VirtualBox

1. Install VirtualBox on Windows or Mac

- 20 GB minimum storage for the VM
- 2 GB RAM allocation
- Open VirtualBox as admin

2. Install Ubuntu 18.04 on VirtualBox (the GUI version)

- Download site: https://www.ubuntu.com/download/desktop

NOTES: Created default install of Ubuntu 18.0.4 - Students will connect directly to Ubuntu and not to a Windows desktop with Ubuntu in VirtualBox. Allocated 25GB storage, 2GB RAM and 2 CPU to installation.

Added updates and dependencies for lab connectivity:
apt upgrade
apt install libglade2-0 openssh-server python-dbus python-gobject open-vm-tools-desktop gnome-session-flashback lightdm

Installing Anaconda for Python 3.6 on Ubuntu

Open Terminal

1. Go to bottom left corner of your screen, click on "Show Applications" button (:::)
2. Type "Terminal" in search bar to find it
3. You can right click on the icon to Add to Favorites, it will pin it to task bar on the left of your screen

Navigate to your home directory:

```
cd
```

Install `curl`:

```
sudo apt-get install curl
```

You will be prompted a password for your user, type it and hit `ENTER`

Download the installer script:

```
curl -O https://repo.anaconda.com/archive/Anaconda3-5.2.0-Linux-x86_64.sh
```

Check sum:

```
sha256sum Anaconda3-5.2.0-Linux-x86_64.sh
```

You should see something like this in your console:

```
09f53738b0cd3bb96f5b1bac488e5528df9906be2480fe61df40e0e0d19e3d48  Anaconda3-5.2.0-Linux-x86_
```

Run the script:

```
bash Anaconda3-5.0.1-Linux-x86_64.sh
```

NOTE: error "no such file or directory" replaced with bash Anaconda3-5.2.0-Linux-x86_64.sh

You should see something like that in your console:

```
Welcome to Anaconda3 5.2.0 (by Continuum Analytics, Inc.)

In order to continue the installation process, please review the license
agreement.
Please, press ENTER to continue
```

Press `ENTER` to continue and then press `ENTER` to read through the license. At the very end you'll get a prompt to accept or decline the terms:

```
Do you approve the license terms? [yes|no]
```

You will then be prompted to confirm the location of the `anaconda3`:

```
 Anaconda3 will now be installed into this location:
 /home/[user-name]/anaconda3

   - Press ENTER to confirm the location
   - Press CTRL-C to abort the installation
   - Or specify a different location below

[/home/[user-name]/anaconda3] >>>
```

It will take a few minutes, just be patient ...

When done, you will be prompted to add your `anaconda3` directory to `PATH`

```
...
installation finished.
Do you wish the installer to prepend the Anaconda3 install location
to PATH in your /home/[user-name]/.bashrc ? [yes|no]
[no] >>>
```

Type `yes` and hit `ENTER`. You should see this in your console:

```
Prepending PATH=/home/[user-name]/anaconda3/bin to PATH in /home/[user-name]/.bashrc
A backup will be made to: /home/[user-name]/.bashrc-anaconda3.bak
```

NOTE: Option to proceed with installation of Microsoft VSCode after above. Selected NO.

To verify the installation, you can list packages installed with `anaconda3` by running

```
conda list
```

NOTE: Conda list did not work - reboot of system was required.

The output will have a long list of install libraries, their viersion, build and channel:

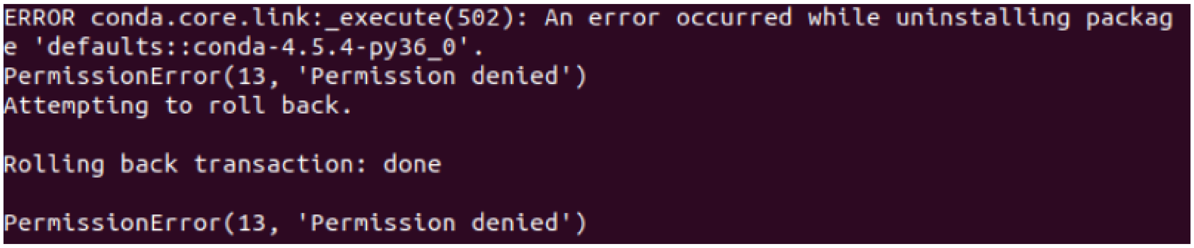
```
# packages in environment at /home/katya/anaconda3:

         #
         # Name                      Version                   Build  Channel
         _ipyw_jlab_nb_ext_conf    0.1.0             py36he11e457_0
         absl-py                   0.2.2                      <pip>
         alabaster                 0.7.10            py36h306e16b_0
         anaconda                  5.2.0                    py36_3
         anaconda-client           1.6.14                   py36_0
```

To install and run a GUI explorer for Anaconda Anaconda Navigator run these two commands

```
conda install -c anaconda anaconda-navigator
```

NOTE Error:



```
ERROR conda.core.link:_execute(502): An error occurred while uninstalling packag
e 'defaults::conda-4.5.4-py36_0'.
PermissionError(13, 'Permission denied')
Attempting to roll back.

Rolling back transaction: done

PermissionError(13, 'Permission denied')
```

```
anaconda-navigator
```

Install nltk and tensorflow with conda

`conda install nltk` **NOTE SAME ERROR AS ABOVE**

`conda install -c conda-forge tensorflow` **NOTE SAME ERROR AS ABOVE**

Installing pip and Rasa.ai stack

Install pip using conda

`conda install pip` **NOTE SAME ERROR AS ABOVE**

Install spaCy

`pip install -U spacy` **NOTE: Failed building wheel errors**

Install Rasa Core

```
pip install rasa_core
```

Install Rasa NLU

```
pip install rasa_nlu
```

Download English language support for spacy

```
python -m spacy download en
```

Replace standard small corpus with medium one

```
python -m spacy download en_core_web_md
python -m spacy link en_core_web_md en --force
```

Install `graphviz`

```
sudo apt-get install graphviz
```

Check version of installed graphviz

```
dot -V
```

Alias `dot` to `graphviz`

```
alias graphviz='dot'
```

Install other dependencies

```
sudo apt-get install python-dev libgraphviz-dev pkg-config
```

Install `pygraphviz`

```
pip install pygraphviz
```

Run rasa visualization function and create a graph

```
python -m rasa_core.visualize -d domain.yml -s data/stories.md -o graph.png
```

Rasa UI

Update your system

Switch to your `home` directory

```
cd
```

Update your system's packages

```
sudo apt-get update
```

Install and configure PostgreSQL

Install `postgresql` and `postgresql-contrib`

```
sudo apt-get install postgresql postgresql-contrib
```

Switch to `postgres` user

```
sudo -i -u postgres
```

Download schema for Rasa NLU

```
wget https://raw.githubusercontent.com/paschmann/rasa-ui/master/resources/dbcreate.sql
```

Initiate postgres process

```
psql
```

You should see the following in your terminal

```
postgres=#
```

Create `rasaui` database by executing the following command

```
create database rasaui;
```

Change password of your `postgres` user to `postgres` *(your real password should be something a bit more robust, this is just for demonstration purposes)*

```
ALTER USER postgres PASSWORD 'postgres';
```

Switch to `rasaui` database by executing the following command

```
\c rasaui
```

Add a schema to your database

```
\i dbcreate.sql
```

Quit the database

```
\q
```

Exit the `postgres` user session

```
exit
```

Install Node.js and npm

Install `curl` and `python-software-properties`

```
sudo apt-get install curl python-software-properties
```

Get and add `nodejs` version 6 (a.k.a. Boron) to PPA - Ubuntu package manager

(if you want to install a more recent version, opt for an even number like 8 or 10, because they are LTS versions of `nodejs`)

```
curl -sL https://deb.nodesource.com/setup_6.x | sudo -E bash -
```

Install `nodejs` using the package manager now

```
sudo apt-get install nodejs
```

Check the version

```
nodejs -v
```

Check if `npm` is installed

```
npm -v
```

If it's not, execute the following command

```
sudo apt-get install npm
```

Now check `npm` version again, it should be all set!

```
npm -v
```

Rasa UI clone & install through npm

Clone Rasa UI from its repository

```
git clone https://github.com/paschmann/rasaui.git
```

Navigate inside of `rasaui` directory

```
cd rasaui
```

Install `node` packages by executing the following

```
npm install
```

`package.json` configuration

- Open `package.json` file, it's located inside of your `rasaui` directory
- Line `47` with `postgresserver` connection string contains information about how Rasa UI should connect to your database
- It is usually in this format

```
"postgres://DB_user_name:DB_password@localhost:5432/DB_name"
```

Testing Servers (`chatbot` repository needs to be cloned to Desktop for this to work!)

- Rasa NLU can be used as a classic HTTP server
- You can find more information and documentation on https://nlu.rasa.com/http.html
- Rasa Core can also be used as a classic HTTP server
- You can find more information and documentation on https://core.rasa.com/http.html

Rasa NLU: start HTTP server

In your terminal, navigate to a folder `rasamodels/test` inside of your `chatbot` directory

```
cd ~/Desktop/chatbot/rasamodels/test
```

Check the contents of the folder

```
ls -al
```

You should have the following in it

```
total 20
drwxr-xr-x 4 user user 4096 Jul 19 14:33 .
drwxr-xr-x 4 user user 4096 Jul 19 14:33 ..
drwxr-xr-x 2 user user 4096 Jul 17 12:04 data
drwxr-xr-x 4 user user 4096 Jul 17 12:08 models
-rw-r--r-- 1 user user  121 Jul 17 10:06 nlu_config.yml
```

To start your server execute the following command

```
python -m rasa_nlu.server -c nlu_config.yml --path models/nlu
```

- Where `-c` parameter points to the NLU configuration file `nlu_config.yml` that is located in `test` directory, and
- `--path` parameter points to the directory `models/nlu` where our NLU models are saved and/or will be saved after we re-train our models through Rasa UI

If your server was successfully started, then you should see the following if you paste `http://localhost:5000` into your browser

A message like this should also appear in your console where the server is currently running

```
2018-07-19 15:15:54-0400 [-] "127.0.0.1" - - [19/Jul/2018:19:15:54 +0000]
"GET /favicon.ico HTTP/1.1" 404 233 "http://localhost:5000/"
"Mozilla/5.0 (X11; Linux x86_64)
AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/67.0.3396.99 Safari/537.36"
```

Rasa Core: start HTTP server

Open a new terminal, navigate to a folder `rasamodels/test` inside of your `chatbot` directory

```
cd ~/Desktop/chatbot/rasamodels/test
```

From your `chatbot/rasamodels/test` directory execute the following command to start your server

```
python -m rasa_core.server -d models/dialogue/test -o rasa-core.log
```

- Where `-d` parameter points to the directory where a trained dialogue model lives
- `-o` parameter points to the file where the server is going to save the log of requests (optional)

If your server was successfully started, then you should see the following if you paste `http://localhost:5005` into your browser

A message like this should also appear in your console where the server is currently running

```
127.0.0.1 - - [2018-07-19 15:55:28] "GET / HTTP/1.1" 200 143 0.001693
127.0.0.1 - - [2018-07-19 15:55:28] "GET /favicon.ico HTTP/1.1" 404 374 0.010400
```

Rasa UI: start application

Open a new terminal, navigate to a folder `rasaui` inside of your `home` directory

```
cd ~/rasaui
```

From your `rasaui` directory execute the following command to start Rasa UI

```
npm start
```

Your console should populate with output of similar kind if your connection was successful.

```
> RasaUI@0.2.0 start /home/[user]/rasaui
> node server/server.js

Rasa UI Server: http://localhost:5001

Express server listening on port 5001

Rasa NLU Connected
Using connection string from: package.json
Rasa NLU Server: http://localhost:5000

Rasa Core Connected
Using connection string from: package.json
Rasa Core Server: http://localhost:5005

Postgres DB Connected
Using connection string from: package.json
Postgres Server: 127.0.0.1:5432
Database:rasaui
Schema:public
```

Notice that we have 4 servers running simultaneously:

1. Rasa UI Server: http://localhost:5001
2. Rasa NLU Server: http://localhost:5000
3. Rasa Core Server: http://localhost:5005
4. Postgres Server: 127.0.0.1:5432

All of these servers run on your local machine, therefore the URLs that you see either include `localhost` or `127.0.0.1`.

If you were to run remote server(s), you would replace the URLs with the IP address or domain name of your remote server(s) and the appropriate ports if you are using alternative ports in `rasaui/package.json` file (lines 40-48 that define the configuration of the servers:

```
"config": {
  "rasanluendpoint": "http://NLU_server_domain:NLU_server_port",
  "rasacoreendpoint": "http://CORE_server_domain:CORE_server_port",
  "coresecuritytoken": "",
  "nlusecuritytoken": "",
  "cacheagents": false,
  "jwtsecret": "your_JWT_secret",
  "postgresserver": "postgres://DB_username:DB_password@DB_server_domain:DB_server_port/DB_n
}
```