EMQX configuration steps

EMQX Cloud Configuration mqtt.broker=qd081a20.ala.dedicated.aws.emqxcloud.com mgtt.port=1883 mqtt.username=chargeghar mqtt.password=5060 mqtt.clientId=iotdemo-server mqtt.ssl=false

EMQX Management API Configuration emqx.api.url=https://qd081a20.ala.dedicated.aws.emqxclo ud.com:8443/api/v5 emqx.api.key=mb450eff10110460 emqx.api.secret=I0HL2xIoXc.3Qzt5H9A.Wg0yWp395-oC # Keep productKey for device identification productKey=powerbank #\u81EA\u5B9A\u4E49Topic\u8DEF\u5F84\u5E26/user\u4 E3Atrue \u4E0D\u5E26/user\u4E3Afalse topicType=true

which details is ststic and which i need to get from emqx cloud console

seperte and make steps

Perfect — let's break down your **EMQX Cloud Configuration** file into:

- 1. Static (Project-defined or custom) values you can define and keep them fixed in your app.
- 2. Dynamic (Cloud Console) values you must get or create these from your EMQX Cloud Console.

Then I'll explain step-by-step how to find or create them 👇



EMQX Cloud Configuration Breakdown

Key	Description	Source	Static / Dynamic
mqtt.broker	The EMQX Cloud broker hostname.	From EMQX Cloud Console → Deployment Details.	Oynamic
mqtt.port	Port used for MQTT connection. Usually 1883 (TCP) or 8883 (SSL).	Default or console.	Static (default)
mqtt.username	MQTT username (set in authentication section of EMQX Cloud).	Create under Console → Authentication → Username/Password.	Dynamic
mqtt.password	The password for the above username.	Same place as username in Console.	Dynamic
mqtt.clientId	Unique client ID for your app or service.	You can define it manually (e.g., iotdemo-server).	Static (custom)
mqtt.ssl	Whether to use SSL (true/false).	Depends on which port you use.	Static (custom)
emqx.api.url	EMQX REST API endpoint (for management).	From Console → API Docs or Deployment Info.	Dynamic
emqx.api.key	API key to access EMQX REST API.	From Console → API Key Management → Create Key.	Dynamic
emqx.api.secret	Secret for the above key.	Shown only once when you create the API key.	Dynamic
productKey	Your internal product identifier (e.g., powerbank).	Custom (used by your app).	Static (custom)
topicType	Custom topic structure flag (true = with /user , false = without /user).	Custom logic.	Static (custom)

| Steps to Set Up EMQX Cloud Configuration

Step 1 — Create Deployment

- 1. Go to EMQX Cloud Console 7.
- 2. Click Create Deployment → choose Region (AWS) and deployment type.
- **3.** Wait until the deployment is running.
- 4. Copy the broker URL (e.g., qd081a20.ala.dedicated.aws.emqxcloud.com).
- Use this in mqtt.broker.

Step 2 — **Get Connection Port**

- Default ports:
 - 1883 → Non-SSL
 - 8883 → SSL/TLS
- Choose based on your mqtt.ssl setting.
- Example:

```
properties
mqtt.port=1883
mqtt.ssl=false
```

Step 3 — Create MQTT Username and Password

- **1.** Go to Deployment → Authentication → Authentication List.
- 2. Click Add Authentication → choose Username & Password.
- 3. Create:
 - Username: chargeghar
 - Password: 5060
- Use them in:

```
mqtt.username=chargeghar
mqtt.password=5060
```

Step 4 — Create API Key for Management API

- **1.** In Console, open **Deployment** → **API Key Management**.
- 2. Click Create API Key.
- **3.** Copy:
 - **API Key ID** → emqx.api.key
 - **API Secret** → emqx.api.secret (save immediately shown once)

Example:

```
emqx.api.key=mb450eff10110460
emqx.api.secret=I0HL2xIoXc.3Qzt5H9A.Wg0yWp395-oC
```

Step 5 — Get EMQX Management API URL

1. Usually formatted as:

```
arduino

https://<br/>broker-host>:8443/api/v5
```

Example:

bash

https://qd081a20.ala.dedicated.aws.emqxcloud.com:8443/api/v5

✓ Use it in emqx.api.url.

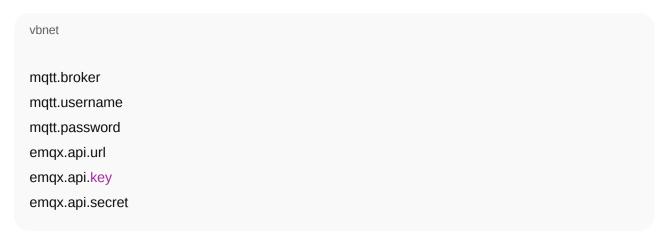
Step 6 — **Set Custom App Config**

Define your own values:

```
mqtt.clientId=iotdemo-server
productKey=powerbank
topicType=true
```

▼ Final Categorization Summary

Dynamic (from EMQX Cloud Console):



Static / Custom (you define):

mqtt.port
mqtt.clientId
mqtt.ssl
productKey
topicType

Would you like me to make this into a **developer setup guide** ($README_emqx_setup.md$) for your project repo?