



## 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 and (2) implementing a microservices communication pipe 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** that uses the **microservices** architecture (e.g., Netflix, Amazon, etc.). **Research** the software and answer the following **questions**.

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

The name of the software is Amazon, and it is a multi-wide store applications for individuals to be able to purchase items online and the items will be delivered to your door.

- b. **Why** was the microservices architecture used for this software?

The microservices architecture system used in this software was used to make shopping functions simpler, and less complex to maintain and keep track information.

- c. How does **communication** happen between the software's microservices?

It first starts with the user's interaction with the software, then sends signals to a holder, which determines which microservices it needs to use. Then the microservice is running, and sends result back to holder, and holder sends the feedback to the user.

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

Amazon CloudWatch – this program collects and tracks the log files of logs, alarms, etc.

AWS Lambda – A method of running codes without having to worry about keeping track of servers.

Amazon S3 – A method of storing work data.

- 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.

- What are microservices? | AWS. (n.d.).  
<https://aws.amazon.com/microservices/>

## 2. PART 2: Pipe Spike

Spike one microservices communication approach that is NOT communication via text file (since you already tried that). 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 message: ``A message from CS361``**. The message must be sent from a program you wrote and received by a different program you wrote.

```
BindMessage.py
context = zmq.Context()
# Display * to user what is happening.
print("Connecting to the cs-361 message server...")
socket = context.socket(zmq.REQ)
socket.connect("tcp://localhost:1111")

# Do 5 request, waiting for a response each time.
for request in range(5):
    print(f"sending request {request} ...")
    socket.send_pyb("A message from CS361")

# Give time by 2.5
time.sleep(2.5)

# Recieves the reply
message = socket.recv()
print(f"Received reply {request} [{message}]")

RequestFile.py
context = zmq.Context()
print("Connecting to the cs-361 message server...")
socket = context.socket(zmq.REQ)
socket.connect("tcp://localhost:1111")

# Send 5 request, waiting for a response each time.
for request in range(5):
    print(f"sending request {request} ...")
    socket.send_pyb("A message from CS361")

# Give time by 2.5
time.sleep(2.5)

# Recieves the reply
message = socket.recv()
print(f"Received reply {request} [{message}]")
```

