

1. Lifecycle Node 란

Lifecycle node 란 노드의 상태를 명확하게 정의하고 관리하는 기능을 제공하는 것을 말한다. 일반적인 ROS2 노드는 시작하면 바로 동작하지만 lifecycle node 는 개발자가 정의한 여러 states 와 transitions 를 거치며 실행된다.

Lifecycle node 는 다음과 같이 4 가지 주요 상태를 가지는데

- 1) Unfigured: 노드가 막 생성된 초기 상태, 아무런 리소스도 할당되지 않은 상태
- 2) Inactive: 노드가 설정되어 필요한 리소스를 할당받은 상태, 실제 작업 수행 x
- 3) Active: 노드가 모든 기능을 정상적으로 수행하는 상태, 콜백 함수가 호출되고 발행가 메시리를 보냄
- 4) Finalized: 노드가 종료된 상태

이렇게 총 4 가지이며 상태 간의 이동은 다음 4 가지의 과정을 사이클하며 이루어진다.

2. QoS 란

ROS2 는 실시간 데이터 전송을 보장하고, DDS 라는 것의 사용으로 노드 간의 동적 검색 기능을 지원하고 있어서 여러 프로그램 간에 통신이 가능하다. QoS 란 DDS 의 서비스 품질로, 데이터 통신의 옵션이라고 할 수 있다. ROS2 는 신뢰성을 중시하는 TCP 통신과 속도를 중시하는 UDP 통신을 선택적으로 사용할 수 있는데, 이를 위해 DDS 의 QoS 를 도입하여 publisher 와 subscriber 를 선언할 때 QoS 를 매개변수 형태로 지정할 수 있어서 원하는 데이터 통신의 옵션 설정을 유저가 직접할 수 있다. QoS 는 다음과 같은 종류가 있다.

1} Reliability

RELIABLE : TCP 처럼 데이터 손실 방지하고 신뢰도 우선

BEST_EFFORT : UDP 처럼 통신 속도를 최우선시

2) History

통신 상태에 따라 정해진 사이즈만큼의 데이터 보관

KEEP_LAST : 정해진 메시지 큐 사이즈 만큼의 데이터 보관 (depth : 메시지 큐 사이즈)

KEEP_ALL : 모든 데이터를 보관 (메시지 큐의 사이즈는 DDS 벤더마다 다름)

3) Durability

데이터 수신하는 Subscriber 가 생성되기 전의 데이터를 폐기할지에 대한 설정

TRANSIENT_LOCAL : Subscription 이 생성되기 전의 데이터도 보관 (Publisher 에 가능)

VOLATILE : Subscription 이 생성되기 전의 데이터는 무효

4) Deadline

정해진 주기 안에 데이터가 발신 및 수신되지 않을 경우 이벤트 함수를 실행

deadline_duration : Deadline 을 확인하는 주기

5) Lifespan

정해진 주기 안에서 수신되는 데이터만 유효 판정

그렇지 않은 데이터는 삭제

lifespan_duration : Lifespan 을 확인하는 주기

6) Liveliness

정해진 주기 안에서 Node, Topic 생사 확인

liveliness : 자동/매뉴얼로 확인할지를 지정하는 옵션

AUTOMATIC / MANUAL_BY_NODE, MANUAL_BY_TOPIC

이 중에서 과제 1 번에서 구현해야 하는 것은 reliability 와 durability 이며 reliability 의 종류에는 메시지를 보내지만 수신 여부를 체크 하지 않는 best_effort 와 반드시 메시지를 도착시키는 reliable 이 있다. Durability 에는 메시지를 실시간으로만 보내서 새로운 구독자가 붙으면 과거 메시지는 받지 못하고 그 이후의 메시지부터 받는 volatile 과 퍼블리셔가 마지막 메시지를 DDS 에 보관했다가 새 구독자가 붙으면 마지막으로 퍼블리시된 메시지를 즉시 전달받고 그 이후부터 계속 수신 가능한 transient_local 이 있다.

3. Lifecycle 예제 학습

```
jy@jy: ~/ros2_ws
jy@jy:~/ros2_ws$ 169x47
[lifecycle_talker-1] [INFO] [1758027154.475362403] [lc_talker]: on_deactivate() is called.
[lifecycle_listener-2] [INFO] [1758027154.475474493] [listener]: notify callback: Transition from state active to deactivating
[lifecycle_listener-2] [INFO] [1758027154.475525699] [listener]: notify callback: Transition from state deactivating to inactive
[lifecycle_listener-2] [INFO] [1758027154.477110414] [listener]: notify callback: Transition from state inactive to cleaningup
[lifecycle_talker-1] [INFO] [1758027154.477444400] [lc_talker]: on_cleanup is called.
[lifecycle_listener-2] [INFO] [1758027154.477574094] [listener]: notify callback: Transition from state cleaningup to unconfigured
[lifecycle_talker-1] [INFO] [1758027154.478255322] [lc_talker]: on_shutdown is called from state unconfigured.
[lifecycle_listener-2] [INFO] [1758027154.478312740] [listener]: notify callback: Transition from state unconfigured to shuttingdown
[lifecycle_listener-2] [INFO] [1758027154.478349599] [listener]: notify callback: Transition from state shuttingdown to finalized
^C[WARNING] [launch]: user interrupted with ctrl-c (SIGINT)
[lifecycle_listener-2] [INFO] [1758027344.564525318] [rclcpp]: signal_handler(signum=2)
[INFO] [lifecycle_talker-1]: process has finished cleanly [pid 15945]
[INFO] [lifecycle_listener-2]: process has finished cleanly [pid 15947]
[lifecycle_talker-1] [INFO] [1758027344.564525308] [rclcpp]: signal_handler(signum=2)
jy@jy:~/ros2_ws$ ros2 launch lifecycle_node test_lifecycle.py
[INFO] [launch]: All log files can be found below /home/jy/.ros/log/2025-09-16-21-55-51-461125-jy-16026
[INFO] [launch]: Default logging verbosity is set to INFO
[INFO] [lifecycle_talker-1]: process started with pid [16037]
[INFO] [lifecycle_listener-2]: process started with pid [16039]
[lifecycle_listener-2] [INFO] [1758027351.779519371] [listener]: notify callback: Transition from state unconfigured to configuring
[lifecycle_talker-1] [INFO] [1758027351.780182855] [lc_talker]: on_configure() is called.
[lifecycle_listener-2] [INFO] [1758027351.780370761] [listener]: notify callback: Transition from state configuring to inactive
[lifecycle_talker-1] [INFO] [1758027351.781534417] [lc_talker]: on_activate() is called.
[lifecycle_listener-2] [INFO] [1758027351.781597633] [listener]: notify callback: Transition from state inactive to activating
[lifecycle_talker-1] [INFO] [1758027353.781932443] [lc_talker]: Lifecycle publisher is active, Publishing: [Lifecycle HelloWorld #1]
[lifecycle_listener-2] [INFO] [1758027353.782096064] [listener]: notify callback: Transition from state activating to active
[lifecycle_listener-2] [INFO] [1758027353.782284351] [listener]: data_callback: Lifecycle HelloWorld #1
[lifecycle_talker-1] [INFO] [1758027354.780344905] [lc_talker]: Lifecycle publisher is active, Publishing: [Lifecycle HelloWorld #2]
[lifecycle_listener-2] [INFO] [1758027354.780819080] [listener]: data_callback: Lifecycle HelloWorld #2
[lifecycle_talker-1] [INFO] [1758027355.780210477] [lc_talker]: Lifecycle publisher is active, Publishing: [Lifecycle HelloWorld #3]
[lifecycle_listener-2] [INFO] [1758027355.780548100] [listener]: data_callback: Lifecycle HelloWorld #3
[lifecycle_talker-1] [INFO] [1758027356.780179979] [lc_talker]: Lifecycle publisher is active, Publishing: [Lifecycle HelloWorld #4]
[lifecycle_listener-2] [INFO] [1758027356.780676585] [listener]: data_callback: Lifecycle HelloWorld #4
[lifecycle_talker-1] [INFO] [1758027357.780053376] [lc_talker]: Lifecycle publisher is active, Publishing: [Lifecycle HelloWorld #5]
[lifecycle_listener-2] [INFO] [1758027357.780480444] [listener]: data_callback: Lifecycle HelloWorld #5
[lifecycle_talker-1] [INFO] [1758027358.780131360] [lc_talker]: Lifecycle publisher is active, Publishing: [Lifecycle HelloWorld #6]
[lifecycle_listener-2] [INFO] [1758027358.780498086] [listener]: data_callback: Lifecycle HelloWorld #6
[lifecycle_talker-1] [INFO] [1758027358.785386522] [lc_talker]: on_deactivate() is called.
[lifecycle_listener-2] [INFO] [1758027358.785577374] [listener]: notify callback: Transition from state active to deactivating
[lifecycle_listener-2] [INFO] [1758027358.785665135] [listener]: notify callback: Transition from state deactivating to inactive
[lifecycle_listener-2] [INFO] [1758027358.787148141] [listener]: notify callback: Transition from state inactive to cleaningup
[lifecycle_talker-1] [INFO] [1758027358.787398703] [lc_talker]: on_cleanup is called.
[lifecycle_listener-2] [INFO] [1758027358.787478079] [listener]: notify callback: Transition from state cleaningup to unconfigured
[lifecycle_talker-1] [INFO] [1758027358.788281571] [lc_talker]: on_shutdown is called from state unconfigured.
[lifecycle_listener-2] [INFO] [1758027358.788337003] [listener]: notify callback: Transition from state unconfigured to shuttingdown
[lifecycle_listener-2] [INFO] [1758027358.788370084] [listener]: notify callback: Transition from state shuttingdown to finalized
```

우선 첫번째 ROS2 공식 홈페이지에 나온 공식 코드를 사용하여 lifecycle_node 를 구현한 결과이다. 공식 홈페이지에서 주어진 자동전환 이벤트가 포함되어 있는 python 런치파일을 사용하였다. 로그 단계별로 우선 노드가 처음 시작하면 unconfigured 상태인데 런치 파일에서 event handler 가 unconfigured->configuring 으로 전환하여 on configure()함수를 콜백하면 publisher 나 subscription 같은 리소스를 준비한다. 그러면 상태는 configuring->inactive->activating->activate 로 전이되고 매초 메시지가 발행되고 listener 가 수신한다. 5 초간 메시지를 발행하면 active->deactivate->inactive 과정을 거치는데 메시지 발행이 중단된다. 비활성화 상태에서는 Inactive->cleaningup->unconfigured 순으로 리소스를 해제한다. 그리고 unconfigured->shutdown-finalized 순으로 프로세스가 종료된다.

4. QoS 예제 학습

두번째, 공식 홈에서 주어진 QoS 예제를 실행한 결과이다. Reliability 정책의 옵션 두개를 각각 다르게 설정하고 실행, 같게 설정하고 실행 후 차이를 확인하였다.

```
jy@jy: ~/ros2_ws
jy@jy: ~/ros2_ws 80x24
jy@jy:~$ cd ros2_ws
jy@jy:~/ros2_ws$ colcon build --packages-select qos_tests_cpp_pkg
Starting >>> qos_tests_cpp_pkg
Finished <<< qos_tests_cpp_pkg [4.79s]

Summary: 1 package finished [4.92s]
jy@jy:~/ros2_ws$ source install/setup.bash
jy@jy:~/ros2_ws$ ros2 run qos_tests_cpp_pkg subscriber_custom_minimal_qos
[WARN] [1758073786.472555574] [subscriber_qos_obj]: New publisher discovered on
topic 'qos_test', offering incompatible QoS. No messages will be sent to it.
Last incompatible policy: RELIABILITY_QOS_POLICY
[INFO] [1758073787.472737725] [publisher_qos_obj]: Publishing: 0:1758073787.4726
67,1758073787472666561
[INFO] [1758073788.472671382] [publisher_qos_obj]: Publishing: 1:1758073788.4726
38,1758073788472638150
[INFO] [1758073789.472885021] [publisher_qos_obj]: Publishing: 2:1758073789.4728
21,1758073789472821341
[INFO] [1758073790.473119746] [publisher_qos_obj]: Publishing: 3:1758073790.4730
43,1758073790473042581
[INFO] [1758073791.473080435] [publisher_qos_obj]: Publishing: 4:1758073791.4730
23,1758073791473023428
[INFO] [1758073792.473377852] [publisher_qos_obj]: Publishing: 5:1758073792.4733
01,1758073792473301548
[INFO] [1758073793.473193400] [publisher_qos_obj]: Publishing: 6:1758073793.4731
38,1758073793473138387
[INFO] [1758073794.473190406] [publisher_qos_obj]: Publishing: 7:1758073794.4731
14,1758073794473114093
[INFO] [1758073795.473237558] [publisher_qos_obj]: Publishing: 8:1758073795.4731
61,1758073795473161164
[INFO] [1758073796.473164777] [publisher_qos_obj]: Publishing: 9:1758073796.4730
89,1758073796473089045
```

사진과 같이 publisher 와 subscriber 가 서로를 발견 하였지만 QoS 가 맞지 않아 통신이 되지 않고 있다. 그러나 정책을 두 노드 모두 reliable 로 맞추어 주면 다음 사진과 같이 정상적으로 통신이 되는 것을 알 수 있다.

```
jy@jy: ~/ros2_ws
jy@jy: ~/ros2_ws 80x31
jy@jy:~$ cd ros2_ws
jy@jy:~/ros2_ws$ source install/setup.bash
jy@jy:~/ros2_ws$ ros2 run qos_tests_cpp_pkg publisher_custom_minimal_qos
[INFO] [1758073118.069704299] [publisher_qos_obj]: Publishing: 0:1758073118.0695
32,1758073118069531744
[INFO] [1758073119.069745182] [publisher_qos_obj]: Publishing: 1:1758073119.0696
29,1758073119069628970
[INFO] [1758073120.069266255] [publisher_qos_obj]: Publishing: 2:1758073120.0691
38,1758073120069137792
[INFO] [1758073121.069063545] [publisher_qos_obj]: Publishing: 3:1758073121.0689
97,1758073121068996093
[INFO] [1758073122.069014250] [publisher_qos_obj]: Publishing: 4:1758073122.0689
51,1758073122068951075
[INFO] [1758073123.069081880] [publisher_qos_obj]: Publishing: 5:1758073123.0689
65,1758073123068964516
[INFO] [1758073124.068979512] [publisher_qos_obj]: Publishing: 6:1758073124.0688
73,1758073124068872738
[INFO] [1758073125.068900385] [publisher_qos_obj]: Publishing: 7:1758073125.0688
32,1758073125068831941
[INFO] [1758073126.069007534] [publisher_qos_obj]: Publishing: 8:1758073126.0687
79,1758073126068778507
[INFO] [1758073127.068991946] [publisher_qos_obj]: Publishing: 9:1758073127.0688
48,1758073127068847924
[INFO] [1758073128.068728745] [publisher_qos_obj]: Publishing: 10:1758073128.068
678,1758073128068678403
[INFO] [1758073129.068858142] [publisher_qos_obj]: Publishing: 11:1758073129.068
697,1758073129068697369
[INFO] [1758073130.068730553] [publisher_qos_obj]: Publishing: 12:1758073130.068
628,1758073130068627936
[INFO] [1758073131.069109019] [publisher_qos_obj]: Publishing: 13:1758073131.068
975,1758073131068974915
[INFO] [1758073132.068567462] [publisher_qos_obj]: Publishing: 14:1758073132.068

jy@jy: ~/ros2_ws
jy@jy: ~/ros2_ws 80x30
Summary: 1 package finished [5.03s]
jy@jy:~/ros2_ws$ source install/setup.bash
jy@jy:~/ros2_ws$ ros2 run qos_tests_cpp_pkg subscriber_custom_minimal_qos
[INFO] [1758073118.070069583] [subscriber_qos_obj]: Data Received = '0:175807311
8.069532,1758073118069531744'
[INFO] [1758073119.069979458] [subscriber_qos_obj]: Data Received = '1:175807311
9.069629,1758073119069628970'
[INFO] [1758073120.069471058] [subscriber_qos_obj]: Data Received = '2:175807312
0.069138,1758073120069137792'
[INFO] [1758073121.069146636] [subscriber_qos_obj]: Data Received = '3:175807312
1.068997,1758073121068996093'
[INFO] [1758073122.069148945] [subscriber_qos_obj]: Data Received = '4:175807312
2.068951,1758073122068951075'
[INFO] [1758073123.069331544] [subscriber_qos_obj]: Data Received = '5:175807312
3.068965,1758073123068964516'
[INFO] [1758073124.069096265] [subscriber_qos_obj]: Data Received = '6:175807312
4.068873,1758073124068872738'
[INFO] [1758073125.069035531] [subscriber_qos_obj]: Data Received = '7:175807312
5.068832,1758073125068831941'
[INFO] [1758073126.069263621] [subscriber_qos_obj]: Data Received = '8:175807312
6.068779,1758073126068778507'
[INFO] [1758073127.069239156] [subscriber_qos_obj]: Data Received = '9:175807312
7.068848,1758073127068847924'
[INFO] [1758073128.068820943] [subscriber_qos_obj]: Data Received = '10:17580731
28.068678,1758073128068678403'
[INFO] [1758073129.069145015] [subscriber_qos_obj]: Data Received = '11:17580731
29.068697,1758073129068697369'
[INFO] [1758073130.069047599] [subscriber_qos_obj]: Data Received = '12:17580731
30.068628,1758073130068627936'
[INFO] [1758073131.069399038] [subscriber_qos_obj]: Data Received = '13:17580731
```

5. lifecycle node 에 QoS 설정

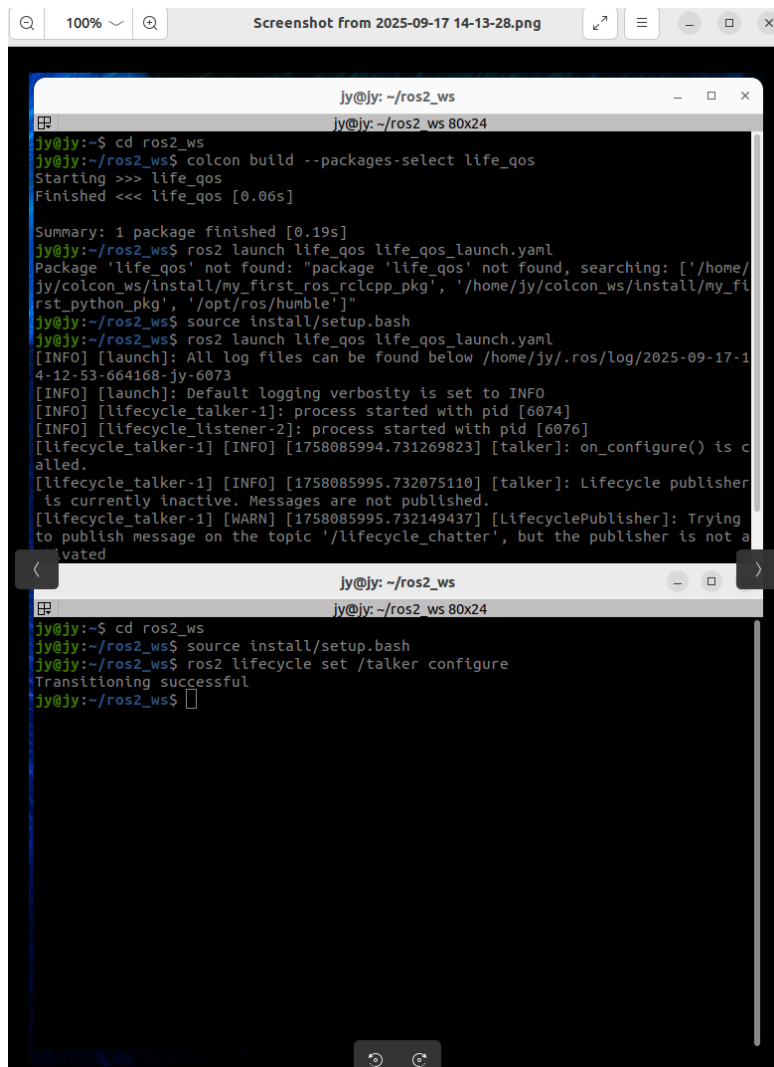
ROS2 에서 퍼블리셔는 생성될 때 QOS 설정을 DDS 에 등록한다.

```
pub_ = this->create_publisher<std_msgs::msg::String>("topic", qos);
```

위의 명령어에서 qos 객체는 reliability, durability 정책을 담고 있기에 on_configure() 안에서 퍼블리셔를 만들고 qos 를 지정한다. 마찬가지로 subscriber 노드는 생성 부분 바로 아래에 qos 설정을 진행해 준다.

다음은 실행 결과 및 사진이다.

- 1) Launch 파일을 실행하여 ros2 lifecycle set /talker configure 세팅을 해준다. 그러면 inactive 상태라는 info 가 수신창에 계속 뜨게 된다.

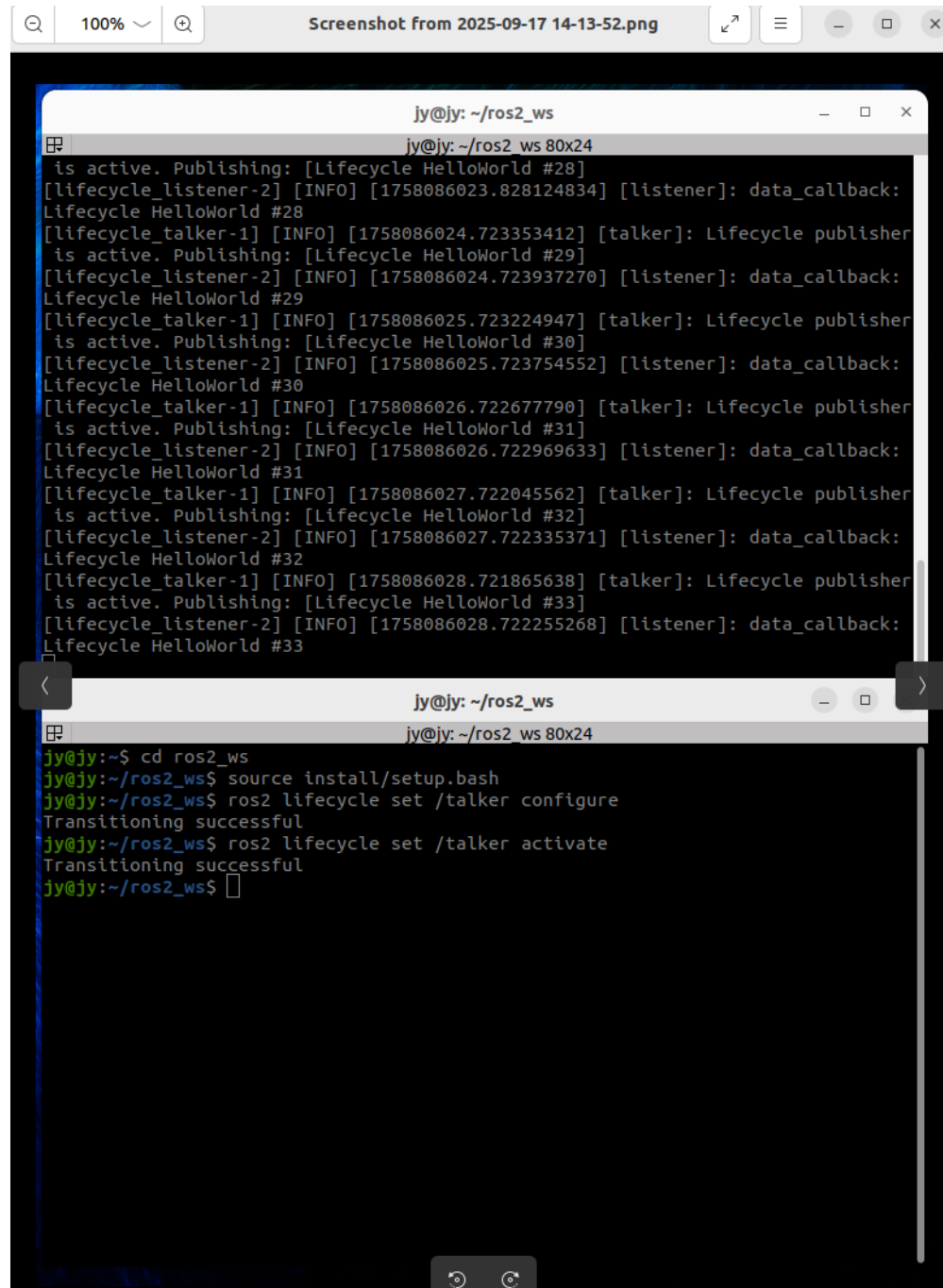


```
jy@jy: ~/ros2_ws
jy@jy:~/ros2_ws$ cd ros2_ws
jy@jy:~/ros2_ws$ colcon build --packages-select life_qos
Starting >>> life_qos
Finished <<< life_qos [0.06s]

Summary: 1 package finished [0.19s]
jy@jy:~/ros2_ws$ ros2 launch life_qos life_qos_launch.yaml
Package 'life_qos' not found: "package 'life_qos' not found, searching: ['/home/
jy/colcon_ws/install/my_first_ros_rclcpp_pkg', '/home/jy/colcon_ws/install/my_fi
rst_python_pkg', '/opt/ros/humble']"
jy@jy:~/ros2_ws$ source install/setup.bash
jy@jy:~/ros2_ws$ ros2 launch life_qos life_qos_launch.yaml
[INFO] [launch]: All log files can be found below /home/jy/.ros/log/2025-09-17-1
4-12-53-664168-jy-6073
[INFO] [launch]: Default logging verbosity is set to INFO
[INFO] [lifecycle_talker-1]: process started with pid [6074]
[INFO] [lifecycle_listener-2]: process started with pid [6076]
[lifecycle_talker-1] [INFO] [1758085994.731269823] [talker]: on_configure() is c
alled.
[lifecycle_talker-1] [INFO] [1758085995.732075110] [talker]: Lifecycle publisher
is currently inactive. Messages are not published.
[lifecycle_talker-1] [WARN] [1758085995.732149437] [LifecyclePublisher]: Trying
to publish message on the topic '/lifecycle_chatter', but the publisher is not a
vated

jy@jy: ~/ros2_ws
jy@jy:~/ros2_ws$ cd ros2_ws
jy@jy:~/ros2_ws$ source install/setup.bash
jy@jy:~/ros2_ws$ ros2 lifecycle set /talker configure
Transitioning successful
jy@jy:~/ros2_ws$
```

- 2) `ros2 lifecycle set /talker activate` 설정을 하면 `activate` 상태로 넘어가게 된다.
그러면 수신창에 HelloWorld 가 계속 출력된다.

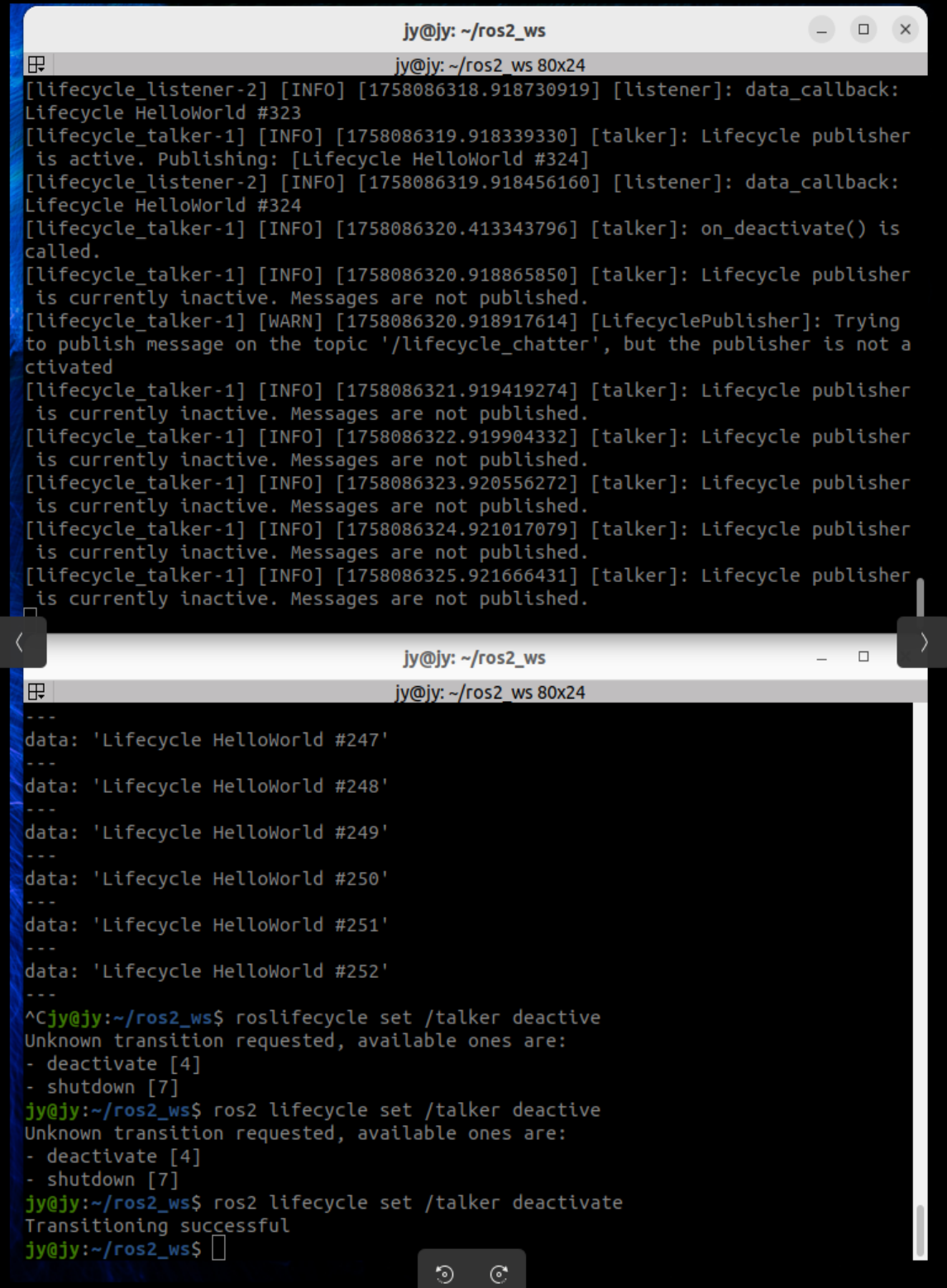


The screenshot shows two terminal windows. The top window displays the output of a ROS2 lifecycle node, showing a sequence of 'HelloWorld' messages and lifecycle state transitions. The bottom window shows the commands used to set the lifecycle state to 'activate'.

```
jy@jy: ~/ros2_ws
jy@jy:~/ros2_ws 80x24
[is active. Publishing: [Lifecycle HelloWorld #28]
[lifecycle_listener-2] [INFO] [1758086023.828124834] [listener]: data_callback:
Lifecycle HelloWorld #28
[lifecycle_talker-1] [INFO] [1758086024.723353412] [talker]: Lifecycle publisher
is active. Publishing: [Lifecycle HelloWorld #29]
[lifecycle_listener-2] [INFO] [1758086024.723937270] [listener]: data_callback:
Lifecycle HelloWorld #29
[lifecycle_talker-1] [INFO] [1758086025.723224947] [talker]: Lifecycle publisher
is active. Publishing: [Lifecycle HelloWorld #30]
[lifecycle_listener-2] [INFO] [1758086025.723754552] [listener]: data_callback:
Lifecycle HelloWorld #30
[lifecycle_talker-1] [INFO] [1758086026.722677790] [talker]: Lifecycle publisher
is active. Publishing: [Lifecycle HelloWorld #31]
[lifecycle_listener-2] [INFO] [1758086026.722969633] [listener]: data_callback:
Lifecycle HelloWorld #31
[lifecycle_talker-1] [INFO] [1758086027.722045562] [talker]: Lifecycle publisher
is active. Publishing: [Lifecycle HelloWorld #32]
[lifecycle_listener-2] [INFO] [1758086027.722335371] [listener]: data_callback:
Lifecycle HelloWorld #32
[lifecycle_talker-1] [INFO] [1758086028.721865638] [talker]: Lifecycle publisher
is active. Publishing: [Lifecycle HelloWorld #33]
[lifecycle_listener-2] [INFO] [1758086028.722255268] [listener]: data_callback:
Lifecycle HelloWorld #33

jy@jy: ~/ros2_ws
jy@jy:~/ros2_ws 80x24
jy@jy:~$ cd ros2_ws
jy@jy:~/ros2_ws$ source install/setup.bash
jy@jy:~/ros2_ws$ ros2 lifecycle set /talker configure
Transitioning successful
jy@jy:~/ros2_ws$ ros2 lifecycle set /talker activate
Transitioning successful
jy@jy:~/ros2_ws$
```

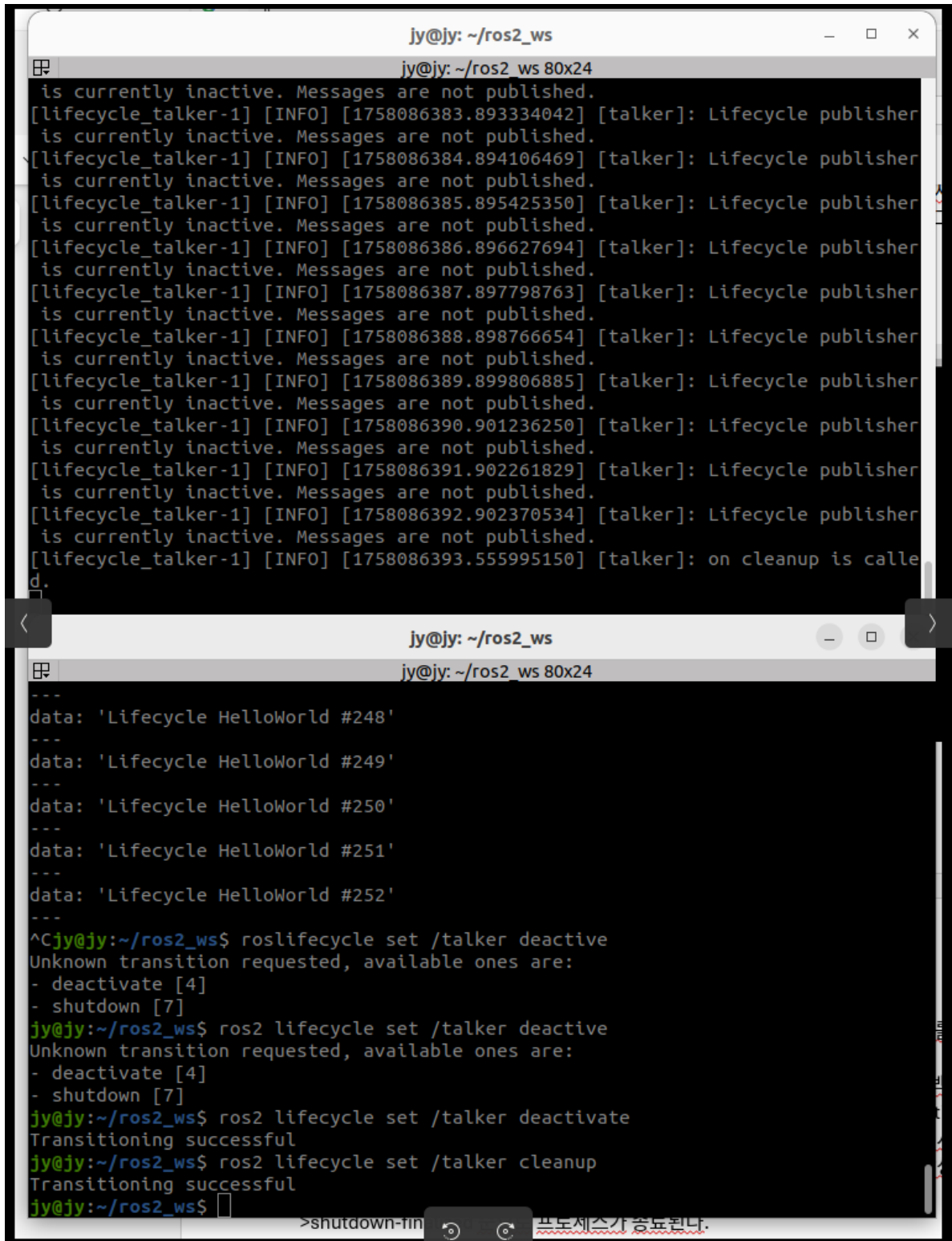
- 3) `ros2 lifecycle set /talker deactivate` 설정을 해주면 다음 사진과 같이 active 상태에서 inactive 상태로 바뀌는 것을 알 수 있다.



```
jy@jy: ~/ros2_ws
jy@jy: ~/ros2_ws 80x24
[lifecycle_listener-2] [INFO] [1758086318.918730919] [listener]: data_callback:
Lifecycle HelloWorld #323
[lifecycle_talker-1] [INFO] [1758086319.918339330] [talker]: Lifecycle publisher
is active. Publishing: [Lifecycle HelloWorld #324]
[lifecycle_listener-2] [INFO] [1758086319.918456160] [listener]: data_callback:
Lifecycle HelloWorld #324
[lifecycle_talker-1] [INFO] [1758086320.413343796] [talker]: on_deactivate() is
called.
[lifecycle_talker-1] [INFO] [1758086320.918865850] [talker]: Lifecycle publisher
is currently inactive. Messages are not published.
[lifecycle_talker-1] [WARN] [1758086320.918917614] [LifecyclePublisher]: Trying
to publish message on the topic '/lifecycle_chatter', but the publisher is not a
ctivated
[lifecycle_talker-1] [INFO] [1758086321.919419274] [talker]: Lifecycle publisher
is currently inactive. Messages are not published.
[lifecycle_talker-1] [INFO] [1758086322.919904332] [talker]: Lifecycle publisher
is currently inactive. Messages are not published.
[lifecycle_talker-1] [INFO] [1758086323.920556272] [talker]: Lifecycle publisher
is currently inactive. Messages are not published.
[lifecycle_talker-1] [INFO] [1758086324.921017079] [talker]: Lifecycle publisher
is currently inactive. Messages are not published.
[lifecycle_talker-1] [INFO] [1758086325.921666431] [talker]: Lifecycle publisher
is currently inactive. Messages are not published.

jy@jy: ~/ros2_ws
jy@jy: ~/ros2_ws 80x24
---
data: 'Lifecycle HelloWorld #247'
---
data: 'Lifecycle HelloWorld #248'
---
data: 'Lifecycle HelloWorld #249'
---
data: 'Lifecycle HelloWorld #250'
---
data: 'Lifecycle HelloWorld #251'
---
data: 'Lifecycle HelloWorld #252'
---
^Cjy@jy:~/ros2_ws$ roslifecycle set /talker deactivate
Unknown transition requested, available ones are:
- deactivate [4]
- shutdown [7]
jy@jy:~/ros2_ws$ ros2 lifecycle set /talker deactivate
Unknown transition requested, available ones are:
- deactivate [4]
- shutdown [7]
jy@jy:~/ros2_ws$ ros2 lifecycle set /talker deactivate
Transitioning successful
jy@jy:~/ros2_ws$
```

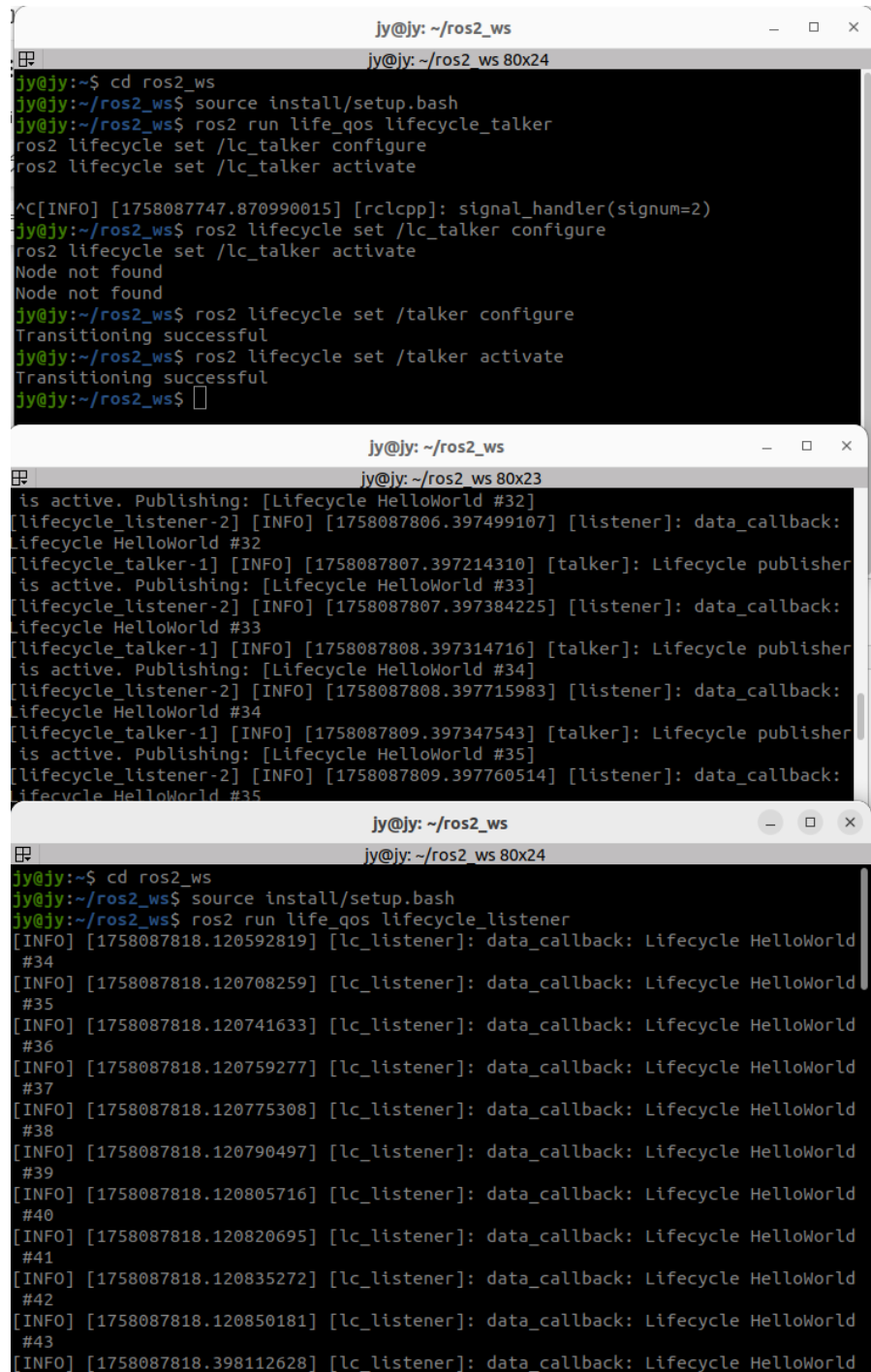

- 4) 그리고 마지막으로 `ros2 lifecycle set /talker cleanup` 설정을 해주면 다시 unconfigured 상태로 돌아오게 된다.



```
jy@jy: ~/ros2_ws
jy@jy: ~/ros2_ws 80x24
is currently inactive. Messages are not published.
[lifecycle_talker-1] [INFO] [1758086383.893334042] [talker]: Lifecycle publisher
is currently inactive. Messages are not published.
[lifecycle_talker-1] [INFO] [1758086384.894106469] [talker]: Lifecycle publisher
is currently inactive. Messages are not published.
[lifecycle_talker-1] [INFO] [1758086385.895425350] [talker]: Lifecycle publisher
is currently inactive. Messages are not published.
[lifecycle_talker-1] [INFO] [1758086386.896627694] [talker]: Lifecycle publisher
is currently inactive. Messages are not published.
[lifecycle_talker-1] [INFO] [1758086387.897798763] [talker]: Lifecycle publisher
is currently inactive. Messages are not published.
[lifecycle_talker-1] [INFO] [1758086388.898766654] [talker]: Lifecycle publisher
is currently inactive. Messages are not published.
[lifecycle_talker-1] [INFO] [1758086389.899806885] [talker]: Lifecycle publisher
is currently inactive. Messages are not published.
[lifecycle_talker-1] [INFO] [1758086390.901236250] [talker]: Lifecycle publisher
is currently inactive. Messages are not published.
[lifecycle_talker-1] [INFO] [1758086391.902261829] [talker]: Lifecycle publisher
is currently inactive. Messages are not published.
[lifecycle_talker-1] [INFO] [1758086392.902370534] [talker]: Lifecycle publisher
is currently inactive. Messages are not published.
[lifecycle_talker-1] [INFO] [1758086393.555995150] [talker]: on cleanup is called.

jy@jy: ~/ros2_ws
jy@jy: ~/ros2_ws 80x24
---
data: 'Lifecycle HelloWorld #248'
---
data: 'Lifecycle HelloWorld #249'
---
data: 'Lifecycle HelloWorld #250'
---
data: 'Lifecycle HelloWorld #251'
---
data: 'Lifecycle HelloWorld #252'
---
^Cjy@jy:~/ros2_ws$ roslifecycle set /talker deactivate
Unknown transition requested, available ones are:
- deactivate [4]
- shutdown [7]
jy@jy:~/ros2_ws$ ros2 lifecycle set /talker deactivate
Unknown transition requested, available ones are:
- deactivate [4]
- shutdown [7]
jy@jy:~/ros2_ws$ ros2 lifecycle set /talker deactivate
Transitioning successful
jy@jy:~/ros2_ws$ ros2 lifecycle set /talker cleanup
Transitioning successful
jy@jy:~/ros2_ws$
```

- 5) 마지막으로 QoS 설정이 되어 있어서 reliable 로 인해 무조건 지연이 발생해도 모든 메시지를 수신 받으며 transient_local 이 설정되어 있어서 늦게 실행된 구독자가 있어도 가장 최근 값 하나를 출력하고 나서 계속하여 수신 받는 것을 볼 수 있다.



```
jy@jy: ~/ros2_ws
jy@jy: ~/ros2_ws 80x24
jy@jy:~$ cd ros2_ws
jy@jy:~/ros2_ws$ source install/setup.bash
jy@jy:~/ros2_ws$ ros2 run life_qos lifecycle_talker
ros2 lifecycle set /lc_talker configure
ros2 lifecycle set /lc_talker activate

^C[INFO] [1758087747.870990015] [rclcpp]: signal_handler(signum=2)
jy@jy:~/ros2_ws$ ros2 lifecycle set /lc_talker configure
ros2 lifecycle set /lc_talker activate
Node not found
Node not found
jy@jy:~/ros2_ws$ ros2 lifecycle set /talker configure
Transitioning successful
jy@jy:~/ros2_ws$ ros2 lifecycle set /talker activate
Transitioning successful
jy@jy:~/ros2_ws$

jy@jy: ~/ros2_ws
jy@jy: ~/ros2_ws 80x23
is active. Publishing: [Lifecycle HelloWorld #32]
[lifecycle_listener-2] [INFO] [1758087806.397499107] [listener]: data_callback:
Lifecycle HelloWorld #32
[lifecycle_talker-1] [INFO] [1758087807.397214310] [talker]: Lifecycle publisher
is active. Publishing: [Lifecycle HelloWorld #33]
[lifecycle_listener-2] [INFO] [1758087807.397384225] [listener]: data_callback:
Lifecycle HelloWorld #33
[lifecycle_talker-1] [INFO] [1758087808.397314716] [talker]: Lifecycle publisher
is active. Publishing: [Lifecycle HelloWorld #34]
[lifecycle_listener-2] [INFO] [1758087808.397715983] [listener]: data_callback:
Lifecycle HelloWorld #34
[lifecycle_talker-1] [INFO] [1758087809.397347543] [talker]: Lifecycle publisher
is active. Publishing: [Lifecycle HelloWorld #35]
[lifecycle_listener-2] [INFO] [1758087809.397760514] [listener]: data_callback:
Lifecycle HelloWorld #35

jy@jy: ~/ros2_ws
jy@jy: ~/ros2_ws 80x24
jy@jy:~$ cd ros2_ws
jy@jy:~/ros2_ws$ source install/setup.bash
jy@jy:~/ros2_ws$ ros2 run life_qos lifecycle_listener
[INFO] [1758087818.120592819] [lc_listener]: data_callback: Lifecycle HelloWorld
#34
[INFO] [1758087818.120708259] [lc_listener]: data_callback: Lifecycle HelloWorld
#35
[INFO] [1758087818.120741633] [lc_listener]: data_callback: Lifecycle HelloWorld
#36
[INFO] [1758087818.120759277] [lc_listener]: data_callback: Lifecycle HelloWorld
#37
[INFO] [1758087818.120775308] [lc_listener]: data_callback: Lifecycle HelloWorld
#38
[INFO] [1758087818.120790497] [lc_listener]: data_callback: Lifecycle HelloWorld
#39
[INFO] [1758087818.120805716] [lc_listener]: data_callback: Lifecycle HelloWorld
#40
[INFO] [1758087818.120820695] [lc_listener]: data_callback: Lifecycle HelloWorld
#41
[INFO] [1758087818.120835272] [lc_listener]: data_callback: Lifecycle HelloWorld
#42
[INFO] [1758087818.120850181] [lc_listener]: data_callback: Lifecycle HelloWorld
#43
[INFO] [1758087818.398112628] [lc_listener]: data_callback: Lifecycle HelloWorld
```

