How to use the container file to create a reproduceable ubuntu worker environment on snowy

User Guide

VERY IMPORTANT FOR CONNECTING FROM OUTSIDE THE CAMPUS: DON'T TRY TO CONNECT TO THE SERVER FROM OUTSIDE THE CAMPUS BEFORE YOU ENABLE 2FA, FOLLOW STEP 2 IN CAMPUS BEFORE YOU TRY IT, OTHERWISE UPPMAX WILL LOCK YOU OUT PERMANENTLY AND YOU WILL NEED TO CONTACT SUPPORT FOR YOUR FIRST 2FA CODE!

1) You need a working username and password on uppmax that is already associated with a project on uppmax. We already have usernames and a project, and you can request a password for that here: https://suprintegration.uppmax.uu.se/getpasswd/

2) If you are connecting from outside the campus, you need a 2FA code, i.e a code that you get from an authenticator application like Google Authenticator(Google Authenticator - Apps on Google Play) or any other authenticator application. Install it on your phone and use these steps to enable it, 2-factor - Uppsala University (uu.se) .

3) When you have your username and password (and 2fa code if you are outside of campus) for uppmax, use this line to connect to the rackham login node, *ssh username@rackham.uppmax.uu.se*

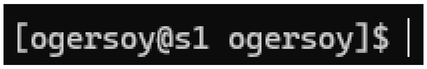
4) Now you are connected to a login node in rackham. Don't try to run any jobs on the login node.

5) To create a worker node that has the gpu compute in snowy, use this in the login node,

   interactive -A uppmax2024-2-18 -t 1:00:00 -M snowy

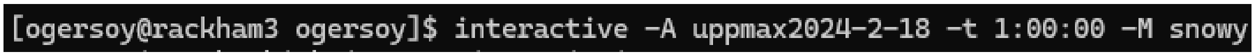
   This will create an interactive node for one hour on snowy, change -t modifier to change the time.

6) Check your node name to make sure you are on **Snowy**, You can understand that by looking at the username and server name that is before every line on ubuntu, for example snowy in my case says:
ogersoy@s1 ogersoy
s1 here means snowy. If it was a rackham worker node it would be rX i.e. r1.
Login nodes have the names rackhamX i.e. rackham1.



```
[ogersoy@s1 ogersoy]$ |
```

Image 1: This is a snowy worker node.



```
[ogersoy@rackham3 ogersoy]$ interactive -A uppmax2024-2-18 -t 1:00:00 -M snowy
```

Image 2: This is a rackham login node.

7) After you make sure you are on a snowy worker node, clone your github repo using the code below. In this example I am using the chatbot branch:

git clone -b centralized-chatbot https://github.com/ogersoy/Project-CS-2024

To update the branch from github for any reason after you clone it use fetch in the **project folder**,

git fetch origin
git pull origin centralized-chatbot

8) To install the requirements from a requirements.txt file, first cd into project folder like so,

cd Project-CS-2024

Then first use these 3 lines to install python, git and git-lfs

module load python3
module load git
module load git-lfs

Then use these steps to create a virtual environment
Install in a Virtual Environment (Recommended)
*A better practice is to avoid using sudo for Python package installations and instead use a **virtual environment** to install packages locally within your project directory.*

1. ***Create a virtual environment*** *in your project directory:*
bash
Copy code

```
python3 -m venv venv
```

2. ***Activate the virtual environment**:*
   bash
   Copy code
   ```
   source venv/bin/activate
   ```
3. ***Install the requirements** in the virtual environment:*
   bash
   Copy code
   ```
   pip3 install -r requirements.txt
   ```
   Since this is a local environment, you won't need sudo and won't run into permission issues.
4. ***Running your project**:*
   Ensure the virtual environment is activated every time you work on the project by running:
   bash
   Copy code
   ```
   source venv/bin/activate
   ```
   Then, run your script or use pip normally.

Check if pip is installed:

```
pip3 --version
```

If not use,

```
module load python3-pip
```

After you make sure pip is running use the command below to install all the requirements:

Use pip3 freeze to check if they are installed correctly.