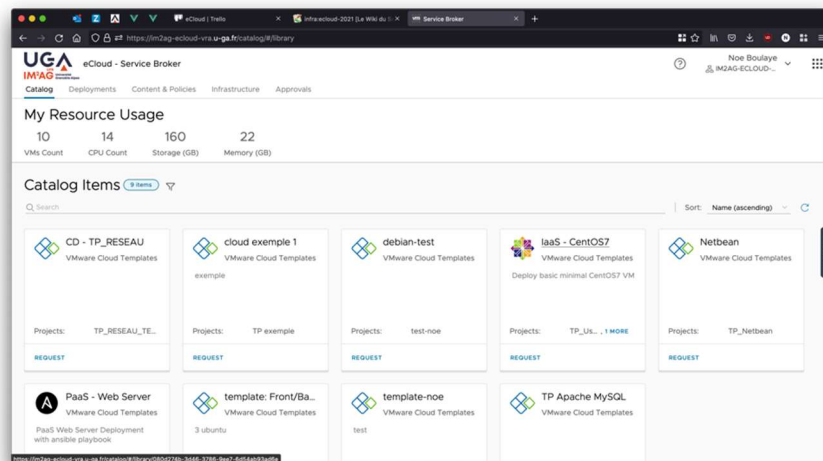
Et s'authentifier avec les identifiants AGALAN

2- Génération d'instance de projet

Une fois connecté, selon le profil la page peut être différente, il faut accéder au « **service Broker** »



Ici le catalogue présente chaque projet qui peut être instancié par l'utilisateur :



Pour instancier un nouveau projet, il faut cliquer sur « **request** » sur celui souhaité
Un formulaire propre au projet demandera des informations, il faudra à minima renseigner un nom de déploiement.

UGA eCloud - Service Broker

Catalog Deployments Content & Policies Infrastructure Approvals

New Request

IaaS - CentOS7 Version 11

Project * TP_Use-Case-1

Deployment Name *

Réseau cible du déploiement labs

Nombre de serveur à déployer 1

3- Accès aux machines du projet

Une fois la demande « submit » effectuée la génération de VM et réseau est initiée.
Il est possible de suivre l'avancement depuis l'onglet « **Deployments** »

UGA Service Broker

Catalog Deployments Content & Policies Infrastructure Approvals

Deployments 1 item

demo

Name	Address	Owner	Project	Status	Expires on	Price
demo		boulayen	TP_Use-Case-1	On	Never	

Après quelques minutes le projet est opérationnel
Il est possible d'accéder à la (ou les) machine virtuelle depuis le même onglet en cliquant sur la **petite flèche à gauche** du nom du projet :

UGA Service Broker

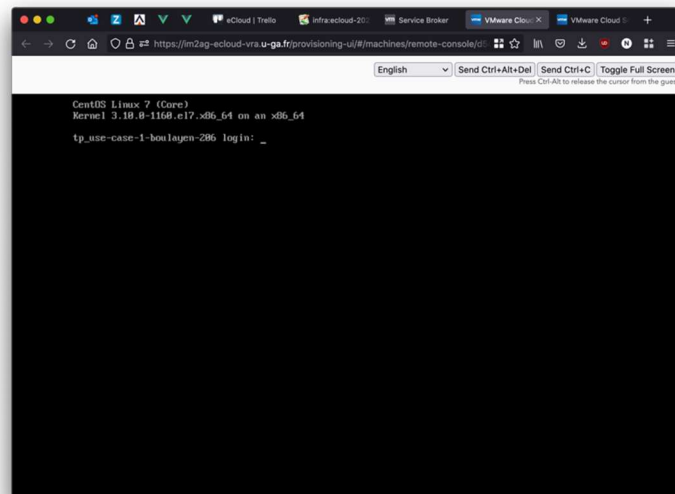
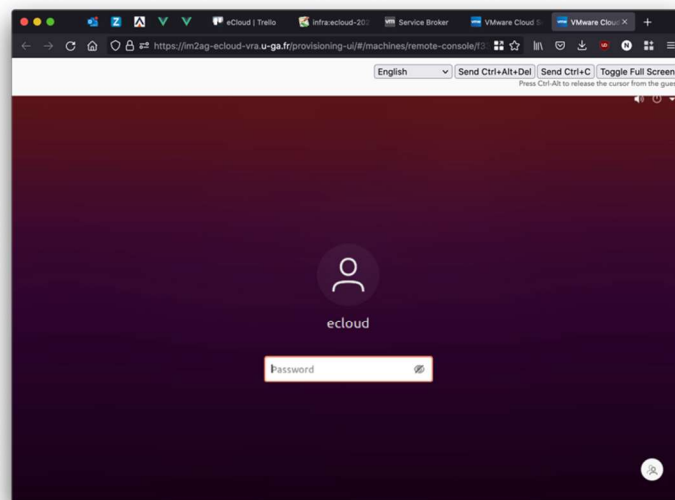
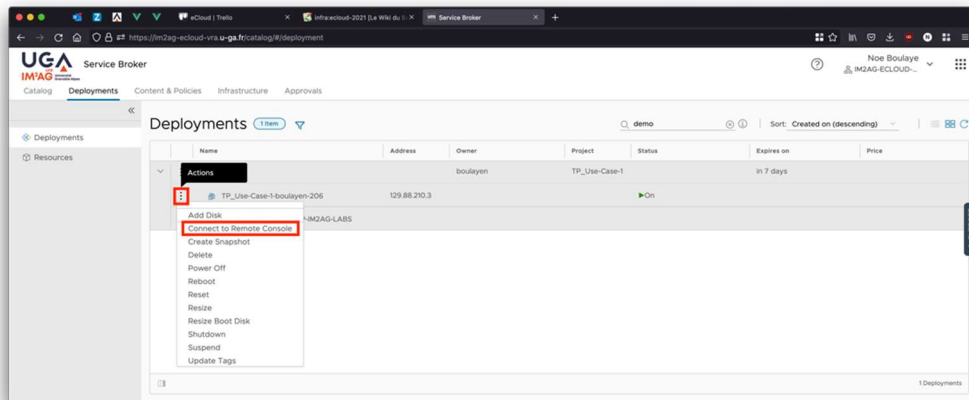
Catalog Deployments Content & Policies Infrastructure Approvals

Deployments 1 item

demo

Name	Address	Owner	Project	Status	Expires on	Price
TP_Use-Case-1-boulayen-206	129.88.210.3			On	in 7 days	
TP-IM2AGIAP-IM2AGIEP-IM2AG-LAB5						

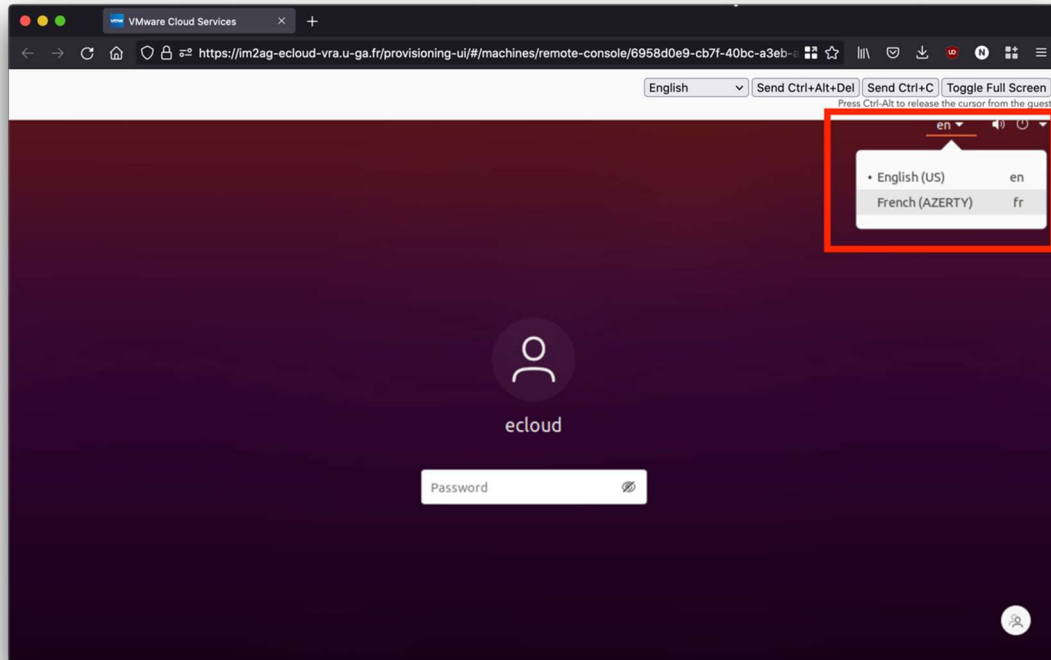
On voit ici l'adresse IP de la VM et son statut (On)
Toujours depuis cet onglet il est maintenant possible d'accéder aux VM avec la console à distance avec le bouton action puis « **connect to remote Console** »



La console peut être avec ou sans interface graphique selon les caractéristiques de la VM du projet.

4- Informations supplémentaires :

- La disposition par défaut du clavier est **QWERTY** mais vous pouvez changer pour **AZERTY** dès la page de login en haut à droite (« en » ou « fr »)



- **À la première connexion** certains programmes peuvent ne pas avoir terminé leur installation, par exemple si VSCode n'est pas disponible immédiatement, **patientez** il apparaîtra dans les applications d'ici **quelques minutes**.
- De même il ne faut pas redémarrer la machine lors des premières minutes après création de la VM.
- **Vous êtes administrateurs** de ces VM grâce à sudo. Nous monitorons l'usage de ces machines, elles doivent rester à usage exclusif des TP pour lesquelles elles vous ont été fournies.
- Nous ne proposons pas de solution de backup.
- Pour vous connecter le user est « ecloud » et le mot de passe : « eCloud2022 » Veillez à changer ce mot de passe lors de votre première connexion (man passwd)
- Vous avez la possibilité de détruire (perte de toutes données) une VM pour en créer une nouvelle, via le « service broker » sur eCloud
- Veillez à ne pas faire cohabiter plusieurs instances d'un projet inutilement.
- Have fun :)

Environnement de développement

L'environnement de développement contient tous les outils préinstallés pour développer un projet industriel :

- JDK 17 + Maven
- npm + node.js
- IDEs : Visual Studio Code, Eclipse
- Administration de Base de Données : DBeaver

Vous pourrez ainsi développer et tester votre application sur une machine virtuelle dédiée. Chaque étudiant dispose de son propre environnement, à vous de vous organiser.

1- Installation du back-end sur la VM

Le back-end contient une application SpringBoot (v.2.6.3) avec un petit bout de code qui appelle la base de données, récupère la première ligne de la table PERSONNE et l'expose via une API REST comme une String.

Cet exemple est un boot minimal qui ne décrit en aucun cas une architecture Spring à reproduire dans vos TPS et projets !

Commissioning de la base de données

Le code est compatible avec un schéma de base simple :

```
USE <YOUR_SCHEMA>;
CREATE TABLE PERSON(
  first_name VARCHAR2(50) NOT NULL,
  last_name VARCHAR2(50) NOT NULL,
  PRIMARY KEY(first_name, LAST_name)
);
INSERT INTO PERSON VALUES ('<YOUR_FIRST_NAME>', '<YOUR_LAST_NAME>');
SELECT * FROM PERSON;
```

Vous pouvez utiliser dbeaver qui est installé sur la VM

Installation du projet

Pour installer le back-end depuis gitlab, exécuter sur la VM :

```
>git clone https://gricad-gitlab.univ-grenoble-alpes.fr/saintmch/uga\_miage\_ecloud\_back\_end.git
```

Changer la configuration de la datasource dans le fichier
src/main/resources/application.properties:

```
spring.datasource.url=jdbc:oracle:thin:@im2ag-oracle.e.ujf-grenoble.fr:1521:IM2AG
spring.datasource.username=<YOUR_USERNAME>
spring.datasource.password=<YOUR_ORACLE_TOKEN>
spring.profiles.active=dev
```

Pour retrouver votre token, c'est ici :

<https://im2ag-wiki.univ-grenoble-alpes.fr/doku.php?id=environnements:oracle>

Compiler le serveur :

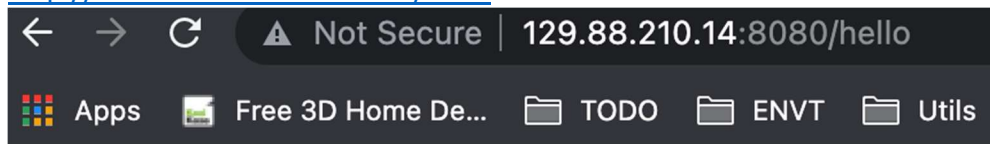
```
>mvn clean install
```

Lancer le serveur :

```
>mvn spring-boot:run
```

Pour tester l'installation, vous pouvez taper cette URL :

<http://<URL DE LA VM>:8080/hello>



"Christophe Saint Marcel"

2- Installation du front-end sur la VM

Le Front-end est basé sur le framework Angular (v.9.0.4).

Installation du projet

Pour installer le front-end depuis gitlab, exécuter sur la VM :

```
>git clone https://gricad-gitlab.univ-grenoble-alpes.fr/saintmch/uga\_miage\_ecloud\_front\_end.git
```

Lancer :

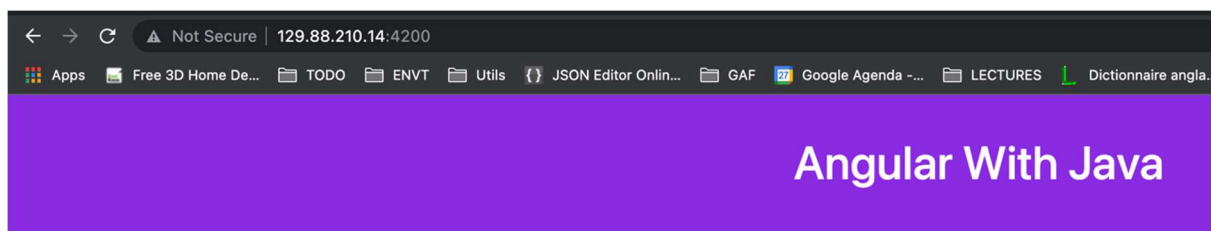
```
>npm install
```

Puis :

```
>npm start
```

Pour tester l'installation, vous pouvez taper cette URL :

<http://<URL DE LA VM>:4200/>



Hello Christophe Saint Marcel

It is now up to you!

3- Configuration de l'environnement

IDE Eclipse

Le plugin Lombok utilisé sur les projets n'est pas encore intégré à l'image de la VM (In progress ...).

Il faut donc bricoler un peu en attendant pour avoir un projet compilable sur la VM.

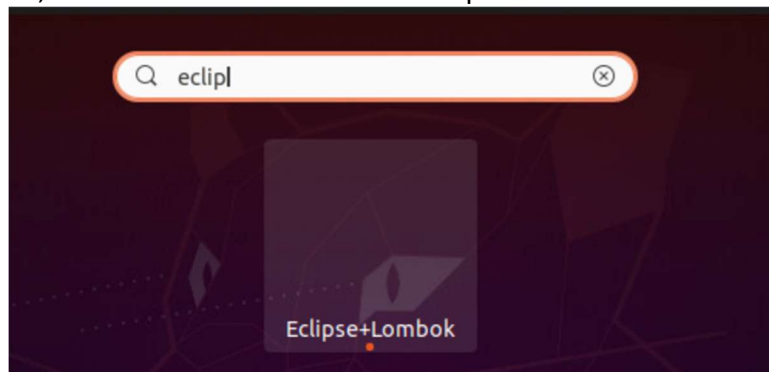
Cela va consister à suivre le guide :

<https://www.xceno.io/posts/2019-05-15-install-lombok-to-eclipse-snap>

A partir du répertoire eclipse du projet back-end que vous avez cloné :

- Copier le fichier eclipse_eclipse.desktop sous /home/ecloud/.local/share/applications/
- Copier les fichiers Lombok.jar et eclipse.ini sous /home/ecloud/snap/eclipse/current

Pour lancer Eclipse, il faudra utiliser le raccourci Eclipse+Lombok



Pour vérifier qu'Eclipse est bien configuré, aller dans l'onglet Help/About Eclipse IDE, vous devez voir que Lombok s'est correctement installé :



IDE VSCode

TODO !

4- TIPS

- Vous pouvez vous connecter en ssh sur la machine. Le mode graphique marche bien mais peut parfois être un peu lent.

- On ne peut pas faire de copier-coller entre votre machine et la VM, vous allez vous rendre compte que c'est une limitation importante. Le transfert de fichiers entre votre machine et la VM peut se faire par scp en utilisant un fichier tampon :

Exemple de transfert VM->Machine

```
>scp ecloud@129.88.210.14:/home/ecloud/Downloads/copy-paste.txt
```

Exemple de transfert Machine->VM

```
>scp copy-paste.txt ecloud@129.88.210.14:/home/ecloud/Downloads/
```

- Pour accéder au serveur ngserve depuis une machine distante, il faut positionner l'option host dans le fichier angular.json :

```
"serve": {  
  "builder": "@angular-devkit/build-angular:dev-server",  
  "options": {  
    "browserTarget": "angular-nodejs-example:build",  
    "proxyConfig": "proxy.conf.json",  
    "host": "0.0.0.0",  
    "port": 4200  
  },  
}
```

5- Références

<https://medium.com/bb-tutorials-and-thoughts/how-to-develop-and-build-angular-app-with-java-backend-87fb603c6e17>

<https://www.xceno.io/posts/2019-05-15-install-lombok-to-eclipse-snap>