Live Streaming with Nginx, RTMP and Kaltura

Jess Portnoy jess.portnoy@kaltura.com, Kaltura, Inc
Abstract

The session will walk attendees through configuring **Nginx** with the **RTMP module** and the **Kaltura HTML5 player** to achieve a fully functional, end to end, FOSS, live streaming solution.

During the session, we will review the various components needed for the solution and conclude with a demo consisting of streaming to **Nginx** over RTMP using **FFmpeg** and playing the stream using the **Kaltura HTML5 player**.

---

**Live Streaming with Nginx, RTMP and Kaltura | FOSDEM 2017**
Main Components

- Nginx [BSD]
- Nginx RTMP Module [BSD]
- FFmpeg [GPL]
- Kaltura Server [AGPLv3]
- Kaltura HTML5 Player [AGPLv3]
Installing Kaltura CE [Community Edition] is easy. We ship both RPM and Deb packages and installation instructions can be found here:

https://github.com/kaltura/platform-install-packages/#documentation-and-guides

The kaltura-server meta package declares all required dependencies and so, for a single server deployment, issuing:

```bash
# $YOUR_FAVOURITE_PACKAGE_MANAGER install kaltura-server
```

should do the trick.
Some Important Packages

- kaltura-nginx: Nginx is available in the official repos of many Linux distributions. The reason we package our own rather than use the official distros package is that Nginx only introduced DSO [Dynamic Shared Objects] support starting from version 1.9.11

- kaltura-base: includes the Kaltura Server code which lives in this repo: https://github.com/kaltura/server

- kaltura-front: a meta package declaring all dependencies needed to run a Kaltura frontend server [Kaltura Management Console, Kaltura’s HTML5 player, FFmpeg and so on]
The paths to the configuration files for kaltura-nginx vary depending on whether you use the Deb or the RPM package. For Deb, the main file is here:

```
/opt/kaltura/nginx/conf/nginx.conf
```

For RHEL:

```
/etc/nginx/nginx.conf
```

As noted previously, the Nginx shipped with the kaltura-nginx package includes the nginx-rtmp-module which in this solution is used for both the streaming and delivery. Here is the basic RTMP module configuration shipped with kaltura-nginx:
rtmp {
server {
  listen 1935; # Listen on standard RTMP port
  chunk_size 4000;

  # This application is to accept incoming stream
  application kLive {
    live on; # Allows live input from above
    dash on; # create DASH fragments and manifest
    # Sets MPEG-DASH playlist and fragment directory
    dash_path /var/tmp/dashme;
    hls on; # create HLS fragments and manifest
    hls_cleanup on;
    hls_sync 100ms;
    hls_fragment 2s;
    # Sets HLS playlist and fragment directory
    hls_path /var/tmp/hlsme/;
  }
}
}
Creating a Live Stream Entry

Next, we need to create the Kaltura Live Entry that will make use of the HLS stream, this is done by going to KMC->Upload->Live Stream Entry.

For Live Stream Type, select Manual Live Stream URLs, provide a meaningful name for the entry, for example My Live Test, and input

http://$YOUR_NGINX_HOST:$NGINX_PORT/hlsme/$DESIRED_STREAM_NAME.m3u8 in the “HLS stream URL” text box.
Streaming

There are several commercial tools for RTMP streaming, perhaps the most common of these is Adobe’s FMLE. While FMLE can certainly be used, in the spirit of FOSS, we will use the ffmpeg CLI binary to stream.

Kaltura makes heavy use of FFmpeg in its transcoding mechanism so every Kaltura Server includes our own build of ffmpeg, provided by the kaltura-ffmpeg package. If you’d rather stream from your desktop and that desktop is not running Linux, you can get ffmpeg binaries for your OS here:

https://ffmpeg.org/download.html
Here is the ffmpeg command to use:

```plaintext
ffmpeg -re -i /path/too/your/video/file -c copy -f flv \
"rtmp://$NGINX_HOST:$PORT/kLive/$DESIRED_STREAM_NAME"
```
Testing playback

Now, all that's left is playing the stream. Go to KMC->your Live Stream Entry->Actions->Preview & Embed, select your HTML5 player from the list and hit play:)

In the same view, you will also find the HTML code needed to embed the player onto external websites.
Conclusion and Next Steps

In this brief tutorial, we’ve detailed how to achieve a completely free and open live video streaming solution.

An important next step is to restrict publishing access [and perhaps playback too, depending on your needs]. To that end, see https://github.com/arut/nginx-rtmp-module/wiki/Directives#access

The Nginx RTMP module also supports many additional, well documented options that are worth exploring: https://github.com/arut/nginx-rtmp-module/wiki/Directives
Future Plans

We plan to achieve a deeper level of integration between Kaltura and the Nginx RTMP module and allow for DVR and automatic stream provisioning.
Credit Where Credit Is Due

My deepest thanks to the *Nginx* and *FFmpeg* projects and to *Roman Arutyunyan* for creating the *Nginx RTMP module* project.
Thank you & Questions
Appendix - Useful Resources

- Nginx Documentation
- Nginx RTMP module
- FFmpeg Documentation
- Kaltura Server Documentation
- Kaltrua CE Packagnig Deb Specs
- Kaltrua CE Packagnig RPM Specs
- Kaltura HTML5 Player GitHub Repo
- Kaltura HTML5 Player Documentation