



## CS361: Assignment 5: 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**? (1+ sentences)

Amazon was one of the first major tech companies that successfully implemented the microservices architecture. With the advent of AWS in 2007, Amazon has since implemented the architecture to many of its services including EC2.

- b. **Why** was the microservices architecture used for this software? (2+ sentences)

The monolith architecture Amazon previously had was architected in multiple tiers, and those tiers had many components in them, but they were all very tightly coupled together, where they behaved like one big monolith. Over time, as the company grows, as you add more developers to it, the code base gets larger and the architecture gets more complex, that monolith is going to add overhead to your process, and that software development lifecycle is going to begin to slow down. Hence, Amazon decided to alternatively use the microservices architecture.

- c. How does **communication** happen between the software's microservices? (2+ sentences)

The solution to the single-purpose function problem was the creation of a rule, to be adhered to by developers, that functions could only communicate with the rest of the world through their own web service APIs. With a microservices approach, the software is composed of small services that communicate over well-defined application programming interfaces (APIs) that can be deployed independently. These services are owned by small autonomous teams. Many of these communications involve services like HTTP requests, FTP, etc.

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

1. **AWS EC2 - Elastic Compute Cloud (EC2)** allows users to rent virtual computers on which to run their own computer applications.

2. **AWS Lambda - Lambda** is an event-driven, serverless computing platform. It is a computing service that runs code in response to events and automatically manages the computing resources required by that code.
3. **AWS Fargate** - Fargate is a serverless compute engine for containers that work with both Amazon Elastic Container Service (ECS) and Amazon Elastic Kubernetes Service (EKS).
4. **AWS RDS - Relational Database Service** is a distributed relational database service. It is a web service running "in the cloud" designed to simplify the setup, operation, and scaling of a relational database for use in applications.

## 2. PART 2: Pipe Spike

Spike one microservices communication approach that is NOT communication via a text file (since you already tried that). Example approaches:

- [RabbitMQ](#)
- Subprocess calls
- [Sockets](#)
- PyZMQ Messaging
- RPyC: Remote Python Call library
- HTTP Request
- os library: system calls
- [ZeroMQ](#)
- FTP

You are NOT limited to the approaches above.

### Requirements for the approaches:

- 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

### Complete the following:

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

Remote Python Call library (RPyC)

- b. Get the approach working. Upload **screenshots** that show the approach being used to **send and receive this message: ``A message from CS361``**

On tab 1 of my terminal:

```
$ rpyc_classic.py
INFO:SLAVE/18812:server started on [127.0.0.1]:18812
INFO:SLAVE/18812:accepted ('127.0.0.1', 57523) with fd 628
```

On tab 1 of my terminal:

```
INFO:SLAVE/18812:accepted ('127.0.0.1', 57523) with fd 628
INFO:SLAVE/18812:welcome ('127.0.0.1', 57523)
INFO:SLAVE/18812:goodbye ('127.0.0.1', 57523)
```

**On tab 2 of my terminal:**

```
$ python3 rpyc_modules.py localhost
3.10.5 (tags/v3.10.5:f377153, Jun 6 2022, 16:14:13) [MSC v.1929 64 bit (AMD64)]
A message from CS361
```

## Citation

Link used for reference: <https://code-maven.com/rpc-with-python-using-rpyc>

## Submission

**PDF or Word** format via Canvas.

## Grading

You are responsible for satisfying all criteria listed in the Canvas rubric for this assignment. You will be able to revise this assignment if you miss points.

## Questions?

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