Oregon State University Ecampus

CS361: Assignment 4: Microservices Case Study & Pipe Spike

Overview

Learn how microservices work in the real world by (1) researching a software product that uses the microservices architecture or provides microservices and (2) implementing a microservices communication approach that is NOT text files.

Instructions

Complete each item below by replacing the highlighted text (**Usability note**: double-click the text to select it).

1. **PART 1: Microservices Case Study**

   Find **well-known software or technology** that uses the **microservices** architecture (e.g., Netflix, Amazon, etc.) or that provides multiple microservices to others. **Research** the software/technology and **answer the following questions**.

   a. What is the **name of the software or technology** and **what is it for**?

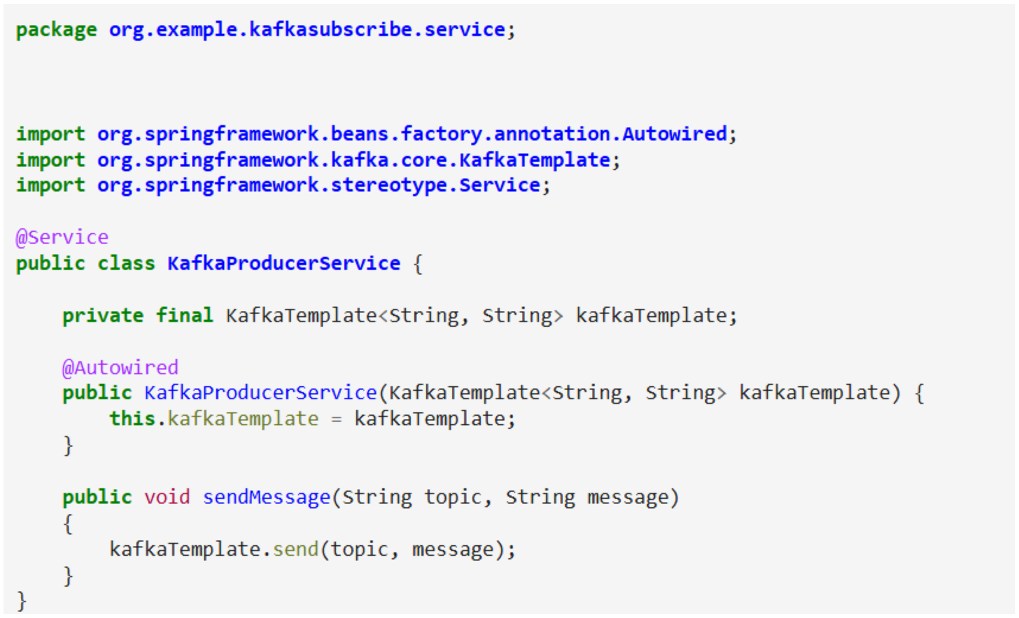
   Netflix: a video streaming platform offering movies, TV shows, and original content.

   b. **Why** were microservices used for this software or technology?

   Netflix moved to microservices because their old system couldn't keep up as they got bigger and needed to handle tons of users all over the world. With microservices, they could scale things separately, so the whole system didn't crash if one part had issues. It also made it way faster to roll out new features since teams could work on their own stuff without waiting on others.

   c. How can the program you listed above **send a message to the microservices**? Explain for at least one microservice. If you're not sure the name of the communication approach, explain as best you can and, if possible, provide a screenshot of example code that sends a message to the microservice. "An API" is not enough detail.

   Netflix uses asynchronous communication to send messages between microservices. One common method is through RESTful APIs or message brokers like Apache Kafka, depending on the service's needs.



```java
package org.example.kafkasubscribe.service;


import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.kafka.core.KafkaTemplate;
import org.springframework.stereotype.Service;

@Service
public class KafkaProducerService {

    private final KafkaTemplate<String, String> kafkaTemplate;

    @Autowired
    public KafkaProducerService(KafkaTemplate<String, String> kafkaTemplate) {
        this.kafkaTemplate = kafkaTemplate;
    }

    public void sendMessage(String topic, String message)
    {
        kafkaTemplate.send(topic, message);
    }
}
```

d. Name and **describe a few microservices** that are part of the software/technology. (3+ microservices)

Streaming Service: It takes care of things like encoding the video, adjusting the quality based on your internet speed, and making sure the video plays smoothly on any device you're using.

Recommendation Engine: It looks at what you've watched and uses machine learning to recommend movies and shows you'll probably enjoy, making your Netflix experience more for you.

Billing Service: It processes your payments, handles subscription upgrades, and makes sure you're charged correctly for your Netflix plan.

e. List your **sources of information**. Provide enough information so that your grader can determine what the source is. If you used online sources, provide links.

- *https://stackoverflow.com/questions/50506101/spring-boot-how-to-communicate-between-microservices*
- *https://blog.dreamfactory.com/microservices-examples#:~:text=Lightweight%20Communication:%20Microservices%20communicate%20with,scaled%20independently%2C%20without%20tight%20coupling*.
- *https://www.techaheadcorp.com/blog/design-of-microservices-architecture-at-netflix/*
- *https://www.geeksforgeeks.org/system-design-netflix-a-complete-architecture/*
- *https://dev.to/gbengelebs/netflix-system-design-backend-architecture-10i3*

2. **PART 2: Pipe Spike**

   Spike one microservices communication approach that is NOT communication via text file (since you already tried that). **Write two programs**: One that sends messages, another that receives those messages.

   ZeroMQ is highly recommended to try out, an intro resource is provided on the Canvas Assignment 4 page.

   Example approaches:
   - ZeroMQ
   - RabbitMQ
   - HTTP Request

   You are NOT limited to the list above.

   Requirements for the approach you choose:

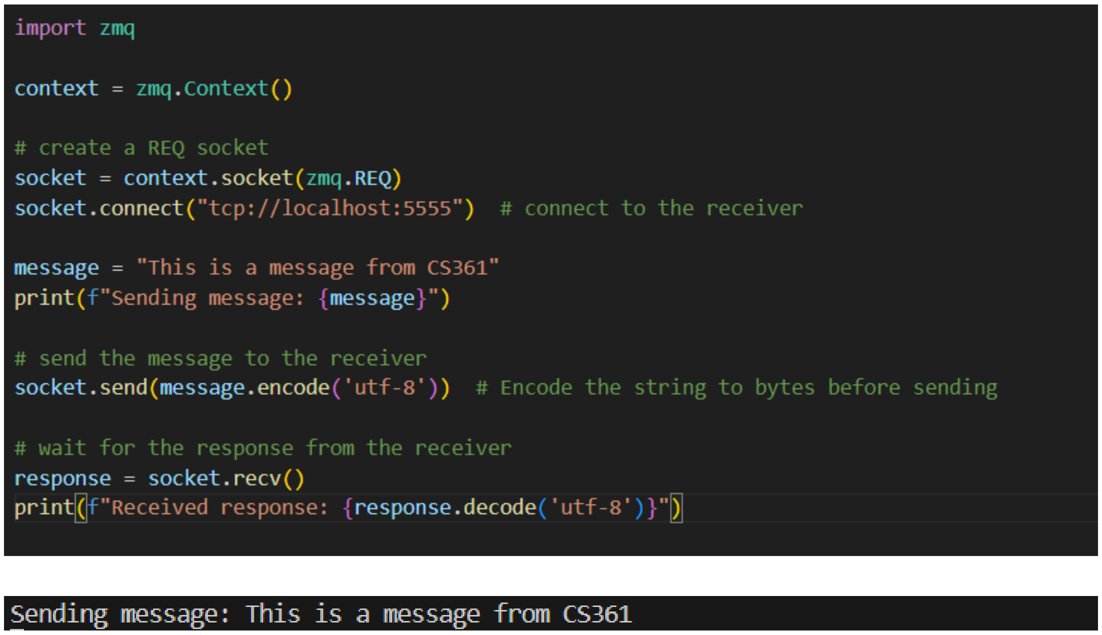
   - Can be used to communicate between processes
   - Can be used to request and provide data
   - Not text files, CSV files, or other similar approaches involving file reads/writes
   - You're allowed to learn from tutorial code and other code you find online, but you're still required to write all your own code (so that you will understand it).

   Complete the following:

   a. **Which approach** did you spike?

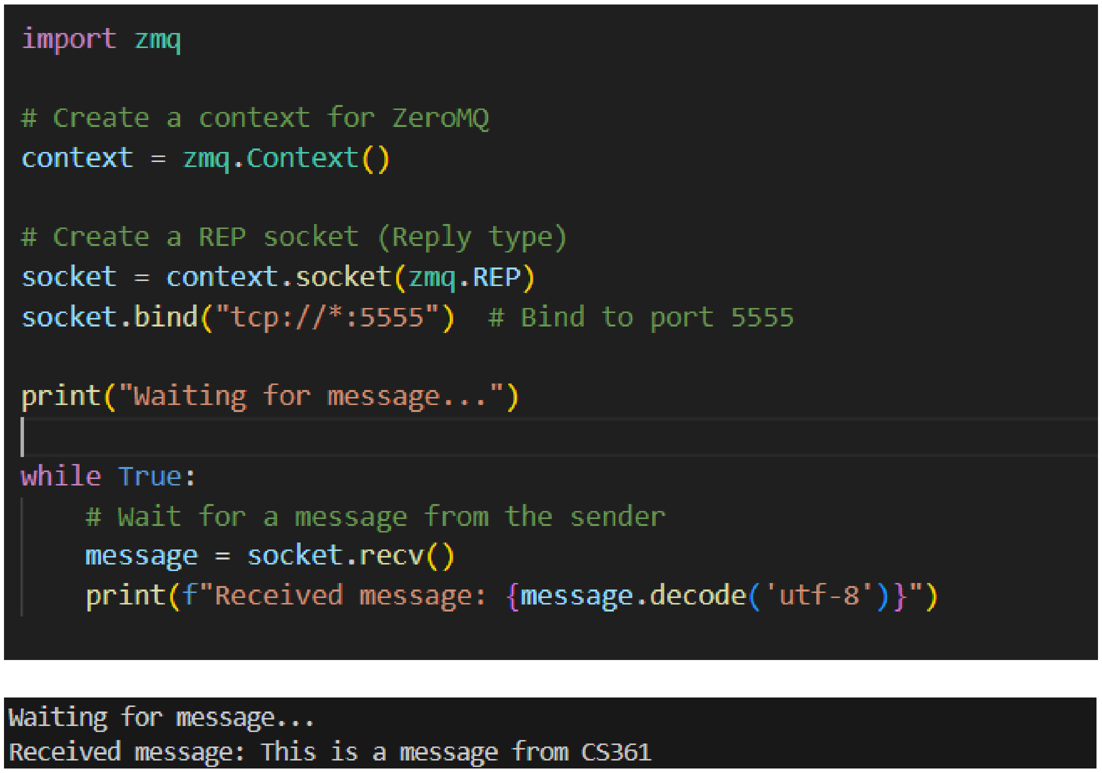
      ZeroMQ

   b. Get the approach working. Upload **screenshots** that show the approach being used to **send and receive this EXACT message: ``This is a message from CS361''**. The message must be sent from a program you wrote and received by a different program you wrote.



```python
import zmq

context = zmq.Context()

# create a REQ socket
socket = context.socket(zmq.REQ)
socket.connect("tcp://localhost:5555")  # connect to the receiver

message = "This is a message from CS361"
print(f"Sending message: {message}")

# send the message to the receiver
socket.send(message.encode('utf-8'))  # Encode the string to bytes before sending

# wait for the response from the receiver
response = socket.recv()
print(f"Received response: {response.decode('utf-8')}")
```

```
Sending message: This is a message from CS361
```



```python
import zmq

# Create a context for ZeroMQ
context = zmq.Context()

# Create a REP socket (Reply type)
socket = context.socket(zmq.REP)
socket.bind("tcp://*:5555")  # Bind to port 5555

print("Waiting for message...")

while True:
    # Wait for a message from the sender
    message = socket.recv()
    print(f"Received message: {message.decode('utf-8')}")
```

```
Waiting for message...
Received message: This is a message from CS361
```

Submission

PDF or Word format via Canvas.

Grading

You are responsible for satisfying all criteria listed in the Canvas rubric for this assignment.

Questions?

Please ask via Ed so that others can benefit from the answers.