Cordial

Release 1.0

Jan 26, 2023

Contents

To use Cordial, you'll need an Amazon Web Services account and a computer with Docker. These instructions will walk you through setting both of these up and testing that Cordial is setup correctly.

We run Cordial in a Docker container to make setup easier. On Ubuntu, you'll be able to work inside of the Docker container with an IDE and other visualization tools, without setting them up on your computer. When you use the run script, your work in `catkin_ws` and `scratch` will be saved even if you close the container, because we are using Docker volumes. You can also pass files back and forth between the container and your computer while it's running by putting them in the `shared` directory in `cordial/docker`.

CHAPTER 1

Getting your Amazon Web Service credentials

For Cordial to speak we use Amazon Polly, which requires an Amazon Web Services account. At our current usage, using Amazon Polly is free up to a certain level), but you will need a credit card to create an account.

1. Create an Amazon Web Services account.
2. Once you sign in, in the top right of the page, click your account name (mine says "Audrow"), then in the drop-down menu click "My Security Credentials," then click "Create New Access Key."
3. Record your access key and keep it somewhere safe. You can do this by downloading this or just viewing it and copy-pasting it to somewhere for later reference.

Note: It is best practice to create separate accounts with less access than your root account and use those access keys, see Amazon's security best practices.

CHAPTER 2

Running Cordial in a Docker Container

1. Setup Docker:

   a. Install Docker:

   ```
   curl -fsSL https://get.docker.com -o get-docker.sh
   sh get-docker.sh
   ```

   b. Set Docker to run without `sudo`:

   ```
   sudo groupadd docker
   sudo gpasswd -a $USER docker
   newgrp docker
   ```

   c. Test that Docker is installed correctly and works without `sudo`:

   ```
   docker run hello-world
   ```

2. Setup Docker-compose:

   a. Install Docker-compose:

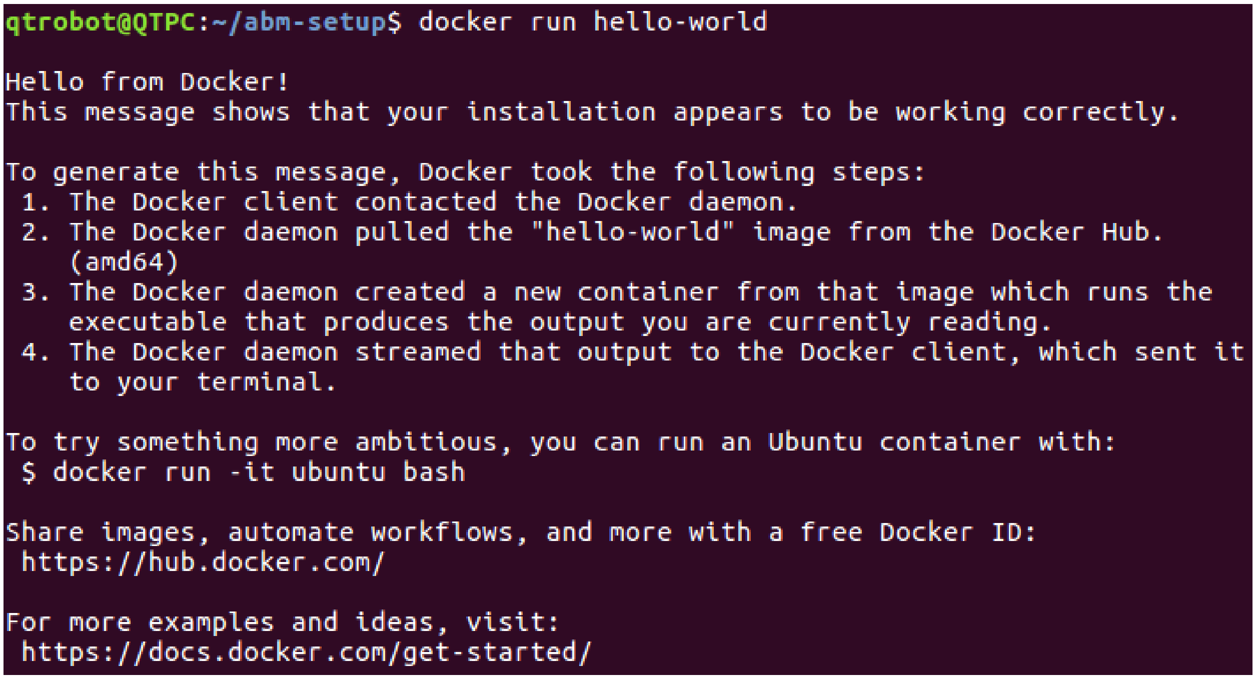
   ```
   sudo curl -L "https://github.com/docker/compose/releases/download/1.25.3/
   ↪docker-compose-$(uname -s)-$(uname -m)" -o /usr/local/bin/docker-compose
   sudo chmod +x /usr/local/bin/docker-compose
   ```

   b. Check that Docker-compose is installed correctly:

   ```
   docker-compose version
   ```

3. Run Cordial's Docker container. The first time it will take a while, say 10 minutes, depending on your computer and internet speed:

```
git clone https://github.com/robotpt/cordial
bash cordial/docker/run.sh
```



```
qtrobot@QTPC:~/abm-setup$ docker run hello-world

Hello from Docker!
This message shows that your installation appears to be working correctly.

To generate this message, Docker took the following steps:
 1. The Docker client contacted the Docker daemon.
 2. The Docker daemon pulled the "hello-world" image from the Docker Hub.
    (amd64)
 3. The Docker daemon created a new container from that image which runs the
    executable that produces the output you are currently reading.
 4. The Docker daemon streamed that output to the Docker client, which sent it
    to your terminal.

To try something more ambitious, you can run an Ubuntu container with:
 $ docker run -it ubuntu bash

Share images, automate workflows, and more with a free Docker ID:
 https://hub.docker.com/

For more examples and ideas, visit:
 https://docs.docker.com/get-started/
```

Fig. 1: What is printed from running the `hello-world` Docker container.

Note: You can pass arguments to the run script to change how the program is run. If you do `bash cordial/docker/run.sh terminal`, the Docker container will run in your current terminal. This is possibly the only way to run Cordial on a Windows or Mac computer without using a virtual machine.

4. Once in the Docker container, configure your AWS credentials by typing `aws configure` in the terminal that pops up. You will be prompted to enter the following:

   a. For `AWS Access Key ID` and `AWS Secret Access Key`, enter the corresponding credentials associated with your AWS account.

   b. For `Default region name`, enter the code the region code that you'd like to use. In Los Angeles, we typically use `us-west-1`. You can find a full list of region names here.

   c. For `Default output format`, enter `json`.

5. Test that everything is setup correctly by running an example in the Docker container, for example:

```
roslaunch cordial_manager example_say_and_ask_on_gui.launch
```

   You can then open a web-browser and go to `localhost:8080` for the face and `localhost:8081` for the GUI. Once these are open, an interaction should begin. You should see the face moving, see the text and some options on the GUI, and hear the voice speaking.

   Troubleshooting:

   - If nothing happens, check the terminal where you ran the above command. It is quite likely that something is wrong with the AWS credentials. Go back to the previous step.
   - If the face moves and GUI displays but you hear no voice, it's possible Docker can't find your sound card or that some other program is occupying your sound card. Try closing anything that might play sound (such as a Google Meet call) and retry. You might also see if you can play any sounds from

the Docker container, by running, for example, `beep`.