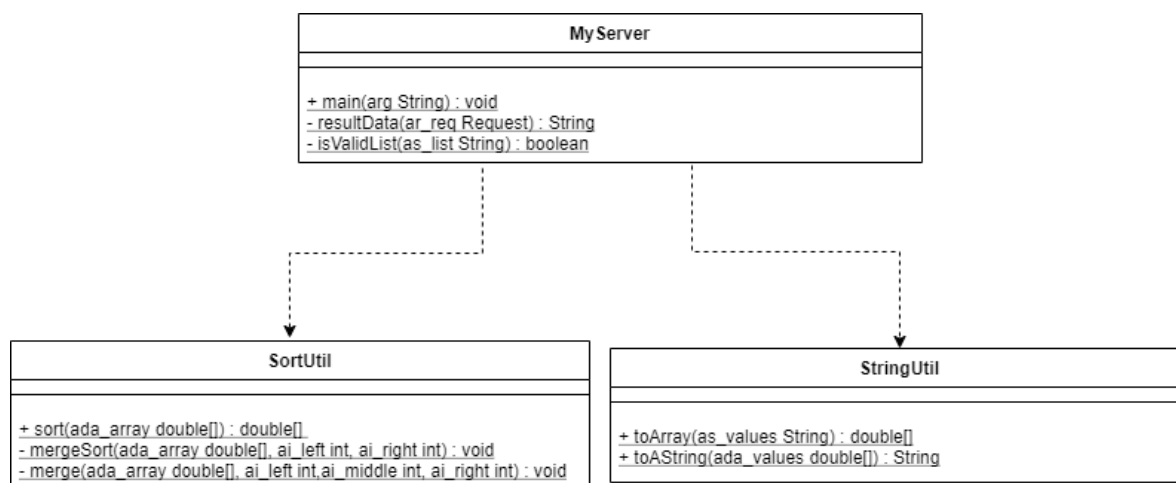


Juan Sebastian Gómez López

Definición del problema.

El problema propuesto en este laboratorio es implementar un servidor web el cual reciba un arreglo de números y devuelva el arreglo ordenado. Este problema con arreglos grandes tiende a requerir una alta cantidad de recursos y procesamiento para poder ordenar el arreglo. Para darle solución a esto se planteó la siguiente arquitectura:



Donde la clase **MyServer** es un servidor Spark el cual publica el path que es capaz de recibir como parámetro la lista que se va a ordenar, luego se apoya en las clases **StringUtil**, para convertir la lista que llega a un arreglo de números y luego la respuesta a **String**, y **SortUtil**, para ordenar el arreglo. El algoritmo de ordenamiento que se implementó dentro de **SortUtil** fue Merge Sort.

Con la aplicación funcionando se creó una imagen de Docker para desplegarla a una instancia EC2 de AWS.

Creación de Auto Scaling group

Primero se crea una instancia EC2 en la cual se va a desplegar la aplicación, para este caso se va a hacer haciendo uso de una imagen de Docker.

```
[ec2-user@ip-172-31-53-10 ~]$ docker ps -a
CONTAINER ID   IMAGE          COMMAND                  CREATED        STATUS        PORTS                               NAMES
0c52fba11728   juance11/taller-7   "java -cp ./classes:..."   39 seconds ago   Up 37 seconds   0.0.0.0:5000->6000/tcp         taller7
```

Una vez con el servicio corriendo debemos asegurarnos de que, tanto Docker como nuestro servicio, se inicien al prender la instancia. Para esto se crea un servicio en `/etc/systemd/system` con nuestro contenedor y lo habilitamos ambos servicios (Docker y el contenedor) para que inicien al prender la máquina.

```

[Unit]
Description=SortService
After=docker.service
Requires=docker.service

[Service]
TimeoutStartSec=0
ExecStartPre=/usr/bin/docker stop taller7
ExecStartPre=/usr/bin/docker rm taller7
ExecStart=/usr/bin/docker run -p 5000:6000 --name taller7 juance11/taller-7

[Install]
WantedBy=multi-user.target

```

Con la maquina configurada creamos una AMI la cual va a tener la información para crear una nueva instancia de nuestra máquina.

Create image

Info

An image (also referred to as an AMI) defines the programs and settings that are applied when you launch an EC2 instance. You can create an image from the configuration of an existing instance.

Instance ID

i-0e7c33232720a7c85

Image name

SortImage

Maximum 127 characters. Can't be modified after creation.

Image description - optional

Image description

Maximum 255 characters

No reboot

☐ Enable

Instance volumes

Volume type	Device	Snapshot	Size	Volume type	IOPS	Delete on termination	Encrypted
EBS	/dev/xvda	Create new snapshot fr...	8	EBS General Purpose SS...	100	<input checked="" type="checkbox"/> Enable	<input type="checkbox"/> Enable

Add volume

During the image creation process, Amazon EC2 creates a snapshot of each of the above volumes.

Cancel

Create image

Ahora creamos un Launch Configuration para configurar como se van a inicializar las maquinas a partir de nuestra imagen. En la creación especificamos el nombre, el AMI de referencia, la llave para conectarnos y el grupo de seguridad donde se van a instanciar las maquinas.

Create launch configuration [Info](#)

Launch configuration name

Name

SortLaunchConfiguration

Amazon machine image (AMI) [Info](#)

AMI

SortImage

Instance type [Info](#)

Instance type

t1.micro (1 vCPUs, 0.5 GiB, EBS Only)

Choose instance type

Security groups [Info](#)

Assign a security group

- ☐ Create a new security group
☒ Select an existing security group

Security groups


Copy to new

View rules

Q Search security groups

< 1 >

	Security group ID	Name	VPC ID	Description
<input checked="" type="checkbox"/>	sg-010fafc06ab16e850	launch-wizard-3	vpc-52d9202f	launch-wizard-3 created 2020-10-26T11:44:20.638-05:00
<input type="checkbox"/>	sg-1aa9cf26	default	vpc-52d9202f	default VPC security group

 Rules with source of 0.0.0.0/0 allow all IP addresses to access your instance. We recommend setting security group rules to allow access from known IP addresses only.

Key pair (login) [Info](#)

Key pair options

Choose an existing key pair

Existing key pair

AREPTALLER5

- ☐ I acknowledge that I have access to the selected private key file (AREPTALLER5.pem), and that without this file, I won't be able to log into my instance.

Ahora creamos el grupo de auto escalamiento a partir del Launch Configuration que creamos anterior mente.

Choose launch template or configuration [Info](#)

Specify a launch template that contains settings common to all EC2 instances that are launched by this Auto Scaling group. If you currently use launch configurations, you might consider migrating to launch templates.

Name

Auto Scaling group name

Enter a name to identify the group.

Must be unique to this account in the current Region and no more than 255 characters.

Launch configuration [Info](#)

[Switch to launch template](#)

Launch configuration

Choose a launch configuration that contains the instance-level settings, such as the Amazon Machine Image (AMI), instance type, key pair, and security groups.

[Create a launch configuration](#)

Launch configuration

Prueba2

Security groups

[sg-010fafc06ab16e850](#)

AMI ID

ami-05ae991a1b5d35b16

Instance type

t2.micro

Date created

Mon Oct 26 2020 15:20:03 GMT-0500
(hora estándar de Colombia)

Key pair name

-

[Cancel](#)[Next](#)

Ahora vamos a crear dos políticas para los procesos de Scale - Out y Scale - in. En la pestaña de Automatic Scaling de nuestro grupo de auto escalado le damos añadir nueva política. Para la creación de la política necesitamos crear una alarma que nos indique el consumo de CPU, le damos en añadir alarma y seleccionamos esta métrica.

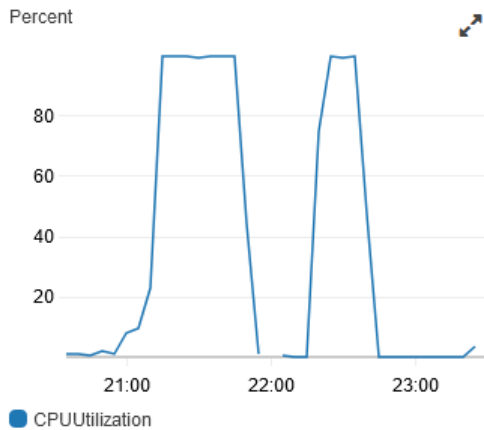
Specify metric and conditions

Metric

[Edit](#)

Graph

This alarm will trigger when the blue line goes above the red line for 1 datapoints within 5 minutes.



Namespace

AWS/EC2

Metric name

CPUUtilization

AutoScalingGroupName

SortGroup

Statistic

Q Average

Period

5 minutes

Adicionalmente le agregamos la condición de que sea mayor al 80% de uso.

Conditions

Threshold type

☒

Static

Use a value as a threshold

☐

Anomaly detection

Use a band as a threshold

Whenever CPUUtilization is...

Define the alarm condition.

☒

Greater

> threshold

☐

Greater/Equal

>= threshold

☐

Lower/Equal

<= threshold

☐

Lower

< threshold

than...

Define the threshold value.

80

Must be a number

► Additional configuration

Cancel

Next

Seleccionamos la alarma que creamos y creamos la politica añadiendo una instancia cuando la condicion de la alarma se cumpla

Policy type

Step scaling ▼

Scaling policy name

Policy1

CloudWatch alarm

Choose an alarm that can scale capacity whenever:

Alarm1 ▼

[Create a CloudWatch alarm](#)

breaches the alarm threshold: CPUUtilization > 80 for 1 consecutive periods of 60 seconds for the metric dimensions:
AutoScalingGroupName = SortGroup

Take the action

Add ▼

1 ▼ capacity units ▼ when 80 ▼ <= CPUUtilization < +infinity

Add step

Instances need

300 ▼ seconds warm up before including in metric

Cancel

Create

Igualmente creamos una politica para el scale in con una alarma cuando el consumo sea menor al 40 %, al finalizar nos deberian quedar nuestras dos politicas creadas.

Policie1

Policy type:
Step scaling

Enabled or disabled?
Enabled

Execute policy when:
Alarm1
breaches the alarm threshold: CPUUtilization > 80 for 1 consecutive periods of 60 seconds for the metric dimensions:
AutoScalingGroupName = SortGroup

Take the action:
Add 1 capacity units when 80 <= CPUUtilization < +infinity

Instances need:
300 seconds to warm up after each step

Policie2

Policy type:
Step scaling

Enabled or disabled?
Enabled

Execute policy when:
Alarm2
breaches the alarm threshold: CPUUtilization <= 40 for 1 consecutive periods of 300 seconds for the metric dimensions:
AutoScalingGroupName = SortGroup

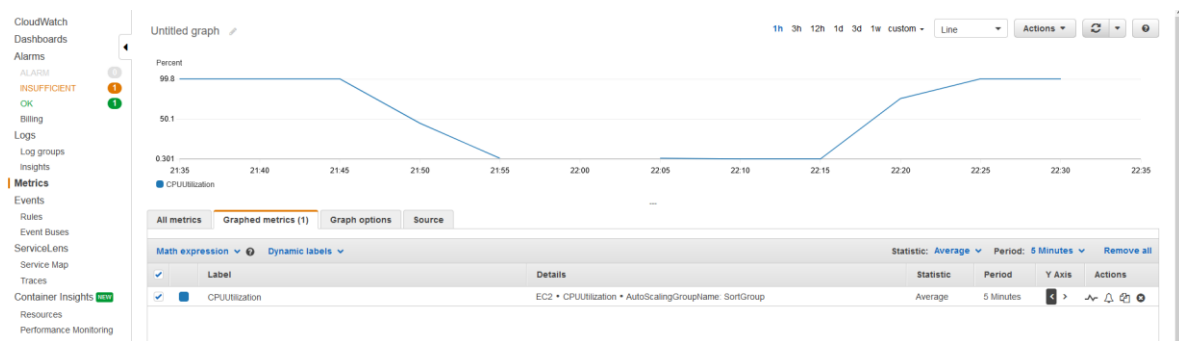
Take the action:
Remove 1 capacity units when 40 >= CPUUtilization > -infinity

PRUEBA

The screenshot shows the AWS Management Console for an EC2 Auto Scaling group named 'SortGroup'. The 'Instance management' tab is selected. The 'Instances (1)' section shows a single instance with ID 'i-0d5e8d9dfb6...', in the 'InService' state, of type 't2.micro', with a 'Weighted capacity' of '-'. The instance is located in 'us-east-1a' and is 'Healthy'. The 'Lifecycle hooks (0)' section is empty, with a 'Create lifecycle hook' button.

Se inicia en 1 sola instancia y porcentaje de uso de CPU en 0. Con la herramienta Apache Jmeter vamos a sobrecargar de peticiones la instancia que tenemos inicialmente

Cuando llega a una carga muy alta se ve la creación de una nueva instancia para suplir la carga.



The screenshot shows the AWS Management Console for the same EC2 Auto Scaling group 'SortGroup'. The 'Instances (2)' section now shows two instances. The first instance, 'i-024c5b0b6a...', is in the 'InService' state, of type 't2.micro', with a 'Weighted capacity' of '-'. The second instance, 'i-0d5e8d9dfb6...', is also in the 'InService' state, of type 't2.micro', with a 'Weighted capacity' of '-'. Both instances are located in 'us-east-1a' and are 'Healthy'. The 'Lifecycle hooks (0)' section remains empty.

Incluso después de un tiempo, al ver que la carga seguía en aumento agrega una nueva instancia.

Instances (3)								
<input type="text" value="Filter instances"/> Refresh Actions								
<div> < 1 > ⊗ </div>								
<input type="checkbox"/>	Instance ID ▲	Lifecycle ▼	Instance type ▼	Weighted capacity ▼	Launch template/configuration ▼	Availability Zone ▼	Health status ▼	Protected from ▼
<input type="checkbox"/>	i-006276406f...	InService	t2.micro	-	Prueba2	us-east-1a	Healthy	Scale in
<input type="checkbox"/>	i-024c5b0b6a...	InService	t2.micro	-	Prueba2	us-east-1a	Healthy	Scale in
<input type="checkbox"/>	i-0d5e8d9dfb6...	InService	t2.micro	-	Prueba2	us-east-1a	Healthy	

Cuando se llega a la tercera instancia se terminamos el proceso de Jmeter para ver la política de Scale-in del grupo de auto escalado

Cuando se activa la alerta de consumo bajo se terminan las dos instancias creadas

Instances (1)								
<input type="text" value="Filter instances"/> Refresh Actions								
<div> < 1 > ⊗ </div>								
<input type="checkbox"/>	Instance ID ▼	Lifecycle ▼	Instance type ▼	Weighted capacity ▼	Launch template/configuration ▼	Availability Zone ▼	Health status ▼	Protected from ▼
<input type="checkbox"/>	i-006276406f...	InService	t2.micro	-	Prueba2	us-east-1a	Healthy	Scale in