# Configure my java spring boot application

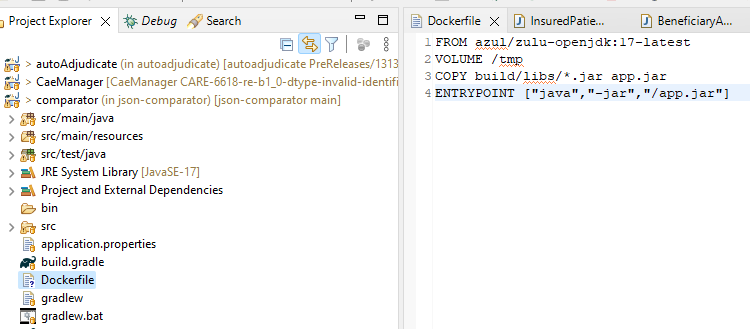
1. **Add docker file to our java spring boot application**

FROM azul/zulu-openjdk:17-latest

VOLUME /tmp

COPY build/libs/\*.jar app.jar

ENTRYPOINT ["java","-jar","/app.jar"]

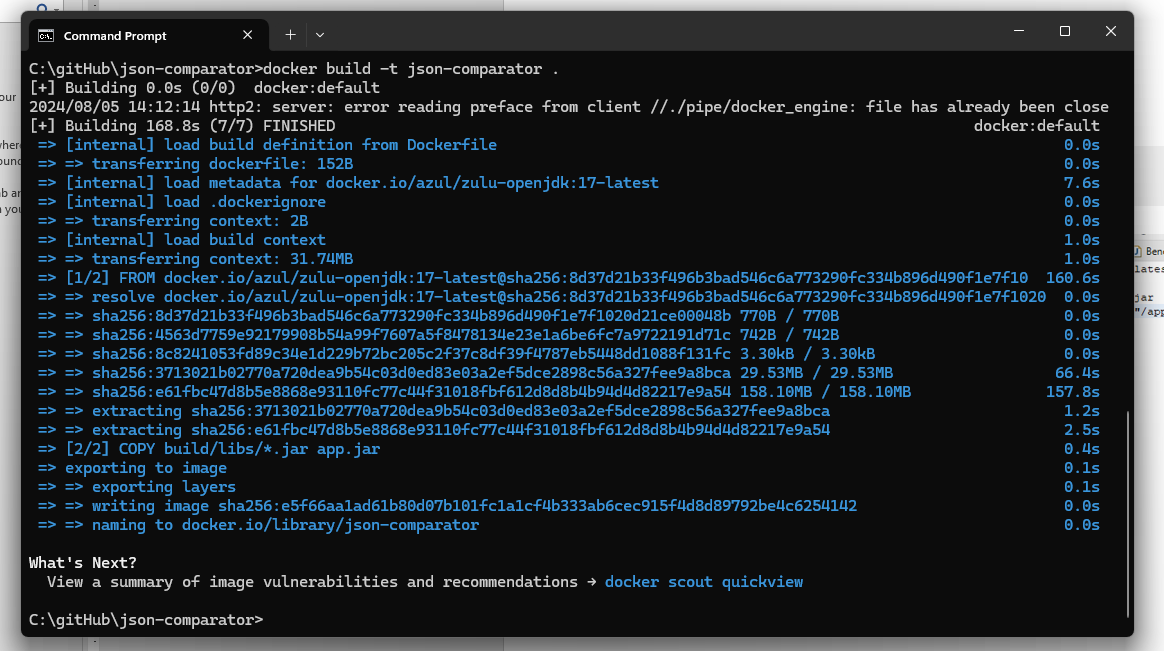
****

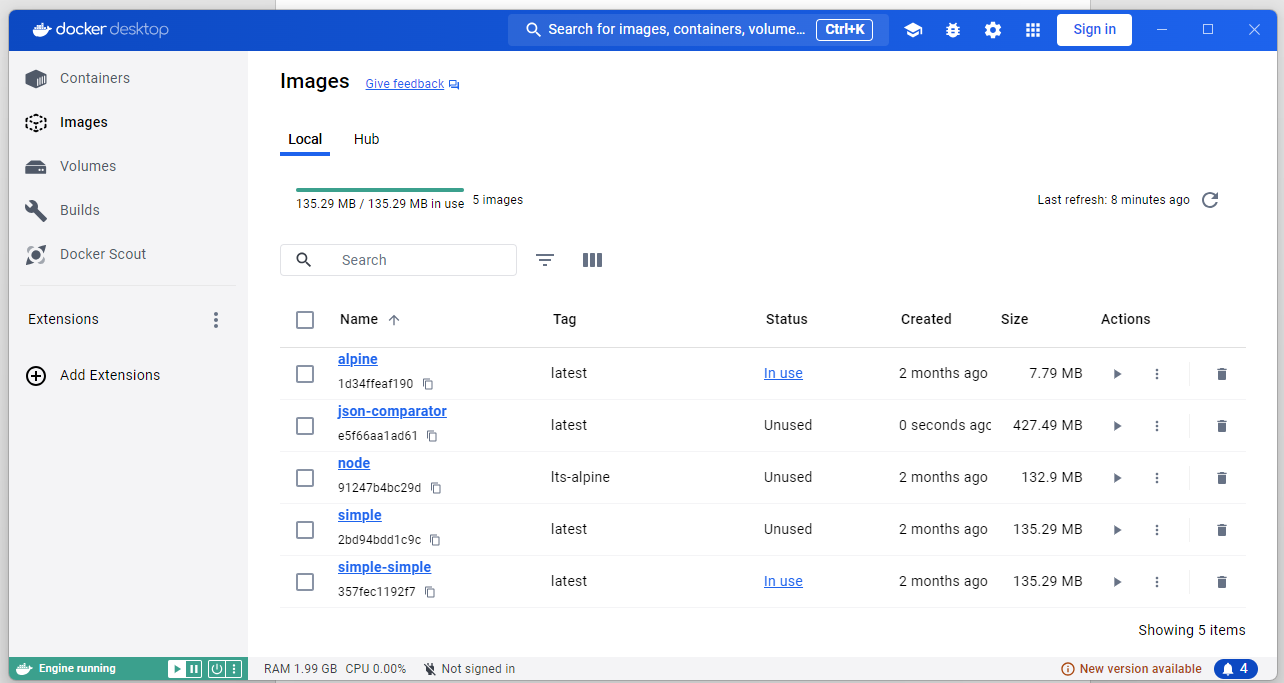
# Create Docker iamge

1. **Build image**

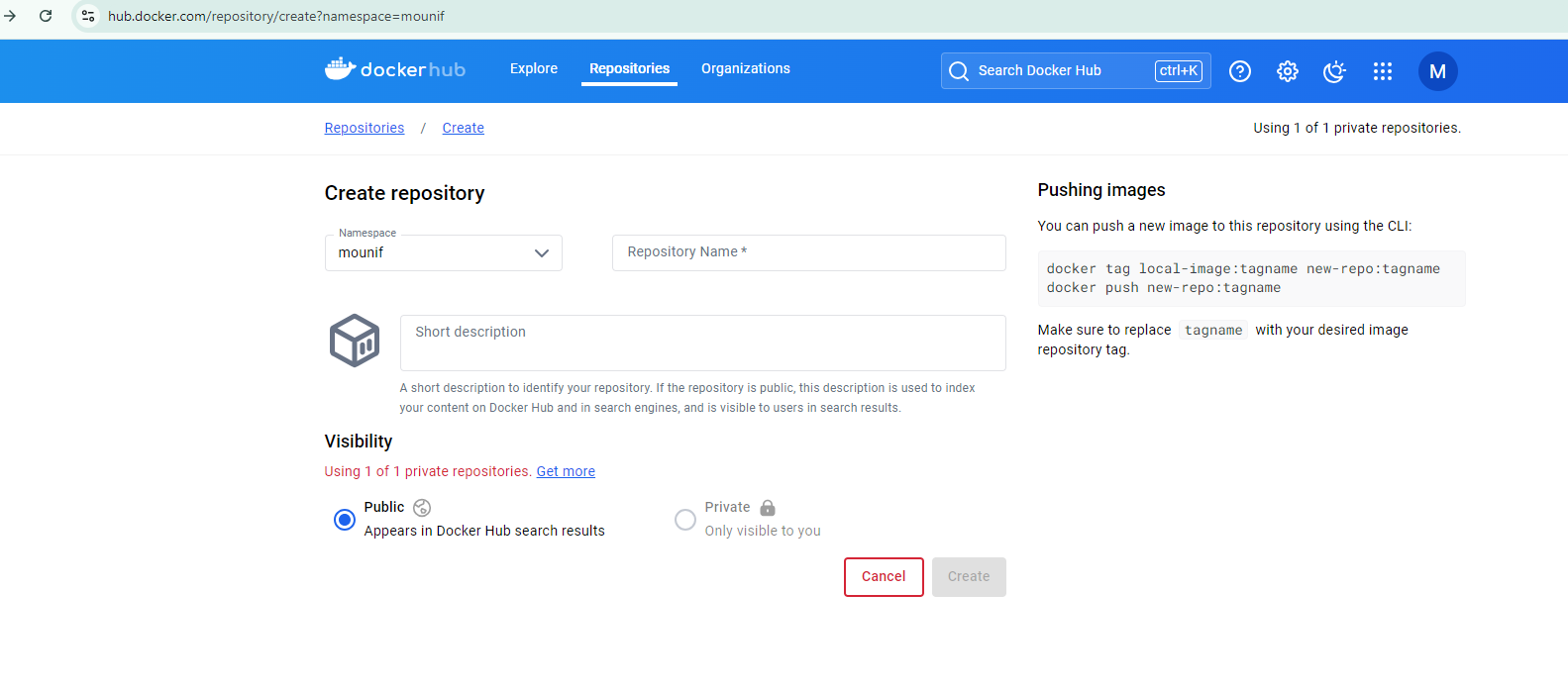
**$ Cd C:\gitHub\json-comparator**

**$ docker build -t json-comparator .**





If you don’t have a docker repository, Go to docker hub and create a new one.

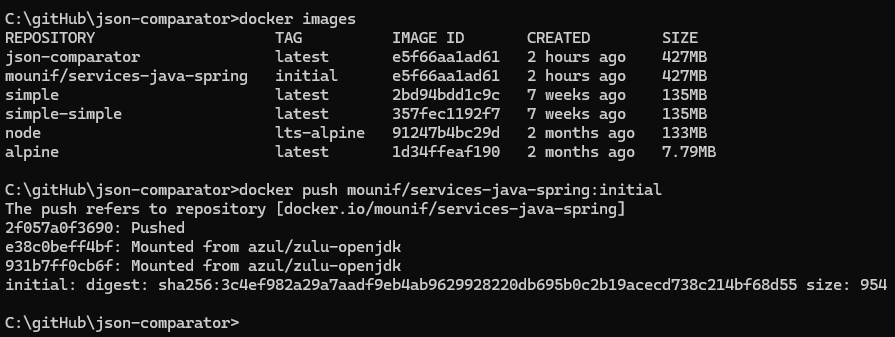


**Tag the Docker image for your repository**

$ docker tag json-comparator mounif/json-comparator:initial

**Docker hub image upload**

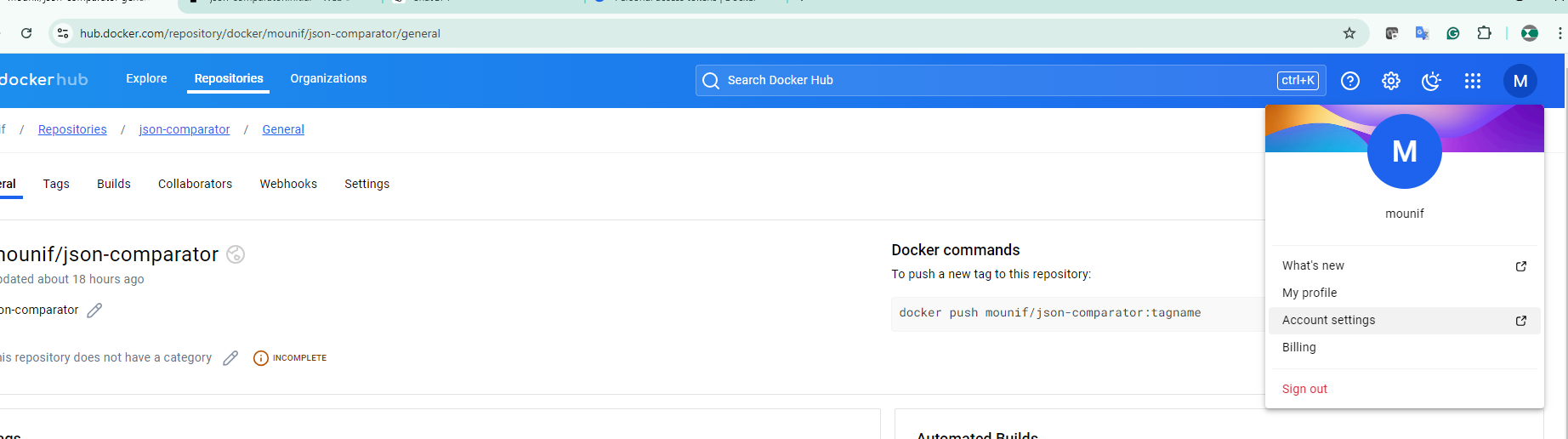
$ docker push mounif/json-comparator:initial



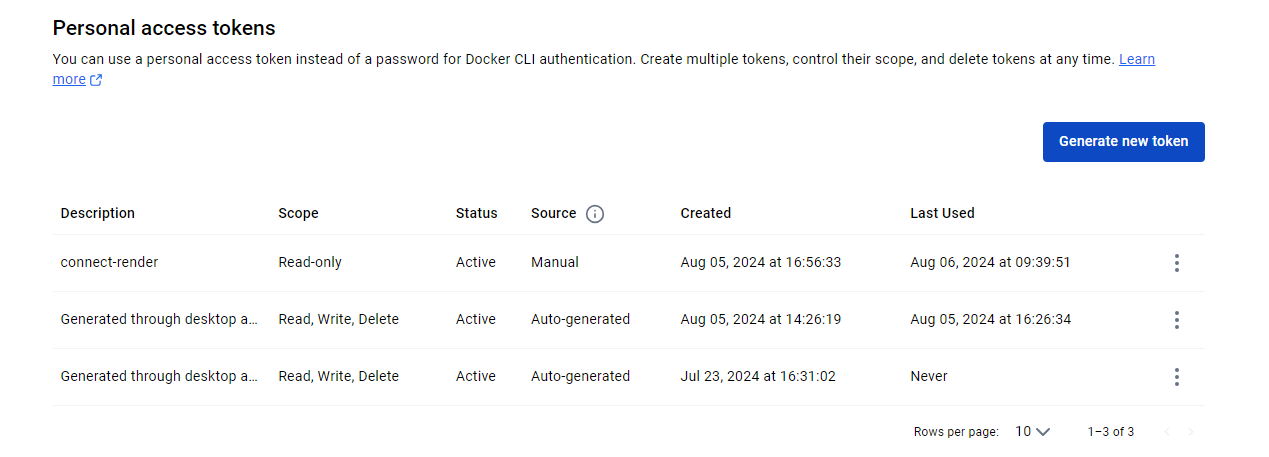
To connect render with your docker repository, you to generate a new token as first

Create Token in docker hub

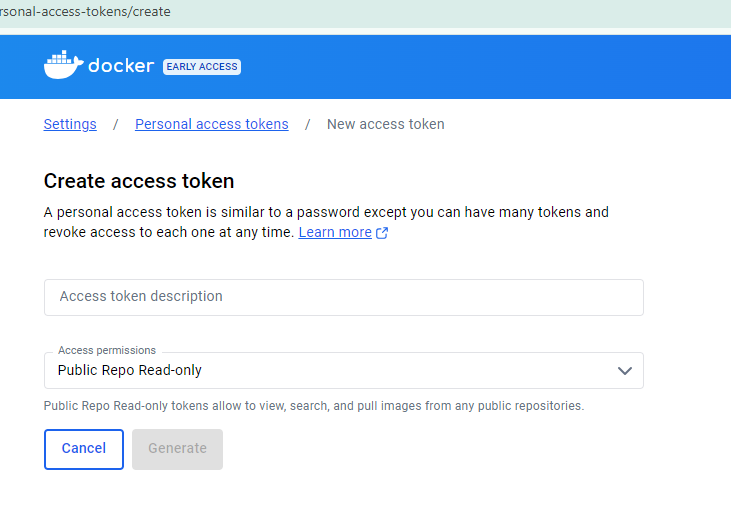
1. Go to account settings

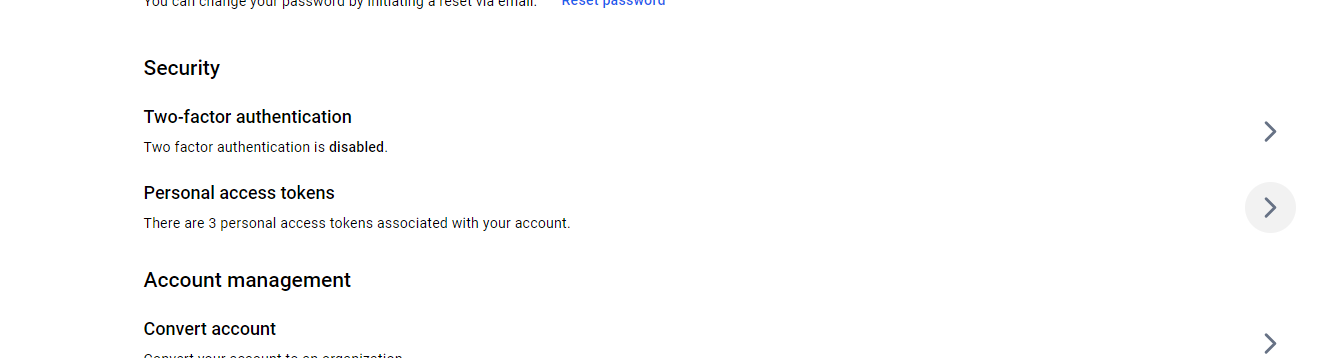


In the section of personal access tokens, click on generate new token



Add name for the token

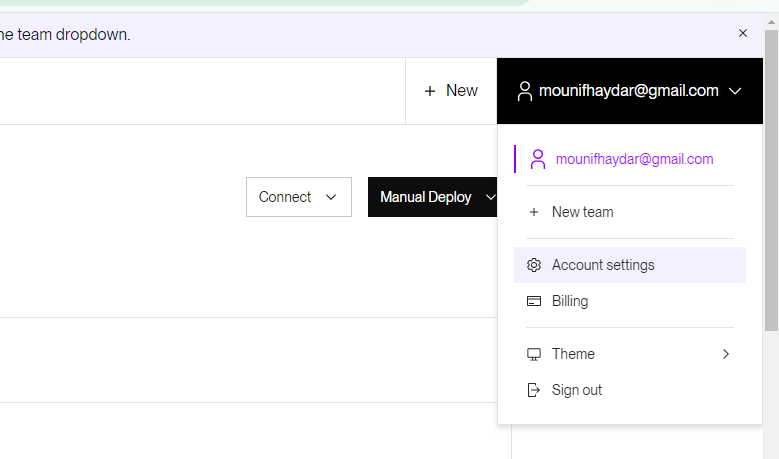




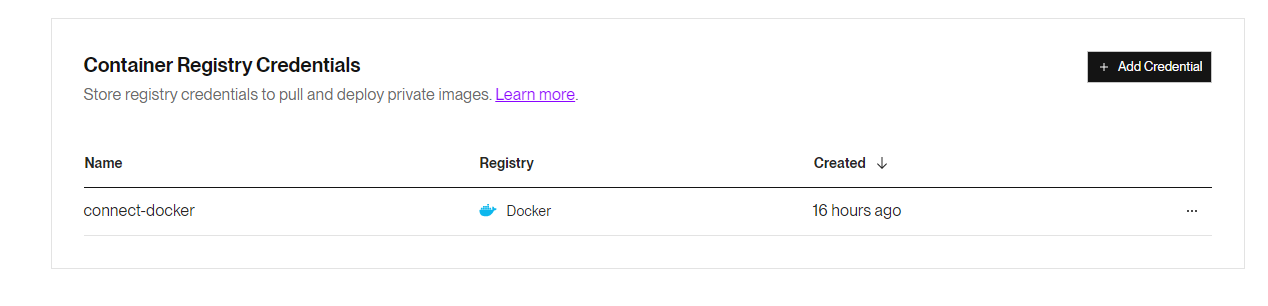
# Publish on render

Add credential to render

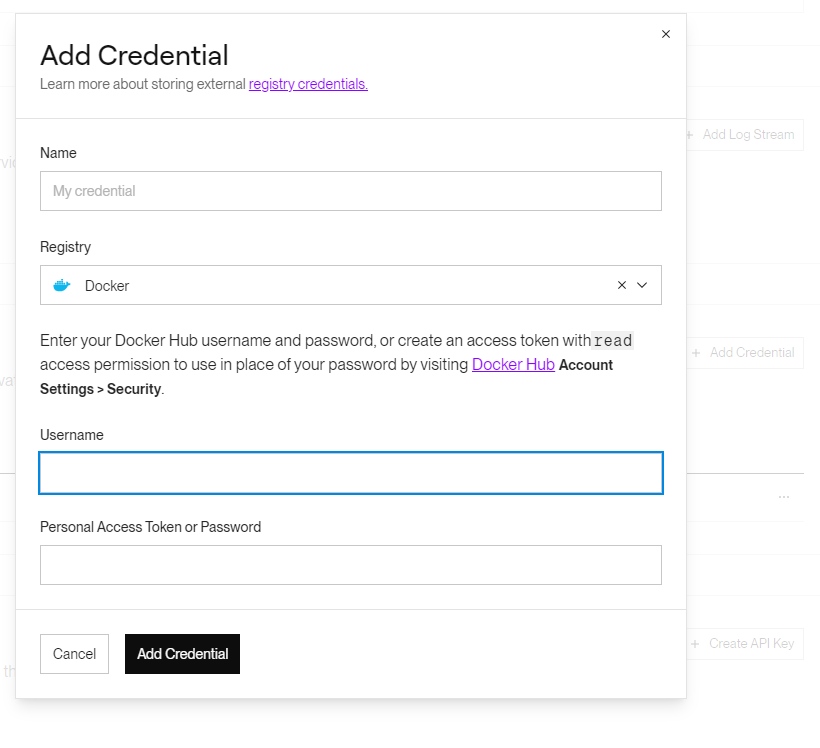
1. Account settings



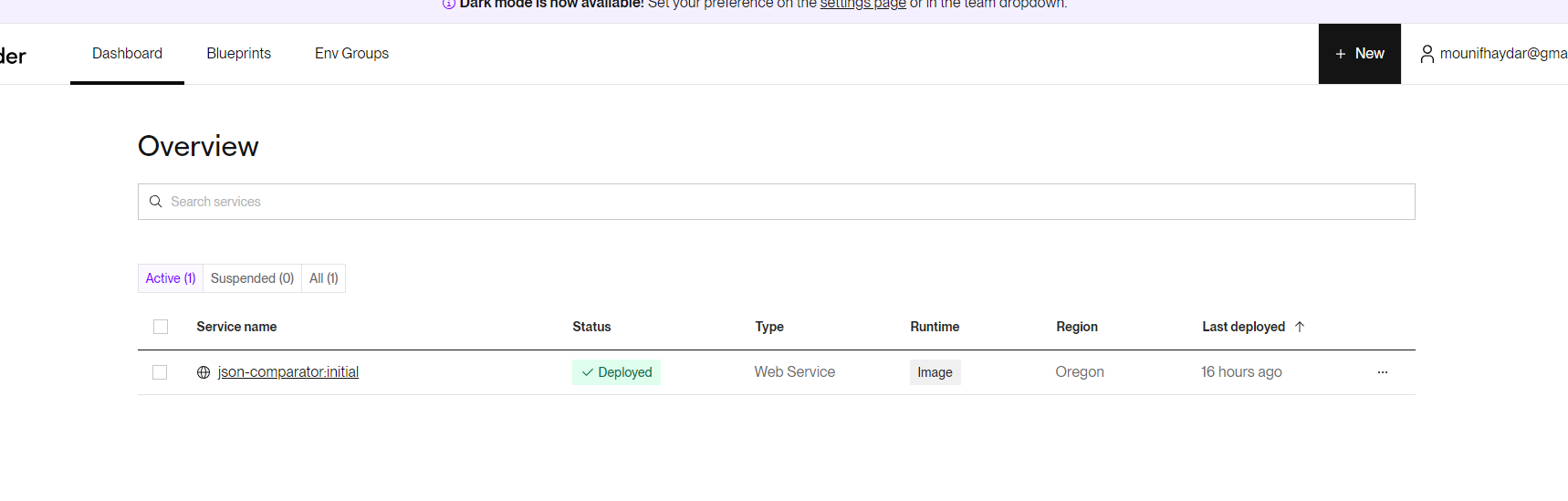
Click on the add credential button



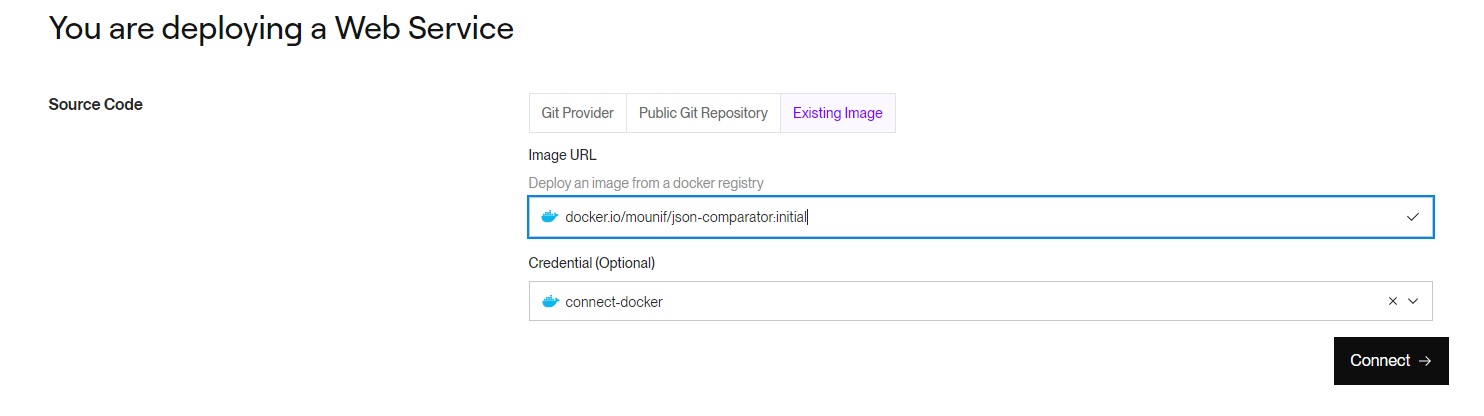
Add a name for the credential, put your docker username and past the Token that already created under the docker account



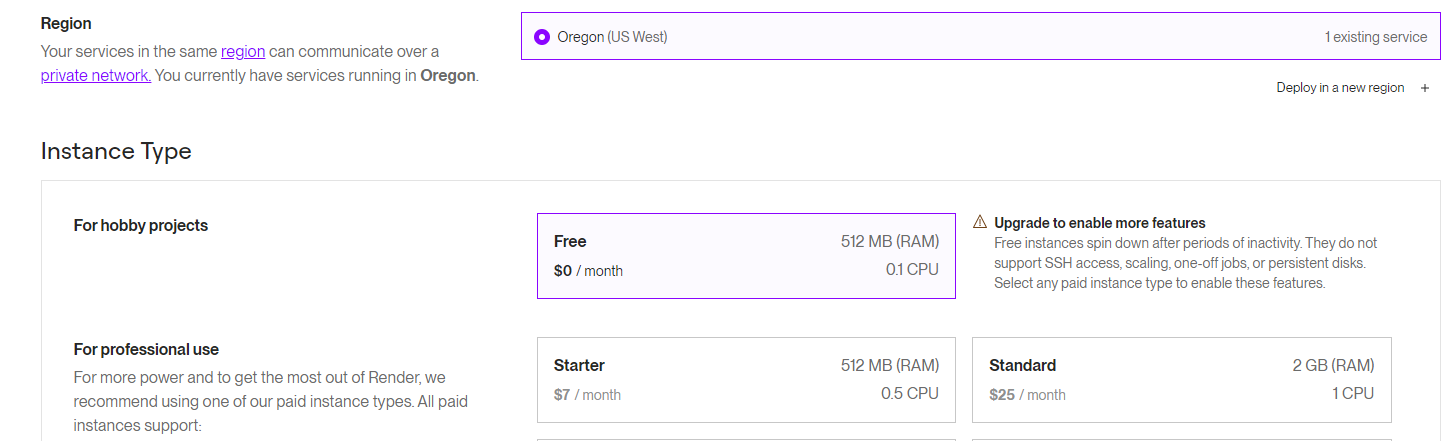
Creating a new webservice, under dashboard click on the button new



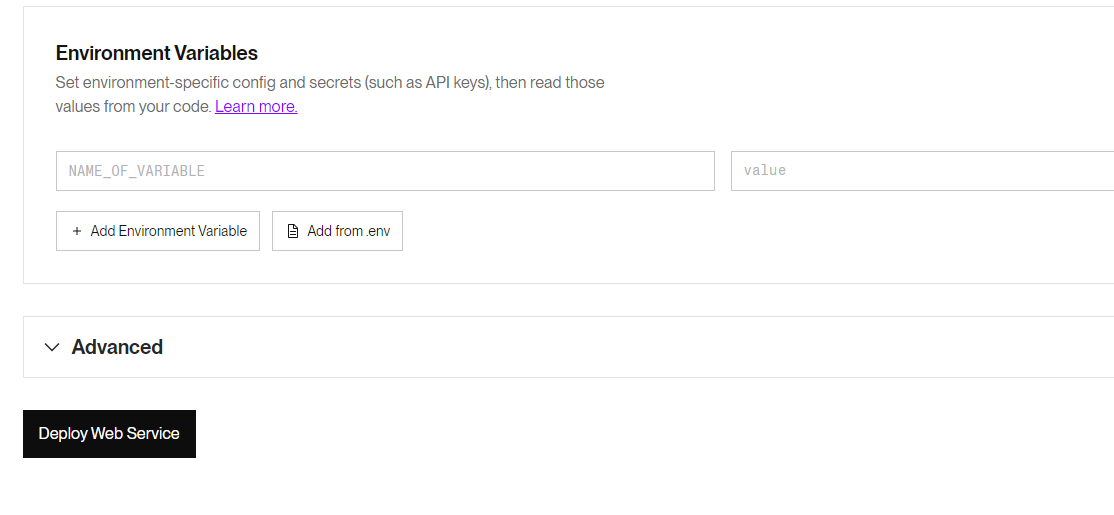
Set image URL and the created credential to allow render to publish



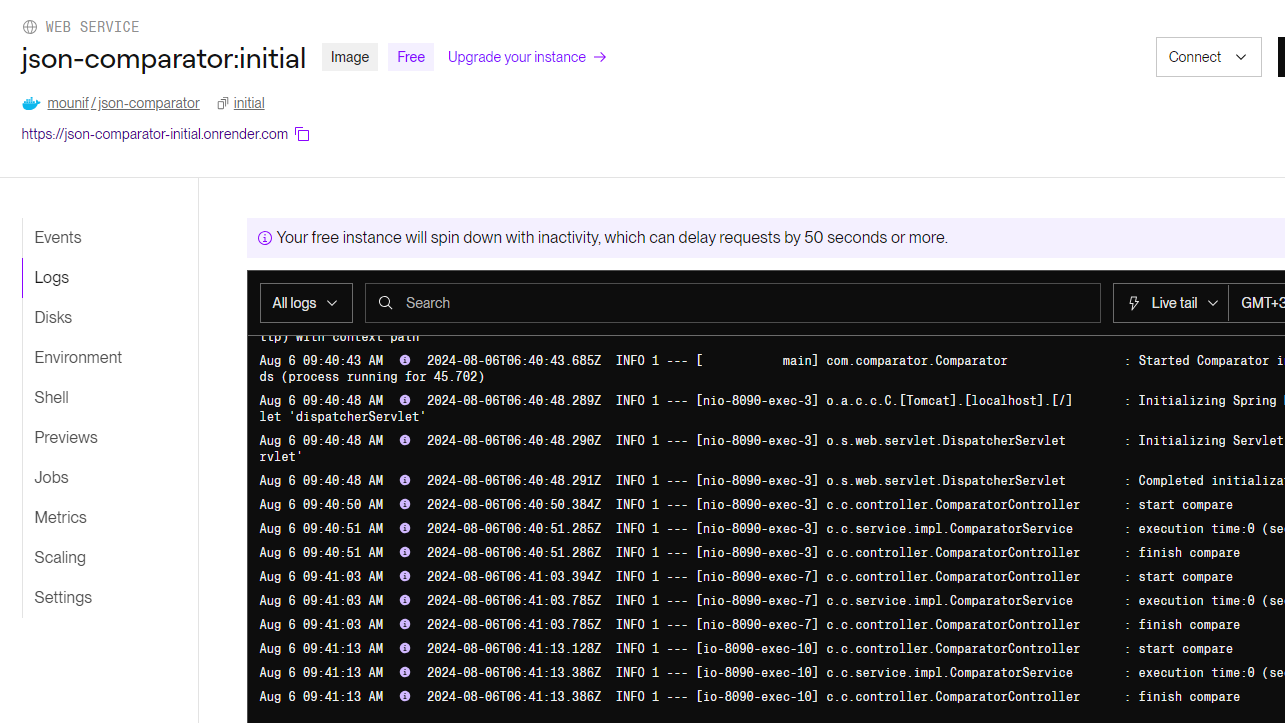
Choose the free plan



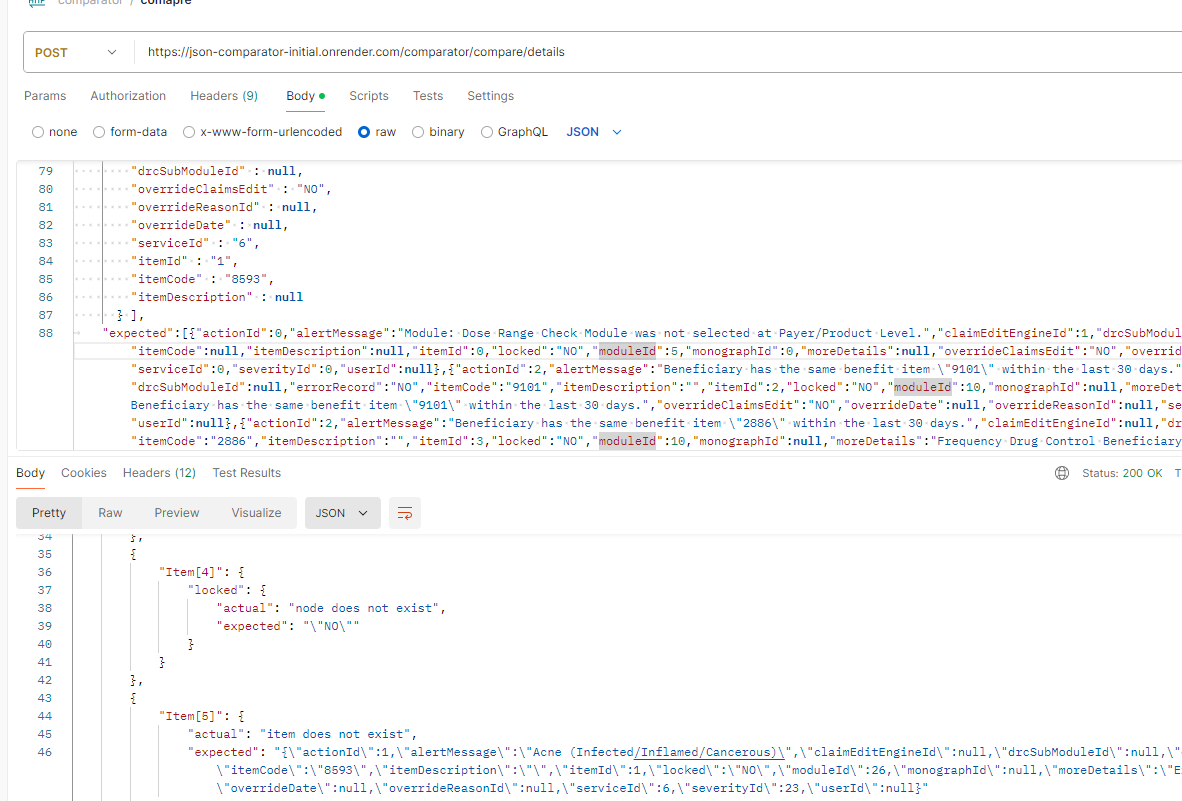
Finally click on deploy web service



The web service are running now under the following link <https://json-comparator-initial.onrender.com>



Test my API: <https://json-comparator-initial.onrender.com/comparator/compare/details>



Index

Reference : <https://hostingtutorials.dev/blog/free-spring-boot-host-with-render>