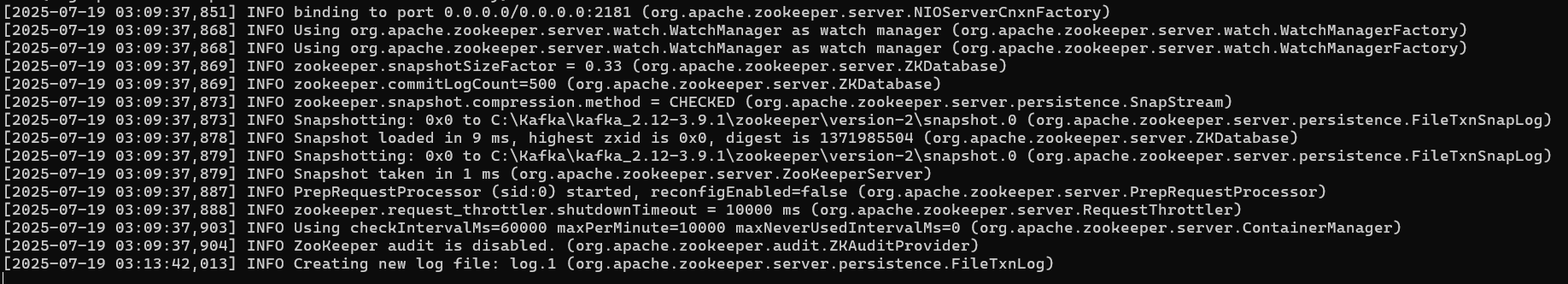
**Hands-on: Kafka Integration with C#**

**Step 1:** Create a Chat Application which uses Kafka as a streaming platform and consume the chat messages in the command prompt.

Code:

zookeeper-server-start.bat C:\Kafka\kafka\_2.12-3.9.1\config\zookeeper.properties

Output:

  
Code:

kafka-server-start.bat C:\Kafka\kafka\_2.12-3.9.1\config\server.properties

Output:

  
Code:

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

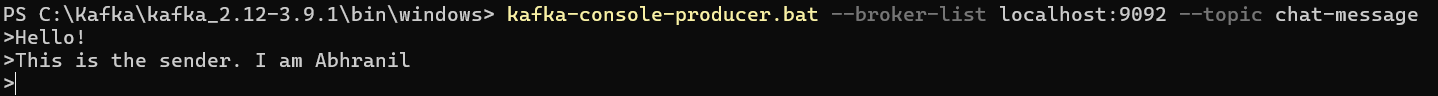
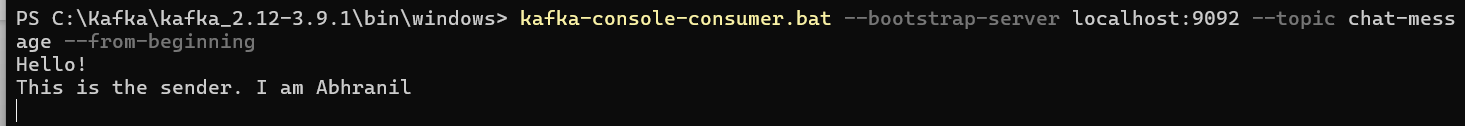
Output:

  
Code:

kafka-console-consumer.bat --bootstrap-server localhost:9092 --topic chat-message --from-beginning

kafka-console-producer.bat --broker-list localhost:9092 --topic chat-message

Output:

**Step 2:** Create a Chat Application using C# Windows Application using Kafka and consume the message in different client applications.

Code:

public partial class Form1 : Form  
{  
 Uri uri = new Uri("http://localhost:9092");  
 string topic = "chat-message";  
 public Form1()  
 {  
 InitializeComponent();  
 }  
   
 private void btnSend\_Click(object sender, EventArgs e)  
 {  
 if (txtMessage.Text == string.Empty)  
 {  
 MessageBox.Show("Please Enter Message", "Warning",  
MessageBoxButtons.OK, MessageBoxIcon.Warning);  
 return;  
 }  
   
 string payload = txtMessage.Text.Trim();  
 var sendMessage = new Thread(() =>  
 {  
 KafkaNet.Protocol.Message msg = new  
KafkaNet.Protocol.Message(payload);  
 var options = new KafkaOptions(uri);  
 var router = new BrokerRouter(options);  
 var client = new Producer(router);  
 client.SendMessageAsync(topic, new List<KafkaNet.Protocol.Message> { msg }).Wait();

});   
 sendMessage.Start();  
 this.Clear();  
 }

private void Clear()  
 {  
 txtMessage.Text = string.Empty;  
 }  
   
 private void btnCancel\_Click(object sender, EventArgs e)  
 {  
 this.Close();  
 }  
}

Output:

