

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.

<u>ALL</u> 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.

← → C O A https://eu-west-1.console.aws.amazon.com/ec2/home?region=eu-west-1#Home:	☆
aws i iii services Q. Search [Alt+5]	3 🗛 🛛 🔿 🖉 irela
EC2 Dishboard X EC2 Global View C O C A	Account attributes
Events You are using the following Amazon EC2 resources in the Europe (Ireland) Region:	Default VPC 🖸
Instances Instances (running) 37 Auto Scaling Groups 9 Dedicated Hosts 0 S	pc-0c735787e36a3c094 Settings
Instance Types Elastic IPs 10 Instances 77 Key pairs 1672 p Launch Templates Laad balancers 5 Placement aroups 1 Security oroups 550 2	Data protection and security Sones
Spot Requests Volumes 89	C2 Serial Console Default credit specification
Reserved Instances Dedicated Hosts Capacity Reservations Launch Instance Service health AWS Health Dashboard C C E	Explore AWS
Timages AMIs AMIs AMIC catalog AMIC tatalog AMIC rear occurred Amic rear occurred retrieving service health information	SPU Powered ML Inference with (
V Elastic Block Store Note: Your instances will launch in the Europe (reland) Region Volumes Zones	save up to 90% on EC2 with Spot
Snapshots Uterycle Manager View in CloudWatch [2] Zone name Zone ID	ingle EC2 ASG. Learn more 🗹
▼ Network & Security ▲ 0 in alarm ④ 1 0K ⊖ 7 insufficient data eu-west-1a euw1-a2 G Security Groups €1 - 0K ⊖ 7 insufficient data €1 - 0K 7	Get Up to 40% Better Price Perfor Get Up to 40% Better Price Perfor General purpose workloads In Ama
Elastic IP's Instances in alarm eu-west-1c euw1-az1	

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.

Create key pair	×
Key pair name Key pairs allow you to connect to your instance securely. Enter key pair name The name can include up to 255 ASCII characters. It can't include leading or trailing Key pair type D PSA	j spaces.
RSA encrypted private and public key pair ED25519 encrypted private public key	rivate and public
Private key file format .pem For use with OpenSSH	
O .ppk For use with PuTTY	
▲ When prompted, store the private key in a secure and accessible your computer. You will need it later to connect to your instant more	e location on ce. <u>Learn</u>
Cancel Ca	reate key pair

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.

Network ACL: acl-07013824	42b2fd7d79				Edit network ACL association
Inbound rules (8)					< 1 > @
Rule number	⊽ Туре	▼ Protocol	▼ Port range		
1	All traffic	All	All	0.0.0/0	⊘ Allow
2	Custom TCP	TCP (6)	0	0.0.0/0	⊘ Allow
3	HTTP* (8080)	TCP (6)	8080	0.0.0/0	⊘ Allow
4	HTTP* (8080)	TCP (6)	8080	0.0.0/0	⊘ Allow
5	Custom TCP	TCP (6)	8761	0.0.0.0/0	⊘ Allow
6	Custom TCP	TCP (6)	8083	0.0.0/0	⊘ Allow
100	All traffic	All	All	0.0.0/0	⊘ Allow
*	All traffic	All	All	0.0.0/0	🛞 Deny
Outbound rules (5) Q Filter outbound rules					< 1 > ©
Rule number	⊽ Туре	▼ Protocol	▼ Port range	▼ Destination	
1	Custom TCP	TCP (6)	8761	0.0.0/0	⊘ Allow
2	Custom TCP	TCP (6)	8083	0.0.0/0	⊘ Allow
3	HTTP* (8080)	TCP (6)	8080	0.0.0/0	⊘ Allow
100	All traffic	All	All	0.0.0/0	⊘ Allow
*	All traffic	All	All	0.0.0/0	🛞 Deny

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

$\leftarrow \rightarrow \mathcal{C}$ $\bigcirc \mathfrak{B}$ ecz	2-3-253-137-21.eu-west-1.o	ompute.amazonaws.com:8761]			☆	ල ය ම එ 🗮 ≡
	🖉 spring	Eureka			номе	LAST 1000 SINCE STARTUP	
System Status							
Environment		test		Current time		2024-08-11T16:03:39 +0000	
Data center		default		Uptime		00:47	
				Lease expiration enabled		false	
				Renews threshold		5	
				Renews (last min)		4	
DS Replicas	tered with Eure	ka					
Application	AMIs	Availability Zones	Status				
GATEWAY-SERVICE	n/a (1)	(1)	UP (1) - ip-172-3	1-3-234.eu-west-1.compute.interna	al:gateway-servio	:e:8080	
ORGANIZATION-SERVICE	n/a (1)	(1)	UP (1) - organiza	tion-service:8083			
General Info							
Name			Value				
total-avail-memory			62mb				
environment			test				
num-of-cpus			1				
current-memory-usage			36mb (58%)				
septer-uptime			00:47				

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

Instances currently registered with Eureka						
Application	AMIs	Availability Zones	Status			
GATEWAY-SERVICE	n/a (1)	(1)	UP (1) - ip-172-31-3-234.eu-west-1.compute.internal:gateway-service:8080			
ORGANIZATION-SERVICE	n/a (1)	(1)	UP (1) - organization-service:8083			
General Info						
Name			Value			
total-avail-memory			62mb			
environment			test			

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.

	Test Plan.jmx (/home/hp/Test Plan.jmx) - Apache JMeter (2.13.20180731)		• 6	3
<u>F</u> ile <u>E</u> dit Search <u>R</u> un <u>O</u> ptions <u>H</u> elp			_	
] + - 4 0 0 1, 5 5 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1	0 🔔	0/0	Г
o Test Plan o Thread Group	Thread Group			
HTTP Request	Name: Thread Group			
View Results Tree	Comments:			
- III workBench				
	Continue Start Next Inread Loop Stop Inread Stop lest Stop lest Stop lest Now			
	Thread Properties			
	Number of Threads (users): 5000			
	Ramp-Up Period (in seconds): 1			
	Loop Count: Forever 3			
	🔲 Delay Thread creation until needed			
	Scheduler			

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

	Test Plan.jmx (/home/hp/Test Plan.jmx) - Apache JMeter (2.13.20180731)	8
Eile Edit Search Run Options Help		
		0 =
	HTTP Request Name: HTTP Request Comments: Web Server Server Name or IP: ec2-34-243-195-9.eu-west-1.compute.amazonaws.com Port Number: 8083 Connect: Response: HTTP Request Implementation: Protocol [http]: Method: GET Content encoding: Dath: [apl/cragatatione/]	
	Redirect Automatically 🖉 Follow Redirects 🖉 Use KeepAlive 🗋 Use multipart/form-data for POST 📄 Browser-compatible headers	
	Send Parameters With the Request:	
	Name: Value Encode? Include Equals?	
	Detail Add Add from Clabourd Delete Up Down	
	Detail Aug Rug Honder Detete Op Down	4
	Send Files With the Request:	- 1
	Pile Patr: Parameter Name: MIME Type:	
	Proxy Server	
	Server Name or IP: Port Number: Username Password]
	Embedded Resources from HTML Files Retrieve All Embedded Resources Use concurrent pool. Size: URLs must match:	
	Source address PHistoriama Use as Monitor Save response as MD5 hash?	

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.

Activitie	es 🗈 Terminal 🔻	Jul 25 11:56 AM		e	🕈 📢 (î -
		ec2-user@ip-172-31-18-17:-/organizations-service	۹ =			
•🝅	2024-07-25 10:56:01.344 DEBUG 318 2024-07-25 10:56:01.422 DEBUG 318 2024-07-25 10:56:01.422 DEBUG 318	<pre>03 [freshExecutor-0] c.n.d.utll.descrilitzerStringGache : clearing global-level cache with size 1 003 [freshExecutor-0] c.n.d.utll.descrilitzerStringGache : clearing app-level serialization cache with size 8 003 [freshExecutor-0] n.d.utl.l.descrilitzerStringGache : clearing app-level serialization cache with size 8</pre>	/eureka//ar	ops/de	elta?:	sta
1	tusCode=200 2024-07-25 10:56:01.422 DEBUG 318	1063 [freshExecutor-0] com.netflix.dlscovery.DlscoveryClient : Got delta update with apps hashcode UP_1_				
	2024-07-25 10:56:01.422 DEBUG 318 2024-07-25 10:56:01.422 DEBUG 318	<pre>003 [freshExecutor-0] com.netflix.discovery.DiscoveryClient : Added instance organization-service:8083 to the existing apps in region null 003 [freshExecutor-0] com.netflix.discovery.DiscoveryClient : The total number of instances fetched by the delta processor : 1</pre>				
•	2024-07-25 10:56:01.423 DEBUG 318 2024-07-25 10:56:01.423 DEBUG 318 is fetching remote regions? false	1003 [freshtxecutor-0] com.netf(X.discovery.Discoveryclient : Ine total number of all instances in the client now is 1 1003 [freshtxecutor-0] com.netf[X.discovery.DiscoveryClient : Completed cache refresh task for discovery. All Apps hash code is Local regional content in the client now is 1	on apps has	hcode	: UP_1	
	2024-07-25 10:56:01.819 INFO 318	063 [Timer-0] c.s.o.config.DynamicHeartbeatService : Sending heartbeat				
	2024-07-25 10:56:01.820 INFO 318	1063 [Timer-0] c.s.o.config.CustomEurekaClient : Heartbeat sent to Eureka at Thu Jul 25 10:56:01 UTC 2024 with interval 30000				
	2024-07-25 10:56:01.821 INFO 318	1063 [Timer-0] C.S.O.Confty.DynamicHeartbeatService : Scheduling next heartbeat with interval: 8494 ms				
	2024-07-25 10:56:10.343 INFO 318	1005 [Timer e] c.s.o.config.UyhamicHeartbeatService : Senoing neartbeat 1005 [Timer e] c.s.o.config.UyhamicHeartbeatService : Senoing neartbeat	-			
	2024-07-25 10:50:10.344 INFO 316	1005 ··· [There's] C.s. o. config. Dupanic Heartheast Service · · · Beat Users Sent to Eartheast with interval : \$10.5010 OF 2024 with the value 30000				
	2024-07-25 10:56:11.749 INFO 318	1663 [Timer-0] C.S.o. configure manufacture transfer to the second state of the se				
	2024-07-25 10:56:11.750 INFO 318	1063 [Timer-0] c.s.o.config.CustomEurekaClient : Heartbeat sent to Eureka at Thu Jul 25 10:56:11 UTC 2024 with interval 30000				
		1863 [Timer-0] c.s.o.config.DynamicHeartbeatService : Scheduling next heartbeat with interval: 8208 ms				
A	2024-07-25 10:56:17.863 INFO 318	063 [Timer-0] c.s.o.config.DynamicHeartbeatService : Sending heartbeat				
_	2024-07-25 10:56:17.863 INFO 318	1063 [Timer-0] c.s.o.config.CustomEurekaClient : Heartbeat sent to Eureka at Thu Jul 25 10:56:17 UTC 2024 with interval 30000				
-	2024-07-25 10:56:17.864 INFO 318	1003 [Time-0] C.S.O.Config.DynamchetartbeatService : Scheduling next heartbeat with interval: 8145 ms				
(2)	2024-07-25 10:50:18.873 INFO 318	1003 [Timer e] C.S.O.Conrig.Uyhanickeartbeatservice : Senoing neartbeat 1062 - [Timer e] C.S.O.Conrig.Uyhanickeartbeatservice : Senoing neartbeat				
	2024-07-25 10:50:18.880 INFO 310	intervel i social de la constructione de la construction de la constru	ma			
	2024-07-25 10:56:20.111 INFO 318	1063 [Titler 0] C.S.O. Config. Dynamic Heartbeat Service : Sending heartbeat				
	2024-07-25 10:56:20.111 INFO 318	1063 [Timer-0] C.S.O.Conflg.CustomEurekaClient : Heartbeat sent to Eureka at Thu Jul 25 10:56:20 UTC 2024 with interval 30000	ms			
1		1863 [Timer-0] c.s.o.config.DynamicHeartbeatService : Scheduling next heartbeat with interval: 8143 ms				
		1063 [Timer-0] c.s.o.config.DynamicHeartbeatService : Sending heartbeat				
	2024-07-25 10:56:26.423 INFO 318	1063 [Timer-0] c.s.o.config.CustomEurekaClient : Heartbeat sent to Eureka at Thu Jul 25 10:56:26 UTC 2024 with interval 30000				
	2024-07-25 10:56:26.424 INFO 318	<pre>1063 [Timer-0] c.s.o.config.DynamicHeartbeatService : Scheduling next heartbeat with interval: 8131 ms</pre>				
	2024-07-25 10:56:27.032 INFO 318	1003 [There'] C.s.o.config.DynamicHeartbeatService : Sending heartbeat				
	2024-07-25 10:50:27.080 INFO 316	1005 [There of C.S. o. config. Dussic Heartheast Service Scheduling next the state with interval 831 ms	MS			
	2024-07-25 10:56:28.473 INFO 316	1 There of class config Dynamic Heartbeat Service Sending bertheat				
	2024-07-25 10:56:28.539 INFO 318	1063 [Timer-0] C.s.o.config.CustonEurekaClient : Heartbeat sent to Eureka at Thu Jul 25 10:56:28 UTC 2024 with interval 30000	ms			
		1863 [Timer-0] c.s.o.config.DynamicHeartbeatService : Scheduling next heartbeat with interval: 8125 ms				
		1863 [t-Conn-Cleaner2] c.n.d.shared.MonitoredConnectionManager : Closing connections idle longer than 30000 SECONDS				
	2024-07-25 10:56:30.776 DEBUG 318	1063 [t-Conn-Cleaner2] c.n.d.shared.NamedConnectionPool : Closing connections idle longer than 30000 SECONDS				
	2024-07-25 10:56:31.329 DEBUG 318	1003 [tbeatExecutor-0] c.n.d.shared.MontforedConnectionManager : Get connection: {}->http://ec2-3-253-137-21.eu-west-1.compute.anazonaws.com	3761, timec	out =	5000	
	8 total allocated: 2 out of 200	total kep	active: 2,	tota	n issu	eo:
• 22	2024-07-25 10:56:31.330 DEBUG 315	063 [theatExecutor-0] c.n.d.shared_NamedConnectionPool : Getting free connection [()->http://ec2-3-253-137-21 eu.west.1 compute amazo	aws.com+87	6116	u111	
	2024-07-25 10:56:31.336 DEBUG 318	1063 [tbeatExecutor-0] c.n.d.shared.MonitoredConnectionManager : Released connection is reusable.	Control			
	2024-07-25 10:56:31.337 DEBUG 318	1063 [tbeatExecutor-0] c.n.d.shared.NamedConnectionPool : Releasing connection [{}->http://ec2-3-253-137-21.eu-west-1.compute.amazonaw	.com:8761]	[[nul]	1	
. 🥒		1063 [tbeatExecutor-0] c.n.d.shared.NamedConnectionPool : Pooling connection [{}->http://ec2-3-253-137-21.eu-west-1.compute.amazonaws.	:om:8761][r	ull]	keep	ali
× .	ve indefinitely					
	2024-07-25 10:56:31.337 DEBUG 318	<pre>i063 [tbeatExecutor-0] c.n.d.shared.NamedConnectionPool : Notifying no-one, there are no waiting threads</pre>				
	2024-07-25 10:56:31.337 DEBUG 318	1003 [tbeatExecutor-0] n.d.s.t.j.AbstractJerseyEureKaHttpClient : Jersey HTTP PUT http://ec2-3-253-137-21.eu-west-1.compute.amazonaws.com:8761	reureka//ap	ps/0	GANIZA	110
• 🥢	2824-87-25 18:56:31 337 period	ma; statuscuppezpo NG3 theatEvenutor.Al com patflix discovery DiscoveryClient DiscoveryClient OBCANTATION.SEPVICE/organization condex PART - Variability	atus: 700			
	2024-07-25 10:56:31.424 DEBUG 318	(10) - [freehEverutor-0] conditionation of control state of control and the control of control of control and control of con	3761 timec	ut -	5888	
	2024-07-25 10:56:31.425 DEBUG 318	(f)-shtps://cc.a.scale.com/cc.a.shared.NamedConnectionPool (f)-shtps://cc.a.scale.a.scale.com/cc.a.shared.NamedConnectionPool (f)-shtps://cc.a.scale.a.scale.com/cc.a.shared.NamedConnectionPool (f)-shtps://cc.a.scale.a.scale.com/cc.a.sc	alive: 2	tota	l issu	ied:
	0, total allocated: 2 out of 200					
	2024-07-25 10:56:31.425 DEBUG 318	<pre>i863 [freshExecutor-0] c.n.d.shared.NamedConnectionPool : Getting free connection [{}->http://ec2-3-253-137-21.eu-west-1.compute.amazo</pre>	naws.com:87	61][1	ull]	

Figure 8: Dynamic heartbeat interval logs