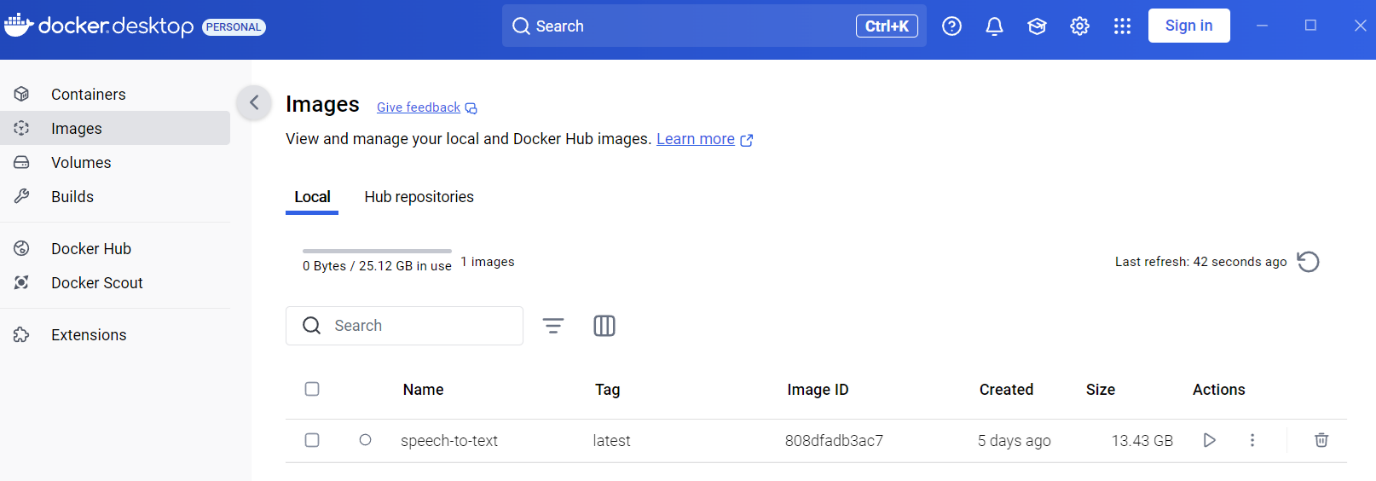
# Running the provided ASR example in Docker Desktop

1. **Open Docker Desktop**: and ensure it is running correctly.
   * Check the docker-installation.docx file for instructions
2. Pull (or download) the repo <https://github.com/Digitalia-Xamk/DigitaliaASR>
   * But probably you did this already since you are reading this document
3. **Navigate to the demo project folder**: Open a terminal (Command Prompt, PowerShell, or any terminal you prefer) and enter the folder.
   * **Important: Modify the** .env file to contain your own HuggingFace api key
4. **Build the Docker Image**: run the following in a terminal you just opened

docker build -t name-the-image .

*This command will build a Docker image named*name-the-image *using the Dockerfile in your project folder.*

* *-t gives a name to the image (change to whatever you like)*
* *. means the build will happen inside this same directory*
* *Note: The first build will take some time thus it installs many things into container and downloads AI models*
* *Next builds will be a lot faster unless specifically run with --no-cache option*

1. **Run the Docker Container**:
   * Once the image is built, go to the "Images" tab (Docker desktop left menu).
   * Click the "Run" button next to your image under Actions.
     + **Note!** If you intend to use GPU(s) you will need to launch via command line (terminal) and include the gpus tag.

* **Only do the following if you have NVIDIA GPU (code is tested with AMD GPUs and most likely won’t work)**

docker run --name name-the-container -d --gpus=all -p 8004:8004 name-the-image

* If you have many GPUs, you can define the one to use, the above command exposes all available GPUs to the container
* Also defined port needs to match the one exposed in Dockerfile
* After this you can jump to step 6
* In the "Run a new Container" dialog, open optional settings and alter the configurations:
  + - Kuva, joka sisältää kohteen teksti, kuvakaappaus, ohjelmisto, Tietokonekuvake

      Kuvaus luotu automaattisesti**Name**: Give your container a name (optional).
    - **Host port**: Map a host port to an exposed container port (dockerfile). Use what ever port you want. Right side example uses the same 8004 as the exposed port.
    - **Volumes** and **Environment Variables**: No need to fill
  + Click "Run".
  + You are automatically moved into containers tab and you should see the logs for the running container
    - After you see text “Device set to use cpu” or “Device set to use cuda” move forward

1. **Access Your Application**: Open a web browser and go to http://localhost:8004 (or the port number you defined) to access your application.
   * You can also click the links under Port(s)

Kuva, joka sisältää kohteen teksti, kuvakaappaus, ohjelmisto, Tietokonekuvake

Tekoälyn generoima sisältö voi olla virheellistä.

* + Starting the app takes some seconds, so if you are too fast you might see page not available info. Just refresh the page few times and you should be fine.

1. These same steps need to be repeated with each separate docker container, so in this case with the separate folders.
   * Ensure that you are using a different image / container name for each service
   * Ensure that the used port won’t overlap with already opened ports.