# Installation local du serveur Keycloak

Lien vers la documentation Keycloak :

<https://www.keycloak.org/getting-started/getting-started-docker>

Commande un peu modifiée à lancer :

* docker run -d -p 4000:8080 -e KC\_BOOTSTRAP\_ADMIN\_USERNAME=admin -e KC\_BOOTSTRAP\_ADMIN\_PASSWORD=admin quay.io/keycloak/keycloak:26.1.0 start-dev

docker images

Une image contenant texte, logiciel, Police, Logiciel multimédia

Description générée automatiquement

docker ps

Une image contenant texte, logiciel, Logiciel multimédia, capture d’écran

Description générée automatiquement

Le conteneur est bien démarré, rendez-vous sur <http://localhost:4000>

**Username :** admin

**Password :** admin

# Configuration du serveur Keycloak

Nous sommes sur l’application Keycloak :

Une image contenant texte, logiciel, Icône d’ordinateur, Logiciel multimédia

Description générée automatiquement

Première chose à faire, créer un nouveau Realm

Une image contenant texte, capture d’écran, Police, nombre

Description générée automatiquement

Une image contenant texte, capture d’écran

Description générée automatiquement

Ensuite ouvrez l’onglet Clients > Create client

Une image contenant texte, capture d’écran, Police

Description générée automatiquement

Next > Next > Save (on configurera plus tard …)

# Création de l’application Client (REACT)

Création de l’application React, vous pouvez cloner mon repo GitHub :

<mettre les liens vers mon repo>

Dans le .env de l’application client (react) bien remplir les variables d’environnement paramétrées plus tôt sur keycloak :

VITE\_KEYCLOAK\_URL=http://localhost:4000/

VITE\_KEYCLOAK\_REALM=myrealm

VITE\_KEYCLOAK\_CLIENTID=myclient

Aller dans Keycloak sur myrealm.

Dans Clients > myclient :

Mettre l’url du frontend dans les valid URIs :

Une image contenant texte, logiciel, capture d’écran, nombre

Description générée automatiquement

**BIEN PENSER A SAVE !**

# Stylisation de la page de connexion

On change le nom affiché ici dans le Realm settings

Une image contenant texte, capture d’écran, nombre, Police

Description générée automatiquement

Le résultat est imminant sur la page de Login sur le client :

Une image contenant texte, capture d’écran, Police, conception

Description générée automatiquement

Dans Realm Settings > Login

Activer le Boolean « **User registration** »

Permet d’ajouter la possibilité de s’inscrire :

Une image contenant texte, capture d’écran, Police, nombre

Description générée automatiquement

**Il y a pas mal d’option disponible intéressante, permettant d’éviter de développer beaucoup de sous module de login/logout/register !**

**Une image contenant texte, capture d’écran, logiciel, Page web

Description générée automatiquement**

Possibilité le modifier les pages dédiés avec des thème préfait de keycloak

# Passons à la partie Authentification des utilisateurs

Une image contenant texte, capture d’écran, logiciel, Icône d’ordinateur

Description générée automatiquement

Créer un nouvel utilisateur :

Une image contenant texte, capture d’écran, nombre, Police

Description générée automatiquement

**Aller dans Credentials**

Une image contenant texte, capture d’écran, Police, nombre

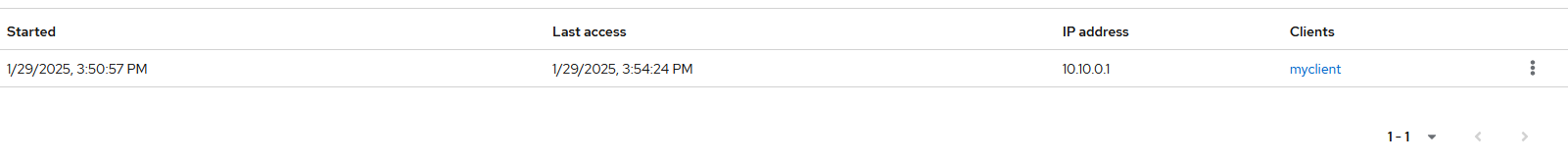
Description générée automatiquement

Se loger dans l’application React :

Une image contenant texte, capture d’écran, Police, blanc

Description générée automatiquement

**Dans Users > Sessions, on devrait voirs que notre client est bien logé**



Possible de déconnecter l’user depuis ici d’ailleurs !

# Mise en place d’une politique de mot de passe

**Authentication > Policies**

Les settings, se définissent ici :

Une image contenant texte, capture d’écran, logiciel, Page web

Description générée automatiquement

Exemple de config, que l’on trouve souvent :

Une image contenant texte, capture d’écran, logiciel, nombre

Description générée automatiquement

Nécessite de changer le password de mon utilisateur

Une image contenant texte, capture d’écran, logiciel, affichage

Description générée automatiquement

# Mise en place d’un 2FA

**Authentication > Required actions**

Une image contenant texte, ligne, Police, nombre

Description générée automatiquement

Test après avoir delete mon utilisateur.

Une image contenant texte, capture d’écran, Police, nombre

Description générée automatiquement

Direct il y a la possibilité de paramétrer son OTP :

Une image contenant texte, capture d’écran, Police

Description générée automatiquement

# Sécurité du serveur

Realm Settings > Keys, on voit les différentes clés de chiffrements disponibles

Une image contenant texte, capture d’écran, nombre, Police

Description générée automatiquement

**Je copie la clé publique du certificat RS256**

Je peux donc utiliser cette clé publique pour restreindre mes routes nodejs, aux utilisateurs authentifiés !

# Création de l’application Serveur + Mise à jour du Client

<voir repo github>

ATTENTION penser à ajouter le proxy dans vite.config.js pour éviter les erreurs cors.

# Sécurité des routes

Copier la clé public RS256 dans Realm Settings > Keys

Dans les variables d’environnements :

PORT=5000

PUBLICKEY=MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEA8yBnW+hft12q7VLksVl2CEA1fu/GPVI4oudLbnhfuD/ZnWnnyPk2WjjyLDbtaAia2tAi92a0Sb/XirRzmpAVNmE+Hd71bUIIqT2qIyECi83cCBOCPgsEo/EmWD9sCmjhlNcXmDd3lAgf2OtDDKdBzz0vGhqM9Lcs1jmAJm61MxRhANddl2hOq2pfov7Es/nL8DsUTfWk9+uwfDWMPJElaVPNBNPzl10fDv57SL8t9aievJSLwzMy0BovmQtuHoJs2aWqGdK2juRlBmGhS2FUD2QPqa7A7gKnMZAztD3S7RBDk39hYr9umwaMvK689wTp1RBcl9iXjZ26LP4HRn0xGwIDAQAB

# Backup du serveur keycloak

Dans Realm settings :

Une image contenant texte, capture d’écran, logiciel, Police

Description générée automatiquement

Une image contenant texte, capture d’écran

Description générée automatiquement

Une image contenant texte, logiciel, Logiciel multimédia, Logiciel de graphisme

Description générée automatiquement