



INSTALLATION GUIDE

TRIP @ US



A0213523U	MEHTA VIDISH PRANAV
A0213496B	ANKEIT TAKSH
A0198539H	WANG KUAN-KAI
A0213514U	ANANDAN NATARAJAN



INSTALLAZTION GUIDE

ISS-VM INSTALLATION

STEP 1: INSTALL ISS VM

This application runs on Ubuntu and hence it is highly recommended to install ISS-VM and virtual box in your PC.

Please follow the instruction at <https://github.com/telescopeuser/iss-vm> to install NUS ISS-VM

STEP 2: GET PROJECT REPOSITORY

Open your terminal in NUS ISS-VM and go to desktop as the current directory. You may do so by following this command:

```
iss-user@iss-vm: ~/Desktop
(base) iss-user@iss-vm:~$ cd Desktop
(base) iss-user@iss-vm:~/Desktop$
```

Make a new directory called Trip@Us Test

```
iss-user@iss-vm: ~/Desktop
(base) iss-user@iss-vm:~/Desktop$ mkdir Trip@UsTest
(base) iss-user@iss-vm:~/Desktop$
```

Cd into the folder

```
iss-user@iss-vm: ~/Desktop/Trip@UsTest
(base) iss-user@iss-vm:~/Desktop/Trip@UsTest$ cd ..
(base) iss-user@iss-vm:~/Desktop$ cd Trip@UsTest
(base) iss-user@iss-vm:~/Desktop/Trip@UsTest$
```



Git clone from project directory: <https://github.com/vid1994/TripAtUs>

```
iss-user@iss-vm: ~/Desktop/Trip@UsTest
(base) iss-user@iss-vm:~/Desktop/Trip@UsTest$ git clone https://github.com/vid1994/TripAtUs
Cloning into 'TripAtUs'...
remote: Enumerating objects: 890, done.
remote: Counting objects: 100% (890/890), done.
remote: Compressing objects: 100% (604/604), done.
remote: Total 890 (delta 399), reused 763 (delta 273), pack-reused 0
Receiving objects: 100% (890/890), 24.56 MiB | 3.84 MiB/s, done.
Resolving deltas: 100% (399/399), done.
Checking connectivity... done.
(base) iss-user@iss-vm:~/Desktop/Trip@UsTest$
```

STEP 3: CREATE CONDA ENVIRONMENT

Conda is already installed in NUS ISS VM. Please create a new virtual environment following below commands. This is done so as to isolate our working environment from our base environment.

```
iss-user@iss-vm: ~/Desktop/Trip@UsTest
(base) iss-user@iss-vm:~/Desktop/Trip@UsTest$ conda -V
conda 4.7.12
(base) iss-user@iss-vm:~/Desktop/Trip@UsTest$ conda create -n TripAtUsTest python=3.6 anaconda
Collecting package metadata (repodata.json): /
```

Once installed, you will be able to see the transactions as done.

```
iss-user@iss-vm: ~/Desktop/Trip@UsTest
yaml          pkgs/main/linux-64::yaml-0.1.7-had09818_2
yapf          pkgs/main/noarch::yapf-0.28.0-py_0
zeromq        pkgs/main/linux-64::zeromq-4.3.1-he6710b0_3
zict          pkgs/main/noarch::zict-1.0.0-py_0
zipp          pkgs/main/noarch::zipp-2.2.0-py_0
zlib          pkgs/main/linux-64::zlib-1.2.11-h7b6447c_3
zstd          pkgs/main/linux-64::zstd-1.3.7-h0b5b093_0

Proceed ([y]/n)? y
Preparing transaction: done
Verifying transaction: done
Executing transaction: done
#
# To activate this environment, use
#
#     $ conda activate TripAtUsTest
#
# To deactivate an active environment, use
#
#     $ conda deactivate
#
(base) iss-user@iss-vm:~/Desktop/Trip@UsTest$
```




Next, activate your conda environment.

```
iss-user@iss-vm: ~/Desktop/Trip@UsTest/TripAtUs
(base) iss-user@iss-vm:~/Desktop/Trip@UsTest/TripAtUs$ conda activate TripAtUsTest
(TripAtUsTest) iss-user@iss-vm:~/Desktop/Trip@UsTest/TripAtUs$
```

Now pip install all packages from requirements.txt file

```
iss-user@iss-vm: ~/Desktop/Trip@UsTest/TripAtUs
(TripAtUsTest) iss-user@iss-vm:~/Desktop/Trip@UsTest/TripAtUs$ pip install -r requirements.txt
Requirement already satisfied: actionlib==1.11.13 in /opt/ros/kinetic/lib/python2.7/dist-packages (from -r requirements.txt (line 1)) (1.11.13)
Requirement already satisfied: alabaster==0.7.12 in /home/iss-user/anaconda3/envs/TripAtUsTest/lib/python3.6/site-packages (from -r requirements.txt (line 2)) (0.7.12)
Collecting amqp==2.5.2
  Using cached amqp-2.5.2-py2.py3-none-any.whl (49 kB)
Requirement already satisfied: anaconda-client==1.7.2 in /home/iss-user/anaconda3/envs/TripAtUsTest/lib/python3.
```

As gdal is best run through conda, please run conda install gdal

```
iss-user@iss-vm: ~/Desktop/Trip@UsTest/TripAtUs
(TripAtUsTest) iss-user@iss-vm:~/Desktop/Trip@UsTest/TripAtUs$ conda install gdal
Collecting package metadata (repodata.json): |
```

Great, now your environment is setup.

STEP 4: CHECK RABBITMQ

Rabbit mq is the message brokerage service that queues tasks for our celery workers. We have already made it created the host and you can run the below command to confirm that it is working.

```
iss-user@iss-vm: ~/Desktop/Trip@UsTest/TripAtUs
(TripAtUsTest) iss-user@iss-vm:~/Desktop/Trip@UsTest/TripAtUs$ sudo rabbitmqctl status
[sudo] password for iss-user:
Status of node 'rabbit@iss-vm' ...
[[pid,16510],
 {running_applications,[{rabbit,"RabbitMQ","3.5.7"},
                        {mnesia,"MNESIA CXC 138 12","4.13.3"},
                        {xmerl,"XML parser","1.3.10"},
                        {os_mon,"CPO CXC 138 46","2.4"}],
```

If you would like to create a separate host, please follow the instructions on this page <https://docs.celeryproject.org/en/stable/getting-started/brokers/rabbitmq.html>

Please ensure that you configure the host in the settings.py file in Django.



STEP 5: RUN CELERY

Celery are the server workers that will run the task asynchronously.

Firstly cd into main directory. Please do note that the directory was changed to system code, hence cd into system code.

```
iss-user@iss-vm: ~/Desktop/Trip@UsTest/TripAtUs/TripAtUs
(TripAtUsTest) iss-user@iss-vm:~/Desktop/Trip@UsTest/TripAtUs$ cd TripAtUs
(TripAtUsTest) iss-user@iss-vm:~/Desktop/Trip@UsTest/TripAtUs/TripAtUs$
```

In order to start the celery server please type the following command

```
iss-user@iss-vm: ~/Desktop/Trip@UsTest/TripAtUs/TripAtUs
(TripAtUsTest) iss-user@iss-vm:~/Desktop/Trip@UsTest/TripAtUs/TripAtUs$ celery -A TripAtUs worker -l info

----- celery@iss-vm v4.4.2 (cliffs)
-----
*****
-- ***** --- Linux-4.15.0-72-generic-x86_64-with-debian-stretch-sid 2020-05-10 12:50:55
-- ***** ---
** ----- [config]
** ----- .> app: TripAtUs:0x7fb956bad7f0
** ----- .> transport: amqp://tripatus:**@localhost:5672/myvhost
** ----- .> results: disabled://
** ----- .> concurrency: 2 (prefork)
** ----- .> task events: OFF (enable -E to monitor tasks in this worker)
** -----
----- [queues]
----- .> celery exchange=celery(direct) key=celery

[tasks]
. Service_Scripts.Async.Plan_My_trip
. Service_Scripts.Async.add
. Service_Scripts.Async.test_celery_worker

[2020-05-10 12:50:55,421: INFO/MainProcess] Connected to amqp://tripatus:**@127.0.0.1:5672/myvhost
[2020-05-10 12:50:55,443: INFO/MainProcess] mingle: searching for neighbors
[2020-05-10 12:50:56,485: INFO/MainProcess] mingle: all alone
[2020-05-10 12:50:56,505: WARNING/MainProcess] /home/iss-user/anaconda3/envs/TripAtUsTest/lib/python3.6/site-packages/
celery/fixups/django.py:203: UserWarning: Using settings.DEBUG leads to a memory
leak, never use this setting in production environments!
[2020-05-10 12:50:56,505: INFO/MainProcess] celery@iss-vm ready.
```

It is important to kill the server when you are not running the application. Please kill the celery work using the below command

```
iss-user@iss-vm: ~/Desktop/Trip@Us/Trip-Us/TripAtUs
(Trip@us) iss-user@iss-vm:~/Desktop/Trip@Us/Trip-Us/TripAtUs$ ps aux | grep -i "celery"
iss-user 28970 0.0 0.0 15976 960 pts/19 S+ 20:37 0:00 grep --color=auto -i celery
(Trip@us) iss-user@iss-vm:~/Desktop/Trip@Us/Trip-Us/TripAtUs$ sudo kill -9 28970
[sudo] password for iss-user:
(Trip@us) iss-user@iss-vm:~/Desktop/Trip@Us/Trip-Us/TripAtUs$
```



STEP 6: FINAL STEP RUN DJANGO

In the same directory, please run the following command to start server at port: 127.0.0.1:8000

```
iss-user@iss-vm: ~/Desktop/Trip@UsTest/TripAtUs/TripAtUs
(TripAtUsTest) iss-user@iss-vm:~/Desktop/Trip@UsTest/TripAtUs/TripAtUs$ python manage.py runserver
Watching for file changes with StatReloader
Performing system checks...

System check identified no issues (0 silenced).
May 10, 2020 - 12:53:27
Django version 2.2.12, using settings 'TripAtUs.settings'
Starting development server at http://127.0.0.1:8000/
Quit the server with CONTROL-C.
[10/May/2020 12:54:28] "GET / HTTP/1.1" 200 10932
[10/May/2020 12:54:28] "GET /static/leaflet/leaflet.js HTTP/1.1" 200 139643
```

You can now open the application in port 127.0.0.1:8000 and start using it 😊

