# Lab 4(Agents: The Rise of Agents)

Fatema Mirza & Mohammad Newaj Jamil

D7001D Network Programming and Distributed Applications

# Lab 4(Agents: The Rise of Agents)

by

Fatema Mirza & Mohammad Newaj Jamil

| | | |
|---|---|---|
| Submission Deadline: | November 5, 2021 | |
| Teachers: | Dr. Evgeny Osipov, | LULEÅ TEKNISKA UNIVERSITET |
| | Ahmed Afif Monrat, | LULEÅ TEKNISKA UNIVERSITET |

# Contents

# 1
# Part I – The Agents Attack

## 1.1. Introduction

A multi agent system creation will be demonstrated in this lab. The architecture will be as follows:

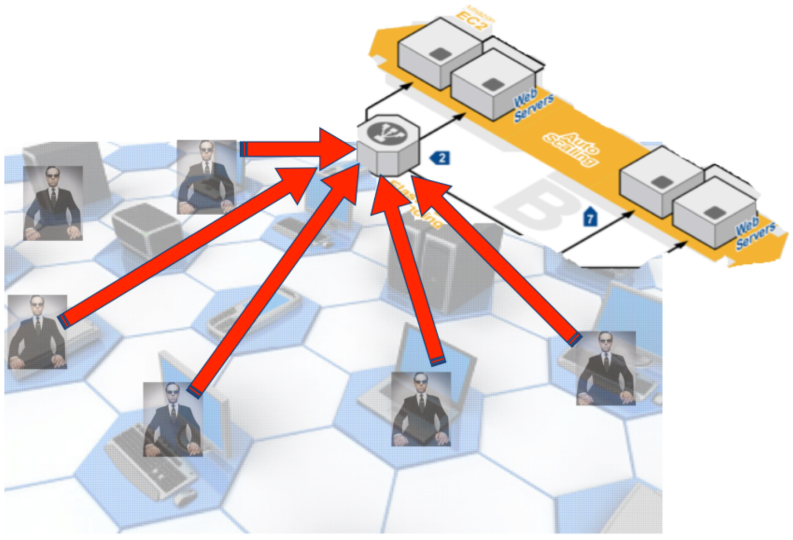
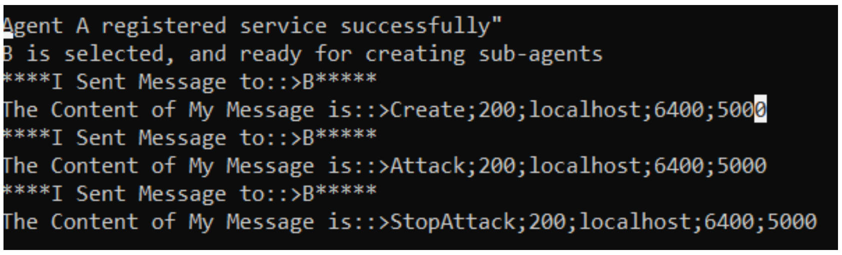


Figure 1.1: Architect Messages

To increase the load on the server, it was opted to increase the number of agents created and observe whether the server can handle the load.

### 1.1.1. Architect

An Architect is an agent (Architect.java and ArchitectFrame.java) which is designed to provide the GUI interface for to create numerous instances of Agent Smith automatically. The Architect implements OneShotBehavior for creating agents, attacking or terminating the attack. The Architect implements a CyclicBehavior to listen to the messages addressed to the Architect.
Architect is capable of sending 3 types of messages to the Broker:



```
Agent A registered service successfully"
B is selected, and ready for creating sub-agents
****I Sent Message to::>B*****
The Content of My Message is::>Create;200;localhost;6400;5000
****I Sent Message to::>B*****
The Content of My Message is::>Attack;200;localhost;6400;5000
****I Sent Message to::>B*****
The Content of My Message is::>StopAttack;200;localhost;6400;5000
```

Figure 1.2: Architect Messages

1. Create - which creates the number of agents as specified in the Number of agents field
2. Attack - Attack the server with the fibonacci calculation
3. Stop attack - Stop the attack

### 1.1.2. Agent Smith

Agent Smith (AgentSmith.java) is an agent that defines the template for all the agents to be created by the Architect. The Agent Smith will make a periodic tcp connection to the specified server and send the request to calculate the Fibonacci range of 40 by implementing TickerBehavior within a CyclicBehavior to listen to the messages addressed to it.

### 1.1.3. Broker

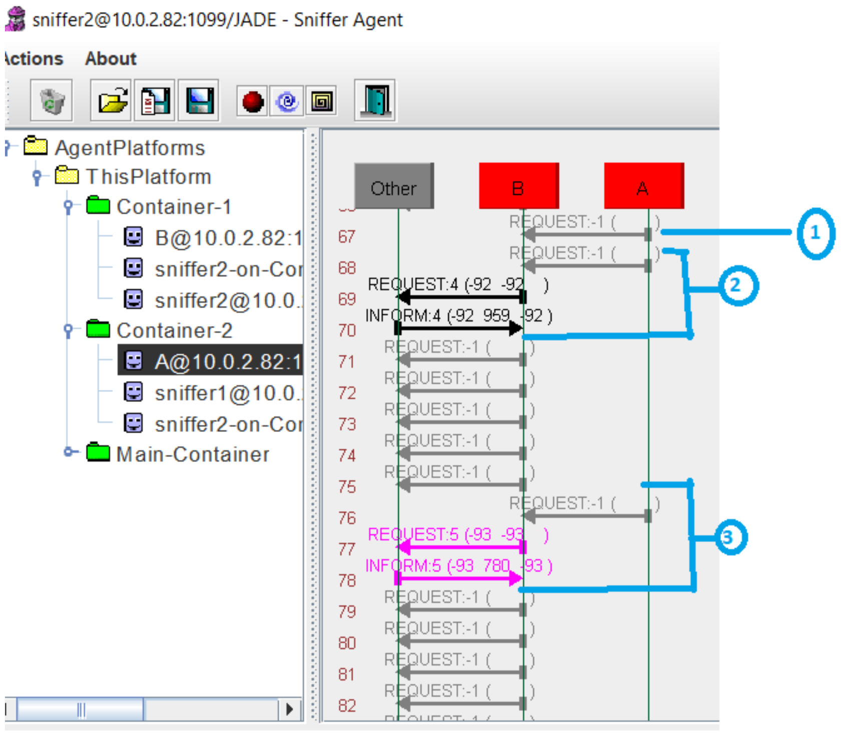
The Broker (Broker.java) will join the input it receives from the GUI and create the Agent Smiths from the template of Agent Smiths as specified in the GUI using the createNewAgent function of the ContainerController by implementing a CyclicBehavior.
Broker is capable of receiving 3 types of messages from the Architect:

1. Create - which creates the number of agents as specified in the Number of agents field from the Architect
2. Attack - Attack the server with the fibonacci calculation
3. Stop attack - Stop the attack

### 1.1.4. Server

A Multithreaded TCP server (MultithrededServer.java) is implemented to serve calculate the Fibonacci sequence (in a recursive manner) and is defined on the port 6400. The Server is hosted on an EC2 instance of AWS so that it can be dynamically autoscaled and load balancing can be performed.

## 1.2. Communication Between Agents



```
67                      REQUEST:-1 (       )
68                      REQUEST:-1 (       )
69  REQUEST:4 (-92  -92    )
70  INFORM:4 (-92  959  -92 )
71      REQUEST:-1 (       )
72      REQUEST:-1 (       )
73      REQUEST:-1 (       )
74      REQUEST:-1 (       )
75      REQUEST:-1 (       )
76                      REQUEST:-1 (       )
77  REQUEST:5 (-93  -93    )
78  INFORM:5 (-93  780  -93 )
79      REQUEST:-1 (       )
80      REQUEST:-1 (       )
81      REQUEST:-1 (       )
82      REQUEST:-1 (       )
```

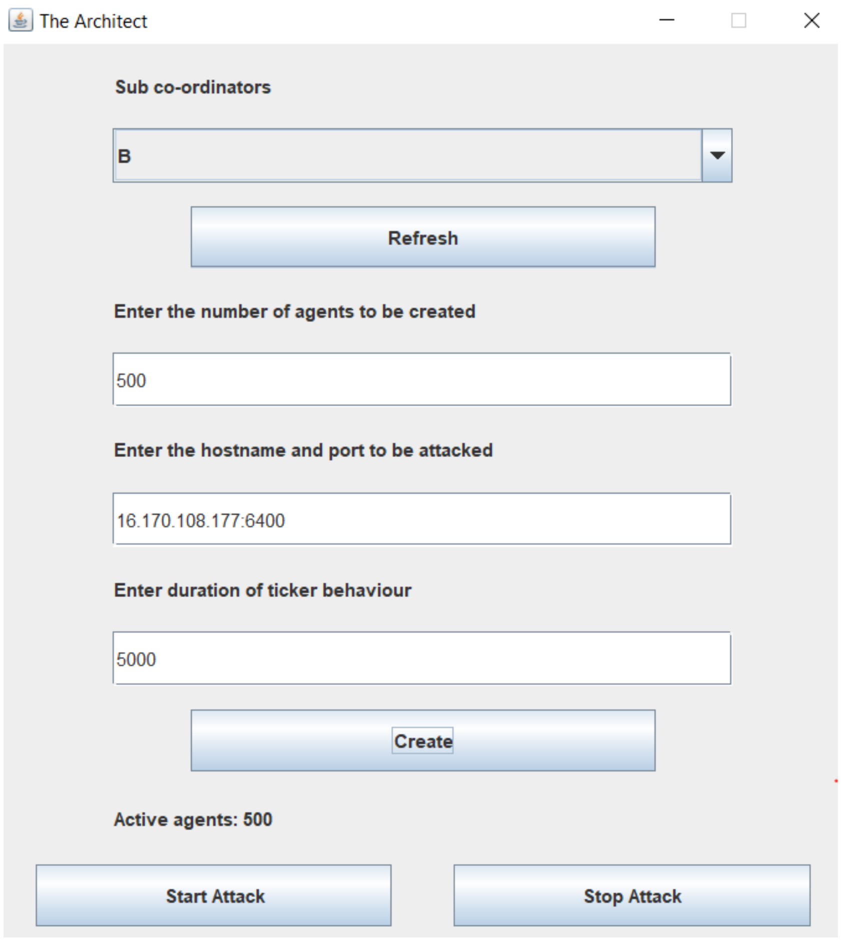
Figure 1.3: Communication demonstrated via the Jade Sniffer

The image above demonstrates how the communication between the agents take place. A represents the Architect. B represents the Broker. Others represent all the Agent Smiths that are created dynamically.

1. In 1, the Architect sends the number of agents to be created to Broker at IP address 16.170.108.177 (Address of the server on AWS) at port 6400.
2. In 2, the Architect sends the Attack request to the Broker, which passes it to the Agent Smiths to attack the server with the Fibonacci calculation.
3. In 3, the Architect sends the Stop Attack request to the Broker, which passes it to the Agent Smiths to stop attacking the server and terminate itself.

## 1.3. Demo of the Multiagent System

The following is input into the GUI of the system to create the agent Smiths.



B

Enter the number of agents to be created

500

Enter the hostname and port to be attacked

16.170.108.177:6400

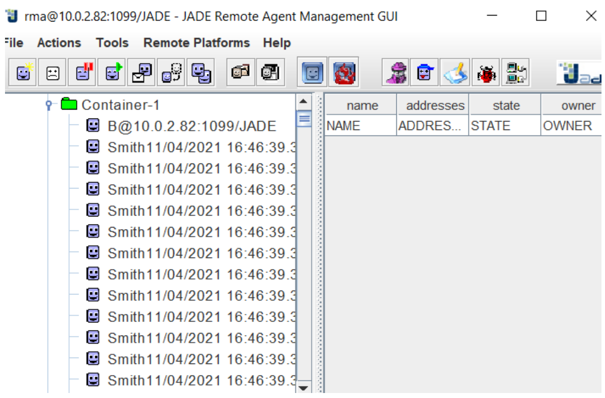
Enter duration of ticker behaviour

5000

Active agents: 500

Figure 1.4: GUI

Once created:

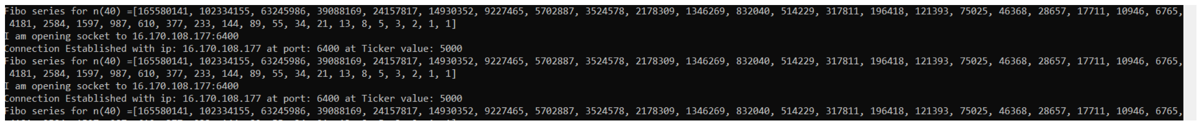


```
Container-1
  B@10.0.2.82:1099/JADE
  Smith11/04/2021 16:46:39.3
  Smith11/04/2021 16:46:39.3
  Smith11/04/2021 16:46:39.3
  Smith11/04/2021 16:46:39.3
  Smith11/04/2021 16:46:39.3
  Smith11/04/2021 16:46:39.3
  Smith11/04/2021 16:46:39.3
  Smith11/04/2021 16:46:39.3
  Smith11/04/2021 16:46:39.3
  Smith11/04/2021 16:46:39.3
  Smith11/04/2021 16:46:39.3
  Smith11/04/2021 16:46:39.3
  Smith11/04/2021 16:46:39.3
```

| name | addresses | state | owner |
|---|---|---|---|
| NAME | ADDRES... | STATE | OWNER |

Figure 1.5: Agent Smiths created

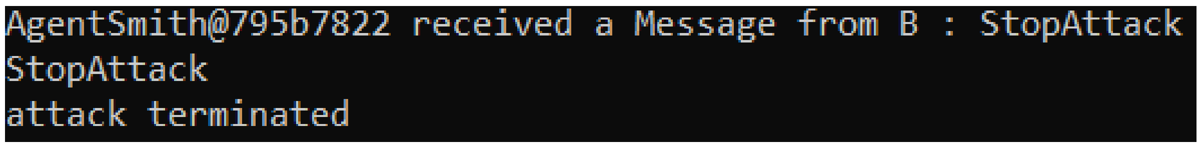
To start the attack, click on the start attack button.



```
Fibo series for n(40) =[165580141, 102334155, 63245986, 39088169, 24157817, 14930352, 9227465, 5702887, 3524578, 2178309, 1346269, 832040, 514229, 317811, 196418, 121393, 75025, 46368, 28657, 17711, 10946, 6765,
 4181, 2584, 1597, 987, 610, 377, 233, 144, 89, 55, 34, 21, 13, 8, 5, 3, 2, 1, 1]
I am opening socket to 16.170.108.177:6400
Connection Established with ip: 16.170.108.177 at port: 6400 at Ticker value: 5000
Fibo series for n(40) =[165580141, 102334155, 63245986, 39088169, 24157817, 14930352, 9227465, 5702887, 3524578, 2178309, 1346269, 832040, 514229, 317811, 196418, 121393, 75025, 46368, 28657, 17711, 10946, 6765,
 4181, 2584, 1597, 987, 610, 377, 233, 144, 89, 55, 34, 21, 13, 8, 5, 3, 2, 1, 1]
I am opening socket to 16.170.108.177:6400
Connection Established with ip: 16.170.108.177 at port: 6400 at Ticker value: 5000
Fibo series for n(40) =[165580141, 102334155, 63245986, 39088169, 24157817, 14930352, 9227465, 5702887, 3524578, 2178309, 1346269, 832040, 514229, 317811, 196418, 121393, 75025, 46368, 28657, 17711, 10946, 6765,
```

Figure 1.6: Result of the Attack

To stop the attack, click on the stop attack button.



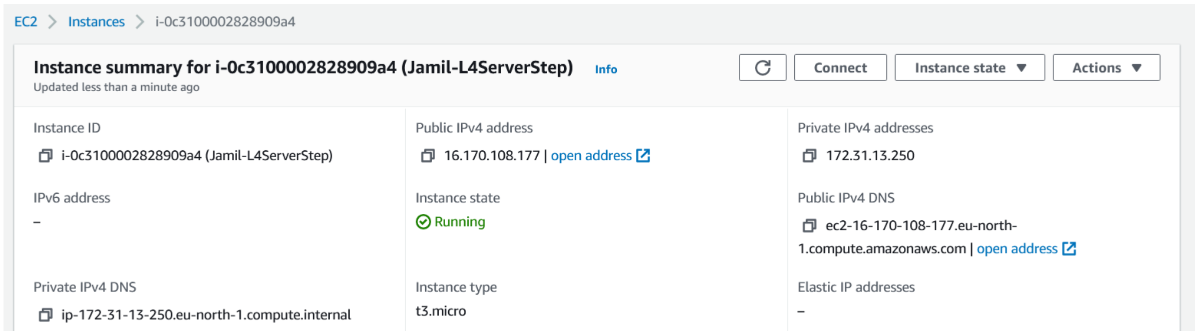
```
AgentSmith@795b7822 received a Message from B : StopAttack
StopAttack
attack terminated
```

Figure 1.7: Result of the Stop Attack

# 2

# Part II – No pasaran!

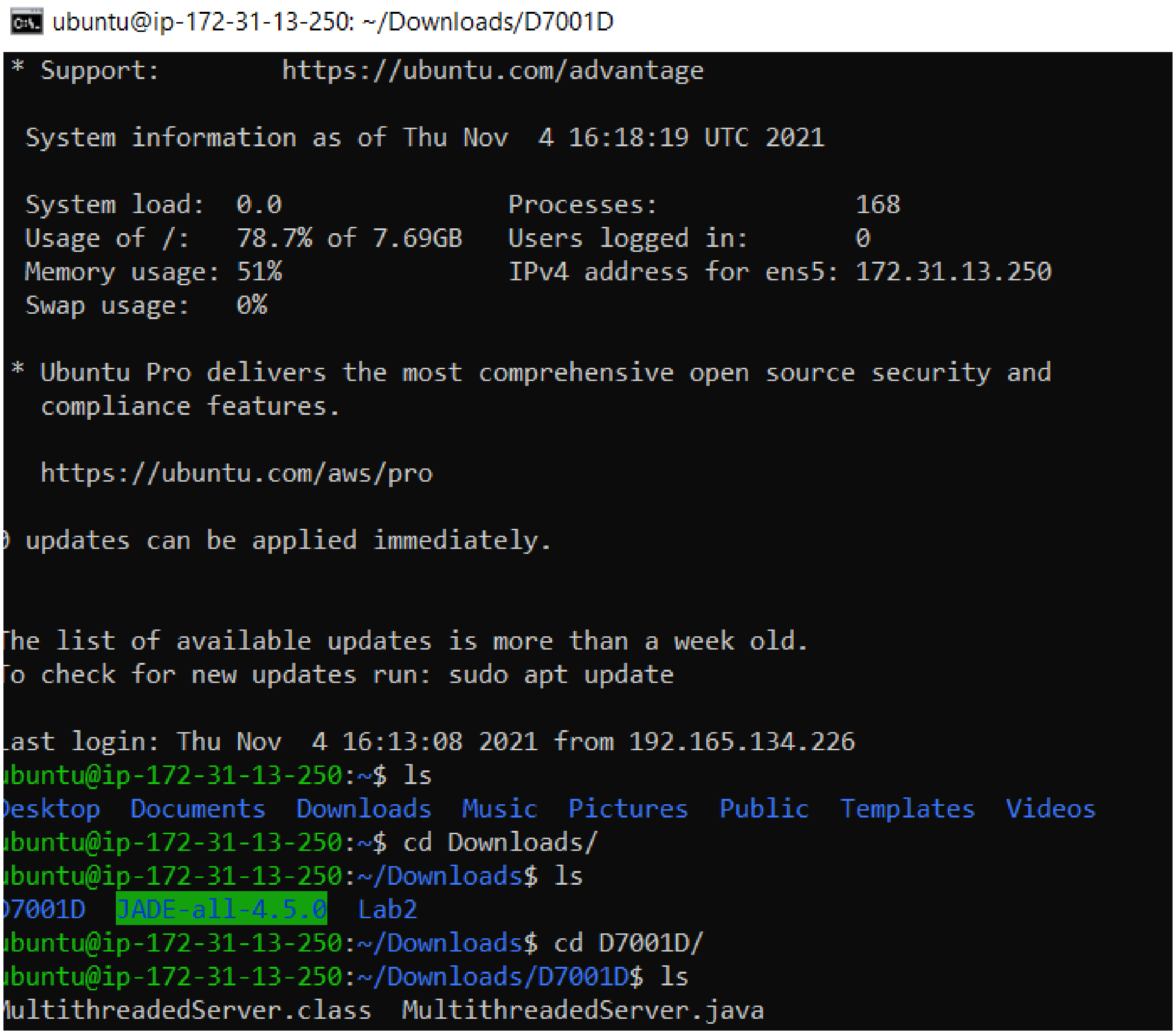
To host the server on AWS, an instance named Jamil-L4ServerStep is created where the required java packages are installed. After that, using the scp command with the key and the file locations, the Multithreaded-Server.java is uploaded to the AWS instance where it is compiled using javac. It is now ready to run.



Instance summary for i-0c3100002828909a4 (Jamil-L4ServerStep)
Updated less than a minute ago

| | | |
|---|---|---|
| Instance ID | Public IPv4 address | Private IPv4 addresses |
| i-0c3100002828909a4 (Jamil-L4ServerStep) | 16.170.108.177 \| open address | 172.31.13.250 |
| IPv6 address | Instance state | Public IPv4 DNS |
| – | Running | ec2-16-170-108-177.eu-north-1.compute.amazonaws.com \| open address |
| Private IPv4 DNS | Instance type | Elastic IP addresses |
| ip-172-31-13-250.eu-north-1.compute.internal | t3.micro | – |

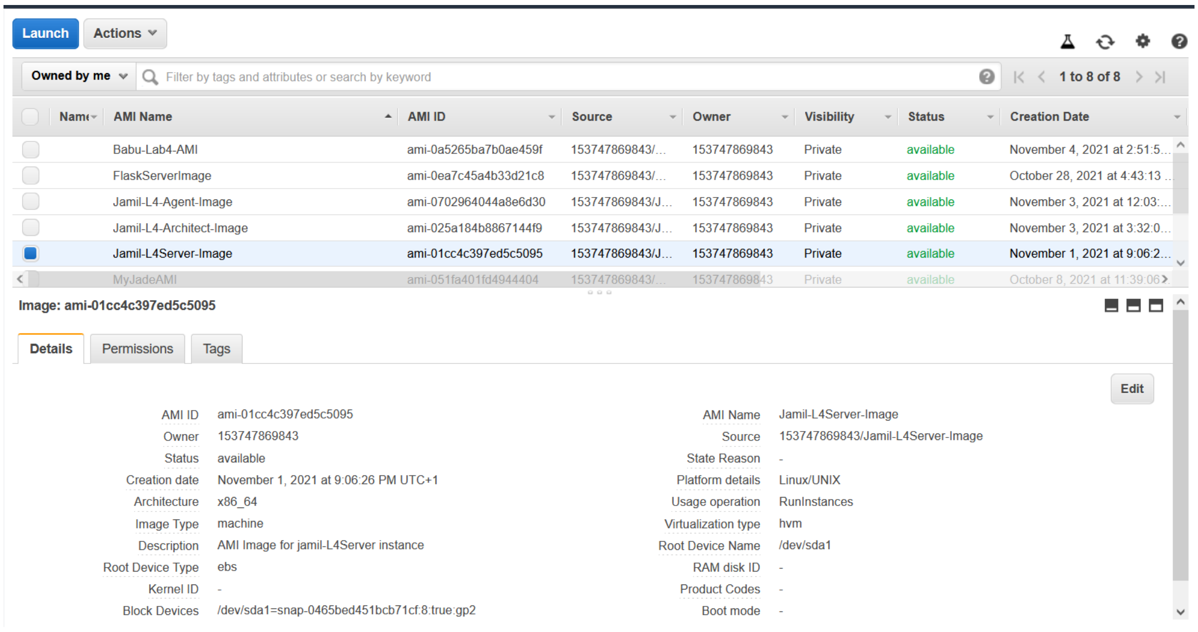
Figure 2.1: Server hosted on AWS



```
ubuntu@ip-172-31-13-250: ~/Downloads/D7001D
 * Support:        https://ubuntu.com/advantage

  System information as of Thu Nov  4 16:18:19 UTC 2021

  System load:  0.0               Processes:             168
  Usage of /:   78.7% of 7.69GB   Users logged in:       0
  Memory usage: 51%               IPv4 address for ens5: 172.31.13.250
  Swap usage:   0%

 * Ubuntu Pro delivers the most comprehensive open source security and
   compliance features.

   https://ubuntu.com/aws/pro

) updates can be applied immediately.


The list of available updates is more than a week old.
To check for new updates run: sudo apt update

.ast login: Thu Nov  4 16:13:08 2021 from 192.165.134.226
ıbuntu@ip-172-31-13-250:~$ ls
Desktop  Documents  Downloads  Music  Pictures  Public  Templates  Videos
ıbuntu@ip-172-31-13-250:~$ cd Downloads/
ıbuntu@ip-172-31-13-250:~/Downloads$ ls
)7001D  JADE-all-4.5.0  Lab2
ıbuntu@ip-172-31-13-250:~/Downloads$ cd D7001D/
ıbuntu@ip-172-31-13-250:~/Downloads/D7001D$ ls
MultithreadedServer.class  MultithreadedServer.java
```

Figure 2.2: Inside Server hosted on AWS

Once the Server is up and running an AMI named Jamil-L4Server-Image has been created (which will be used in a dual fold purpose - that is to create the new instances for autoscaling as well as a safety measure in case the server configurations are lost).



Owned by me | Filter by tags and attributes or search by keyword | 1 to 8 of 8

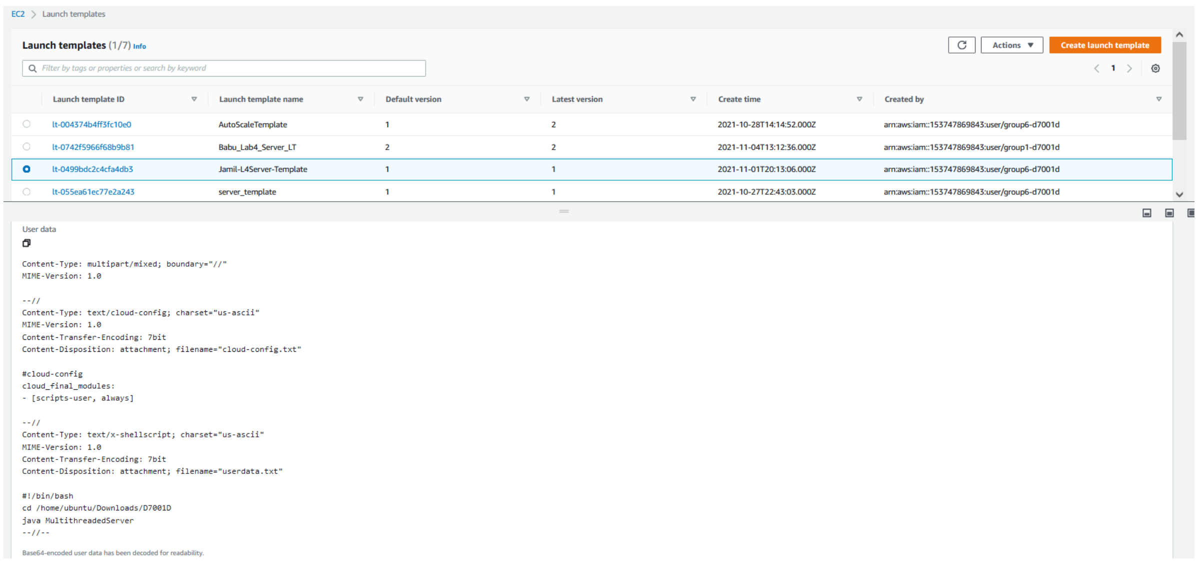
| Name | AMI Name | AMI ID | Source | Owner | Visibility | Status | Creation Date |
|---|---|---|---|---|---|---|---|
| | Babu-Lab4-AMI | ami-0a5265ba7b0ae459f | 153747869843/... | 153747869843 | Private | available | November 4, 2021 at 2:51:5... |
| | FlaskServerImage | ami-0ea7c45a4b33d21c8 | 153747869843/... | 153747869843 | Private | available | October 28, 2021 at 4:43:13 ... |
| | Jamil-L4-Agent-Image | ami-0702964044a8e6d30 | 153747869843/J... | 153747869843 | Private | available | November 3, 2021 at 12:03:... |
| | Jamil-L4-Architect-Image | ami-025a184b8867144f9 | 153747869843/J... | 153747869843 | Private | available | November 3, 2021 at 3:32:0... |
| | Jamil-L4Server-Image | ami-01cc4c397ed5c5095 | 153747869843/J... | 153747869843 | Private | available | November 1, 2021 at 9:06:2... |
| | MyJadeAMI | ami-051fa401fd4944404 | 153747869843/... | 153747869843 | Private | available | October 8, 2021 at 11:39:06 |

Image: ami-01cc4c397ed5c5095

| | | | |
|---|---|---|---|
| AMI ID | ami-01cc4c397ed5c5095 | AMI Name | Jamil-L4Server-Image |
| Owner | 153747869843 | Source | 153747869843/Jamil-L4Server-Image |
| Status | available | State Reason | - |
| Creation date | November 1, 2021 at 9:06:26 PM UTC+1 | Platform details | Linux/UNIX |
| Architecture | x86_64 | Usage operation | RunInstances |
| Image Type | machine | Virtualization type | hvm |
| Description | AMI Image for jamil-L4Server instance | Root Device Name | /dev/sda1 |
| Root Device Type | ebs | RAM disk ID | - |
| Kernel ID | - | Product Codes | - |
| Block Devices | /dev/sda1=snap-0465bed451bcb71cf:8:true:gp2 | Boot mode | - |

Figure 2.3: Server AMI

Once the AMI is created, a template named Jamil-L4Server-Template is created. To ensure that the autoscaled instances are capable of running itself without any commands, the commands are written in the user data.



EC2 > Launch templates
Launch templates (1/7) Info
Actions
Create launch template
Filter by tags or properties or search by keyword

| Launch template ID | Launch template name | Default version | Latest version | Create time | Created by |
|---|---|---|---|---|---|
| lt-004374b4ff3fc10e0 | AutoScaleTemplate | 1 | 2 | 2021-10-28T14:14:52.000Z | arn:aws:iam::153747869843:user/group6-d7001d |
| lt-0742f5966f68b9b81 | Babu_Lab4_Server_LT | 2 | 2 | 2021-11-04T13:12:36.000Z | arn:aws:iam::153747869843:user/group1-d7001d |
| lt-0499bdc2c4cfa4db3 | Jamil-L4Server-Template | 1 | 1 | 2021-11-01T20:13:06.000Z | arn:aws:iam::153747869843:user/group6-d7001d |
| lt-055ea61ec77e2a243 | server_template | 1 | 1 | 2021-10-27T22:43:03.000Z | arn:aws:iam::153747869843:user/group6-d7001d |

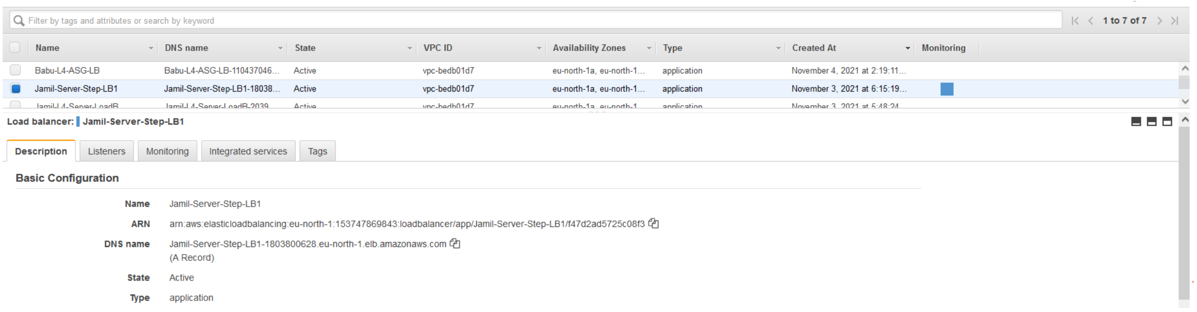
User data

```
Content-Type: multipart/mixed; boundary="//"
MIME-Version: 1.0

--//
Content-Type: text/cloud-config; charset="us-ascii"
MIME-Version: 1.0
Content-Transfer-Encoding: 7bit
Content-Disposition: attachment; filename="cloud-config.txt"

#cloud-config
cloud_final_modules:
- [scripts-user, always]

--//
Content-Type: text/x-shellscript; charset="us-ascii"
MIME-Version: 1.0
Content-Transfer-Encoding: 7bit
Content-Disposition: attachment; filename="userdata.txt"

#!/bin/bash
cd /home/ubuntu/Downloads/D7001D
java MultithreadedServer
--//--
```

Base64-encoded user data has been decoded for readability.

Figure 2.4: Server Template

Following that load balancers and target groups for autoscaling are created.



Filter by tags and attributes or search by keyword — 1 to 7 of 7

| | Name | DNS name | State | VPC ID | Availability Zones | Type | Created At | Monitoring |
|---|---|---|---|---|---|---|---|---|
| | Babu-L4-ASG-LB | Babu-L4-ASG-LB-110437046... | Active | vpc-bedb01d7 | eu-north-1a, eu-north-1... | application | November 4, 2021 at 2:19:11... | |
| ✓ | Jamil-Server-Step-LB1 | Jamil-Server-Step-LB1-18038... | Active | vpc-bedb01d7 | eu-north-1a, eu-north-1... | application | November 3, 2021 at 6:15:19... | |
| | Jamil-L4-ServerLoadB | Jamil-L4-ServerLoadB-2039 | Active | vpc-bedb01d7 | eu-north-1a, eu-north-1 | application | November 3, 2021 at 5:48:24 | |

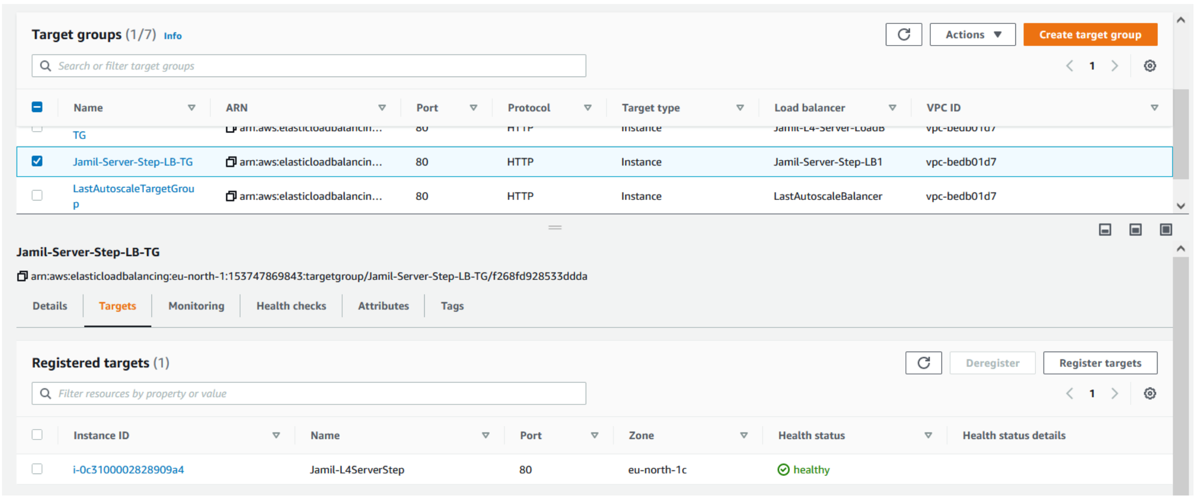
Load balancer: Jamil-Server-Step-LB1

Description | Listeners | Monitoring | Integrated services | Tags

Basic Configuration

| | |
|---|---|
| Name | Jamil-Server-Step-LB1 |
| ARN | arn:aws:elasticloadbalancing:eu-north-1:153747869843:loadbalancer/app/Jamil-Server-Step-LB1/f47d2ad5725c08f3 |
| DNS name | Jamil-Server-Step-LB1-1803800628.eu-north-1.elb.amazonaws.com (A Record) |
| State | Active |
| Type | application |

Figure 2.5: Server Load Balancer



Target groups (1/7) Info

Actions | Create target group

Search or filter target groups

| | Name | ARN | Port | Protocol | Target type | Load balancer | VPC ID |
|---|---|---|---|---|---|---|---|
| | TG | arn:aws:elasticloadbalancin... | 80 | HTTP | Instance | Jamil-L4-Server-LoadB | vpc-bedb01d7 |
| ✓ | Jamil-Server-Step-LB-TG | arn:aws:elasticloadbalancin... | 80 | HTTP | Instance | Jamil-Server-Step-LB1 | vpc-bedb01d7 |
| | LastAutoscaleTargetGroup | arn:aws:elasticloadbalancin... | 80 | HTTP | Instance | LastAutoscaleBalancer | vpc-bedb01d7 |

Jamil-Server-Step-LB-TG

arn:aws:elasticloadbalancing:eu-north-1:153747869843:targetgroup/Jamil-Server-Step-LB-TG/f268fd928533ddda

Details | Targets | Monitoring | Health checks | Attributes | Tags

Registered targets (1)

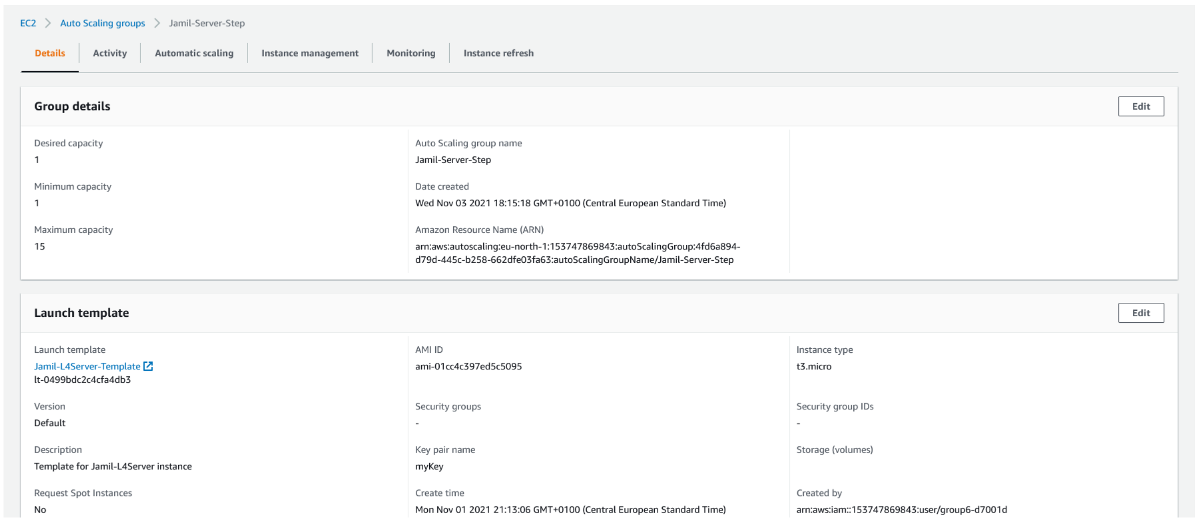
Deregister | Register targets

Filter resources by property or value

| | Instance ID | Name | Port | Zone | Health status | Health status details |
|---|---|---|---|---|---|---|
| | i-0c3100002828909a4 | Jamil-L4ServerStep | 80 | eu-north-1c | ✓ healthy | |

Figure 2.6: Server Target Group

After careful consideration, step scaling was decided to be used as the scaling policy for the autoscale. With step scaling and simple scaling, it is possible to choose scaling metrics and threshold values for the CloudWatch alarms that invoke the scaling process. This also allows for the definition of how the Auto Scaling group should be scaled when a threshold is in breach for a specified number of evaluation periods (AWS, 2021).



**Group details**

Desired capacity
1

Minimum capacity
1

Maximum capacity
15

Auto Scaling group name
Jamil-Server-Step

Date created
Wed Nov 03 2021 18:15:18 GMT+0100 (Central European Standard Time)

Amazon Resource Name (ARN)
arn:aws:autoscaling:eu-north-1:153747869843:autoScalingGroup:4fd6a894-d79d-445c-b258-662dfe03fa63:autoScalingGroupName/Jamil-Server-Step

**Launch template**

Launch template
Jamil-L4Server-Template
lt-0499bdc2c4cfa4db3

Version
Default

Description
Template for Jamil-L4Server instance

Request Spot Instances
No

AMI ID
ami-01cc4c397ed5c5095

Security groups
-

Key pair name
myKey

Create time
Mon Nov 01 2021 21:13:06 GMT+0100 (Central European Standard Time)

Instance type
t3.micro

Security group IDs
-

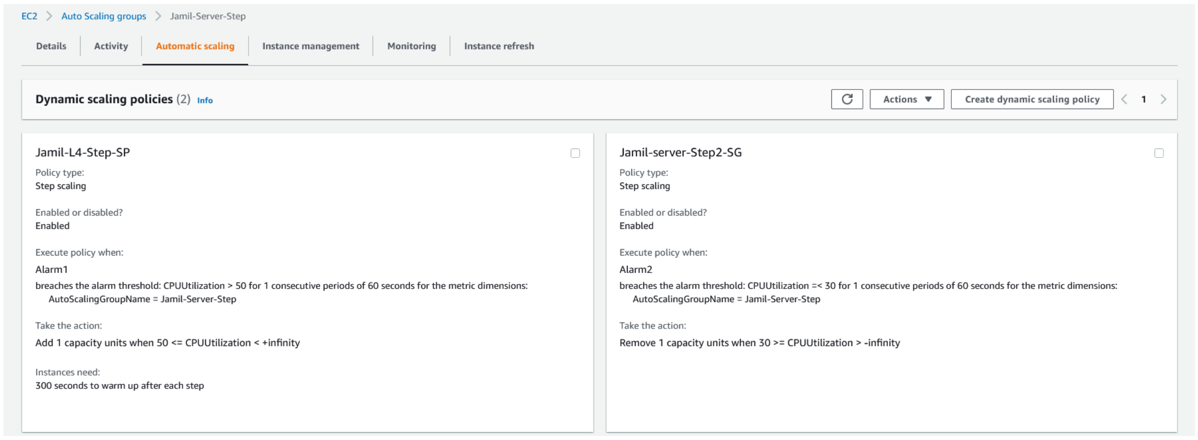
Storage (volumes)

Created by
arn:aws:iam::153747869843:user/group6-d7001d

Figure 2.7: Server AutoScale Group

Dynamic scaling policy was introduced.

1. If the CPU utilization is more than 50% for a period of 1 minute, create a new instance.
2. If the CPU utilization is less than 30% for a period of 1 minute, terminate the latest created instance.



**Dynamic scaling policies** (2)

Jamil-L4-Step-SP
Policy type:
Step scaling

Enabled or disabled?
Enabled

Execute policy when:
Alarm1
breaches the alarm threshold: CPUUtilization > 50 for 1 consecutive periods of 60 seconds for the metric dimensions:
AutoScalingGroupName = Jamil-Server-Step

Take the action:
Add 1 capacity units when 50 <= CPUUtilization < +infinity

Instances need:
300 seconds to warm up after each step

Jamil-server-Step2-SG
Policy type:
Step scaling

Enabled or disabled?
Enabled

Execute policy when:
Alarm2
breaches the alarm threshold: CPUUtilization =< 30 for 1 consecutive periods of 60 seconds for the metric dimensions:
AutoScalingGroupName = Jamil-Server-Step

Take the action:
Remove 1 capacity units when 30 >= CPUUtilization > -infinity

Figure 2.8: Server AutoScale Group Dynamic Policies

The Architect is parameterized with 9350 Agent Smiths and the attack is started. It can be observed that the CPU Utilization is now steadily increasing.

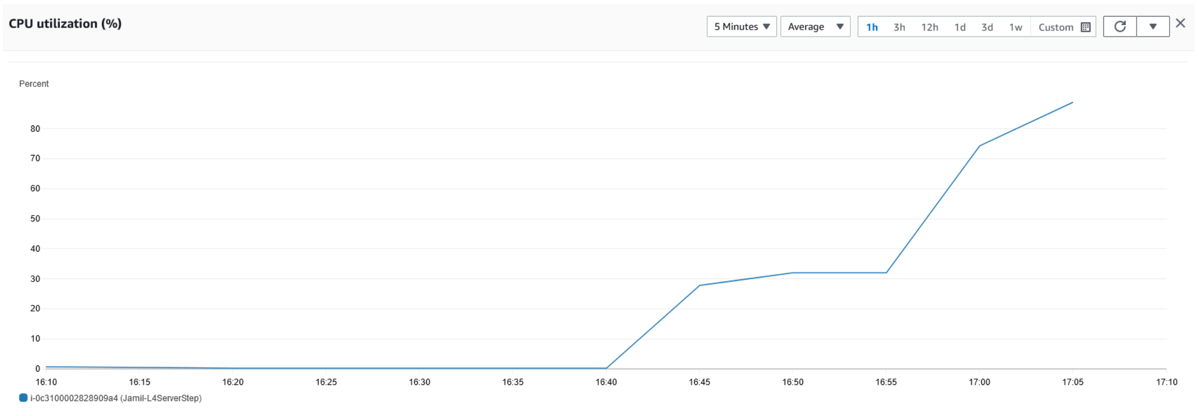
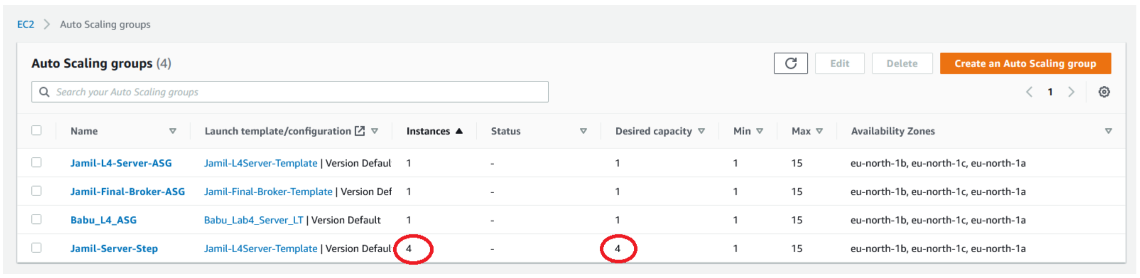


Figure 2.9: CPU Utilization increasing

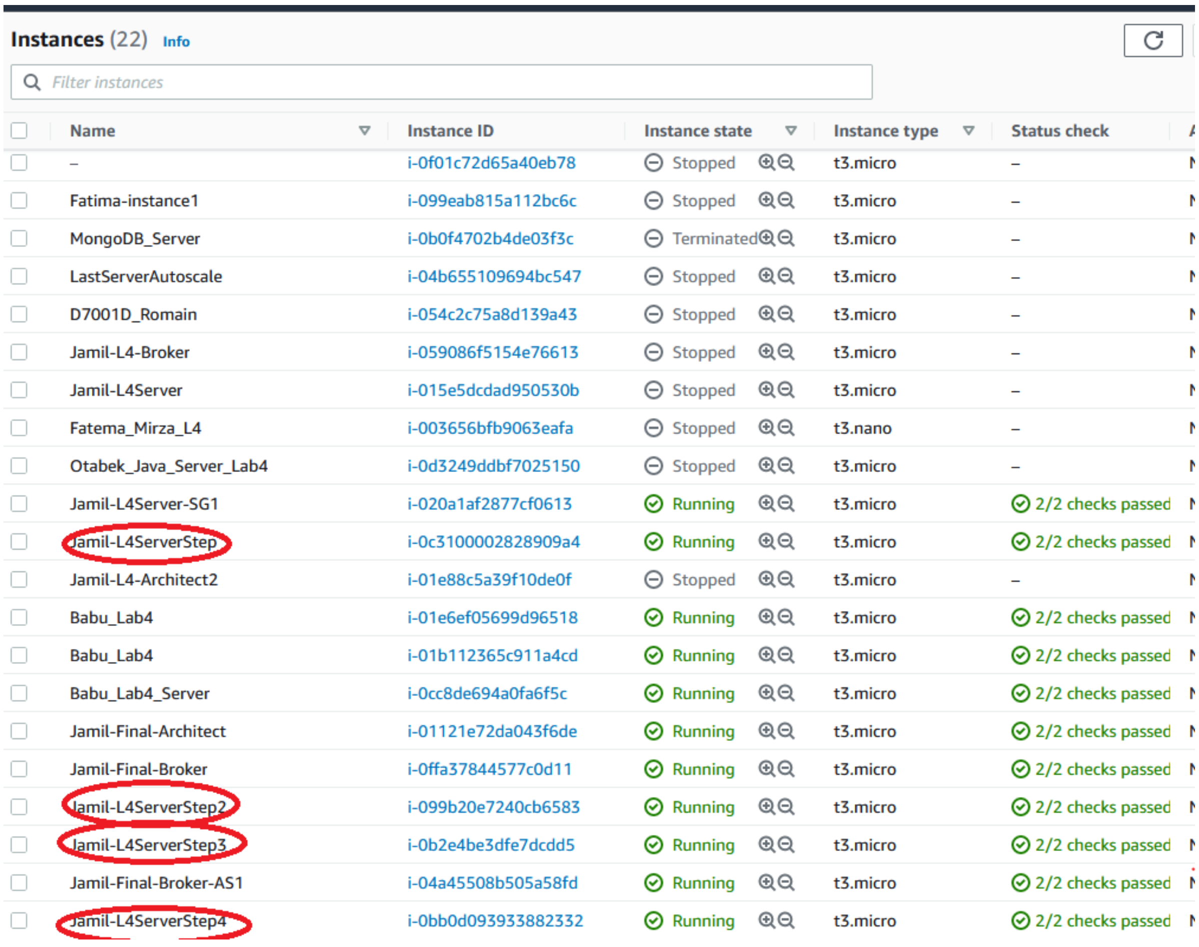
Due to this increase in CPU utlization, new instances are being created by the autoscaling group using the launch template defined.


EC2 > Auto Scaling groups

Auto Scaling groups (4)

| Name | Launch template/configuration | Instances | Status | Desired capacity | Min | Max | Availability Zones |
|---|---|---|---|---|---|---|---|
| Jamil-L4-Server-ASG | Jamil-L4Server-Template \| Version Defaul | 1 | - | 1 | 1 | 15 | eu-north-1b, eu-north-1c, eu-north-1a |
| Jamil-Final-Broker-ASG | Jamil-Final-Broker-Template \| Version Def | 1 | - | 1 | 1 | 15 | eu-north-1b, eu-north-1c, eu-north-1a |
| Babu_L4_ASG | Babu_Lab4_Server_LT \| Version Default | 1 | - | 1 | 1 | 15 | eu-north-1b, eu-north-1c, eu-north-1a |
| Jamil-Server-Step | Jamil-L4Server-Template \| Version Defaul | 4 | - | 4 | 1 | 15 | eu-north-1b, eu-north-1c, eu-north-1a |

Figure 2.10: Instances from autoscale group



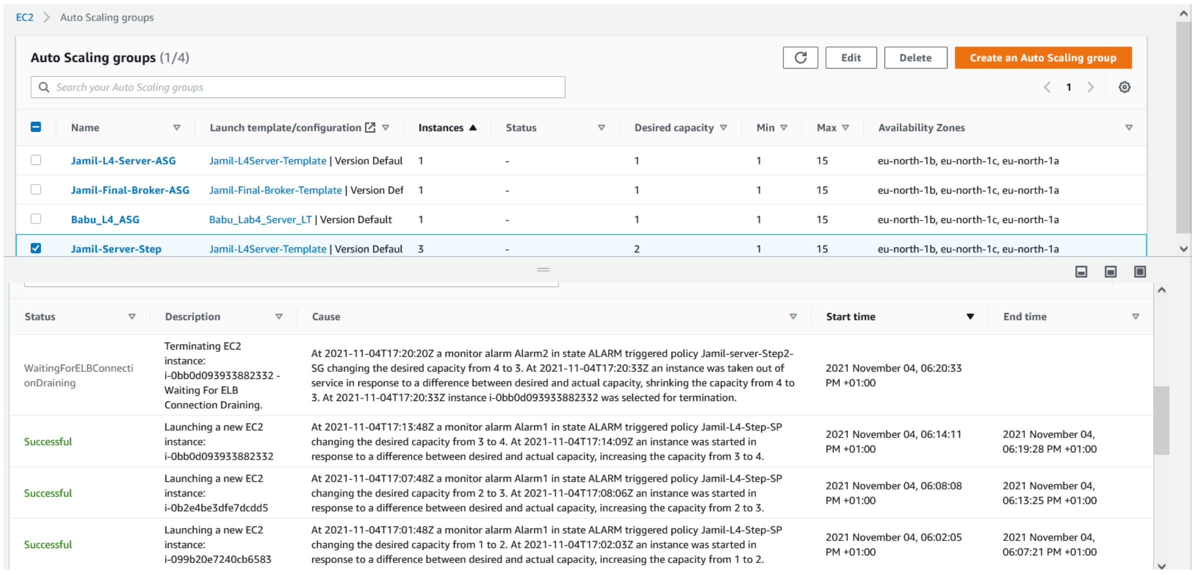
**Instances** (22) Info

Filter instances

| Name | Instance ID | Instance state | Instance type | Status check |
|---|---|---|---|---|
| – | i-0f01c72d65a40eb78 | Stopped | t3.micro | – |
| Fatima-instance1 | i-099eab815a112bc6c | Stopped | t3.micro | – |
| MongoDB_Server | i-0b0f4702b4de03f3c | Terminated | t3.micro | – |
| LastServerAutoscale | i-04b655109694bc547 | Stopped | t3.micro | – |
| D7001D_Romain | i-054c2c75a8d139a43 | Stopped | t3.micro | – |
| Jamil-L4-Broker | i-059086f5154e76613 | Stopped | t3.micro | – |
| Jamil-L4Server | i-015e5dcdad950530b | Stopped | t3.micro | – |
| Fatema_Mirza_L4 | i-003656bfb9063eafa | Stopped | t3.nano | – |
| Otabek_Java_Server_Lab4 | i-0d3249ddbf7025150 | Stopped | t3.micro | – |
| Jamil-L4Server-SG1 | i-020a1af2877cf0613 | Running | t3.micro | 2/2 checks passed |
| Jamil-L4ServerStep | i-0c3100002828909a4 | Running | t3.micro | 2/2 checks passed |
| Jamil-L4-Architect2 | i-01e88c5a39f10de0f | Stopped | t3.micro | – |
| Babu_Lab4 | i-01e6ef05699d96518 | Running | t3.micro | 2/2 checks passed |
| Babu_Lab4 | i-01b112365c911a4cd | Running | t3.micro | 2/2 checks passed |
| Babu_Lab4_Server | i-0cc8de694a0fa6f5c | Running | t3.micro | 2/2 checks passed |
| Jamil-Final-Architect | i-01121e72da043f6de | Running | t3.micro | 2/2 checks passed |
| Jamil-Final-Broker | i-0ffa37844577c0d11 | Running | t3.micro | 2/2 checks passed |
| Jamil-L4ServerStep2 | i-099b20e7240cb6583 | Running | t3.micro | 2/2 checks passed |
| Jamil-L4ServerStep3 | i-0b2e4be3dfe7dcdd5 | Running | t3.micro | 2/2 checks passed |
| Jamil-Final-Broker-AS1 | i-04a45508b505a58fd | Running | t3.micro | 2/2 checks passed |
| Jamil-L4ServerStep4 | i-0bb0d093933882332 | Running | t3.micro | 2/2 checks passed |

Figure 2.11: Instances from instance list

From the activity log, it can be observed that the alarm triggers have been set off.



EC2 > Auto Scaling groups

Auto Scaling groups (1/4)

| | Name | Launch template/configuration | Instances | Status | Desired capacity | Min | Max | Availability Zones |
|---|---|---|---|---|---|---|---|---|
| ☐ | Jamil-L4-Server-ASG | Jamil-L4Server-Template \| Version Defaul | 1 | - | 1 | 1 | 15 | eu-north-1b, eu-north-1c, eu-north-1a |
| ☐ | Jamil-Final-Broker-ASG | Jamil-Final-Broker-Template \| Version Def | 1 | - | 1 | 1 | 15 | eu-north-1b, eu-north-1c, eu-north-1a |
| ☐ | Babu_L4_ASG | Babu_Lab4_Server_LT \| Version Default | 1 | - | 1 | 1 | 15 | eu-north-1b, eu-north-1c, eu-north-1a |
| ✓ | Jamil-Server-Step | Jamil-L4Server-Template \| Version Defaul | 3 | - | 2 | 1 | 15 | eu-north-1b, eu-north-1c, eu-north-1a |

| Status | Description | Cause | Start time | End time |
|---|---|---|---|---|
| WaitingForELBConnectionDraining | Terminating EC2 instance: i-0bb0d093933882332 - Waiting For ELB Connection Draining. | At 2021-11-04T17:20:20Z a monitor alarm Alarm2 in state ALARM triggered policy Jamil-server-Step2-SG changing the desired capacity from 4 to 3. At 2021-11-04T17:20:33Z an instance was taken out of service in response to a difference between desired and actual capacity, shrinking the capacity from 4 to 3. At 2021-11-04T17:20:33Z instance i-0bb0d093933882332 was selected for termination. | 2021 November 04, 06:20:33 PM +01:00 | |
| Successful | Launching a new EC2 instance: i-0bb0d093933882332 | At 2021-11-04T17:13:48Z a monitor alarm Alarm1 in state ALARM triggered policy Jamil-L4-Step-SP changing the desired capacity from 3 to 4. At 2021-11-04T17:14:09Z an instance was started in response to a difference between desired and actual capacity, increasing the capacity from 3 to 4. | 2021 November 04, 06:14:11 PM +01:00 | 2021 November 04, 06:19:28 PM +01:00 |
| Successful | Launching a new EC2 instance: i-0b2e4be3dfe7dcdd5 | At 2021-11-04T17:07:48Z a monitor alarm Alarm1 in state ALARM triggered policy Jamil-L4-Step-SP changing the desired capacity from 2 to 3. At 2021-11-04T17:08:06Z an instance was started in response to a difference between desired and actual capacity, increasing the capacity from 2 to 3. | 2021 November 04, 06:08:08 PM +01:00 | 2021 November 04, 06:13:25 PM +01:00 |
| Successful | Launching a new EC2 instance: i-099b20e7240cb6583 | At 2021-11-04T17:01:48Z a monitor alarm Alarm1 in state ALARM triggered policy Jamil-L4-Step-SP changing the desired capacity from 1 to 2. At 2021-11-04T17:02:03Z an instance was started in response to a difference between desired and actual capacity, increasing the capacity from 1 to 2. | 2021 November 04, 06:02:05 PM +01:00 | 2021 November 04, 06:07:21 PM +01:00 |

Figure 2.12: Autoscale activity

After the attack on the server has stopped, the CPU utilization is as follows:

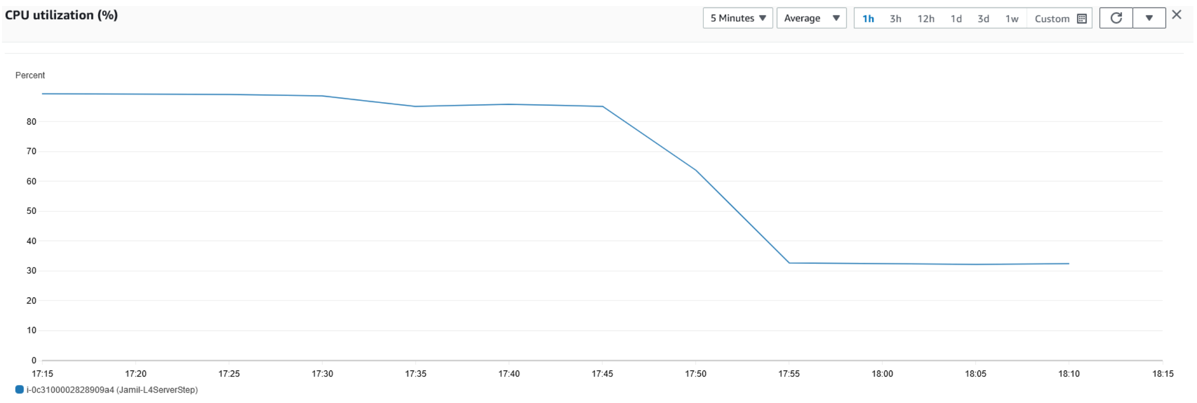
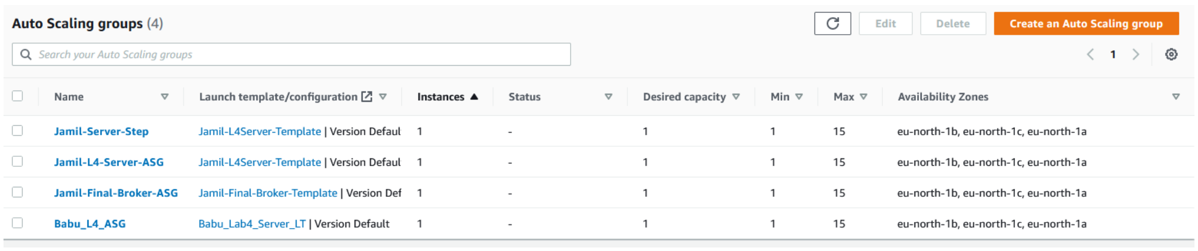


Figure 2.13: CPU Utilization after terminating attack

And the number of instances have now reduced to 1.



| Name | Launch template/configuration | Instances | Status | Desired capacity | Min | Max | Availability Zones |
|---|---|---|---|---|---|---|---|
| Jamil-Server-Step | Jamil-L4Server-Template \| Version Defaul | 1 | - | 1 | 1 | 15 | eu-north-1b, eu-north-1c, eu-north-1a |
| Jamil-L4-Server-ASG | Jamil-L4Server-Template \| Version Defaul | 1 | - | 1 | 1 | 15 | eu-north-1b, eu-north-1c, eu-north-1a |
| Jamil-Final-Broker-ASG | Jamil-Final-Broker-Template \| Version Def | 1 | - | 1 | 1 | 15 | eu-north-1b, eu-north-1c, eu-north-1a |
| Babu_L4_ASG | Babu_Lab4_Server_LT \| Version Default | 1 | - | 1 | 1 | 15 | eu-north-1b, eu-north-1c, eu-north-1a |

Figure 2.14: Number of instances in the autoscale group after terminating attack

To demonstrate how the number of instances created changed with the number of agents during autoscaling, a table was created.

Table 2.1: Table for how the number of instances created changed with the number of agents

| Number of Agents | Number of instances |
|---|---|
| 2000 | 1 |
| 4000 | 2 |
| 9000 | 4 |
| 9350 | 4 |
| 9500 | Ran out of Thread |

# 3
# Part III: The war of worlds

A summary of the finding will be demonstrated below:

1. The autoscale was implemented via step scaling which is superior compared to simple scaling. This is because step scaling allows the setting of dynamic scaling out as well as scaling in policies.

2. Increasing the range of Fibonacci sequence affects both the CPU utilization as well as the time required to process it. To focus solely on the number of agents in the attack, the range of the Fibonacci sequence was to be at 40.

3. On the CPU Utilization parameter load balancing, it takes about 4 instances to serve the server load for 9350 Agent Smiths.

4. At this instant, the server is capable of serving 9350 Agent Smiths. This is not a load balancing issue; but rather a problem of the number of threads that can be initiated at once (jade.tilab, 2010). To solve this a number of solutions were tried unsuccessfully:

   (a) The Architect was moved to an AWS instance to be scaled. However, this involves a GUI operation which needs a VNC Server, it was not possible to scale the Architect to prevent it from running out of threads.

   (b) It was assumed that creating two instances of the Broker and then creating the agents (half with one broker and half with other broker) would solve the problem; however, it failed since it was still running on one machine, and hence ran out of threads.

   (c) It was assumed that perhaps creating half of Agent Smiths in one container and the rest in another container would solve the problem. However, the threads still ran out because it was still running on machine.

   (d) It was assumed that perhaps creating two main containers might solve the problem but the tread creation problem persisted because they were still on the same machine.

   (e) Laptops were connected to mobile hotspot to create an adhoc network. It was then attempted to create half the Agent Smith in one machine and the rest in other remote machine. However this did not work as well because it was not possible to create Agents half and half in one machine and rest in another within one java program (stackoverflow, 2013).

   (f) MainContainer was used in place of AgentContainer to remotely create the Architect and the Broker. This has worked on local but the port on which it was used in, it was blocked and caused the OS to hang severely.

# Bibliography

AWS. (2021). *Step and simple scaling policies for amazon ec2 auto scaling*. Retrieved November 4, 2021, from https://docs.aws.amazon.com/autoscaling/ec2/userguide/as-scaling-simple-step.html

jade.tilab. (2010). *[jade-develop] how many agents can a container cope with?* Retrieved November 4, 2021, from https://webcache.googleusercontent.com/search?q=cache:IPKmLk7sbj4J:https://jade.tilab.com/pipermail/jade-develop/2010q1/014838.html+&cd=2&hl=en&ct=clnk&gl=se

stackoverflow. (2013). *Running multi-platform from an external java application*. Retrieved November 4, 2021, from https://stackoverflow.com/questions/25104854/running-multi-platform-from-an-external-java-application