

Task 0 – Software Installation (for Ubuntu)

NOTE: We recommend to use **Ubuntu OS ONLY** for this theme, since we have thoroughly tested the software installation on it at our end. If you still wish to use **Windows OS** and face any error in future, then you will have to resolve it by yourself.

The installation of all software/libraries has been tested on **64-bit, Ubuntu 16.04 and 18.04** machines. We recommend you to use any one of these versions of **Ubuntu OS**. These software/libraries have to be installed **only on 64-bit OS**.

This document contains instructions to install the following software/libraries on **Ubuntu OS**:

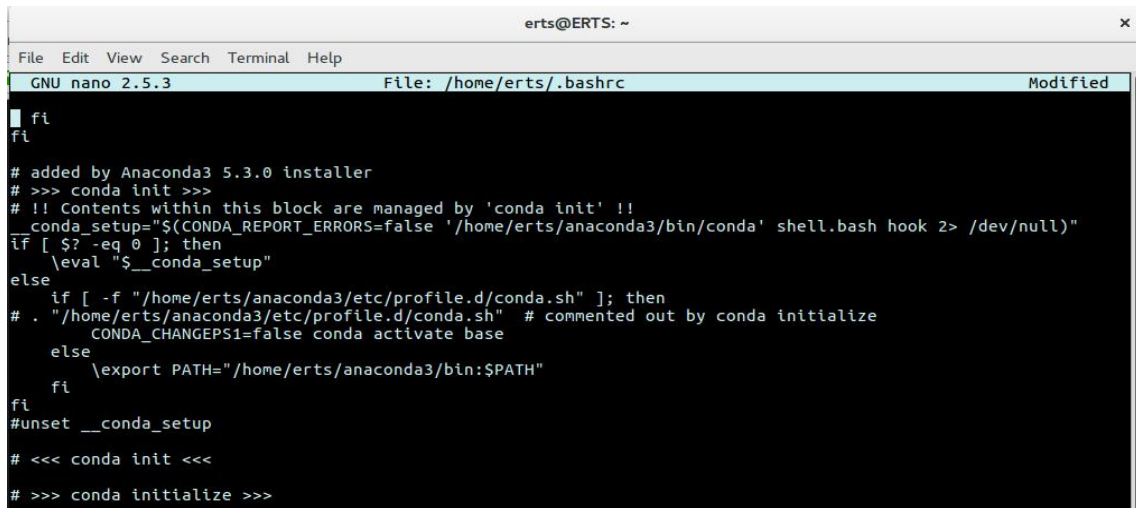
- Miniconda for Python 3
- NumPy
- OpenCV
- PySerial
- Visual Studio Code

1. Miniconda for Python 3:

- Conda is a package management system, created specifically to assist working with Python packages in a cross-platform fashion. Conda also has the concept of an environment, which is an independent, self-contained install of Python and packages.
- There are two options for getting Conda: Anaconda and Miniconda. [[reference](#)].
- We will be using **Python 3** for this theme. You don't have to install Python 3 separately. It comes by default with installation of Miniconda.

NOTE: If you already have Anaconda installed in your system, then kindly follow the below steps, else skip to the [Installation steps for Miniconda](#).

- Open Terminal and type: “**sudo nano ~/.bashrc**” and press **Enter**, enter the password when asked. Scroll down to the last to see the lines as ones shown in Figure 1. If you don't see these lines on your Terminal, it means Anaconda is not installed on your system. These lines were added by Anaconda installer at the time when you installed Anaconda.



```

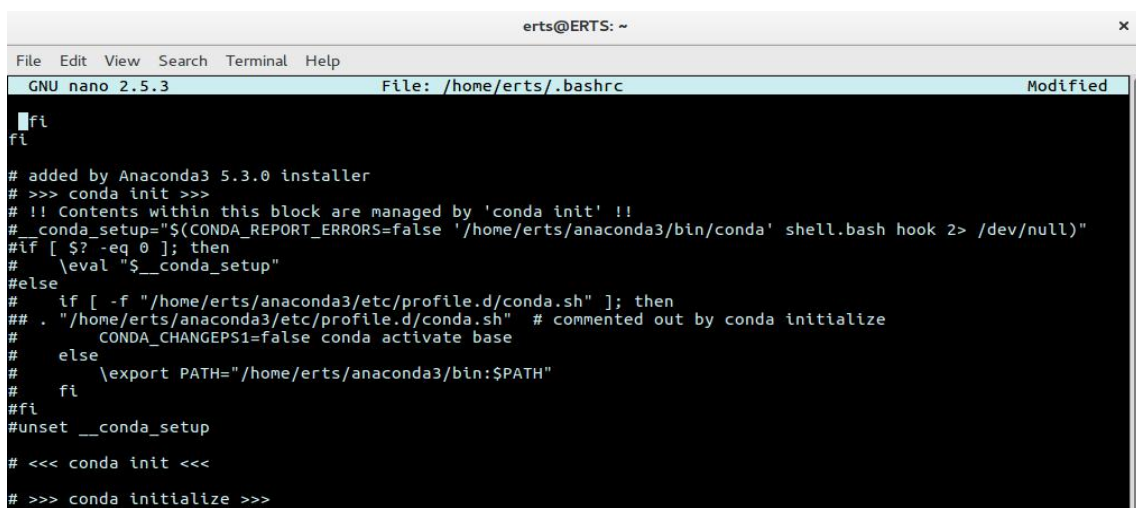
erts@ERTS: ~
File Edit View Search Terminal Help
GNU nano 2.5.3 File: /home/erts/.bashrc Modified
fi
fi

# added by Anaconda3 5.3.0 installer
# >>> conda init >>>
# !! Contents within this block are managed by 'conda init' !!
__conda_setup="$({CONDA_REPORT_ERRORS=false '/home/erts/anaconda3/bin/conda' shell.bash hook 2> /dev/null}"
if [ $? -eq 0 ]; then
    \eval "$__conda_setup"
else
    if [ -f "/home/erts/anaconda3/etc/profile.d/conda.sh" ]; then
        # "/home/erts/anaconda3/etc/profile.d/conda.sh" # commented out by conda initialize
        CONDA_CHANGEPS1=false conda activate base
    else
        \export PATH="/home/erts/anaconda3/bin:$PATH"
    fi
fi
#unset __conda_setup
# <<< conda init <<<
# >>> conda initialize >>>

```

Figure 1: Before adding comments

- Both Anaconda and Miniconda are associated by a common keyword “**conda**”. Hence, we need to disable Anaconda initialization. This will help in referring only to Miniconda when we use “**conda**” keyword next time. We will disable it by commenting certain lines using “**#**” at the start. After doing so, terminal will be as shown in Figure 2.



```

erts@ERTS: ~
File Edit View Search Terminal Help
GNU nano 2.5.3 File: /home/erts/.bashrc Modified
fi
fi

# added by Anaconda3 5.3.0 installer
# >>> conda init >>>
# !! Contents within this block are managed by 'conda init' !!
# __conda_setup="$({CONDA_REPORT_ERRORS=false '/home/erts/anaconda3/bin/conda' shell.bash hook 2> /dev/null}"
# if [ $? -eq 0 ]; then
#     \eval "$__conda_setup"
# else
#     if [ -f "/home/erts/anaconda3/etc/profile.d/conda.sh" ]; then
#         # "/home/erts/anaconda3/etc/profile.d/conda.sh" # commented out by conda initialize
#         CONDA_CHANGEPS1=false conda activate base
#     else
#         \export PATH="/home/erts/anaconda3/bin:$PATH"
#     fi
# fi
#unset __conda_setup
# <<< conda init <<<
# >>> conda initialize >>>

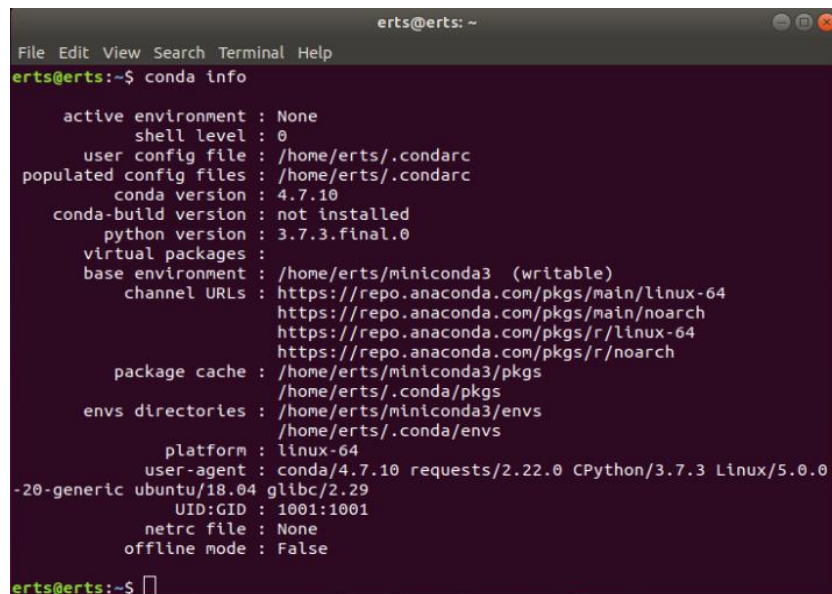
```

Figure 2: After commenting the lines using “**#**”

- Use the following shortcuts in sequence to save the changes and exit.
 - Save or Write-Out: **Ctrl + O** then **Enter**
 - Exit: **Ctrl + X**
- You will be back to the Terminal screen. For changes to take effect, type in Terminal: “**source ~/.bashrc**” and press **Enter**.

Installation Steps for Miniconda:

- Download **Miniconda for Python 3** for **64-bit** OS from [here](#).
- We need to make this downloaded file executable. Right-click on the folder where the downloaded file is present (for example, **Downloads** folder) and select **Open in Terminal** option.
- Type: “**sudo chmod u+x Miniconda3-latest-Linux-x86_64.sh**” and press **Enter**.
- Now type: “**./Miniconda3-latest-Linux-x86_64.sh**” and press **Enter**.
- To continue the installation process, press **Enter**. Accept the license terms by typing “**yes**”. Confirm the location of installation by pressing **Enter**. It will install the necessary libraries.
- Type “**yes**” when asked to initialize **Miniconda3** by running **conda init**.
- Once the installation completes, type in Terminal: “**source ~/.bashrc**” **OR** close the current Terminal and open the new one, for changes to take effect.
- Now type in Terminal: “**conda config --set auto_activate_base false**”, then type: “**source ~/.bashrc**”.
- Congrats, **Miniconda for Python 3** is successfully installed. Let’s verify now.
- In Terminal, type “**conda info**”. You will see the list of information related to the Miniconda, the output will be as shown in Figure 3.



```

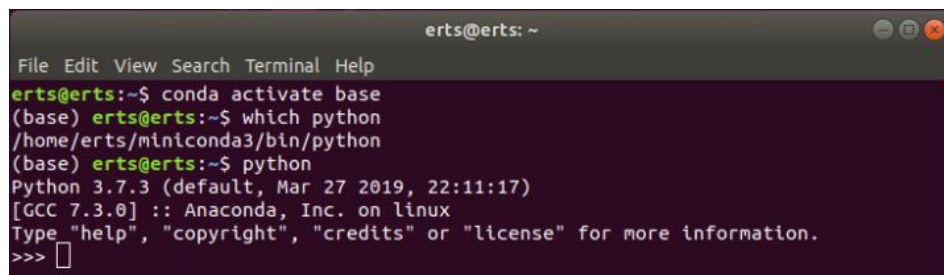
erts@erts: ~
File Edit View Search Terminal Help
erts@erts:~$ conda info

active environment : None
  shell level      : 0
  user config file : /home/erts/.condarc
populated config files : /home/erts/.condarc
conda version      : 4.7.10
conda-build version: not installed
python version     : 3.7.3.final.0
virtual packages   :
  base environment : /home/erts/miniconda3 (writable)
  channel URLs      : https://repo.anaconda.com/pkgs/main/linux-64
                    https://repo.anaconda.com/pkgs/main/noarch
                    https://repo.anaconda.com/pkgs/r/linux-64
                    https://repo.anaconda.com/pkgs/r/noarch
  package cache     : /home/erts/miniconda3/pkgs
                    /home/erts/.conda/pkgs
  envs directories  : /home/erts/miniconda3/envs
                    /home/erts/.conda/envs
  platform          : linux-64
  user-agent        : conda/4.7.10 requests/2.22.0 CPython/3.7.3 Linux/5.0.0-20-generic ubuntu/18.04 glibc/2.29
  UID:GID           : 1001:1001
  netrc file        : None
  offline mode      : False

erts@erts:~$
  
```

Figure 3: “conda info” output

- To activate the Python that comes with Miniconda installation, type in Terminal: “**conda activate base**”, then “**which python**” and then “**python**”. You will see output similar as shown in Figure 4.
- Whenever any environment is activated, its name appears in round braces (**env_name**) before. For example, on executing “**conda activate base**”, the (**base**) appears in the start of the line in Terminal.
- The **Python** version (**3.7.3**) might be different but that’s okay as long as it is **3.7.x**. Now type “**exit()**” to come out of Python console.

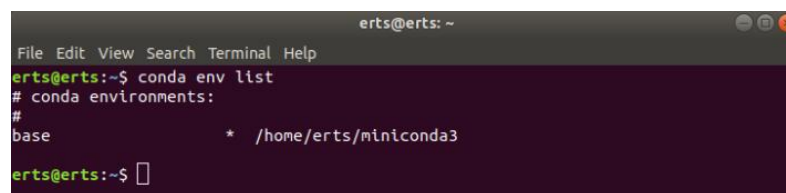


```

erts@erts: ~
File Edit View Search Terminal Help
erts@erts:~$ conda activate base
(base) erts@erts:~$ which python
/home/erts/miniconda3/bin/python
(base) erts@erts:~$ python
Python 3.7.3 (default, Mar 27 2019, 22:11:17)
[GCC 7.3.0] :: Anaconda, Inc. on linux
Type "help", "copyright", "credits" or "license" for more information.
>>>
  
```

Figure 4: Miniconda Python location and version output

- Deactivate the **base** environment by typing: “**conda deactivate**” OR “**source deactivate**”. Use this command to deactivate any active environment.
- Miniconda creates a default environment named **base** at the installation directory, let’s check this out. Type “**conda env list**” to get the output as shown in Figure 5. The * against the environment name indicates the current active environment.



```

erts@erts: ~
File Edit View Search Terminal Help
erts@erts:~$ conda env list
# conda environments:
#
base                * /home/erts/miniconda3
erts@erts:~$
  
```

Figure 5: “conda env list” output

2. NumPy, OpenCV, PySerial:

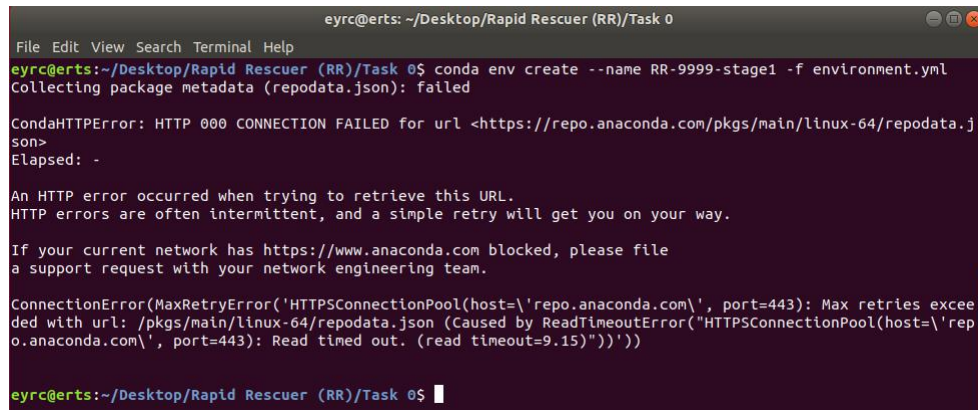
- The libraries necessary to complete the tasks in **Stage 1** are packaged into the “**environment.yml**” file located in the same folder where this document is present.
- Open new Terminal inside the folder where this file is present. Create an environment for Stage 1 with the name “**RR-<Team_ID>-stage1**” by typing the following command:

“**conda env create --name RR-<Team_ID>-stage1 -f environment.yml**”

- Replace **<Team_ID>** with your **Team ID**. If your **Team ID** is **9999**, type:

“conda env create --name RR-9999-stage1 -f environment.yml”

- Make sure that you have good Internet connection while running this command, else you will encounter an error shown in Figure 6.



```
eyrc@erts: ~/Desktop/Rapid Rescuer (RR)/Task 0
File Edit View Search Terminal Help
eyrc@erts:~/Desktop/Rapid Rescuer (RR)/Task 0$ conda env create --name RR-9999-stage1 -f environment.yml
Collecting package metadata (repodata.json): failed

CondaHTTPError: HTTP 000 CONNECTION FAILED for url <https://repo.anaconda.com/pkgs/main/linux-64/repodata.j
son>
Elapsed: -

An HTTP error occurred when trying to retrieve this URL.
HTTP errors are often intermittent, and a simple retry will get you on your way.

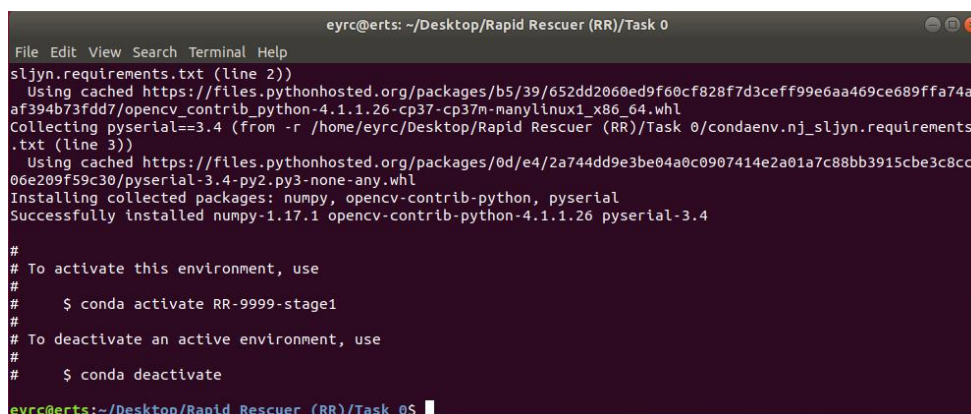
If your current network has https://www.anaconda.com blocked, please file
a support request with your network engineering team.

ConnectionError(MaxRetryError('HTTPSConnectionPool(host='repo.anaconda.com', port=443): Max retries exce
ded with url: /pkgs/main/linux-64/repodata.json (Caused by ReadTimeoutError('HTTPSConnectionPool(host='rep
o.anaconda.com', port=443): Read timed out. (read timeout=9.15)'))')

eyrc@erts:~/Desktop/Rapid Rescuer (RR)/Task 0$
```

Figure 6: Internet connection error encountered while creating the environment

- Once the environment is created, you will see the output on the Terminal as shown in Figure 7.



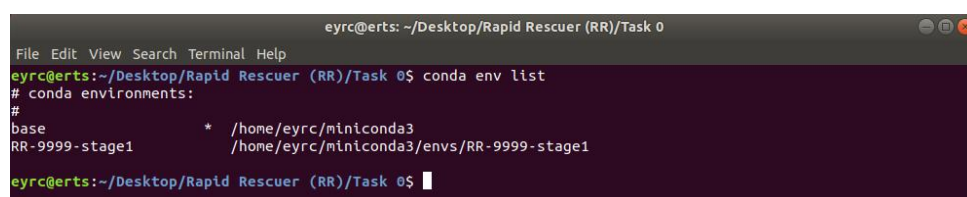
```
eyrc@erts: ~/Desktop/Rapid Rescuer (RR)/Task 0
File Edit View Search Terminal Help
sljyn.requirements.txt (line 2))
Using cached https://files.pythonhosted.org/packages/b5/39/652dd2060ed9f60cf828f7d3ceff99e6aa469ce689ffa74a
af394b73fdd7/opencv_contrib_python-4.1.1.26-cp37m-manylinux1_x86_64.whl
Collecting pyserial==3.4 (from -r /home/eyrc/Desktop/Rapid Rescuer (RR)/Task 0/condaenv.nj_sljyn.requirements
.txt (line 3))
Using cached https://files.pythonhosted.org/packages/0d/e4/2a744dd9e3be04a0c0907414e2a01a7c88bb3915cbe3c8cc
06e209f59c30/pyserial-3.4-py2.py3-none-any.whl
Installing collected packages: numpy, opencv-contrib-python, pyserial
Successfully installed numpy-1.17.1 opencv-contrib-python-4.1.1.26 pyserial-3.4

#
# To activate this environment, use
#
#   $ conda activate RR-9999-stage1
#
# To deactivate an active environment, use
#
#   $ conda deactivate

eyrc@erts:~/Desktop/Rapid Rescuer (RR)/Task 0$
```

Figure 7: “conda env create --name RR-9999-stage1 -f environment.yml” output

- If you now type: “conda env list”, you will see the newly created environment **RR-9999-stage1** as shown in Figure 8. Your Conda environment is successfully created.



```
eyrc@erts: ~/Desktop/Rapid Rescuer (RR)/Task 0
File Edit View Search Terminal Help
eyrc@erts:~/Desktop/Rapid Rescuer (RR)/Task 0$ conda env list
# conda environments:
#
base                  *  /home/eyrc/miniconda3
RR-9999-stage1        /home/eyrc/miniconda3/envs/RR-9999-stage1

eyrc@erts:~/Desktop/Rapid Rescuer (RR)/Task 0$
```

Figure 8: “conda env list” output after environment creation

3. Visual Studio Code:

- Download the portable version **ONLY** of VS Code from this [link](#). The downloaded file name will be: **code-stable-1571154019.tar.gz**.
- We need to extract this file in the folder its downloaded (for example, **Downloads** folder). Open the Terminal in this folder and type this command to extract this file: **“tar -xvf code-stable-1571154019.tar.gz”**.
- You have installed VS Code successfully.

Congrats, you have installed all the required software necessary for Stage 1 !!

In order to verify the installation of above software/libraries, follow the steps given in [Test_Setup_Read_Me.pdf](#) document in 2. *Test Setup* folder.