Escuela colombiana de ingeniería Julio Garavito

Taller de Patrones de Arquitectura-ESB – PhD. Luis Daniel Benavides Navarro

Juan pablo Ospina Henao

Instalación ServeMix

Requerimientos

For running Apache ServiceMix itself, you'll need

- Java Runtime Environment (JRE) 1.6.x (Java 6) or
  Java Runtime Environment (JRE) 1.7.x (Java 7)
- About 100 MB of free disk space for the default assembly

If you're developing your own integration applications and OSGi bundles, you'll also need

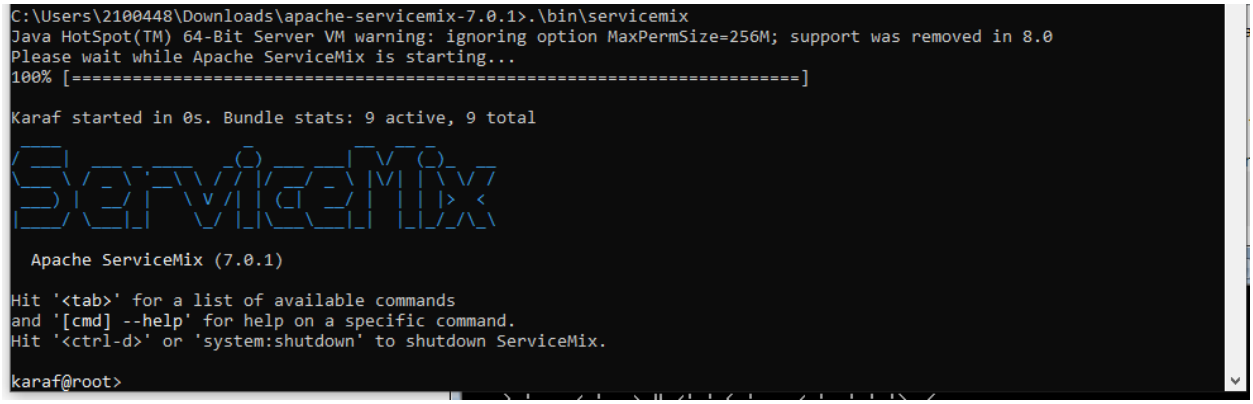
- Java Developer Kit (JDK) 1.6.x (Java 6) or
  Java Developer Kit (JDK) 1.7.x (Java 7)
- Apache Maven 3.0.4 or higher

Descargar ServiceMix

1. Se debe acceder al siguiente Url para acceder al recurso de ServiceMix

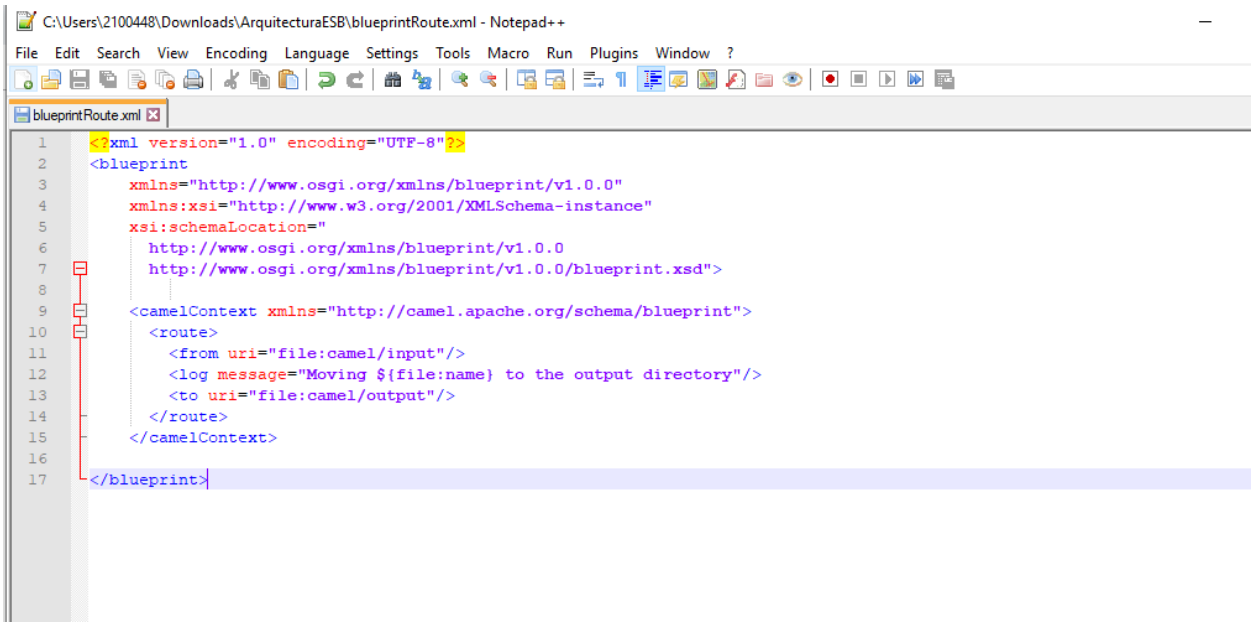
https://www-eu.apache.org/dist/servicemix/servicemix-7/7.0.1/apache-servicemix-7.0.1.zip

2. Pasamos a descomprimir el archivo .zip y se instala el servicio en la consola de comandos



Configuración ServiceMix

Debemos crear un archivo en el directorio deploy el cual contiene el archivo ruta

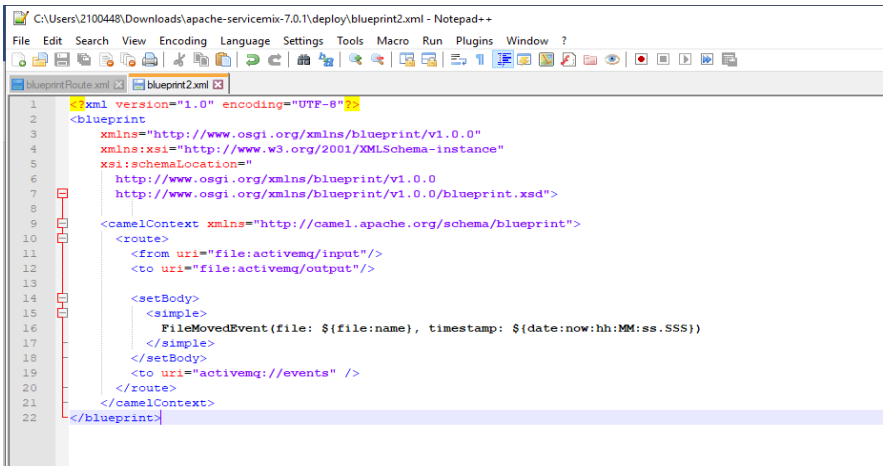


Para ver el funcionamiento de la ruta se prueba pasando un archivo en camel/input y debe salir un archivo en camel/output

Para corroborar se debe digitar el log:display para ver si se creo una ruta

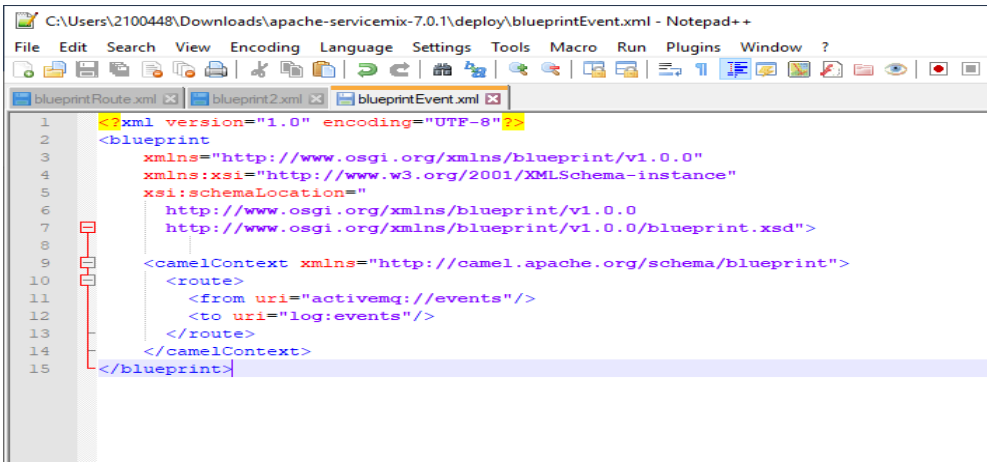
Ahora se debe realizar el mismo procedimiento, pero implementado activeMQ

Creamos el archivo blueprint2.xml



Para corroborar el funcionamiento pasamos a realizar el paso de archivo de input a output en

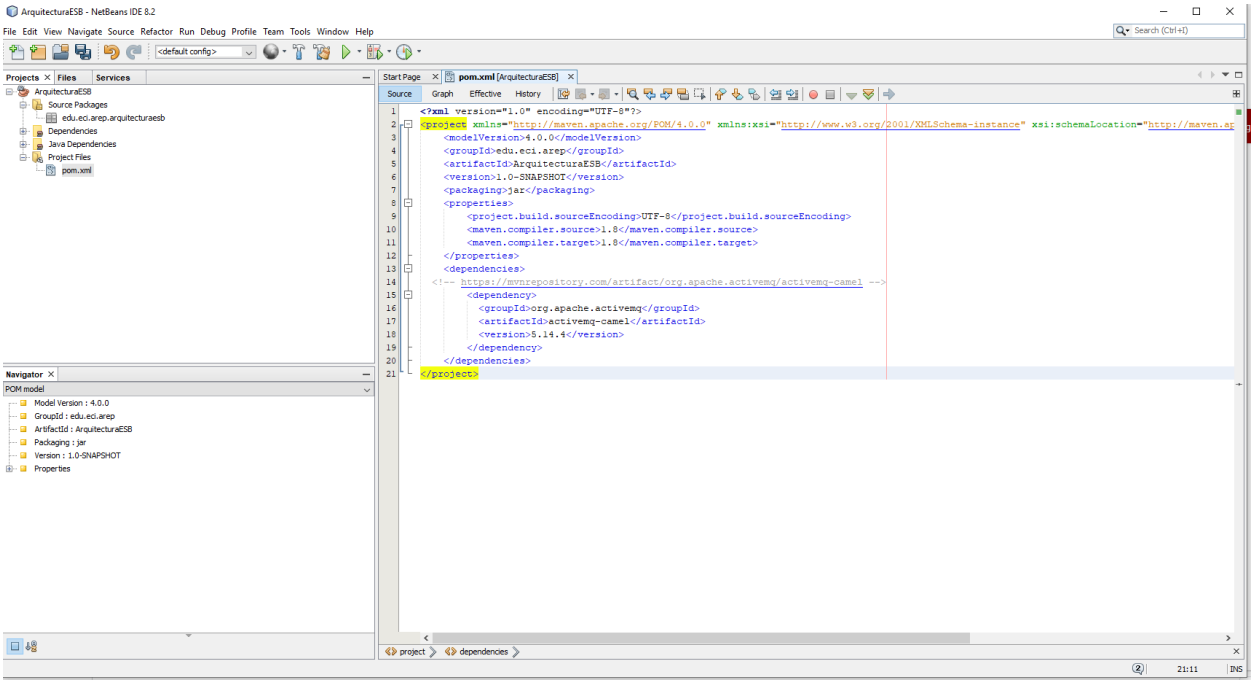
Con le fin de capturar todos los eventos del canal respectivo activemq se debe agregar el siguiente archivo en el directorio deploy



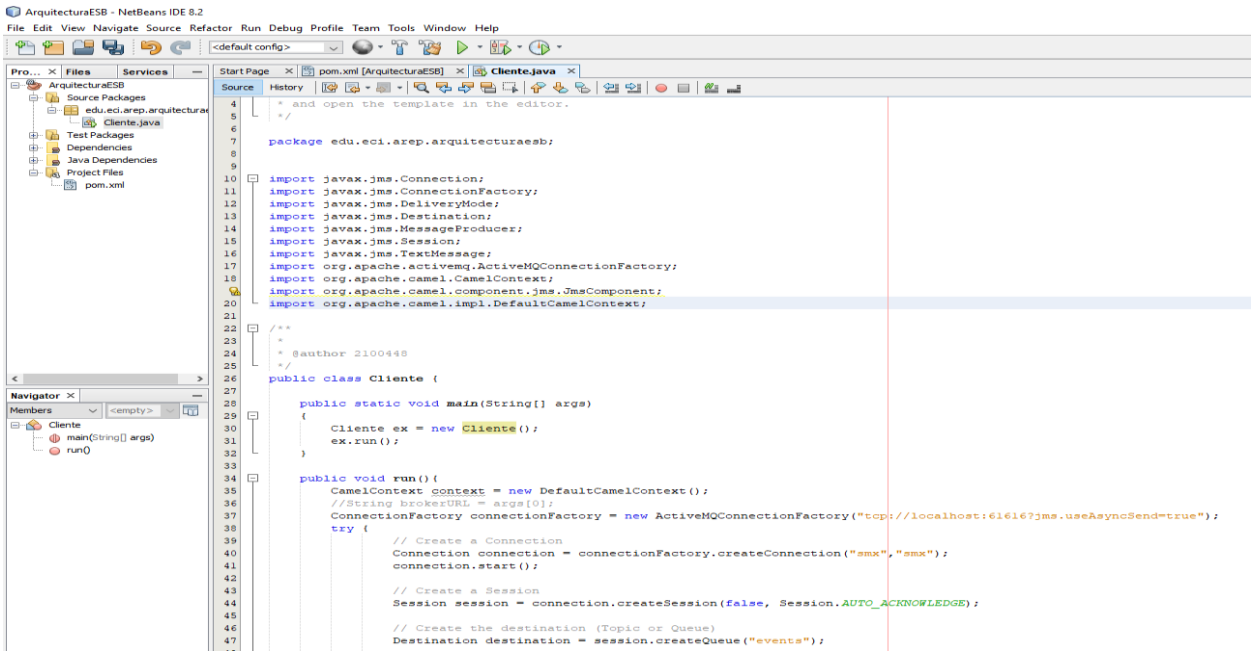
Para corroborar que si estén llegando los eventos en el log de consola.

Arquitectura ESB

Se crea un proyecto java Maven en NetBeans y Agregamos las dependencias requeridas:

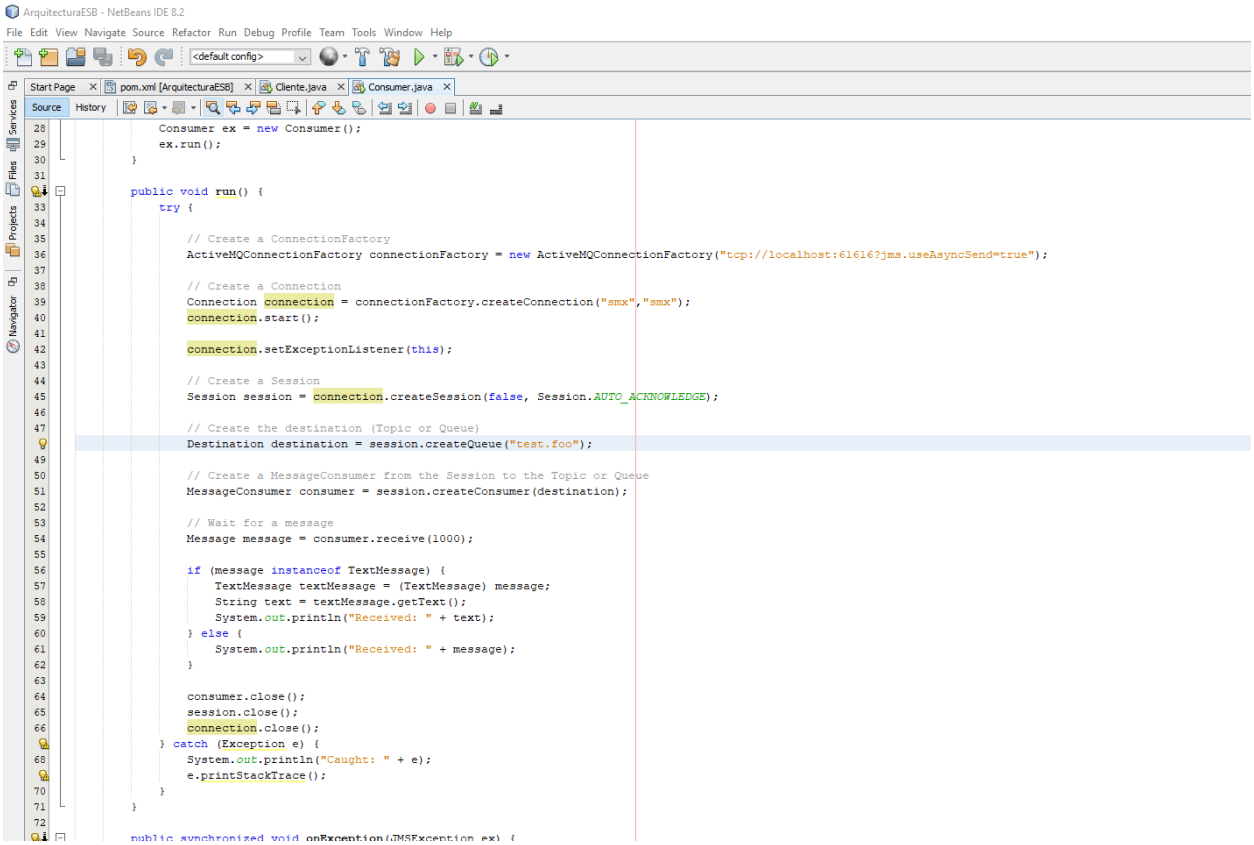


Y creamos la clase cliente como se ve a continuación:



Se debe crear una clase consumidor la cual es la que recibe todos los mensajes que crea em el cliente

Se deben configurar la seguridad y la ruta entre los dos como se muestra en la siguiente imagen:



Mensaje

Cliente

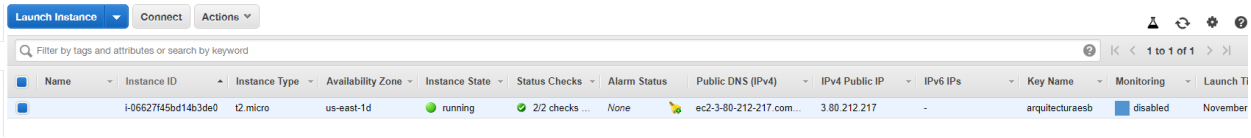
```
Sent message: 1354011814 : main
------------------------------------------------------------------------
BUILD SUCCESS
```

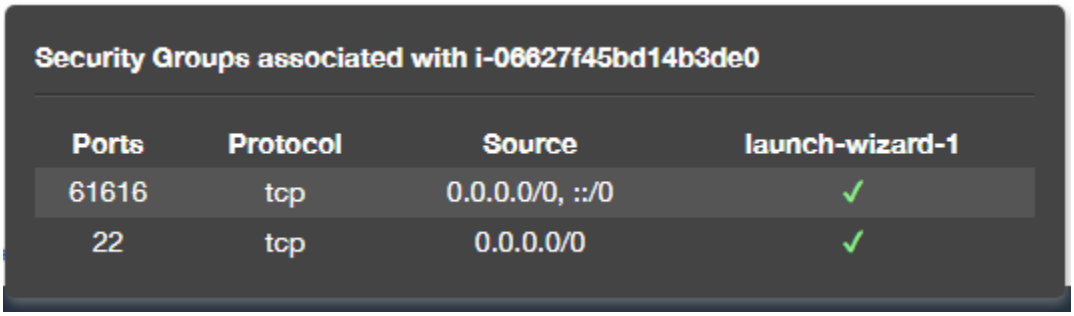
Consumer

```
SLF4J: See http://www.slf4j.org/codes.html#StaticLoggerBinder for further details.
Received: Hello world! From: main : 1740189450
------------------------------------------------------------------------
BUILD SUCCESS
```

EC2 AWS

Para ello debemos habilitar el puerto 61616 en el security group

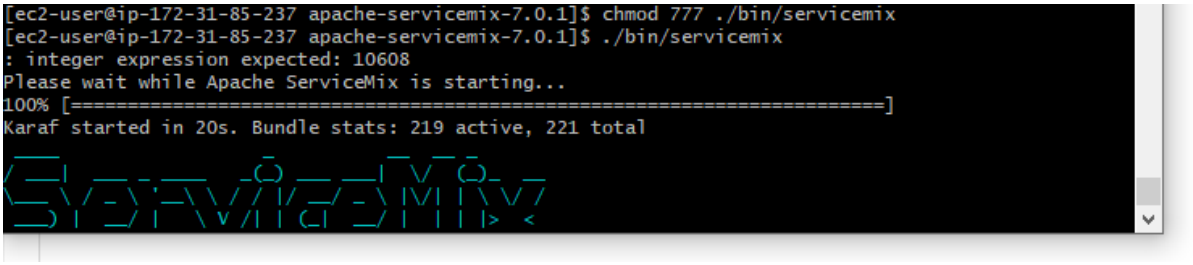




Y se conecta a la maquina

Debemos pasar el código fuente de servieMX para ello usamos sftp para pasar archivos desde nuestra maquina hasta la alojada en aws.

Pasamos luego a correr el servicio



Y por último cambiamos en el código de cliente y consumidor la url donde se conecta

Cliente

```
public class Cliente {

    public static void main(String[] args)
    {
        Cliente ex = new Cliente();
        ex.run();
    }

    public void run(){
        CamelContext context = new DefaultCamelContext();
        //String brokerURL = args[0];
        ConnectionFactory connectionFactory = new ActiveMQConnectionFactory("tcp://ec2-3-80-212-217.compute-1.amazonaws.com:61616?jms.useAsyncSend=true");
        try {
                // Create a Connection
                Connection connection = connectionFactory.createConnection("smx","smx");
                connection.start();

                // Create a Session
                Session session = connection.createSession(false, Session.AUTO_ACKNOWLEDGE);

                // Create the destination (Topic or Queue)
                Destination destination = session.createQueue("test.foo");
```

Consumer

```
public class Consumer implements Runnable, ExceptionListener {
        public static void main(String[] args)
        {
            Consumer ex = new Consumer();
            ex.run();
        }

        public void run() {
            try {

                // Create a ConnectionFactory
                ActiveMQConnectionFactory connectionFactory = new ActiveMQConnectionFactory("tcp://ec2-3-80-212-217.compute-1.amazonaws.com:61616?jms.useAsyncSend=true");

                // Create a Connection
                Connection connection = connectionFactory.createConnection("smx","smx");
                connection.start();
```

Luego pasamos a probar el servicio en aws

Desde el cliente mandamos el mensaje:

```
    public void run(){
        CamelContext context = new DefaultCamelContext();
        //String brokerURL = args[0];
        ConnectionFactory connectionFactory = new ActiveMQConnectionFactory("tcp://ec2-3-80-212-217.compute-1.amazonaws.com:61616?jms.useAsyncSend=true");
        try {
                // Create a Connection
                Connection connection = connectionFactory.createConnection("smx","smx");
                connection.start();

                // Create a Session
                Session session = connection.createSession(false, Session.AUTO_ACKNOWLEDGE);
```

```
tput - Run (ArquitecturaESB)
  Running NetBeans Compile On Save execution. Phase execution is skipped and output directories of dependency projects (with Compile on Save turned on) will be used instead of their jar artifa
  Scanning for projects...

  ------------------------------------------------------------------------
  Building ArquitecturaESB 1.0-SNAPSHOT
  ------------------------------------------------------------------------

  --- exec-maven-plugin:1.2.1:exec (default-cli) @ ArquitecturaESB ---
  SLF4J: Failed to load class "org.slf4j.impl.StaticLoggerBinder".
  SLF4J: Defaulting to no-operation (NOP) logger implementation
  SLF4J: See http://www.slf4j.org/codes.html#StaticLoggerBinder for further details.
  Sent message: 846238611 : main
```

Y revisamos el mensaje recibido:

```
public class Consumer implements Runnable, ExceptionListener {
    public static void main(String[] args)
    {
        Consumer ex = new Consumer();
        ex.run();
    }

    public void run() {
        try {

            // Create a ConnectionFactory
            ActiveMQConnectionFactory connectionFactory = new ActiveMQConnectionFactory("tcp://ec2-3-80-212-217.compute-1.amazonaws.com:61616?jms.useAsyncSend=true");

            // Create a Connection
            Connection connection = connectionFactory.createConnection("smx","smx");
            connection.start();
```

```
ut - Run (Consumer)
 Running NetBeans Compile On Save execution. Phase execution is skipped and output directories of dependency projects (with Compile on Save turned on) will be used instead of their jar artifacts.
 Scanning for projects...

 ------------------------------------------------------------------------
] Building ArquitecturaESB 1.0-SNAPSHOT
 ------------------------------------------------------------------------

] --- exec-maven-plugin:1.2.1:exec (default-cli) @ ArquitecturaESB ---
 SLF4J: Failed to load class "org.slf4j.impl.StaticLoggerBinder".
 SLF4J: Defaulting to no-operation (NOP) logger implementation
 SLF4J: See http://www.slf4j.org/codes.html#StaticLoggerBinder for further details.
- Received: Hello world! From: main : 1709366259
 ------------------------------------------------------------------------
```

Diagrama de arquitectura

Diagrama de despliegue