



Serving Media with NGINX Plus

Table of Contents

3	About NGINX Plus
3	Using this Guide
4	Prerequisites and System Requirements
5	Serving Media with NGINX Plus
9	NGINX Plus for Streaming Media – Technical Detail

This guide explains how to use NGINX Plus as a streaming media server. It provides instructions for configuring NGINX Plus for both live and on demand video streams.

About NGINX Plus

[NGINX Plus](#) is the commercially supported version of the open source NGINX software. NGINX Plus is a complete application delivery platform, extending the power of NGINX with a host of enterprise-ready capabilities that are instrumental to building web applications at scale:

- Full-featured HTTP and TCP load balancing
- High-performance reverse proxy
- Caching and offload of dynamic and static content
- Adaptive streaming to deliver audio and video to any device
- Application-aware health checks and high availability
- Advanced activity monitoring available via a dashboard or API
- Management and real-time configuration changes with DevOps-friendly tools

Using this Guide

After reviewing the "Prerequisites and System Requirements" on [page 4](#), perform the instructions in the following sections.

Prerequisites and System Requirements

- Linux system to host NGINX Plus (in on-premise and private-cloud deployments). To avoid potential conflicts with other applications, we recommend you install NGINX Plus on a fresh system. For the list of Linux distributions supported by NGINX Plus, see [NGINX Plus Technical Specifications](#).
- NGINX Plus R6 or later.

The instructions assume you have basic Linux system administration skills, including the following. Full instructions are not provided for these tasks.

- Installing Linux software from vendor-supplied packages
- Editing configuration files
- Copying files between a central administrative system and Linux servers
- Running basic commands to start and stop services
- Reading log files

Serving Media with NGINX Plus

NGINX Plus provides a powerful solution for most streaming media needs, utilizing the famous performance and reliability of the open source NGINX software.

At NGINX, our customers often ask how best to use NGINX and NGINX Plus to solve their media streaming needs, including:

- They want to replace their current media streaming solution because of unsatisfactory performance, reliability, or ease of use. They are therefore looking to integrate a new origin media server into their existing infrastructure.
- They are building a new infrastructure for media streaming and are looking for a high-performance and reliable solution that can act as an origin media server while also providing load balancing and caching.

Some other streaming media solutions specifically target and limit themselves to media functionality, but end up lacking the level of performance and reliability customers demand. NGINX Plus instead provide a comprehensive application delivery platform with proxying, load balancing, caching, and web serving along with features specifically designed for streaming media.

The open source NGINX software is famous for high performance and reliability; it's used by over 130 million sites overall, including over 40% of the world's 10,000 busiest websites. It is already a popular choice for serving MP4 and Flash video on demand (VOD) content using file download, progressive download, and pseudo-streaming, as well as live streaming using Real-Time Message Protocol (RTMP), Apple HTTP Live Streaming (HLS), and Dynamic Adaptive Streaming over HTTP (DASH). There is no better example for this than Netflix, the world's most popular video streaming service, delivering content to more than 50 million subscribers using NGINX as a critical part of their architecture.

NGINX Plus builds on the open source NGINX software and adds enterprise-ready features, including additional streaming media features, such as adaptive bit-rate streaming for VOD (using HLS and Adobe HTTP Dynamic Streaming [HDS]), bandwidth control for MP4 streaming, and enhanced session logging. As a complete application delivery platform, NGINX Plus offers enhanced web serving and load-balancing features like application health checks, session persistence, and enhanced monitoring statistics. NGINX Plus customers also get true enterprise-grade support that provides direct access to the core NGINX engineering team.

Video Streaming Overview

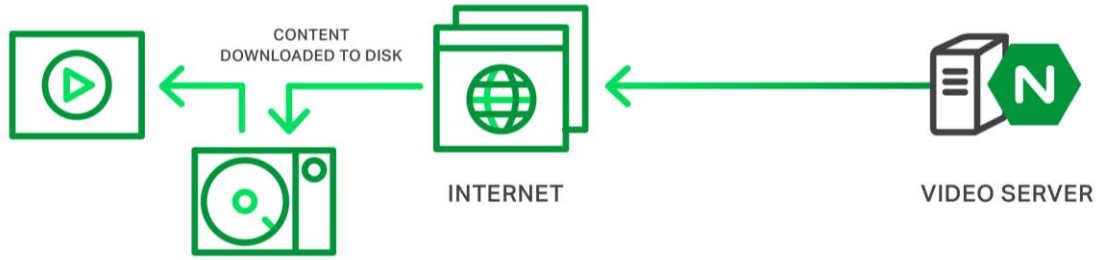
Video streaming can be broken into two major categories: video on demand (VOD) and live streaming. Within each category, NGINX Plus supports multiple methods for serving content. This section discusses the methods at a high level and the following section, **NGINX Plus for Streaming Media – Technical Detail**, explains in detail how to configure NGINX Plus to support each method.

Video on Demand

VOD content is pre-recorded and stored on a server from which a player on a client device can retrieve it. There are several ways to support VOD with NGINX Plus.

File Download

With this method, the complete video file is downloaded via HTTP and saved to the client device. Playback does not begin until the entire file is downloaded, which often results in delays because video files are large. If there are network issues, the download might never complete.



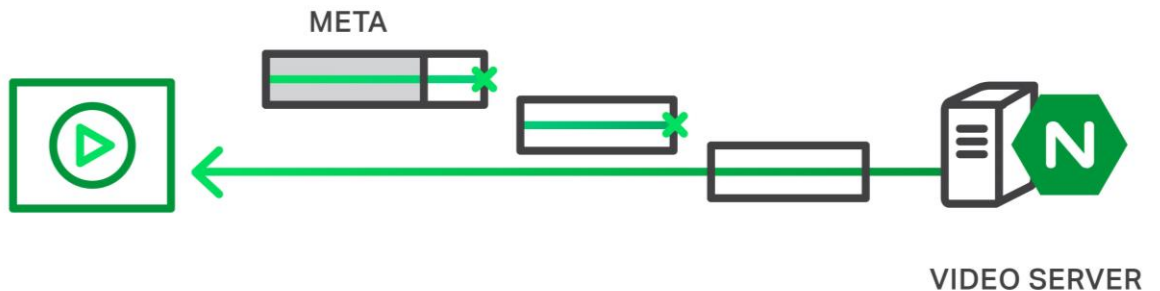
Progressive Download

Progressive download improves on the file download method by reducing the delay before playback begins. The video is still downloaded as a single file, but the metadata (information that tells the video player how to play the video) is at the front of the file, so the video player can begin playback as soon as it buffers the metadata and a small amount of video data. The player continues to download and save the rest of the file via HTTP as it plays what is already downloaded. Progressive download does not fully support “trick play” features such as fast forwarding or jumping to a later part of the video, unless the corresponding part of the file has already been downloaded.



Pseudo-Streaming

HTTP pseudo-streaming is an advance on progressive download because the player can seek forward or backward in a video. The player calculates the desired part of the video in terms of a byte offset or number of seconds from the start the video, and can buffer the data without necessarily having to write it to disk. It is no longer necessary to download the file from start to finish; the player can now stop the stream of data and move to a different point.



Pseudo-streaming is supported by two modules in NGINX Plus: the MP4 module for MP4 files and the FLV module for Flash files.

Adaptive Bit-Rate Streaming

HTTP adaptive bit-rate streaming is currently the most sophisticated method for streaming media delivery. It uses a completely different paradigm from the progressive download and pseudo-streaming methods discussed previously, repackaging the media file into a series of smaller segments, each with just a few seconds of video, which makes it very easy for the player to skip forward or backward in the video. A playlist or manifest file is created to act as a table of contents for the segments.



Another difference is that the player does not typically store video data locally. Media providers can make available several versions of the same content, each encoded with a different bit rate to provide a different level of sound and picture quality. In this case, there is also a master manifest or playlist pointing to the manifest or playlist for each bit rate. The video player is able to switch seamlessly between the different versions at any time during playback, so the viewer can select the desired quality level and the player can adjust the quality automatically based on real-time network conditions. The player can more easily recover from network interruptions and a persistent connection to the server is not required, which is a particular benefit on mobile networks.

Two popular methods for adaptive bit-rate streaming over HTTP:

- HTTP Live Streaming (HLS) from Apple. A file encoded in H264+ACC format is re-packaged into a series of short segment files with a playlist (also known as the index file). The video player first requests the playlist and then the segment files.
- HTTP Dynamic Streaming (HDS) from Adobe. The content is re-packaged in three types of files: a manifest file, containing information about the video that the player needs; one or more fragment files, containing the segments; and an index file for each fragment file, containing information about the segments.

Real Time Messaging Protocol

The Real Time Messaging Protocol (RTMP) can be used for VOD, but is primarily used for live streaming and so is described in the next section.

Live Streaming

With live streaming, video content is distributed from the video source to clients in real time, without being recorded first. NGINX Plus supports live streaming using RTMP, which can also be converted to HLS and Dynamic Adaptive Streaming over HTTP (DASH).

RTMP

RTMP is a TCP-based streaming protocol from Adobe. RTMP is supported across various browsers by means of a Flash plugin.

HLS

HLS for live streaming uses the same playlist and segment files as described previously for HLS VOD, except that new segment files are continuously created and the playlist is marked as incomplete, which causes the player to keep requesting a new playlist until the stream has finished and the playlist is marked complete.

DASH

DASH, like HLS and HDS, uses the concept of segments and the equivalent of a playlist or manifest file, known as a Media Presentation Description (MPD) file. However, DASH treats the video stream as a single file (does not create segment files) , so the MPD file points to offsets in the origin file rather than to segment files.

NGINX Plus for Streaming Media

Technical Detail

This section explains how to configure NGINX Plus for each of the media streaming methods.

Video on Demand

To support video on demand (VOD), NGINX Plus uses its standard HTTP functionality, and the MP4, FLV, HLS, and F4F (HDS) modules. The NGINX RTMP module can also be used for VOD, but is more commonly applied to live streaming, so we discuss it in that section below.

File Download

The HTTP functionality in NGINX Plus is all that is required to enable the download of complete media files; no specialized modules are needed. The following example configuration serves MP4 files from the **/usr/share/nginx/html/video** directory:

```
server {
    listen 80;

    root /usr/share/nginx/html;

    location /video {
    }
}
```

Progressive Download

Like basic file download, progressive download of files with metadata at the beginning of the file doesn't require any specialized modules or configuration. The basic configuration shown previously for file download also enables progressive download.

Pseudo-Streaming

Configuration for pseudo-streaming depends on the media format:

- To serve FLV files, include the `flv` directive, which enables the core NGINX FLV module. For more information on the FLV module please see: http://nginx.org/en/docs/http/nginx_http_flv_module.html
- To serve MP4 files, include the `mp4` directive, which enables the core NGINX MP4 module. The module can even handle media files that do not have metadata at the beginning of the file; it reads the entire file to locate the metadata before playback begins.

You can also include the `mp4_limit_rate` directive to enable the bandwidth management feature for MP4 files, limiting the amount of bandwidth a client can use to a specified factor of the detected video bit rate. Clients with faster connections still experience smooth video streaming without using more bandwidth than they actually need, which could negatively impact clients with slower connections. The related `mp4_limit_rate_after` directive controls the delay before bandwidth limiting begins, so that clients can consume unlimited bandwidth as they initially buffer content.

For more information on the MP4 module please see: http://nginx.org/en/docs/http/nginx_http_mp4_module.html

The following example configuration configures pseudo-streaming of MP4 and FLV files from the **/usr/share/nginx/html/video** directory. For MP4 files, clients can use unlimited bandwidth for first 15 seconds of downloading and up to 1.2 times the bit-rate after that:

```

server {
    listen 80;

    root /usr/share/nginx/html;

    location ~ ^/video/.*\.flv$ {
        flv;
    }

    location ~ ^/video/.*\.mp4$ {
        mp4;
        mp4_limit_rate 1.2
        mp4_limit_rate_after 15s;
    }
}

```

Adaptive Bit-Rate Streaming

NGINX Plus supports adaptive bit-rate streaming using both HLS and HDS.

HLS

NGINX Plus handles HLS content differently depending on whether the video files are already segmented into playlist and segment files. If the files are already segmented, NGINX Plus serves them as normal HTTP requests. If the files are not segmented, the advanced NGINX Plus HLS module converts them to HLS dynamically. In this case, when a client requests an HLS playlist file (file extension **.mp4.m3u8**) or segment file (**.mp4.ts**), NGINX Plus locates the corresponding MP4 file and dynamically creates the necessary playlist or segment. The following example serves HLS content from the **/usr/share/nginx/html/hls** directory where the M4 files are located:

```

server {
    listen 80;

    root /usr/share/nginx/html;

    location /hls {
        hls;
    }
}

```

To support adaptive bit-rate streaming of different quality versions of the same content, you manually create a master playlist that points to the playlist for each bit-rate. The NGINX Plus HLS module then dynamically creates the playlist and segment files for each bit-rate.

For example, the following is an example master playlist for serving a video that has two quality versions:

```

#EXTM3U
#EXT-X-STREAM-INF:PROGRAM-ID=1,BANDWIDTH=545600,RESOLUTION=416x234,
CODECS="avc1.42e00a,mp4a.40.2"
/hls/myvideo_low.mp4.m3u8
#EXT-X-STREAM-INF:PROGRAM-ID=1,BANDWIDTH=1755600,RESOLUTION=640x360,
CODECS="avc1.42e00a,mp4a.40.2"
/hls/myvideo_high.mp4.m3u8

```

For more information on the HLS module please see: http://nginx.org/en/docs/http/nginx_http_hls_module.html

HDS

To serve HDS content, include the `f4f` directive to enable the F4F module. The content must be pre-processed with tools such as `ffmpeg`, and the manifest, index, and segment files made accessible to NGINX Plus. The following example serves HDS content from the **/usr/share/nginx/html/hds** directory where the manifest, index, and segment files are located:

```

server {
    listen 80;

    root /usr/share/nginx/html;

    location /hds {
        f4f;
    }
}

```

For more information on the F4F module please see:
http://nginx.org/en/docs/http/nginx_http_f4f_module.html

Live Streaming

The NGINX RTMP module supports live streaming.

RTMP

To configure standard RTMP support, include an rtmp block in the main context. NGINX accepts a live stream and forwards it to an RTMP capable player. The following example serves a simple live broadcast:

```

rtmp {
    server {
        listen 1935;
        application live {
            live on;
        }
    }
}

```

HLS

The NGINX RTMP module can translate an RTMP stream into HLS format. The following example pulls an RTMP stream from **tv2.example.com** and creates HLS files in the **/tmp/tv2** directory, from where NGINX uses HTTP to serve them with a URL of **http://NGINX-hostname/tv2/stream-name.m3u8**:

```

rtmp {
    server {
        listen 1935;
        application tv {
            live on;
            hls on;
            hls_path /tmp/tv2;
            hls_fragment 15s;
            pull rtmp://tv2.example.com:443/root/new name=tv2 static;
        }
    }
}

http {
    server {
        listen 80;
        location /tv2 {
            root /tmp;
        }
    }
}

```

DASH

The NGINX RTMP module can also translate an RTMP stream into DASH format. The following example pulls an RTMP stream from **tv2.example.com** and creates DASH files in the **/tmp/tv2** directory where

they are served using HTTP to the path **`http://NGINX-hostname/tv2/stream-name.mpd`**:

```
rtmp {
    server {
        listen 1935;
        application tv {
            live on;
            dash on;
            dash_path /tmp/tv2;
            dash_fragment 15s;
            pull rtmp://tv2.example.com:443/root/new name=tv2 static;
        }
    }
}

http {
    server {
        listen 80;
        location /tv2 {
            root /tmp;
        }
    }
}
```

For more information on the RTMP module please see: <https://github.com/arut/nginx-rtmp-module>

Summary

NGINX Plus provides a high performance, reliable and full-featured application delivery platform including a rich set of media serving functionality along with support provided by the NGINX team.