**A Step-by-step Guideline on Containerizing the Microservices and Upload on Artifact Registry**

**Containerizing Microservices (Use Auth as example)**

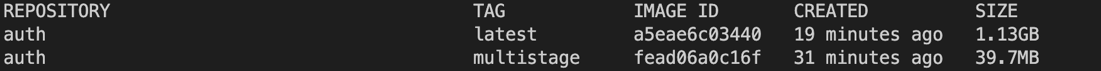
1. In each microservice folder, add a new DockerFile and DockerFile.multistage
2. Make sure the firebase json file is also copy over to the new directory – app by adding this command in the DockerFile and multistage file:

COPY eti-assignment-2-firebase-adminsdk-6r9lk-85fb98eda4.json /app/

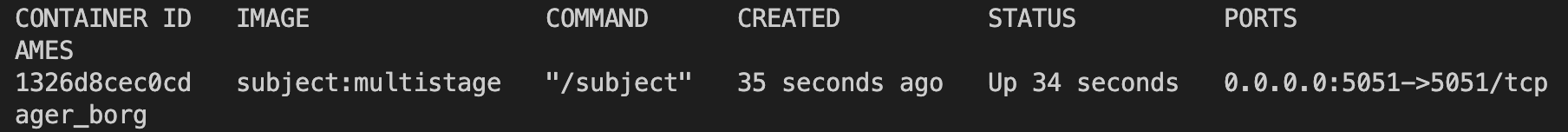
1. Build the image with a tag auth

docker build --tag auth .

1. Build the multi-stage image with the same tag to reduce the image size

docker build -t auth:multistage -f Dockerfile.multistage .  
docker build -t subject:multistage -f Dockerfile.multistage .  
docker build -t tutoring:multistage -f Dockerfile.multistage .  
docker build -t chatting:multistage -f Dockerfile.multistage .  
docker build -t payment:multistage -f Dockerfile.multistage .  


1. Run the image in a container in detach mode

docker run -d -p 5050:5050 auth:multistage  
docker run -d -p 5051:5051 subject:multistage  
docker run -d -p 5052:5052 tutoring:multistage  
docker run -d -p 5053:5053 chatting:multistage  
docker run -d -p 5054:5054 payment:multistage  


Text

Description automatically generated

After containerized all the microservices

A picture containing graphical user interface

Description automatically generated

Publish on Docker Registry

Graphical user interface, application, Teams

Description automatically generated

Tagging the image name to the registry

**docker tag auth:multistage chenhannp/auth  
docker tag subject:multistage chenhannp/subject  
docker tag tutoring:multistage chenhannp/tutoring  
docker tag chatting:multistage chenhannp/chatting  
docker tag payment:multistage chenhannp/payment**

**Text

Description automatically generated**

Push to the registry

**docker push chenhannp/auth  
docker push chenhannp/subject  
docker push chenhannp/tutoring  
docker push chenhannp/chatting  
docker push chenhannp/payment**

**Graphical user interface, application, Teams

Description automatically generated**

Deploy image On Google Cloud Platform  
1. Artifact Registry

Graphical user interface, text, application, email

Description automatically generated

2. Pull image from docker registry

docker pull chenhannp/auth;docker pull chenhannp/subject;docker pull chenhannp/tutoring;docker pull chenhannp/chatting;docker pull chenhannp/payment;docker pull chenhannp/react-app

3. Tag the image with a new path to the registry

docker tag chenhannp/auth asia-southeast1-docker.pkg.dev/eti-assignment-2-376806/docker-auth/auth;docker tag chenhannp/subject asia-southeast1-docker.pkg.dev/eti-assignment-2-376806/docker-subject/subject;docker tag chenhannp/tutoring asia-southeast1-docker.pkg.dev/eti-assignment-2-376806/docker-tutoring/tutoring;docker tag chenhannp/chatting asia-southeast1-docker.pkg.dev/eti-assignment-2-376806/docker-chatting/chatting;docker tag chenhannp/payment asia-southeast1-docker.pkg.dev/eti-assignment-2-376806/docker-payment/payment; docker tag chenhannp/react-app asia-southeast1-docker.pkg.dev/eti-assignment-2-376806/docker-react-app/react-app

4. Push to the artifact registry

docker push asia-southeast1-docker.pkg.dev/eti-assignment-2-376806/docker-auth/auth;docker push asia-southeast1-docker.pkg.dev/eti-assignment-2-376806/docker-subject/subject;docker push asia-southeast1-docker.pkg.dev/eti-assignment-2-376806/docker-tutoring/tutoring;docker push asia-southeast1-docker.pkg.dev/eti-assignment-2-376806/docker-chatting/chatting;docker push asia-southeast1-docker.pkg.dev/eti-assignment-2-376806/docker-payment/payment; docker push asia-southeast1-docker.pkg.dev/eti-assignment-2-376806/docker-react-app/react-app

Graphical user interface, application

Description automatically generated

4. Deploy the container on Google Cloud Run

4 CPUS for Chatting microservice

Graphical user interface, text, application, email

Description automatically generated

Graphical user interface, text, application, email

Description automatically generated

Graphical user interface, text, application, email

Description automatically generated

URLs:

Auth: <https://auth-4dcnj7fm6a-uc.a.run.app>

Subject: <https://subject-4dcnj7fm6a-uc.a.run.app>

Tutoring: <https://tutoring-4dcnj7fm6a-uc.a.run.app>

Chatting: <https://chatting-4dcnj7fm6a-uc.a.run.app>

Payment: <https://payment-4dcnj7fm6a-uc.a.run.app>

Test with postman:

**Text

Description automatically generated**

**Containerize the React App**

**Added Dockerfile and .dockerignore**

docker build -t react-app:latest .

docker tag react-app chenhannp/react-app

docker push chenhannp/react-app

**In Artifact Registry**

docker pull chenhannp/react-app

docker tag chenhannp/react-app asia-southeast1-docker.pkg.dev/eti-assignment-2-376806/docker-react-app/react-app

docker push asia-southeast1-docker.pkg.dev/eti-assignment-2-376806/docker-react-app/react-app

Graphical user interface, text, application, email

Description automatically generated

<https://react-app-4dcnj7fm6a-uc.a.run.app>

fixed the problem of having to on CORS middleware to access the microservices:

* added cors middleware package in Go

fixed the problem of react page reload failed with 404

* added nginxconf file and added with dockerfile

Graphical user interface, application, email

Description automatically generated

**Commands to run as a whole**

1. Build images

docker build -t auth:multistage -f Dockerfile.multistage .; **docker tag auth:multistage chenhannp/auth; docker push chenhannp/auth**  
docker build -t subject:multistage -f Dockerfile.multistage .; **docker tag subject:multistage chenhannp/subject; docker push chenhannp/subject**  
docker build -t tutoring:multistage -f Dockerfile.multistage .; **docker tag tutoring:multistage chenhannp/tutoring; docker push chenhannp/tutoring**  
docker build -t chatting:multistage -f Dockerfile.multistage . ;**docker tag chatting:multistage chenhannp/chatting; docker push chenhannp/chatting**  
docker build -t payment:multistage -f Dockerfile.multistage .; **docker tag payment:multistage chenhannp/payment; docker push chenhannp/payment**docker build -t react-app:latest .;docker tag react-app chenhannp/react-app;docker push chenhannp/react-app

2. Pull image from docker registry

docker pull chenhannp/auth;

docker pull chenhannp/subject;

docker pull chenhannp/tutoring;

docker pull chenhannp/chatting;

docker pull chenhannp/payment;

docker pull chenhannp/react-app

3. Tag the image with a new path to the registry

docker tag chenhannp/auth asia-southeast1-docker.pkg.dev/eti-assignment-2-376806/docker-auth/auth;

docker tag chenhannp/subject asia-southeast1-docker.pkg.dev/eti-assignment-2-376806/docker-subject/subject;

docker tag chenhannp/tutoring asia-southeast1-docker.pkg.dev/eti-assignment-2-376806/docker-tutoring/tutoring;

docker tag chenhannp/chatting asia-southeast1-docker.pkg.dev/eti-assignment-2-376806/docker-chatting/chatting;

docker tag chenhannp/payment asia-southeast1-docker.pkg.dev/eti-assignment-2-376806/docker-payment/payment;

docker tag chenhannp/react-app asia-southeast1-docker.pkg.dev/eti-assignment-2-376806/docker-react-app/react-app

4. Push to the artifact registry

docker push asia-southeast1-docker.pkg.dev/eti-assignment-2-376806/docker-auth/auth;

docker push asia-southeast1-docker.pkg.dev/eti-assignment-2-376806/docker-subject/subject;

docker push asia-southeast1-docker.pkg.dev/eti-assignment-2-376806/docker-tutoring/tutoring;

docker push asia-southeast1-docker.pkg.dev/eti-assignment-2-376806/docker-chatting/chatting;

docker push asia-southeast1-docker.pkg.dev/eti-assignment-2-376806/docker-payment/payment;

docker push asia-southeast1-docker.pkg.dev/eti-assignment-2-376806/docker-react-app/react-app