



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 is an online streaming platform that allows users to watch movies, TV shows, and documentaries.

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

Netflix has used microservices to handle massive scalability requirements, reduce risks of deployment, enable continuous deployment, and isolate failures. They have broken down their services into independently deployable units which allows them to create new technologies quickly, scale efficiently, and recover quickly from failures.

- 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.

I believe that Netflix uses RESTful HTTP APIs for communication between their microservices. A typical microservice interaction might involve one service making an HTTP request to another service to retrieve user recommendations. For example:

```
fetch('https://api.netflix.com/recommendations/user/1234', {
  method: 'GET',
  headers: { 'Authorization': 'Bearer TOKEN' }
})
.then(response => response.json())
.then(data => console.log(data));
```

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

- 1) Recommendations Microservice: Personalized movie and show recommendations are generated based on user history and preferences
- 2) Content Delivery Microservice: Manages efficient streaming of video content, ensuring quality and minimal latency.
- 3) User Authentication Microservice: Handles user log, authentication, and authorization securely.

- 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.

These are the sources I used:

<https://roshanccloudarchitect.me/understanding-netflixs-microservices-architecture-a-cloud-architect-s-perspective-5c345f0a70af>

<https://netflixtechblog.com/tagged/microservices>

<https://www.techaheadcorp.com/blog/design-of-microservices-architecture-at-netflix/>

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?

I spiked the ZeroMQ approach

- 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.

sender.py U X

assignments > assignment4 > sender.py > ...

```
1 import zmq
2
3 # Create context and socket
4 context = zmq.Context()
5 socket = context.socket(zmq.REQ)
6 socket.connect("tcp://localhost:5555")
7
8 # Send message
9 message = "This is a message from CS361"
10 print(f"Sending message: {message}")
11 socket.send_string(message)
12
13 # Wait for reply (acknowledgement)
14 reply = socket.recv_string()
15 print(f"Received reply: {reply}")
```

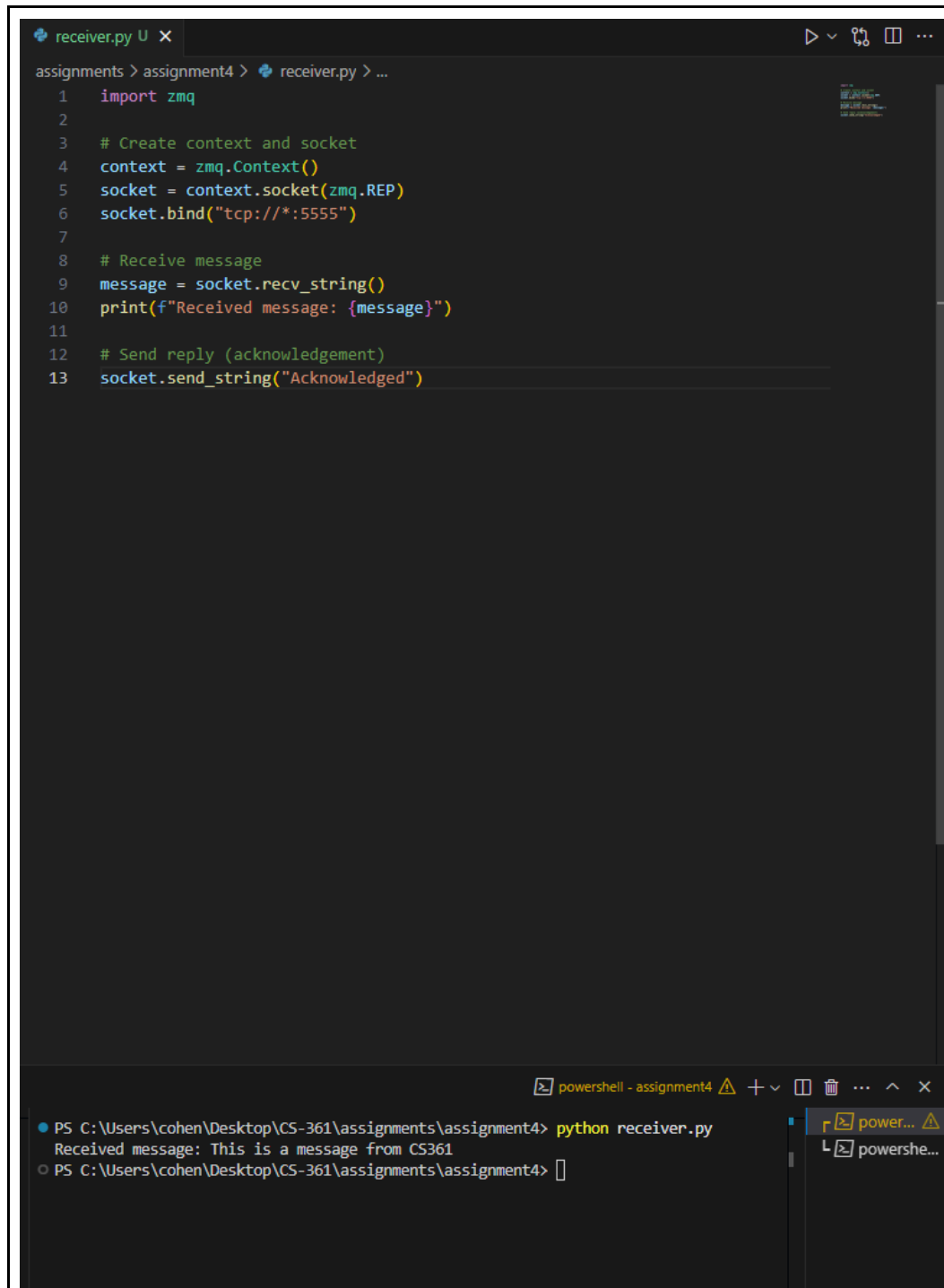
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

PS C:\Users\cohen\Desktop\CS-361\assignments\assignment4> python sender.py

Sending message: This is a message from CS361

Received reply: Acknowledged

PS C:\Users\cohen\Desktop\CS-361\assignments\assignment4>



The image shows a code editor window with a file named `receiver.py` open. The code is a Python script using the `zmq` library to create a socket, bind it to `tcp://*:5555`, receive a message, and send an acknowledgment. Below the code editor is a terminal window showing the execution of the script. The terminal output shows the received message: "This is a message from CS361".

```
receiver.py U x
assignments > assignment4 > receiver.py > ...
1  import zmq
2
3  # Create context and socket
4  context = zmq.Context()
5  socket = context.socket(zmq.REP)
6  socket.bind("tcp://*:5555")
7
8  # Receive message
9  message = socket.recv_string()
10 print(f"Received message: {message}")
11
12 # Send reply (acknowledgement)
13 socket.send_string("Acknowledged")
```

powershell - assignment4 ⚠ + ▢ 🗑 ... ^ x

- PS C:\Users\cohen\Desktop\CS-361\assignments\assignment4> python receiver.py
Received message: This is a message from CS361
- PS C:\Users\cohen\Desktop\CS-361\assignments\assignment4>

power... ⚠
powershe...

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.