

✓ Node-RED + PLC + HiveMQ + Kafka Integration: Full Setup Guide

🧰 Prerequisites (Software Needed)

1. **Node-RED**
 - Installed via npm: `npm install -g --unsafe-perm node-red`
 2. **MQTT Broker**
 - Use **HiveMQ Cloud**
 - For HiveMQ Cloud: Create an account → get broker URL, port, username/password
 3. **PLC Software / Ethernet-IP**
 - PLC must support Ethernet/IP (e.g., Allen Bradley CompactLogix)
 - IP address of PLC must be known (e.g., 192.168.1.60)
 4. **Optional: Kafka**
 - Local or remote Kafka broker running on port (e.g., 9092)
 - Ensure `kafka-console-producer` and `kafka-console-consumer` are working
-

👤💻 Step-by-Step Instructions

2️⃣ Install Node-RED

Open terminal or CMD:

```
bash
CopyEdit
npm install -g --unsafe-perm node-red
```

3️⃣ Start Node-RED

```
bash
CopyEdit
node-red
```

- Open browser: `http://127.0.0.1:1880`
-

🔧 Configure Nodes in Node-RED (Based on your screenshots)

📦 [1] ETH-IP Node (PLC)

- Add **eth-ip** in node (Install from "Manage Palette" if not available)
- Set PLC IP: `192.168.1.60`
- Mode: All Tags

- Tags:

```
css
CopyEdit
Date_Time[0] to Date_Time[5] (all type DINT)
```

[2] Delay Node

- Set to limit to 1 message/minute to avoid flooding:
 - Mode: Rate Limit
 - Messages: 1 msg/min

[3] Function Node (Optional for transformation)

Example function code:

```
js
CopyEdit
msg.payload = {
  tag0: msg.payload["Date_Time[0]"],
  tag1: msg.payload["Date_Time[1]"],
  tag2: msg.payload["Date_Time[2]"]
};
return msg;
```

[4] MQTT Out Node (to HiveMQ)

- Server: Create new → set broker like `broker.hivemq.com` or HiveMQ Cloud
- Topic: PLC
- QoS: 1 (recommended)

HiveMQ Cloud Setup

If using **HiveMQ Cloud**:

1. Register: <https://www.hivemq.com/cloud/>
2. Create a broker instance
3. Note:
 - **Host URL**
 - **Port (e.g., 8883 for TLS)**
 - **Username/password**

☐ Kafka Integration (Optional)

- Add **Kafka Producer** node from palette manager
- Server: `localhost:9092` or actual Kafka server IP
- Topic: `plc-data`

How to Install Missing Nodes

1. Click top-right **menu** $\equiv \rightarrow$ **Manage Palette**
2. Go to **Install** tab
 - o Search:
 - `node-red-contrib-eth-ip` for PLC
 - `node-red-contrib-mqtt-broker`
 - `node-red-contrib-kafka`

Testing

- Deploy the flow
- View Debug output (right panel)
- Confirm PLC values flow to MQTT $\rightarrow$ Kafka or HiveMQ

Summary

Task	Tool or Config
Node-RED Platform	<code>node-red</code>
PLC Comm (ETH/IP)	<code>eth-ip</code> in node with IP <code>192.168.1.60</code>
Rate Limiting	<code>delay</code> node - 1 msg/min
MQTT Publishing	<code>mqtt</code> out node with HiveMQ topic <code>PLC</code>
Optional Kafka Push	Kafka Producer Node
HiveMQ Dashboard	Shows MQTT messages live

.

Node-RED interface showing a flow diagram and a debug console.

Flow Diagram:

- Input: `192.168.1.60.0` (IP address)
- Node: `limit 1 msg/m` (Rate limiter)
- Node: `function 1` (Function node)
- Node: `PLC` (PLC node)
- Node: `debug 1` (Debug node)
- Node: `modbus-client` (Modbus client node)

Debug Console:

```
["Date_Time[0]":2025,"Date_Time[1]":7,"Date_Time[2]":22,"Date_Time[3]":14,"Date_Time[4]":24,"Date_Time[5]":53]
7/22/2025, 2:25:52 PM node debug 1
msg.payload: string(110)
+
["Date_Time[0]":2025,"Date_Time[1]":7,"Date_Time[2]":22,"Date_Time[3]":14,"Date_Time[4]":25,"Date_Time[5]":53]
7/22/2025, 2:26:52 PM node debug 1
msg.payload: string(110)
+
["Date_Time[0]":2025,"Date_Time[1]":7,"Date_Time[2]":22,"Date_Time[3]":14,"Date_Time[4]":26,"Date_Time[5]":53]
7/22/2025, 2:27:52 PM node debug 1
msg.payload: string(110)
+
["Date_Time[0]":2025,"Date_Time[1]":7,"Date_Time[2]":22,"Date_Time[3]":14,"Date_Time[4]":27,"Date_Time[5]":53]
7/22/2025, 2:28:52 PM node debug 1
msg.payload: string(110)
+
["Date_Time[0]":2025,"Date_Time[1]":7,"Date_Time[2]":22,"Date_Time[3]":14,"Date_Time[4]":29,"Date_Time[5]":53]
7/22/2025, 2:29:52 PM node debug 1
msg.payload: string(110)
+
["Date_Time[0]":2025,"Date_Time[1]":7,"Date_Time[2]":22,"Date_Time[3]":14,"Date_Time[4]":30,"Date_Time[5]":53]
7/22/2025, 2:30:52 PM node debug 1
msg.payload: string(110)
+
["Date_Time[0]":2025,"Date_Time[1]":7,"Date_Time[2]":22,"Date_Time[3]":14,"Date_Time[4]":31,"Date_Time[5]":53]
```

Node-RED interface showing a flow diagram, a debug console, and an edit dialog for the PLC node.

Flow Diagram:

- Input: `192.168.1.60.0` (IP address)
- Node: `limit 1 msg/m` (Rate limiter)
- Node: `function 1` (Function node)
- Node: `PLC` (PLC node)
- Node: `debug 1` (Debug node)
- Node: `modbus-client` (Modbus client node)

Debug Console:

```
["Date_Time[0]":2025,"Date_Time[1]":7,"Date_Time[2]":22,"Date_Time[3]":14,"Date_Time[4]":24,"Date_Time[5]":53]
7/22/2025, 2:26:52 PM node debug 1
msg.payload: string(110)
+
["Date_Time[0]":2025,"Date_Time[1]":7,"Date_Time[2]":22,"Date_Time[3]":14,"Date_Time[4]":25,"Date_Time[5]":53]
7/22/2025, 2:27:52 PM node debug 1
msg.payload: string(110)
+
["Date_Time[0]":2025,"Date_Time[1]":7,"Date_Time[2]":22,"Date_Time[3]":14,"Date_Time[4]":26,"Date_Time[5]":53]
7/22/2025, 2:28:52 PM node debug 1
msg.payload: string(110)
+
["Date_Time[0]":2025,"Date_Time[1]":7,"Date_Time[2]":22,"Date_Time[3]":14,"Date_Time[4]":27,"Date_Time[5]":53]
7/22/2025, 2:29:52 PM node debug 1
msg.payload: string(110)
+
["Date_Time[0]":2025,"Date_Time[1]":7,"Date_Time[2]":22,"Date_Time[3]":14,"Date_Time[4]":29,"Date_Time[5]":53]
7/22/2025, 2:30:52 PM node debug 1
msg.payload: string(110)
+
["Date_Time[0]":2025,"Date_Time[1]":7,"Date_Time[2]":22,"Date_Time[3]":14,"Date_Time[4]":30,"Date_Time[5]":53]
7/22/2025, 2:32:52 PM node debug 1
msg.payload: string(110)
+
["Date_Time[0]":2025,"Date_Time[1]":7,"Date_Time[2]":22,"Date_Time[3]":14,"Date_Time[4]":32,"Date_Time[5]":53]
```

Edit eth-ip in node dialog:

- Properties: `192.168.1.60.0`
- Mode: `All tags`
- Name: `Name`
- Buttons: `Delete`, `Cancel`, `Done`
- Footer: `Enabled`

Node-RED interface showing a flow with a 'limit 1 msg/m' node and a 'function 1' node. The 'Edit eth-ep endpoint node' dialog is open, displaying a 'Tag list' with columns for 'Date_Time' and 'DINT'. The 'debug' console shows a series of messages with timestamps and payloads.

Edit eth-ep endpoint node

Delete Cancel Update

Properties

Connection Tags

Tag list

Date_Time	DINT
Date_Time[0]	DINT
Date_Time[1]	DINT
Date_Time[2]	DINT
Date_Time[3]	DINT
Date_Time[4]	DINT
Date_Time[5]	DINT

Import Export

debug

```
["Date_Time[0]":2025,"Date_Time[1]":7,"Date_Time[2]":22,"Date_Time[3]":14,"Date_Time[4]":25,"Date_Time[5]":53]
7/22/2025 2:26:52 PM node debug 1
msg.payload: string(110)
+
["Date_Time[0]":2025,"Date_Time[1]":7,"Date_Time[2]":22,"Date_Time[3]":14,"Date_Time[4]":25,"Date_Time[5]":53]
7/22/2025 2:27:52 PM node debug 1
msg.payload: string(110)
+
["Date_Time[0]":2025,"Date_Time[1]":7,"Date_Time[2]":22,"Date_Time[3]":14,"Date_Time[4]":25,"Date_Time[5]":53]
7/22/2025 2:28:52 PM node debug 1
msg.payload: string(110)
+
["Date_Time[0]":2025,"Date_Time[1]":7,"Date_Time[2]":22,"Date_Time[3]":14,"Date_Time[4]":25,"Date_Time[5]":53]
7/22/2025 2:29:52 PM node debug 1
msg.payload: string(110)
+
["Date_Time[0]":2025,"Date_Time[1]":7,"Date_Time[2]":22,"Date_Time[3]":14,"Date_Time[4]":25,"Date_Time[5]":53]
7/22/2025 2:30:52 PM node debug 1
msg.payload: string(110)
+
["Date_Time[0]":2025,"Date_Time[1]":7,"Date_Time[2]":22,"Date_Time[3]":14,"Date_Time[4]":25,"Date_Time[5]":53]
7/22/2025 2:31:52 PM node debug 1
msg.payload: string(110)
+
["Date_Time[0]":2025,"Date_Time[1]":7,"Date_Time[2]":22,"Date_Time[3]":14,"Date_Time[4]":25,"Date_Time[5]":53]
7/22/2025 2:32:52 PM node debug 1
msg.payload: string(110)
+
["Date_Time[0]":2025,"Date_Time[1]":7,"Date_Time[2]":22,"Date_Time[3]":14,"Date_Time[4]":25,"Date_Time[5]":53]
```

Node-RED interface showing a flow with a 'limit 1 msg/m' node and a 'function 1' node. The 'Edit delay node' dialog is open, displaying properties for 'Rate Limit', 'All messages', and 'Drop intermediate messages'. The 'debug' console shows a series of messages with timestamps and payloads.

Edit delay node

Delete Cancel Done

Properties

Action Rate Limit

All messages

Rate 1 msg(s) per 1 Minute

allow msg rate (in ms) to override rate

Drop intermediate messages

Name

debug

```
["Date_Time[0]":2025,"Date_Time[1]":7,"Date_Time[2]":22,"Date_Time[3]":14,"Date_Time[4]":25,"Date_Time[5]":53]
7/22/2025 2:27:52 PM node debug 1
msg.payload: string(110)
+
["Date_Time[0]":2025,"Date_Time[1]":7,"Date_Time[2]":22,"Date_Time[3]":14,"Date_Time[4]":25,"Date_Time[5]":53]
7/22/2025 2:28:52 PM node debug 1
msg.payload: string(110)
+
["Date_Time[0]":2025,"Date_Time[1]":7,"Date_Time[2]":22,"Date_Time[3]":14,"Date_Time[4]":25,"Date_Time[5]":53]
7/22/2025 2:29:52 PM node debug 1
msg.payload: string(110)
+
["Date_Time[0]":2025,"Date_Time[1]":7,"Date_Time[2]":22,"Date_Time[3]":14,"Date_Time[4]":25,"Date_Time[5]":53]
7/22/2025 2:30:52 PM node debug 1
msg.payload: string(110)
+
["Date_Time[0]":2025,"Date_Time[1]":7,"Date_Time[2]":22,"Date_Time[3]":14,"Date_Time[4]":25,"Date_Time[5]":53]
7/22/2025 2:31:52 PM node debug 1
msg.payload: string(110)
+
["Date_Time[0]":2025,"Date_Time[1]":7,"Date_Time[2]":22,"Date_Time[3]":14,"Date_Time[4]":25,"Date_Time[5]":53]
7/22/2025 2:32:52 PM node debug 1
msg.payload: string(110)
+
["Date_Time[0]":2025,"Date_Time[1]":7,"Date_Time[2]":22,"Date_Time[3]":14,"Date_Time[4]":25,"Date_Time[5]":53]
7/22/2025 2:33:52 PM node debug 1
msg.payload: string(110)
+
["Date_Time[0]":2025,"Date_Time[1]":7,"Date_Time[2]":22,"Date_Time[3]":14,"Date_Time[4]":25,"Date_Time[5]":53]
```

Node-RED interface showing a flow configuration and the 'Edit mqtt out node' dialog.

Flow Configuration:

- Inject node (192.168.1.60.0) connected to a 'limit 1 msg/m' node.
- The 'limit 1 msg/m' node is connected to a 'function 1' node.
- The 'function 1' node is connected to a 'mqtt-out' node (labeled 'nodered-client').

Edit mqtt out node Dialog:

- Properties:**
 - Server: Hive
 - Topic: PLC
 - QoS: 1
 - Retain: Retain
 - Name: Name
- Tip: Leave topic, qos or retain blank if you want to set them via msg properties.

Debug Console:

```
7/22/2025, 2:27:52 PM node debug 1
msg.payload: string(110)
{"Date_Time[0]":2025,"Date_Time[1]":7,"Date_Time[2]":22,"Date_Time[3]":14,"Date_Time[4]":26,"Date_Time[5]":53}

7/22/2025, 2:28:52 PM node debug 1
msg.payload: string(110)
{"Date_Time[0]":2025,"Date_Time[1]":7,"Date_Time[2]":22,"Date_Time[3]":14,"Date_Time[4]":27,"Date_Time[5]":53}

7/22/2025, 2:29:52 PM node debug 1
msg.payload: string(110)
{"Date_Time[0]":2025,"Date_Time[1]":7,"Date_Time[2]":22,"Date_Time[3]":14,"Date_Time[4]":30,"Date_Time[5]":53}

7/22/2025, 2:30:52 PM node debug 1
msg.payload: string(110)
{"Date_Time[0]":2025,"Date_Time[1]":7,"Date_Time[2]":22,"Date_Time[3]":14,"Date_Time[4]":31,"Date_Time[5]":53}

7/22/2025, 2:31:52 PM node debug 1
msg.payload: string(110)
{"Date_Time[0]":2025,"Date_Time[1]":7,"Date_Time[2]":22,"Date_Time[3]":14,"Date_Time[4]":32,"Date_Time[5]":53}

7/22/2025, 2:32:52 PM node debug 1
msg.payload: string(110)
{"Date_Time[0]":2025,"Date_Time[1]":7,"Date_Time[2]":22,"Date_Time[3]":14,"Date_Time[4]":33,"Date_Time[5]":53}
```

Node-RED interface showing a flow configuration and the 'Edit mqtt out node' dialog.

Flow Configuration:

- Inject node (192.168.1.60.0) connected to a 'limit 1 msg/m' node.
- The 'limit 1 msg/m' node is connected to a 'function 1' node.
- The 'function 1' node is connected to a 'mqtt-out' node (labeled 'nodered-client').

Edit mqtt out node Dialog:

- Properties:**
 - Name: Hive
- Connection:**
 - Server: 9b44d1ea5cd4690b0dac9511f6344ae.s1.eu
 - Port: 8883
 - Connect automatically: ☒
 - Use TLS: ☒ TLS configuration
 - Protocol: MQTT V3.1.1
 - Client ID: node-red-client1
 - Keep Alive: 180
 - Session: ☐ Use clean session
 - ☐ Automatically unsubscribe when disconnecting

Debug Console:

```
7/22/2025, 2:28:52 PM node debug 1
msg.payload: string(110)
{"Date_Time[0]":2025,"Date_Time[1]":7,"Date_Time[2]":22,"Date_Time[3]":14,"Date_Time[4]":26,"Date_Time[5]":53}

7/22/2025, 2:29:52 PM node debug 1
msg.payload: string(110)
{"Date_Time[0]":2025,"Date_Time[1]":7,"Date_Time[2]":22,"Date_Time[3]":14,"Date_Time[4]":27,"Date_Time[5]":53}

7/22/2025, 2:30:52 PM node debug 1
msg.payload: string(110)
{"Date_Time[0]":2025,"Date_Time[1]":7,"Date_Time[2]":22,"Date_Time[3]":14,"Date_Time[4]":30,"Date_Time[5]":53}

7/22/2025, 2:31:52 PM node debug 1
msg.payload: string(110)
{"Date_Time[0]":2025,"Date_Time[1]":7,"Date_Time[2]":22,"Date_Time[3]":14,"Date_Time[4]":31,"Date_Time[5]":53}

7/22/2025, 2:32:52 PM node debug 1
msg.payload: string(110)
{"Date_Time[0]":2025,"Date_Time[1]":7,"Date_Time[2]":22,"Date_Time[3]":14,"Date_Time[4]":32,"Date_Time[5]":53}

7/22/2025, 2:33:52 PM node debug 1
msg.payload: string(110)
{"Date_Time[0]":2025,"Date_Time[1]":7,"Date_Time[2]":22,"Date_Time[3]":14,"Date_Time[4]":33,"Date_Time[5]":53}

7/22/2025, 2:34:52 PM node debug 1
msg.payload: string(110)
{"Date_Time[0]":2025,"Date_Time[1]":7,"Date_Time[2]":22,"Date_Time[3]":14,"Date_Time[4]":34,"Date_Time[5]":54}
```


Node-RED interface showing a flow configuration. The flow includes an inject node, a debug node, a complete node, a catch node, a status node, a link in node, a link call node, a link out node, a comment node, a function node, a switch node, a change node, a range node, a template node, a delay node, a trigger node, an exec node, and a Star node. The flow is connected to a PLC node and a nodered-client node. The debug console shows the following log entries:

```
["Date_Time[0]":2025,"Date_Time[1]":7,"Date_Time[2]":22,"Date_Time[3]":14,"Date_Time[4]":28,"Date_Time[5]":53]
7/22/2025, 2:29:52 PM node debug 1
msg.payload : string(110)
=
["Date_Time[0]":2025,"Date_Time[1]":7,"Date_Time[2]":22,"Date_Time[3]":14,"Date_Time[4]":29,"Date_Time[5]":53]
7/22/2025, 2:30:52 PM node debug 1
msg.payload : string(110)
=
["Date_Time[0]":2025,"Date_Time[1]":7,"Date_Time[2]":22,"Date_Time[3]":14,"Date_Time[4]":30,"Date_Time[5]":53]
7/22/2025, 2:31:52 PM node debug 1
msg.payload : string(110)
=
["Date_Time[0]":2025,"Date_Time[1]":7,"Date_Time[2]":22,"Date_Time[3]":14,"Date_Time[4]":31,"Date_Time[5]":53]
7/22/2025, 2:32:52 PM node debug 1
msg.payload : string(110)
=
["Date_Time[0]":2025,"Date_Time[1]":7,"Date_Time[2]":22,"Date_Time[3]":14,"Date_Time[4]":32,"Date_Time[5]":53]
7/22/2025, 2:33:52 PM node debug 1
msg.payload : string(110)
=
["Date_Time[0]":2025,"Date_Time[1]":7,"Date_Time[2]":22,"Date_Time[3]":14,"Date_Time[4]":33,"Date_Time[5]":53]
7/22/2025, 2:34:52 PM node debug 1
msg.payload : string(110)
=
["Date_Time[0]":2025,"Date_Time[1]":7,"Date_Time[2]":22,"Date_Time[3]":14,"Date_Time[4]":34,"Date_Time[5]":54]
7/22/2025, 2:35:52 PM node debug 1
msg.payload : string(110)
=
["Date_Time[0]":2025,"Date_Time[1]":7,"Date_Time[2]":22,"Date_Time[3]":14,"Date_Time[4]":35,"Date_Time[5]":54]
```

HiveMQ Cloud interface showing the "Your Clusters" section. The interface includes a sidebar with "Organizations" and "Clusters Overview". The main content area displays a cluster named "Serverless" with the status "Running". The cluster details include the URL, Port (TLS), and Started time.

Serverless (FREE) Running

URL: 9b44d1ea5cdd4690b0dac9511f6344ae.s1.eu.hivemq.cloud

Port (TLS): 8883

Started: 2025-07-17T13:36:59Z

[Manage Cluster](#)

Kafka Remote Consumer Setup

UI for Apache Kafka

HiveMQ Cloud

console.hivemq.cloud/clusters/9b44d1ea5cd4690b0dac9511f6344ae

HiveMQ Cloud

Search in documentation

Ctrl+K

ORGANIZATIONS

Jaydlp's organi: ▾

CLUSTERS OVERVIEW

+

Free #1

Free #1

Overview

Access Management

Integrations

Web Client

Getting Started

Connection Details

Comprehensive details and statistics for your cluster

URL

9b44d1ea5cd4690b0dac9511f6344ae.s1.eu.hivemq.cloud

Port

8883

Websocket Port

8884

TLS MQTT URL

9b44d1ea5cd4690b0dac9511f6344ae.s1.eu.hivemq.cloud:8883

TLS Websocket URL

9b44d1ea5cd4690b0dac9511f6344ae.s1.eu.hivemq.cloud:8884/mqtt

Cluster Info

Basic cluster information

Usage for This Billing Cycle

Real-time cluster statistics for your active billing period.

Access Management

Cluster access statistics

Cluster ID: 9b44d1ea5cd4690b0dac9511f6344ae

Kafka Remote Consumer Setup

UI for Apache Kafka

HiveMQ Cloud

console.hivemq.cloud/clusters/9b44d1ea5cd4690b0dac9511f6344ae/access-management

HiveMQ Cloud

Search in documentation

Ctrl+K

ORGANIZATIONS

Jaydlp's organi: ▾

CLUSTERS OVERVIEW

+

Free #1

Free #1

Overview

Access Management

Integrations

Web Client

Getting Started

Authentication

Credentials Active

Hide

Currently you have not created any credentials. Fill out the following form to create an access credentials pair and limit access to your HiveMQ Cloud MQTT instance. To learn more check out our Security Fundamentals guide.

NAME	PERMISSION	ACTIONS
Admin	PUBLISH_SUBSCRIBE	

Add new credential

Client Certificate

This functionality is available in Cloud Starter and higher tier plans.

Compare plans

JSON Web Token

This functionality is available in Cloud Starter and higher tier plans.

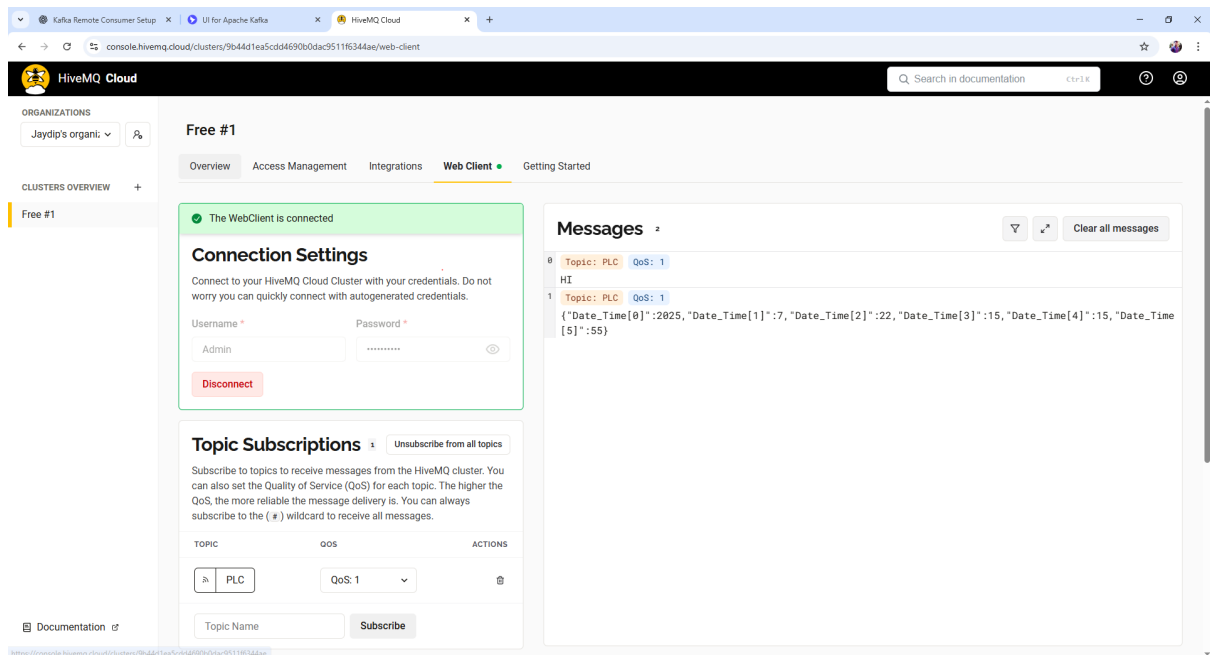
Compare plans

Authorization

Permissions

Create Permissions.

Edit



kafka_2.13-3.6.1.tgz software want in both pc server and client

PC1 (Kafka Server) Software

- Java 17 (e.g., Temurin)
- Apache Kafka 2.13-3.6.1 (Windows)
- ZeroTier One (Windows)
- Zookeeper (comes with Kafka)
- Kafka UI (optional: kafka-ui-api-v0.7.2.jar)

PC1 Setup

1. Install all software
2. Join ZeroTier network and approve device in <https://my.zerotier.com>
3. Get IP using: `zerotier-cli listnetworks`
4. Modify server.properties:

`broker.id=0`

`listeners=PLAINTEXT://0.0.0.0:9092`

`advertised.listeners=PLAINTEXT://<ZT_IP>:9092`

`zookeeper.connect=localhost:2181`

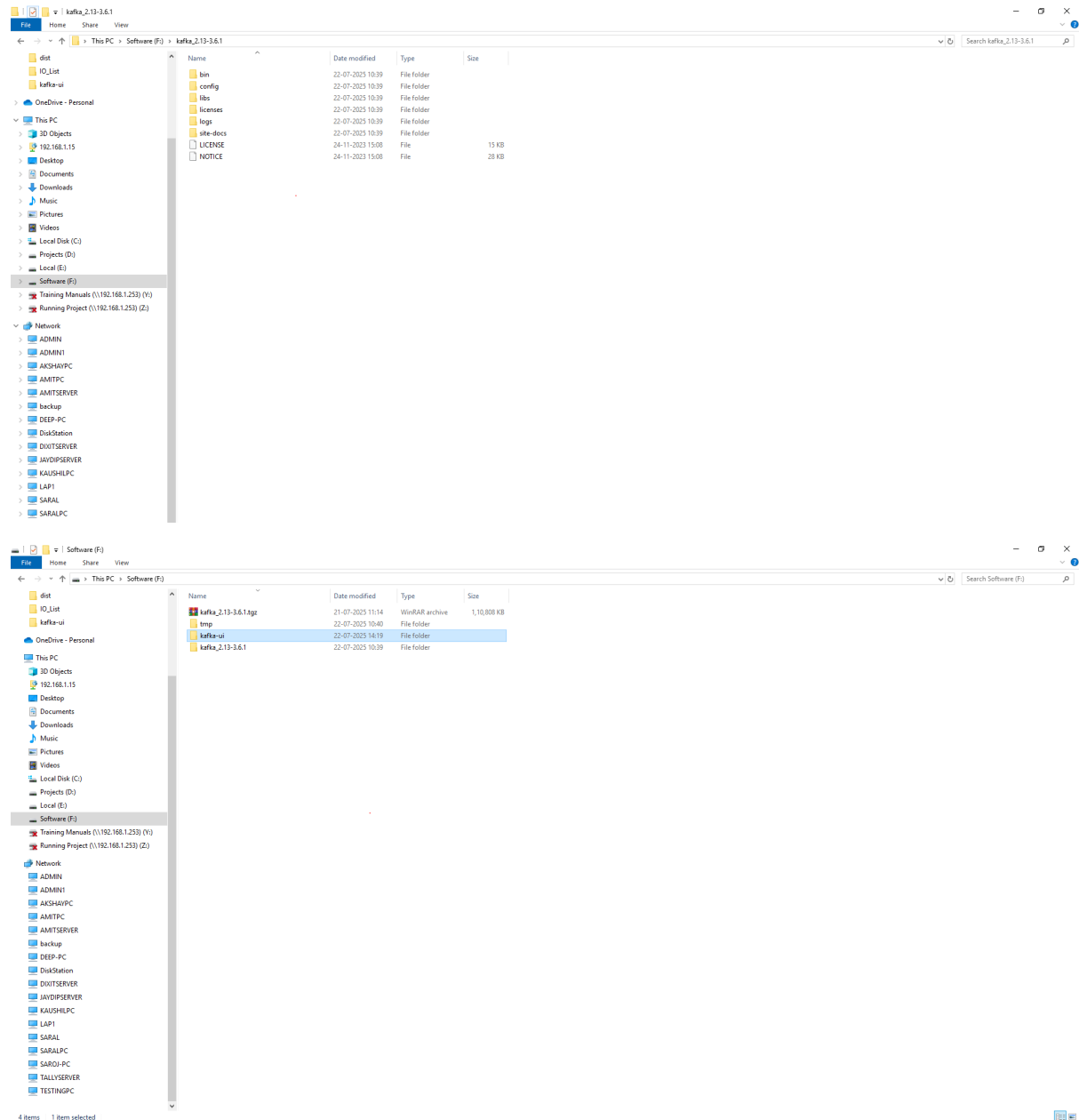
5. Start Kafka:

`zookeeper-server-start.bat ../../config/zookeeper.properties`

`kafka-server-start.bat ../../config/server.properties`

6. Create topic:

kafka-topics.bat --create --topic PLC --bootstrap-server localhost:9092 --partitions 1 --replication-factor 1



Folder look like this

PC2 (Kafka Client) Software

- Java 17
- Apache Kafka 2.13-3.6.1
- ZeroTier One

PC2 Setup

1. Install software
2. Join same ZeroTier network and approve device
3. Confirm IP with: zerotier-cli listnetworks
4. Test:

telnet <PC1_ZT_IP> 9092

5. Start consumer:

cd /d F:\kafka_2.13-3.6.1\bin\windows

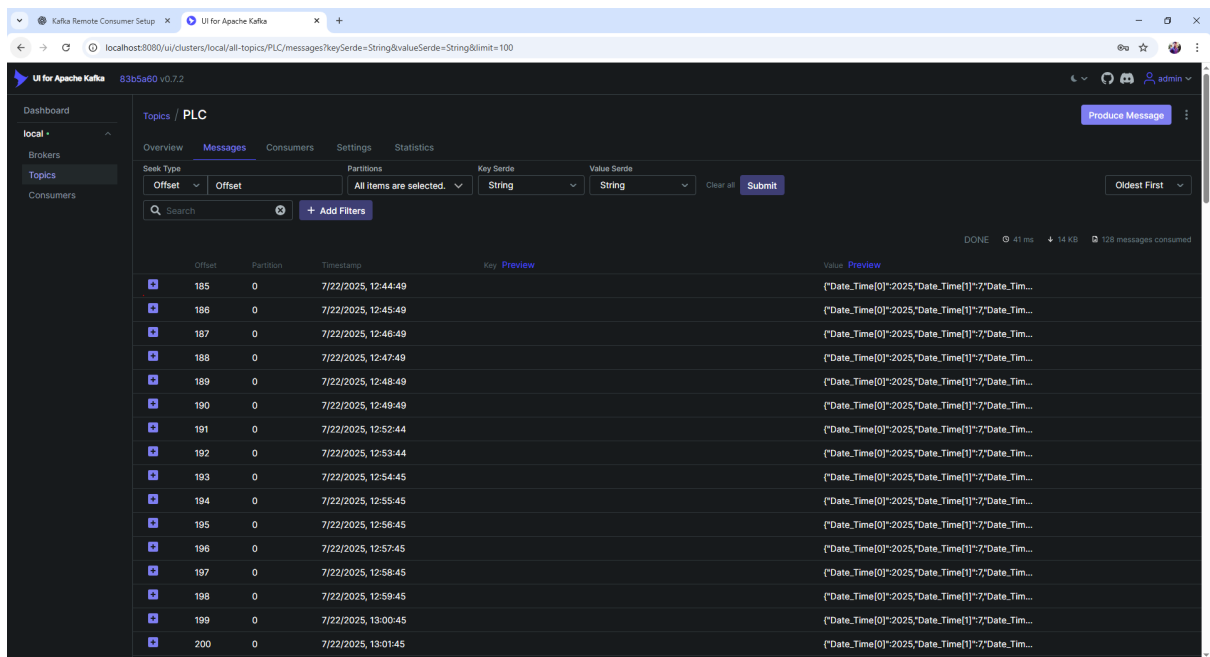
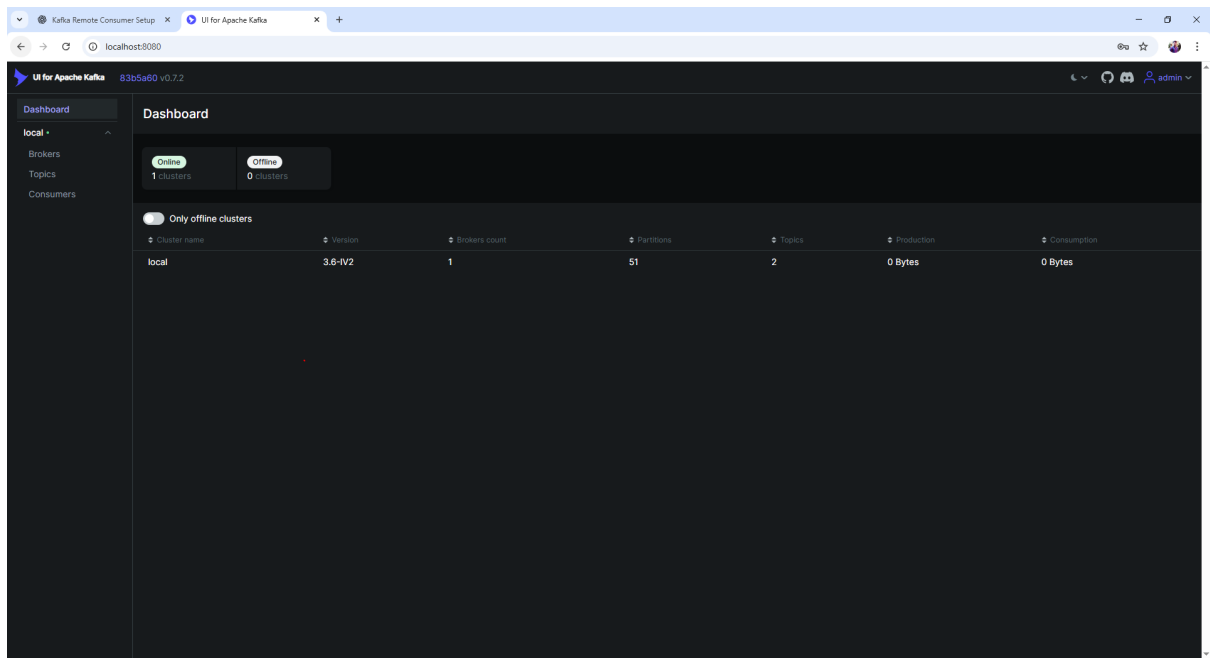
kafka-console-consumer.bat --bootstrap-server 192.168.194.109:9092 --topic PLC --from-beginning

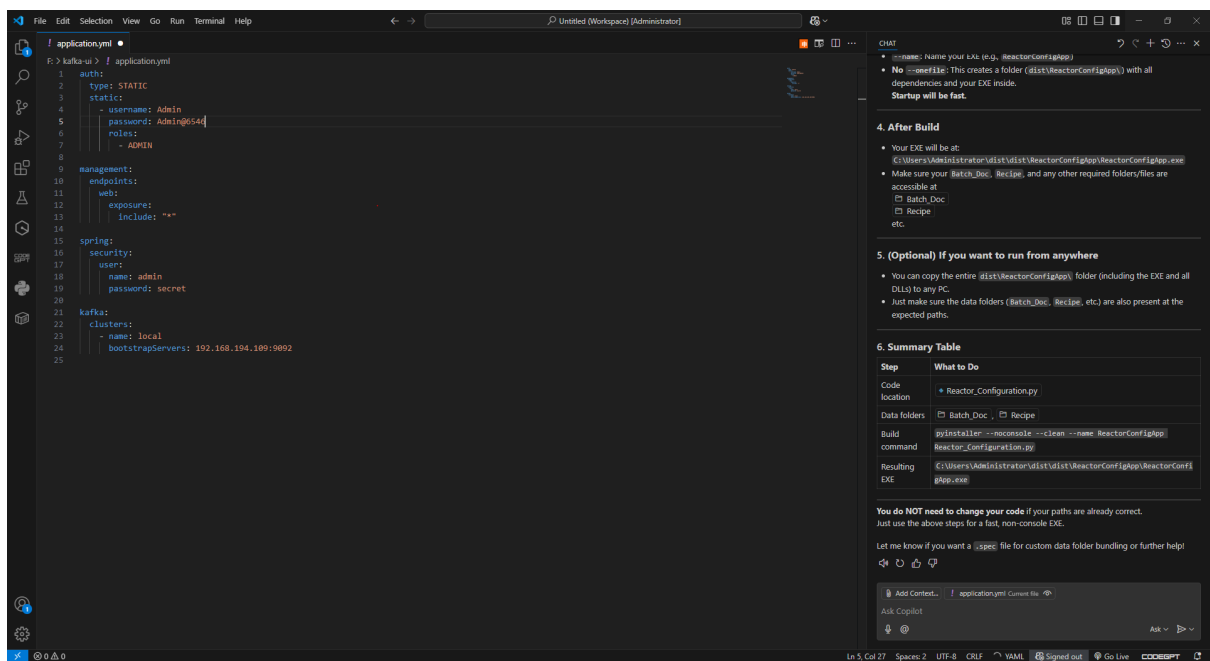
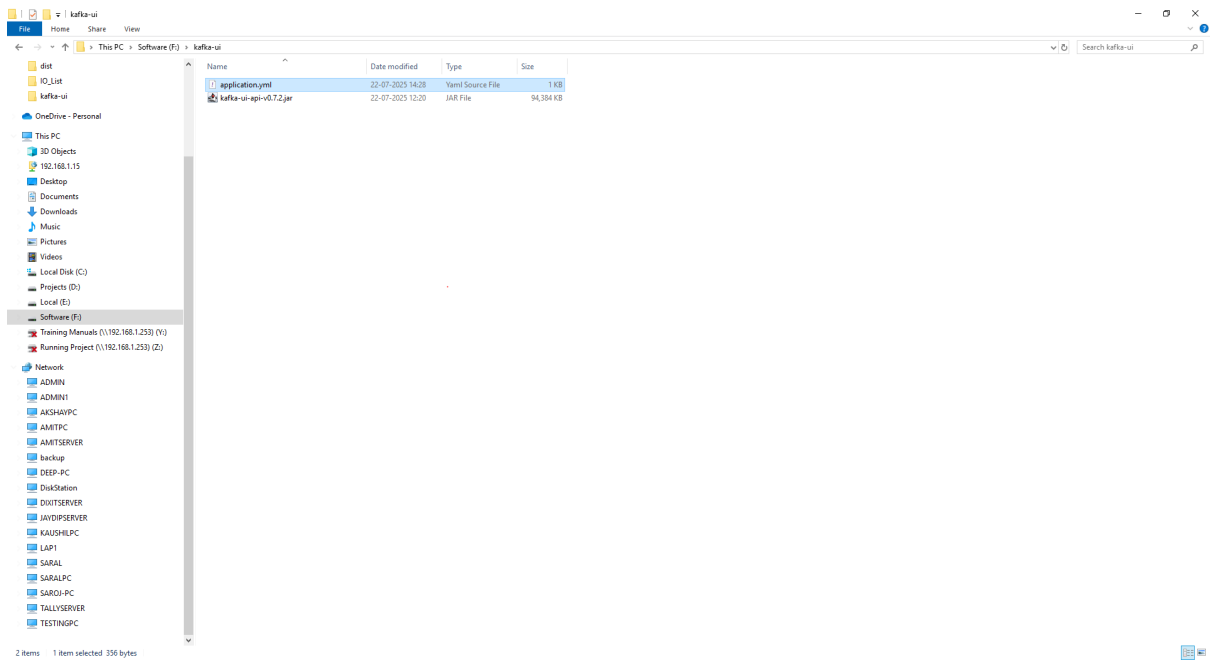
```
Command Prompt - kafka-console-consumer.bat --bootstrap-server 192.168.194.109:9092 --topic PLC --group test-group --from-beginning
Microsoft Windows [Version 10.0.19045.6093]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Dixit>cd /d D:\kafka_2.13-3.6.1\bin\windows

D:\kafka_2.13-3.6.1\bin\windows>kafka-console-consumer.bat --bootstrap-server 192.168.194.109:9092 --topic PLC --group test-group --from-beginning
[2025-07-22 12:02:18,925] WARN [Consumer clientId=console-consumer, groupId=test-group] Bootstrap broker 192.168.194.109:9092 (id: -1 rack: null) disconnected (org.apac
he.kafka.clients.NetworkClient)
[2025-07-22 12:02:36,645] WARN [Consumer clientId=console-consumer, groupId=test-group] Bootstrap broker 192.168.194.109:9092 (id: -1 rack: null) disconnected (org.apac
he.kafka.clients.NetworkClient)
[2025-07-22 12:02:59,011] WARN [Consumer clientId=console-consumer, groupId=test-group] Connection to node -1 (/192.168.194.109:9092) could not be established. Broker m
ay not be available. (org.apache.kafka.clients.NetworkClient)
[2025-07-22 12:02:59,011] WARN [Consumer clientId=console-consumer, groupId=test-group] Bootstrap broker 192.168.194.109:9092 (id: -1 rack: null) disconnected (org.apac
he.kafka.clients.NetworkClient)
[2025-07-22 12:03:20,284] WARN [Consumer clientId=console-consumer, groupId=test-group] Connection to node -1 (/192.168.194.109:9092) could not be established. Broker m
ay not be available. (org.apache.kafka.clients.NetworkClient)
[2025-07-22 12:03:20,284] WARN [Consumer clientId=console-consumer, groupId=test-group] Bootstrap broker 192.168.194.109:9092 (id: -1 rack: null) disconnected (org.apac
he.kafka.clients.NetworkClient)
[2025-07-22 12:03:41,696] WARN [Consumer clientId=console-consumer, groupId=test-group] Connection to node -1 (/192.168.194.109:9092) could not be established. Broker m
ay not be available. (org.apache.kafka.clients.NetworkClient)
[2025-07-22 12:03:41,696] WARN [Consumer clientId=console-consumer, groupId=test-group] Bootstrap broker 192.168.194.109:9092 (id: -1 rack: null) disconnected (org.apac
he.kafka.clients.NetworkClient)
[2025-07-22 12:04:03,657] WARN [Consumer clientId=console-consumer, groupId=test-group] Connection to node -1 (/192.168.194.109:9092) could not be established. Broker m
ay not be available. (org.apache.kafka.clients.NetworkClient)
[2025-07-22 12:04:03,657] WARN [Consumer clientId=console-consumer, groupId=test-group] Bootstrap broker 192.168.194.109:9092 (id: -1 rack: null) disconnected (org.apac
he.kafka.clients.NetworkClient)
{"Date_Time[0]":2025,"Date_Time[1]":7,"Date_Time[2]":22,"Date_Time[3]":11,"Date_Time[4]":59,"Date_Time[5]":53}
{"Date_Time[0]":2025,"Date_Time[1]":7,"Date_Time[2]":22,"Date_Time[3]":12,"Date_Time[4]":8,"Date_Time[5]":53}
{"Date_Time[0]":2025,"Date_Time[1]":7,"Date_Time[2]":22,"Date_Time[3]":12,"Date_Time[4]":1,"Date_Time[5]":53}
{"Date_Time[0]":2025,"Date_Time[1]":7,"Date_Time[2]":22,"Date_Time[3]":12,"Date_Time[4]":2,"Date_Time[5]":53}
{"Date_Time[0]":2025,"Date_Time[1]":7,"Date_Time[2]":22,"Date_Time[3]":12,"Date_Time[4]":3,"Date_Time[5]":53}
{"Date_Time[0]":2025,"Date_Time[1]":7,"Date_Time[2]":22,"Date_Time[3]":12,"Date_Time[4]":4,"Date_Time[5]":53}
{"Date_Time[0]":2025,"Date_Time[1]":7,"Date_Time[2]":22,"Date_Time[3]":12,"Date_Time[4]":5,"Date_Time[5]":53}
{"Date_Time[0]":2025,"Date_Time[1]":7,"Date_Time[2]":22,"Date_Time[3]":12,"Date_Time[4]":6,"Date_Time[5]":53}
{"Date_Time[0]":2025,"Date_Time[1]":7,"Date_Time[2]":22,"Date_Time[3]":12,"Date_Time[4]":7,"Date_Time[5]":54}
```

[illegible][illegible]





ZeroTier Download Documentation API Community Account Networks Logout

[Create A Network](#)

Your Networks

Networks: **1 / 3**
Included Devices: **3 / 10**

Upgrade to Essential to add more networks or devices.

Essential includes unlimited networks, admins, SSO seats, custom routes, and 10 free devices. Additional devices are billed at \$2/month.

[Upgrade to Essential](#)

SEARCH
1 networks...

NETWORK ID	NAME	DESCRIPTION	SUBNET	NODES	CREATED
db64858fedc2ede1	jrdarbar1995		192.168.194.0/24	1	2025-07-22

ZeroTier Download Documentation API Community Account Networks Logout

Networks

jrdarbar1995 Network ID: db64858fedc2ede1 Included Devices: **3 / 10**
Upgrade to Essential for more devices and unlimited networks/admins/sso [Upgrade to Essential](#)

Members

Search all columns... 3 total members 3 filtered members

AUTHORIZATION: All (3) Authorized (3) Not authorized (0) ACTIVITY: All (1) Inactive (1) Active (2)

[Reset Filters](#)

[Refresh](#)

☐	Edit	Auth	Address	Name/Desc	Managed IPs	Last Seen	Version	Physical IP
☐	☒	☒	BE220FF5A0 e2:63:e0:e2:7a:25		192.168.194.178	52 minutes	1.14.0	2401:4900:5555:c1b4:1c04:dcff:fe01:ceF7
☐	☒	☒	C48527597F e2:29:47:cad5fa		192.168.194.36	1 minute	1.14.2	2401:4900:5555:c1b4:cc04:baa1:a5a0:83a1
☐	☒	☒	EABA255462 e2:07:40:cd:b9e7		192.168.194.109	1 minute	1.14.2	122.179.136.231

[E-Mail Join Instructions](#) [Manually Add Member](#)

[Invite](#) [Add New Member](#)

[Use old members list](#)

[Members Help](#)

[Settings](#)

On Both PCs:

- Kafka and Zookeeper installed (PC1: Broker, PC2: Client/Consumer)
- Java JDK installed
- Kafka configured and tested locally
- ZeroTier One installed

PC1 (Kafka Broker) – Configuration Steps

1. Install & Join ZeroTier Network

- Install ZeroTier One
- Launch the app → copy the **Node ID**
- Go to <https://my.zerotier.com> → Create an account
- Create a **new network**
- Add PC1's **Node ID** to the network → click “Authorize”
- Take note of the **Managed IP address** (e.g., 10.147.20.2)

2. Configure `server.properties`

Edit Kafka's config file:

```
properties
CopyEdit
# Path: kafka/config/server.properties
advertised.listeners=PLAINTEXT://10.147.20.2:9092
listeners=PLAINTEXT://0.0.0.0:9092
```

3. Restart Kafka & Zookeeper

```
bash
CopyEdit
zookeeper-server-start.bat ..\..\config\zookeeper.properties
kafka-server-start.bat ..\..\config\server.properties
```

PC2 (Kafka Consumer or Client) – Configuration Steps

1. Install & Join Same ZeroTier Network

- Install ZeroTier One
- Join the same **network ID**
- Authorize PC2 from ZeroTier dashboard
- Note its **Managed IP** (e.g., 10.147.20.3)

2. Test ZeroTier Connection

From PC2, run:

```
bash
CopyEdit
ping 10.147.20.2
```

If response comes, ZeroTier is working.

3. Run Kafka Console Consumer

```
bash
```

```
CopyEdit
cd /d C:\kafka_2.13-3.6.1\bin\windows
kafka-console-consumer.bat --bootstrap-server 10.147.20.2:9092 --topic
your-topic-name --from-beginning
```

If you're using SSL:

```
bash
CopyEdit
kafka-console-consumer.bat --bootstrap-server 10.147.20.2:9093 --topic
your-topic-name --consumer.config config
```