Order Placement:

- A user places an order via Orders Service (Server-1).
- Orders Service calls Customer Service (Server-12) to validate customer details.
- Once validated, Orders Service saves the order details in the database.

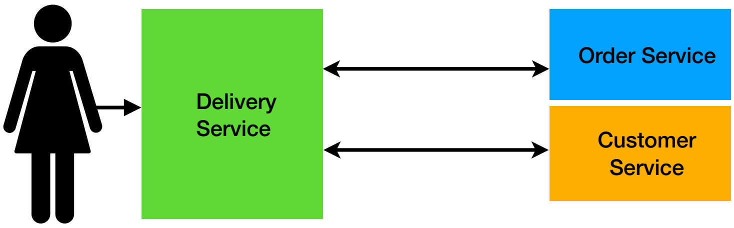
2. Delivery Assignment:

- Delivery Service (Server-2) queries Orders Service to get pending orders.
- It also queries Customer Service (Server-12) to get the customer address.
- Once the delivery is scheduled, it updates Orders Service.

. Delivery Assignment:

- Delivery Service (Server-2) queries Orders Service to get pending orders.
- It also queries Customer Service (Server-12) to get the customer address.
- Once the delivery is scheduled, it updates Orders Service.

Architecture of the micro services:



Below is the ansible playbook deploys three micro services:

```
---
- name: Deploy Microservices
  hosts: all
  become: yes
  tasks:

  - name: Deploy Orders Service on Server-1
    hosts: server-1
    tasks:
      - name: Copy Orders Service JAR
        copy:
          src: orders-service.jar
          dest: /opt/microservices/orders-service.jar
      - name: Start Orders Service
        command: java -jar /opt/microservices/orders-service.jar
        async: 30
        poll: 0
      - name: Ensure Orders Service is running
        wait_for:
          port: 8081
          timeout: 30

  - name: Deploy Customer Service on Server-12
    hosts: server-12
    tasks:
      - name: Copy Customer Service JAR
        copy:
          src: customer-service.jar
          dest: /opt/microservices/customer-service.jar
      - name: Start Customer Service
        command: java -jar /opt/microservices/customer-service.jar
        async: 30
        poll: 0
      - name: Ensure Customer Service is running
        wait_for:
          port: 8082
          timeout: 30

  - name: Deploy Delivery Service on Server-2
    hosts: server-2
    tasks:
      - name: Copy Delivery Service JAR
        copy:
          src: delivery-service.jar
          dest: /opt/microservices/delivery-service.jar
      - name: Start Delivery Service
        command: java -jar /opt/microservices/delivery-service.jar
        async: 30
        poll: 0
      - name: Ensure Delivery Service is running
        wait_for:
          port: 8083
          timeout: 30
```

Environment Configuration

- server-1 runs Orders Service on port 8081.
- server-3 runs Customer Service on port 8082.
- server-2 runs Delivery Service on port 8083.
- The Orders Service calls
  http://server-12:8082/customer/{id} for customer details.
- The Delivery Service calls:
  http://server-1:8081/orders/{id} to fetch orders.
  http://server-12:8082/customer/{id} to get the customer address.