

Guía Laboratorio Kurento

Kurento es un Media Server con WebRTC y un conjunto de APIs de cliente que simplifican el desarrollo de aplicaciones de video avanzadas para plataformas web y smartphone. Las características de Kurento Media Server incluyen comunicaciones de grupo, transcodificación, grabación, mezcla, difusión y enrutamiento de flujos audiovisuales.

El servidor de medios WebRTC es sólo una especie de "middleware multimedia" (que se encuentra en medio de las partes que establecen una comunicación), permitiendo gestionar el tráfico compartido entre las partes.

Los Media Server son capaces de procesar flujos de tráfico de tipo media y ofrecer diferentes características:

- **Grupos de Comunicación:** Distribuye el flujo de tráfico de media que un par de usuarios genera entre varios receptores, es decir, se actúa como unidad de conferencia múltiple, MCU.
- **Mezclar:** transformar varios flujos de tráfico entrantes en un único flujo de media compuesto.
- **Transcodificación:** adaptación de codecs y formatos entre clientes incompatibles.
- **Grabación:** almacenando de forma persistente del tráfico intercambiados entre las partes.

De esta manera el esquema de utilización de un media server comparado con una comunicación Peer-to-peer es el siguiente:

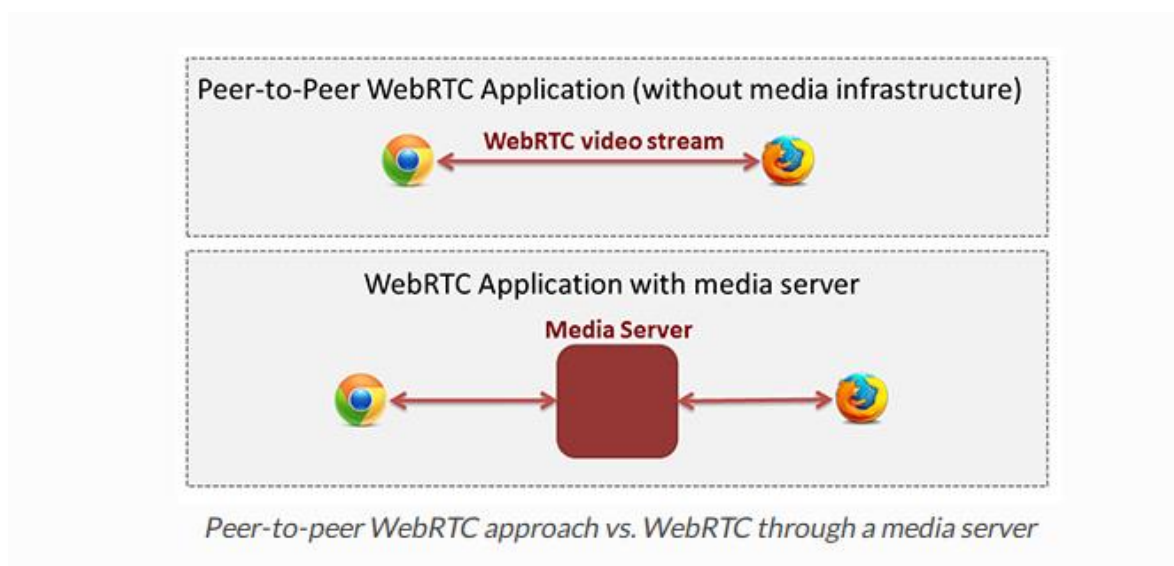


Figura 1. WebRTC vs WebRTC Media Server

Para este laboratorio es necesario tener lo siguiente:

Guide Develop By: Juan Felipe Gómez Manzanares. M Sc
Telematic Engineer.



- Ubuntu 16.04 LTS. Se recomienda instalarlo en el ambiente virtual de Proxmox.
- Computador con cámara.

Instalación Kurento.

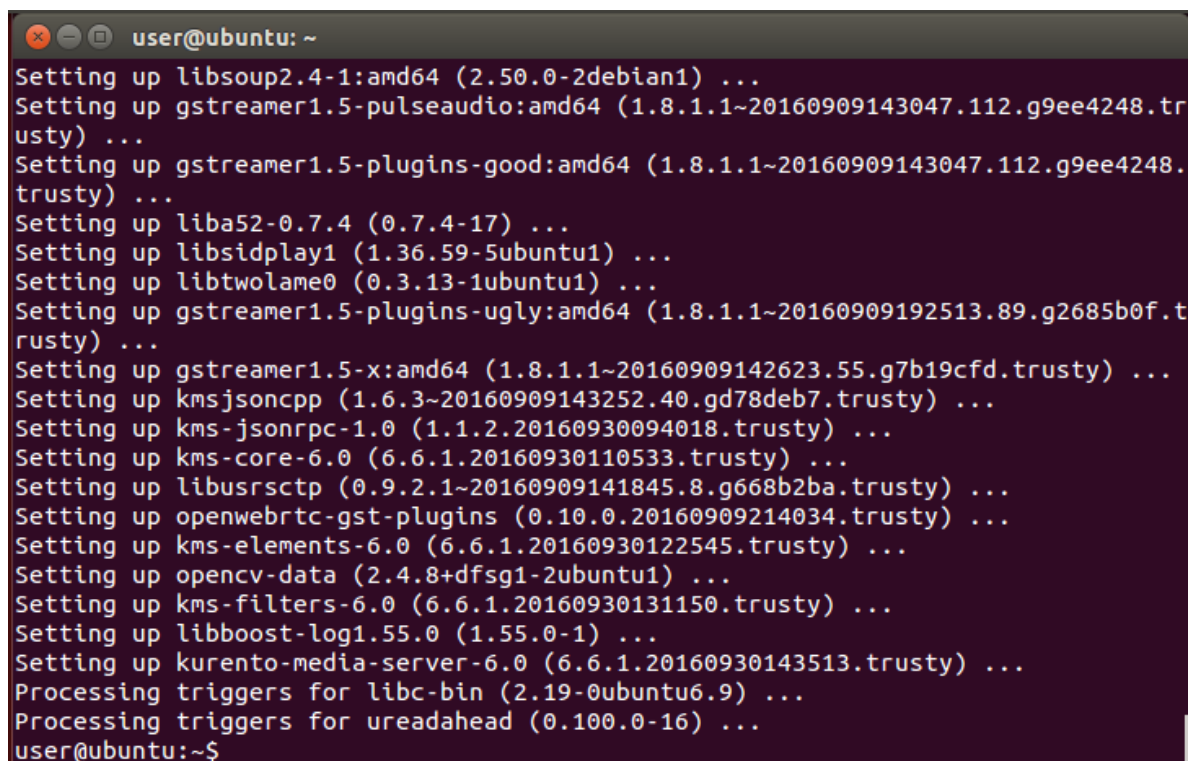
Para el proceso de instalación del Media Server Kurento, nos basaremos en la siguiente guía:

https://doc-kurento.readthedocs.io/en/stable/user/installation_dev.html

Ejecutamos los siguientes comandos en la terminal de Ubuntu:

```
echo "deb http://ubuntu.kurento.org trusty kms6" | sudo tee
/etc/apt/sources.list.d/kurento.list
wget -O - http://ubuntu.kurento.org/kurento.gpg.key | sudo apt-key
add -
sudo apt-get update
sudo apt-get install kurento-media-server-6.0
```

Al finalizar tendremos el siguiente mensaje de confirmación:



```
user@ubuntu: ~
Setting up libsoup2.4-1:amd64 (2.50.0-2debian1) ...
Setting up gstreamer1.5-pulseaudio:amd64 (1.8.1.1~20160909143047.112.g9ee4248.tr
usty) ...
Setting up gstreamer1.5-plugins-good:amd64 (1.8.1.1~20160909143047.112.g9ee4248.
trusty) ...
Setting up liba52-0.7.4 (0.7.4-17) ...
Setting up libsndfile1 (1.36.59-5ubuntu1) ...
Setting up libtwolame0 (0.3.13-1ubuntu1) ...
Setting up gstreamer1.5-plugins-ugly:amd64 (1.8.1.1~20160909192513.89.g2685b0f.t
rusty) ...
Setting up gstreamer1.5-x:amd64 (1.8.1.1~20160909142623.55.g7b19cfd.trusty) ...
Setting up kmsjsoncpp (1.6.3~20160909143252.40.gd78deb7.trusty) ...
Setting up kms-jsonrpc-1.0 (1.1.2.20160930094018.trusty) ...
Setting up kms-core-6.0 (6.6.1.20160930110533.trusty) ...
Setting up libusrsctp (0.9.2.1~20160909141845.8.g668b2ba.trusty) ...
Setting up openwebrtc-gst-plugins (0.10.0.20160909214034.trusty) ...
Setting up kms-elements-6.0 (6.6.1.20160930122545.trusty) ...
Setting up opencv-data (2.4.8+dfsg1-2ubuntu1) ...
Setting up kms-filters-6.0 (6.6.1.20160930131150.trusty) ...
Setting up libboost-log1.55.0 (1.55.0-1) ...
Setting up kurento-media-server-6.0 (6.6.1.20160930143513.trusty) ...
Processing triggers for libc-bin (2.19-0ubuntu6.9) ...
Processing triggers for ureadahead (0.100.0-16) ...
user@ubuntu:~$
```

Figura 2. Finalización proceso Instalación

Procedemos a Iniciar Kurento y continuar con la guía:

```
sudo service kurento-media-server start
```

Laboratorio Inicial Kurento

Una vez instalado Kurento, ya podemos realizar nuestra primera aplicación utilizando el media server. Para este caso usaremos una aplicación ya desarrollada en Java por el equipo de Kurento.

Antes de empezar debemos instalar Apache Maven y Java JDK usando los siguientes comandos:

```
sudo apt-get install maven
sudo apt-get install default-jdk
```

Una vez instalados ambos paquetes podemos proceder con la descarga e implementación de la aplicación Demo de Kurento:

```
git clone https://github.com/Kurento/kurento-tutorial-java.git
cd kurento-tutorial-java/kurento-hello-world
git checkout 6.6.0
mvn compile exec:java
```

Una vez se esté ejecutando la aplicación, tendremos el siguiente mensaje en la terminal y podremos acceder a la aplicación a través de: <https://localhost:8443/>

```
handler of type [class org.springframework.web.servlet.resource.ResourceHttpRequestHandler]
2017-04-19 09:27:39.319 INFO 19114 --- [WorldApp.main()] o.s.w.s.handler.SimpleUrlHandlerMapping : Mapped URL path [/**] onto handler
of type [class org.springframework.web.servlet.resource.ResourceHttpRequestHandler]
2017-04-19 09:27:39.428 INFO 19114 --- [WorldApp.main()] o.s.w.s.handler.SimpleUrlHandlerMapping : Mapped URL path [/**/favicon.ico] o
nto handler of type [class org.springframework.web.servlet.resource.ResourceHttpRequestHandler]
2017-04-19 09:27:39.874 INFO 19114 --- [WorldApp.main()] o.s.j.e.a.AnnotationMBeanExporter : Registering beans for JMX exposure
on startup
2017-04-19 09:27:39.903 INFO 19114 --- [WorldApp.main()] o.s.c.support.DefaultLifecycleProcessor : Starting beans in phase 2147483647
2017-04-19 09:27:40.844 INFO 19114 --- [WorldApp.main()] s.b.c.e.t.TomcatEmbeddedServletContainer : Tomcat started on port(s): 8443 (ht
tps)
2017-04-19 09:27:40.870 INFO 19114 --- [WorldApp.main()] o.k.tutorial.helloworld.HelloWorldApp : Started HelloWorldApp in 8.357 seco
nds (JVM running for 23.72)
```

Figura 3. Aplicación Media Server Inicia

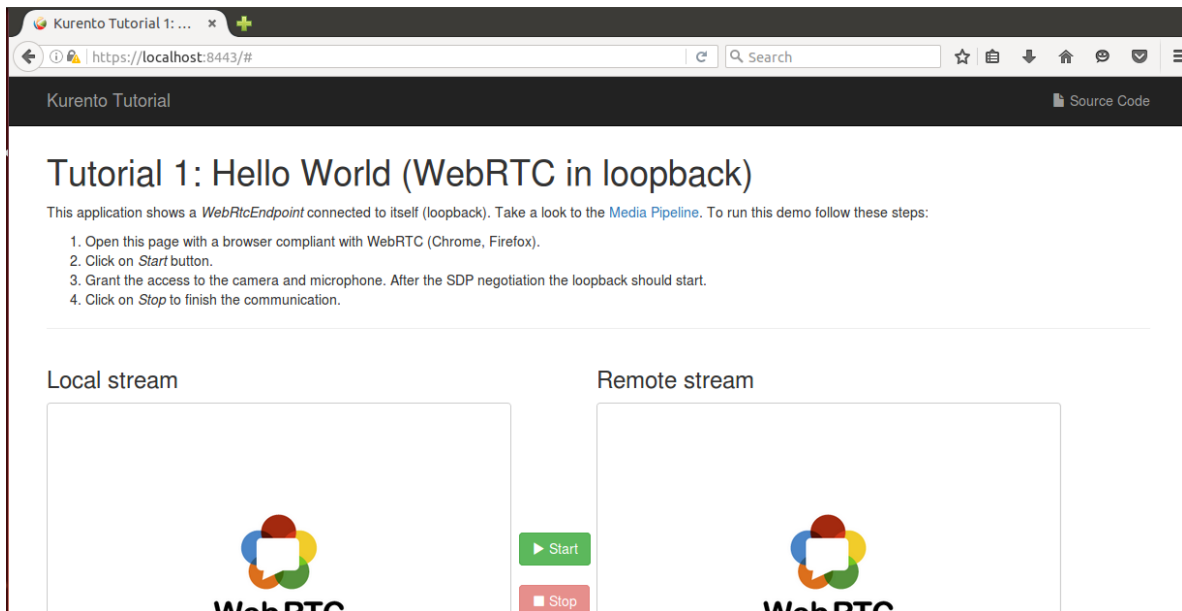


Figura 4. Aplicación Web KurentoMediaServer.

En caso de tener problemas para tener acceso remoto a la aplicación de Kurento, podemos compilar la aplicación de la siguiente manera:

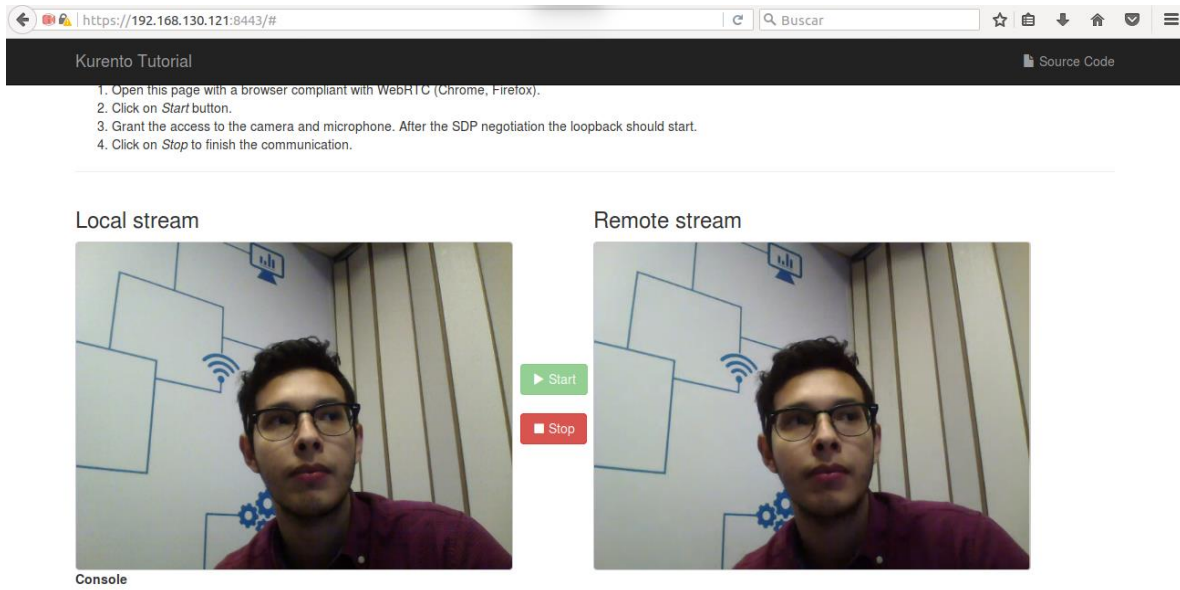
```
mvn compile exec:java -Dkms.url=ws://IP_KMS:KMS_PORT/kurento
```

Una vez damos al botón de start se inicia el streaming y tendremos algo muy parecido al primer laboratorio realizado de WebRTC, como se aprecia en la siguiente captura de pantalla:

Por último, para lograr tener un comportamiento seguro y adecuado de la aplicación deberemos configurar los Secure Sockets, basándonos en la siguiente guía: <http://bit.ly/2q1rOGy>

Importante solo seguir los pasos destinados para Java.

De esta manera terminamos el laboratorio del hola mundo.



- ? Como pueden apreciar dentro del archivo descargado del repositorio de GitHub se encuentran otros ejemplos de las funcionalidades de utilizar Media Servers como Kurento. Para el informe de este laboratorio deben probar al menos 2 de los ejemplos brindados por Kurento, explicando de manera clara:
- Objetivo del ejemplo seleccionado.
 - Funcionalidades que utiliza.
 - Protocolo de comunicación.
 - Aplicaciones en un entorno empresarial.

Estos ejemplos que utilicen no se pueden repetir entre las parejas, es decir, deben ponerse de acuerdo entre los grupos para no repetir ejemplos. Ejemplo repetido no será tenido en cuenta. Al igual que como ocurrió en el primer laboratorio de WebRTC, ambos integrantes de las parejas deben aparecer en las capturas.