MQTT

▼ Step 1 — Installing Mosquitto

First, log in using your non-root user and update the package lists using `apt update`:

```
sudo apt update
```

Now, install Mosquitto using `apt install`:

```
sudo apt install mosquitto mosquitto-clients
```

Log in to your server a second time, so you have two terminals side-by-side. In the new terminal, use `mosquitto_sub` to subscribe to the test topic:

```
mosquitto_sub -h localhost -t test
```

- `h` is used to specify the hostname of the MQTT server, and `t` is the topic name. You'll see no output after hitting `ENTER` because `mosquitto_sub` is waiting for messages to arrive. Switch back to your other terminal and publish a message:

```
mosquitto_pub -h localhost -t test -m "hello world"
```

The options for `mosquitto_pub` are the same as `mosquitto_sub`, though this time we use the additional `-m` option to specify our message. Hit `ENTER`, and you should see **hello world** pop up in the other terminal. You've sent your first MQTT message!

▼ Step 2 — Configuring MQTT Passwords

Mosquitto includes a utility to generate a special password file called `mosquitto_passwd`. This command will prompt you to enter a password for the specified username, and place the results in `/etc/mosquitto/passwd`.

```
sudo mosquitto_passwd -c /etc/mosquitto/passwdsammy
```

Now we'll open up a new configuration file for Mosquitto and tell it to use this password file to require logins for all connections:

```
sudo nano /etc/mosquitto/conf.d/default.conf
```

This should open an empty file. Paste in the following:

/etc/mosquitto/conf.d/default.conf

```
allow_anonymous false
password_file /etc/mosquitto/passwd
listener 1886 0.0.0.0

```

Be sure to leave a trailing newline at the end of the file.

`allow_anonymous false` will disable all non-authenticated connections, and the `password_file` line tells Mosquitto where to look for user and password information. Save and exit the file.

Now we need to restart Mosquitto and test our changes.

```
sudo systemctl restart mosquitto
```

Try to publish a message without a password:

```
mosquitto_pub -h localhost -t "test" -m "hello world"
```

The message should be rejected:

```
Output
Connection Refused: not authorised.
Error: The connection was refused.
```

Before we try again with the password, switch to your second terminal window again, and subscribe to the 'test' topic, using the username and password this time:

```
mosquitto_sub -h localhost -t test -u "sammy" -P "password"
```

It should connect and sit, waiting for messages. You can leave this terminal open and connected for the rest of the tutorial, as we'll periodically send it test messages.

Now publish a message with your other terminal, again using the username and password:

```
mosquitto_pub -h localhost -t "test" -m "hello world" -u "sammy" -P "pa
ssword"
```

The message should go through as in Step 1. We've successfully added password protection to Mosquitto. Unfortunately, we're sending passwords unencrypted over the internet. We'll fix that next by adding SSL encryption to Mosquitto.