At first set up mosquitto broker **1**
Mosquitto broker installation 1
MQTT **2**
Python Part 2
MQTT_example.py 2
C++ Part 3
Paho.mqtt.cpp Installation 3
Paho.mqtt.c Releases - Installation from source.zip 4
Installation from ..tar.gz.zip (There is no tar.gz present for cpp) 5
Alternatives 5
Coding Part CPP 6
Subscriber 6
Publisher 8
Publisher to client_sub 10

At first set up mosquitto broker

Mosquitto broker installation

Need this pkg
`sudo apt-get install mosquitto mosquitto-clients`
Check version:
`mosquitto -v`
If Mosquitto is running, you should see a message like this:

~~~
```
1621104855: mosquitto version 1.6.9 starting
1621104855: Using default config.
1621104855: Opening ipv4 listen socket on port 1883.
1621104855: Opening ipv6 listen socket on port 1883.
1621104855: mosquitto version 1.6.9 running
```
~~~

Test Mosquitto:

You can test whether Mosquitto is running correctly by using the Mosquitto clients to publish and subscribe to topics. To publish to a topic, use the mosquitto_pub command. For example:

Subscribe to topic:
`mosquitto_sub -t "test/topic"`

Open another terminal and publish a msg:
`mosquitto_pub -t "test/topic" -m "Hello, world!"`

You see this msg in other terminal. If you see it then mosquitto is up.

MQTT

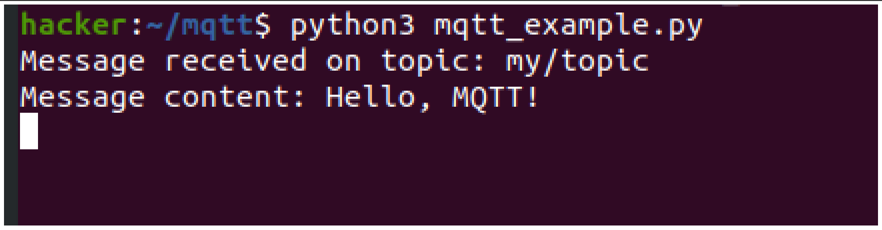
Python Part

It will need paho-mqtt,

```
`pip install paho-mqtt`
```

MQTT_example.py

~~~
```
import paho.mqtt.client as mqtt

# Callback function for when a message is received
def on_message(client, userdata, message):
    print("Message received on topic:", message.topic)
    print("Message content:", message.payload.decode())

# Set up MQTT client
client = mqtt.Client()
client.on_message = on_message
client.connect("localhost", 1883)
client.subscribe("my/topic")
~~~

```
# Start the MQTT client loop
client.loop_start()

# Publish a message to the "my/topic" topic
client.publish("my/topic", "Hello, MQTT!")

# Wait for messages
while True:
    pass

# Disconnect from the MQTT broker
client.loop_stop()
client.disconnect()

```
```

Run it with python3:
You will see the following msg:

```
```
Message received on topic: my/topic
Message content: Hello, MQTT!
```
```



```
hacker:~/mqtt$ python3 mqtt_example.py
Message received on topic: my/topic
Message content: Hello, MQTT!
```

C++ Part

Paho.mqtt.cpp Installation

Clang will not find the library mosquitto.h, so, install following pkg,

```
sudo apt-get install libmosquitto-dev
```

Now check, it will generally be present inside `/usr/include`,

```
find / -name mosquitto.h
```
```

**At first need these paho-mqtt c++ library and that will need paho-mqtt c library:**
Generally the pkg is `sudo apt-get install libpaho-mqtt3c-dev` for C and `sudo apt-get install libpaho-mqttpp3-dev` for CPP. But this pkg is only available for ubuntu-22. So, install these from github (either from source or from releases).

## Paho.mqtt.c Releases - Installation from source.zip

Download `source.zip`: https://github.com/eclipse/paho.mqtt.c/releases
**Unzip the file and `cd` to that directory *(no need to create a build dir)* then do following**,

~~~
```
cmake .
make
sudo make install
```
~~~

## Do the same for CPP

Download `source.zip`: https://github.com/eclipse/paho.mqtt.cpp/releases
Unzip the file and cd to that directory then do the following,

~~~
```
cmake .
make
sudo make install
```
~~~

All the necessary libraries will be present inside src. Now, you can either copy these files to your local dir `usr/local/lib` or, add it to the path like this:

```
`export LD_LIBRARY_PATH=/usr/local/lib:${LD_LIBRARY_PATH}`
```

Change the `/usr/local/lib` with your library path.

While doing this for cpp, CMake was not able to find the PahoMqttC,

~~~
```
CMake Error at
/usr/share/cmake-3.16/Modules/FindPackageHandleStandardArgs.cmake:146
(message):
  Could NOT find PahoMqttC (missing: PAHO_MQTT_C_LIBRARIES)
Call Stack (most recent call first):
  /usr/share/cmake-3.16/Modules/FindPackageHandleStandardArgs.cmake:393
~~~

```
(_FPHSA_FAILURE_MESSAGE)
  cmake/FindPahoMqttC.cmake:33 (FIND_PACKAGE_HANDLE_STANDARD_ARGS)
  src/CMakeLists.txt:26 (find_package)
```
```

Then added the path manually.

```
`export LD_LIBRARY_PATH=/usr/local/lib:$LD_LIBRARY_PATH`
```

[Always encountered this error and solved after adding specifying 2 path at the beginning of the CMakeLists.txt]/

If adding the path to the LD_LIBRARY_PATH environment variable did not work, you can try specifying the path to the PahoMqttC libraries explicitly in your CMakeLists.txt file. You can do this by adding the following lines before the `find_package` command:

```
set(PAHO_MQTT_C_LIBRARIES "/usr/local/lib/libpaho-mqtt3c.so")
set(PAHO_MQTT_C_INCLUDE_DIRS "/usr/local/include")
```

Then ran cmake .,make then sudo make install

After installing from source all the libraries will automatically be added to "`/usr/local/lib`"

Installation from ..tar.gz.zip (There is no tar.gz present for cpp)

Download the file named likely: Eclipse-Paho-MQTT-C-1.3.12-Linux.tar.gz.zip
Unzip it and cd to the dir.
All the files will be present inside the lib dir. Simply copy the lib dir in `/usr/local/lib`

```
`sudo cp lib/* /usr/local/lib`
```

Or, add it to the path,

```
`export LD_LIBRARY_PATH=/usr/local/lib:${LD_LIBRARY_PATH}`
```

Change the ``/usr/local/lib`` with your library path.
```

Alternatives

Follow this for arm:
https://ubuntu.pkgs.org/22.04/ubuntu-universe-amd64/libpaho-mqtt-dev_1.3.9-1build1_amd64.deb.html

Package	Version	Arch	Repository
libpaho-mqtt-dev_1.3.9-1build1_arm64.deb	**1.3.9**	**arm64**	**Ubuntu Universe Official**
libpaho-mqtt-dev	**All**	**All**	**All**

Coding Part CPP

Subscriber

Code, now, let's write a client subscriber and check it with mosquitto broker. Let's goooo:
client_subscriber.cpp

```
#include <iostream>
#include <cstring>
#include <cstdlib>
#include <chrono>
#include <thread>
#include "mqtt/async_client.h"

const std::string SERVER_ADDRESS("tcp://localhost:1883");
const std::string CLIENT_ID("mqtt_cpp_async_subscriber");
const std::string TOPIC("my/topic");

class mqtt_callback : public virtual mqtt::callback
{
public:
    void connection_lost(const std::string& cause) override {
        std::cout << "\nConnection lost" << std::endl;
        if (!cause.empty())
            std::cout << "cause: " << cause << std::endl;
    }
    void delivery_complete(mqtt::delivery_token_ptr token) override {
        std::cout << "\nDelivery complete for token: "
                  << (token ? token->get_message_id() : -1) << std::endl;
    }

    void message_arrived(mqtt::const_message_ptr msg) override {
        std::cout << "Message received on topic: " << msg->get_topic() <<
std::endl;
        std::cout << "Message content: " << msg->to_string() << std::endl;
    }
};

int main(int argc, char* argv[])
{
    mqtt::async_client client(SERVER_ADDRESS, CLIENT_ID);
    mqtt_callback cb;
    client.set_callback(cb);

    mqtt::connect_options connOpts;
    connOpts.set_keep_alive_interval(20);
    connOpts.set_clean_session(true);

    try {
        std::cout << "Connecting to MQTT server: '" << SERVER_ADDRESS << "'..
<< std::flush;
        mqtt::token_ptr conntok = client.connect(connOpts);
        conntok->wait();
        std::cout << "OK\n" << std::endl;

        std::cout << "Subscribing to topic: " << TOPIC << "..." << std::flush
        mqtt::token_ptr subtok = client.subscribe(TOPIC, 0);
        subtok->wait();
        std::cout << "OK\n" << std::endl;

        while (true) {
            std::this_thread::sleep_for(std::chrono::seconds(1));
        }

        std::cout << "Disconnecting from MQTT server..." << std::flush;
        mqtt::token_ptr disconntok = client.disconnect();
        disconntok->wait();
        std::cout << "OK" << std::endl;
    }
    catch (const mqtt::exception& exc) {
        std::cerr << "Error: " << exc.what() << std::endl;
        return 1;
    }

    return 0;
}
```

Now, run the file,

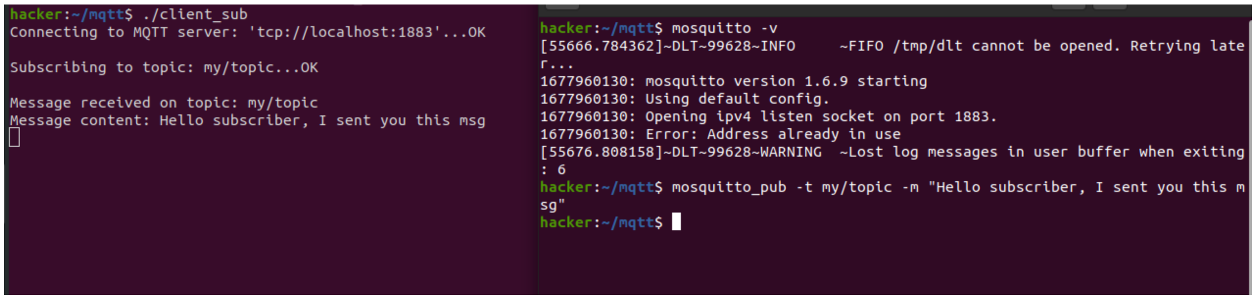
```
clang++-14 client_subscriber.cpp -o client_sub -lpaho-mqttpp3
-lpaho-mqtt3c -lpaho-mqtt3a -lmosquitto
```

The -lmosquitto option links your code against the Mosquitto library during the compilation process, allowing you to use the Mosquitto functions in your program.

Check if mosquitto is up in other terminal and do following,

```
mosquitto -v
mosquitto_pub -t my/topic -m "Hello subscriber, I sent you this msg"
```

You should see this hello message is sent to the other terminal.



```
hacker:~/mqtt$ ./client_sub
Connecting to MQTT server: 'tcp://localhost:1883'...OK

Subscribing to topic: my/topic...OK

Message received on topic: my/topic
Message content: Hello subscriber, I sent you this msg
```

```
hacker:~/mqtt$ mosquitto -v
[55666.784362]~DLT~99628~INFO     ~FIFO /tmp/dlt cannot be opened. Retrying late
r...
1677960130: mosquitto version 1.6.9 starting
1677960130: Using default config.
1677960130: Opening ipv4 listen socket on port 1883.
1677960130: Error: Address already in use
[55676.808158]~DLT~99628~WARNING  ~Lost log messages in user buffer when exiting
: 6
hacker:~/mqtt$ mosquitto_pub -t my/topic -m "Hello subscriber, I sent you this m
sg"
hacker:~/mqtt$ 
```

Publisher

Now, let's write a publisher and check if it is working with mqtt broker or, not:

publisher.cpp

```
#include <iostream>
#include <cstring>
#include <cstdlib>
#include <chrono>
#include <thread>
#include "mqtt/async_client.h"
```

```cpp
const std::string SERVER_ADDRESS("tcp://localhost:1883");
const std::string CLIENT_ID("mqtt_cpp_async_publisher");
const std::string TOPIC("my/topic");

class callback : public virtual mqtt::callback
{
public:
    void connection_lost(const std::string& cause) override {
        std::cout << "\nConnection lost" << std::endl;
        if (!cause.empty())
            std::cout << "\tcause: " << cause << std::endl;
    }

    void delivery_complete(mqtt::delivery_token_ptr token) override {
        std::cout << "\nDelivery complete for token: "
                  << (token ? token->get_message_id() : -1) << std::endl;
    }
};

int main(int argc, char* argv[])
{
    mqtt::async_client client(SERVER_ADDRESS, CLIENT_ID);
    callback cb;
    client.set_callback(cb);

    mqtt::connect_options connOpts;
    connOpts.set_keep_alive_interval(20);
    connOpts.set_clean_session(true);

    try {
        std::cout << "Connecting to MQTT server: '" << SERVER_ADDRESS <<
"'..." << std::flush;
        client.connect(connOpts)->wait();
        std::cout << "OK\n" << std::endl;

        std::string payload = "Hello, MQTT!";
        mqtt::message_ptr pubmsg = mqtt::make_message(TOPIC, payload);
        pubmsg->set_qos(0);
        client.publish(pubmsg)->wait();
        std::cout << "Published message: " << payload << std::endl;

        std::cout << "Disconnecting from MQTT server..." << std::flush;
        client.disconnect()->wait();
        std::cout << "OK" << std::endl;
    }
    catch (const mqtt::exception& exc) {
        std::cerr << "Error: " << exc.what() << std::endl;
        return 1;
    }

    return 0;
}
```

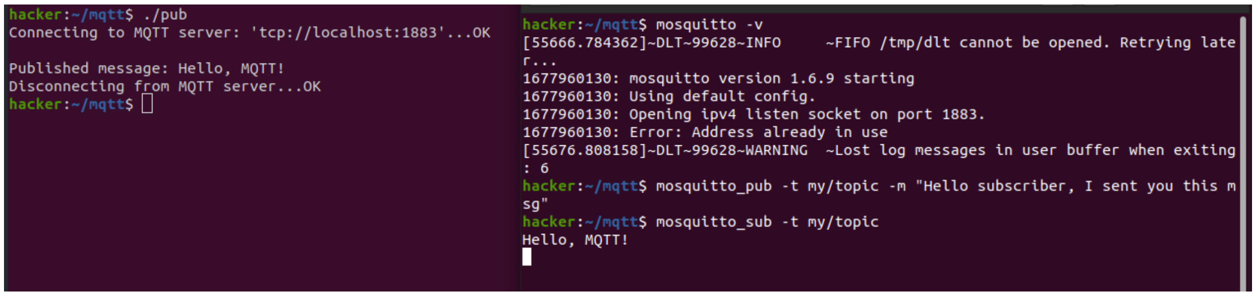
Now, compile it,

```
clang++-14 publisher.cpp -o pub -lpaho-mqttpp3 -lpaho-mqtt3c -lpaho-mqtt3a
-lmosquitto
```

Open another terminal and do following,

```
```
mosquitto -v
mosquitto_sub -t my/topic
```
```

Now, run that binary `./pub`, you will see the msg sent to the publisher.



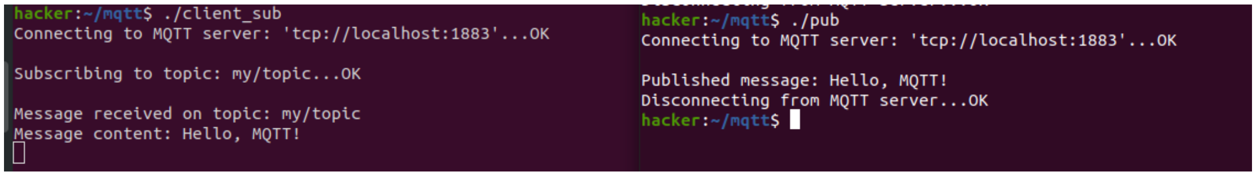
```
hacker:~/mqtt$ ./pub
Connecting to MQTT server: 'tcp://localhost:1883'...OK

Published message: Hello, MQTT!
Disconnecting from MQTT server...OK
hacker:~/mqtt$
```

```
hacker:~/mqtt$ mosquitto -v
[55666.784362]~DLT~99628~INFO     ~FIFO /tmp/dlt cannot be opened. Retrying late
r...
1677960130: mosquitto version 1.6.9 starting
1677960130: Using default config.
1677960130: Opening ipv4 listen socket on port 1883.
1677960130: Error: Address already in use
[55676.808158]~DLT~99628~WARNING  ~Lost log messages in user buffer when exiting
: 6
hacker:~/mqtt$ mosquitto_pub -t my/topic -m "Hello subscriber, I sent you this m
sg"
hacker:~/mqtt$ mosquitto_sub -t my/topic
Hello, MQTT!
```

Publisher to client_sub

Sending msg from publisher(publisher.cpp) to client_subscriber(client_subscriber.cpp) i.e., cpp to cpp

If this both of these works with the mosquitto broker sub and pub, then it will also work with both pub and client executable. Check it in different terminal.



```
hacker:~/mqtt$ ./client_sub
Connecting to MQTT server: 'tcp://localhost:1883'...OK

Subscribing to topic: my/topic...OK

Message received on topic: my/topic
Message content: Hello, MQTT!
```

```
hacker:~/mqtt$ ./pub
Connecting to MQTT server: 'tcp://localhost:1883'...OK

Published message: Hello, MQTT!
Disconnecting from MQTT server...OK
hacker:~/mqtt$
```

*******MQTT CPP tutorial*****:**
https://github.com/redboltz/mqtt_cpp/tree/master
https://github.com/redboltz/mqtt_cpp/wiki

If getting this error,error: unable to load plugin '-lpaho-mqttpp3': '-lpaho-mqttpp3: cannot open

```
shared object file: No such file or directory'
Makefile:36: recipe for target 'modified.ll' failed
make: *** [modified.ll] Error 1
```

Then you forgot to add the plugin path, you should find it in `/usr/local/lib`, so add it to path,

```
export LD_LIBRARY_PATH=/usr/local/lib
```