

KMM

LAN PRAKTIKOA

VIDEO ON DEMAND ETA VIDEO
STREAMING

2020/12/03

Aritz Ganuza
Jon Dorronsoro

AURKIBIDEA

1.Sarrera	3
2.Plangintza	4
Helburuak	4
Atazak	4
Ordu estimazioa	4
Pertsonak	5
Kostua	5
3. Landutako Soluzioa	6
Zerbitzaria	6
Video on Demand	8
Video Streaming	9
4. Arakatzailerako Bezeroak	12
HTML5	12
VideoJS	12
5. Adaptive Streaming	14
DASH	14
6. Ondorioak	15
7. Erreferentzia bibliografikoak	16
8. Eranskinak	17

1.Sarrera

Txosten honetan Komunikazio Mugikorrek eta Multimedia ikasgaiko lehen lan praktikoaren nondik norakoak azalduko dira. Lan praktiko honek bi helburu nagusi dauzka. Alde batetik, media-streaming zerbitzu bat abiatzea, zeinak aurre-grabatutako edukiak (bideoa) eta baita zuzeneko emanaldiak eskaini beharko dituen. Bestetik, txostenak idazteko eta aurkezpenak egiteko gaitasunak lantzea.

Enuntziatuan esaten digute enpresa bat gurekin harremanetan jarri dela gure zerbitzuak kontratatzeko. Beraien komunikazio-estrategiaren parte bezala, webgune bat nahi dute, bertan aurrez grabatutako bideoak eta zuzeneko ekitaldiak zabaltzeko. Beraz, gure eginkizuna, webgune bat eta streaming zerbitzari bat martxan jartzea izango da. Streaming zerbitzariak aurrez grabatutako eta denbora errealeko edukiak zerbitzatu beharko ditu. Erabiltzaileak eduki hauek webgunearen bidez atzitu ahal izango ditu (arakatzaile bat erabiliz).

Txosten honetan, arestian aipatutako eginkizuna lortzeko garatu den soluzioa azalduko da, kode zatiak erakutsiz. Horrez gain, lana burutzeko beharrezko plangintza ere agertuko da, soluzio finala lortu eta gero ateratako ondorioekin batera.

2. Plangintza

Helburuak

1. Streaming zerbitzu bat abiaraztea. Zerbitzu honek bai Video on Demand (Vod) bai denbora errealeko streaming-a gauzatu ahalko ditu.
2. Multimedia plataformen funtzionamenduan sakondu.
3. Txosten teknikoak zein aurkezpenak gauzatzeko gaitasunak hobetu.

Atazak

- Web orria zerbitzatu
- VoD inplementatu (bi bezero ezberdinetan)
- Live Streaming-a inplementatu

Ordu estimazioa

Ondorengo taulan (1. Taula) proiektua hasi aurretik garaturiko denboraren aurreikuspena eta azkenean jasotako amaierak denborak ikus daitezke.

	AURREIKUSPENA	AMAIERA DENBORAK
INFORMATU	5	3
WEB ORRIA	4	2
VoD	6	4
LIVE STREAMING	8	11
TXOSTENA	10	7
AURKEZPENA	5	3
GUZTIRA	38	30

1. **Taula:** Ordu estimazioa eta azken emaitzak.

Pertsonak

Lanaren banaketa txostenean eta aurkezpenean bakarrik eman da 2. Taulan ikus daitekeen moduan, beste atazak batera egin ditugu.

	JON	ARITZ
SARRERA		✓//✓
PLANGINTZA	✓//✓	
LANDUTAKO SOLUZIOA		✓//✓
ARAKATZAILEKO BEZEROA	✓//✓	
ADAPTIVE STREAMING	✓//✓	
ONDORIOAK		✓//✓
ERREFERENTZIAK	✓	✓
ERANSKINAK	✓	✓
DEMOA	✓	✓

2.Taula: lanaren banaketa txostena//aurkezpena formatuan

Kostua

Erabili ditugun software guztiak doakoak, eta ahal zen neurrian libreak, izan dira, beraz kostua 0 €-koa izan da 3. Taulan ikus daitekeen moduan. Lan-indar moduan izandako kostua 30hX7,43€=222,9€ko kostua suposatuko luke gutxieneko soldata suposatuz.

SOFTWAREA	PREZIOA	SOFTWARE MOTA
OBS	0	LIBREA
NGINX	0	LIBREA
APACHE	0	KODE IREKIKOA
HTML5	0	(LENGOAI DA)
VIDEO-JS	0	LIBREA

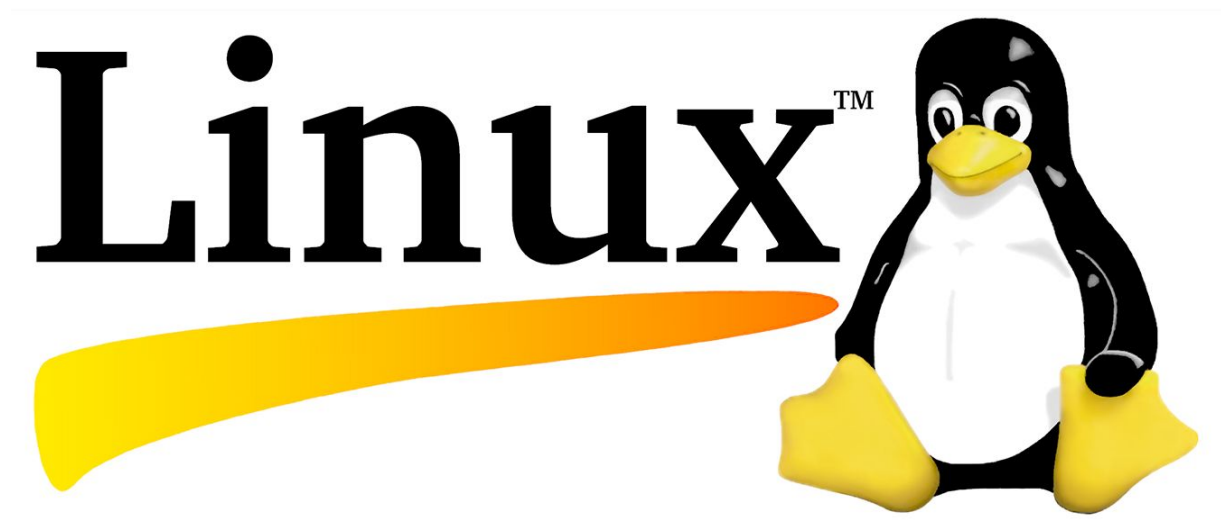
3.Taula: Erabilitako softwareak eta prezioa

3. Landutako Soluzioa

Txostenaren atal honetan proposatutako soluzioa garatzeko jarraitu ditugun pausuak eta erabilitako softwarea azalduko da, 3 azpi ateltan banatuz, lehenik zerbitzaria, bigarrenik Video on Demand eta bukatzeko Streaming zerbitzua.

Zerbitzaria

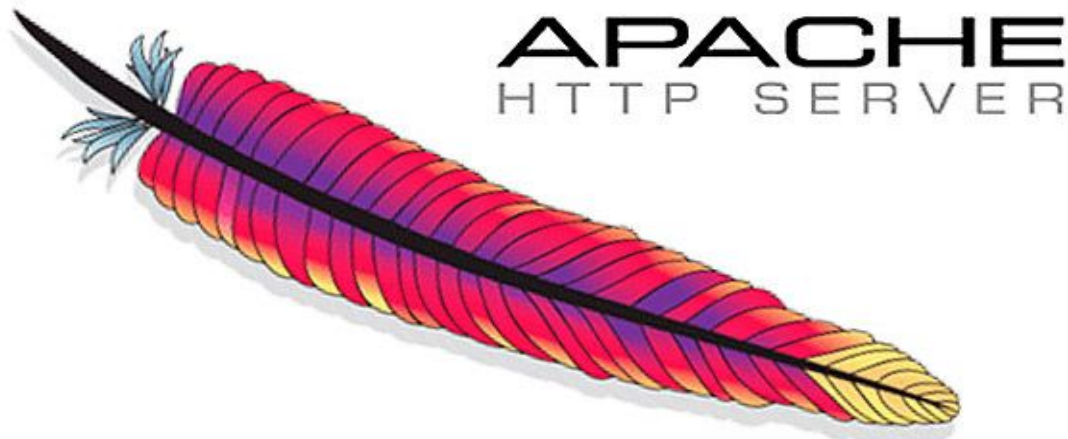
Hasteko, landutako soluzioa garatzeko Linux makina bat erabili dugula azpimarratzekoa da, batik bat, Sistema Eragile honek eskaintzen duen erraztasuna eta malgutasuna zerbitzariak martxan jarri eta hauek kudeatzeko orduan.



3.1 Irudia: Linux Sistema Eragilearen logoa (Tux)

Guk proposatutako soluzioa garatzeko lehen pausua zerbitzari bat martxan jartzea izan zen, gure aurrez grabatutako bideoak eta bideo streaming-ak gordetzeko euskarri bezala erabiliko genuena, eta web orri baten bitartez atzigarri egongo zena. Honetarako Apache erabili genuen, html web orri bat zerbitzatzeko nahiko delako, eta era berean instalatu eta abian jartzeko oso erraza delako, batez ere Linux makina batean.

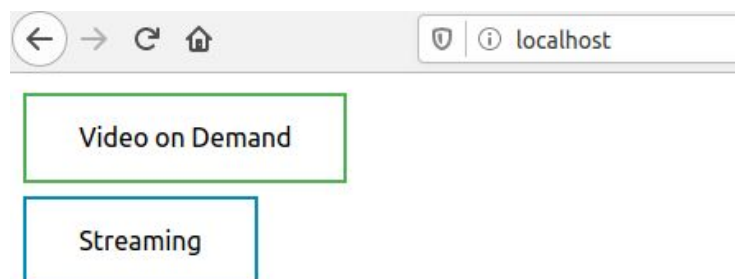
Apache HTTP zerbitzaria, edo soilik Apache, web zerbitzari mota bat da, eta HTTP/1.1 protokoloa inplementatzen duten plataformetan erabili dezakegu. RFC 2616 araudia jarraitzen du. Multiplataforma software bat denez, Unix zerbitzarietan, Macintosh eta Windows-en funtzionatzen du. Apache web zerbitzaria deitzen dugun arren, ez da zerbitzari fisiko bat. Zerbitzari batean exekutatzen den softwarea baizik. Gure lana zerbitzari baten eta webguneetako bezero arakatzailen (Firefox, Google Chrome, Safari etab.) konexioa ezartzea da, eta haien artean fitxategiak bidaliko dituzte bitartean (bezero-zerbitzariaren egitura erabiliz).



3.2 Irudia: Apache HTTP Server-en logoa

Kokapena ezberdina izan daiteke, `/var/www` edo `/srv/www` edo baliteke beste banaketetan batean beste leku batean egotea. Gure kasuan `/var/www` barruan html izeneko karpeta egoten da, eta hor kokatu behar dira webguneoko fitxategiak, bai kodea eta bai bideo eta argazkiak.

Behin hau lortuta, gure webgunearen diseinua egitea bakarrik ez zaigu falta. Kasu honetan, diseinu simple eta intuitibo bat aukeratu dugu, txostenean eskatutako funtzionalitateak eskaintzera mugatuz, hau da, VoD-etarako botoi bat eta Streaming-erako beste bat aurkeztuz.



3.3 Irudia: Web orriko hasiera orria

Video on Demand

Apache-ren bitartez gure web orria martxan jarrita daukagula, VoD funtzionalitatea gehitzeko prest gaude. Horretarako html orri berri bat osatu genuen Video on Demand izeneko botoi berda sakatzean irekiko duguna. Hemen gure zerbitzarian gordetako hainbat bideo ikusteko gaitasuna izango dugu, bakoitza arakatzaille bezero desberdin batekin eskaintzen dena. Hala ere ez dugu arakatzaillea hauen inguruan sakonduko, txosteneko hurrengo puntuan sakonago azalduko dira eta.

VideoJS



HTML5

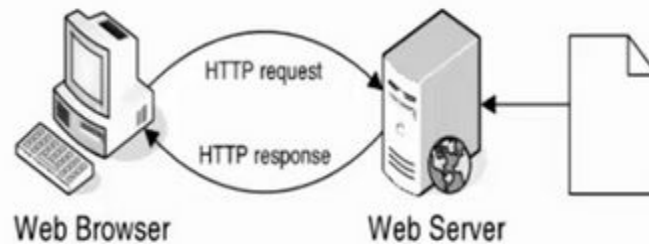


DASH



3.4 Irudia: VoD orria

Soluzio honetan VIDEOJS, HTML5, eta DASH-en bitartez zerbitzatuko ditugu bideoak. Atal honi dagokion arkitektura oso sinplea da, eta edozein web aplikaziok erabiltzen duen berdina da. Web arakatzaileraren bitartez Apache zerbitzariari HTTP request bat bidaltzen dio, eta honek bere biltegiatzean (kasu honetan gure ordenagailuko /var/www/html karpetan) gordetako bideoak zerbitzatzeko dituzten arakatzailerako bezero desberdinak erabiliz.



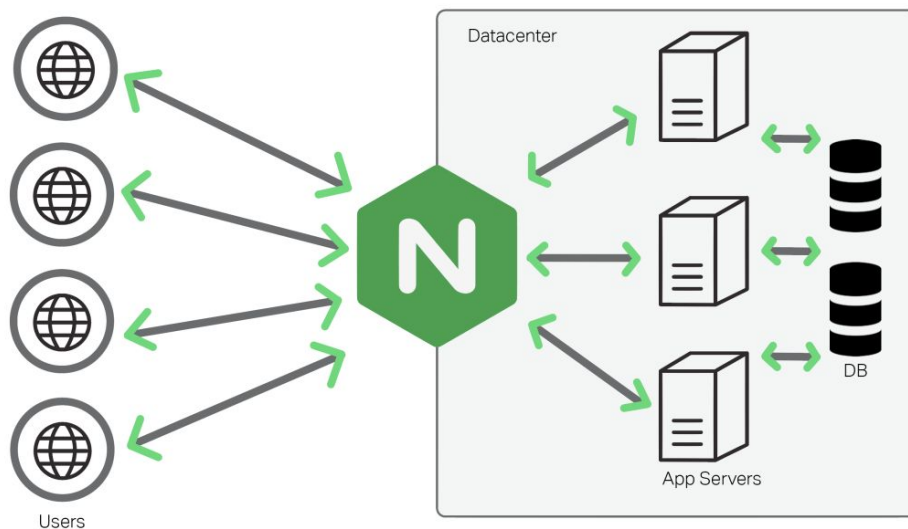
3.5 Irudia: Apache arkitektura

Video Streaming

Live streaming-a egiteko Open Broadcaster Software (OBS) erabili dugu. OBS Studio audio eta bideo grabazioak eta edizioak egiteko balio duen softwarea da, bideo-tutorialak prestatzeko erabil daiteke, (streaming) egiteko, bideo-jokoetan lortutako balentriak grabatzeko, etab. Software librea da eta Microsoft Windows, GNU/Linux eta macOS-ekin erabil daiteke.

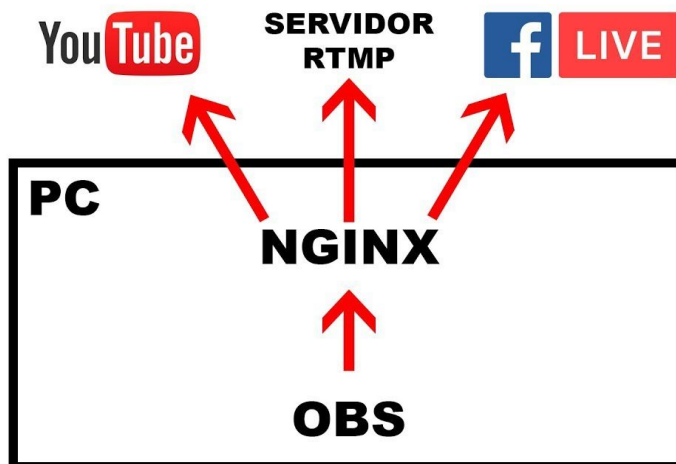
OBS-k live streaming fluxua RTMP (Real Time Messaging Protocol) protokoloa erabiliz kanporatzen du. Protokolo hau TCP-ren gainean lan egiten du, baina gure Apache web orrian HTTP bidez bideo streaming bat abiatzeko, protokolo desberdin bat erabili behar dugu. Gure kasuan, HLS (HTTP Live Streaming) protokoloa erabiliko dugu. HLS moldaketa-transmisioko komunikazio-protokoloa da, Apple Inc-ek garatu eta 2009an sortua. Protokoloaren euskarria hedatuta dago baliabideen erreproduzitaileetan, web-nabigatzaileetan, gailu mugikorretan eta transmisio-baliabideen zerbitzarietan. 2019tik aurrera, bideoaren industriari buruzko urteroko inkesta batek aurkitu du streaming-formatu ezagunena dela. HLS MPEG-DASH sistemaren antzekoa da, non fluxu orokorra hausten duen HTTP oinarri duten fitxategien deskarga txikien segida batean. Eskuragarri dauden fluxuen zerrenda bat, bit-abiadura desberdinetan kodetua, bezeroari bidaltzen zaio M3U erreproduktzio-zerrenda zabaldua erabiliz.

Protokolo aldaketa hau burutu eta gure Apache zerbitzariari bidaltzeko, beste zerbitzari bat beharrezkoa zaigu eta lan honetarako Nginx aukeratu dugu. Nginx alderantzizko proxy gisa, karga-balancer gisa, postako proxy gisa eta HTTP cache gisa ere erabil daitekeen web zerbitzari bat da. Nginx software librea eta kode irekikoa da, eta BSD lizentzian 2 klausularen arabera argitaratu da. Web zerbitzarien frakzio handi batek Nginx erabiltzen du, sarritan karga-balancer gisa.

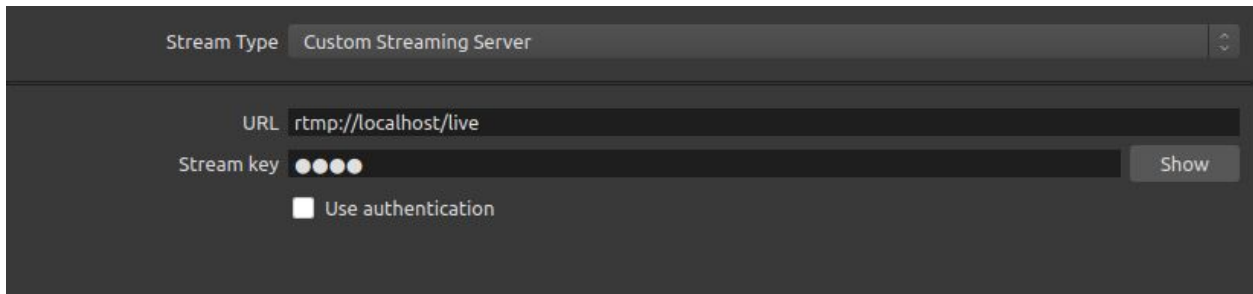


3.6 Irudia: Nginx-en arkitektura

Laburbilduz, honakoa da gure zerbitzarien arkitektura: OBS-k bideoa grabatu eta RTMP biartez gure Nginx zerbitzariari bidaltzen dio (ikus 3.8 irudia), eta honek, HLS protokoloan kanporatu eta Apache zerbitzariari bidaltzen dio (ikus 3.9 eta 3.10 irudia), gure web orrian atzigarri izateko (ikus 3.11 irudia).



3.7 Irudia: Live streaming-aren arkitektura



3.8 Irudia: OBS-k rtmp zerbitzariari (Nginx) streaming-a bidali

```
rtmp {
  server {
    listen localhost:1935;
    ping 30s;
    notify_method get;

    application live {
      live on;

      # sample HLS
      hls on;
      hls_path /var/www/html/hls;
      hls_sync 100ms;
      hls_fragment 3;
      hls_playlist length 60;
      allow play all;
    }
  }
}

types {
  application/vnd.apple.mpegurl m3u8;
  video/mp2t ts;
}

root /var/www/html/hls;
}
```

3.9 Irudia: rtmp hls protokolora bihurtu

3.10 Irudia: Apache zerbitzarira bidali streaming-a hls

```
<!DOCTYPE html>
<html lang="en">
<head>
  <link href="https://vjs.zencdn.net/7.1/video-js.min.css" rel="stylesheet">
  <script src="https://vjs.zencdn.net/7.1/video.min.js"></script>
</head>
<body>
  <video id="player" class="video-js vjs-default-skin" controls >
    <source src="http://localhost/hls/test.m3u8" type="application/x-mpegURL" />
  </video>

  <script>
    var options, video;

    options = {
      autoplay: true,
      muted: true
    };

    video = videojs('player', options);
```

bidez

3.11 Irudia: Apache zerbitzariak streaming-a atzigarri utzi videojs erabiliz.

4. Arakatzailerako Bezeroak

Gure proiektuan bi bezero inplementatu ditugu, HTML5 eta VideoJS. Aurregrabatutako bideoak ikusteko bi aukera eskaini arren biek bideo bera eskaintzen dute batetik besterako ezberdintasunak argiago uzte aldera.

HTML5

Online bideoak ikusteko sorturiko etiketa de HTML5-ek eskaintzen duen `<video>`. 4.1 irudian ikus daitekeen moduan oso erraza da erabiltzeko, nahikoa da bideoaren iturria eta bideo mota jartzearekin `<video>` etiketaren barruan. Bestetik bideo leihatilaren tamaina alda dezakegu bideoari hobekien datorkionera.

```
<h1>HTML5</h1>
<video width="640" height="480" controls>
  <source src="movie.mp4" type="video/mp4">
  <source src="movie.ogg" type="video/ogg">
  Your browser does not support the video tag.
</video>
```

4.1 Irudia: HTML5-eko bideoaren kodea

VideoJS

VideoJS HTML5-en gainean eraikitako bideo erreproduzitzaila weba da. HTML5-ez gain, egungo streaming plataforma gehienak, Youtube, Vimeo eta Flash jasaten ditu. Horrez gain kode irekiko proiektua da eta izena adierazten duen moduan JavaScriptez dago programatua.

VideoJS erabiltzeko lehenik eta behin liburutegiak inportatu behar ditu gure web orrira, 4.2 irudian agertzen den modura. Bi modutan egin dezakegu hau, lehenik txostenean agertzen den eran, link bitartez, edo bestela deskargatu eta gure zerbitzaritik exekutatu dezakegu.

```
<head>
  <script type="module" src="https://unpkg.com/dash-video-element"></script>
  <link href="https://vjs.zencdn.net/7.8.4/video-js.css" rel="stylesheet" />
  <!-- If you'd like to support IE8 (for Video.js versions prior to v7) -->
  <script src="https://vjs.zencdn.net/ie8/1.1.2/videojs-ie8.min.js"></script>
</head>
```

4.2 Irudia: VideoJS-eko liburutegiaren inportazioa

Erabiltzeko modua HTML5-en antzekoa da, `<video>` etiketan informazio gehiago jartzeko aukera eskaintzen du, ezberdintasun nagusia `class` atala izanik, HTML5-ean jartzea ez zen beharrezkoa, hemen `video-js` dela adierazi beharko zaio.

```
<h1>VideoJS</h1>
<video
  id="my-video"
  class="video-js"
  controls
  preload="auto"
  width="640"
  height="480"
  poster="poster.jpg"
  data-setup="{}"
>
  <source src="movie.mp4" type="video/mp4" />
  <source src="movie.webm" type="video/webm" />
  <p class="vjs-no-js">
    To view this video please enable JavaScript, and consider upgrading to a
    web browser that
  <a href="https://videojs.com/html5-video-support/" target="blank"
```

4.3 Irudia: VideoJS-eko bideoaren kodea

Funtsean biak aztertu ostean argi ikus daiteke VideoJS aukera hobea dela, bai plataforma anitzekin duen elkargarritasunagatik bai Adaptive streaming-ari eskaintzen dizkion erreztasunengatik.

5. Adaptive Streaming

Ahalik eta modu efizienteenean internetetik datuak transmititzeko teknika da. Teknika honen funtsa erabiltzaileen gaitasunen arabera bidalketa egitean datza, jakinik gaitasun hauek aldagarriak direla edozein unetan.

Prozesu hau aurrera eraman ahal izateko bideo bat, adibidez, kalitate bakar batean gordeta izan behar hainbat ezberdinetan (480p, 720p, 1080p...) gordetzen da. Honela erabiltzailearen banda-zabalera baxua bada kalitate kaxkarragoa duen bideo bat bidaliko zaio.

Aurreko teknikarekin, Progressive Streaming, eskaintzen duen abantaila nagusia kalitate gama da. Progressive Streaming-ean bideoa kalitate bakarrean gordetzen zen eta banda-zabalera baxuegia baldin bazen buffering-ak pairatu behar ziren.

DASH

DASH abiarazteko hainbat aldagaketa burutu behar dira kodearen gainean. Argiena etiketa bera da, kasu honetan <dash-video> izatera pasatuz. Bestetik bideoaren formatua ere aldatu beharra dago. Azkenik 5.1 argazkian ikusten den script-a behar da erreprodukzioa gauzatu dadin.

```
<script src="/home/user/Downloads/cdnjs/dash.all.min.js">
function init() {
  var video,
      player,
      url = "https://dash.akamaized.net/akamai/bbb_30fps/bbb_30fps.mpd";

  video = document.querySelector("video");
  player = dashjs.MediaPlayer().create();
  player.initialize(video, url, false);
}
</script>
```

5.1 Irudia: DASH martxan jartzeko kodea

6. Ondorioak

Nahiz eta egindako plangintza aurreikuspena eta bukaeran pasatako ordu kopurua desberdina izan, hau da, espero baino ordu gutxiago sartu ditugun, esan nahiko genuke ez zaigula erraza iruditu. Batez ere informazio bilketa fasean, ezjakintasun nabari batekin egin genuen topo, eta ondorioz gure beharratara moldatuko zen softwarea aurkitzea eta hau ulertzea kostatzen zaitzaigun.

Esan beharra dago, VoD atala burutzeko ez genituela ez zailtasun ez duda handirik izan, baina Streaming atalera iristean, arkitektura eta protokolo desberdinak ulertu eta hauek kudeatzeko oztopo nabari izan genituen. Fase honetan, azpimarragarria da Nginx-en bitartez protokolo aldaketa gauzatzeko pasa genuen denbora, honen konplexutasuna dela medio, eta interneten dagoen informazio faltagatik ere.

Azkenean, modu simple batean bitartez diseinatu dugu gure soluzioa, eskakizun minimoak era intuitibo eta efiziente batean garatu ahal izateko. Zentzu horretan software eta zerbitzari desberdinen arteko erlazioa eta arkitekturari eman diogu garrantzia, web orriaren diseinua bigarren plano batean utziz, garrantzi gutxiago zeukala ulertzen baitugu. Hala ere, proposatutako soluzioa modu txukun eta argi batean antolatu dugu.

Bukatzeko, esan dezakegu media-streaming zerbitzu bat martxan jartzeko erabiltzen diren teknologia eta software erabilienean inguruan (Apache, Nginx, HTML5, VideoJs, DASH, protokolo ezberdinak...) asko ikasteko oso baliagarria izan da lehen lan praktikoa hau, batez ere, hauen jakintza teorikoa praktikan jartzean bakarrik barneratzen delako uste dugulako.

7. Erreferentzia bibliografikoak

<https://www.w3schools.com/>

<https://www.apache.org/>

<https://videojs.com/>

<https://github.com/Dash-Industry-Forum/dash-video-element>

https://developer.mozilla.org/es/docs/Web/HTML/Transision_adaptativa_DASH

<https://docs.microsoft.com/es-es/azure/media-services/previous/media-services-embed-mpeg-dash-in-html5>

<https://videojs.github.io/videojs-contrib-dash/>

<https://reference.dashif.org/dash.js/latest/samples/index.html>

<https://ffmpeg.org/>

<https://ffmpeg.org/ffmpeg.html>

<https://www.how2shout.com/tools/free-best-open-source-video-streaming-servers-software.html>

<https://support.google.com/youtube/answer/3068031?hl=en>

<https://www.nobile-engineering.com/wordpress/index.php/2018/10/30/video-streaming-hls-apache-nginx/>

8. Eranskinak

Hemengo link-ean eskuragarri dago web orria eta readme.txt fitxategia, zeinetan komandoak dauden:

<https://cryptpad.fr/drive/#/2/drive/edit/BkeA8V2gzkLDmb2l+47eXAer/>