

ROS Jazzy Jalisco

--- What's new for ROS 2 core ---

see more details to

<https://docs.ros.org/en/jazzy/Releases/Release-Jazzy-Jalisco.html>



Jazzy Documentation

Installation, Tutorials, How-to Guides and Packages are available.



Supported Platform

- Tier 1 platforms
 - **Ubuntu 24.04 Noble** (amd64 / aarch64)
 - Windows 10 (VS 2019) (amd64)
- Tier 2 platforms
 - RHEL 9 (amd64)
- Tier 3 platforms
 - macOS (amd64)
 - Debian Bookworm (amd64)

New Features

Preview RMW Zenoh

A new ROS MiddleWare (RMW) that integrates [Zenoh](#) with ROS 2 and [rmw_zenoh](#) is now available. However, it is still a preview because there are some known bugs in it, and we aren't quite ready to commit to it for the long term.

[rmw_zenoh](#) is one of `Non-DDS` RMW implementations.

Note that [rmw_zenoh](#) requires `Zenoh Router` daemon running.

To use, we need to compile [rmw_zenoh](#) from the source.



```
### Start the zenoh router
```

```
ros2 run rmw_zenoh_cpp rmw_zenoh
```

```
### Talker and Listener
```

```
RMW_IMPLEMENTATION=rmw_zenoh_cpp ros2 run demo_nodes_cpp talker
```

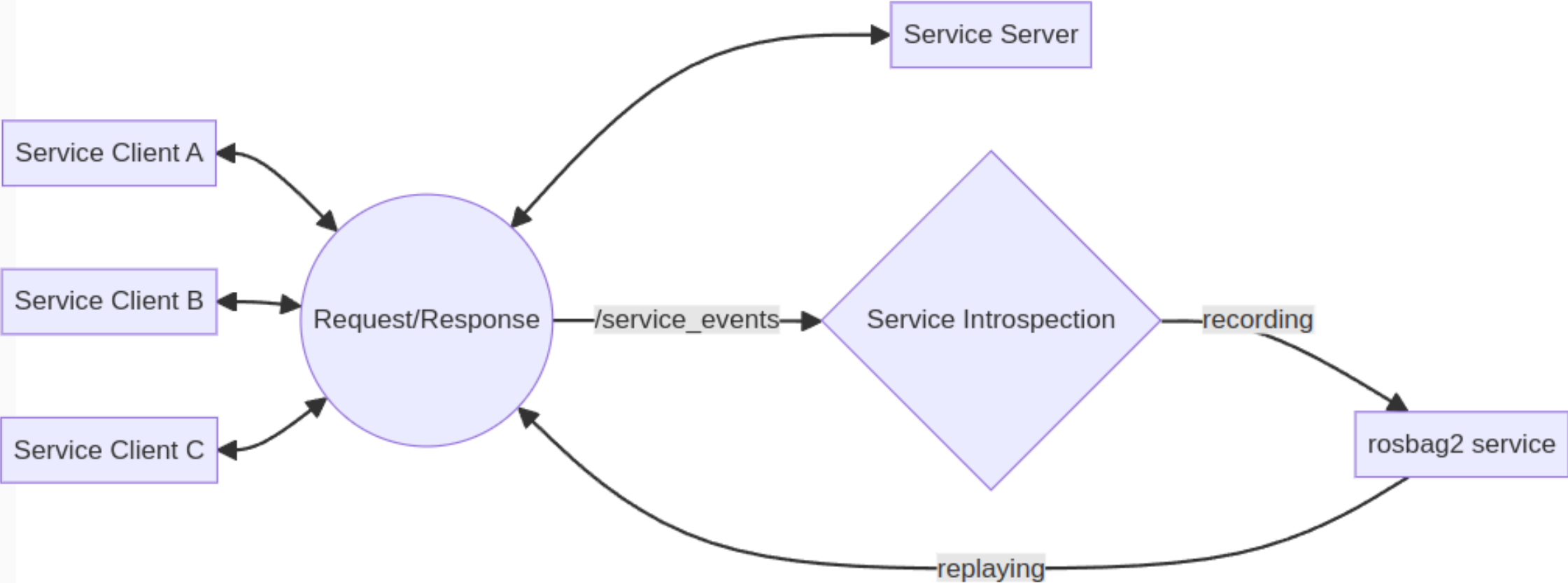
```
RMW_IMPLEMENTATION=rmw_zenoh_cpp ros2 run demo_nodes_cpp listener
```

- By default, discovery traffic is only in local host system. If network communication is needed, we need to [configure zenoh router and restart](#).
- uses CDR as the serialization format. (Compatible with DDS based RMWs)

rosvbag2 record and replay service

This feature depends on [Service introspection](#) implemented in Iron. rosvbag2 takes advantage of it to record all service requests and responses, and also replays service data from the bag file.

This feature allows you to debug/enhance/test/simulate the ROS 2 services more efficiently.



- Recording

```
# All services and all topics
ros2 bag record --all
# All topics
ros2 bag record --all-topics
# All services
ros2 bag record --all-services
```

- Playback

```
ros2 bag play --publish-service-requests bag_path
```

Please check out [Managing Service Data Tutorial](#) and [Design Document](#) for more information.

Core Feature Development

- [message_filters](#)
 - Now we can use `TypeAdapter` to the subscription, so that user defined message type can be declared and converted by user callback for the filtering process.
- [rcl](#)
 - `~/get_type_description` service is added to support upcoming feature [Evolving Message Type](#).

- `rclcpp`
 - `get_service_typesupport_handle` API is added to resolve service type at runtime. this is required by rosbag2 service playback which requires dynamic service type resolving.
- `ros2cli`
 - Now you can specify the log file name prefix with `--ros-args --log-file-name filename`. This is useful when you run with `ros2 run` because prefix is `python` by default.

- `rosbag2`
 - `Player` and `Recorder` are now exposed as `rclcpp` components. this brings the capability that you can record and play via intra-process communication with high performance.
 - Compression threads priority control support.
 - Split bag files during conversion.
 - bag files self-contain metadata. this means each bag file is self-contained, ready to play on anywhere.

- `ros2action`
 - `type` sub-command is now supported to look up the action type.

```
ros2 action type /fibonacci  
action_tutorials_interfaces/action/Fibonacci
```

Core Updates from Iron Irwini

- `rcl`
 - `rcl_timer_call_with_info` is added to allow user application know expected timer call time and when actual timer is called. e.g user application can know the jitter and delay for the timer.
 - `rcl_wait improvement` to avoid spurious wake-ups.
- `rclcpp`
 - Data Racy Condition Improvement.
 - Enhancement `rclcpp::WaitSet`.
 - `rclcpp::get_typesupport_handle` is deprecated.
 - Actual and expected call time when timer is called.
 - Entity Execution Order is NOT predictable (W.I.P).

- `rclcpp_action`
 - `Safely release goal handle`, to drop the handle in a locked context.
- `rclpy`
 - `forbid parameter to be declared statically without initialization`.
 - Several `TypeDescriptionsInterface` enhancement for `Evolving Message Type`.

- **rosvbag2**
 - **--exclude CLI option was renamed to the --exclude-regex** (part of rosvbag2 service development).
 - **Add node name to the Read(Write)SplitEvent message** to support multiple rosvbag2 instance.
 - **Signal Handling Enhancement.**
 - **Add topic_id returned by storage to the TopicMetadata**, performance improvement with id but topic names via storage backend and tell multiple topics with id.
 - **Improve performance in SqliteStorage::get_bagfile_size()**

- `ros2_tracing`
 - publication and subscription end to end tracing is agnostic from middleware details, no source build is required anymore
 - Generic publisher and subscription tracing supported
 - New `ros2 trace start/pause/resume/stop` sub-commands are supported.

```
$ ros2 trace start session_name # Configure tracing session and start tracing
$ ros2 trace pause session_name # Pause tracing after starting
$ ros2 trace resume session_name # Resume tracing after pausing
$ ros2 trace stop session_name # Stop tracing after starting or resuming
```

see more details for https://github.com/ros2/ros2_tracing/pull/70 and [readme](#)

See you at **ROSCon 2024**

Odense, Denmark

October 21st - 23rd, 2024

