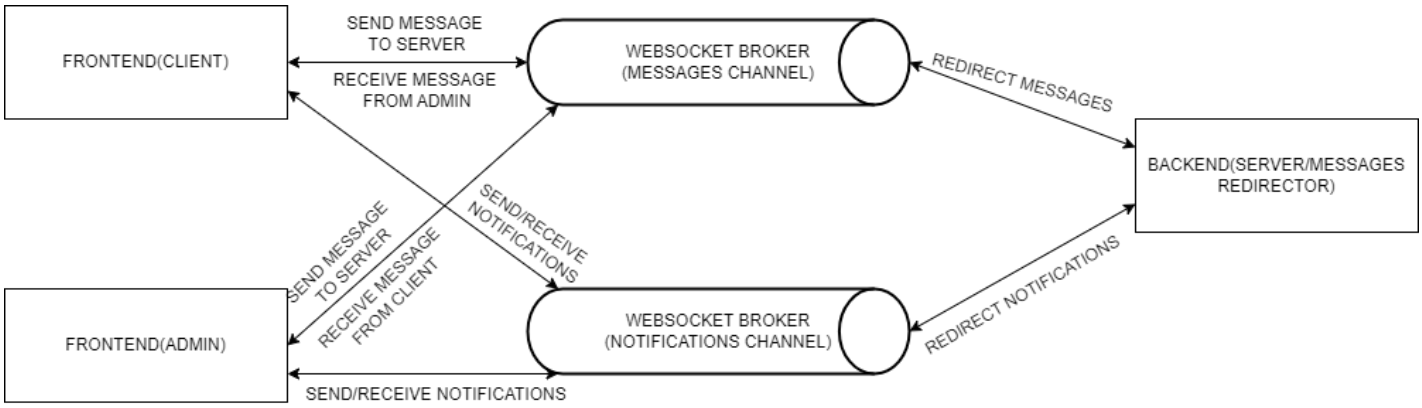
DOCUMENTAȚIE ASSIGNMENT 3 – ONLINE PLATFORM

Realizator: Pop Alexandra

Grupa: 30641, TI

1. Arhitectura conceptuală



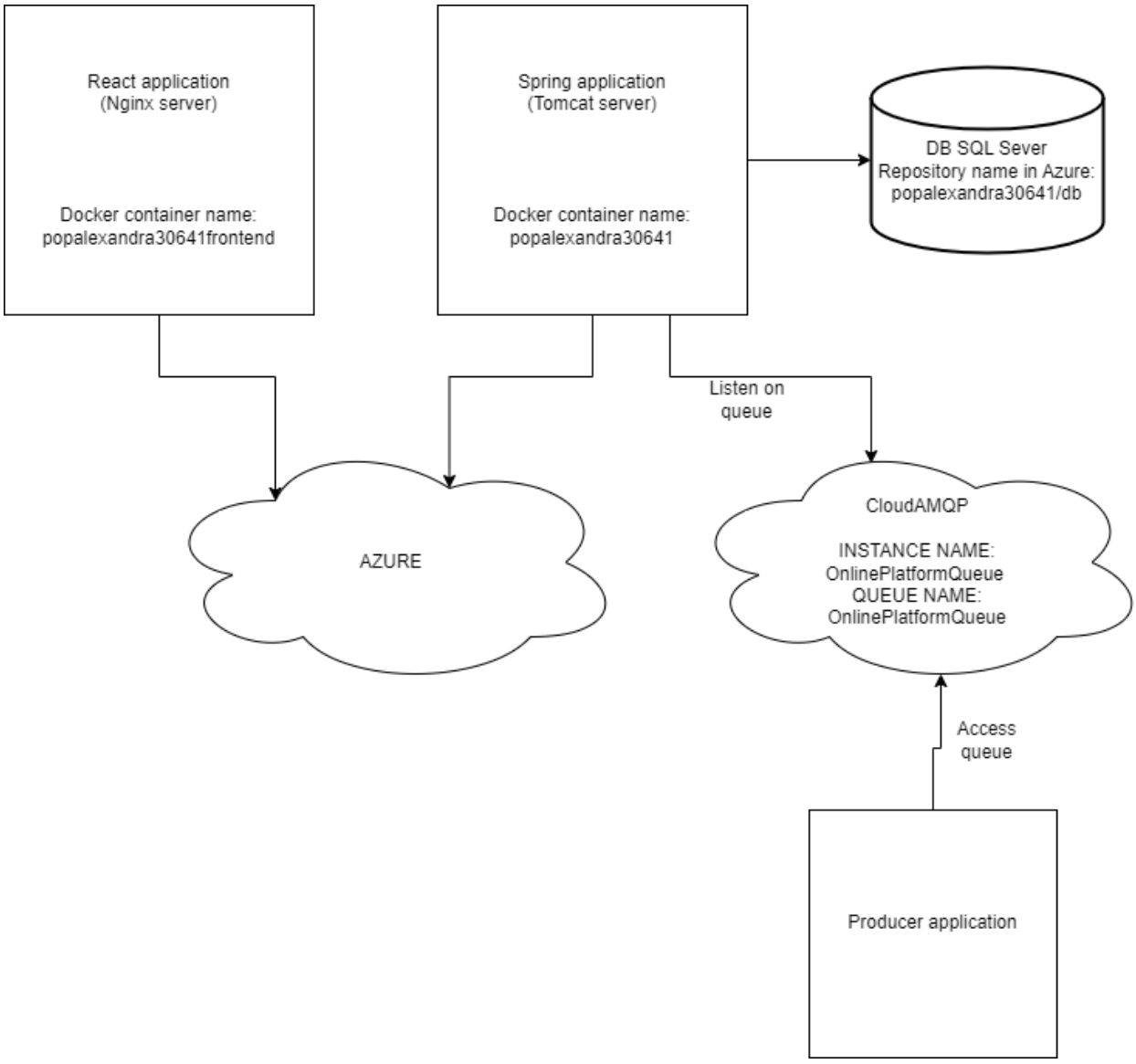
Arhitectura sistemului distribuit realizat are în plus componentele prezentate în imaginea de mai sus, ce compun funcționalitățile unui chat între un client și un admin al aplicației, acestea fiind integrate în aplicațiile de backend și frontend de până acum.

Pentru comunicarea cu mesaje între clienți și admin, este disponibilă doar comunicarea admin-ului cu un singur client la un moment dat, însă inițiativa de comunicare trebuie să vină de la client, admin-ul neavând capacitatea de a începe o conversație (nu are partea de chat disponibilă până la primirea unui mesaj de la un client). Schimbarea de mesaje e făcută prin websockets, printr-un canal separat de cel de notificări de energie și notificări de chat. Orice user poate apela o metodă post din backend pentru ca acesta să redirecționeze mesajul prin canalul destinat mesajelor către user-ul de la celălalt capăt (în funcție de id-ul dat ca parametru în corpul dto-ului mesajului).

În plus, userii participanți la conversația din chat pot vedea când celălalt participant scrie un mesaj înapoi (”typing”), sau când acesta i-a citit mesajele trimise (”seen”) cu ajutorul unor notificări transmise tot prin intermediul websocket-urilor între useri. Când un user scrie un mesaj, se apelează o metodă post specifică, și informația este redirecționată către celălalt utilizator. La fel și pentru ”seen”, ambele tipuri de notificări având constrângeri în plus pentru o funcționare corectă.

Când backend-ul primește un request de la frontend pe unul din canalele de mesaje sau notificări de chat, acesta face broadcast cu mesajul/notificarea de transmis tuturor utilizatorilor conectați la canal, iar în frontend se face filtrarea acestor mesaje/notificări pentru fiecare user în funcție de destinatarul lor (id-ul user-ului către care a fost trimis să fie egal cu id-ul user-ului conectat).

2. UML Deployment diagram



Aplicațiile de frontend, backend și baza de date sunt puse pe cloud-ul de Azure, în aceleași containere de la assignment-ul 1 si 2, însă am făcut un nou release cu datele nou adăugate (brokerele din backend pentru mesaje și notificări, controller-ul pentru transmitere de mesaje de la frontend la backend și invers ce conține două metode de tip ”post”, și dto-ul pentru mesajele transmise ce conține: id-ul și numele complet al celui care trimite mesajul, id-ul destinatarului și mesajul propriu-zis).

Aplicația de sender este rulată pe mașina locală și transmite datele într-o coadă care este stocată în CloudAMQP.

Backend-ul și sender-ul pot comunica foarte ușor deoarece coada este într-un mediu cloud, nu local.

3. Readme file containing build and execution considerations

Pentru rularea aplicației de backend pe mașina locală, trebuie mai întâi să se introducă comanda ”git clone https://github.com/DS202230641PopAlexandra/DS2022_30641_Pop_Alexandra_1_Backend.git” în fișierul în care se dorește stocarea proiectului. Apoi se deschide proiectul în IntelliJ și se rulează fișierul ”DS2020Application.java”, iar aplicația va fi funcțională.

Pentru rularea aplicației de frontend, trebuie utilizată comanda de ”git clone https://github.com/DS202230641PopAlexandra/DS2022_30641_Pop_Alexandra_1_Frontend.git” în folder-ul dorit, apoi după deschiderea proiectului în ide-ul IntelliJ, trebuie rulate în terminalul ide-ului comenzile ”npm install” pentru ca toate librăriile specificate în ”package.json” și ”package-lock.json” să fie instalate (acestă comandă trebuie rulată la fiecare reclonare a proiectului), și ”npm run start” pentru a porni aplicația (se va deschide automat pagina html principală în browser-ul vostru default).

Pentru ca aplicația de frontend (react app) să aibă acces la datele din baza de date și pentru a putea trimite request-uri, trebuie ca aplicația de backend să fie funcțională.

Simulatorul de device-uri se poate clona pe masina locală cu comanda ”git clone https://github.com/DS202230641PopAlexandra/DS2022_30641_Pop_Alexandra_2.git” în fișierul în care se dorește stocarea proiectului. Prin pornirea acestuia, se va începe transmiterea de date, ele putând fi văzute în consola proiectului sau în consola backend-ului.