Object Serialization Server

Description
In this problem, you are required to implement a simple server that handles object serialization and deserialization. The server listens for incoming connections, accepts messages from clients, and processes these messages by deserializing them. The messages are compressed using zlib and serialized using JSON.

Input
The input will be a compressed and serialized JSON message containing the following fields:
- username: A string representing the username of the sender.
- text: A string representing the message text.
- timestamp: A string representing the timestamp of the message in the format '%Y-%m-%d %H:%M:%S.%f'.

Implementation
You need to implement the following methods:
- Message.serialize(self): This method should serialize the Message object into a compressed JSON string.
- Message.deserialize(serialized_message): This method should deserialize the compressed JSON string back into a Message object.
- Ensure your deserialization function handles the decompression and decoding correctly.
- You may use Python's standard libraries json, zlib, and datetime for this task.

Output (with unit test)

test attribute passed: Received message: found in log messages
test attribute passed: Username: Alice found in log messages
test attribute passed: Text: Hello, World! found in log messages