National College of Ireland

Configuration Manual

MSc Research Project
Cloud Computing

Soumya Mohanan
Student ID: x23104767

School of Computing
National College of Ireland

Supervisor: Yasantha Samarawickrama

National College of Ireland
Project Submission Sheet
School of Computing

Student Name:	Soumya Mohanan
Student ID:	x23104767
Programme:	Cloud Computing
Year:	2023-2024
Module:	MSc Research Project
Supervisor:	Yasantha Samarawickrama
Submission Due Date:	12/08/2024
Project Title:	Configuration Manual
Word Count:	793
Page Count:	8

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.

ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature:	Soumya Mohanan
Date:	12th August 2024

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST:

Attach a completed copy of this sheet to each project (including multiple copies).	☐
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	☐
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	☐

Assignments that are submitted to the Programme Coordinator office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Configuration Manual

Soumya Mohanan
x23104767

1 Create AWS EC2 instances for the Microservices

1. Sign in to the AWS Management Console and open the Amazon EC2 console at https://console.aws.amazon.com/ec2 and click on launch instance.

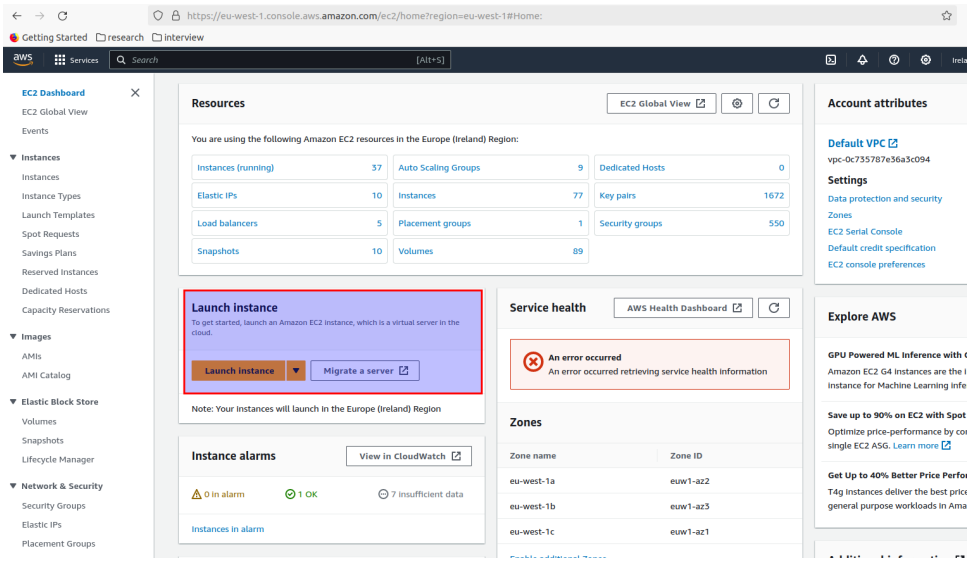

Figure 1: AWS EC2 Create Instance Page

2. Choose an Amazon Machine Image (AMI) and Instance Type: Amazon Linux 2 AMI and t2.micro instance type eligible for the AWS free tier.

   - **Network Settings:**
     - Select default VPC.
     - Subnet preference set to "No preference."
     - Auto-assign public IP enabled.
     - New security group created with the following rules:
       * Allow SSH traffic from anywhere (0.0.0.0/0).
       * Allow HTTPS traffic from the internet (0.0.0.0/0).
       * Allow HTTP traffic from the internet (0.0.0.0/0).

3. Create a new key pair with the following options selected:
   Key pair type : RSA
   Private key format : .pem

   Store the private key in secured place for establishing secure connection to the EC2 instances in later procedures.

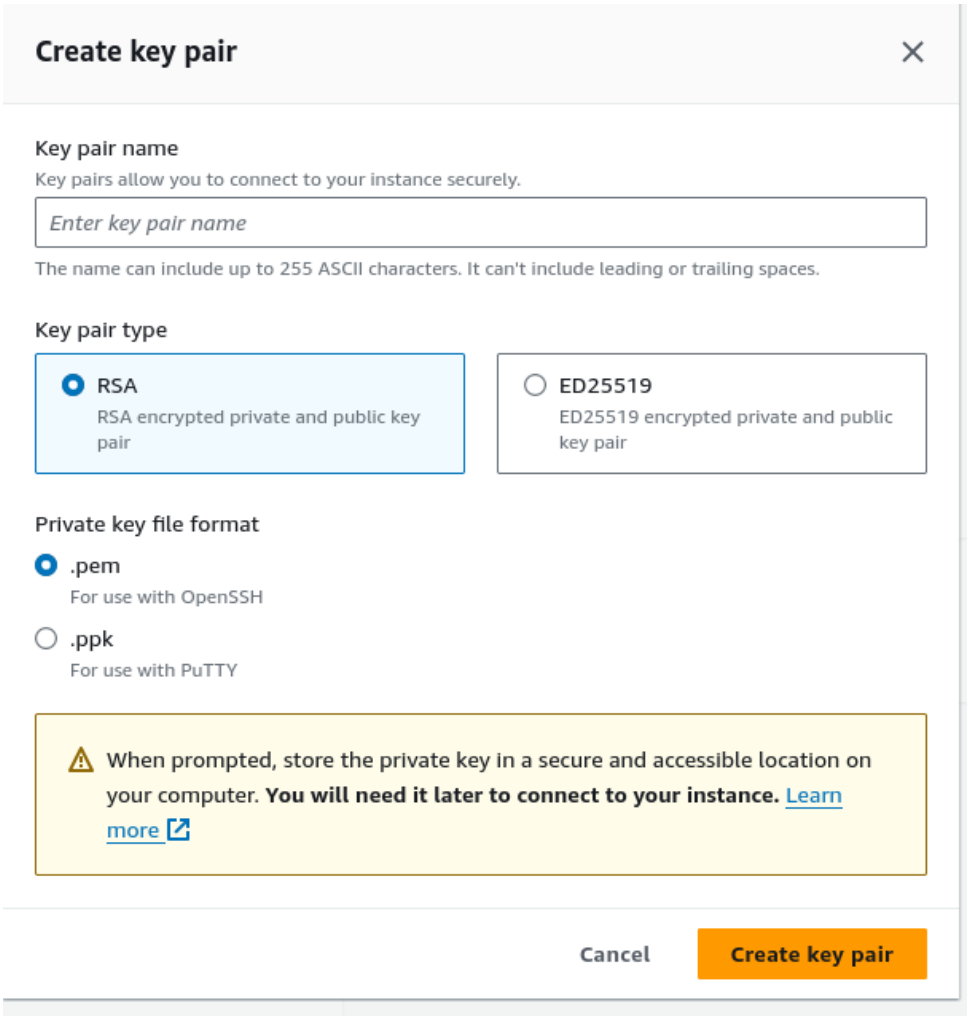

Figure 2: Create New key pair

4. After these settings are performed click on "Launch Instance".

2 Change the newtork ACL inbound and outbound rules

Change the newtork ACL rules for ec2 instance to allow traffic from other microservice ports and external traffic by selecting the appropriate Network ACL. Add rules by editing "Inbound Rules" and "Outbound Rules" tabs as shown in the image for port 8080, 8083 and 8761.

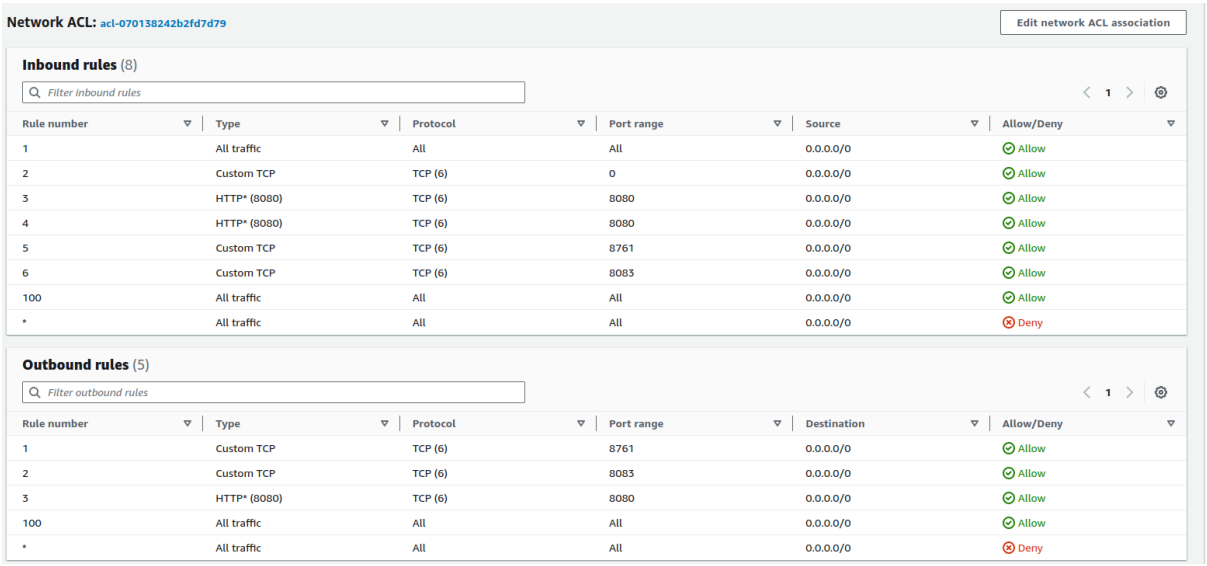

Figure 3: Inbound and outbound rules required

3 Create JAR File and Transfer to the EC2 instance for the eureka server

Copy the application code from the code artifacts to the EC2 instance using the below steps:

1. Create a jar file for the application using the command:

```
mvn clean package
```

2. Change the permissions of the `.pem` file downloaded earlier when the new key pair was created:

```
chmod 400 /path/to/your-key.pem
```

3. Securely transfer the JAR file to the EC2 Instance using SCP:

```
scp -i /path/to/your-key.pem /path/to/local/file.jar ec2-user
<EC2-Public-IP>:/remote/directory
```

4 Start the Eureka Server First

1. Connect to the EC2 Instance created earlier using SSH:

```
ssh -i /path/to/your-key.pem ec2-user@<EC2-Public-IP>
```

2. Install OpenJDK

```
sudo amazon-linux-extras install java-openjdk11 -y
```

3. Install Maven

```
    sudo yum install maven -y
```

4. Navigate to the folder where the jar file was copied.

5. Start the Spring Boot Eureka server on the EC2 Instance using the below command:

```
java -jar /remote/directory/file.jar
```

6. The eureka server will start on port 8761

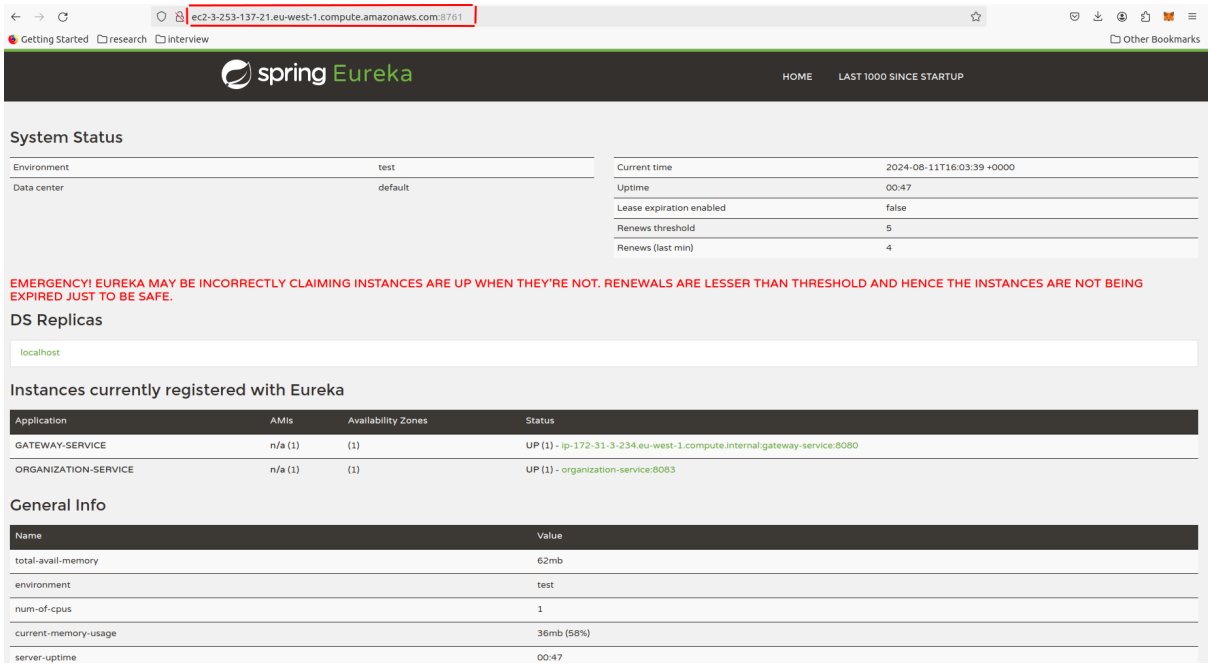

Figure 4: Eureka Dashboard

5 Run the Zuul Gateway and Organisation Service

Follow the same steps as in Step 1 to Step 4 for starting the Eureka service with the respective application code and start the service.

- The Zuul Gateway will start on port 8080, accessible at:

```
http://<ec2_address>:8080
```

- The Organisation service will start on port 8083, accessible at:

```
http://<ec2_address>:8083
```

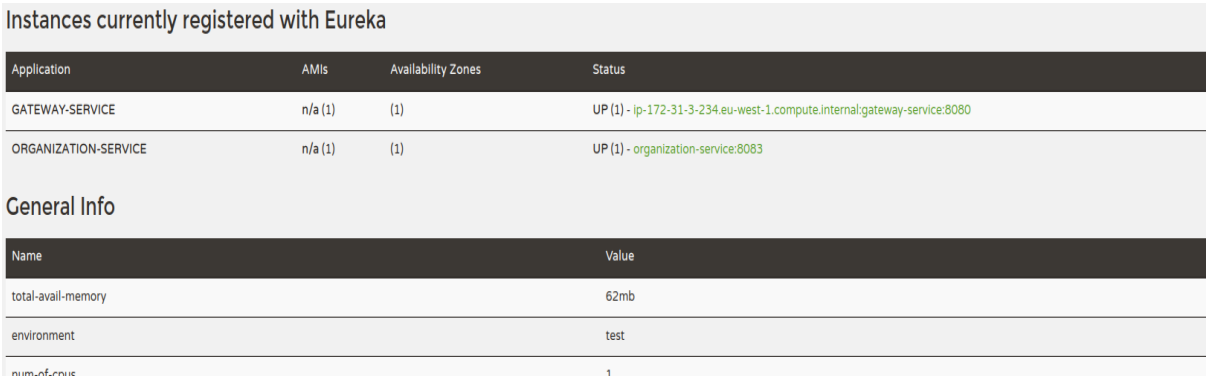

Figure 5: Services Registered on Eureka Dashboard

6 Load Testing using the Jmeter tool

Step 1: Install JMeter

1. Download JMeter from the official Apache JMeter website.
2. Extract the downloaded archive.
3. Launch JMeter by running the `jmeter` script in the `bin` directory.

Step 2: Create a Test Plan

1. Start JMeter by running the `jmeter` script in the `bin` directory.
2. Add a Test Plan

Step 3: Add Thread Group

1. Thread Group: Right-click on the Test Plan, then choose Add Threads (Users).
2. Configure Thread Group:
   - **Number of Threads (Users)**: Set the number of virtual users you want to simulate. 5000 in this case.
   - **Ramp-Up Period**: The time JMeter should take to start all the users.
   - **Loop Count**: Set how many times to execute the test.

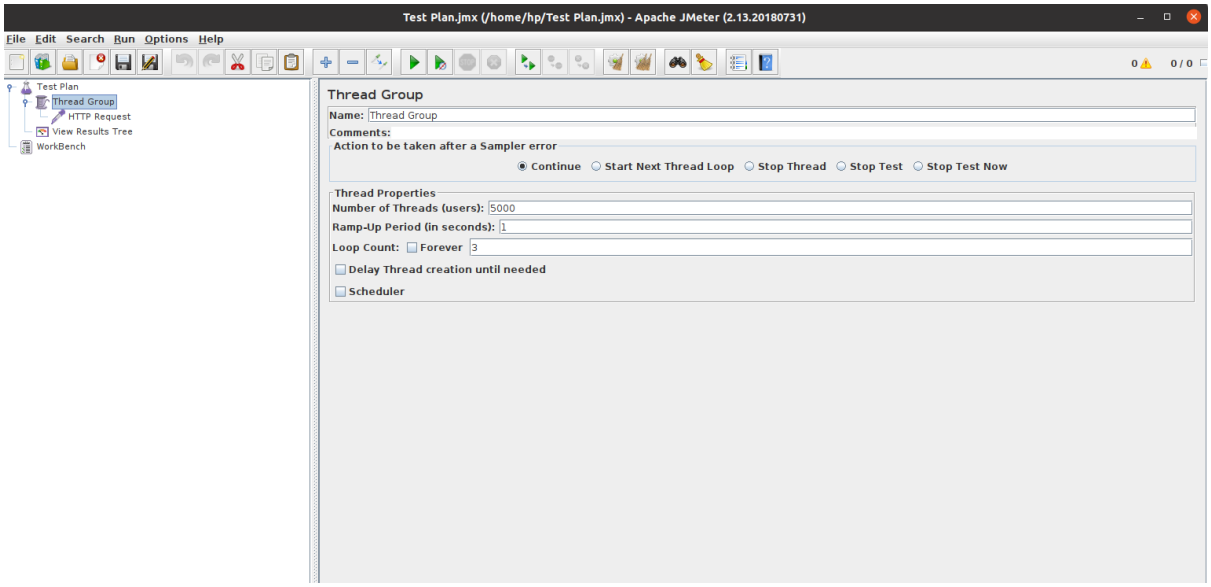

Figure 6: Thread Group settings

Step 4: Add HTTP Request Sampler

1. HTTP Request: Right-click on the Thread Group, then choose `Add > Sampler > HTTP Request`.

2. Configure HTTP Request:

   - **Server Name or IP**: Enter the domain name or IP address of the server.
   - **Path**: Specify the API endpoint here it is `/api/organisations/1`.
   - **Method**: Choose the HTTP method - GET

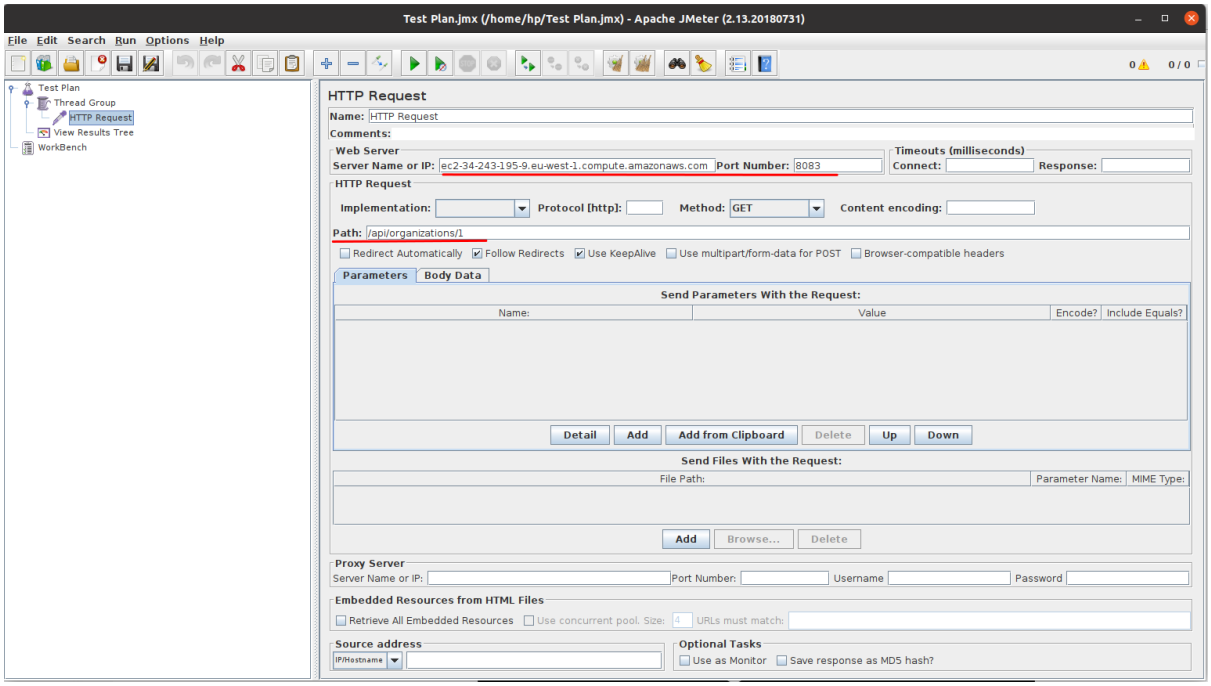

Figure 7: Request Sampler settings

Step 5: Add Listeners

1. View Results: Right-click on the Thread Group, then choose `Add > Listener`.
2. Listeners:
   - **View Results Tree**: Provides detailed logs of each request.

Step 6: Run the Test

1. Click on the green `Play` button in the JMeter toolbar to begin the load test.

7 Custom Logging of the dynamic heartbeat interval

Custom logging can be checked on the log details of the ec2 instance on which the service is running.

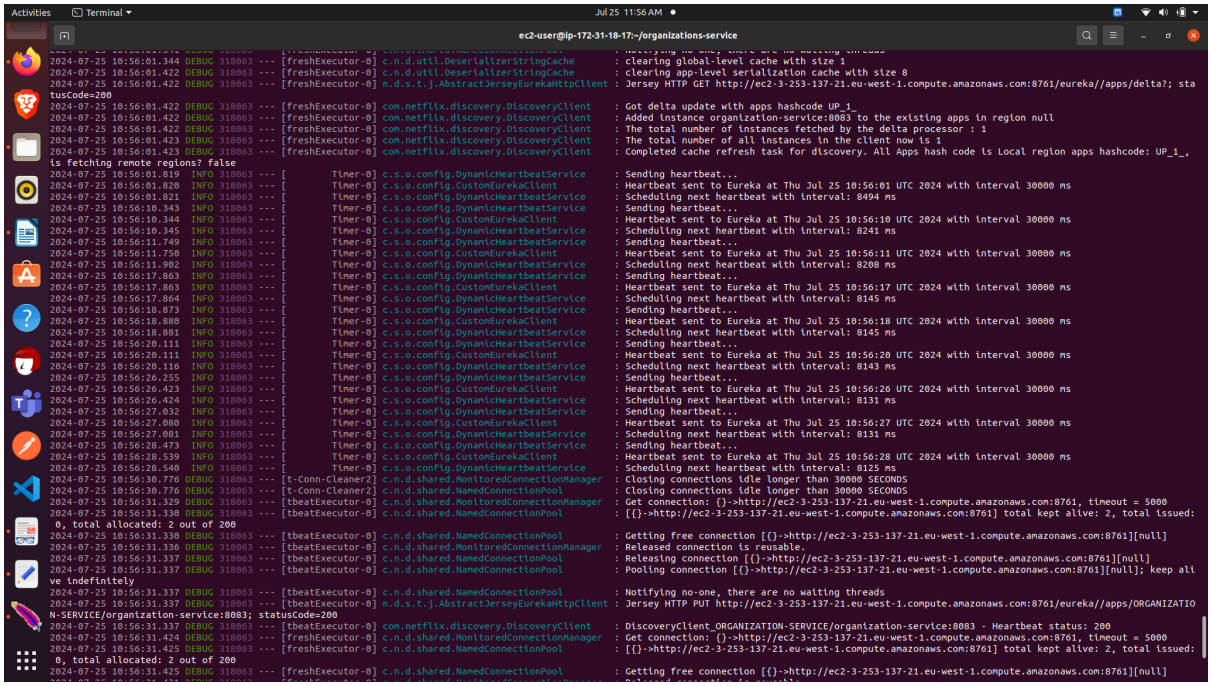

Figure 8: Dynamic heartbeat interval logs