**Poly-verification suite**

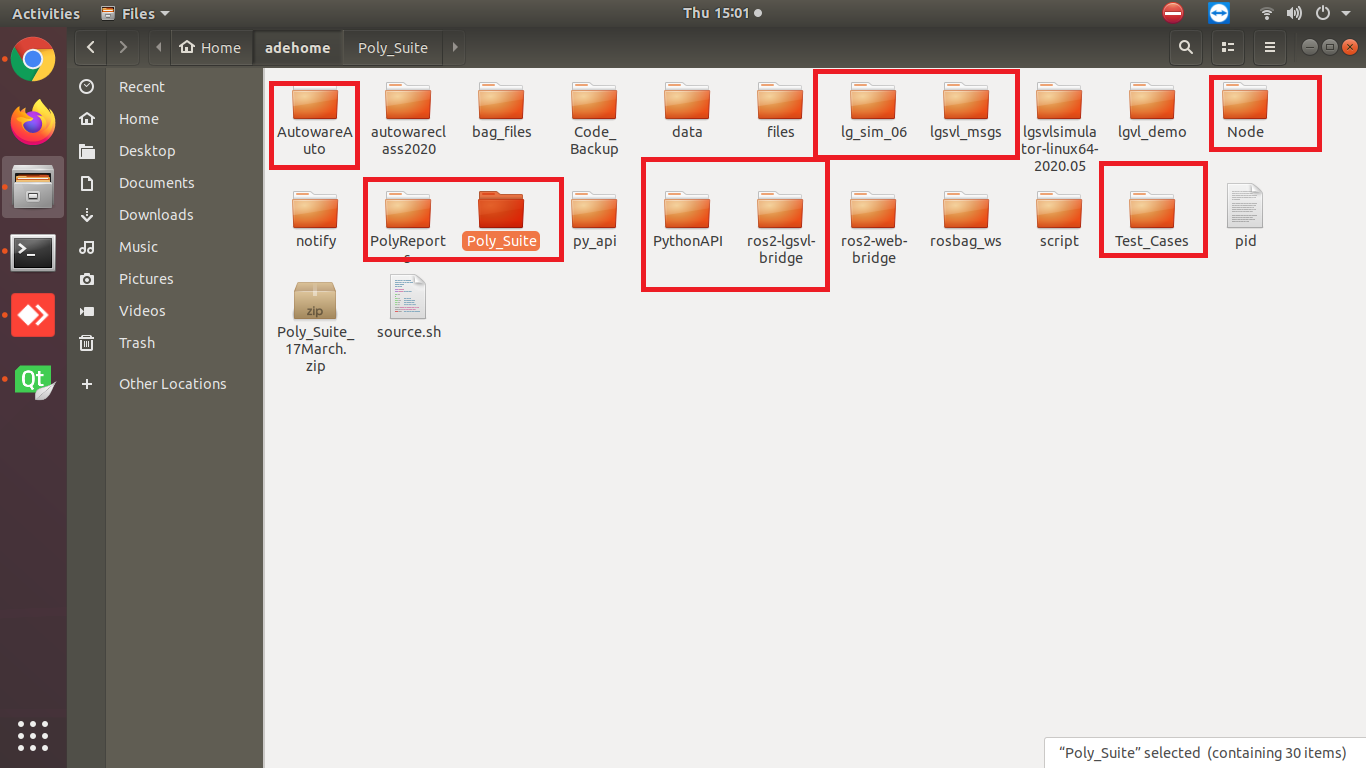
18 March 2021

This document contains the information of how to run and setup the Poly-Verification Suit in local machine. All the pre requisites need to be install in the adehome folder

## Pre requisites

* AutowareAuto
* Lgsvl\_msgs
* LG\_Simulator
* Packages(Node)-
  + Perception Validation (Node\_lg\_groundtruth\_data\_ws)
  + Control Validation (Node\_control\_validation\_ws)
* LGSVL\_Bridge
* Test\_Cases
* Libraries-
  + **Watchdog**
    - Steps to install the watchdog library is mention in the Node\_Setup document, please refer it.
  + **Pandas**
    - Open terminal and run below command

**$ sudo python3 -m pip install pandas**



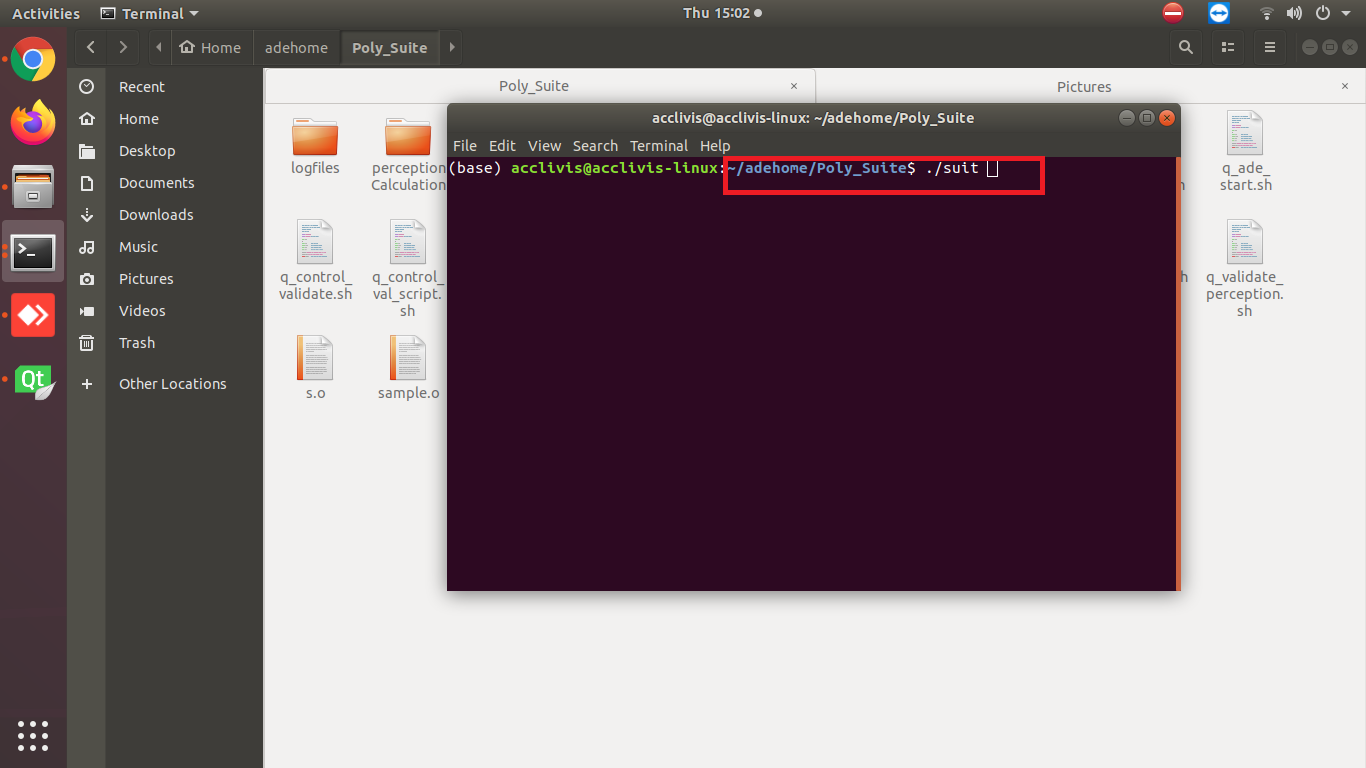
## Steps to run the Poly\_Suite

Keep the following files and folder inside adehome as shown in the above screenshot-

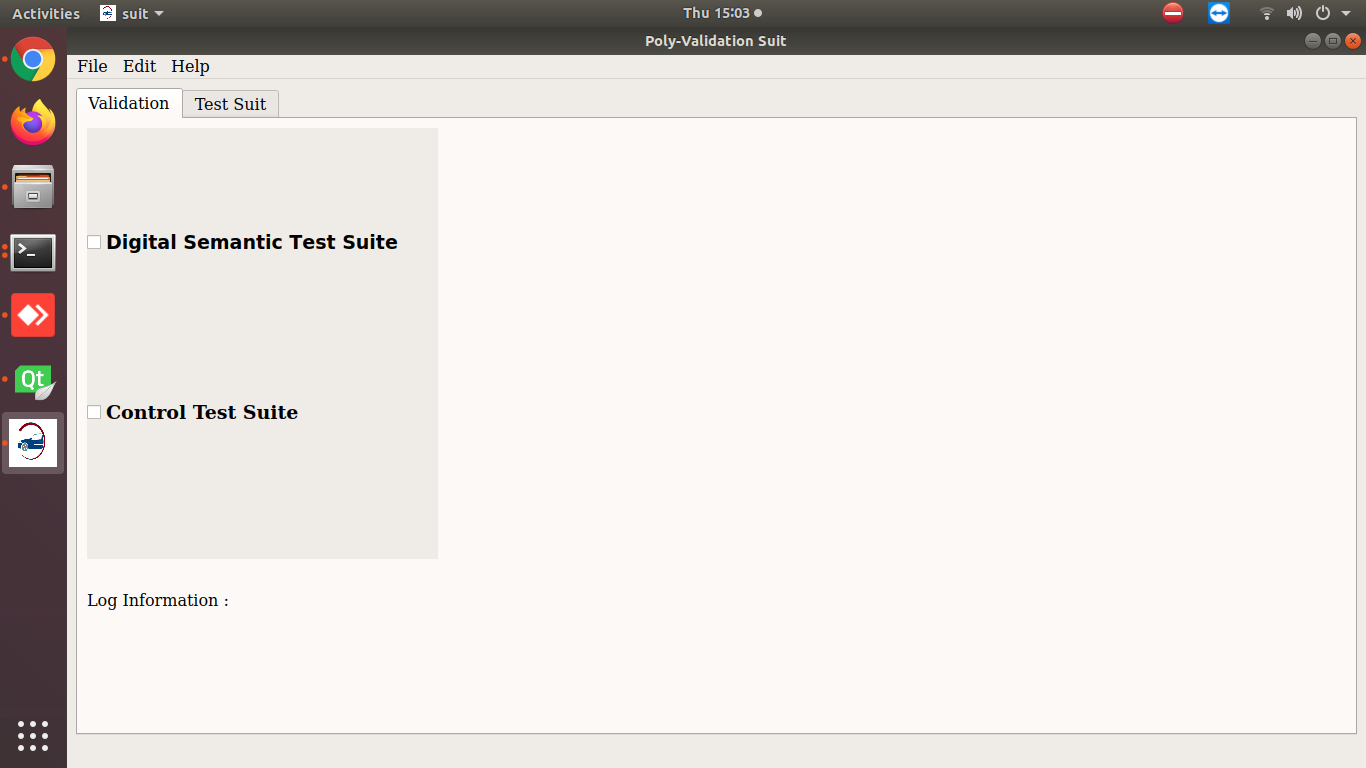
* AutowareAuto
* LG\_Simulator
* ros2-lgsvl-bridge
* lgsvl\_msgs
* Packages/Node
  + Preception Validation (Node\_lg\_groundtruth\_data\_ws)
  + Control Validation (Node\_control\_validation\_ws)
* PolyReports
* Poly\_Suite
* Test\_cases

1. Go to the Poly\_Suite folder
2. Open terminal and run below command:

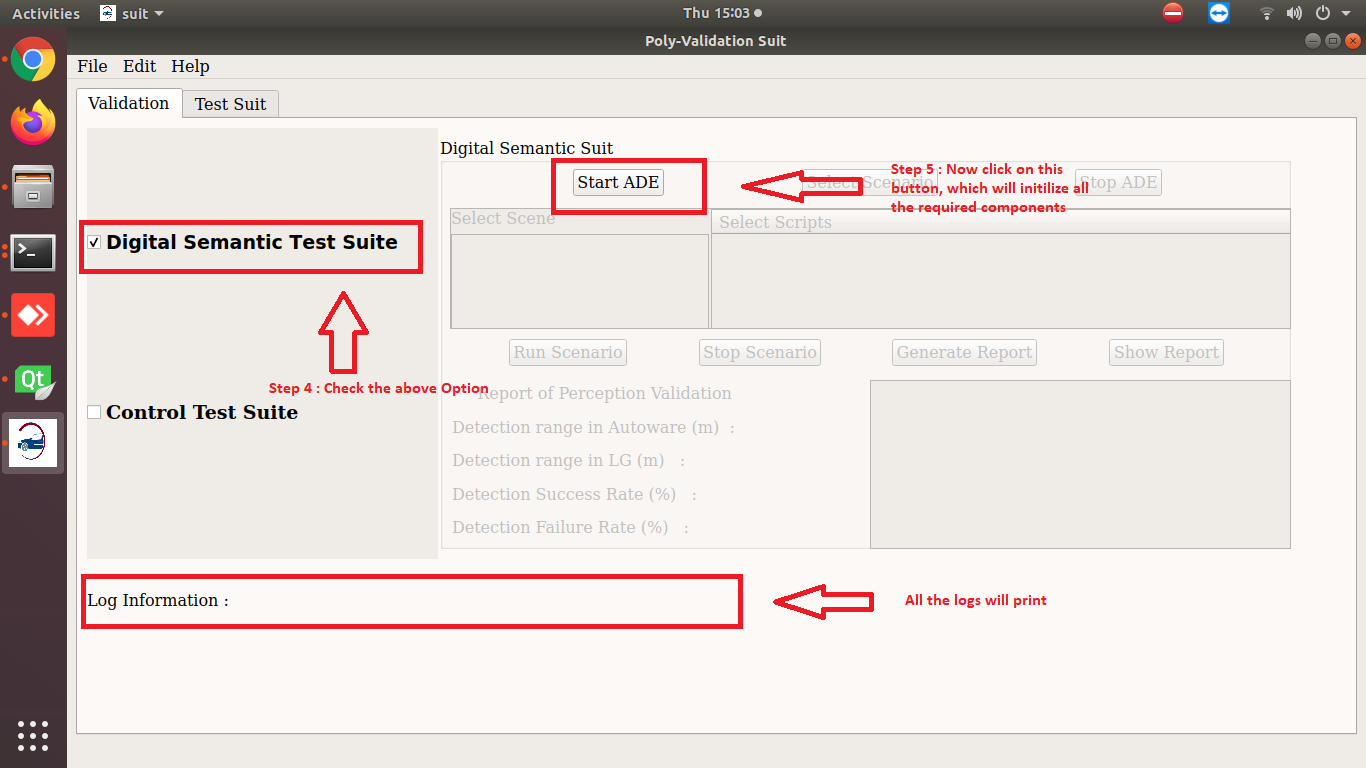
**$ ./suit**

****

1. It will open the Poly\_Suite



1. Select “**Digital Schematic Test Suit**”
2. Now Click on **Start ADE** Button



1. It will start the ADE docker and enter into the ADE docker and also it initiates the LGSVL\_Bridge, LGSVL Simulator, Perception Validation node, Rviz2 with perception stack in background and Browser.

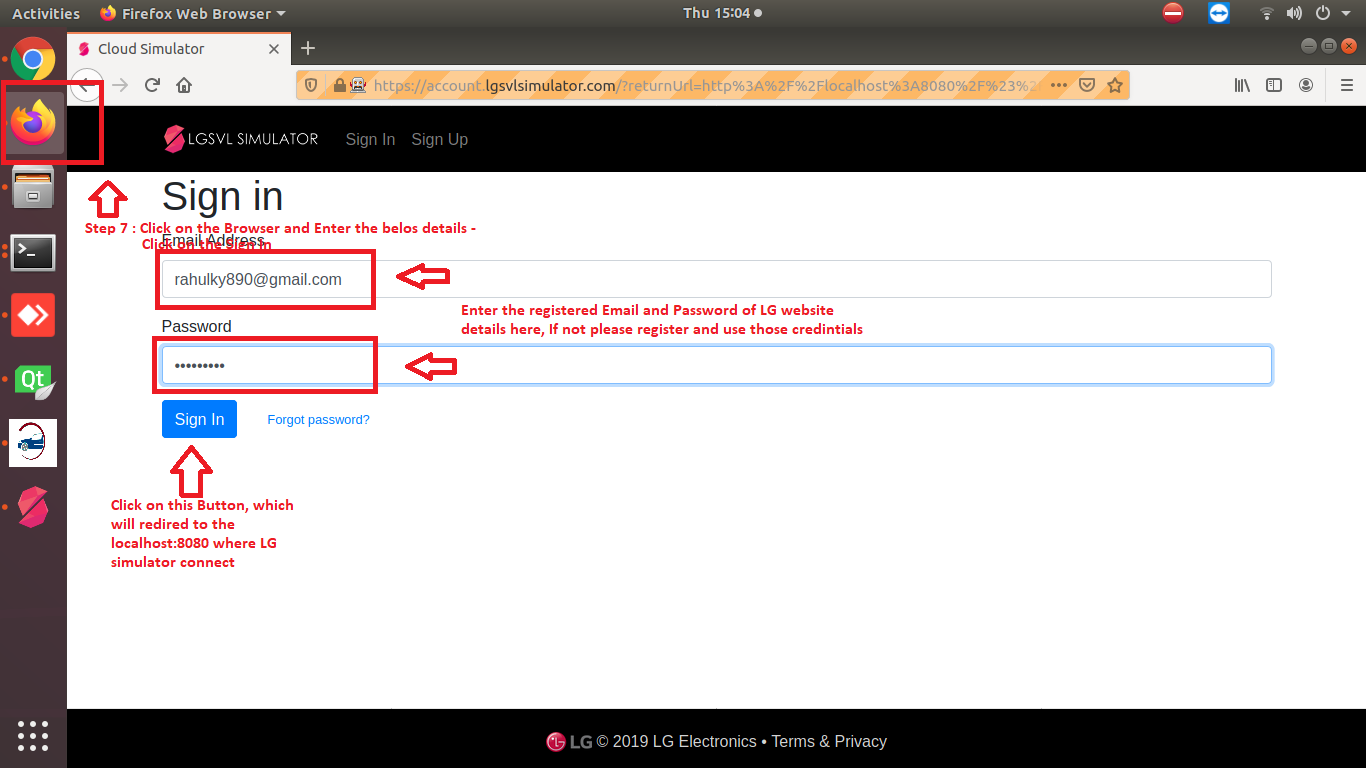
**Note:** Logs of initiating of the above modules will be visible on the Poly\_Suite. Make sure the above modules and applications are running successfully.

If any of is not working/start then please Click on Stop ADE button to stop the ADE docker and Start again.

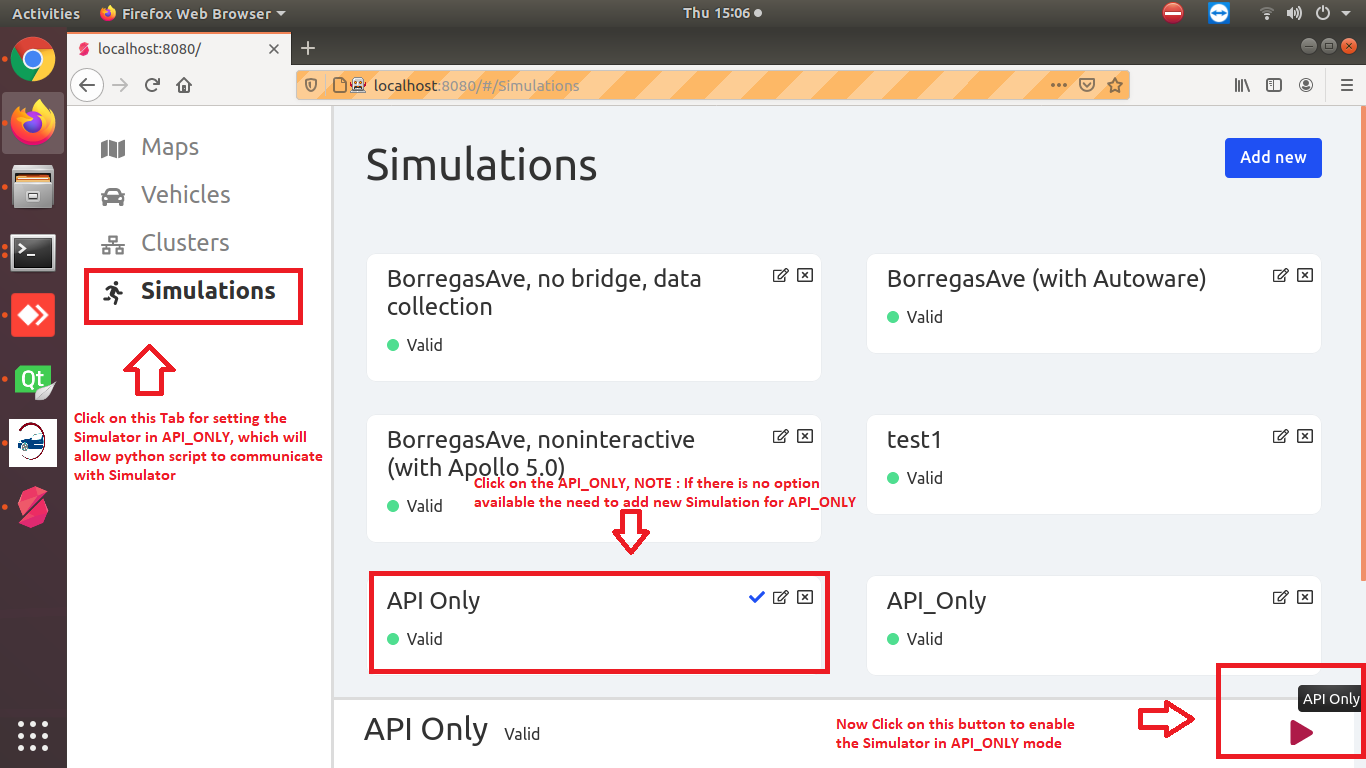


1. Now Click on the Opened web browser and check if it is login successfully on the lg simulator web site or it is asking for the credential.

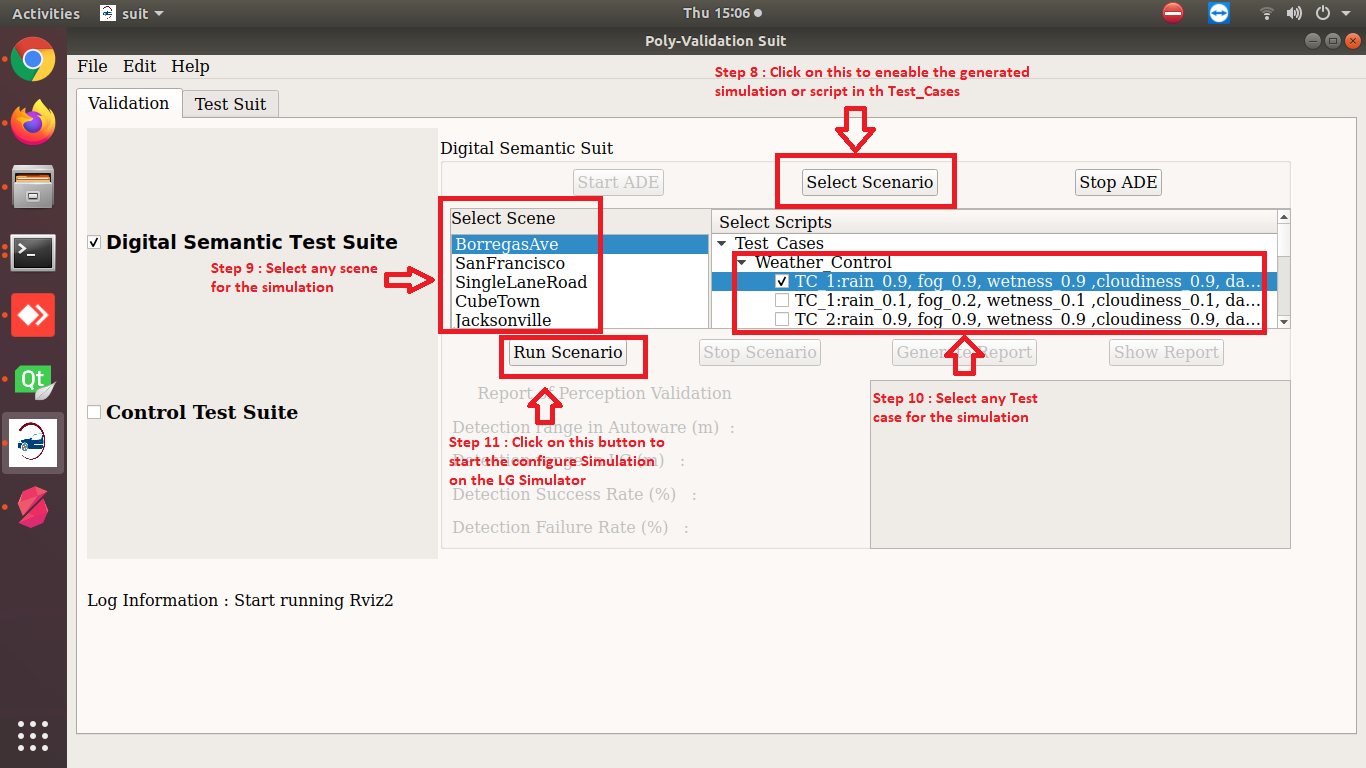
* If it is asking for the credential then please login with the below credentials-
  + **ID:** [**rahulky890@gmail.com**](mailto:rahulky890@gmail.com)
  + **Password: rahul@123**

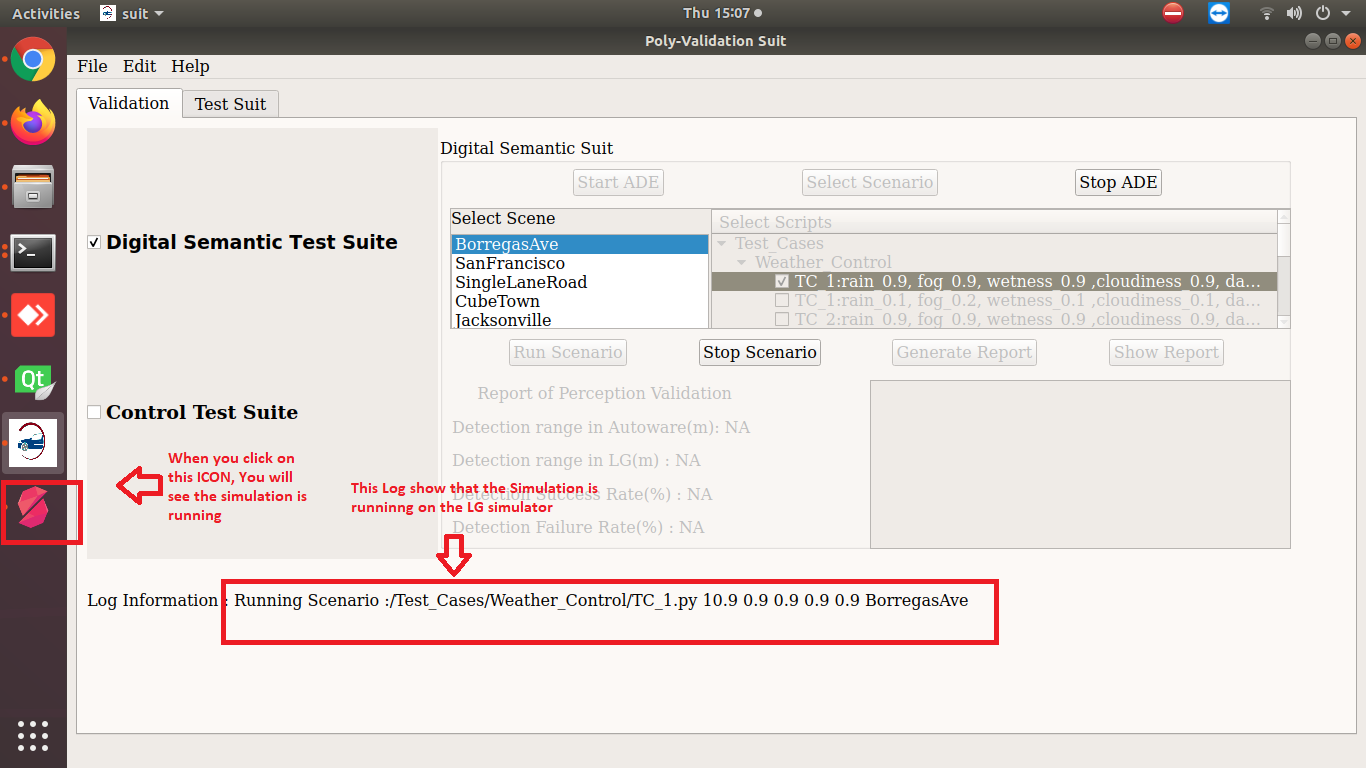
****

* Once it is successfully logged in, click on the **Simulation tab** and Select **API\_ONLY** mode.
* Now Click on the **RUN** **|>** button on the web page which is shown on the right bottom corner.

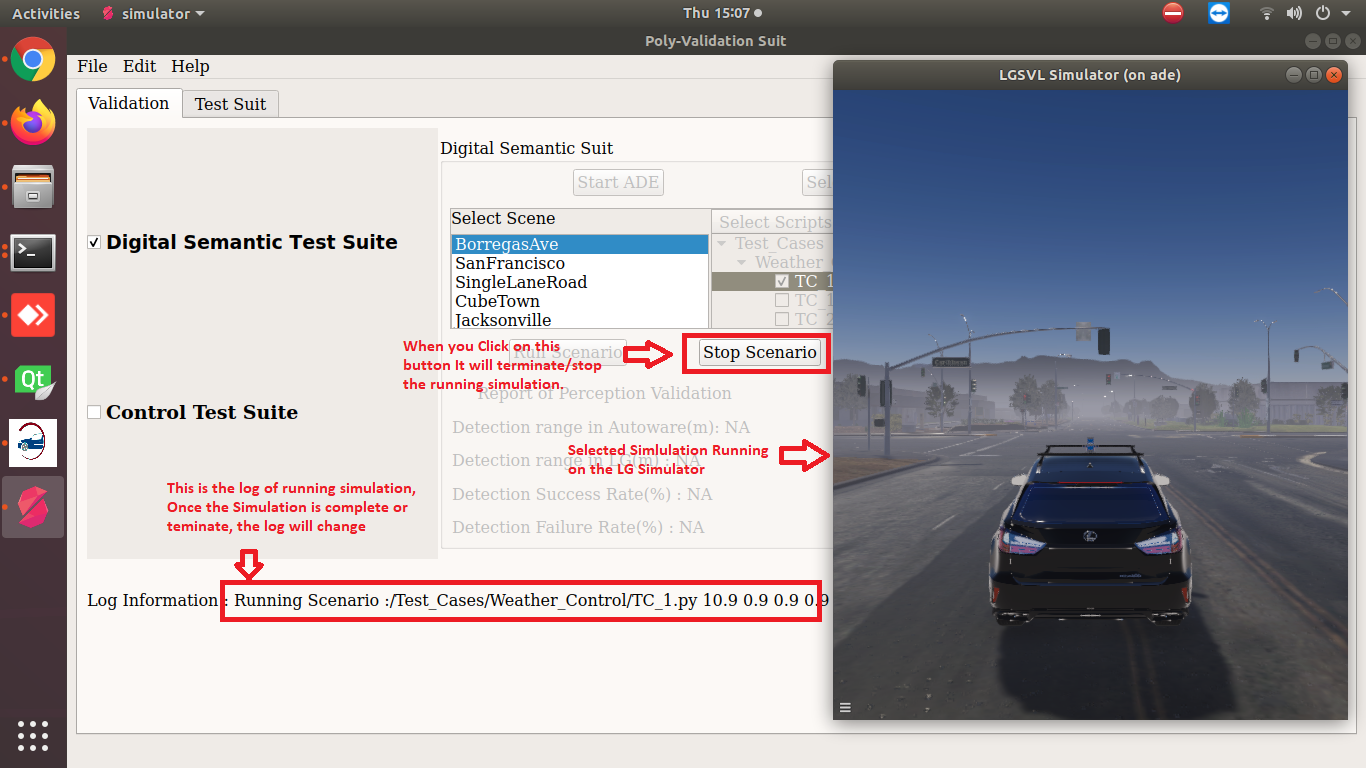


1. Now go to the Poly\_Suite application and Click on Select Scenario button, it will enable the Scenes.
2. Select any scene for lgsvl simulator to run the simulation
3. Now select/mark any single scenario from the list of scenarios
4. Click on the **Run Scenario** Button, it will run the scenario which will be visible on the Lg Simulator application.





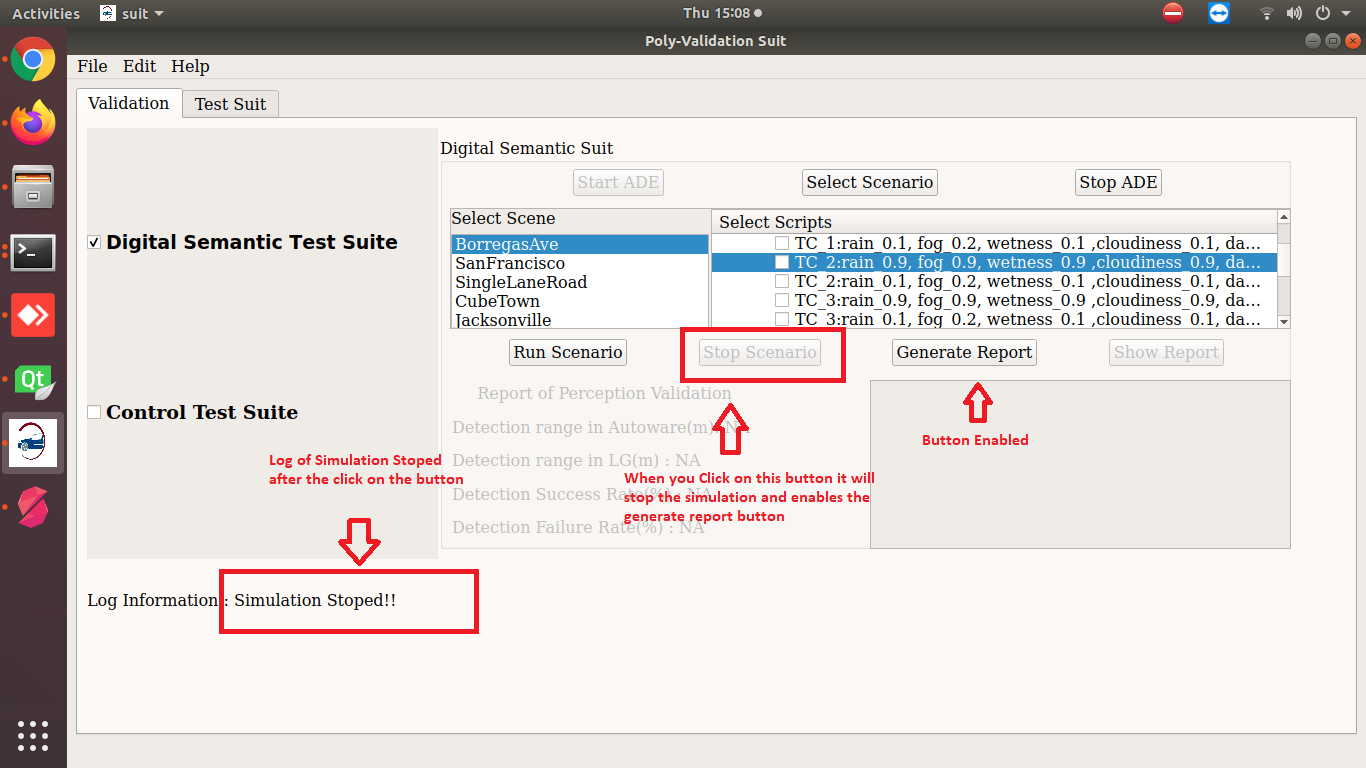
1. Once the Simulation is completed, you can see in the log information about the completion of the simulation.



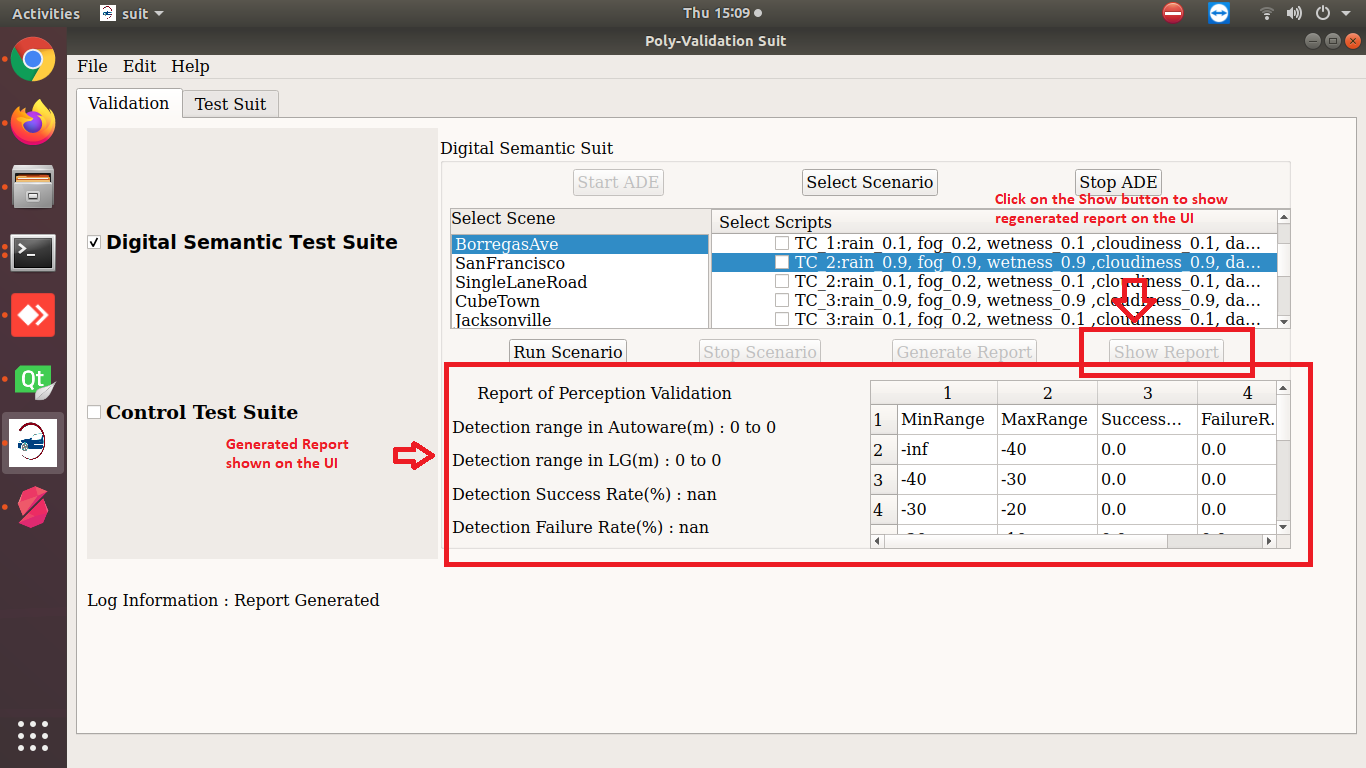
1. Now click on the **Stop Scenario** button which stop/terminate the simulation and enable the **Generate Report** button and also generate files of the simulation which will be shown in the below path-

**Poly\_Report/Validation\_report/<Date\_\_TimeOfRunningTestCase/TestCaseName/>**

1. Now Click on the **Generate Report** button to generate the report



1. Now Click on **Show Report** button, which will present the consolidate report of the simulation on the UI.



1. Again, follow steps from step 9 to 16 to run another simulation.
2. To Stop the Running docker click on the **Stop ADE** button, it will stop LGSVL\_Bridge, LG\_Simulator, Rviz2 and all the other component/module running into the docker.

## Configure the Scenarios

If you want to change weather condition in simulation for your test case then, please follow the below steps:

1. Go to the **Test\_Cases -> Weather\_Control**
2. Now open config.txt file
3. Set your parameters in given pattern ex**.: rain\_0.9, fog\_0.9, wetness\_0.9, cloudiness\_0.9, damage\_0.9**

If you want to add your test case in the test case list then please copy and paste your python script in this folder.

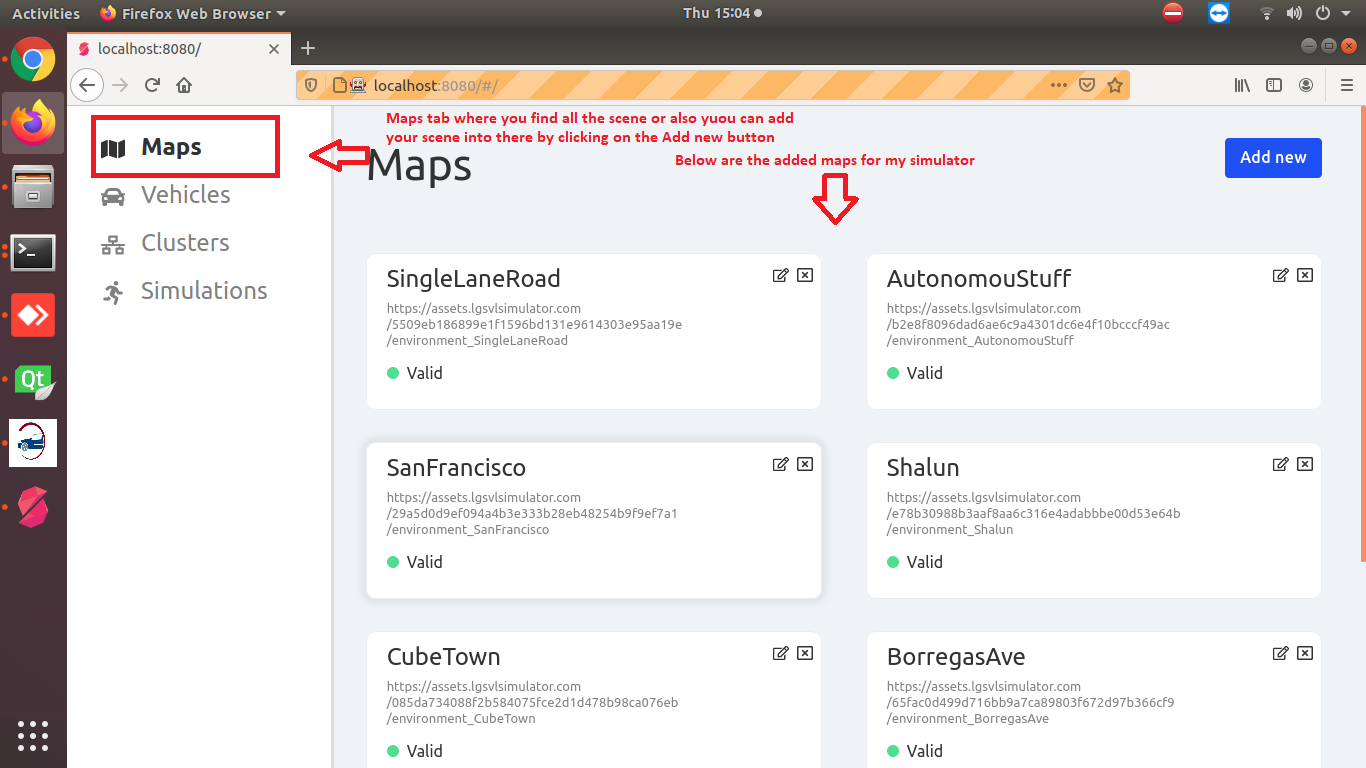
If user want to set the list of scenes for the simulation need to edit the scene.txt file which is in the **Test\_Cases** directory.

Only those scene will be run in the simulation which are available or downloaded from the lg website.

## Things to remember:

* Download and add map for the lg simulator
* Download and add Vehicle for the lg simulator

When login the lg simulator, it will provide the default the Vehicle and its configuration with some Maps on it in the Vehicle and Maps tab respectively.



If it is not available the please add manually using the below link-

Add Vehicle: <https://www.lgsvlsimulator.com/docs/vehicles-tab/#how-to-add-a-vehicle>

Add Maps: <https://www.lgsvlsimulator.com/docs/maps-tab/#how-to-add-a-map>

Make sure the vehicle configuration is using the **ROS2** bridge and also need to add the **3D\_Ground Truth** sensor in the Sensor configuration.

**{**

**"type": "3D Ground Truth",**

**"name": "3D Ground Truth",**

**"params": {**

**"Frequency": 10,**

**"Topic": "/simulator/ground\_truth/3d\_detections"**

**},**

**"transform": {**

**"x": 0,**

**"y": 1.975314,**

**"z": -0.3679201,**

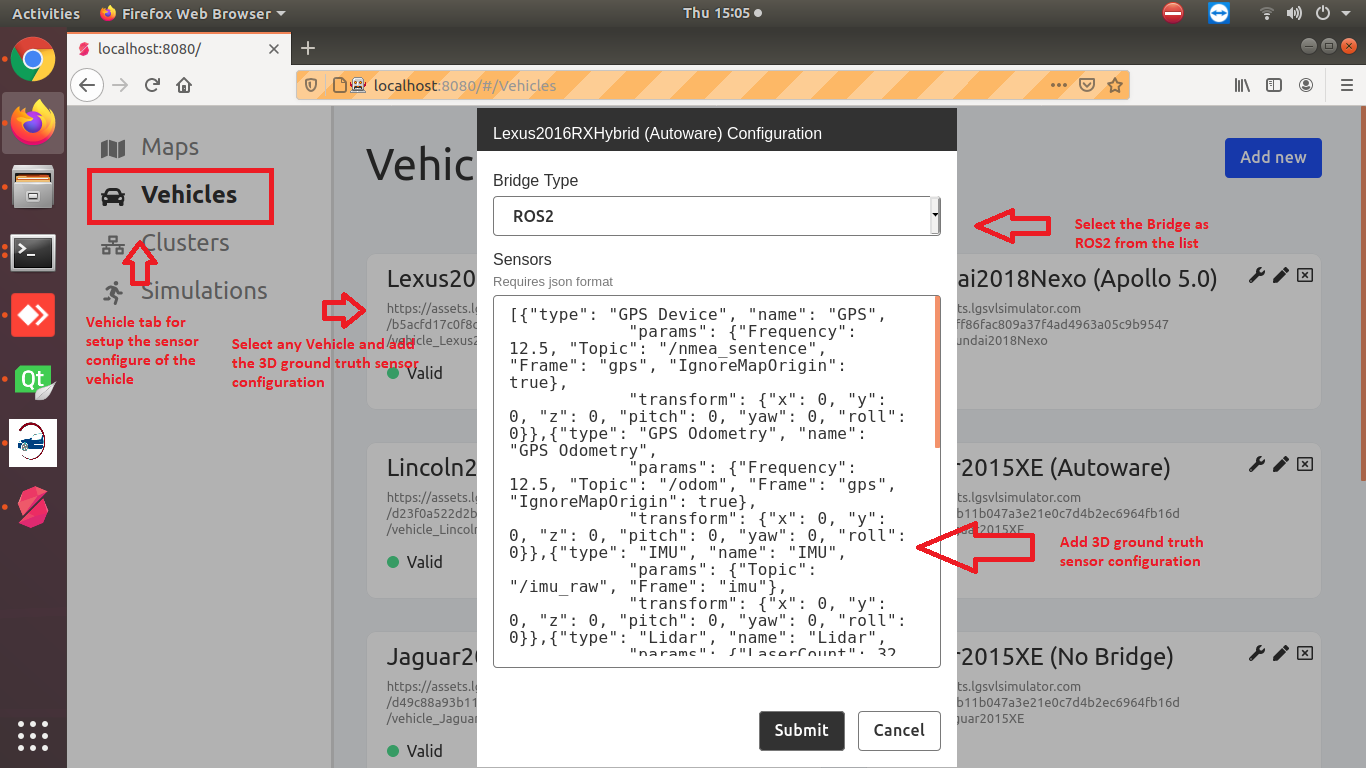
**"pitch": 0,**

**"yaw": 0,**

**"roll": 0**

**}**

**}**

****

That vehicle which you have configured need to be added/modified in the test cases files and below in the below path-

**Test\_Case/Weather\_Control/**

## Note-

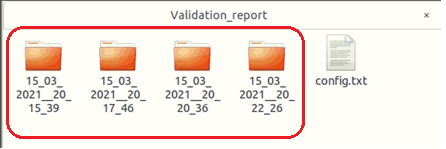
All the pre requisite need to be install and setup before running the Poly\_Suite application as this application has dependencies of the above module and packages.

## Data Gathering and Report Generation for Perception Validation

**Report Directory:** “Poly\_Report/Validation\_report”

“Run Scenario” button runs selected scenario and saves required data in pre-defined directory “Poly\_Report/Validation\_report”.

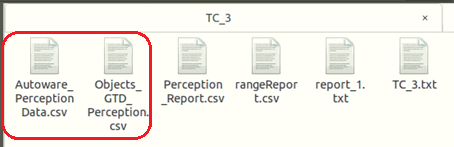
While running first scenario after launching, Polyverif suit creates a new folder in above directory with date and time as shown in below figure.



**Data Storage Files:**

1. Objects\_GTD\_Perception.csv
2. Autoware\_Perception\_data.csv

While running scenario, Polyverif suit captures two data files “Objects\_GTD\_Perception.csv” and “Autoware\_Perception\_data.csv”.



**“Objects\_GTD\_Perception.csv”** file stores objects/vehicles position with respect to ego vehicle, orientation speed and many other parameters, those are mentioned below. As this is captured from LGSVL simulator, this is considered as ground truth data of all objects. In file, all distances are taken from ego vehicle.

1. sensor\_name - Currently, we are using only Lidar data for perception validation so sensor\_name is “lidar”.
2. frame\_id - frameid.
3. timestamp\_sec - timestamp in seconds
4. timestamp\_nanosec - timestamp in nanosecond
5. available - true if data available else false
6. verified - true if data verified by tester or false
7. label – Vehicle label
8. position\_x – x position of object
9. position\_y - y position of object
10. position\_z - z position of object
11. size\_x
12. size\_y
13. size\_z
14. orientation\_x
15. orientation\_y
16. orientation\_z
17. orientation\_w
18. linear\_velocity\_x
19. linear\_velocity\_y
20. linear\_velocity\_z
21. angular\_velocity\_x
22. angular\_velocity\_y
23. angular\_velocity\_z

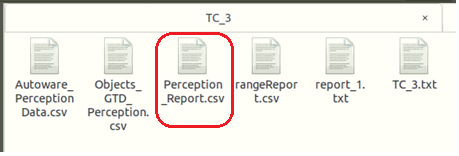
**“Autoware\_Perception\_data.csv**” file stores results of Autoware perception module on Lidar raw data. Captured parameters are below:

1. sensor\_name - Currently, we are using only Lidar data for perception validation so sensor\_name is “lidar”.
2. frame\_id - frameid.
3. timestamp\_sec - timestamp in seconds
4. timestamp\_nanosec - timestamp in nanosecond
5. available - true if data available else false
6. verified - true if data verified by tester or false
7. vehicle\_label - Vehicle label
8. signal\_label
9. class\_likelihood
10. centroid\_x
11. centroid\_y
12. centroid\_z
13. size\_x
14. size\_y
15. size\_z
16. corner\_1\_x
17. corner\_1\_y
18. corner\_1\_z
19. corner\_2\_x
20. corner\_2\_y
21. corner\_2\_z
22. corner\_3\_x
23. corner\_3\_y
24. corner\_3\_z
25. corner\_4\_x
26. corner\_4\_y
27. corner\_4\_z
28. orientation\_x
29. orientation\_y
30. orientation\_z
31. orientation\_w
32. velocity
33. heading
34. heading\_rate
35. value

**Report:**

Perception\_Report.csv

“Generate report” button reads “Objects\_GTD\_Perception.csv” and “Autoware\_Perception\_data.csv” file and process Perception Validation to validate objects visibility by ego vehicle. This process generates “Perception\_Report.csv” with below parameters:



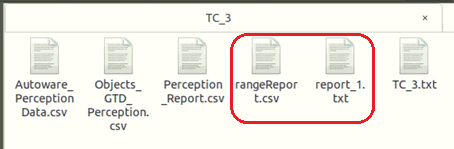
1. timestamp\_sec
2. timestamp\_nanosec
3. label
4. position\_x
5. position\_y
6. position\_z
7. size\_x
8. size\_y
9. size\_z
10. match\_found

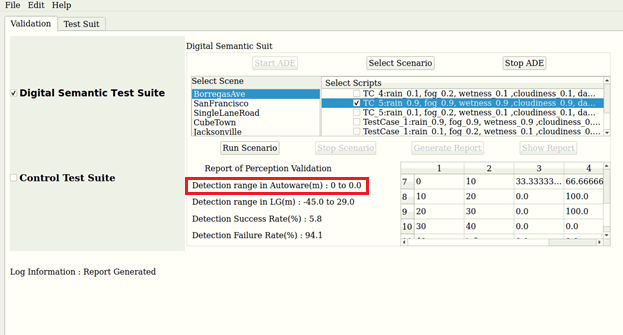
**Details on Parameters (shown in UI):**

**Parameters storage files:**

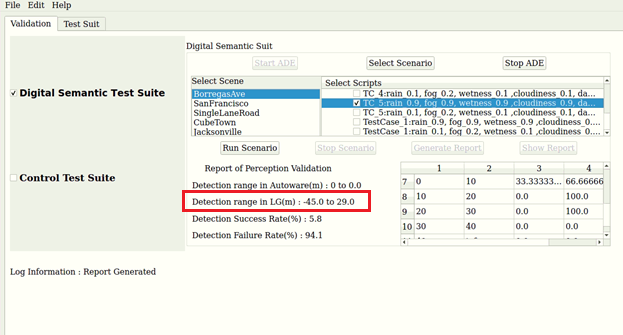
* rangeReport.csv
* report\_1.txt

In last stage, “Show Report” button reads “Perception\_Report.csv” and consolidates data to display on UI, output data is also stored in “rangeReport.csv” and “report\_1.txt” for future reference.

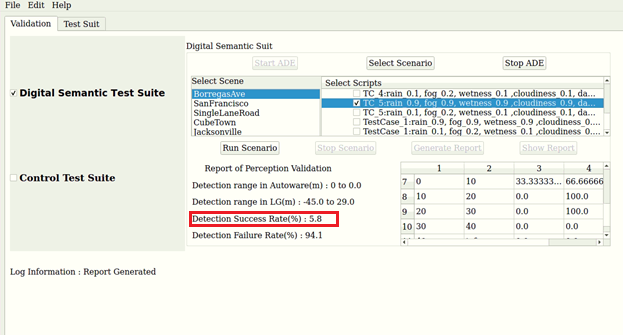




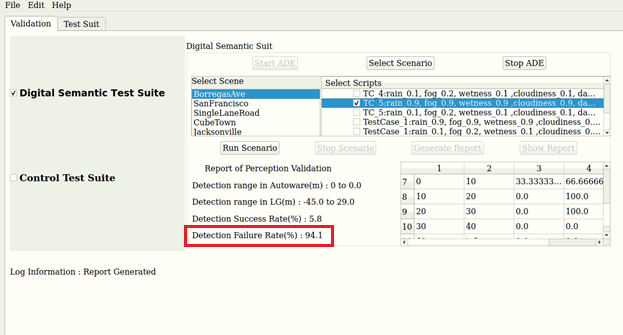
Detection range in Autoware (m) - Maximum and minimum distance of objects/vehicles Autoware perception able to detect, negative value for backside and positive for front side of ego vehicle



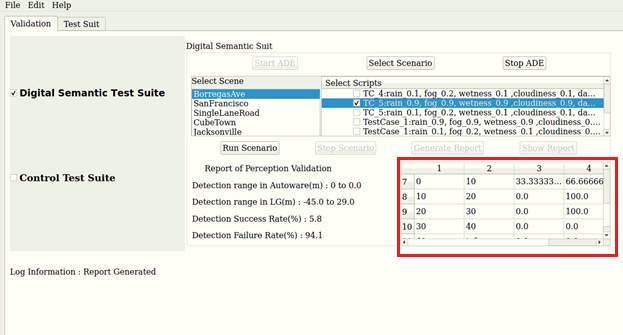
Detection range in LG (m) - Maximum and minimum distance of objects/vehicles available in LGSVL simulator (actual distance from ego vehicle) able to detect, negative value for backside and positive for front side of ego vehicle



Detection Success Rate (%) - Overall vehicle detection success rate, this parameter validates accuracy of perception module.



Detection Failure Rate (%) – Overall vehicle detection failure rate, this parameter validates failure rate of perception module



Above list is showing detection success/failure rate of perception module in a specific range. E.g. in above image when distance is less detection success rate is higher than overall detection success rate.

This report shows 4 columns:

* minRange - start range
* maxRange - end range
* SuccessRate - detection success rate (in percentage)
* FailureRate – detection failure rate (in percentage)