Install Autoware from Sources - release/v1.0 Branch

Created	@August 12, 2025 10:24 AM
Tags	

What changes

1. Update ansible/roles/tensorrt/tasks/main.yaml file
    a. How to do
2. Revert 896fd14 commit in autoware repositories.
3. Edit two files to update dependencies
    a. Files
        i. src/universe/autoware.universe/perception/multi_object_tracker/CMakeLists.txt
        ii. src/universe/autoware.universe/perception/multi_object_tracker/package.xml
    b. How to do
4. Create directories and symbolic links for grid_map_core eigen plugins
    a. How to do

Steps

1. Install required. modules

```
# Update and install system dependencies
sudo apt-get update -y
sudo apt-get install -y \
    build-essential \
    cmake \
    git \
    libboost-all-dev \
    libssl-dev \
    libzmq3-dev \
    pkg-config \
    python3-dev \
    python3-pip \
    python3-ven
```

2. Clone `autowarefoundation/autoware` and move to the directory.

```
# Create a workspace directory
mkdir -p $HOME/workspace
cd $HOME/workspace

# Clone the Autoware repository
git clone -b release/v1.0 https://github.com/autowarefoundat
ion/autoware.git
cd autoware
```

3. Set your git username and email address if not defined

```
# Configure git
git config --global user.email "all4dich@gmail.com"
git config --global user.name "Sunjoo Park"
```

4. Revert 896fd14 commit

    a. Some of repositories declared in autowares.repos drop 'v1.0' tag from the repository. So use commit's hash

    b. Revert 896fd14 commit

```
# Revert a specific commit
# Some of repos defined in autoware.repos remove 'v1.0' from
the repository
git revert --no-edit 896fd14
```

5. Edit Tensorrt task's dependency.

    a. Install libnvinfer-headers-dev and ibnvinfer-headers-plugin-dev as `tensorrt_version` version (defined here)

    b. If this patch may not applied, your build command tries to install these two modules as the latest version. But this action causes the problem and the process will be terminated

c. Patch Command

```
#Apply patch to the tensorrt package
patch -p1 << 'EOF'
diff --git a/ansible/roles/tensorrt/tasks/main.yaml b/ansible/roles/tensorrt/tasks/main.yaml
index df85ae7..388484b 100644
--- a/ansible/roles/tensorrt/tasks/main.yaml
+++ b/ansible/roles/tensorrt/tasks/main.yaml
@@ -20,6 +20,8 @@
       - libnvinfer-plugin-dev={{ tensorrt_version }}
       - libnvparsers-dev={{ tensorrt_version }}
       - libnvonnxparsers-dev={{ tensorrt_version }}
+      - libnvinfer-headers-dev={{ tensorrt_version }}
+      - libnvinfer-headers-plugin-dev={{ tensorrt_version }}
     allow_change_held_packages: true
     allow_downgrade: true
     update_cache: true
EOF
```

6. Run `./setup-dev-env.sh`. Use your sudo password for `BECOME password:`

```
ubuntu@yjwon-ubuntu2204:~/work/autoware$ ./setup-dev-env.sh -y
Setting up the build environment can take up to 1 hour.
>  Are you sure you want to run setup? [y/N] y
/home/ubuntu/.local/bin has been been added to PATH, but you need to open a new terminal or re-login for this
    PATH change to take effect. Alternatively, you can source your shell's config file with e.g. 'source
    ~/.bashrc'.

You will need to open a new terminal or re-login for the PATH changes to take effect. Alternatively, you can
source your shell's config file with e.g. 'source ~/.bashrc'.

Otherwise pipx is ready to go! ✨ 🌟 ✨
  installed package ansible 6.7.0, installed using Python 3.
```

```
10.12
  These apps are now globally available
    - ansible
    - ansible-community
    - ansible-config
    - ansible-connection
    - ansible-console
    - ansible-doc
    - ansible-galaxy
    - ansible-inventory
    - ansible-playbook
    - ansible-pull
    - ansible-test
    - ansible-vault
done! ✨ 🌟 ✨
ansible-galaxy collection install -f -r /home/ubuntu/work/autoware/ansible-galaxy-requirements.yaml
Starting galaxy collection install process
Process install dependency map
Starting collection install process
Installing 'autoware.dev_env:0.1.0' to '/home/ubuntu/.ansible/collections/ansible_collections/autoware/dev_env'
Created collection for autoware.dev_env:0.1.0 at /home/ubuntu/.ansible/collections/ansible_collections/autoware/dev_env
autoware.dev_env:0.1.0 was installed successfully
ansible-playbook autoware.dev_env.universe --ask-become-pass --extra-vars tensorrt_install_devel=true --extra-vars data_dir=/home/ubuntu/autoware_data --extra-vars rosdistro=humble --extra-vars rmw_implementation=rmw_cyclonedds_cpp --extra-vars base_image=ubuntu:22.04 --extra-vars cuda_base_image=ubuntu:22.04 --extra-vars prebuilt_base_image=ubuntu:22.04 --extra-vars cuda_version=12.3 --extra-vars cudnn_version=8.9.5.29-1+cuda12.2 --extra-vars tensorrt_version=8.6.1.6-1+cuda12.0
BECOME password:
```

7. Use Y for these questions.

```
[Warning] Some Autoware components depend on the CUDA, cuDNN and TensorRT NVIDIA libraries which have end-user license agreements that should be reviewed before installation.
```

```
Install NVIDIA libraries? [y/N]: y
[Warning] Should the ONNX model files and other artifacts be downloaded alongside setting up the development environment.
Download artifacts? [y/N]: y
```

8. Create `src` directory and import repos into it

```
cd autoware
mkdir src
vcs import src < autoware.repos
```

9. Install dependent ROS packages

```
source /opt/ros/humble/setup.bash
# Make sure all previously installed ros-$ROS_DISTRO-* packages are upgraded to their latest version
sudo apt update && sudo apt upgrade
rosdep update
rosdep install -y --from-paths src --ignore-src --rosdistro $ROS_DISTRO
```

   You may to input 'return' key repeatly

10. Apply patches for two files

    a. Files

       i. src/universe/autoware.universe/perception/multi_object_tracker/CMakeLists.txt

       ii. src/universe/autoware.universe/perception/multi_object_tracker/package.xml

    b. Patch Command

```
patch -p1 << 'EOF'
diff --git a/src/universe/autoware.universe/perception/multi_object_tracker/CMakeLists.txt b/perception/multi_object_tracker/CMakeLists.txt
index 3e379bcfd1..055e414790 100644
--- a/src/universe/autoware.universe/perception/multi_object_tracker/CMakeLists.txt
+++ b/src/universe/autoware.universe/perception/multi_object_tracker/CMakeLists.txt
@@ -2,6 +2,7 @@ cmake_minimum_required(VERSION 3.14)
 project(multi_object_tracker)
 find_package(autoware_cmake REQUIRED)
+ament_auto_find_build_dependencies()
 autoware_package()

 # Ignore -Wnonportable-include-path in Clang for mussp
diff --git a/src/universe/autoware.universe/perception/multi_object_tracker/package.xml b/perception/multi_object_tracker/package.xml
index e3172dfd22..f343492b2c 100644
--- a/src/universe/autoware.universe/perception/multi_object_tracker/package.xml
+++ b/src/universe/autoware.universe/perception/multi_object_tracker/package.xml
@@ -24,7 +24,7 @@
   <depend>tier4_autoware_utils</depend>
   <depend>tier4_perception_msgs</depend>
   <depend>unique_identifier_msgs</depend>
-
+  <depend>diagnostic_updater</depend>
   <test_depend>ament_lint_auto</test_depend>
   <test_depend>autoware_lint_common</test_depend>
EOF
```

11. Install grid_map packages

```
sudo apt-get install -y ros-humble-grid-map ros-humble-grid-map-core ros-humble-grid-map-cv ros-humble-grid-map-msgs ros-humble-grid-map-ros ros-humble-grid-map-rviz-plugin
```

12. Create directories and symbolic links for grid_map_core eigen plugins

```
sudo mkdir -p /opt/ros/humble/include/grid_map_core/eigen_plugins
sudo ln -s /opt/ros/humble/include/grid_map_core/grid_map_core/eigen_plugins/FunctorsPlugin.hpp /opt/ros/humble/include/grid_map_core/eigen_plugins/FunctorsPlugin.hpp || echo "INFO: Already Done"
sudo ln -s /opt/ros/humble/include/grid_map_core/grid_map_core/eigen_plugins/Functors.hpp /opt/ros/humble/include/grid_map_core/eigen_plugins/Functors.hpp || echo "INFO: Already Do
```

```
ne"
sudo ln -s /opt/ros/humble/include/grid_map_core/grid_map_co
re/eigen_plugins/DenseBasePlugin.hpp /opt/ros/humble/includ
e/grid_map_core/eigen_plugins/DenseBasePlugin.hpp || echo "I
NFO: Already Done"
```

13. Build autoware workspace

```
# Build the workspace
. /opt/ros/humble/setup.bash
cd $HOME/workspace/autoware
colcon build --symlink-install --cmake-args -DCMAKE_BUILD_TY
PE=Release
```

```
#26 645.2 Summary: 334 packages finished [10min 45s]
#26 645.2   204 packages had stderr output: accel_brake_map_
calibrator autonomous_emergency_braking autoware_ad_api_spec
s autoware_auto_common autoware_auto_geometry autoware_auto_
perception_rviz_plugin autoware_auto_tf2 autoware_point_type
s autoware_utils awapi_awiv_adapter bag_time_manager_rviz_pl
ugin behavior_path_avoidance_by_lane_change_module behavior_
path_avoidance_module behavior_path_external_request_lane_ch
ange_module behavior_path_goal_planner_module behavior_path_
lane_change_module behavior_path_planner behavior_path_plann
er_common behavior_path_side_shift_module behavior_path_star
t_planner_module behavior_velocity_crosswalk_module behavior
_velocity_planner_common bezier_sampler bluetooth_monitor bo
ost_io_context boost_serial_driver boost_tcp_driver boost_ud
p_driver bytetrack cluster_merger compare_map_segmentation c
omponent_interface_specs component_interface_utils control_p
erformance_analysis control_validator costmap_generator cros
swalk_traffic_light_estimator cuda_utils detected_object_fea
ture_remover detected_object_validation detection_by_tracker
diagnostic_converter dummy_diag_publisher dummy_infrastructu
re dummy_perception_publisher duplicated_node_checker eagley
e_can_velocity_converter eagleye_coordinate eagleye_fix2kml
eagleye_geo_pose_fusion eagleye_gnss_converter eagleye_navig
ation eagleye_rt ekf_localizer elevation_map_loader emergenc
y_handler euclidean_cluster external_cmd_converter external_
cmd_selector external_velocity_limit_selector fake_test_node
```

```
fault_injection freespace_planner freespace_planning_algorithms frenet_planner front_vehicle_velocity_estimator geography_utils glog_component gnss_poser goal_distance_calculator grid_map_utils ground_segmentation gyro_odometer heatmap_visualizer image_diagnostics image_projection_based_fusion image_transport_decompressor interpolation joy_controller kalman_filter kinematic_evaluator landmark_manager lane_departure_checker lanelet2_extension lanelet2_map_preprocessor lidar_apollo_segmentation_tvm lidar_apollo_segmentation_tvm_nodes lidar_centerpoint lidar_centerpoint_tvm livox_tag_filter llh_converter localization_error_monitor localization_evaluator localization_util map_based_prediction map_height_fitter map_loader map_projection_loader map_tf_generator mission_planner motion_utils motion_velocity_smoother mpc_lateral_controller mrm_comfortable_stop_operator mrm_emergency_stop_operator multi_object_tracker ndt_omp ndt_scan_matcher nebula_common nebula_decoders nebula_examples nebula_hw_interfaces nebula_ros object_merger object_range_splitter object_recognition_utils object_velocity_splitter objects_of_interest_marker_interface obstacle_avoidance_planner obstacle_collision_checker obstacle_cruise_planner obstacle_stop_planner obstacle_velocity_limiter occupancy_grid_map_outlier_filter osqp_interface pacmod_interface path_sampler path_smoother perception_utils pid_longitudinal_controller planning_debug_tools planning_evaluator planning_test_utils planning_topic_converter planning_validator pointcloud_preprocessor pointcloud_to_laserscan pose2twist predicted_path_checker probabilistic_occupancy_grid_map pure_pursuit qp_interface radar_crossing_objects_noise_filter radar_fusion_to_detected_object radar_object_clustering radar_object_tracker radar_scan_to_pointcloud2 radar_static_pointcloud_filter radar_threshold_filter radar_tracks_msgs_converter radar_tracks_noise_filter raw_vehicle_cmd_converter route_handler rtc_interface rtc_replayer rtklib_bridge sampler_common scenario_selector shape_estimation shift_decider signal_processing simple_object_merger simple_planning_simulator static_centerline_optimizer steer_offset_estimator stop_filter surround_obstacle_checker system_error_monitor system_monitor tensorrt_classifier tensorrt_yolo tensorrt_yolox tier4_api_utils tier4_auto_msgs_converter tier4_autoware_utils tier4_debug_rviz_plugin tier4_debug_tools tier4_logging_level_configure_rviz_plugin tier4_pcl_extensions tier4_pla
```

```
nning_rviz_plugin time_utils topic_state_monitor tracking_object_merger traffic_light_arbiter traffic_light_classifier traffic_light_fine_detector traffic_light_map_based_detector traffic_light_multi_camera_fusion traffic_light_occlusion_predictor traffic_light_ssd_fine_detector traffic_light_utils traffic_light_visualization trajectory_follower_base trajectory_follower_node tree_structured_parzen_estimator tvm_utility twist2accel vehicle_info_util vehicle_velocity_converter velodyne_monitor yabloc_common yabloc_image_processing yabloc_particle_filter yabloc_pose_initializer
#26 DONE 645.6s
```

14. Launch Autoware

```
cd $HOME/workspace/autoware
source install/setup.bash
ros2 launch autoware_launch autoware.launch.xml vehicle_model:=sample_vehicle sensor_model:=sample_sensor_kit map_path:=/home/ubuntu/workspace/Town10
```