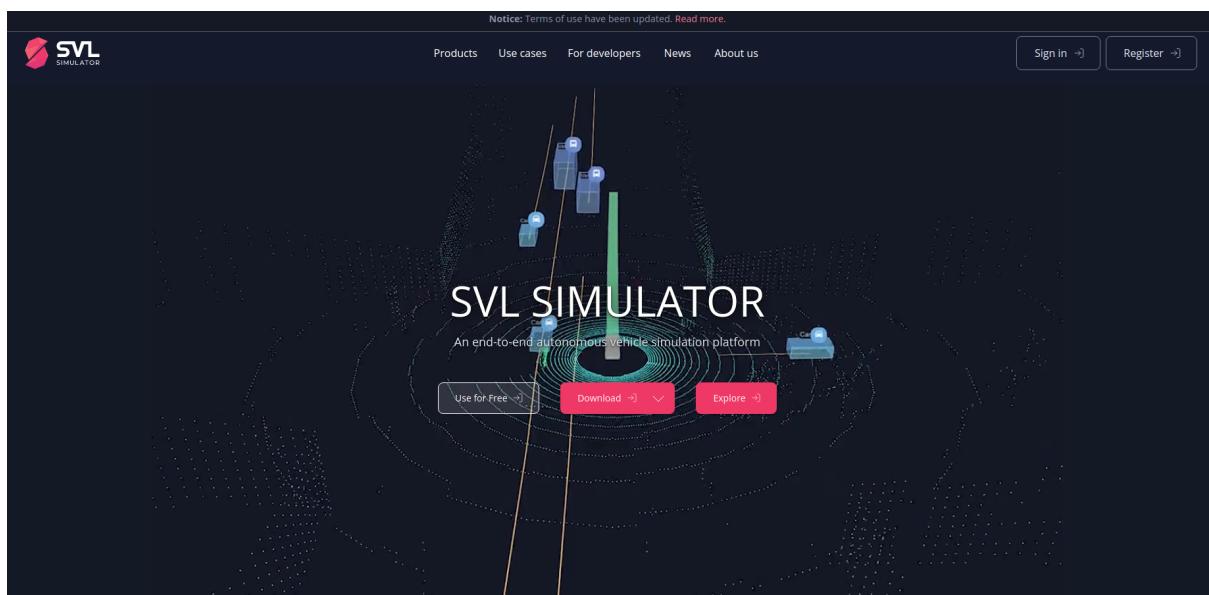


LGSVL SIMULATOR + ROS2 + Autoware.Auto Setup Ver.1.0

LGSVL Simulator

SVL Simulator

- SVL Simulator is a simulation platform used for autonomous vehicle and robotic system development.



Requirements

1. Ubuntu 20.04
2. ROS Foxy
3. Docker CE
4. Nvidia-smi
5. Nvidia-container-toolkit

1. Docker setting

1. 이전 버전 제거

```
$ sudo apt-get remove docker docker-engine docker.io containerd runc
```

2. 레포지토리를 사용한 설치

```
$ sudo apt-get update  
$ sudo apt-get install \  
    apt-transport-https \  
    ca-certificates \  
    curl \  
    gnupg-agent \  
    software-properties-common
```

Do you want to continue [Y/n] 와 같이 묻는 내용 나오면 Y를 입력한다.

3. 다음 명령을 실행해서 도커의 GPG 키를 추가

```
$ curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo apt-key add -
```

4. fingerprint 값 확인

```
$ sudo apt-key fingerprint 0EBFCD88
```

5. 레포지토리 추가

```
$ sudo add-apt-repository \  
    "deb [arch=amd64] https://download.docker.com/linux/ubuntu \  
    $(lsb_release -cs) \  
    stable"
```

6. 도커 패키지 설치

```
$ sudo apt-get update && sudo apt-get install docker-ce docker-ce-cli containerd.io
```

7. 접근권한 설정 (중요)

```
$ sudo /usr/sbin/groupadd -f docker  
$ sudo /usr/sbin/usermod -aG docker `user`  
$ sudo chown root:docker /var/run/docker.sock
```

2. Nvidia-container-toolkit

- <https://docs.nvidia.com/datacenter/cloud-native/container-toolkit/install-guide.html>

3. SVL Simulator Download

1. Download

Release 2021.3 · lgsvl/simulator

New Release 2021.3
2021.3
Added

- Content creation:
 - Annotation size support...

Contributors: 1

Assets (4)

- svisimulator-linux64-2021.3.zip (209 MB)
- svisimulator-windows64-2021.3.zip (184 MB)
- Source code (zip)
- Source code (tar.gz)

리눅스 버전 다운로드

- \$ mkdir -p ~/adehome
- 다운받는 경로는 /home/유저이름/adehome 에 다운받고 압축풀기

4. Install ADE

1. ADE ?

- a. The ADE Development Environment is a modular Docker-based tool to ensure that all developers in a project have a **common, consistent development environment.**

2. ADE Installation 및 업데이트

```
$ cd ~/.local/bin
$ wget https://gitlab.com/ApexAI/ade-cli/uploads/f6c47dc34cffbe90ca197e00098bdd3f/ade+x86_64
$ mv ade+x86_64 ade
$ chmod +x ade
$ ./ade --version
```

```
$ ./ade update-cli  
$ ./ade --version
```

3. Setup ADE home and project checkout

```
$ cd ~/adehome  
$ touch .adehome  
$ cd ~/adehome  
$ git clone https://gitlab.com/autowarefoundation/autoware.auto/AutowareAuto.git  
$ cd AutowareAuto  
$ git checkout tags/1.0.0 -b release-1.0.0
```

4. Set the Autoware.Auto containers with NVIDIA setup

```
$ vim ~/adehome/AutowareAuto/.aderc-amd64-foxy-lgsvl-nvidia
```

| 아래 내용 Copy & Paste

 export ADE_DOCKER_RUN_ARGS="--cap-add=SYS_PTRACE --net=host --privileged --add-host ade:127.0.0.1 -e RMW_IMPLEMENTATION=rmw_cyclonedds_cpp --runtime=nvidia -ti --rm -e DISPLAY -v /tmp/.X11-unix:/tmp/.X11-unix \ -e NVIDIA_VISIBLE_DEVICES=all \ -e NVIDIA_DRIVER_CAPABILITIES=compute,utility,display"
export ADE_GITLAB=gitlab.com
export ADE_REGISTRY=registry.gitlab.com
export ADE_DISABLE_NVIDIA_DOCKER=false
export ADE_IMAGES="
registry.gitlab.com/autowarefoundation/autoware.auto/autowareauto/amd64/ade-foxy:master

registry.gitlab.com/autowarefoundation/autoware.auto/autowareauto/amd64/binary-foxy:master

registry.gitlab.com/autowarefoundation/autoware.auto/ade-lgsvl/foxy:2020.06
nvidia/cuda:11.0-base

"

5. Start ADE Container

```
$ cd ~/adehome/AutowareAuto  
$ ade --rc .ade-amd64-foxy-lgsvl-nvidia start --update --enter
```

아래 사진처럼 ade 로 진입 성공

```
sc@sc-home:~$ cd adehome/
sc@sc-home:~/adehome$ cd AutowareAuto/
sc@sc-home:~/adehome/AutowareAuto$ ade --rc .aderc-amd64-foxy-lgsvl-nvidia start --update --enter
INFO: Using non-default .aderc-amd64-foxy-lgsvl-nvidia configuration file
master: Pulling from autowarefoundation/autoware.auto/autowareauto/amd64/ade-foxy
Digest: sha256:3080f412cd49c8dd6be6d34db7e0c75baa20d3cffa56b740f4c68a660c8ca42a
Status: Image is up to date for registry.gitlab.com/autowarefoundation/autoware.auto/autowareauto/amd64/ade-foxy:master
registry.gitlab.com/autowarefoundation/autoware.auto/autowareauto/amd64/ade-foxy:master
master: Pulling from autowarefoundation/autoware.auto/autowareauto/amd64/binary-foxy
a0d0a0d46f8b: Already exists
3c3fa3e7f630: Pull complete
Digest: sha256:38a2bcae33a694b0e01ebc197d769739574a5bf12502768a2790cb86763296f1
Status: Downloaded newer image for registry.gitlab.com/autowarefoundation/autoware.auto/autowareauto/amd64/binary-foxy:master
registry.gitlab.com/autowarefoundation/autoware.auto/autowareauto/amd64/binary-foxy:master
Creating volume container for registry.gitlab.com/autowarefoundation/autoware.auto/autowareauto/amd64/binary-foxy:master
2020.06: Pulling from autowarefoundation/autoware.auto/ade-lgsvl/foxy
Digest: sha256:1b3d3ea7e056291527e82d212ccaf6e7149a972d578fa6aeee4a366b2a8a3700
Status: Image is up to date for registry.gitlab.com/autowarefoundation/autoware.auto/ade-lgsvl/foxy:2020.06
registry.gitlab.com/autowarefoundation/autoware.auto/ade-lgsvl/foxy:2020.06
11.0-base: Pulling from nvidia/cuda
Digest: sha256:774ca3d612de15213102c2dbbba55df44dc5cf9870ca2be6c6e9c627fa63d67a
Status: Image is up to date for nvidia/cuda:11.0-base
docker.io/nvidia/cuda:11.0-base
Starting ade with the following images:
ade-foxy | ab2d6fb1014 | master | registry.gitlab.com/autowarefoundation/autoware.auto/autowareauto/amd64/ade-foxy:master
binary-foxy | 99d478f92a98 | master | registry.gitlab.com/autowarefoundation/autoware.auto/autowareauto/amd64/binary-foxy:master
foxy | 2020.06 | 2020.06 | registry.gitlab.com/autowarefoundation/autoware.auto/ade-lgsvl/foxy:2020.06
cuda | n/a | 11.0-base | nvidia/cuda:11.0-base

ade_registry.gitlab.com_autowarefoundation_autoware.auto_autowareauto_amd64_binary-foxy_master
ade_registry.gitlab.com_autowarefoundation_autoware.auto_ade-lgsvl_foxy_2020.06
ade_nvidia_cuda_11.0-base
non-network local connections being added to access control list

Current default time zone: 'Asia/Seoul'
Local time is now: Thu Nov 11 22:07:29 KST 2021.
Universal Time is now: Thu Nov 11 13:07:29 UTC 2021.

Adding user sc to group video
Adding user sc to group dialout
ADE startup completed.
Entering ade with following images:
ade-foxy | ab2d6fb1014 | master | registry.gitlab.com/autowarefoundation/autoware.auto/autowareauto/amd64/ade-foxy:master
binary-foxy | 99d478f92a98 | master | registry.gitlab.com/autowarefoundation/autoware.auto/autowareauto/amd64/binary-foxy:master
foxy | 2020.06 | 2020.06 | registry.gitlab.com/autowarefoundation/autoware.auto/ade-lgsvl/foxy:2020.06
cuda | n/a | 11.0-base | nvidia/cuda:11.0-base

sc@ade:~$ _
```

5. Install lgsvl-bridge & lgsvl-msgs

```
sc@sc-home:~$ apt-cache search ros-foxy-lgsvl*
ros-foxy-lgsvl-bridge - LGSVL Simulator Bridge
ros-foxy-lgsvl-bridge-dbgsym - debug symbols for ros-foxy-lgsvl-bridge
ros-foxy-lgsvl-msgs - Message definitions for interfacing with the LGSVL Simulator for ROS and ROS 2.
ros-foxy-lgsvl-msgs-dbgsym - debug symbols for ros-foxy-lgsvl-msgs
```

```
$ sudo apt -y install ros-foxy-lgsvl*
```

6. SVL Simulator Run

터미널 새로 열어서 ade enter 를 입력하면 자동으로 ade 컨테이너에 진입.

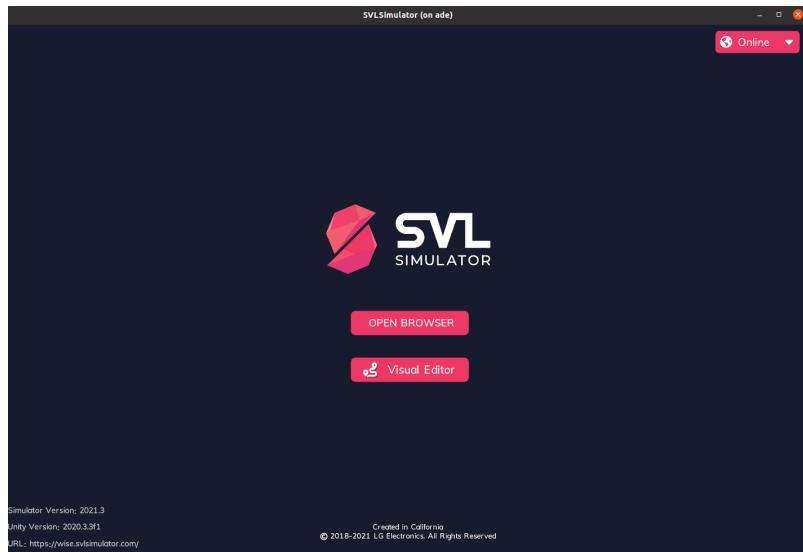
(무조건 4번의 과정이 선행되어야함)

파일 구조를 살펴보면 다음과 같다.

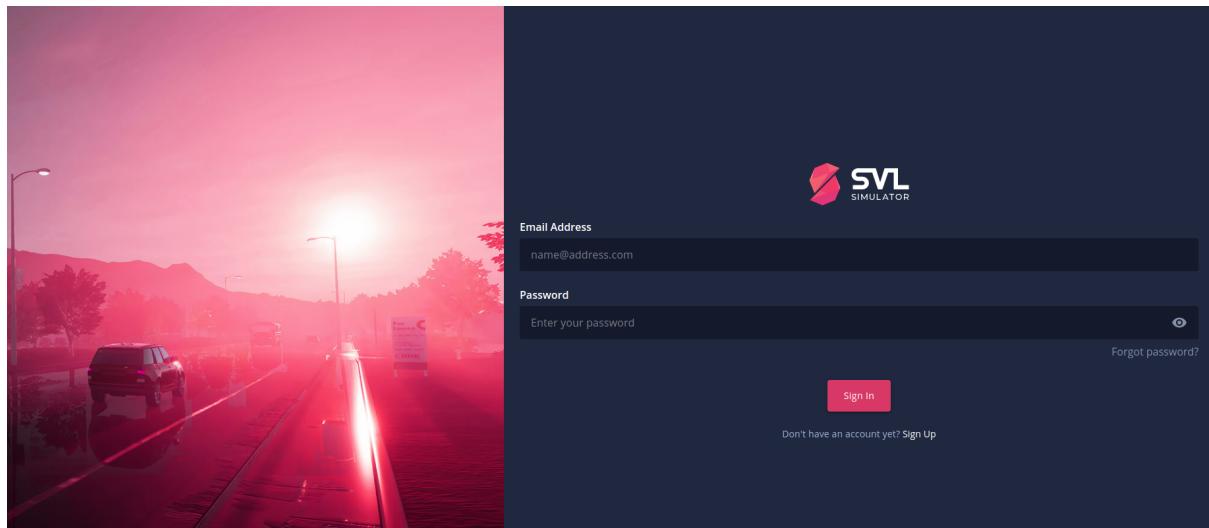
```
sc@ade:~$ ll
total 84
drwxrwxr-x 15 sc  sc  4096 Nov  8  00:40 .
drwxr-xr-x  1 root root 4096 Nov 11 22:07 ..
-rw-rw-r--  1 sc  sc    0 Nov  2  00:08 .adehome
drwxrwxr-x 12 sc  sc  4096 Nov  7 22:37 AutowareAuto/
-rw-----  1 sc  sc  2645 Nov  8  01:27 .bash_history
-rw-r--r--  1 sc  sc   220 Feb 25 2020 .bash_logout
-rw-r--r--  1 sc  sc  4485 Oct  4 19:38 .bashrc
drwxr-xr-x  6 sc  sc  4096 Nov  7 16:16 .cache/
drwxr-xr-x  5 sc  sc  4096 Nov  7 15:20 .config/
drwx----- 3 sc  sc  4096 Nov  2  00:40 .dbus/
drwx----- 2 sc  sc  4096 Nov  7 15:47 Downloads/
drwxr-xr-x  3 sc  sc  4096 Nov  7 15:20 .local/
drwxr-xr-x  3 sc  sc  4096 Nov  7 15:20 .mono/
drwx----- 3 sc  sc  4096 Nov  7 15:20 .pki/
-rw-r--r--  1 sc  sc   807 Feb 25 2020 .profile
drwxr-xr-x  3 sc  sc  4096 Nov  2  00:41 .ros/
drwxr-xr-x  2 sc  sc  4096 Nov  2  00:49 .rviz2/
drwx----- 2 sc  sc  4096 Nov  1 17:35 .ssh/
drwxr-xr-x  4 sc  sc  4096 Oct  1 09:15 svlsimulator-linux64-2021.3/
-rw-----  1 sc  sc   761 Nov  8  00:40 .viminfo
drwxr-xr-x  2 root root 4096 Nov  4 21:33 .Xauthority/
```

```
$ cd svlsimulator-linux64-2021.3/
$ ./simulator
```

실행 화면은 다음과 같다.

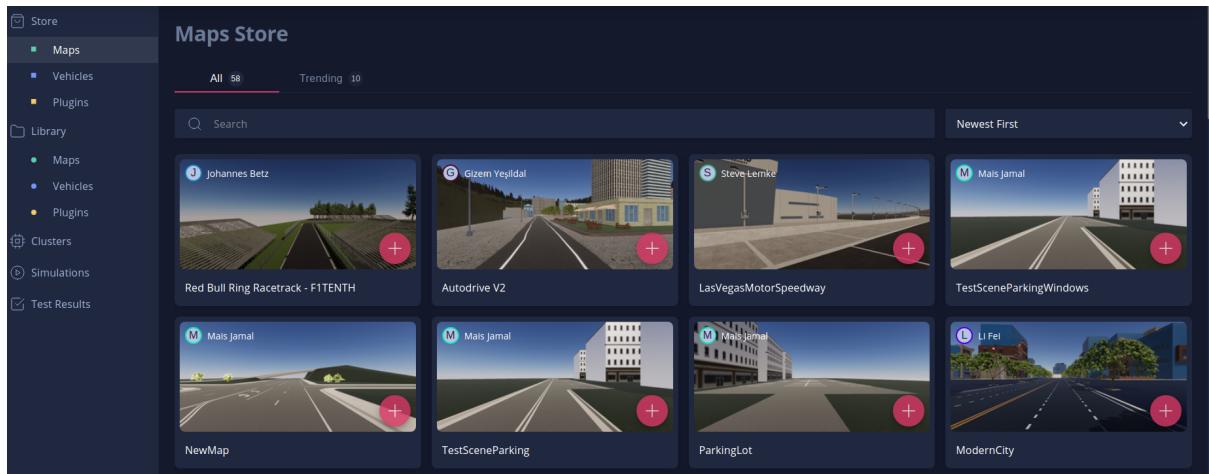


1. 우측 상단 부분 ONLINE 표시
2. OPEN BROWSER 클릭 : LGSVL web이 열림



시뮬레이터를 사용하기 위해서는 개인 계정이 있어야합니다.

개인 계정 생성 후에, 로그인 하시면 되겠습니다.



좌측 카테고리 설명을 드리자면 ,

Store에서 제가 원하는 Map/Vehicle/각종 Plugin 추가할 수 있음 → Library에 저장

개인 환경에서 실험을 진행하기 위해서는 **Clusters**를 생성해야합니다.



ADD New 를 클릭하여 본인 cluster를 생성하면 되겠습니다.

그 다음, Simulation Tap에 들어가서 New Simulation 클릭.

Simulations > New simulation

General

Simulation Name
TEST FOR ADE

Description
Enter a quick description of the simulation

Tags
Enter comma separated values. Example: tag1, tag2, tag3

Select Cluster
ade_cluster

Create test report

Headless mode

Interactive mode

Next

Select Cluster 부분 : 본인 계정 Cluster 선택 후 Next 클릭

General

Test case

Autopilot

Publish

Runtime Template
Random Traffic

Map
BorregasAve

Vehicle
Lexus2016RXHybrid

Autoware.Auto

2021.3, 2021.2, 20...

+ Add another vehicle

- Runtime Template : Random Traffic 선택
- Map : BorregasAve
- Vehicle : Lexus2016RXHybrid - Autoware.Auto : 오토웨어 차량 제한

Date and weather

Simulation Date	Time Of Day
November 11 <input type="button" value=""/>	12:00 PM <input type="button" value=""/>
Set the environment time of day of the simulation.	
Rain	Fog
0	0
Defines the amount of rain. This introduces particle droplet effects falling from the sky and camera post-processing effects. 0.0 - no rain, 1.0 - maximum raining.	
Wetness	Cloudiness
0	0
Defines the amount of water/wetness that covers the road and sidewalks. 0.0 - surface is dry, 1.0 - surface is fully covered with puddles.	
Defines the amount of clouds in the sky. 0.0 - sky is clear, 1.0 - sky is fully covered.	
Traffic	
(?) Random Traffic <input checked="" type="checkbox"/>	(?) Random Pedestrians <input checked="" type="checkbox"/>
(?) Random Bicyclists <input checked="" type="checkbox"/>	(?) Use Pre-defined Seed <input checked="" type="checkbox"/>

- Date and weather : 시뮬레이션 할 날짜/시간대/강수/안개/노면상태/구름상태 설정 가능
- Traffic : 교통 정보와 관련된 object를 임의로 맵에 설정
- 설정 완료 후에 하단 Next 클릭

General	Test case	Autopilot	Publish
----------------	------------------	------------------	----------------

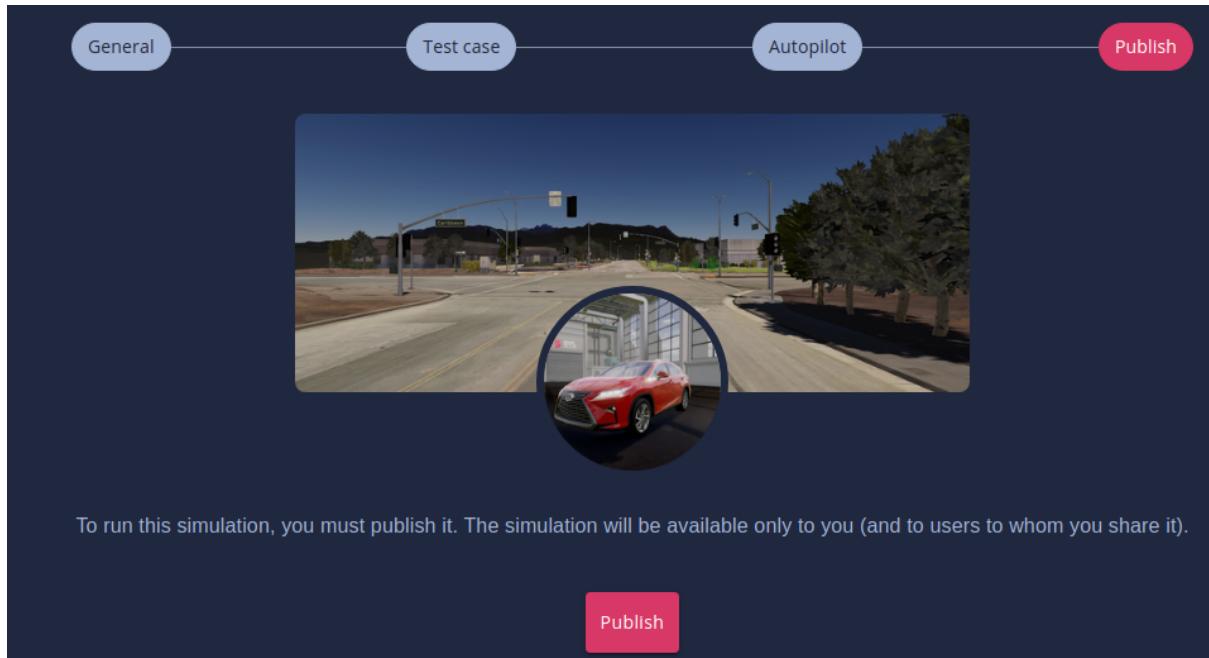
Lexus2016RXHybrid

Autopilot	Bridge Connection
Autoware.Auto (Apex.AI) <input type="button" value=""/>	localhost:9090 <input type="button" value=""/>

(?) Apex.Autonomy is a fork of Autoware.Auto that has been made so robust and reliable that it can be used in safety-critical applications. Apex.Autonomy is developed in sync with future releases of Autoware.Auto and APIs stay compatible to Autoware.Auto.

<input type="button" value="Save as draft"/>	Next
--	------

- Autopilot : Autoware.Auto(Apex.AI) 설정
- Bridge Connection : local machine 환경에서 9090 실시
- 설정 완료 후에 하단 Next 클릭

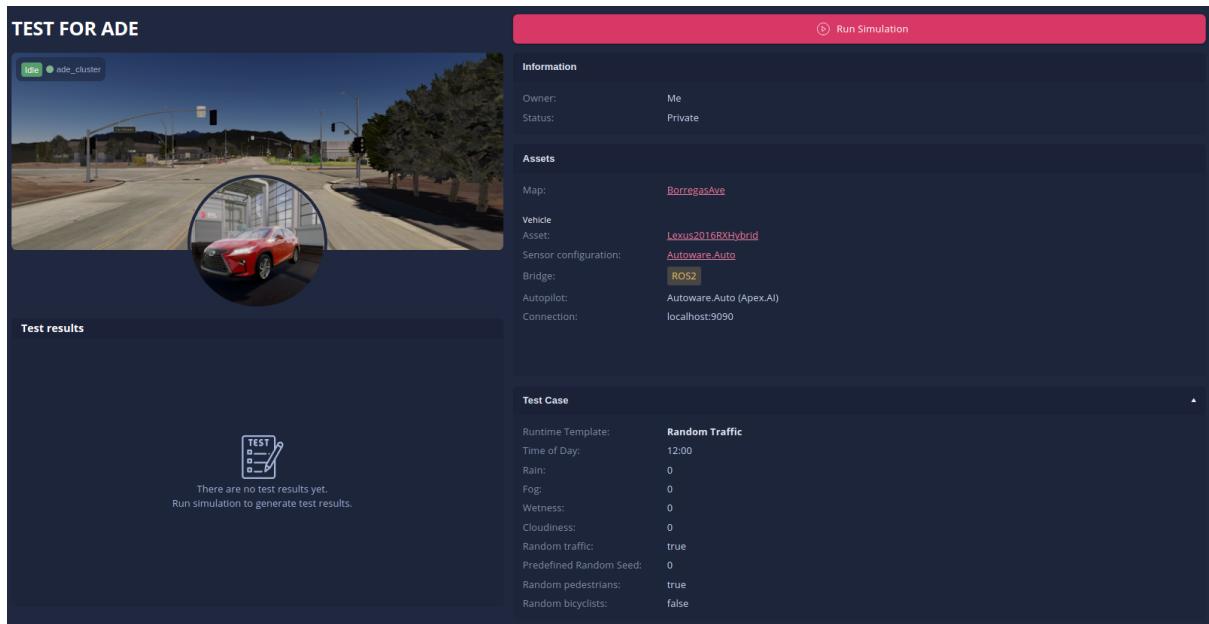


- publish 클릭
- 터미널 창에서 lgsvl bridge를 실행시킨다.

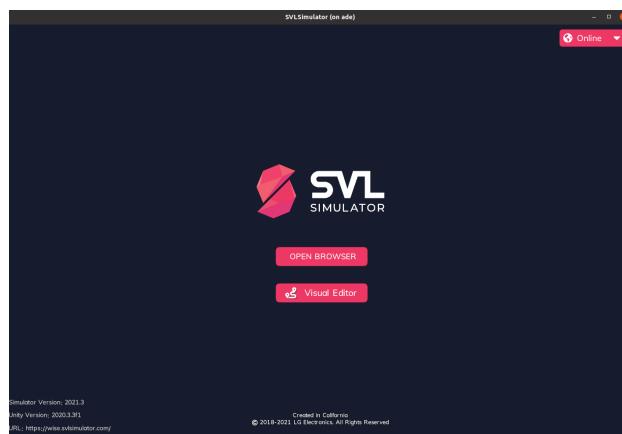
```
$ lgsvl_bridge
```

```
sc@sc-home:~/adehome$ lgsvl_bridge
[INFO] [1636641793.782130970] [lgsvl-bridge]: Listening on port 9090
```

- 포트 연결 확인 후에 다시 시뮬레이터 웹으로 돌아와서 RUN Simulation 버튼 클릭



실행화면이 다음과 같이 바뀐다.





7. Simulator 상세 조작 설명

Controls

Simulator

` - Cinematic/Free camera
1-0 - Agent select and follow cam
F1 - Toggle help
F12 - Reset active agent
N - Toggle NPCs
P - Toggle Pedestrians

Camera

WASD - Forward/Back Strafe
QE - Up/Down
Mouse RightClick - Look
Left Shift - Boost
WS - Zoom

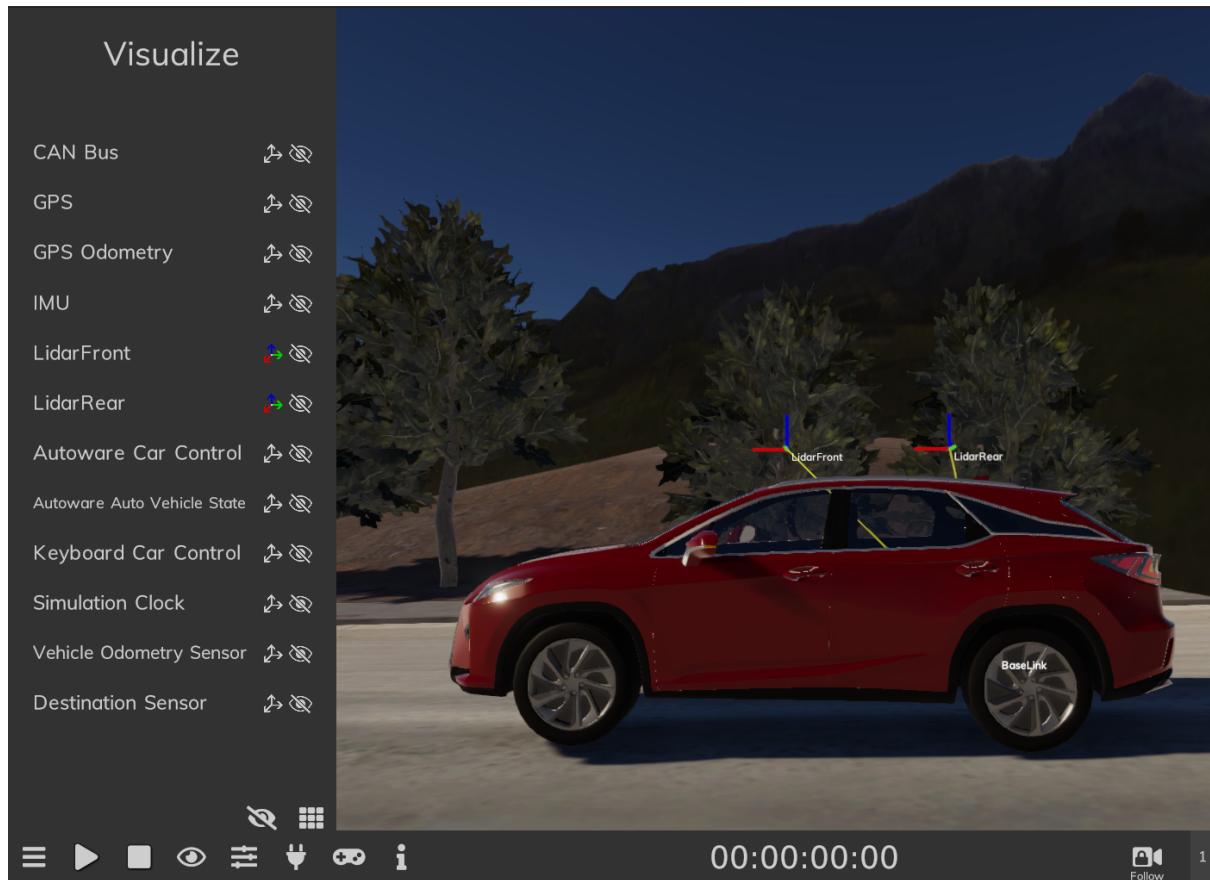


00:00:00:00

- 시뮬레이터 조작 방법



- Bridge 상태 : 양호 - 플레이 버튼을 누르면 플랫폼, 센서 데이터 통신 시작



- 센서간 tf 관계도 도식화 가능하며, eye 표시를 해제하면 시뮬레이터 상에서 센서 데이터 출력 가능



- 주행하는데 필요한 실험 환경도 시뮬레이터 안에서 설정 가능합니다.

8. 센서 데이터 확인



- PLAY 버튼을 누르면 실험 시간이 기록됨과 동시에 센서 및 차량 데이터가 Publish되는 것을 확인할 수 있습니다.

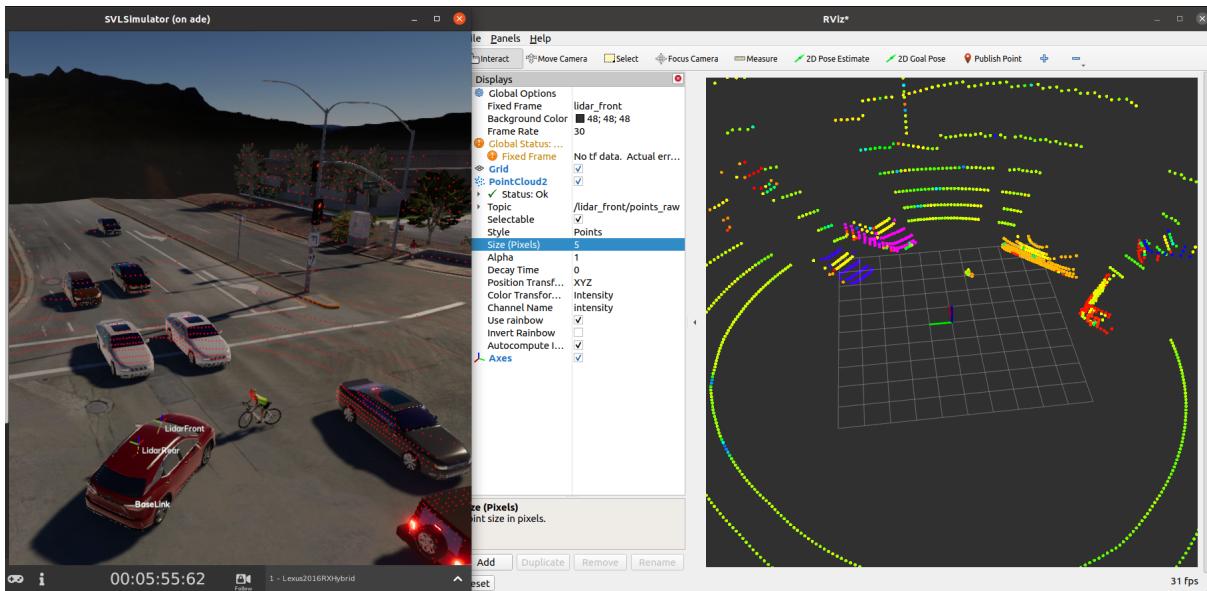
```

sc@sc-home:~$ ros2 topic list
/clicked_point
/gnss/fix
/goal_pose
 imu/imu_raw
/initialpose
/lgsvl/clock
/lgsvl/gnss_odom
/lgsvl/state_report
/lgsvl/vehicle_control_cmd
/lgsvl/vehicle_odom
/lgsvl/vehicle_state_cmd
/lidar_front/points_raw
/lidar_rear/points_raw
/localization/initialpose
/parameter_events
/planning/goal_pose
/rosout
/tf
/tf_static
sc@sc-home:~$ _

sc@sc-home:~$ mkdir -p ~/ros2_lgsvl_bag_data
sc@sc-home:~$ cd ~/ros2_lgsvl_bag_data/
sc@sc-home:~/ros2_lgsvl_bag_data$ ros2 bag record -a
[INFO] [1636642957.691504576] [rosbag2_storage]: Opened database 'rosbag2_2021_11_12-00_02_37/rosbag2_2021_11_12-00_02_37_0.db3' for READ_WRITE.
[INFO] [1636642957.701600855] [rosbag2_transport]: Listening for topics...
[INFO] [1636642957.702338470] [rosbag2_transport]: Subscribed to topic '/rosout'
[INFO] [1636642957.702792755] [rosbag2_transport]: Subscribed to topic '/parameter_events'
[INFO] [1636642958.412659280] [rosbag2_transport]: Subscribed to topic '/lgsvl/vehicle_state_cmd'
[INFO] [1636642958.413309379] [rosbag2_transport]: Subscribed to topic '/goal_pose'
[INFO] [1636642958.413793376] [rosbag2_transport]: Subscribed to topic '/lgsvl/vehicle_odom'
[INFO] [1636642958.414283048] [rosbag2_transport]: Subscribed to topic '/tf'
[INFO] [1636642958.414801098] [rosbag2_transport]: Subscribed to topic '/localization/initialpose'
[INFO] [1636642958.416418536] [rosbag2_transport]: Subscribed to topic '/gnss/fix'
[INFO] [1636642958.417030527] [rosbag2_transport]: Subscribed to topic '/lidar_front/points_raw'
[INFO] [1636642958.417930766] [rosbag2_transport]: Subscribed to topic '/clicked_point'
[INFO] [1636642958.418738879] [rosbag2_transport]: Subscribed to topic '/lgsvl/clock'
[INFO] [1636642958.421146827] [rosbag2_transport]: Subscribed to topic '/imu/imu_raw'
[INFO] [1636642958.422146561] [rosbag2_transport]: Subscribed to topic '/initialpose'
[INFO] [1636642958.422888650] [rosbag2_transport]: Subscribed to topic '/planning/goal_pose'
[INFO] [1636642958.423572283] [rosbag2_transport]: Subscribed to topic '/lgsvl/state_report'
[INFO] [1636642958.424230147] [rosbag2_transport]: Subscribed to topic '/lidar_rear/points_raw'
[INFO] [1636642958.425433029] [rosbag2_transport]: Subscribed to topic '/lgsvl/gnss_odom'
[INFO] [1636642958.427590332] [rosbag2_transport]: Subscribed to topic '/lgsvl/vehicle_control_cmd'
[INFO] [1636642958.428282826] [rosbag2_transport]: Subscribed to topic '/tf_static'
^C[INFO] [1636642968.605035672] [rclcpp]: signal_handler(signal_value=2)
sc@sc-home:~/ros2_lgsvl_bag_data$ ll
total 12
drwxrwxr-x 3 sc sc 4096 Nov 12 00:02 ./
drwxr-xr-x 30 sc sc 4096 Nov 12 00:02 ../
drwxrwxr-x 2 sc sc 4096 Nov 12 00:02 rosbag2_2021_11_12-00_02_37/
sc@sc-home:~/ros2_lgsvl_bag_data$ _

```

- ROS2 환경에서 토픽 리스트와 센서 데이터를 직접 확인해볼수 있으며, 이때 발생한 데이터도 record 할 수 있습니다.



- 시각화 툴 rviz도 정상적으로 작동되는 것을 알 수 있습니다.

앞으로의 계획

- Sensor configuration - camera, radar ..
- Autoware.auto perception module test

```
sc@ade:~/AutowareAuto (master % u=)$ cd src/
sc@ade:~/AutowareAuto/src (master % u=)$ ll
total 64
drwxrwxr-x 16 sc sc 4096 Nov  2 00:09 .
drwxrwxr-x 12 sc sc 4096 Nov  7 22:37 ..
drwxrwxr-x 24 sc sc 4096 Nov  2 00:05 common/
drwxrwxr-x 16 sc sc 4096 Nov  2 00:05 control/
drwxrwxr-x 12 sc sc 4096 Nov  2 00:05 drivers/
drwxrwxr-x  2 sc sc 4096 Nov  7 17:37 external/
drwxrwxr-x  3 sc sc 4096 Nov  2 00:05 fusion/
drwxrwxr-x  5 sc sc 4096 Nov  2 00:05 launch/
drwxrwxr-x  7 sc sc 4096 Nov  2 00:05 localization/
drwxrwxr-x  5 sc sc 4096 Nov  2 00:05 mapping/
drwxrwxr-x  7 sc sc 4096 Nov  2 00:05 perception/
drwxrwxr-x 17 sc sc 4096 Nov  2 00:05 planning/
drwxrwxr-x  4 sc sc 4096 Nov  2 00:05 prediction/
drwxrwxr-x  4 sc sc 4096 Nov  2 00:05 system/
drwxrwxr-x 21 sc sc 4096 Nov  2 00:05 tools/
drwxrwxr-x  3 sc sc 4096 Nov  2 00:05 urdf/
```

- 시나리오 자동 생성 프로그램 개발

참고 사이트

<https://autowarefoundation.gitlab.io/autoware.auto/AutowareAuto/index.html>

<https://www.svlsimulator.com/docs/>