

**KLUSTERIPALVELINJÄRJESTELMÄN KÄYTTÖÖNOTTO JA  
KONFIGUROINTI SEKÄ STREAMAUSPALVELUN ASENNUS**



Ammattikorkeakoulututkinnon opinnäytetyö  
Tieto- ja viestintäteknikka, insinööri (AMK), Riihimäen kampus

Syksy 2019

Taavi Köntti

---

Tekijä	Taavi Köntti	Vuosi 2019
Työn nimi	Klusteripalvelinjärjestelmän käyttöönotto ja konfigurointi sekä streamauspalvelun asennus	
Ohjaajat	Jari Mustajärvi	

---

## TIIVISTELMÄ

Opinnäytetyön tarkoituksena oli kehittää Keski-Uudenmaan koulutuskuntayhtymä Keudan Nurmijärven toimipisteelle, opiskelijoiden opintoihin tarkoitettu mediapalvelinkokonaisuus.

Palvelinkokonaisuus piti rakentaa entuudestaan tuntemattomista komponenteista, joita koulu oli saanut lahjoituksena, ja niille etsittiin käyttötarkoitusta.

Mediapalvelimen keskeisimmät tavoitteet olivat tarjota Mediaopiskelijoiden opintoihin ja harjoituksiin liittyen videoiden streamauspalvelu, eli live-lähetyksien harjoittelu koulun sisäisesti. Toisena tavoitteena oli videoiden jako VOD-tyyppisenä suoratoistona opiskelijoiden kesken.

Tilaustyön kustannukset haluttiin pitää lähellä nollaa, joten työssä sovelletut tekniikat ja ohjelmistot olivat joko täysin ilmaisia, tai niistä oli lisenssit tilaajalla jo valmiiksi.

Palvelinkokonaisuus haluttiin myös rakennettavan Windows-ympäristöön, sillä sen ylläpito tulisi jäämään ICT-opiskelijoiden vastuulle, ja Windows on Keudan ICT-opintojen keskeisin toimintaympäristö.

Tämä opinnäytetyön kirjallinen osuus keskittyy palvelinkokonaisuuden kehitykseen ja käytettyjen tekniikoiden sovellukseen työssä. Työn kaikki tavoitteet saavutettiin, mutta kompromissien vuoksi muutamalla heikkoudella

Avainsanat Palvelin, stream, klusteri

Sivut 79 sivua ja liitteitä 30 sivua

---

Author	Taavi Köntti	Year 2019
Subject	Deployment and configuration of a cluster server and development of a streaming service.	
Supervisors	Jari Mustajärvi	

---

ABSTRACT

The objective of this thesis project was to develop a media server entity for the students at Keuda Vocational School, at the campus located in Nurmijärvi.

The server was to be created out of unknown components, which had been donated to the school, and were looking for a useful purpose.

The essential objectives of the server entity were to provide two kinds of services for media program students to be used in their studies and exercises. The first one was to make live streaming possible, and the second one of the tasks was to make video sharing possible via VOD-type streaming, both within the schools internal network for training purposes.

It was required that the expenses were to be kept close to nothing, and therefore the selected methods and programs were either free, or the school had licenses to them beforehand. The server entity was to be developed within the Windows environment, because server maintenance and administration would be left as the responsibility of the ICT students, and their main working environment was Windows.

The written part of the thesis project focuses on the stages in the development process of this media server, and it examines how the selected methods and programs were applied into practice. The objectives of the project were achieved, but with a few weaknesses.

Keywords Server, stream, clustering

Pages 79 pages and appendices 30 pages

## Sisällys

1	Johdanto .....	1
2	Laitteisto ja testaus .....	2
2.1	HPE BladeSystem c7000 enclosure .....	2
2.2	HP ProLiant BL460 Gen7 Server Blade .....	4
2.3	HP ProLiant BL460 Gen6 Server Blade .....	8
2.4	HP ProLiant BL460c Gen6 Server Blade .....	8
2.5	SAS-kiintolevyt .....	9
2.6	Kytkimet .....	10
2.7	Käytettävissä oleva laitteisto .....	11
3	Palvelinjärjestelmän suunnittelu .....	12
3.1	Vaatimukset .....	12
3.1.1	Ohjelmistoympäristö .....	12
3.1.2	Streamipalvelin .....	12
3.1.3	Videon katselualusta .....	12
3.1.4	Videon suoratoistopalvelu .....	12
3.1.5	Aktiivihakemisto .....	13
3.2	Suunnittelu .....	13
3.2.1	Palvelut .....	14
3.2.2	Palvelimet ja roolit .....	15
3.3	Suunnitelman yhteenveto .....	15
4	Palvelimien asennus ja konfigurointi .....	17
4.1	Käyttöjärjestelmän asennus iLO moduulilla .....	17
4.1.1	iLO manager .....	18
4.1.2	Etäyhteyden muodostaminen palvelimeen .....	20
4.1.3	Windows Server 2016 asennus .....	22
4.2	Aktiivihakemisto .....	25
4.2.1	ADDS alustus .....	25
4.2.2	ADDS Roolin asennus .....	26
4.2.3	Domainin muodostus ja mainostus .....	28
4.3	Klusterointi .....	32
4.3.1	ISCSI muodostus .....	34

4.3.2	Klusterin muodostus .....	36
4.4	IIS.....	40
5	Palvelujen asennus ja Kehitys.....	42
5.1	Sovelletut tekniikat websivustolla .....	42
5.1.1	HTML5 ja CSS3.....	42
5.1.2	JavaScript.....	43
5.1.3	PHP .....	44
5.1.4	Plyr.....	46
5.1.5	FFmpeg.....	48
5.2	Streamimoottori.....	49
5.2.1	Mistserver .....	50
5.2.2	Mistserverin asennus.....	51
5.2.3	Toiminta Windowsissa .....	54
5.3	Websivut .....	55
5.3.1	Käyttöliittymä.....	56
5.3.2	VOD ja stream sivut.....	57
5.3.3	Streamin aloitus .....	60
5.3.4	Videon lähetys.....	64
5.3.5	Videoiden arkistointi ja poisto .....	69
6	Lopputulos JA Parannettavaa .....	73
6.1	Palvelinkoneet ja SAN .....	73
6.2	Streamimoottori.....	74
6.3	Videon lataus palvelimeen.....	74
	Lähteet.....	76

## Kuvat

Kuva 1	HP BladeSystem c7000 edestä ja takaa (Sanghera, 2016).....	3
Kuva 2	c7000 interconnect bay -paneelit.....	4
Kuva 3	BL460 Server Blade -etupaneeli (helpmax, n/a).....	5
Kuva 4	SUV-adapteri.....	7
Kuva 5	HP Dual port SAS Hot-swap kiintolevy.....	9
Kuva 6	Cisco Catalyst 3750G-24TS (CNET, 2010).....	11
Kuva 7	Palvelinjärjestelmän suunnitelma .....	16

Kuva 8 Server Blade BIOS ja iLO manager .....	19
Kuva 9 iLO käyttöliittymän kirjautumisruutu .....	20
Kuva 10 iLO3 webkäyttöliittymä.....	21
Kuva 11 .NET Frameworkin lisäys .....	22
Kuva 12 Etäyhteysportaali iLO:n ylitse .....	23
Kuva 13 Onnistunut imagen lataus .....	24
Kuva 14 Add roles and features -näkyvä.....	27
Kuva 15 ADDS roolin valinta .....	28
Kuva 16 Muistutus domainin mainostuksesta ja aloituslinkki. ....	29
Kuva 17 Uuden domainin muodostus .....	30
Kuva 18 Failover clusterin kytkentäperiaate .....	33
Kuva 19 iSCSI Palvelinrooli.....	35
Kuva 20 iSCSI Levyn luominen .....	35
Kuva 21 Failover Clustering - featuren lisäys.....	37
Kuva 22 Nodejen lisäys klusterin muodostuksessa (Kameshwaran, 2017).....	39
Kuva 23 IIS roolin asennus.....	40
Kuva 24 PHP yleismuuttujan lisäys (Loges, 2017) .....	45
Kuva 25 PHP Module mapping .....	45
Kuva 26 Plyr - HTML5-videotoistin .....	47
Kuva 27 MistServer (DDVTech) .....	50
Kuva 28 Streamitekniikkojen latenssit (Ruether, 2020) .....	51
Kuva 29 MistController .....	52
Kuva 30 Mistserver manager.....	53
Kuva 31 Windowsin Suorita / Run työkalu .....	54
Kuva 32 Registry key – muokkaus (expert-advice, 2017).....	55
Kuva 33 Websivuston etusivu.....	56
Kuva 34 Videot - sivu .....	58
Kuva 35 Välimuistin poisto-ohjeet .....	59
Kuva 36 Lähetykset - sivu alias streamisivu.....	60
Kuva 37 Streamin aloitus 1 .....	61
Kuva 38 Streamin aloitus 2 - valmiustila .....	62
Kuva 39 Aloita stream 3 - lähetys käynnissä .....	63

Kuva 40 Streamin aloitusohjeet - sivu .....	64
Kuva 41 Uuden videon lähetys .....	65
Kuva 42 Uuden videon lähetys aloitettu onnistuneesti .....	66
Kuva 43 Videon lähetys onnistunut.....	67
Kuva 44 Soittimen virheilmoitus .....	68
Kuva 45 Videon poisto - tunnistautuminen.....	70
Kuva 46 Ladattujen videoiden alkuperäiskappaleiden poistoskripti.....	71
Kuva 47 Arkisto .....	72

## **Liitteet**

Liite 1	Videon lataus palvelimeen, sekä videon enkooderin käynnistys ja HTML-sivun generoiminen – PHP-koodi
Liite 2	Videon lataus - HTML
Liite 3	IP osoiteen tarkastus sekä uudelleenohjaus – PHP-koodi
Liite 4	Videolista - HTML
Liite 5	Generoitu HTML-sivu esimerkki ja Plyr instanssin rakennus
Liite 6	Videon poisto manuaalisesti – PHP-koodi
Liite 7	Streamin aloitus, julkaisu, sekä lopetus
Liite 8	Websivun tyyllittely - CSS

## Lyhenteet ja Käsitteet

<b>.NET framework</b>	Microsoftin luoma monenlaisien sovellusten kehitysalusta
<b>ADDS</b>	Active Directory Domain Services, Microsoftin palvelinkäyttöjärjestelmien hallitun alueverkon tietokanta
<b>Batch</b>	Windows käyttöjärjestelmien skriptikieli
<b>BIOS</b>	Basic input/output system, tietokoneen emolevyyn sijoitettu ohjelma, joka vastaa mm. käyttöjärjestelmän lataamisesta.
<b>bitrate</b>	Videon toistossa käytetty termi, joka käsittää liikkuvan datan määrää videon toistossa
<b>Cache</b>	Välimuisti. Tietokoneen prosessorin yhteydessä nopeimman tason käyttömuisti. Webselaimen yhteydessä selaimen paikallisesti tallennettavat websivuston tiedostot.
<b>CSV</b>	Cluster shared volumes, tekniikka klusterin nodejen jaetun tallennustilan luomiseen
<b>DB9</b>	Tiedonsiirron standardi, yhdeksänpinninen sarjaliitin
<b>DC</b>	Domain Controller, Toimialueen hallintakone, joka sisältää aktiivihakemiston tietokannan
<b>DDR3</b>	kolmannen sukupolven DDR muisti, tietokoneen RAM

<b>DHCP</b>	Dynamic Host Configuration Protocol. IP-osoitteiden jakoon kehitetty palvelu.
<b>DNS</b>	Domain Name System. Nimenselvennyspalvelu, joka muuntaa isäntänimet IP osoitteiksi
<b>Domain</b>	Hallittu jaettujen resurssien toimialue lähiverkossa
<b>DSRM</b>	Directory services restore mode. ADDS:n virheenkorjaustyökalu
<b>Enkoodaus</b>	Videon pakkaukseen eri tekniikoilla yleistermi
<b>Failover cluster</b>	Useasta palvelimesta muodostettu kokonaisuus, joka toimii kuin yksi palvelin jakaen työtaakkaansa ja parantaen virheensietoa.
<b>Firmware</b>	Laitteen omalla käskykannalla toimiva ohjelmisto, joka huolehtii laitteen perustoiminnasta
<b>FQDN</b>	Fully qualified domain name. Palvelimen kokopitkä nimi domainissa, joka muodostuu kaikista domainin tasoista ja palvelimen isäntänimestä
<b>FTP</b>	File Transfer Protocol, Tiedonsiirtoon verkon yli kehitetty protocol
<b>GB</b>	Gigabyte, gigatavu. Tidostokoon mittari
<b>GPO</b>	Group policy object, aktiivihakemiston käyttäjien ja tietokoneiden hallintaa suorittava käskykokonaisuus

<b>GPU</b>	Graphics processor unit, tietokoneen komponentti tai sen osa, joka vastaa datan muuntamisesta grafiikaksi.
<b>HDD</b>	Hard disk drive. Tietokoneen mekaaninen kiintolevy
<b>HP</b>	Hewlett Packard, tietotekniikan laitevalmistaja
<b>Hot-swap</b>	Ominaisuus, jolla voidaan komponentteja liittää tai poistaa, kun laitteessa on virta päällä
<b>iframe</b>	HTML elementti, mikä mahdollistaa muiden sivujen katselun yhden HTML tiedoston sisällä
<b>IIS</b>	Internet Information Services. Microsoftin Web palvelimen luontiin käytettävä palvelu
<b>iLO</b>	Integrated Lights-out module. HP:n kehittämä tietokoneen sisäinen hallinnointiyksikkö, joka toimii herätevirralla
<b>Interconnect bay</b>	HP:n rakkikotelon laitteiden verkkokorttien keskitetty ohjainpaneeli
<b>iSCSI</b>	Internet SCSI
<b>Kellotaajuus</b>	Termi, jolla viitataan komponenttien datankäsittelynopeuteen
<b>Koodekki</b>	Videon pakkaustekniikan yleistermi

<b>LOM</b>	Lights out module, yleistermi kaikkien laitevalmistajien vastaaville moduuleille. katso iLO
<b>Mezzanine card</b>	Tietokoneen muihin ohjainkortteihin kytkettävä lisäohjainkortti
<b>MIME</b>	Multipurpose Internet Mail Extensions. Standardi tiedoston formaatin määrittämiseen
<b>NAS</b>	Network Attached Storage. Verkon yli käytettävä tallennusmedia
<b>NetBIOS</b>	Network Basic input/output system. Vanhentunut keino toimialueelle tunnistautumiseen.
<b>NLB</b>	Network Load Balancer. Useamman palvelimen ryhmä, joka jakaa verkkokäyttäjät keskenään.
<b>RAM</b>	Random Access Memory. Laitteen keskusmuisti, käyttömuisti, ohjelmien työmuisti
<b>RDIMM</b>	Registered dual in-line memory module. Tietokoneiden muistikomponentti, joka antaa vakautta ja skaalautuvuutta tekniikkansa ansiosta
<b>RDM</b>	Remote Device Management. Protokolla, joka mahdollistaa kaksisuuntaisen etäyhteyden
<b>RPM</b>	Rounds per minute. tietotekniikassa viitaa mekaanisten komponenttien pyörimisnopeuteen

<b>RTMP</b>	Real time message protocol. Protokolla, jota käytetään pääsääntöisesti jatkuvan videolähteen siirtämiseen verkon yli
<b>SAN</b>	Storage Area Network. Dedikoitu lähiverkko verkkotallennusmedialle
<b>SAS</b>	Serial attached SCSI. Sarjakytkennäinen SCSI
<b>SCSI</b>	Small Computer System Interface on standardien joukko tiedon välittämiseksi tietokoneen ja oheislaitteiden välillä
<b>SPOF</b>	Single point of failure. Järjestelmäkokonaisuuden toimivuuden heikko kohta.
<b>Thread</b>	Tietokoneen prosessorin osio, jolla laskentaa suoritetaan, ns. looginen ydin
<b>USB</b>	Universal serial bus. Sarjaportti yleisien plug and paly laitteiden liittämiseen ja tiedonsiirtoon
<b>VGA</b>	Video Graphics Array. Tietokoneen grafiikkadatansiirron liitintyyppi
<b>VLAN</b>	Virtual LAN. Virtuaalinen lähiverkko. Voidaan muodostaa samaan lähiverkkoon eroteltuja osia.
<b>VOD</b>	Video on demand. Tilausvideo, verkosta ladattava video, jota siirretään verkon yli katselun yhteydessä

**Volume**

Kiintolevyn tai levyjen osio, joka näkyy yhtenä erillisenä osiona tietokoneelle

**Wizard**

Käyttäjän ohjaustyökalu, joka auttaa käyttäjää asentamaan sovelluksia ilman komentorivikomentoja

## 1 JOHDANTO

Keski-Uudenmaan koulutuskuntayhtymä Keuda on ammatillista koulutusta järjestävä opetuslaitos, jonka kaikki 11 toimipistettä sijaitsevat Uudellamaalla. Päätoimipiste sijaitsee Järvenpäässä, jonka lisäksi toimipisteitä löytyy Keravalta, Tuusulasta, Mäntsälästä, Sipoosta, Nurmijärveltä sekä Helsingistä. Keudalta voi valmistua yli 50:lle eri tutkintonimikkeelle. Henkilöstöä Keudalla on noin 650, joista kaksi kolmasosaa on opetushenkilöstöä. (Keuda, n.d)

Opinnäytetyö suoritetaan Keudan Nurmijärven toimipisteellä. Nurmijärvellä voi opiskella ammatillisen perustutkinnon Autoalalta, Liiketaloudesta, Media- ja kuvallisen ilmaisun alalta, Rakennusalalta, Ravintola-alalta, Sosiaali- ja terveysalalta, Sähkö- ja automaatioalalta ja Tieto- ja viestintätekniiikan alalta (Keuda, n.d). Tulen viittamaan tässä työn kirjallisessa osassa Media- ja kuvallisen ilmaisun linjaa lyhyesti medialinjana, sekä Tieto- ja viestintäteknikkaa ICT-linjana.

ICT-linja on saanut lahjoituksena tietoteknistä laitteistoa, johon sisältyi enimmäkseen Hewlett-Packardin palvelinkalustoa. Samanaikaisesti Medialinjalla oli tarve omalle mediapalvelimelle. Palvelimen pääasiallinen tarkoitus olisi mahdollistaa opiskelijoiden opintoihin liittyviä harjoituksia ja tehtäviä koulun sisäisesti, esimerkiksi videon streamausta, ilman että julkiseen verkkoon lähtisi oppilaiden materiaalia. Tämän lisäksi oppilaat voisivat jakaa palvelimella VOD - tyyppisesti materiaalia toisilleen, sekä opettajille.

Tässä opinnäytetyössä on siis tarkoitus ottaa käyttöön entuudestaan tuntematon laitteisto, asentaa se, sekä suunnitella ja kehittää sille toimiva kokonaisuus, joka vastaisi Medialinjan toiveita. Työn tilaajana toimii näin ollen Keuda Nurmijärven ICT-linjan päällikkö, ja se toteutetaan yhteistyössä Nurmijärven Medialinjan kanssa.

Työn kustannukset halutaan pitää mahdollisimman pieninä, joten työssä käytettävät laitteisto pyritään kokoamaan koululta valmiiksi olemassa olevista laitteista sekä ohjelmistoista, joihin joko koululla on lisenssit, tai ne ovat ilmaisia vapaan lähdekoodin ohjelmistoja.

Palvelinkokonaisuuden ylläpito tulisi olemaan ICT-opiskelijoiden vastuulla, joten ympäristöksi oli työn tilausvaiheessa määritelty Windows, joka on opiskelijoiden pääasiallinen toimintaympäristö, ja siihen koululla oli lisenssit valmiina. Käyttöjärjestelmäksi päättyi siis Windows server 2016 standardiversio graafisella käyttöliittymällä.

## 2 LAITTEISTO JA TESTAUS

Työn ensimmäisenä vaiheena oli tehdä laitteista ns. kartoitus. Ensisilmäyksellä voidaan todeta, että laitteisto on pääasiassa HP:n, eli Hewlett-Packardin kalustoa.

Palvelinkokonaisuus tulee näin ollen toteutettua pääasiassa HP:n laitteistolla.

Alla olevassa listassa laitteisto kokonaisuudessaan

- 1 x HPE BladeSystem c7000
- 6 x HP ProLiant BL460 Gen7 Server Blade
- 6 x HP ProLiant BL460 Gen6 Server Blade
- 4 x HP ProLiant BL460c Gen6 Server Blade
- 1 x HP ProCurve 2610-24 Switch
- 1 x Cisco Catalyst 3750G-24TS
- 28 x HP Dual Port 72GB 6Gb/s SAS Hot-swap HDD

Näistä laitteista ei kuitenkaan päätynyt työhön kuin osa. Syyt laitteiden poisjättämiseen olivat joko se, että laite oli tekniikaltaan liiaksi vanhentunut, tai laite ei muuten ollut yhteensopiva. Osa laitteista oli myös rikki.

### 2.1 HPE BladeSystem c7000 enclosure

Kokonaisuuden keskustana toimii kyseinen Rakkipalvelinkotelo (Kuva 1). Koteloon sopii 16 Server Blade -nimellä tunnettua normaalipituista räkkietokonetta, tai vaihtoehtoisesti 8 täysipitkää konetta. Lisäosan avulla täysipitkiäkin voidaan liittää koteloon 16 kappaletta (HPEsupport, 2011). Kyseisestä räkkikotelosta on tehty kolmea versiota, joista työssä käytetty on päivitetty ensimmäisen versio. Päivitetty versio sisälsi hallintapaneelin, johon

kuuluu etupuolen alaosassa näkyvä ohjausyksikkö, joka on pieni kosketusnäyttö neljällä fyysisellä napilla.

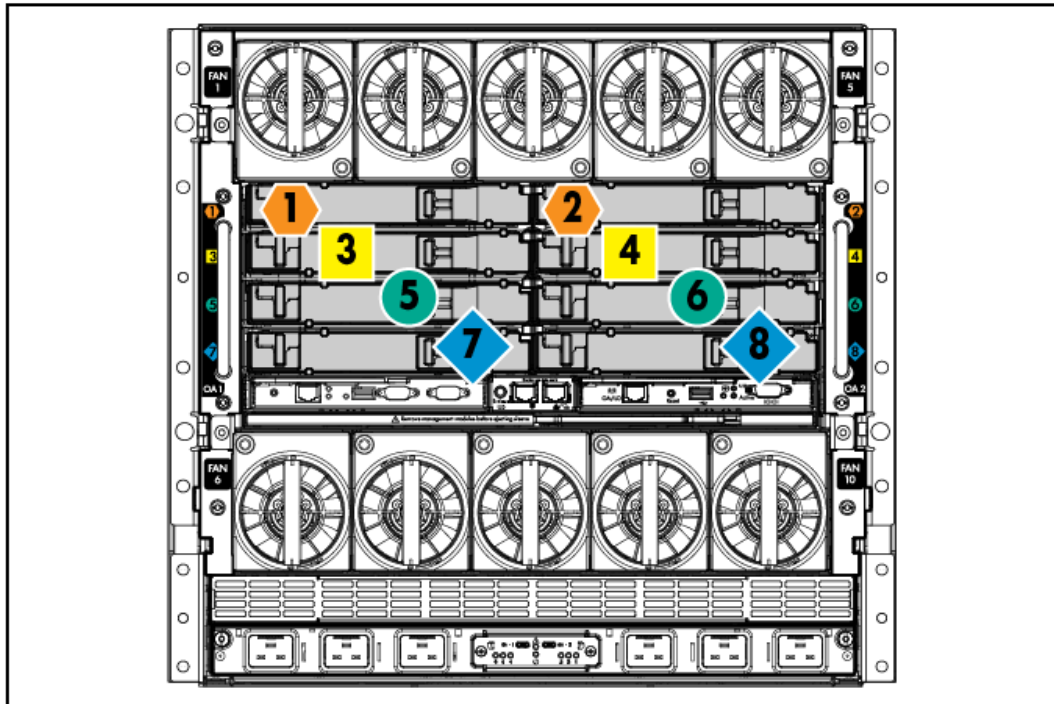
Kuva 1 HP BladeSystem c7000 edestä ja takaa (Sanghera, 2016)



Vakiovarusteltuna räkkikoteloon ei sisälly kuin virranjakoyksikkö ja edellä mainittu hallintapaneeli. Ensimmäinen omistaja tai jälleenmyyjä siis määrittelee mitä osia koteloon otetaan, kun se tilattiin uutena HP:lta.

Työssä käytettävään koteloon oli otettu suurin mahdollinen määrä jäähdytysyksiköitä, 10 kappaletta. Loputkin osat olivat HP:n suositusten mukaan 8 ns. interconnect bayta (kuva 2), joista meille tulee käyttöön vain kaksi ensimmäistä. Tämä siitä syystä, että myöhemmin selvisi, ettei käytössä olevilla tietokoneilla ollut Mezzanine kortteja lisäyhteyksiä varten.

Kuva 2 c7000 interconnect bay -paneelit



Server blade signal	Interconnect bay	Interconnect bay labels
FlexibleLOM 1 and 2	1 and 2	Orange hexagon
Mezzanine 1	3 and 4	Yellow square
Mezzanine 2 and 3	5 and 6	Green circle
Mezzanine 2 and 3	7 and 8	Blue diamond

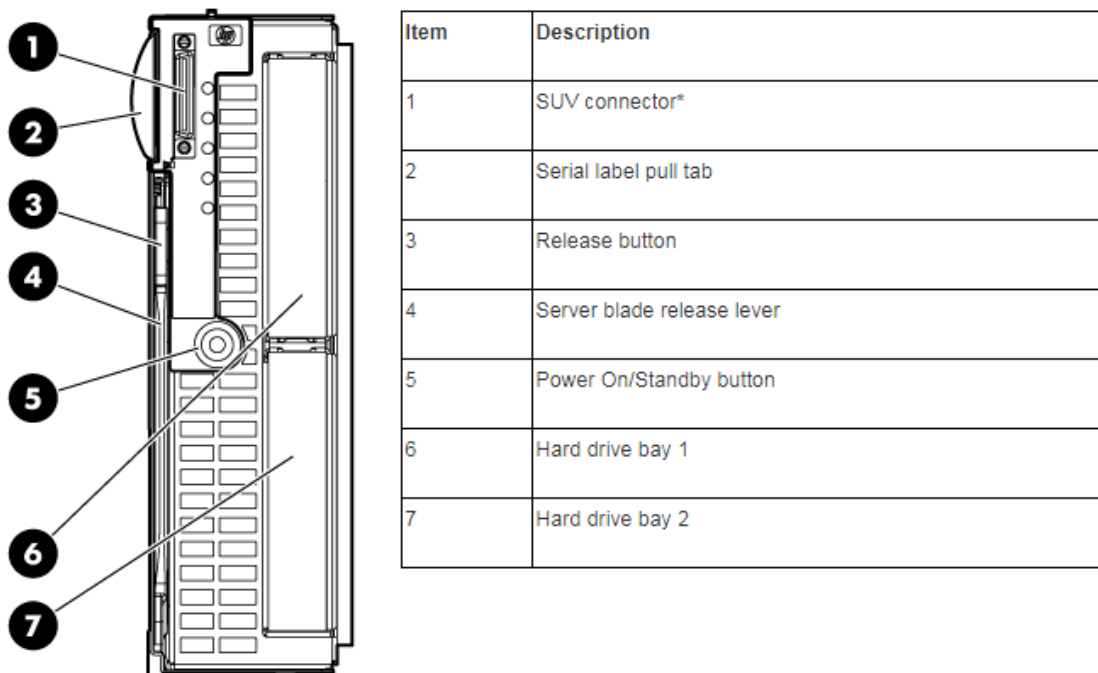
Räkkikotelon testauksessa ei ilmennyt mitään ongelmia, kaikki osat vaikuttivat toimivan moitteetta. Tosin testaus oli pakostakin toteutettava yhdessä siihen kuuluvien tietokoneiden kanssa, joten varmuutta ei koskaan voinut saada mezzaninekorttien takapaneeleista, sillä niitä ei voinut tietokoneista puuttuvien mezzaninekorttien vuoksi testata.

## 2.2 HP ProLiant BL460 Gen7 Server Blade

Räkkiin sijoitettavat tietokoneet kantavat HP:n lanseeraamaa Server Blade nimeä. Server Bladeja on huomattavan monta mallia enemmän kuin räkkikoteloita. Parhaat tietokoneet, jotka lahjoitetusta laitteistoista löytyivät, olivat siis ProLiant BL460 seitsemännenten sukupolven Server Bladeja.

Kuten rakkikotelossa, myös Server Bladeissa piti päättää, mitä ominaisuuksia niihin haluaa tilausvaiheessa. Tutkittuani sisuskaluja, näin että kyseiset Gen 7 -palvelinkoneet olivat identtisiä. Samalla tuli huomattua, että ne eivät tehokkuudeltaan vaikuttaneet välttämättä parhailta niille suunniteltuun tehtävään, siis streamauksen käsittelyyn. Tiesin siis tässä vaiheessa jo, että haluan muodostaa streamipalvelun palvelinklusterista, jotta oppilaiden sitä rasittaessa harjoituksillaan, se ei antaisi täysin periksi. Ainoa looginen ratkaisu oli rasituksen jako useamman palvelinkoneen kesken.

Kuva 3 BL460 Server Blade -etupaneeli (helpmax, n/a)



BL460 G7 -palvelinkoneen sisuksiin lukeutui:

- prosessorit: 1 kpl Intel Xeon E5640 prosessoria
  - 4 ydintä
  - 8 threadia
  - 12 MB cache
  - 2,67 GHz kellotaajuus
- RAM: 3 x 2GB 2000MHz DDR3 muistikampaa
- 1 kpl NC553i Dual Port FlexFabric 10Gb, eli kaksi NICia yhdessä verkkokortissa
- iLO3 prosessori ja sille omistettu ylimääräinen NIC
- Integroitu GPU
- Intel 5520 Chipset.
- 2 x Hot-swap paikkaa SAS-kiintolevyille.

Listasta huomaa heti, että muistin määrä vaikuttaa aivan liian vähäiseltä. Myös prosessorin laskentateho vaikuttaa hieman heikolta ensisilmäyksellä nykyaikaisempiin palvelinprosessoreihin verrattuna. Se on silti aivan riittävä työn tarpeisiin, sillä käyttäjiä ei mitä todennäköisimmin tule olemaan liikaa yhtäaikaisesti, sillä tarve oli vain Nurmijärven toimipisteen sisäverkon käyttäjille. Suurin ongelma on siis RAMin määrä, mitä tulisimme tarvitsemaan videodatan käsittelyssä reilusti.

Server Bladeissa ei turvallisuussyistä löydy USB- tai muita sarjaportteja, eikä myöskään minkään tyyppisiä liittimiä esimerkiksi näytön yhdistämiseen. Kaikki liitännät tehtiin HP:n omalla SUV-adapterilla (kuva 4). SUV tulee sanoista Serial USB VGA. Adapterin avulla saamme siis yhdistettyä näytön VGA porttiin, näppäimistön ja/tai hiiren USB-yhteydellä ja mahdollisen lisälaitteen DB9 tyyppisellä sarjaliittimellä.

Kuva 4 SUV-adapteri



SUV-liittimestä jäi laitteiston testauksessa ristiriitainen mielipide itselleni. Toisaalta se toimi yllättävän hyvin, siinä on hyvin toimiva lukitusmekanismi, joten näytön ja näppäimistön siirtäminen palvelinkoneesta toiseen liittimen avulla oli todella helppoa ja nopeaa. HP oli siis onnistunut kehittämään liittimen hyvinkin käyttäjäystävälliseksi. Mutta kääntöpuolena ilmeni, että liitin ei ollut kestävyydeltään luotettava. Laitteistotestauksen yhteydessä ilmeni, että kahdesta palvelinkoneesta oli kyseinen liitin mennyt rikki, ja siksi koneita oli mahdotonta päästä asentamaan ja konfiguroimaan. Koneissa kun ei muita liityntämahdollisuuksia ole.

Yhdestä Gen 7 -palvelinkoneesta oli myös virtalähde hajonnut, joten se, ja toinen liitinvikaisista koneista, joka osui myös G7-koneisiin, vähennettynä toimivista laitteista, jättää työhön jäi neljä toimivaa BL460 Gen 7 -palvelinkonetta.

### 2.3 HP ProLiant BL460 Gen6 Server Blade

Ulkonäöllisesti ja ulkoisilta ominaisuuksiltaan täysin samanlainen kuin Gen 7 -kone (kuva 3), tämä palvelintietokone eroaa vain sisäisiltä komponenteiltaan. Eroavat komponentit listattu alle:

- Prosessori: Intel Xeon X5550
  - 4 ydintä
  - 8 threadia
  - 8MB cache
  - 2,66 GHz kellotaajuus
- RAM: 3 x 6GB 1333 MHz DDR3
- Intel 5500 Chipset

Kyseessä on siis työtä ajatellen jälleen täysin pätevä tietokone muilta ominaisuuksiltaan, mutta vieläkin huonompi RAM, kuin vuotta uudemmassa Gen 7 -koneessa. Muistin 1333MHz nopeus on siis G7 konettakin hitaampi, vaikka muistin määrä yhtä suuri onkin. Työn tilaajan mielestä totutetaan työ valmiilla komponenteilla, ja tulevaisuudessa harkitaan sitten tehojen lisäyksiä.

Toinen SUV-liitinvikaisista tietokoneista oli kyseenomaisia Gen 6 -koneita, joten käyttöön ei jäänyt kuin viisi BL460 Gen 6 -konetta. Yksi koneista hajosi myös työn rakennusvaiheessa rakkikotelon antaessa virheilmoituksen media error.

### 2.4 HP ProLiant BL460c Gen6 Server Blade

Nämä tietokoneet eivät päätyneet työhön mukaan. Palvelinkoneiden iLO prosessori oli vanhempaa mallia muihin Gen 6 -palvelikoneisiin verrattuna. Muissa malleissa oli siis iLO3, mutta näissä halvemmissä malleissa oli käytetty iLO2:ta. Tämä esti käyttöjärjestelmän asennuksen tietokoneille, sillä tietoturvasyistä käyttöjärjestelmiä ei voinut Server Bladeille asentaa USB:n kautta, ja muita ulkoisia mahdollisuuksia käyttöjärjestelmän asennukseen ei ollut tarjolla (HPESupport, 2011).

iLO2 käyttää niin vanhaa .NET frameworkia, että käytössä olevilla laitteillani sitä ei enää tuettu. Ongelman olisi voinut kiertää hankkimalla vanhemman, esimerkiksi Windows XP, käyttöjärjestelmän omaavan tietokoneen asennustöihin. Työn tilaajan kanssa keskusteltuani, oli hän sitä mieltä, ettei ole järkevää enää tukea niin vanhaa tekniikkaa. Jätimme siis kyseiset palvelinkoneet kokonaan pois työstä.

## 2.5 SAS-kiintolevyt

Jokaisessa HP:n Server Blade -koneessa on paikka kahdelle Hot-swap ominaisuuden omaavalle kiintolevylle. HP tuottaa valmiita Hot-swap kiinnityskehyksen omaavia kiintolevyjä valmiskäyttöön Bladeserver-ympäristöissä (kuva 5), ja lahjoitetun laitteiston yhteydessä niitä olikin 28 kappaletta mukana. Kaikki kiintolevyt olivat samanlaisia 2,5 tuuman mekaanisia HDD levyjä. Ne olivat tilaltaan varsin pieniä, kukin vain 72GB.

Kuva 5 HP Dual port SAS Hot-swap kiintolevy



Tässä ilmeni työn ensimmäinen ylitsepääsemätön ongelma. Palvelimemme tulisivat mediapalvelujensa vuoksi tarvitsemaan paljon tallennustilaa. Vaikka pystyisimme hyödyntämään tekniikoita, joilla voisi yhdistää kaikkien laitteiden tallennustilan yhdeksi, ottaen tietenkin huomioon käyttöjärjestelmien varaaman tilan, olisi silti käytettävissä liian pieni tallennustila kunnollisiin mediapalveluihin.

Juttelin siis työn tilaajan kanssa ja päädyimme hankkimaan 4 kappaletta tallennustilaltaan suurempia kiintolevyjä. Hankittavat levyt tulivat olemaan 1,2TB suuruisia, muutoin ominaisuuksiltaan yhteneviä muiden kiintolevyjen kanssa. Kaikki kiintolevyt olivat SAS-

tekniikan ja 7200 RPM omaavia levyjä. Kiintolevyjen teoreettinen tiedonsiirtonopeus on 6GB/s.

## 2.6 Kytkimet

Yhteyden muodostamiseksi toisiinsa ja käyttäjille, tarvitsivat palvelinkoneet laadukkaan kytkimen. Olisin halunnut toteuttaa työn HP:n laitteistolla kokonaan, mutta HP:n oma ProCurve-kytkin ei soveltunut hitaiden nopeuksiensa vuoksi työhön. Tulevaa palvelinklusteria varten vaatimuksena oli vähintään GigabitEthernet (Microsoft, 2018). Tämä onneksi löytyi Ciscon Catalyst 3750G -kytkimestä. Todennäköisesti laitteiston lahjoittaja oli huomannut saman ongelman aikanaan, ja siksi Ciscon kytkin laitteistossa olikin mukana, vaikka muutoin laitteisto oli vain HP:n omia laitteita.

Suositus on, että laitteisiin olisi useampi reitti yhden pisteen virheiden, eli SPOF:n, välttämiseksi (Rouse, 2012), mutta annetuissa olosuhteissa verkkoreitin seuraava hyppy kulkee väkisinkin yhden ja saman kytkimen kautta Keudan sisäverkossa. Kokonaisuudessa on jo SPOF, joka on vaikutusmahdollisuuteni ulkopuolella, joten päätin että työ voidaan toteuttaa vain yhdellä kytkimellä. Näin ollen HP:n ProCurve -kytkin jäi kokonaan pois työstä.

Ciscon Catalyst 3750G-24TS -kytkimessä oli kaikki työhön tarvittavat ominaisuudet. Kaikki sen 24 porttia olivat kukin Gigabit Ethernet -portteja, joilla saimme palvelinympäristön kytkettyä toisiinsa 1000Base-T liittymillä. Kytkin kylläkin syö melko paljon tehoja, 190W. Ciscon laitteistolle tyypillisesti, siihen pystyy, jos tarve tulee, asentamaan myös VLAN verkkoja, vaikka pärjäämme työssä sen oletus VLAN:ssa. (CNET, n.d)

Kuva 6 Cisco Catalyst 3750G-24TS (CNET, 2010)



## 2.7 Käytettävissä oleva laitteisto

Ilmenneistä syistä useampi laite jää näin ollen hyödyntämättä työhön. Alle on listattu laitteet, jotka oli mahdollista ottaa työn seuraavaan vaiheeseen, eli suunnitteluun, mukaan.

- HP Bladesever enclosure c7000
- 4 x HP Server Blade BL460 Gen 7
- 5 x HP Server Blade BL460 Gen 6
- 14 x HP Dual Port 72GB 6Gb/s SAS Hot-swap HDD
- 4 x HP Dual Port 1.2TB 6Gb/s SAS Hot-swap HDD
- Cisco Catalyst 3750G-24TS -kytkin

Huomioitavana seikkana se, että vaikka kaikki kiintolevyt vaikuttivat toimivalta, vain yhdeksän palvelinkoneen käyttömahdollisuuden myötä kiintolevykoja ei ollut kuin 18. Joten työhön otettiin nuo neljä suurempaa kiintolevyä ja loput paikat täytettiin pienillä.

### 3 PALVELINJÄRJESTELMÄN SUUNNITTELU

Nyt kun käytettävissä oleva laitteisto oli selvillä, pystyin siirtymään työn suunnitteluvaiheeseen. Suunnitteluvaihe kannattaa ehdottomasti aloittaa kertaamalla mitkä olivat työn tilaajan vaatimukset.

#### 3.1 Vaatimukset

##### 3.1.1 Ohjelmistoympäristö

Työn tilaaja haluaa työn toteutettavan Microsoftin Windows -ympäristössä. Palvelinjärjestelmän ylläpito jäisi tulevaisuudessa Keudan ICT-opiskelijoiden tehtäväksi koulutusmielessä, ja Keuda painottaa palvelinopinnoissaan Windows-ympäristöä.

##### 3.1.2 Streamipalvelin

Palvelinkokonaisuuden tärkein tehtävä on mahdollistaa medialinjan opiskelijoiden stream tekniikoiden harjoittelu suljetussa sisäverkossa. Palvelun tavoite on ottaa siihen tarkoitetuilta laitteilta tai ohjelmistoilta RTMP dataa ja ohjata se käyttäjien nähtäväksi adaptiivisella katselumahdollisuudella. Työn tilauksessa käytettiin kuvailuna 'Youtuben kaltainen streamaus'.

##### 3.1.3 Videon katselualusta

Jotta streameja ja muuta mediaa voidaan katsella, täytyy niitä varten olla jokin alusta, missä käyttäjät voivat niitä toistaa ja katsella.

##### 3.1.4 Videon suoratoistopalvelu

Tämä ei ollut varsinainen vaatimus ja voidaan ennemminkin luokitella vahvaksi toivomukseksi. Jälleen Youtubea esikuvana käyttäen, toivoi tilaaja palvelua, johon voidaan ladata videoita, joita voidaan sitten myöhemmin VOD tyyppisenä toistaa katselualustalla.

### 3.1.5 Aktiivihakemisto

Alkuperäinen kuvaus oli, että palveluiden käyttäjät kuuluisivat saman Windowsin aktiivihakemiston alle. Tämä lähinnä siitä syystä, että domainin ja aktiivihakemiston valvominen sisäverkossa olisi hyvää harjoitusta ICT-opiskelijoille ja kaikkien käyttäjien valvominen olisi helpompaa, jos muut kuin aktiivihakemiston käyttäjät eivät pääsisi käsiksi palveluihin.

Lyhyen keskustelun jälkeen muutin tahtomattani tilaajan mielipidettä siten, että videoita olisi muidenkin kuin aktiivihakemiston käyttäjien kiva päästä katselemaan. Työn uuteen kuvaukseen siten kuului yhä aktiivihakemisto ja uusi domain, mutta vain streamauspalvelun käyttäjät tulisivat olla aktiivihakemiston käyttäjiä. Kaikilla sisäverkkoon pääsevillä olisi näin ollen mahdollisuus katsella streameja, toistaa ja ladata VODEja, ja hyödyntää muita mahdollisia palveluja.

## 3.2 Suunnittelu

Itse suunnitteluosan jaoin kahteen vaiheeseen. Ensimmäinen vaihe oli pyydettyjen palvelujen toteutus teoriassa, mitä tekniikoita ja ohjelmia tulisi palvelimille asentaa. Toinen vaihe on palvelinkoneiden asennusten suunnittelu ja palvelinroolien jako palvelinkoneiden kesken.

Tiesin myös heti, että käyttöjärjestelmiksi valikoituisi Windows server 2016, sillä keksin jo laitekartoituksen yhteydessä haluavani muodostaa streamipalvelun palvelinklusterilla, jotta räsitus saataisiin jaettua. Tämä on tehty huomattavan paljon helpommaksi server 2016 ja uudemmissa versioissa, kuin niitä edeltäneissä versioissa (Microsoft, 2018). Kyseinen fakta yhdistettynä siihen, että Server 2016 oli ICT-opiskelijoiden pääasiallinen opiskelukohde, valinta oli helppo.

Räsitusjaon vaihtoehtoista Windowsin NLB ominaisuutta ei voinut käyttää, sillä NLB on staattista räsituksen jakoa, ja streamipalvelimen tulee käsitellä jatkuvasti syötettävää streamidataa. Palvelinklusteri tarjoaa mahdollisuuden jatkuvasti muokattavan datan käsittelyyn. (K2, viitattu 2020)

### 3.2.1 Palvelut

Aloitin palveluiden suunnittelun miettimällä videoiden toistoalustaa, ja mitään muuta järkevää vaihtoehtoa en keksinyt, kuin sille omistettu WEB-palvelin ja verkkosivut. Verkkosivuille toteutettaisiin livestreamien katselumahdollisuus, VOD videoiden katselumahdollisuus, videoiden palvelimelle lataus- ja pakkaustoiminnot, sekä tietenkin Streamin aloitus- ja ohjesivut aktiivihakemiston käyttäjille.

Tuumasin myös, myös että Videon latauspalvelun voisi sijoittaa samalle palvelimelle Web serverin kanssa. Tämä helpottaisi palvelujen toteutusta, mutta tuottaisi taas huomattavan paljon lisää rasiitusta Web-palvelimelle.

Tässä vaiheessa sainkin idean, että en toteuttaisikaan palvelinkokonaisuutta yhdellä klusterilla, vaan kahdella erillisellä klusterilla. Näin voisin yhden klusterin omistaa kokonaisuudessaan streamipalvelimeksi, jossa ei paljon muuta työtä tehtäisi kuin streamien muodostamiseen edellytettävän moottorin, niin sanotun stream-enginen laskutoimitukset. Streamiklusteri ottaisi streamidatan vastaan, käsittelisi ja pakkaisi sen adaptiiviseksi, ja lähettäisi sen eteenpäin web-palvelimelle.

Toisessa klusterissa olisi siten kaikki muut palvelut, eli edellä mainitut verkkosivut ja videon latauspalvelu, sekä mahdolliset muut palvelut. Latauspalvelu vaatii myös enkooderin, että palvelimelle lähetetyt videot tulisivat olemaan aina muodossa, jota voidaan verkon yli toistaa. Kaikkea videodataa emme pysty verkon yli normaalinopeuksilla toistamaan, sillä videot voivat kuvauksen jälkeen olla esimerkiksi täysin pakkaamattomia ja näin ollen voivat olla huiman suuria tiedostoja, joiden verkon yli toistaminen ei ole käytännöllistä (Jenkins, 2020).

Youtube jälleen esikuvana, videot enkoodataan useampaan muotoon niiden palvelimelle latauksen yhteydessä. Hyvällä enkooderilla voimme toteuttaa yhdestä videosta monta erilaatua videota erilaisille toistolaitteille, sekä eriäville verkon nopeuksille käyttäjäkohtaisesti. (Jenkins, 2020)

### 3.2.2 Palvelimet ja roolit

Seuraavaksi piti miettiä mitkä roolit olisi jokaiselle palvelinkoneelle jaettava. Aloitin suunnittelun miettimällä, että kaikki koneet tulisi liittää aktiivihakemiston yhteydessä luotavaan uuteen domainiin.

Jokainen Palvelin, joka hallinnoi ADDS:aa, luokitellaan Domain Controlleriksi ja suositus on, että DC palvelimia on vähintään kaksi kappaletta. Suurin syy tähän on se, että ongelmatilanteessa koko domain ei kaadu, vaikka yksi Domain controller putoaisi pois käytöstä hetkellisesti. (Vincent, 2019)

Päätin siis asettaa kaksi palvelinkonetta Domain Controllereiksi. Nämä koneet eivät tulisi liitettäväksi kumpaakaan palvelinklusteriin, vaan niiden tehtävänä on vain hallinnoida domainia ja aktiivihakemistoa, jonne streamipalvelun käyttäjät muodostettaisiin. DC:t ovat koko palvelinkokonaisuuden pohjana, joten päätin valjastaa siihen kaksi kappaletta parempia Gen 7 Server Bladeja, parantaakseni hieman luotettavuutta, vaikkakin vain hyvin vähän.

Loput palvelinkoneet voimme siis valjastaa kahteen klusteriin, toiset streamipalvelulle, toiset verkkosivuille ja latauspalvelulle. Ensimmäisen klusterin ajattelin siis valjastaa streamipalvelulle, joten päätin ottaa loput Gen 7 -koneista tähän klusteriin, niiden ollessa laskentateholtan hieman parempia loppuihin koneisiin verrattuna. Jäljelle jäävät koneet voin näin ollen valjastaa toiseen klusteriin, jossa loput palvelut tulisivat olemaan.

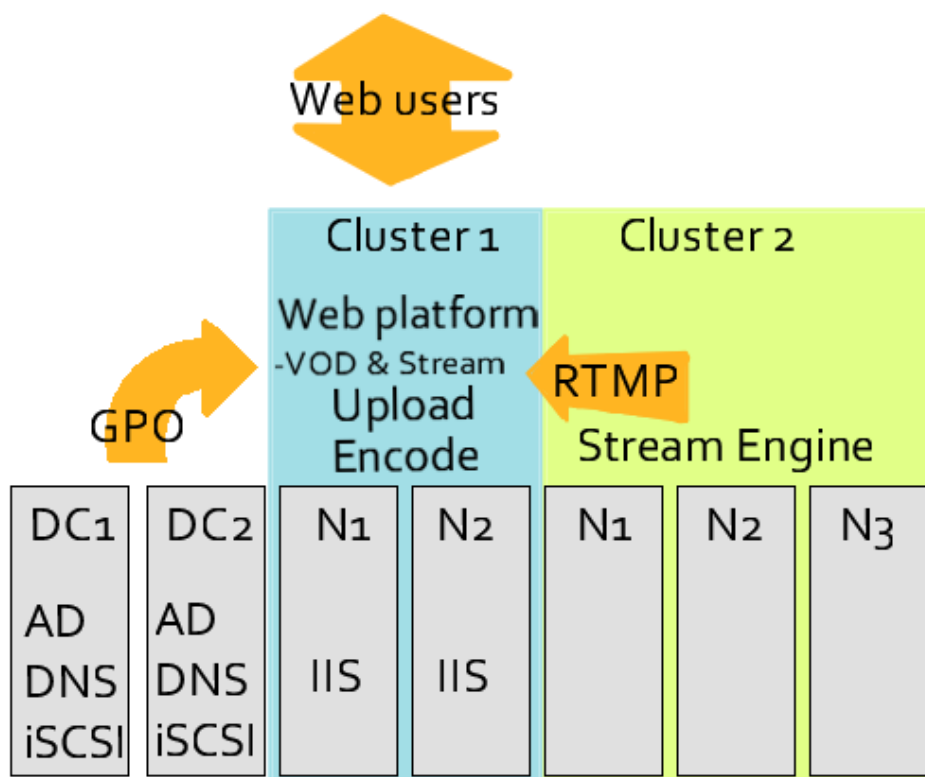
### 3.3 Suunnitelman yhteenveto

Kuvassa 7 näkyy lopullisen suunnitelman piirrosversio. Kuva on piirretty jäljitellen c7000 rakkikoteloon sijoitettavien palvelinkoneiden fyysisiä sijainteja, näin ollen asennus ja kehitysvaiheissa on helppoa havainnoida missä mikäkin palvelu ja rooli sijaitsevat.

Palvelimen sisään kirjoitetut roolit viittaavat Windowsin tarjoamiin rooleihin, mitkä tulee asentaa ja konfiguroida, ja koneiden ulkopuolelle jäävien klusterin yhteyteen kirjoitetut palvelut tulisi olla kolmannen osapuolen tuottamia palveluja tai ohjelmistoja, tai vaihtoehtoisesti kehittää itse.

DC1 ja DC2 ovat hallintakoneita, jotka ohjaavat aktiivihakemiston käyttäjät webalustalle Group Policy Objektilla. Palvelinkoneille siis asennetaan Aktiivihakemisto, jonka yhteydessä niille asennetaan myös DNS palvelu, joiden kautta nimenselvennykset toteutetaan. Tämä on vaadittava ominaisuus ADDS:n yhteyteen (Nelson, 2016). Hallintaroolin lisäksi koneet toimivat iSCSI levyjen fyysisenä sijaintina. Asensin myös DHCP palvelimen DC-koneille, vaikka suunnitelmassa en sitä toistaiseksi tarvinnutkaan, mutta oli hyvä varautua mahdollisiin muutoksiin ja laajennuksiin. Se ei kuitenkaan näy suunnitelmassa.

Kuva 7 Palvelinjärjestelmän suunnitelma



Klusteri yksi toimii web palvelimena, jolle kehitän websivuston, joka toimii videoiden katselualustana. Videoita on siis tarkoitus päästä katsomaan muutkin kuin aktiivihakemiston käyttäjät, mutta heillä ei tulisi olla oikeutta streamaukseen alustalla. Videoiden katselun lisäksi käyttäjille pitää luoda mahdollisuus ladata palvelimelle omia videoita, jotka vastaavasti VOD palvelua varten tulee käsitellä ja pakata useampaan eri tiedostokokoon.

Klusterin 2 tehtävänä on toimia streamimoottorina. Moottorin tehtävä on pakata ja muodostaa adaptiivinen, siis useampi erilaatuinen, streami reaaliajassa webalustalle

sopivaksi (Ozer, 2019). Streamidata sitten lähetetään rtmp protokollalla webserverille, josta käyttäjät pääsevät sitä tarkastelemaan http protokollan yli. Microsoft ei tarjonnut helposti asennettavaa streamimoottoria, joten joudun etsiä käyttöön soveltuvan moottorin jostain muualta. Microsoft panostaa streamitekniikassa Azure palveluunsa, ja sen käyttöönotto pienessä sisäverkon palvelinkokonaisuudessa ei tuntunut luentevalta, koska Azure on sidottu Microsoftin pilvipalveluihin.

## 4 PALVELIMIEN ASENNUS JA KONFIGUROINTI

Käyttöjärjestelmäksi palvelimille valikoitui Microsoft Windows Server 2016. Valinta perustui kahteen seikkaan, joista ensimmäinen on se, että kyseinen käyttöjärjestelmä on edelleen eniten käytetty käyttöjärjestelmä Keudan ICT-linjan palvelinjärjestelmien opinnoissa.

Työssä toteutuvien palvelimien ylläpito tulisi jäämään Keudan oppilaille, joten toistaiseksi oppilaille kaikista tunnetuin käyttöjärjestelmä oli tulevaisuutta ajatellen hyvä vaihtoehto.

Toinen valintaperuste oli se, että tiesin suunnittelun yhteydessä haluavani muodostaa useamman palvelimen klusterin, Windows Server käyttöjärjestelmien tarjoamalla Failover clustering ominaisuudella, ja tämän ominaisuuden käyttöönotto oli parantunut Windows Server 2016 versiossa huomattavasti edeltäjiinsä nähden (Microsoft, 2018).

Asennustyö ei kuitenkaan suju ilman haasteita HP:n Bladeserver -järjestelmässä, sillä turvallisuuden vuoksi HP estää mm. käyttöjärjestelmän asennuksen USB-asennustikulta, vaikka SUV-adapterilla meidän on mahdollista USB-laitteita liittää palvelinkoneisiin. HP:n asennusohjeen mukaan Palvelinkoneille tulisi asentaa käyttöjärjestelmä niiden iLO moduulin kautta niin sanotulla virtuaalisella DVD-aseamalla (Hewlett Packard, 2010).

### 4.1 Käyttöjärjestelmän asennus iLO moduulilla

Blade Serverien sisältämä iLO moduuli tarkoittaa palvelinkoneen sisäänrakennettua moduulia, jolla voi hallita etäyhteydellä kyseistä palvelinkonetta, vaikka se olisi käytännössä pois päältä, eli vain herätevirralla. Juuri tästä ominaisuudesta moduuli saakin nimensä,

integrated lights-out module. ILO on HP:n vastine muille vastaaville LOM tekniikoille. Moduuli on käytännössä oma itsenäinen tietokoneensa, rakennettuna toisen tietokoneen sisälle. ILO:n käyttöjärjestelmä on HP:n kehittämä, samaa nimeä kantava, moduuliin kovakoodattu käyttöjärjestelmä. Siihen pääsee käsiksi ethernetin yli tietokoneella, jolla käyttöliittymään avataan web selaimella, kirjoittamalla navigointipalkkiin kyseenomaisen iLO moduulin IP-osoite. (Hewlett Packard, 2010)

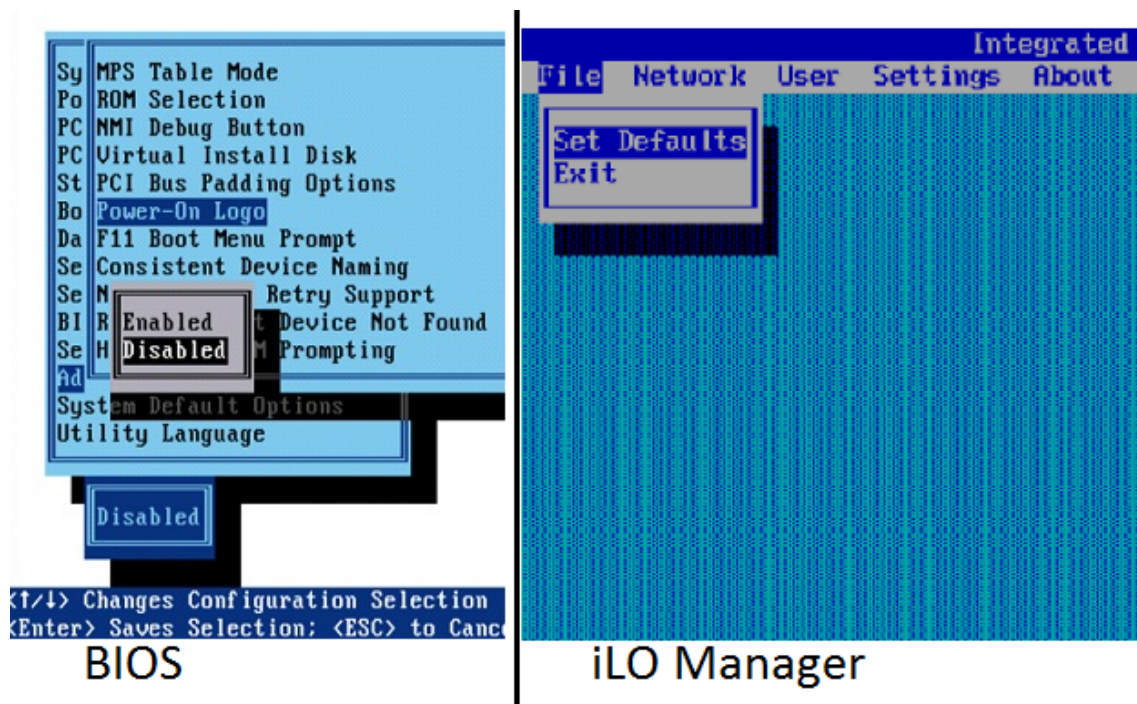
Räkkikotelon takaosasta löytyy hallintapaneeli, jonka ethernetporttiin liitytään tietokoneella. Tämän jälkeen liitetyn tietokoneen IP osoite tuli muuttaa samaan aliverkkoon missä iLO moduulin IP osoite sijaitsee. Tämän jälkeen saamme suoran yhteyden moduuliin webselaimessa.

#### 4.1.1 iLO manager

Saadakseni selville mikä on moduulin IP osoite, minun täytyi selvittää se Server Bladen yhteydessä olevasta iLO Managerista. HP on halunnut piilottaa iLO managerin. Aluksi siis, että on edes mahdollisuus päästä siihen, oli käytävä ensin Blade Serverin BIOS valikossa (kuva 8). Täältä täytyi poistaa käynnistyksen yhteydessä esitettävä grafiikka, ns. power-on logo.

Tämän jälkeen, kun tietokone käynnistyi uudelleen, pystyimme valitsemaan iLO Manageriin siirtymisen. iLO Managerissa pystyimme IP-tietojen lisäksi tarkastelemaan ja luomaan iLO käyttäjiä. ILO vaatii käyttöä varten käyttäjätiedot, sillä iLO:n tarkoitus on hallita sen omaavaa palvelinkonetta, ja käyttäjätunnuksilla rajoitetaan täysin ulkopuolisten pääsyä käsiksi hallinta-alustaan.

Kuva 8 Server Blade BIOS ja iLO manager



iLO manager tarjoaa kahdenlaista oikeustasoa, perus eli standard, sekä järjestelmänvalvoja eli administrator tasoa. Tarvitsin tarkoitustani varten, eli virtuaalisen DVD-aseman kautta käyttöjärjestelmän asennukseen, järjestelmänvalvojan oikeudet. Päätin luoda kokonaan uuden käyttäjän itselleni näillä oikeuksilla.

IP asetuksista huomasin, että iLO saa IP osoitteensa DHCP palvelimelta. Päätelin, että ProLiant rakkikoteloissa on DHCP palvelin sisäänrakennettu juuri siihen tarkoitukseen, että iLO moduuleille jaetaan rakkikotelon järjestyksenmukaiset IP osoitteet. Varmistin asian vaihtamalla Server Bladejen fyysistä paikkaa kotelossa, ja katsomalla, että moduulit tosiaan saivat fyysisen sijaintinsa mukaan jaetun IP osoitteen rakkikotelolta. Ne jaettiin paikallisella IP osoitealueella, 192.168.0.10 osoitteesta eteenpäin järjestyksessä, seuraava osoite näin ollessa 192.168.10.11 ja niin edelleen.

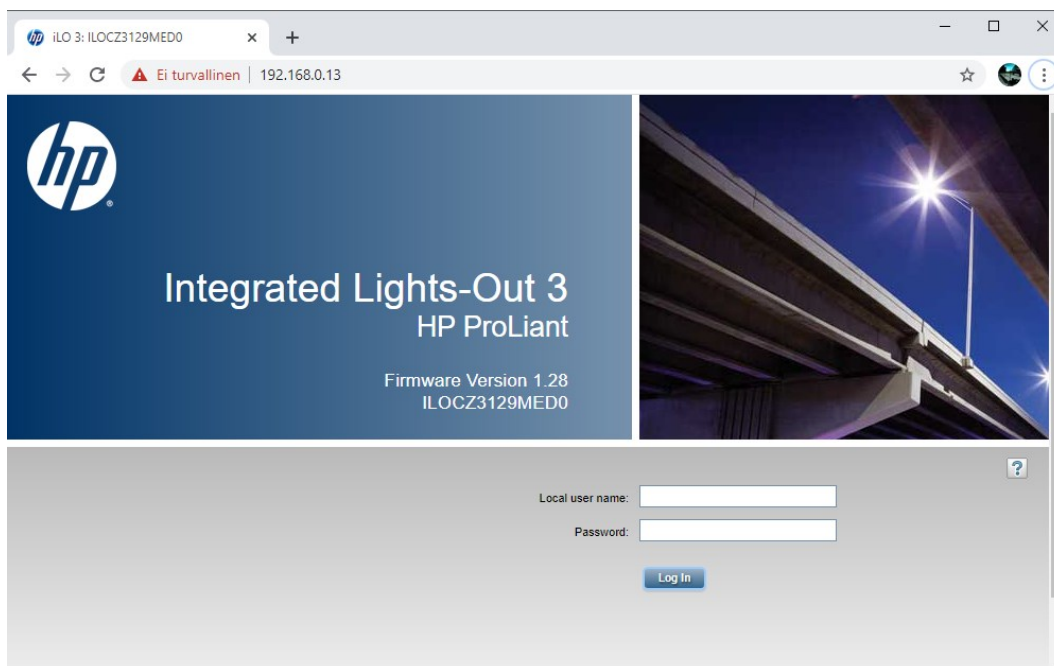
Tämä olisi nopeuttanut asennustyötä, koska se mahdollisti pääsyn yhdistämään jokaiseen moduuliin ennalta tiedettyjen IP osoitteiden vuoksi. Näin olisi säästyttävä vaivalta, jossa jokaisen palvelinkoneen kohdalla pitää päästä iLO manageriin. Ongelmaksi nousi se, että työn laitteisto oli käytettyä, ja edellinen omistaja tai omistajat olivat muokanneet moduulien käyttäjätietoja satunnaisiksi. Siispä jouduin jokaiseen iLO moduuliin käydä luomassa uuden

järjestelmänvalvojaoikeudet omaavan käyttäjän. Poistin samalla muut, tarpeettomat käyttäjät.

#### 4.1.2 Etäyhteyden muodostaminen palvelimeen

Nyt kun käyttäjätiedot ja IP osoitteet olivat selvillä, Pääsimme iLO:n käyttöliittymään tietokoneen webselaimella. Selaimen osoiteriviin tuli vain syöttää halutun moduulin IP osoite, ja mikäli tietokoneemme oli asetettu samaan aliverkkoon, saimme yhteyden moduulin käyttöliittymään (kuva 9).

Kuva 9 iLO käyttöliittymän kirjautumisruutu

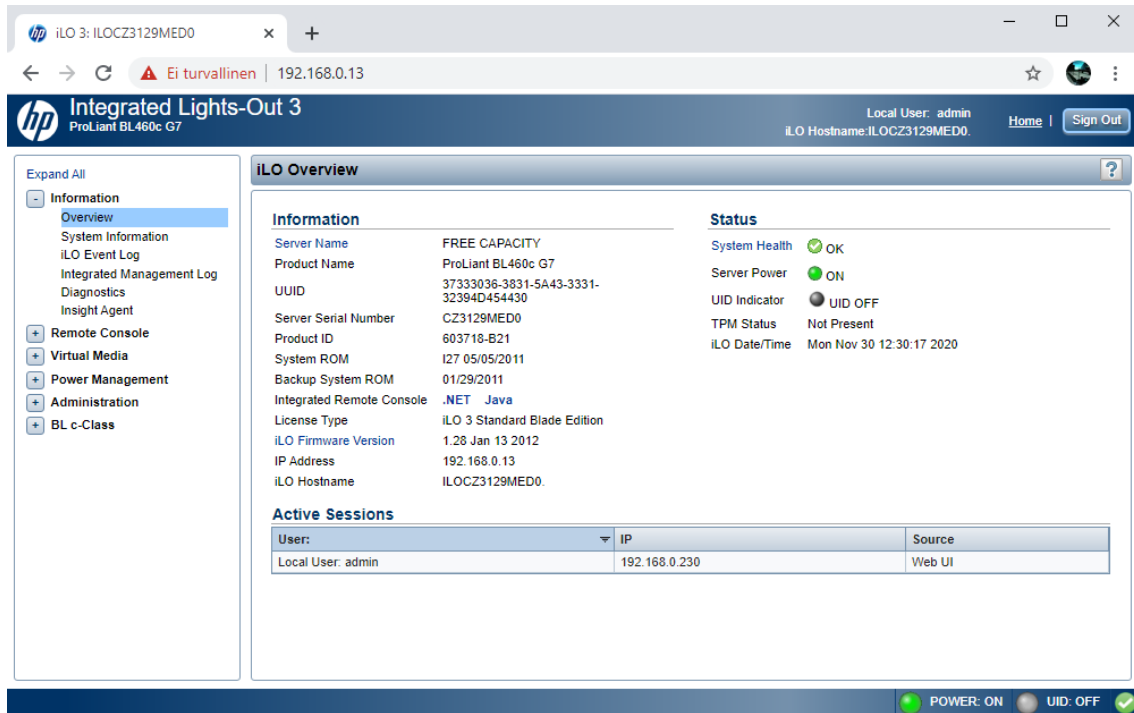


Aloitusruudussa iLO vaatii kirjautumisen luomallani käyttäjällä. Olin luonut jokaiselle iLO moduulille samannimisen, järjestelmänvalvojan oikeudet omaavan käyttäjän. Ne syötettyä aloitusruutuun, ohjataan webmanagerin työpöydälle (kuva 10). Tästä pääsee tarkastelemaan iLO moduulin tietoja, sekä sen omistavan Server Bladen tietoja.

Huomasin, että moduulin firmware on hyvin vanha, joka olisi varmaan kannattanut päivittää. Palvelimeltahan ei ollut pääsyä internetiin, joten emme voineet suorittaa helppoa internetpäivitystä. Siispä tuomasin, että koska ainoa tavoitteemme on saada

käyttöjärjestelmä asennettua etäyhteydellä palvelinkoneelle, ja jos onnistun sen tekemään, ei firmwaren päivittämiseen olisi ollut ehdotonta.

Kuva 10 iLO3 webkäyttöliittymä

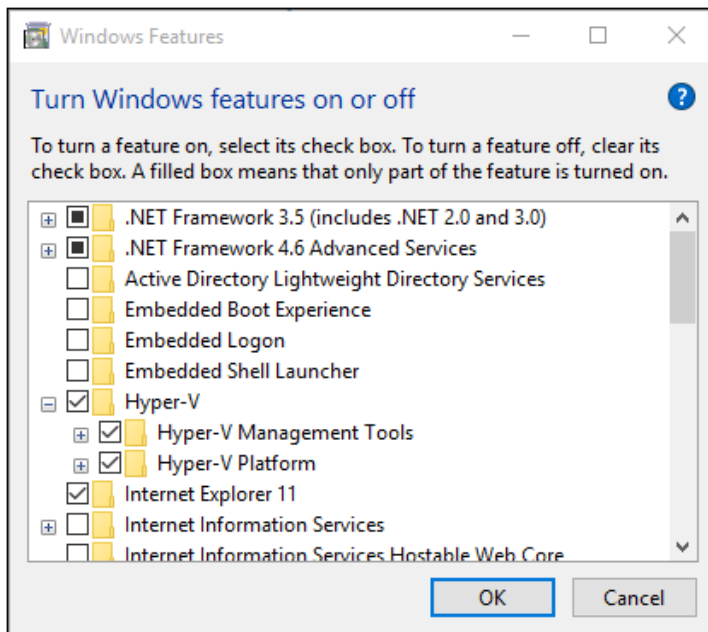


Managerissa näemme etäyhteyden muodostamiseen muutamana eri mahdollisuutena *Remote Console*n alla, tai suoraan pääruudussa. Vaihtoehdot olivat yhdistäminen .NET:llä, tai vaihtoehtoisesti Javalla. Tietokoneessani, jolla olin etäyhteyden muodostanut, ei ollut Javaa asennettuna, joten päätin kokeilla ensin .NET vaihtoehtoa. Se ei onnistunut suoraan. Selkeää dokumentointia ei löytynyt, mutta löysin internetin keskustelupalstalta ratkaisun pulmaan, jonka lähteen valitettavasti hukkasin. ILO3 käyttää vanhentunutta .NET frameworkia, joka ei ole oletuksella Windows 10 käyttöjärjestelmässä aktivoitu (Perkins, 2016). Samalla selvisi myös, että Microsoftin RDM toiminto ei ole yhteensopiva kuin Internet Explorer selaimessa (Perkins, 2016).

Minun tuli tehdä päätös päivitänkö moduulin firmwaren, vai asennanko vanhan .NET paketin koneelleni. Valitsin jälkimmäisen vaihtoehdon sillä perusteella, että se tuntui nopeammalta, ja mitään takuita ei toimivuudesta olisi ollut uudellakaan firmwarella.

.NET 2 ja 3 käytiin asentamassa Windowsin asetuksista navigoimalla *apps and features* kohtaan, jonka alta löytyy valinta *programs and features*. Tämän sisällä täytyi vielä valita *Turn Windows Features on or off*. Sieltä pääsemme lisäämään kaikennäköisiä ominaisuuksia Windowsille, joista löytyy myös .NET frameworkkien vanhat versiot (kuva 11). Valintojen jälkeen tietokone asentaa uudet ominaisuudet ja mahdollisesti käynnistyy uudelleen, uusien ominaisuuksien voimaantumisen varten.

Kuva 11 .NET Frameworkin lisäys

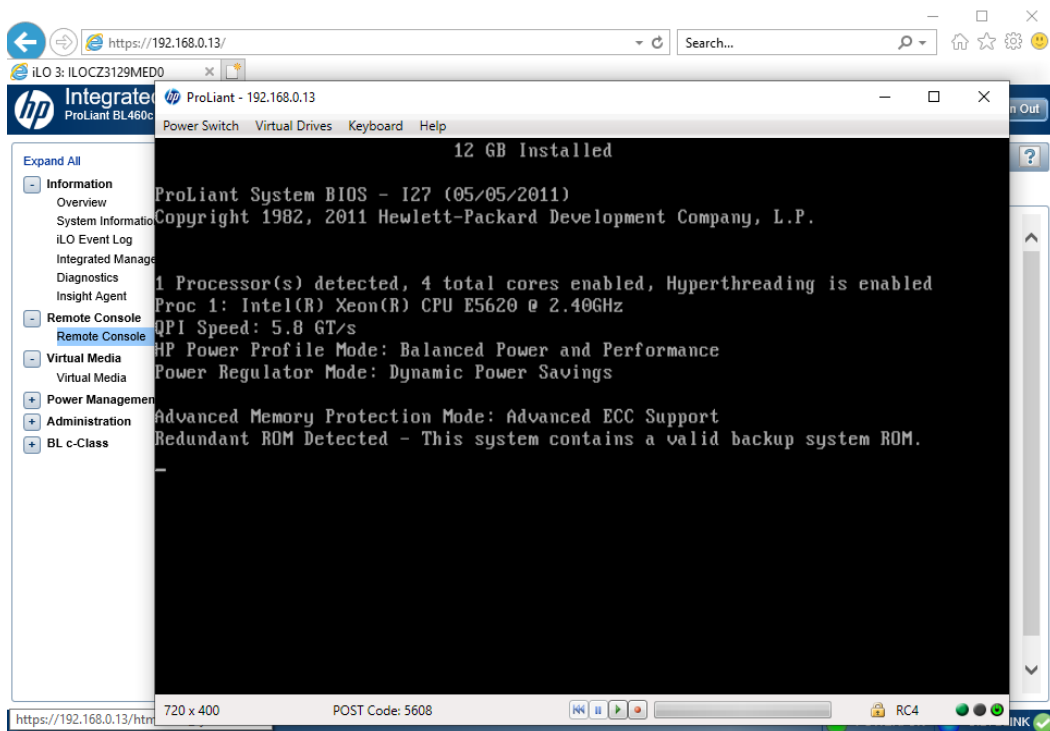


Tehtyäni tarvittavat lisäykset ja vaihdettuani käyttämään Internet Explorer selainta, etäyhteyden muodostaminen tulisi onnistua Windowsin RDM:lla.

#### 4.1.3 Windows Server 2016 asennus

Muodostettua etäyhteys koneeseen, meille avautui mahdollisuus liittää etäyhteysportaaliin (kuva 12) virtuaalinen DVD-asema, johon voimme asettaa Windows Server 2016 imagetiedoston. Näin saimme virtualisoitua vanhanaikaisen tavan asentaa käyttöjärjestelmä DVD levytä.

Kuva 12 Etäyhteysportaali iLO:n ylitse

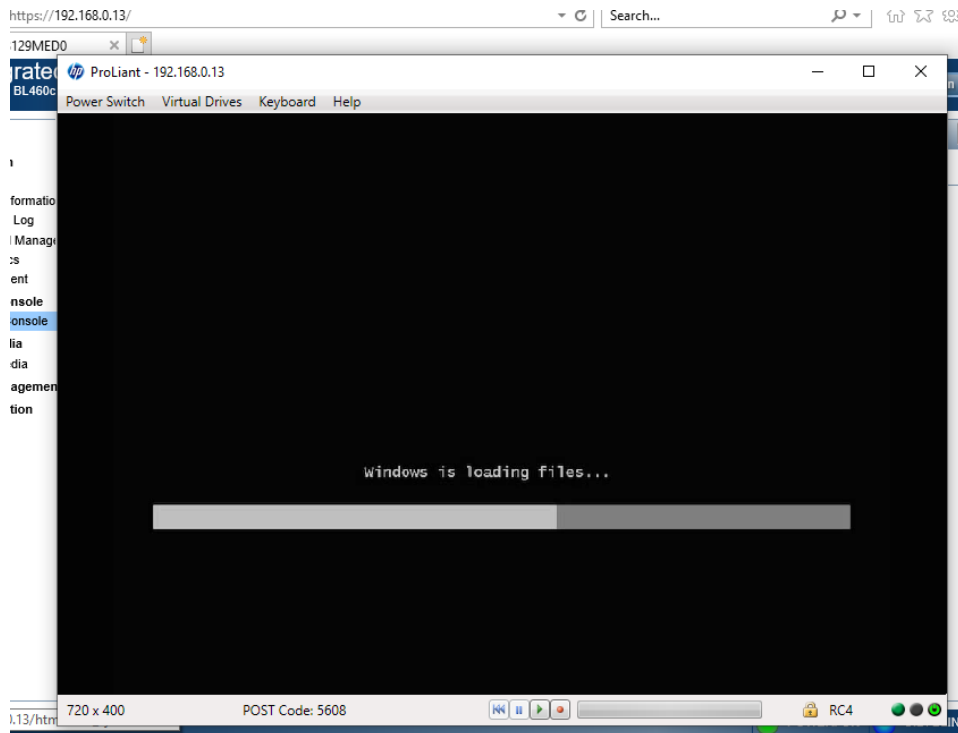


Microsoft tarjoaa eri käyttöliittymille ja niiden versioille imagetiedostoja ladattavaksi verkkosivuillaan ilmaiseksi. Käyttöjärjestelmät vaativat täydelliseen toimintaan silti lisenssin, joka täytyy Microsoftilta ostaa. Yleensä lisenssi annetaan oston yhteydessä yksityisille käyttäjille uniikkina aktivointiavaimena, joka tulee syöttää Windowsin aktivointiasetuksissa. (Rodrigues, 2015)

Keudalla on lisenssit hankittu ja Keudan lisenssillä pystyy aktivoimaan samalla aktivointiavaimella satoja eri tietokoneita. Myös imagetiedosto löytyi ISO tyyppisenä valmiiksi työn tilaajalta saamaltani kannettavalta kovalevyasemalta. Kopion ISO-tiedoston omalle tietokoneelleni, millä olin etäyhteyden palvelinkoneeseen muodostanut, ja latsin sen virtuaaliseen DVD-asemaan.

Nyt pystyimme etäportaaliin käynnistämään palvelinkoneen, mutta emme ole täysin valmiita vielä aloittamaan käyttöjärjestelmän asennusta, sillä vielä pitää tehdä pieni muutos palvelinkoneen BIOS asetuksiin. Käynnistyskohde ja –järjestys tuli asettaa, eli boot sequense. Palvelinkoneelle oli kerrottava, että käynnistyksen yhteydessä haetaan ensimmäiseksi DVD-asemalta mahdollinen käynnistyksen kohde. Kun tämä oli tehty, Pystyimme viimein aloittamaan Käyttöjärjestelmän asennuksen palvelinkoneelle (kuva 13).

Kuva 13 Onnistunut imagen lataus



Käyttöjärjestelmän asennus Windowsin asennustyökalulla oli varsin suoraviivaista ja ei poikkea tavallisesta Windowsin käyttöjärjestelmien asennuksesta juurikaan, joten en käsittele sitä tarkemmin tässä. Tärkein huomioitava seikka käyttöjärjestelmän asennuksen yhteydessä oli oikean version valinta. Pystymme halutessamme asentamaan Server 2016 käyttöjärjestelmän core version, joka on vain komentorivityökalun sisältävä käyttöliittymä. Valitsin mieluummin version, jossa on graafinen käyttöliittymä, vaikka kaikki windowsin tarjoamat palvelut ovatkin mahdollista asentaa komentorivin tekstikomennoilla. Valitsin myös englanninkielisen version omien mieltymyksieni vuoksi.

Sama prosessi täytyi tehdä kokonaisuudessaan jokaiselle palvelinkoneelle erikseen. Tämä oli näin ollen paljon aikaa vievää työtä, mutta yksi hopeareunus löytyi. Se oli, että etäyhteyden vuoksi pystyin osittain aloittamaan valmiissa asennuksissa palveluiden sekä konfigurointien asennuksen, sillä etäyhteys oli ainoa keino kiertää yhden ainoan SUV-adapterin tuomat rajoitukset. Vaikka olisin onnistunut asentamaan käyttöjärjestelmän USB-muistista, eli en olisi tarvinnut iLO:a, olisi koko palvelinkokonaisuus ollut lukittuna yhden käyttöjärjestelmän asennukseen kerrallaan. Jälkikäteen ajateltuna kaikki aika ja vaiva mikä iLO:n toimintakuntoon saattamiseen upposi, maksoi mahdollisesti itsensä takaisin.

## 4.2 Aktiivihakemisto

Palveluiden ja roolien asennus tuli aloittaa aktiivihakemistosta, koska aktiivihakemiston yhteydessä luodaan uusi toimialue. Haluamme liittää kaikki palvelimemme tähän toimialueeseen, joten loogisinta oli näin ollen aloittaa siitä.

Aktiivihakemisto, englanninkieliseltä nimeltään Active Directory, on Windows-ympäristössä toimialueen eli domainin käyttäjätietokanta ja hakemistopalvelu. Se sisältää tietoa mm. käyttäjistä, tietokoneista ja verkon resursseista kyseisessä domainissa. Se mahdollistaa keskitetyn resurssien jakamisen käyttäjille ja sen tärkein tehtävä on hallita domainin käyttäjiä ja resursseja. Se tuli uutena ominaisuutena Windows servereille 2008 versiossa. (Microsoft, 2012)

### 4.2.1 ADDS alustus

Ennen ADDS asennuksen aloitusta, sekä muutenkin hyvänä tapana juuri asennetun käyttöjärjestelmän kanssa, on tarkistaa kolme seikkaa. Ensimmäisenä kannattaa vaihtaa tietokoneen nimi. Tämä johtuu siitä, että sen muuttaminen aktiivihakemiston ja uuden domainin asennuksen jälkeen on todella hankalaa, ja haluamme tulevaa domainin hallintaa varten tunnistaa palvelimet helposti. Tunnistuksen helpottamiseksi on pidetty yleisenä tapana nimetä palvelinkone sen tulevan roolin mukaan (Nelson, 2016). Tässä tapauksessa siis rooli on Domain Controller, joten tietokoneen nimeksi saatoimme asettaa yksinkertaisesti DC1. Numero yksi sen vuoksi, että meillä voi, ja on suositeltavaa, olla useampi Domain Controller. Tämä taas siitä syystä, että ikävissä tapauksissa missä palvelin lakkaa toimimasta, esimerkiksi hajonneen komponentin vuoksi, meidän domain ei kaadu (Nelson, 2016). Tässä siis sovelletaan englanninkielistä kultaista sääntöä: *two is one, one is none* (Kim, 2016).

Toinen asia, mikä pitää tarkistaa, on aika. Aika säädetään tarkastamalla kellonaika, päivämäärä, sekä aikavyöhyke. Aikavyöhykkeen yhteydessä kannattaa tarkistaa, mikäli löytyy automaattinen kesä ja talviakiaan siirtyminen, ja miettiä tuleeko se pitää palvelinympäristössä päällä. Tärkeintä on, että kaikissa laitteissa on samat asetukset.

Windowsin domainympäristö on herkkä ajan suhteen. Kaikki domainin sisältämät laitteet tulee olla korkeintaan viiden minuutin erotuksella toisistaan. Jos eroa on enemmän, tapahtuu paljon virheitä. (Nelson, 2016)

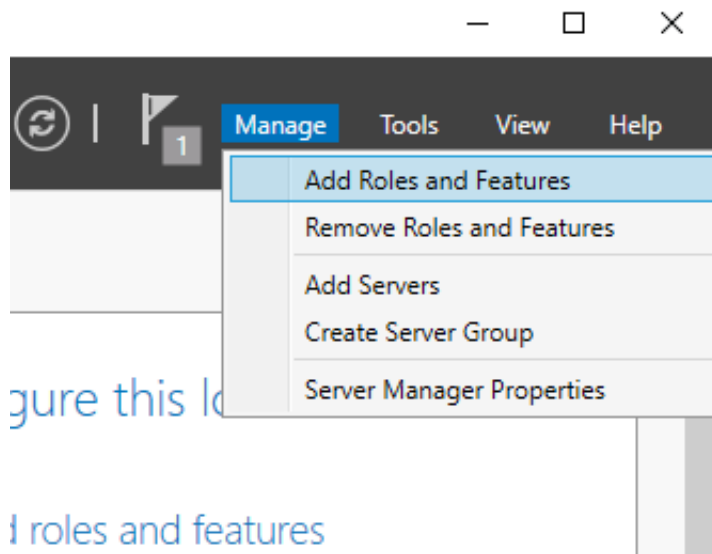
Viimeinen asetus, mikä tulee ennen ADDS asennusta tehdä, on oikeat IP asetukset. Aktiivihakemisto asennetaan palvelimen sen hetkisen IP osoitteen päälle ja IP:n vaihtaminen on tietokoneen nimen tavoin hankalaa ADDS asennuksen jälkeen. Näin ollen palvelinkoneille halutaan asettaa staattinen IP osoite, joka ei tule muuttumaan. (Nelson, 2016)

Tätä varten tuli siis selvittää IP verkko, missä palvelimet tulisi toimia ja vapaat osoitteet. Olin ne työn tilaajan kanssa selvittänyt jo aikaisemmin, joten pystyin nyt aloittamaan aktiivihakemiston, sekä uuden domainin asennuksen.

#### 4.2.2 ADDS Roolin asennus

Halusimme siis asentaa Active Directory Domain Service:n ja sen hallintatyökalut. Windows Servereille löytyy käyttöjärjestelmän yhteydessä asennettu Server Manager työkalu. Sen voi siis käytännössä olettaa olevan osa käyttöjärjestelmää. Kaikki Windowsin tarjoamat palvelut ja ominaisuudet voidaan asentaa Server Managerista käynnistettävällä *add roles and features* asennusohjelmalla, eli ns. wizardilla. Tämä aloitetaan valitsemalla Windowsin Server Managerin yläpalkissa sijaitsevasta *Manage* painikkeesta valinta *add roles and features* (kuva 14).

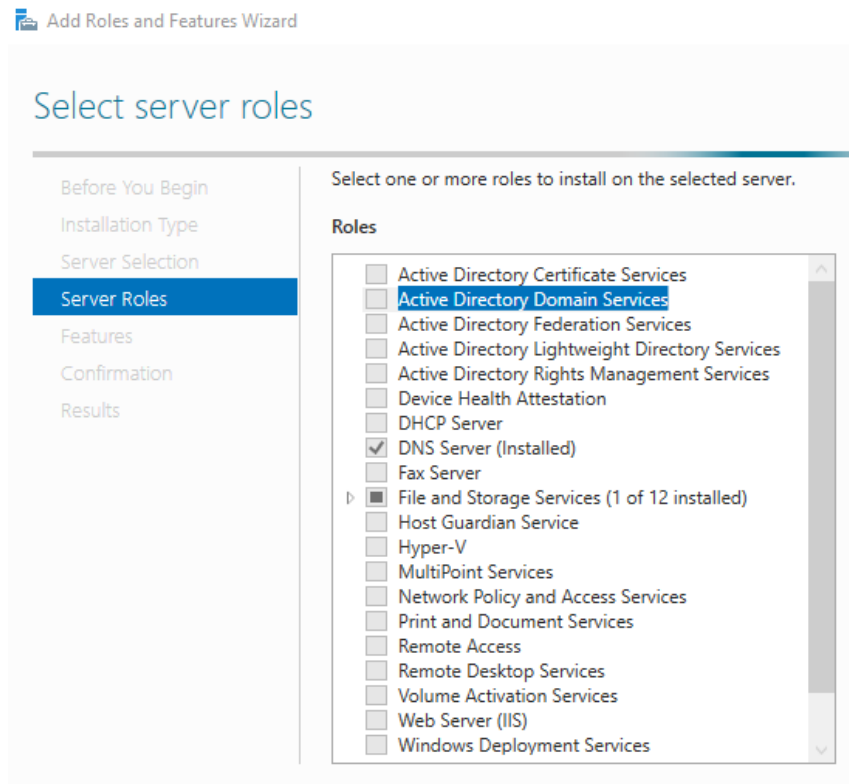
Kuva 14 Add roles and features -näkyvä



Valitsemme asennusohjelman alussa sijaitsevasta valintaikkunasta *Role or Feature based installation* vaihtoehdon, sillä olemme asentamassa uutta roolia palvelimelle. Seuraavaksi valitsemme Kohteen mihin asennamme uuden roolin. Jos palvelinta ei ole valmiiksi liitetty johonkin toimialueeseen tai palvelinryhmään, niin palvelinlistassa ei pitäisi näkyä kuin kyseessä oleva paikallinen palvelinkone. Samassa kohdassa voimme vielä tarkistaa, että tietokoneen nimi ja IP näyttävät oikeilta.

Seuraavassa valintaruudussa valitsemme asennettavan roolin. Haluamme asentaa aktiivihakemiston ja domainin, joten valitsemme oikean roolin, Active Directory Domain Services (kuva 15). Asennustyökalu osaa lisätä automaattisesti tarvittavat lisäominaisuudet, mitä ADDS tarvitsee toimiakseen kunnolla. Voimme siis mennä asennusohjelman loppuun ja käynnistää itse aktiivihakemiston asennuksen.

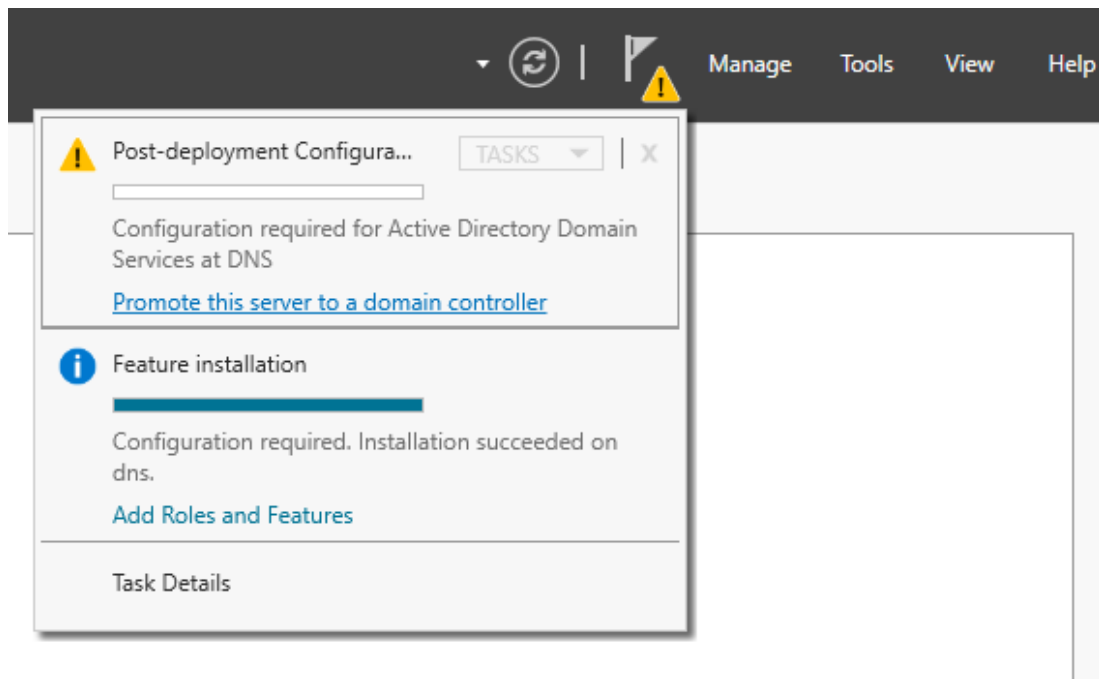
Kuva 15 ADDS roolin valinta



#### 4.2.3 Domainin muodostus ja mainostus

Kun aktiivihakemisto on asennettu, meiltä puuttuu vielä itse domain, mitä aktiivihakemisto tulee hallitsevan. Server Manager osaa kertoa käyttäjälle, että mainostus tulee tehdä erikseen (kuva 16). Haluamme muodostaa uuden domainin, ja sitä varten tarvitsemme hallintakoneen, eli domain controllerin. Meillä on tarvittavat roolit asennettu, mutta kukaan muu lähiverkossa ei sitä tiedä, ennen kuin me mainostamme sitä.

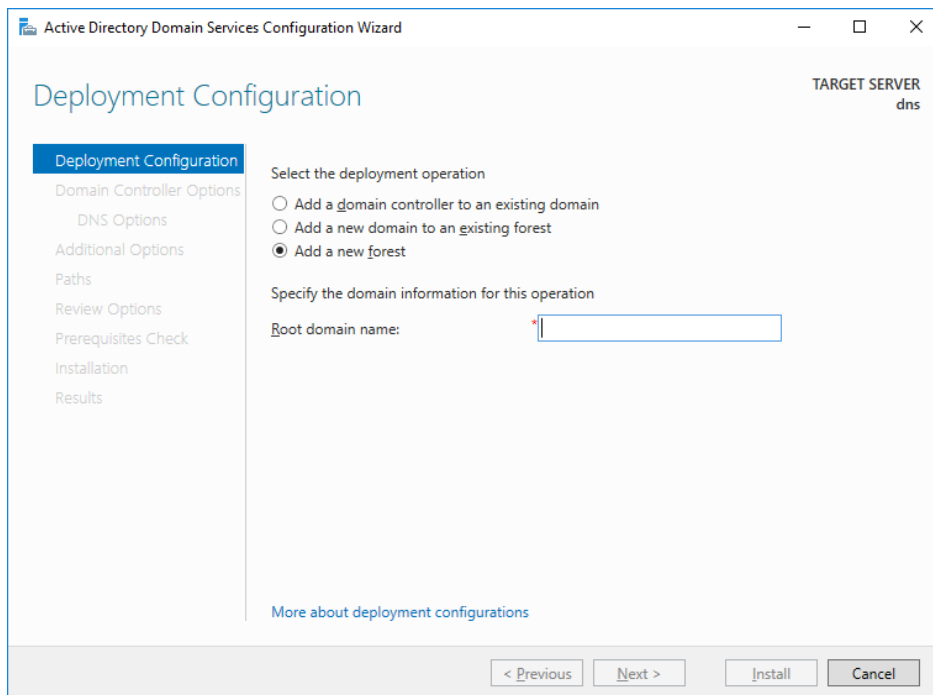
Kuva 16 Muistutus domainin mainostuksesta ja aloituslinkki.



Painamalla Managerin tarjoamaa linkkiä, joka sijaitsee muistutuksen yhteydessä, pääsemme aloittamaan aktiivihakemiston jälkiasennuksen ja domainin muodostuksen

Ensimmäisenä meidän tulee valita, miten luomme uuden domainin, sillä siihen on useampi vaihtoehto (kuva 17). Ylin vaihtoehto tulisi valita, kun lisäämme aktiivihakemiston toimialueeseen toisen hallintapalvelimen, eli Domain Controllerin. Toinen vaihtoehto on luoda uusi domain jo olemassa olevan domainin sisälle, ja kolmantena vaihtoehtona on kokonaan uuden alueen luominen.

Kuva 17 Uuden domainin muodostus



*Forest* on vain laajempi käsite toimialueesta. Tämä muodostuu siitä, että meille voi olla yhden toimialueen sisällä muita toimialueita, joiden alla vastaavasti lisää toimialueita. Näistä koostuva kokonaisuus tunnetaan nimityksellä *Tree*. Voimme myös ryhmyttää eri ylitason domaineja keskenään. Näin useampi tree muodostaa forestin. (Petters, 2020)

Meillä ei ole käytössä valmiita toimialueita, joten valitsemme kokonaan uuden forestin luomisen (kuva 17). Tämän yhteydessä meidän pitää nimetä toimialueemme juuritason nimi. Nimiehdotuksen olin saanut työn tilaajalta etukäteen ja syötin sen tekstikenttään.

Seuraavassa valintaikkunassa valitsemme domainin yhteydessä asennettavat lisätoiminnot. Tässä tarjotaan vaihtoehtoa nimenselvennyspalvelun luomiselle, DNS:lle. Aktiivihakemiston domain tarvitsee toimiakseen nimenselvennystä, joten Microsoft on järkevästi asettanut DNS palvelun asennusmahdollisuuden domainin asennuksen sisälle. Meillä voisi olla kuitenkin DNS palvelin erikseen, jolloin palvelua ei tarvitsisi asentaa DC koneelle lainkaan. Tässä työssä järkevin ratkaisu on kuitenkin asentaa DNS ADDS:n yhteydessä.

Asennusohjelma vaatii vielä samassa ruudussa asettamaan DSRM salasanan ennen pääsyä asennuksessa eteenpäin. DSRM on virhetilanteiden ilmetessä käytettävä domainin

palautusominaisuus, jota järjestelmänvalvojat eivät välttämättä tule tarvitsemaan koskaan (Petters, 2020). Tästä syystä salasana olisi hyvä dokumentoida, tai on hyvin todennäköistä, ettei ole mitään tietoa mikä salasana on, jos palautusominaisuudelle tulisi joskus tarvetta.

Seuraavassa ruudussa asennusohjelma haluaa asettaa netBIOS nimen. NetBIOS on vanhentunut kommunikointiprotokolla, millä vanhat laitteet, mitä todennäköisesti ei uusiin domaineihin koskaan liitetä, käyttävät mm. juuri domainiin liittymiseen. Asennuksen yhteydessä siis laitamme vielä varmuuden vuoksi vanhoja laitteita varten protokollan toimintavalmiuteen, ja sitä varten netBIOS tarvitsee nimen, jolle on yleensä käytäntönä antaa domainin nimen etuosa. (Barrie, 2009)

Muita asetuksia ei paljoa enää ole, ja asennusohjelman loppupuolella tehdään ensin valintojen yhteenveto normaalisti, mutta myös ajetaan valintojen validointi. Mikäli ikävän nälöisiä virheilmoituksia ei tule, Painamme asenna, ja asennuksen jälkeen palvelin käynnistyy uudelleen ja huomaamme että palvelin ei ole enää paikallispalvelin, vaan osa juuri luomaamme domainia. Kirjautumissivulla siis kirjaudumme uuteen domainiin, vaikka käyttäjätiedot ovatkin muuttumattomat ja sijitsevat paikallisesti. Aktiivihakemiston asennus on näin ollen valmis.

Tein suositusten mukaan vielä toisen domain controllerin, että palvelinkokonaisuudelle ei käy huonosti, jos ensimmäinen DC lakkaa syystä tai toisesta toimimasta. Toinen DC otta silloin ohjat käsiin. Tätä varten Aktiivihakemiston tietokanta tulee päivittää DC koneiden välillä ahkeralla tahdilla. (Tonias, 2018)

Ensimmäiseksi uudelle palvelimelle tuli asentaa ADDS rooli, ja tämä kävi samoin kuin ensimmäisellä kerrallakin. Asennus eroaa vasta mainostuksen luomisessa, sillä nyt meillä on valmis domain jo olemassa, emmekä halua tehdä toista toimialuetta, vaan lisätä toisen hallintalaitteen jo olemassa olevaan aktiivihakemiston toimialueeseen. Valitsemme siis näkyvästä valintaruudussa ylimmän vaihtoehdon, *Add a domain controller to an existing domain* (kuva 17).

Samassa ruudussa syötetään kohdenimi domainille, johon mielimme asentaa tämän uuden hallintayksikön. Domainiin liittyminen tapahtuu tämän yhteydessä. Domainiin liittyessä

tarvitaan järjestelmänvalvojanoikeus tasoisen tunnistautumisen domainiin. Syötämme siis Ensimmäisen Domain Controllerin Administrator tunnukset, joka samalla on toistaiseksi ainoa käyttäjä Aktiivihakemistossamme.

Viimeinen valinta, mikä eroaa ensimmäisen DC:n asennuksesta, on kohde mistä kopioidaan yleisiä, kopiontia edellyttäviä domainin tietoja. Meillä ei ole kuin yksi DC entuudestaan, joten ainoa vaihtoehtomme on valita kyseinen palvelin.

Lopussa valitsemme asenna, ja asennuksen mentyä loppuun meillä on toimiva domain kahdella hallintakoneella. Myöhemmin tulisi tietysti lisätä kaikki käyttäjät mitä haluamme aktiivihakemistoon, mutta toistaiseksi Aktiivihakemiston asennus oli valmis.

#### 4.3 Klusterointi

Lopuille palvelinkoneille olin suunnitteluvaiheessa päättänyt muodostaa klusterit, sillä työn tilaajan tarpeisiin nähden tarvitsimme palvelimille tulevien palvelujen tarjoamiseen klusteroinnin tuomia hyötyjä.

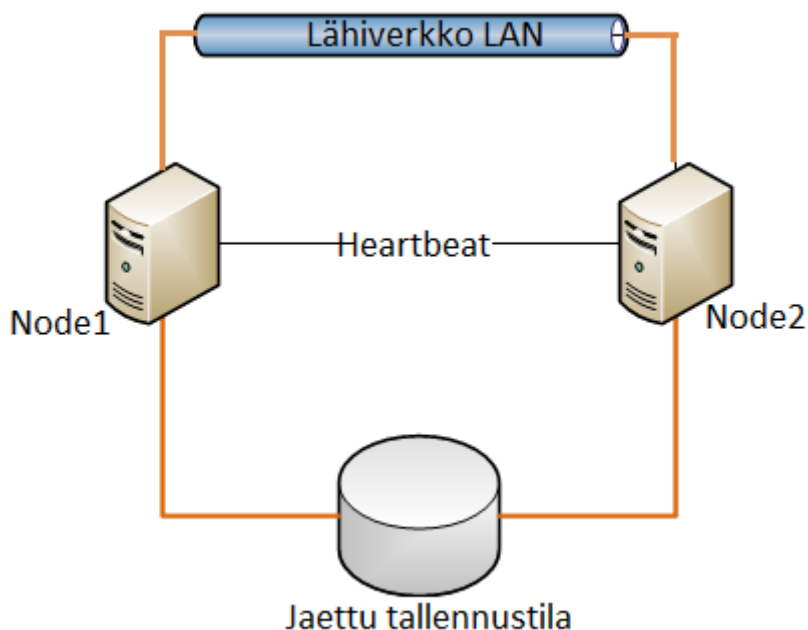
Klusterilla viitataan tietotekniikassa useamman laitteen, yleensä tietokoneen, liittämistä toisiinsa lähiverkossa, sillä ajatuksella, että useamman koneen ryhmästä, siis klusterista, muodostetaan yksi kokonaisuus. Ikään kuin muodostaisimme usean laitteen yhteistyöllä yhden laitteen. Yksittäiset, ominaisuuksiltaan samankaltaiset laitteet tunnetaan nimityksellä node. (Pfister, 1998)

Klusteroinnilla saavutetaan kaksi suurta hyötyä, hyvä vikasietoisuus sekä suurempi laskentateho. Suuri laskentateho perustuu siihen, että klusteri pystyy hyödyntämään kaikkien siinä olevien laitteiden omaa suorituskykyä keskitettynä samaan tehtävään. Hyvä vikasietoisuus taas tulee siitä, että klusteri ei tarvitse kaikkia siinä olevia nodeja pysyäkseen toimintakuntoisena. Jos siis jokin node esimerkiksi hajoaa, hyvin asennettu klusteri pysyy täysin toimintakykyisenä, menettäen vain hieman suorituskykyään. (Pfister, 1998)

Klusterointi itsessään on jo vanha idea, mutta se levisi suureen käyttöön vasta internetin mukana tulleiden palvelintarpeiden myötä, ja on ollut jatkuvasti kehittyvää tähän päivään asti. (Pfister, 1998)

Windows Serverin tarjoama Failover clustering ominaisuus toimii teoriassa kuvan 18 esittämällä tavalla. Meillä on yksi päänode, joka seuraa jatkuvasti klusterin muita nodeja kuuntelemalla tietokoneiden heartbeat signaalia. Heartbeat, eli sydämenlyönti, on siis sähköinen signaalipulssi, minkä lähettämällä nodet ilmoittavat olevansa toimintakunnossa. Jos pulssia ei kuulu tarpeeksi pitkään aikaan, päänode tekee oletuksen, että kyseinen node on pudonnut pois pelistä, eli ns. kuollut, ja nodella olleet tehtävänannot jaetaan loppujen nodejen kesken uudelleen. Mikäli päänode lakkaa toimimasta, failover klusterin ominaisuuksiin kuuluu uuden väliaikaisen päänoden määrittäminen, ja se määritellään nodejen sen hetkisen kuorman mukaan (Microsoft, 2018).

Kuva 18 Failover clusterin kytkentäperiaate



Klusterin muodostukseen suositellaan normaalisti jaettua tallennustilaa, erillisessä, vain siihen tarkoitukseen omistetussa verkossa, joka tunnetaan nimellä Storage Area Network, SAN. Se rakennetaan tavanomaisesti laadukkaista ja nopeista komponenteista. Klusterit siis

käyttäisivät tätä verkkoa ja siinä sijaitsevaa tallennustilaa jakamaan tiedostoja, keskustelemaan ja muutenkin toimimaan keskenään. (Bauer, 2018)

Minulla ei ollut laadukkaan SAN:n rakentamiseen vaadittavia komponentteja käytettävissäni, joten suositusten vastaisesti olin suunnitellut rakentaa jaetun tallennustilan samaan verkkoon kaikkien muiden palvelinympäristön palikoiden kanssa. Tämä ei ole hyvä käytäntö, mutta työn tilaajan kanssa sovimme, että ratkaisu on silti käyttötärpeisiimme siedettävä.

Teimme jaetun tallennustilan SCSI:lla. SCSI eli Small Computer System Interface on standardi tiedon välittämiseksi tietokoneiden ja mahdollisten oheislaitteiden välillä. Sitä varten pitää koneessa olla SCSI-ohjain, ja vastaavasti mahdollisissa oheislaitteissa tulisi olla oma SCSI-kontrollerinsa, joka hoitaa väylälle liikennöinnin. Teoriassa SCSI laite voisi olla mikä vain sillä standardi on täysin laiteriippumaton. (Wikipedia, n.d)

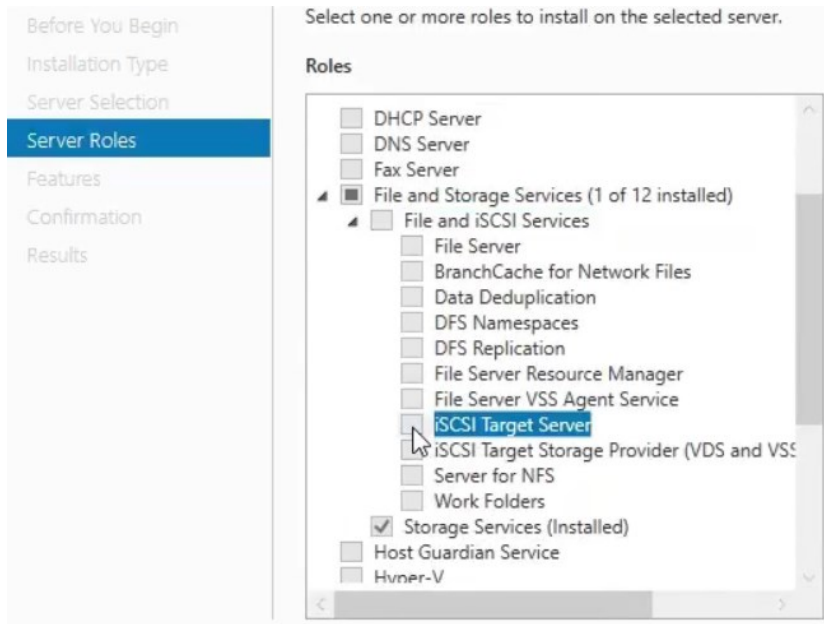
Kaikki palvelinkoneemme ja kovalevyimme ovat SCSI yhteensopivia, kiintolevyimme tarkemmin ottaen ovat SAS-kiintolevyjä, eli sarjaväyläisiä SCSI kiintolevyjä. Suuremmat SAS-kiintolevyt hankittiin juuri tästä syystä, että saamme Laadukkaan NAS:n luotua iSCSI avulla, eli internet SCSI:n.

#### 4.3.1 iSCSI muodostus

Käytämme iSCSI:n muodostamiseen samoja palvelimia, mitkä toimivat Domain Controllereina. Ensimmäisessä suunnitelmassani iSCSI Tiedostopalvelin olisi omistettu vain sitä tarkoitusta varten, mutta ymmärsin pian, että muiden puutteiden vuoksi en olisi siitä saanut juurikaan hyötyä. Siksi ajattelin, että DC koneet voisivat ottaa tehtävän, sillä näin pienen ADDS:n hallinnointi on kevyttä ja passiivista, joten koneet saivat nyt verkkolevyn hallinnan vielä hoidettavakseen.

iSCSI:n muodostus aloitetaan, kuten mikä tahansa muukin rooli, Server Managerin *add roles and features* valinnasta (kuva 14). Valitsemme *role or feature based installation*, ja itse rooliksi *iSCSI Target server* (kuva 19). Muita valintoja ei tule tehtäväksi roolin asennuksessa.

Kuva 19 iSCSI Palvelinrooli



Roolin asennuksen jälkeen navigoimme Server Managerissa *File and Storage Service* kohtaan ja avaamme sen alta löytyvän iSCSI valinnan. iSCSI:n sisältä löydämme kaikki iSCSI:a käyttävät levyasemat, jotka palvelumme sisältää. Me luomme vasta ensimmäistä, joten emme näe ikkunassa muuta valintaa kuin uuden levyn luominen (kuva 20).

Kuva 20 iSCSI Levyn luominen



Server Blade -koneissa on kaksi paikkaa SAS-kiintolevyille. Asetamme DC-koneiden toiseen paikkaan hankkimamme isommat kiintolevyt, jotka omistamme täysin iSCSI:lle. Valitsemme siis levyn kohteeksi koko levyn kattavan Volumen. Näin ollen emme myöskään tarvitse kasvuasetuksia, sillä toistaiseksi meillä ei ole tarjota kasvumahdollisuutta.

Jos meillä olisi entuudestaan iSCSI kohde olemassa, voisimme seuraavaksi uuden levyn asettaa valmiiseen iSCSI kohteeseen, mutta luomme ensimmäistä levyä, joten haluamme myös luoda uuden kohteen.

Seuraavaksi meillä olisi tilaisuus asettaa käyttäjäkohteita. Emme ole vielä asentaneet klusteria, joka tulee aseman ainoaksi jäseneksi mitä todennäköisimmin, joten emme toistaiseksi lisää jäseniä, teemme vain tyhjän iSCSI levyn. Asennuksen pystyi nyt viemään loppuun ja uusi levymme ilmestyy asennuksen jälkeen *File and Storage* palvelun alle näkyviin.

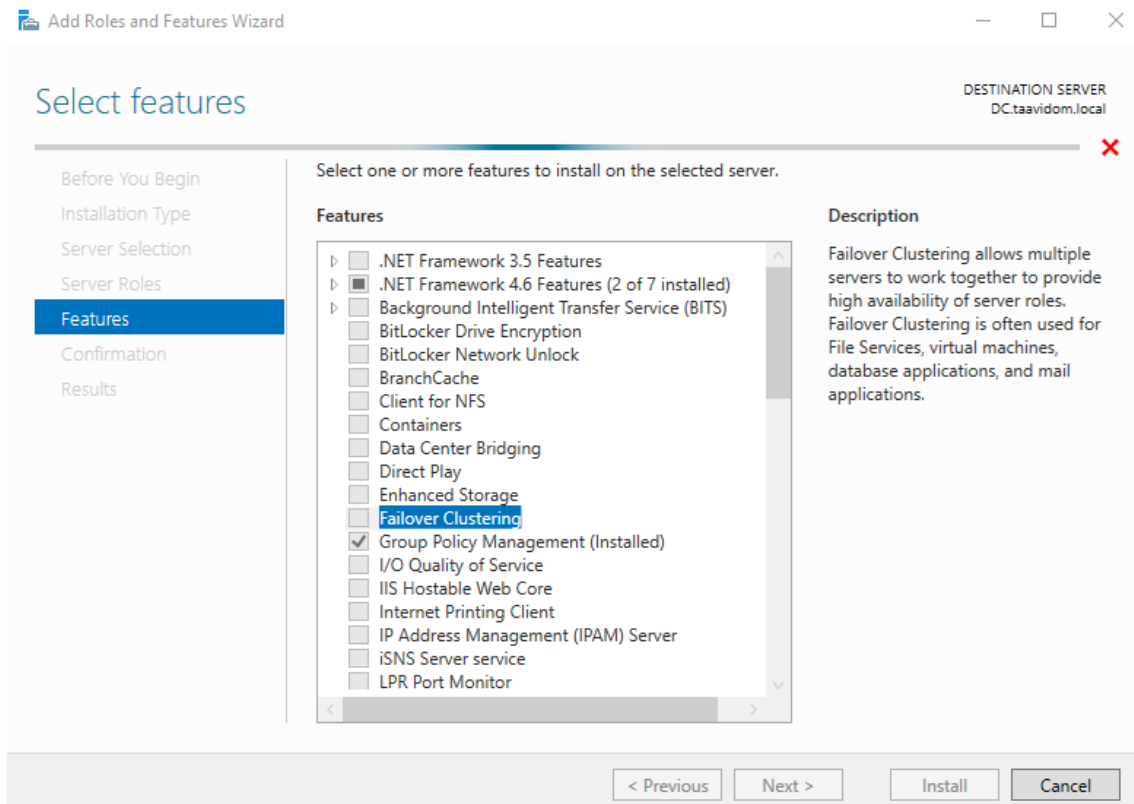
#### 4.3.2 Klusterin muodostus

Itse klusterin muodostus on onnistuttu tekemään melko suoraviivaiseksi ja helpoksi nykyaikaisilla käyttöjärjestelmillä, kuten työssä käytettävässä Windows Server 2016 versiossa, kunhan laitteisto vastaa klusterin muodostamisen edellytykset. Tärkein edellytys on nopea yhteys nodejen välille (Microsoft, 2018). Gigabit Ethernetiä hitaammilla yhteyksillä klusteria ei pystynyt muodostamaan. Sain huomata tämän Kokeillessani asennusta HP ProCurve kytkimellä, ja se oli työn tämä vaihe, kun kyseinen kytkin jäi kokonaan pois.

Uuden klusterin muodostuksessa kannattaa asennus tehdä mahdollisimman nopeasti jokaiselle tietokoneelle, joka halutaan asettaa klusterin nodeksi. Mitä identtisempää laitteisto ja ohjelmisto nodejen välillä on, sitä varmemmin klusterin muodostus onnistuu (Kameshwaran, 2017). Jos Yrität muodostaa klusteria tietokoneiden välille, joilla on eri ohjelmistoversio tai erilaisia ohjelmia asennettu, saattaa synkronisoinnin kanssa tulla haasteita.

Kuten kaikki muutkin roolit, klusterin asennus aloitetaan Windowsin Server Managerissa valitsemalla *add roles and features* (kuva 14). Failover cluster ei ole rooli, vaan ominaisuus eli feature. Siksi navigoimme wizardissa *add features* kohtaan suoraan ja valitsemme Failover clustering ominaisuuden (kuva 21).

Kuva 21 Failover Clustering - featuren lisäys



Tämä tulee tehdä jokaiselle koneelle, mikä halutaan asettaa klusteriin. Tein siis asennuksen jokaiselle palvelinkoneelle, paitsi DC koneille, vaikka kaikkia koneita ei asetetakaan samaan klusteriin. Muita asetuksia ei featuren asennuksessa tehdä, joten valitsimme *install feature*.

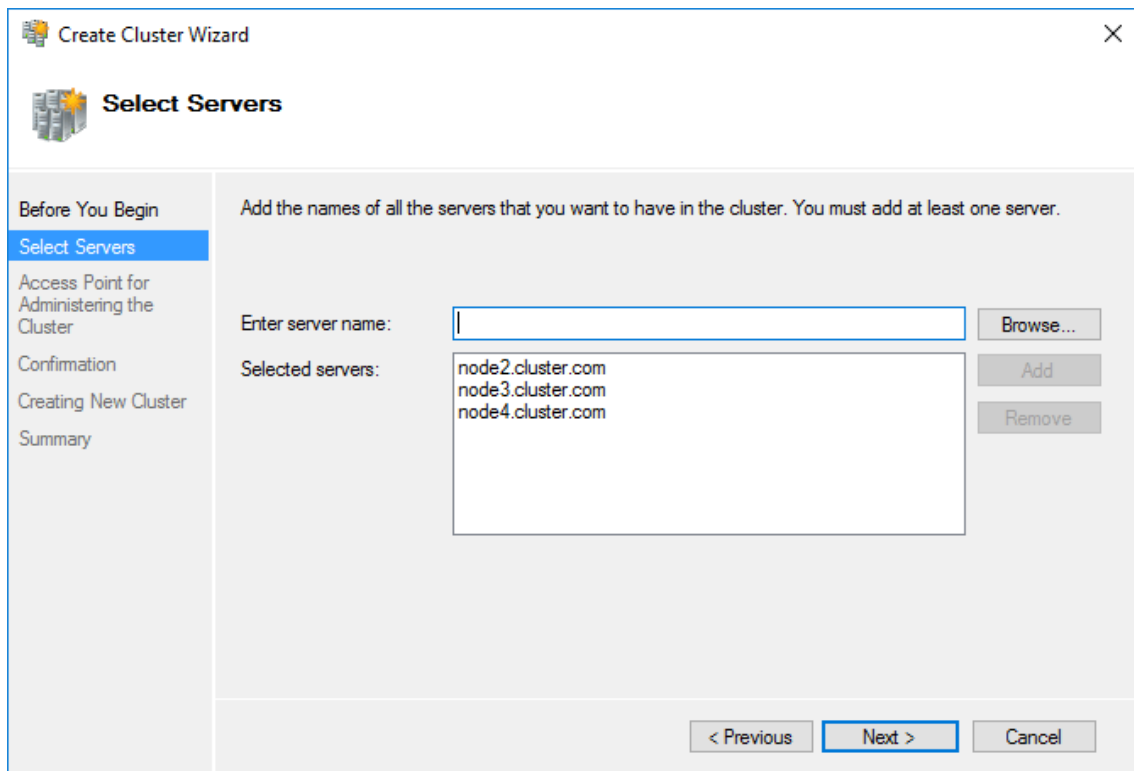
Näin helposti oli klusterointiominaisuus käytössä, mutta ennen kuin aloitin klusterin muodostamisen, liitin kaikki palvelinkoneet, jotka tulevat klusterin nodeksi, aiemmin tekemääni iSCSI levyyn käyttäjiksi. Se onnistuu Server 2016 käyttöjärjestelmän Server managerin tools valintojen alta. Valitsimme sieltä iSCSI initiator. Se avaa meille uuden wizardin, jolla liitetään kone iSCSI:n käyttäjäksi. Annamme kohdeserverin missä iSCSI sijaitsee ja asennusohjelman pitäisi osata kertoa, että kohteesta löytyy tekemämme iSCSI levy. Viemme wizardin loppuun, mutta ennen kuin iSCSI on todellisuudessa käytössä, pitää iSCSI sijainnissa vielä käydä aktivoimassa käyttäjät. Kun käyttäjät ovat aktivoitu ja iSCSI toimii, olemme lähes valmiita aloittamaan klusterin muodostamisen.

Sitä ennen palvelinkoneet, jotka halutaan liittää klusteriin, tulee liittää samaan domainiin. Domainiin liittäminen tapahtuu yksinkertaisesti. Liitettävässä koneessa tulee navigoida

järjestelmän asetuksiin, sen alta valitaan turvallisuusasetukset, eli *system and security*. Tämän alta navigoidaan vielä *about system* kohtaan, jonka alta löytyvät vaihtoehdot tietokoneen nimen vaihdolle, ja työryhmiin tai domainiin liittymiselle. Valittaessa domainiin liittyminen, aukeaa liittymisikkuna, jolle ensin kerrotaan kohde, missä domain sijaitse. Minun tapauksessani piti antaa toinen kahdesta domain controllerista, joilla ADDS palvelu on asennettu. Täytyy myös muistaa, että jos koneen FQDN eli nimi domainin sisällä toimi, tulee tarkastaa liitettävän koneen DNS asetukset. Myös IP osoitteella voi yrittää liittyä, minkä pitäisi toimia. Tämän jälkeen Pyydetään järjestelmänvalvojan oikeuden omaavan käyttäjän tunnistautuminen. Käyttäjän tulee olla DC koneen käyttäjä. Kun tunnistautuminen onnistuu, liitetään kone domainiin, jonka yhteydessä tietokone käynnistyy uudeleen itsestään ja käynnistyksen jälkeen, tulee kirjautumisruudussa näkyviin, että paikallisen koneen sijaan kirjaututaan domainiin.

Kun kaikki palvelinkoneet ovat liitetty domainiin, aloitamme klusterin muodostuksen valitsemalla Server Managerin toolsin alta Failover Cluster Manager. Managerissa emme näe vielä yhtään klusteria, vaan tulee sellainen nyt muodostaa. Valitsemme *Create new cluster*, joka avaa jälleen uuden wizardin klusterin muodostukseen. Ensimmäisessä valintaruudussa syötämme kaikkien nodejen FQDN nimen tai IP osoitteen (kuva 22).

Kuva 22 Nodejen lisäys klusterin muodostuksessa (Kameshwaran, 2017)



Seuraavassa kohdassa valitsemme klusterille nimen, sekä IP osoitteen. Nimi kannattaa valita jälleen sen mukaan, mikä tulisi olemaan klusterin rooli. Ensimmäisen klusterini tarkoitus oli siis toimia webalustana Videoiden ja streamien katseluun. Nimesin näin ollen klusterini webcluster nimiseksi. IP osoite, mikä klusterille syötetään, tulee siis olemaan IP osoite, mihin kaikki tuleva liikenne ohjataan. Klusteri jakaa tulevan liikenteen tämän jälkeen klusterin sisäisesti (Kameshwaran, 2017). Sama IP osoite on siis osoite mihin tulevalle websivustolle pyrkivät käyttäjät hakeutuvat. Halusin siis tehdä samalla merkinnän DNS palvelimelle rekisteritiedon klusterin IP:lle, joka tulisi olemaan websivuni nimi. Olimme työn tilaajan kanssa päättäneet, että sivu tulisi olla opiskelijoille helposti muistettava, joten tein nimenselvennyksen nimelle *keuda.mediaserveri*.

Viimeiseksi tehdään yhteenveto ja wizard juoksee tarkastuksen. Mikäli ei vakavia virheilmoituksia tule, voidaan wizard ajaa loppuun ja klusteri muodostaa. (Kameshwaran, 2017)

Klusteri on siis nyt muodostettu, mutta sille tulee vielä asettaa tekemämme iSCSI levy käyttöön. Se onnistuu Failover Cluster Managerissa löytyvästä vaihtoehdosta *Add Disk*

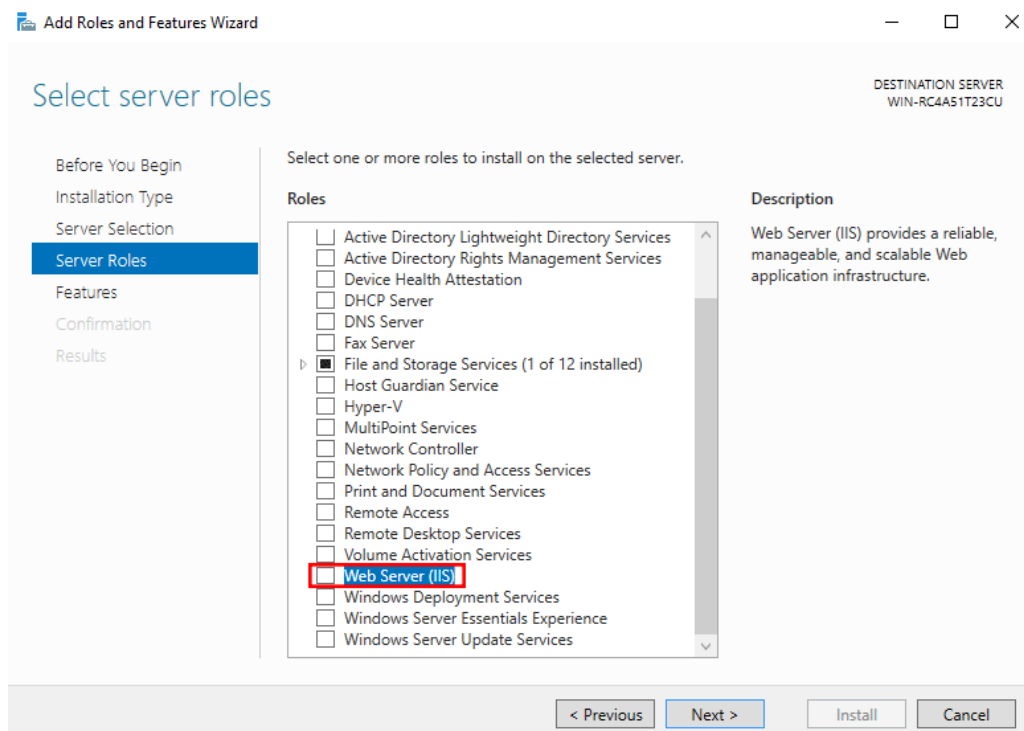
valinnasta. Mikäli aikaisemmin iSCSI:iin liittyminen onnistui, pitäisi iSCSI levyn olla näkyvässä valintana. Kun levy on valittu, toimii klusteri halutulla tavalla. Tästä eteenpäin tehdessä muutoksia johonkin nodeen, ne replikoituvat muihin nodeihin automaattisesti.

Toinen klusteri muodostettiin lopuista koneista samalla lailla, kuitenkin sillä erotuksella, että teimme streamien tallennusta varten toisen iSCSI levyn, johon adaptiiviset streamit tallennetaan. Se on siis erillisellä levyllä muista Web serverin videoista, joten web serverille täytyi myöhemmin käydä asettamassa pääsy streamiklusterin iSCSI levyyn.

#### 4.4 IIS

Seuraavaksi oli vuorossa Web palvelimen asennus. Toimiessamme Windows Server ympäristössä, meillä on sen luontiin olemassa varsin hyvä palvelu nimeltä Internet Information Services Windowsilla. Sen asennus palvelimelle aloitetaan, kuten muutkin palvelut (kuva 14). Valitsemme roolin *Web Server (IIS)* ja sen yhteydessä automaattisesti valittavat ominaisuudet (kuva 23).

Kuva 23 IIS roolin asennus



IIS:n yhteydessä on paljon hyödyllisiä lisäpalikoita, joista en tiennyt vielä, tulisinko tarvitsemaan niistä jotain. Onneksi uusien osien lisääminen käy yhtä vaivattomasti, kuin muidenkin Windowsin palvelujen asentaminen, joten jätimme tällä kertaa valitsematta enempää ominaisuuksia IIS:lle.

Menemme asennusohjelman loppuun ja valitsemme *install*. Asennus kesti kohdallani yllättävän pitkään, mutta en huomannut sen toiminnassa jälkeinpäin tarkasteltuna mitään erikoista. Arvelen, että hitaus johtui klusterista, vaikka en siitä mitään virallista tietoa löytänytään.

IIS:n Web palvelin toimii seuraavalla tavalla. Käytössämme on ensinnäkin ominaisuus nimeltä Application pool. Se on käytännössä Windowsille rakennettava prosessi, joka sisältää web serverin käyttämät ohjelmat. Itse Web sivu on vain yksi ohjelma Application poolin sisällä. Tämän takia Windowsin IIS Managerissa nämä kaksi asiaa on eroteltu toisistaan. Meillä on mahdollisuus muodostaa samalla palvelimella useampia websivuja, ja niiden asentamisen voi suunnitella tarpeisiinsa nähden monella tavalla. Esimerkiksi jos websivujen toiminnallisuus on hyvin samankaltainen toisen sivun kanssa, voimme pistää samaan Application pooliin molemmat sivut, jotka käyttävät siis samaa isäntäprosessia, vaikka ovatkin erilliset sivut. Monesti kuitenkin haluamme luoda uuden Application poolin uuden websivun yhteydessä, koska näin voidaan muut websivun hyödyntämät ohjelmistot pitää erillään muista prosesseista. Tällä on monia hyötyjä, mutta suurimpana hyötynä on varmasti se, jos jokin sovellus kaatuu, niin se ei kaada mukanaan koko web palvelinta, vaan vain sen prosessin minkä sisällä se toimi. (Microsoft, 2016)

Minulla ei suunnitelmassa ole kuin yhden websivun luominen, joten pystyimme lähteä kehittämään sitä IIS:n yhteydessä luoduilla oletus Application poolilla ja websivulla.

## 5 PALVELUJEN ASENNUS JA KEHITYS

Ympäristö oli nyt toimintavalmiina palvelujen kehitystä varten. Olin päättänyt, että palvelujen keskustana toimii websivusto, josta pääsee käsiksi niin videoiden ja streamien katseluun, sekä streamien aloittamiseen. Websivujen kehitys oli pitkäkestoista ja epäsuoraa, ja sen kehittäminen tapahtui sivuavasti kaikkien muiden palvelujen ja tekniikoiden käyttöönoton yhteydessä.

Käyn tässä kappaleessa ensin läpi tekniikat, joilla webalusta kehitettiin, ja sovellukset, joita se käyttää. Tämän jälkeen käyn läpi streamimoottorin toiminnan asennuksineen ja viimeiseksi käyn läpi websivuston käyttöliittymän, sekä tavat millä websivun osat ja toiminta toteutettiin

### 5.1 Sovelletut tekniikat websivustolla

#### 5.1.1 HTML5 ja CSS3

Websivun pohjana toimii aina esityskieli. Esityskieli on tapa kertoa selaimelle, jolla sivuja tarkastellaan, miten annetut tekstit ja muut osat tulisi sivulla näyttää. Hyper Text Markup language, siis HTML, on käytännössä ainoa esityskieli mitä internetissä käytetään, mutta siitä on aikojen saatossa ollut useita eri versioita. Tämän hetken uusin ja yleisesti parhaana pidetty versio on nimeltään HTML5. (MDN docs, n.d)

Cascading Style Sheet, eli CSS on puolestaan tapa tyyllitellä HTML tekstiä ja elementtejä. Sillä siis muokataan HTML:lla muodostettua kokonaisuutta halutun näköiseksi, esimerkiksi elementtien kokoon, väriin, sijaintiin sekä muihin ulkonäöllisiin seikkoihin vaikuttaen. CSS versioita on ollut muutama ja uusin, monipuolisin, sekä ominaisuuksiltaan rikkain on nykyinen CSS3, joka on HTML5:n kanssa valloittanut internetin kaikkien sivustojen markkinaosuutta huimasti (MDN docs, n.d -a). Oli siis helppo valinta millä versioilla lähden sivujani työstämään.

HTML muodostuu kokonaisuudessaan niin sanotuista tageista. Kaikki sisältö, mitä HTML:lla tuotetaan, kääritään tagien sisään, joitain poikkeuksia lukuun ottamatta. Nämä poikkeudet

ovat silti tageja itsessään. HTML dokumentti koostuu yksinkertaisimmillaan dokumenttityypin määrittelystä, sekä html tageista. Kuitenkin käytännössä se vaatii lisäksi vähintään head ja body tagit. Headin sisään asetetaan kaikki tiedot mitä sivulle annetaan sen toiminnallisuuden edistämiseksi, joita ei haluta webkäyttäjälle esittää. Kyse on ns. metadatasta. Bodyn sisään sitten luodaan itse websivun sisältö. (MDN docs, n.d -a)

### 5.1.2 JavaScript

Pystymme luomaan sisältöä ja muokkaamaan sen ulkonäköä HTML ja CSS kielillä. Mitä emme pysty toteuttamaan niillä, on itse toiminnallisuus sivuilla. HTML:n ominaisuudet ovat hyvin rajoittuneet, joten tarvitsemme avuksi tämän tasoisessa websivustossa jonkin toiminnallisuuden antavan työkalun. Tätä varten webkehittäjillä on JavaScript ohjelmointikieli.

JavaScript on juuri webkehitykseen tehty ohjelmointikieli. Jälleen kerran lähes kaikki suuremmat sivustot käyttävät JavaScriptia luomaan toiminnallisuutta sivuillaan (MDN docs, n.d -b). Tulin tarvitsemaan työssäni JavaScriptia mm. Videosoitin asetuksissa ja HTML elementtien muokkaamisessa tarpeiden mukaisesti, esim. piilottamisessa, kun käyttäjän ei niitä tarvitse nähdä, sekä toimimaan joissain tapauksina tulkkina palvelimen skriptikielelle.

JavaScriptin vahvuus ja heikkous tulevat samasta asiasta. JavaScript on niin monipuolinen ja muokkauskykyinen, että sillä pystyy toteuttamaan lähes kaiken mitä mieleen vain juolahtaa. Samasta syystä JavaScriptin käyttö voi olla kuitenkin välillä hyvinkin hankalaa. Liian hyvän taipuisuuden ja virheensietokykyensä vuoksi, joskus tulee luotua sitä käyttäen ongelmia, joita itse kieli ei edes tiedosta ongelmiksi. Itse olen todella tykästynyt kieleen, mutta monet myös inhoavat sitä tästä syystä.

JavaScriptin toinen suuri vahvuus on sen olemassa olevien erilaisten kirjastojen lukemattomat määrät, joilla sen toiminnallisuutta vielä parannetaan. Yksi tunnetuimmista kirjastoista kulkee nimellä jQuery. Se on kehitetty päätavoitteenaan helpottaa ja nopeuttaa ohjelmointia tarjoamalla tuhansia valmiita funktioita ja oikoteitä (Hodge, 2017). En itse asentanut työhön ylimääräisiä kirjastoja, sillä websivustoni olivat loppupeleissä melko suoraviivaiset, joten pärjäsinkin puhtaalla JavaScriptillä.

### 5.1.3 PHP

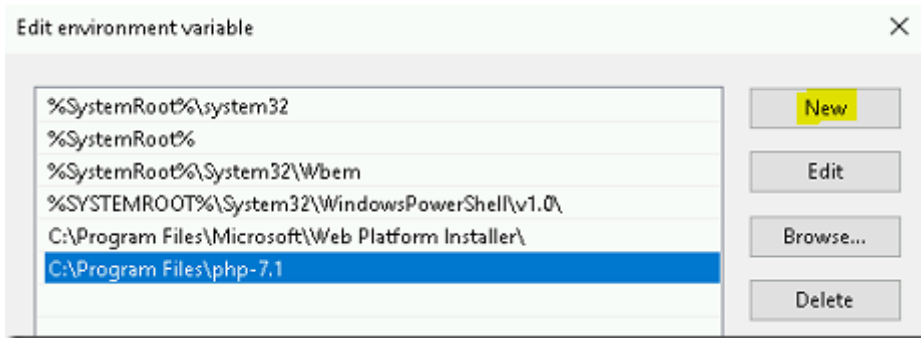
Pystyäksemme suorittamaan toimintoja ja komentoja palvelimella, tarvitsimme käyttöömmemme myös jonkin tyyppisen skriptauskielen. Valitsin kieleksi PHP:n seuraavin perustein. PHP on hyvin suosittu kieli palvelinympäristössä, lähes kaikki vuorovaikutusta tarvitsevat websivut hyödyntävät joko Microsoftin ASP- tai PHP-kieltä (Bhagat, 2019). PHP:n käyttöön löytää siis varmasti paljon ohjeita ja neuvoja. Toisena perusteena oli, että itselläni oli jo hieman kokemusta siitä valmiiksi. Tästä syystä valitsin PHP:n, vaikka toiminkin Microsoftin ohjelmistossa, ja ASP olisi ollut parilla klikkauksella käytössä palvelimilla.

PHP tulee asentaa erikseen palvelimelle. Sen eri versiot löytyvät PHPn virallisilta verkkosivuilta. Valitsin version, joka oli työn tekohetkellä uusi, mutta kuitenkin tarpeeksi vanha. Uusissa versioissa saattaa olla uutuuden ongelmia, sekä mitä vanhempi versio, sitä enemmän sillä on käyttäjien tarjoamaa tukea (perspectives, 2012). Päädyn version 7.2 valintaan. Se ladataan kokonaisuudessaan pakatussa kansiossa, jonka siirsin jälleen USB-muistilla palvelinkoneelle, missä kansio puretaan. PHP:n ja Windowsin asetuksia piti silti säätää, että saadaan PHP kytkettyä käyttöön.

Ensimmäisenä piti asentaa PHP:n lisäosa, PHP Wincache. Wincache on ominaisuus, jolla PHP pystyy hyödyntämään Windowsin välimuistia, nopeuttaen toimintaansa huomattavasti. Sen sai ladattua Microsoftin omilta sivuilta hakemalla PHP wincache hakusanalla. Tulee valita yhtenevä versio lisäosan ja PHP:n kanssa. Kun se oli ladattu, tulee vielä purkaa se oikeaan sijaintiin, joka on PHP:n juurihakemiston alla oleva hakemisto lisäosille, joka on nimetty *ext* nimiseksi. Vielä täytyi muokata PHP asetustiedostoa *php.ini* sen juurihakemistossa siten, että PHP ymmärtää käyttäjä lisäosaansa. Ini tiedostoon lisätään rivi *extension = php\_wincache.dll*. (Loges, 2017)

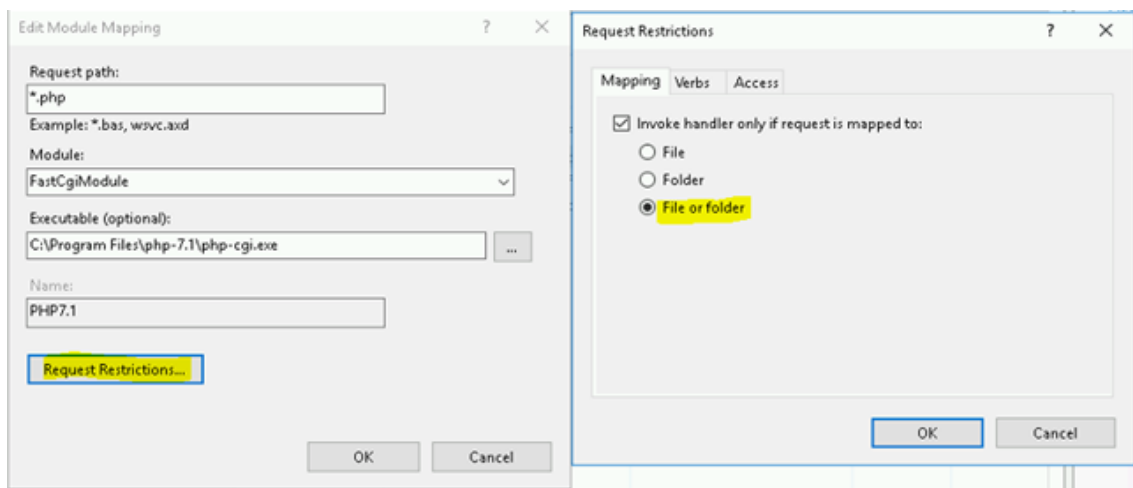
Seuraavaksi tuli kertoa käyttöjärjestelmälle, että PHP kuuluu yleismuuttujiin tässä windows käyttöjärjestelmässä. Se onnistuu Menemällä edityneisiin järjestelmäasetuksiin ja Avata Järjestelmän tiedot. Tämän alle on asetettu painike *environment variables*. Sinne lisäämme polun, jossa PHP:n ydintiedostot sijaitsevat tietokoneella (kuva 24). Tämän jälkeen Windows saatetaan joutua käynnistämään uudelleen, kuten omassa tapauksessani. (Loges, 2017)

Kuva 24 PHP yleismuuttujan lisäys (Loges, 2017)



Viimeisenä asiana PHP:n käyttöönottoa varten, oli kertoa IIS palvelulle, miten tulee toimia php tiedostojen kanssa. Avataan IIS Manager, navigoidaan websivulle, avataan *Handler Mappings* ja sen alla valitsemme Uuden moduulin kytkennän, *edit module mappings*. Lisäämme aukeavassa valikossa tiedot PHP nimisille tiedostoille, se sijainnin mistä käsittelijä löytyy, eli PHP juurihakemiston (kuva 25). Tämän jälkeen PHP on käyttövalmiina. (Loges, 2017)

Kuva 25 PHP Module mapping



Tulen siis hoitamaan myös videon lataamisen palvelimeen PHP:n avulla, joka edellyttää edelleen IIS:n, sekä PHP:n asetusten hienosäätöä. Jälkeenpäin tarkasteltuna, minun olisi täytynyt valita jokin toinen keino videon palvelimelle lataamiseen. PHP:lla lataamisessa oli parkin ongelmaa, mutta suurin ja ehdottomasti vakavin ongelma tuli esille, kun selvisi, että PHP ei osaa paloitella tiedoston siirtoa, vaan se käsittelee tiedoston siirron aina yhdellä suurella integer muuttujalla (PHP Manual, n.d). Tämä estää tiedostokokoa olemasta noin

2GB suurempia, vaikka kaikki PHP:n muut rajoittimet ohitettaisiinkin. Tämä tuli jäämään työ yhdeksi suurimmasta heikkoudesta.

Muita rajoittimia silti täytyy muokata, sillä oletusarvona tiedoston latauskoko on vain 50MB (PHP Manual, n.d). Tämä ei riitä missään tapauksessa tarpeisiimme, sillä videotiedostojen koko jää vain hyvin harvassa tapauksessa tuhon lukuun tai sen alle. Normaalisti videotiedosto, pituudestaan riippuen, voi olla useita Gigatavuja suuri. Haluamme siis asettaa PHP:n ini tiedostosta asetukset *post\_max\_size* sekä *upload\_max\_filesize* suuremmiksi, tai vaihtoehtoisesti rajattomaksi käyttämällä annettavana arvona nollaa (PHP Manual, n.d). Suuren koon rajoituksella tai rajoittamattoman väliltä valittaessa ei suurempaa väliä ole, sillä integer rajoittaa tiedostokoon puolestamme liian suurilla arvoilla.

Myös IIS:n puolelta tulee muuttaa asetuksia. Oletus Application poolilla on myös rajoitteita, jotka tulee muuttaa tai poistaa, ensimmäisenä tiedostokoko. Application poolin asetuksia pääsee muuttamaan valitsemalla IIS managerissa Application pools, sen alta oikea pooli jota halutaan muokata ja *Advanced settings*. Tämä avaa Application Poolin kaikki muokattavissa olevat muuttujat. Sen alta löydämme myös vaihtoehdon sovelluksille sallittuun tiedostokoon käsittelyyn.

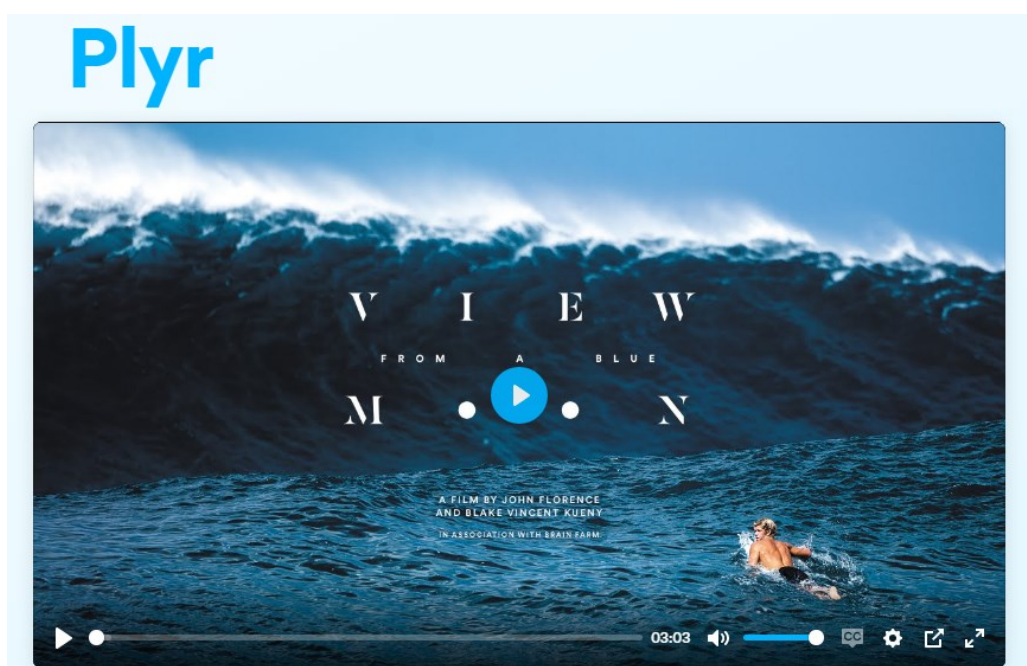
Tämäkään ei kuitenkaan riitä vielä, sillä sain videon latausta testaillessani huomata, että isoilla tiedostokoilla kaikki lähtee toimimaan kuten pitääkin, mutta PHP törmää säännöllisen ajan jälkeen aina virheeseen ja katkaisee latauksen. En tiennyt mistä ongelma johtuu, eikä PHP:n lokitiedoissakaan selvinnyt syyn alkuperää. Pitkän selvityksen jälkeen oivalsin itse, että Application poolin asetuksissa oli toinenkin kohta mikä tuli muuttaa. Se oli valinta sovellukselle sallitulle ajankestolle. Isoilla tiedostoilla lataus vei niin pitkään aikaa, että application pool aikakatkaisi PHP:n prosessin. Asetin vaihtoehdoksi rajattoman ajan arvolla 0.

#### 5.1.4 Plyr

Tarvisin websivuilla videoistimen, jolla katseltaisiin sekä VOD videot, että suorat lähetykset streamaaajilta. HTML5 yhteensopivia videoistimia on tarjolla hyvinkin monia vaihtoehtoja, mutta päädyin valinnassani toistimeen nimeltä Plyr. Tein valinnan seuraavin perustein.

Tarvitsin toistimen, millä pystyy muodostamaan adaptiivisen videoiden ja lähetysten katselukokoemuksen. Tällä tarkoitetaan sitä, että meillä on monta lähdettä samasta videosta, jotka eroavat tiedostokooltaan toisistaan, mahdollistaen vaihtoehtoja eri verkkonopeuksien omaaville katsojille. Tämä myös vaikuttaa suoraan videoiden laatuun. Mitä parempi laatu, sitä suurempi tiedostokoko, ja mitä suurempi tiedostokoko sitä suurempi bitrate, joka puolestaan on suoraan verrannollinen verkkokaistan käyttöön (Brown, 2020). Plyr tarjosi hyvän ja helpponäköisen keinon toteuttaa adaptiivinen ominaisuus. Toinen valintakriteeri oli se, että parhaimmisto videoitoimista on maksullisia, ja yksi työn vaatimuksista oli käyttää ilmaisohjelmistoa. Plyr oli siis mielestäni paras ominaisuuksiltaan, joka oli täysin ilmainen.

Kuva 26 Plyr - HTML5-videoitistin



Plyr on vapaan lähdekoodin projekti, jonka on kehittänyt Sam Potts. Potts on australialainen freelancer kehittäjä, jonka suurimmaksi projektikseen on noussut Plyr sen saaman suosion myötä. Plyrin ominaisuuksiin kuuluu mm. responsiivisuus, streamausmahdollisuudet, picture on picture, usean lähteen adaptiivinen rakenne, tuki 4K kuvanlaadulle ja monet muut ominaisuudet, mutta suurin vahvuus on vapaa lähdekoodi, mikä on toteutettu puhtaalla JavaScript koodilla, mahdollistaen helpon ja täydellisen muokkausmahdollisuuden. (Potts, n.d)

Plyr asennetaan lataamalla Pottsin GitHub sivulta projekti kokonaisuudessaan itselle. Tulen tarvitsemaan tätä websivullani, joten sijoitin koodin omaan hakemistoonsa, heti websivuni juurihakemiston alle. HTML:n se lisätään ensin lisäämällä Plyrin JavaScript tiedosto link tagilla, ja sen jälkeen JavaScriptiä hyödyntäen luoda uusi instanssi soittimesta, minkä yhteydessä valitaan halutut asetukset. HTML-videotagin alle annamme lähteet mitä toistimella katsotaan. (liite 5)

Toistimen videolähteeseen saattoi myös asettaa suoraan tulevan streamin, ja toistin osaa sen parsia käyttäjälle katselukelpoiseksi. En tullut käyttäneeksi adaptiivista rakennetta streamien yhteydessä, mutta se olisi ollut yksinkertaista tällä soittimella.

#### 5.1.5 FFmpeg

Palvelun yksi keskeisin työkalu on videoiden enkooderi, eli videon pakkaustyökalu. Verkon yli videoiden katselua rajoittaa verkon kaistanleveys, joten VODien luontia varten täytyy palvelimelle ladattu video pystyä pakkaamaan se verkon yli katseluun sopivaksi. Tässä haetaan niin sanottua kultaista keskitietä kuvanlaadun ja tiedostokoon välillä.

Video tulee myös pakata useaan kertaan luoden monta videotiedostoa ladatusta videosta. Käyttäjillä voi olla eritasoisia kaistanopeuksia, joita varten pitää olla kevyempiä tiedostoja toistettavaksi, ja kaikilla laitteilla emme pääse samanlaisiin ruudunkokoihin, joten videoiden resoluutiot pitää miettiä laitekohtaisesti. (Ozer, 2019)

Valitsin tähän tehtävään työkalun nimeltä FFmpeg. FFmpeg on työkalu, millä koostetaan ja ohjataan suuria määriä eri videonkäsittelyn työkalua (Wikipedia, n.d). Tarvitsin käyttööni enkoodereita, joita FFmpegillä on hurjasti tarjolla.

FFmpeg on alun perin Linux-ympäristöön kehitetty, mutta siitä on julkaistu paketteja, jotka toimivat myös Windowsilla (Wikipedia, n.d). FFmpegillä ei siis työn tekovaiheessa varsinaista graafista käyttöliittymää ollut, vaan Linuxille tyyppillisesti, sitä ohjataan tekstikomennoin. Tekstikomennot toimivat Windowsin command prompt komentotulkissa.

Valitsin FFmpegin siitä syystä, että tiesin kokemuksen perusteella, että sen käyttö on luontevaa PHP:n kanssa. Vaihtoehtoisena ohjelmana katselin handbrake nimistä ohjelmaa,

mikä on luotu juuri videon adaptiivisen rakenteen nopeaan saavuttamiseen, ja sitäkin olisi voinut ohjata komennoilla PHP:n kautta. En onnistunut testausvaiheessa handbrake komentoja onnistuneesti luomaan, joten päätin käyttää ohjelmaa, jonka tiesin toimivan.

FFmpeg asennukseen Windowsille on luotu valmiiksi esirakennettuja paketteja, joten asentaminen kävi helposti. Paketteihin ohjataan suoraan, FFmpegin kotisivuilta valittaessa käyttöjärjestelmäksi Windows (FFmpeg, n.d). Valitsin paketin, josta löytyi kaikki yleisimmät videokodekkien enkooderit. En halunnut turhaan liikaa työkaluja, joita en tule tarvitsemaan.

Kun FFMpeg oli ladattu, tuli se PHP:n tavoin asettaa yleismuuttujaksi tietokoneella (kuva 24). Tällä kertaa poluksi asetettiin FFmpegin juurihakemiston alla oleva bin hakemisto, joka sisälsi sovelluksen niin sanotun juoksettavan ohjelman.

Tämän jälkeen pystyimme komennoilla ohjaamaan FFmpegiä ja luomaan uusia videotiedostoja pakkaamalla vanhasta videosta uusi tiedosto haluamallamme enkooderilla.

## 5.2 Streamimoottori

Streamimoottorin asennus täytyi aloittaa oikean ohjelmiston valinnalla. Microsoft on panostanut stream tekniikoissaan Azure alustansa, joten Microsoftin tarjoamat streamausmahdollisuudet eivät tulleet pilveen sidotun toimintansa ansiosta kyseeseen. Täytyi siis valita kolmannen osapuolen palvelu.

Sisäverkossa toimimisen lisäksi työn tilaajan vaatimukseen kuului ilmaisohjelmien hyödyntäminen. Tästä syystä en tullut myöskään valinneeksi Wowza Stream Engineä, joka olisi ollut hyvä vaihtoehto kaikin puolin, ja lisäksi minulla oli opiskelujen pohjalta siitä jo hieman kokemusta. Päätin siis etsiä mahdollisimman samankaltaisen ilmaisohjelman Wowzan kanssa.

En oikeastaan löytänyt ilmaisohjelmistoja montaakaan kappaletta. Ainoa minkä löysin, joka muistutti toimivuudeltaan pitkälti Wowzaa, oli moottori nimeltä Mistserver. Vaihtoehtojen puutteessa valitsin sen ilman toista ajatusta, mistä myöhemmin sain kärsiä.

### 5.2.1 Mistserver

Mistserver on DDVTech nimisen firman omistuksessa oleva osittain vapaan lähdekoodin projekti. Mistserver on projektina melko nuori, mutta on nopeassa kehityksessä. DDVTech väittää Mistserverin olevan ainutlaatuinen toimintaperiaatteiltaan. Jokainen käyttäjän luoma yhteys käsitellään Mistserverissä erillisenä prosessina, tarkoittaen, että kaikki yhteydet, jotka streamia kuljettavat, ovat täysin omia entiteettejään. Tällä saavutetaan oletettavasti hyvä virheensietokyky, josta on hyötyä suurella käyttäjäkunnalla. (DDVTech, n.d)

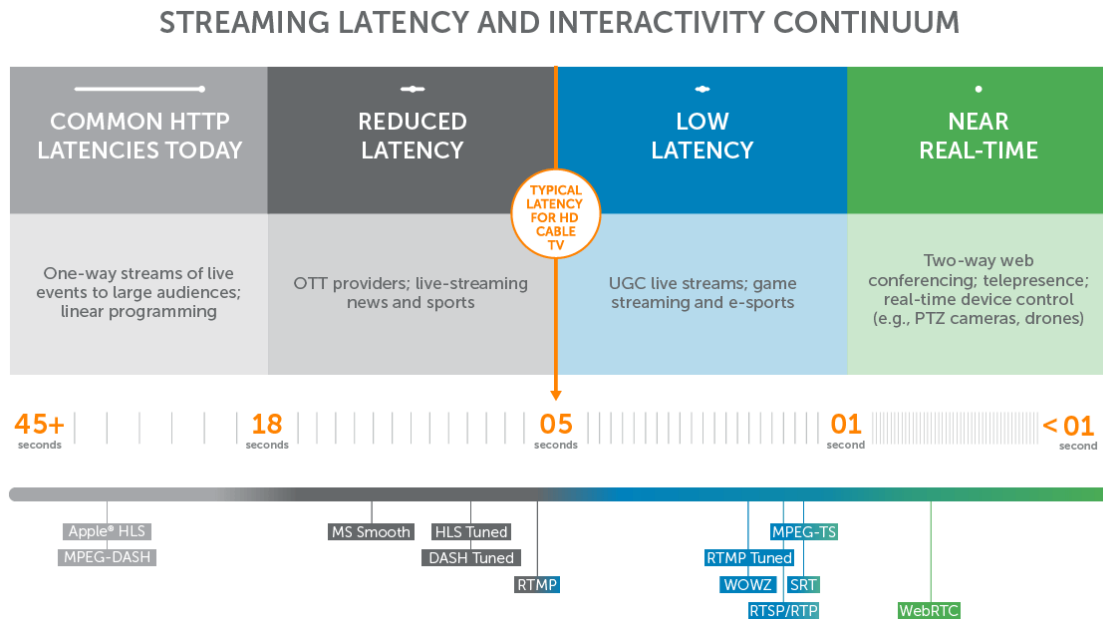
Kuva 27 MistServer (DDVTech, n.d)



Mistserverillä tehtävä streamaus on pohjimmiltaan kuitenkin hyvin perinteistä. Videodata kulkee verkon yli käyttäen Real Time Message Protokollaa, RTMP:ta. RTMP on ollut streamauksessa niin yleinen tekniikka, että ne toimivat lähes synonyymeina toisilleen yhdessä vaiheessa (Ruether, 2020). RTMP Protokolla toimii TCP protokollan päällä, kuten myös yleisimmän verkkoliikenteen käsittelevä HTTP protokolla. Näin ollen RTMP muodostaa käyttäjien laitteisiin yhteyden TCP porteilla. Portteja on kuitenkin useita käytössä ja niillä saavutetaan rinnakkaisen tiedonsiirron suomat nopeammat datapakettien siirrot.

RTMP on streamitekniikoita verrattaessa niin sanottu kultaisen keskitien kulkija. Tämä tarkoittaa, että RTMP:lla saavutetaan hyvänlaatuista streamia ilman suuria viiveitä. Viive on kuitenkin noin 5 sekunin luokkaa, joten reaaliajasta ei voi täysin puhua. (Ruether, 2020)

Kuva 28 Streamitekniikkojen latenssit (Ruether, 2020)



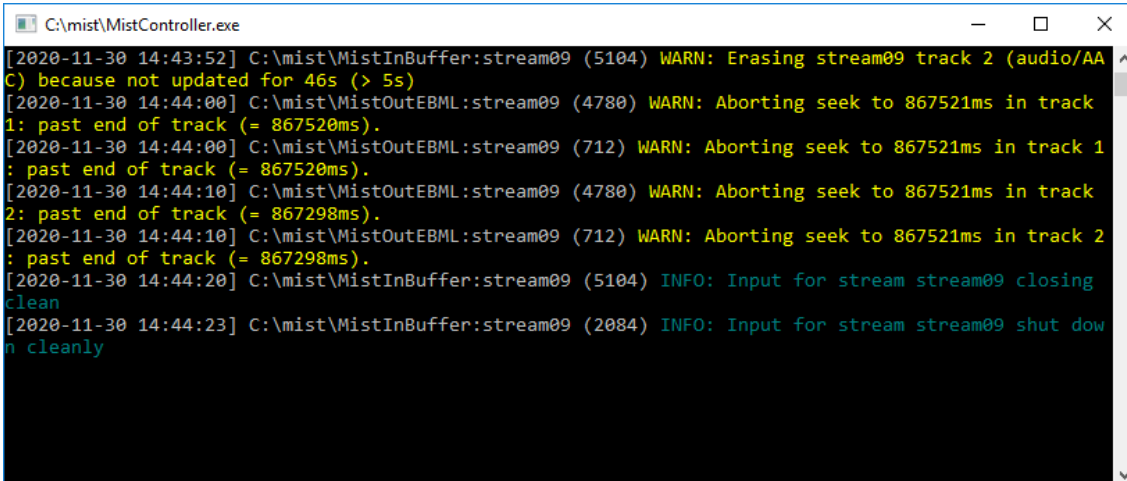
Mistserverillä oli kuitenkin suuria puutteita, jotka ymmärsin niin myöhäisessä vaiheessa, että päätös tehtiin toteuttaa työ silti Mistserverin avulla. Suurin puutteena mainittakoon se, että se tosiaan on osittain vapaan lähdekoodin ohjelmisto, mutta osa tärkeistä ominaisuuksista on jätetty maksumuurin taakse. Esimerkiksi Streampalvelimille tyypillinen Push ominaisuus puuttuu ilmaisversiosta, joka olisi ollut melko tärkeä ominaisuus minun kohdallani. Push ominaisuus tarkoittaa sitä, että uuden streamiprosessin voivat käyttäjät muodostaa itse, joka mahdollistaa helpon toimivuuden websivulla (manual, 2020). Nyt jouduin luomaan prosessit etukäteen, sekä sitomaan ne kiinni tiettyyn IP osoitteeseen. Tämä rajoittaa toimintaa huomasti, ja tulevaisuuden laajennuksia varten on hyvin työläs ja kaikinpuolin huono asia.

### 5.2.2 Misterverin asennus

Mistserver on ladattavissa DDVTechnin GitHub sivulta. Latasin itselleni ensin koko Mistserverin sisältävän pakatun tiedostokansion, zip tiedoston, tietokoneelle missä oli verkkoyhteys. Siirsin sen sitten ulkoisella USB-muistilla palvelinkoneelle, jossa purin ja asensin sen.

Tämän kummempaa ei Mistserverin käyttöönotto varsinaisesti ole. Kun se on purettu koneelle, käynnistetään se MistController nimisellä sovelluksella, joka sijoittaa Mistserverin juurihakemistossa. Sovellus lähtee käyntiin ja jää ohjaamaan verkkoliikennettä moottorille (kuva 29).

Kuva 29 MistController



```
C:\mist\MistController.exe
[2020-11-30 14:43:52] C:\mist\MistInBuffer:stream09 (5104) WARN: Erasing stream09 track 2 (audio/AA
C) because not updated for 46s (> 5s)
[2020-11-30 14:44:00] C:\mist\MistOutEBML:stream09 (4780) WARN: Aborting seek to 867521ms in track
1: past end of track (= 867520ms).
[2020-11-30 14:44:00] C:\mist\MistOutEBML:stream09 (712) WARN: Aborting seek to 867521ms in track 1
: past end of track (= 867520ms).
[2020-11-30 14:44:10] C:\mist\MistOutEBML:stream09 (4780) WARN: Aborting seek to 867521ms in track
2: past end of track (= 867298ms).
[2020-11-30 14:44:10] C:\mist\MistOutEBML:stream09 (712) WARN: Aborting seek to 867521ms in track 2
: past end of track (= 867298ms).
[2020-11-30 14:44:20] C:\mist\MistInBuffer:stream09 (5104) INFO: Input for stream stream09 closing
cleanly
[2020-11-30 14:44:23] C:\mist\MistInBuffer:stream09 (2084) INFO: Input for stream stream09 shut dow
n cleanly
```

Kun MistContoller on laitettu käyntiin, pääsemme selaimella sen hallintaliittymään, kun navigoimme selaimella tietokoneen IP osoitteeseen tai local loopback osoitteeseen 127.0.0.1 ja tarkennamme vielä kohdeportiksi Mistserverin oletusportin 4343. Tämä avaa meille graafisen hallintapaneelin selaimessa, jossa pääsemme konfiguroimaan asetuksia (kuva 30).

Vasta työn tässä vaiheessa ymmärsin, että osa Mistserverin ominaisuuksista oli väitetyt avoimen lähdekoodin sijaan maksuseinän takana. Huomasin että kahteen hyvinkin tärkeään ominaisuuteen en päässyt käsiksi. Ensimmäinen oli automatisoitu streamien luonti push toiminnolla. Tämä tarkoitti, että jouduin luomaan streamit manuaalisesti Mistserverille, ja se tarkoitti, että streamit tulisivat olevan sidottuja siihen IP osoitteeseen, jonka niille luomisen yhteydessä antaa. Tämän vuoksi koko webpalvelimeni käyttömukavuus laski, sillä aina kun aktiivihakemiston käyttäjiä ja tietokoneita muutettaisiin, täytyisi luoda uudet instanssit streamimoottorillekin. Sen lisäksi pitäisi konfiguroida DHCP palvelin, jolle antaisi varaukset laitteille, jotta niiden IP osoitteet pysyisivät moottorin streamien kanssa yhtenevinä, tai vaihtoehtoisesti IP osoitteet jäisi järjestelmänvalvojan vastuulle asettaa manuaalisesti.

Toinen, vielä ehkä suurempi ongelma tuli siinä, että adaptiivisen rakenteen streamia ei ollut mahdollista luoda. Käyttäjät olisivat lukittuna yhteen bitrateen, joka tietysti aiheuttaa sen, ettei käyttäjille voinut tarjota todella hyvälaatuista videota, ja vastaavasti kaikista hitaimpien kaistanopeuksien omaavilla streamit eivät tulisi pätkimättä toimimaan.

Juttelin asiasta työn tilaajan kanssa ja hän teki päätöksen, että jatkamme Mistserverin kanssa, sillä minulla ei ollut esittää vaihtoehtoista ilmaista ratkaisua. Kyse oli vain harjoitteluympäristöstä, joten yhdenlaatuinen streami oli työn tilaajan mukaan hyväksyttävä tulos. Jälkikäteen ajatellen olisin voinut yrittää streamimoottorin rakentaa itse FFmpegin avulla, mutta se olisi vienyt varmasti aikaa paljon.

Streamit luotiin moottorille selaimen käyttöliittymän *streams* välilehdessä (kuva 30). Loin moottorille 11 streamia. Yhden medialinjan studion streamilaitteelle, ja 10 aktiivihakemiston käyttäjille. Aktiivihakemiston käyttäjät lähettävät push tekniikalla datan moottorille, jossa moottori käsittelee sen ja lähettää webpalvelimelle rtmp protokollalla.

Kuva 30 Mistserver manager

The screenshot shows the MistServer Management Interface in a web browser. The interface is titled "MistServer Management Interface" and shows a "CONNECTED" status. The main content area is titled "Streams" and contains a table of stream configurations. The table has columns for Stream name, Source, Status, and Connections. All streams listed are currently "Unavailable".

Stream name	Source	Status	Connections
mainstream	push://192.168.73.220	Unavailable	0
stream01	push://192.168.73.221	Unavailable	0
stream02	push://192.168.73.222	Unavailable	0
stream03	push://192.168.73.223	Unavailable	0
stream04	push://192.168.73.224	Unavailable	0
stream05	push://192.168.73.225	Unavailable	0
stream06	push://192.168.73.226	Unavailable	0
stream07	push://192.168.73.227	Unavailable	0
stream08	push://192.168.73.228	Unavailable	0
stream09	push://192.168.73.229	Unavailable	0
stream10	push://192.168.73.230	Unavailable	0

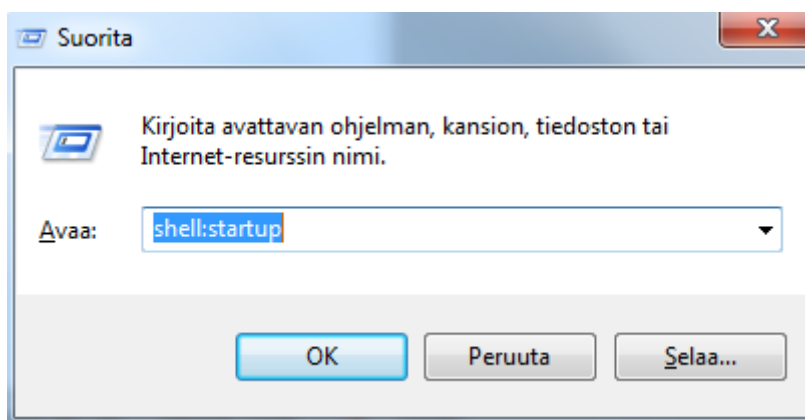
Muita asetuksia moottoriin ei nyt sitten tarvinnut tehdä, sillä takaiskut poistivat hieman konfiguraatiotyötä.

### 5.2.3 Toiminta Windowsissa

Mistserver on palvelimen näkökulmasta ohjelma, eikä palvelu. Tästä syystä nousi yksi iso ongelma sen käytettävyyden kanssa. Se ei nimittäin lähde käyntiin, ennen kuin tietokoneen käyttäjä sen käynnistää. Jos esimerkiksi sähkökatkoksen vuoksi palvelin sammuu, se osaa käynnistyä kyllä uudelleen sähköjen palaututtua, mutta ohjelmat eivät lähde palvelujen tapaan käyntiin tietokoneen automaattisen käynnistymisen yhteydessä. Tämä ongelma tulisi siis kiertää, ettei aina, kun syystä tai toisesta palvelin käynnistyy uudelleen, joudu palvelimen ylläpitäjä tulemaan fyysisesti paikan päälle käynnistämään mistserver uudelleen, ennen kuin streamipalvelu jälleen toimii.

Ensimmäiseksi Misterverin käynnistysohjelma tuli asettaa Windowsin käynnistymisen yhteyteen automaattiseksi. Se onnistuu navigoimalla ensin Windowsin tätä ominaisuutta varten tehtyyn startup hakemistoon. Se saattaa sijaita hieman eri paikoissa käyttöjärjestelmän version mukaan, mutta sitä varten voimme käyttää seuraavaa keinoa. Painamme ensin Windows painiketta + R kirjainta. Meille aukeaa suorita ikkuna, johon kirjoittamalla *shell:startup* pääsemme suoraan haluamaamme kohteeseen (kuva 31).

Kuva 31 Windowsin Suorita / Run työkalu



Tähän kansioon päästyämme riittää, kun viemme pikakuvakkeen MistControllerista kansion sisään. Pikakuvakkeen voi luoda helposti klikkaamalla hiiren oikealla napilla MistControllerin

käynnistysohjelmaa, ja valitsemalla luo pikakuvake. Tämä pikakuvake sitten viedään startup kansioon. Kaikki ohjelmat tämän kansion sisällä käynnistyvät, kun Windowsiin kirjaudutaan.

Mistä jää enää ongelman suurempi puoli. Kun Windows palvelin käynnistyy, siihen tulee turvallisuussyistä kirjautua järjestelmänvalvojana, mikäli mielii päästä toimimaan koneella. Meidän tulee kiertää kyseinen turvallisuusasetus, mikä tietenkään ei ole suositeltavaa normaalissa ympäristössä, mutta tässä työssä voitiin sallia.

Päädyin ratkaisuun, missä asetan käyttäjien salasanan vanhenemisasetukset pois päältä, ja ajan Windowsin rekisteritietoja muokkaamalla asetuksen, joka syöttää kirjautumistiedot automaattisesti. Se tehtiin kuvan 32 ohjeilla.

Kuva 32 Registry key – muokkaus (expert-advice, 2017)

#### Registry key technique.

1. Run exe
2. Navigate to:  
`HKLM\Software\Microsoft\Windows NT\CurrentVersion\winlogon`
3. Set or create the following keys:  
`(DWORD) AutoAdminLogon = 1`  
`(String) DefaultUserName = Your user name`  
`(String) DefaultPassword = Your password`
4. Now restart and lets to see it works!

Tämän jälkeen Windows osasi kirjautua sisään käynnistyksen yhteydessä automaattisesti ja Mistserver käynnistysohjelma käynnistyy kirjautumisen yhteydessä automaattisesti. Onnistuin siis ohittamaan ongelman, vaikkakin kyseenalaisin keinoin.

### 5.3 Websivut

Esittelen seuraavaksi tekemäni websivut. Näytän, miten olen suunnitellut käyttäjän näkökulmasta alustan toiminnallisuuden, ja kerron sitten avainkohtia, miten toteutin kyseisen toiminnan tai ominaisuuden.

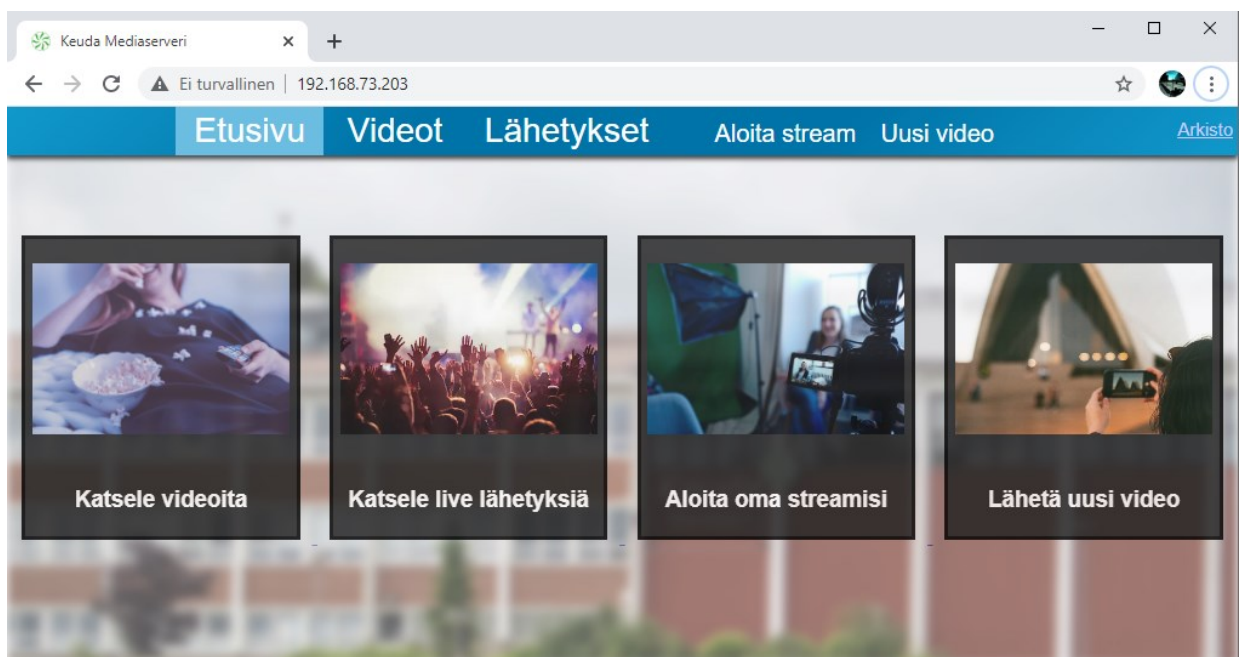
Websivut kasvoivat ehkä turhankin suureksi koodiltaan, joten en valitettavasti kaikkea koodia tässä kirjallisessa osiossa ala kattamaan, mutta mielestäni tärkeät osat laitan liitteiksi, joihin viittaa tässä osiossa.

### 5.3.1 Käyttöliittymä

Tavoitteeni oli tehdä käyttöliittymästä helppokäyttöinen, samalla kuitenkin pyrkiessäni silmää miellyttävään ulkoasuun. Sivustolla liikkumisen olin pyrkinyt tekemään mahdollisimman yksinkertaiseksi, jotta oppilaat osaisivat toimia sivustolla heti ensimmäisellä kerralla. Samalla yritin pitää myös yhtenevää ilmettä Keudalle tyypillisen grafiikkasuunnittelun kanssa, ainakin oman tulkintani mukaan.

Websivujen oletusdokumentiksi, eli sivu, joka avataan automaattisesti käyttäjän navigoidessa websivulle ilman tarkentavaa polkua, tein tervetuloa sivulle tyypillisen etusivun (kuva 33). Etusivun sisältö on suuret graafiset ja interaktiiviset linkit sivuston tarjoamiin palveluihin. Etusivu, sekä muutkin sivut, oli rakennettu responsiiviseksi CSS tyyllittelyllä. Graafiset linkit muovautuivat webselaimen tai laitteen näytön mukaan niin, että siitä saisi käyttäjät selvää, vaikka selaimen kokoa muutettaisiin. kaikki sivuston käyttämä CSS tyyllittely on sijoitettu yhteen CSS tiedostoon (liite 8), joka osoitetaan jokaiselle sivulle erikseen.

Kuva 33 Websivuston etusivu



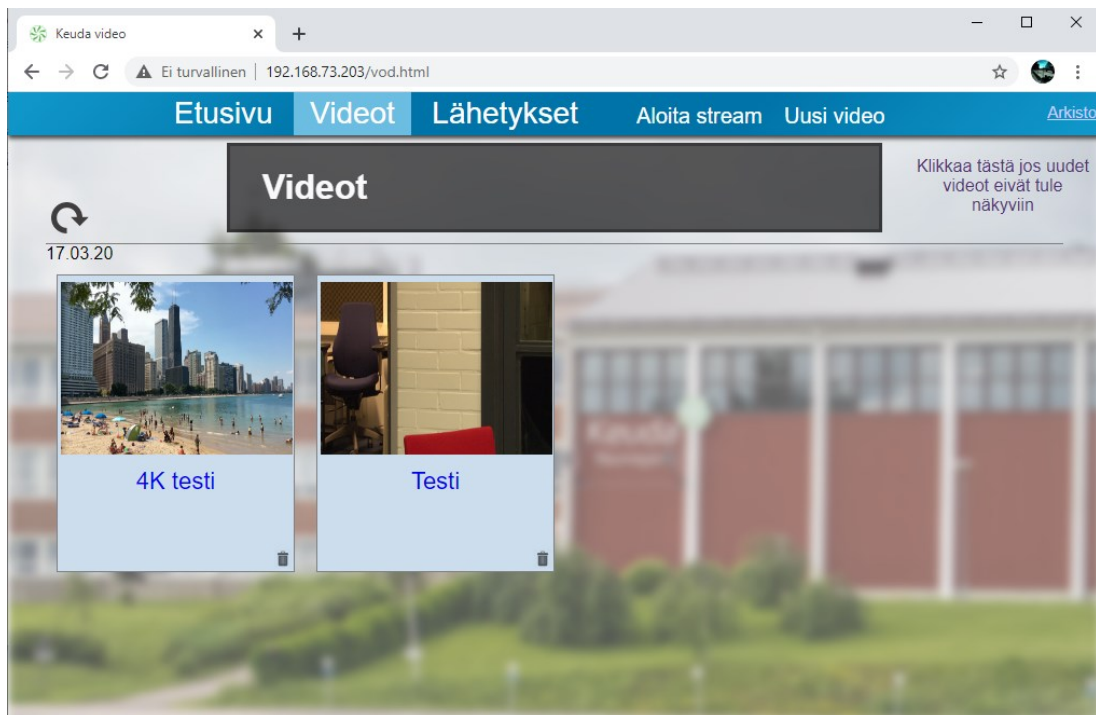
Sivun ylälaudassa sijaitsee navigointipalkki, josta voi liikkua sivun eri palveluihin, navigointipalkin samalla kertoen käyttäjälle korostusvärillä, millä sivulla hän sillä hetkellä on. Navigointipalkki pysyy samanlaisena kaikilla sivuston sivuilla, tietenkin korostaen aina sen kyseisen sivun, jolla käyttäjä sillä hetkellä on. Tämä toteutettiin HTML luokilla. HTML luokka on HTML elementteihin sisällytettävä attribuutti, johon viitataan CSS tiedostossa tyyllittelyä varten (w3schools, n.d). Myös sivustolla käytetty taustakuva, joka on sumennettu kuva Keuda Nurmijärven rakennuksesta, lisätään CSS:lla sivuston jokaiselle sivulle.

Käytin CSS3:n tarjoamaa flex ominaisuutta graafisten linkkien hallitsemiseen. Ilman sitä laatikot muovautuivat sisällön ja selaimen koon mukaan siten, että ne eivät olleet rivillä samassa tasossa. Flex-muotoilu auttoi pitämään lokerot aina samalla tasolla.

### 5.3.2 VOD ja stream sivut

VOD katseluun tehty *Videot* sivu (kuva 34), on toteutettu myös minimalistiseksi. Videot, jotka palvelimelle lähetetään, lisätään erilliseen HTML tiedostoon, joka esitetään Videot sivulla iframe elementillä. Iframe on keino esittää muita websivuja jonkin toisen sivun sisällä. Saavutin iframen avulla keinon päivittää listaa ilman, että Videot sivulta tarvitsisi poistua tai selainta päivittää. Päivitys toteutetaan iframen sisällä (liite 4). Kuvassa on näkyvissä kaksi esimerkkiä käyttäjän palvelimelle lähetetyistä videoista. Ne tulevat näkyviin linkin muodossa lähetyksen yhteydessä generoitavan html koodin avulla (liite 1), joille annetaan aina sama luokka CSS tyyllittelyä varten. Linkit sisältävät tyyllitellyn HTML lokeron, jonka sisään asetetaan käyttäjän videolle antama nimi ja videon enkoodauksen yhteydessä tuotettava kuvake.

Kuva 34 Videot - sivu



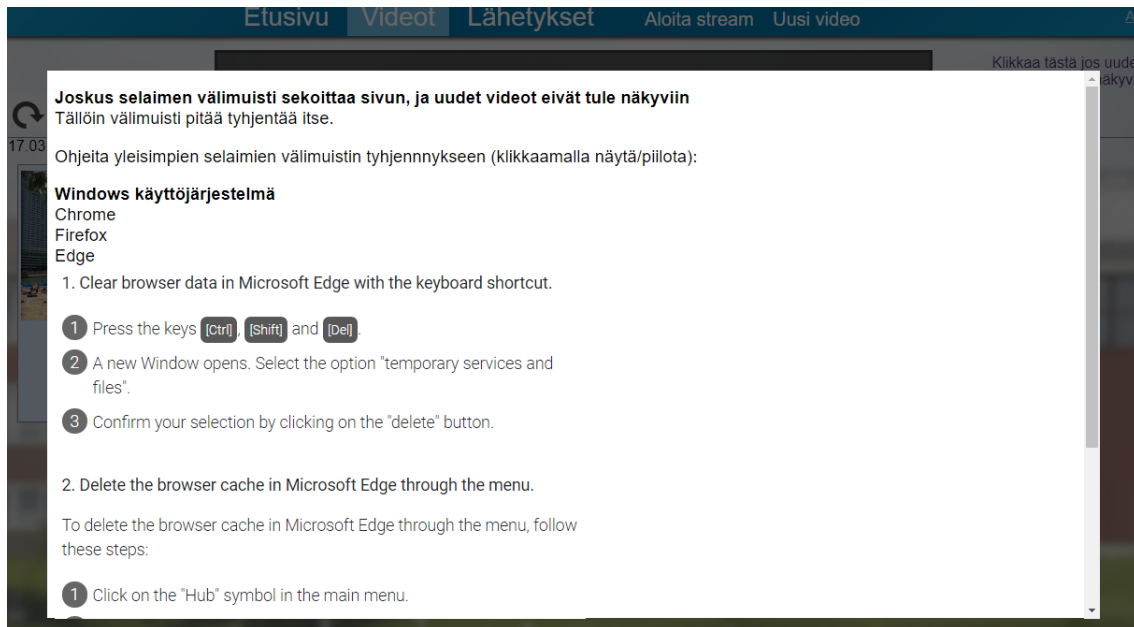
Tein videolistalle myös manuaalisen päivitysmahdollisuuden, joka näkyy kuvassa vasemmalla ylhäällä, päivitystä kuvaavana interaktiivisena ikonina. Kun hiiren vie ikonin päälle, se alkaa pyöriä, jonka voi olettaa olevan tuttu päivityksen merkki webkäyttäjille. Toteutin tämän siksi, että automaattinen päivitys ei tuntunut toimivan aivan kaikilla selaimilla.

Itse videon katselua varten käyttäjä valitsee haluamansa videon klikkaamalla sitä. Tämä lataa iframen sisään videon lähetyksen yhteydessä generoitavan HTML-sivun, joka sisältää vain Plyr soittimen, jonka lähteiksi on asetettu kaikki videon eri laatuiset versiot, joista käyttäjä sitten pääsee valitsemaan haluamansa laadun (liite 5). Sivulla on myös napin muodossa linkki, joka ohjaa iframen takaisin videolistalle. Kaikki siis tapahtuu poistumatta koskaan Videot sivulta.

Huomasin myös, että jotkin selaimet latasivat iframen päivityksen yhteydessä sivun suoraan paikallisesta välimuististaan, vaikka sivusto käskee lataamaan sivun uudelleen palvelimelta (liite 4). Tämä tuntui ongelmalta, jota ei voinut ohittaa palvelimen puolelta, vaan ongelma olisi ratkaistava käyttäjän päädystä selaimen asetuksista. En siis voi käyttäjien puolesta ratkaista selainkohtaista ongelmaa, joten jouduin kiertämään sen. Tuotin ohjeet, miten yleisimmillä eri webselaimilla välimuisti poistettaisiin itse. Se näkyy sivun oikealla ylhäällä

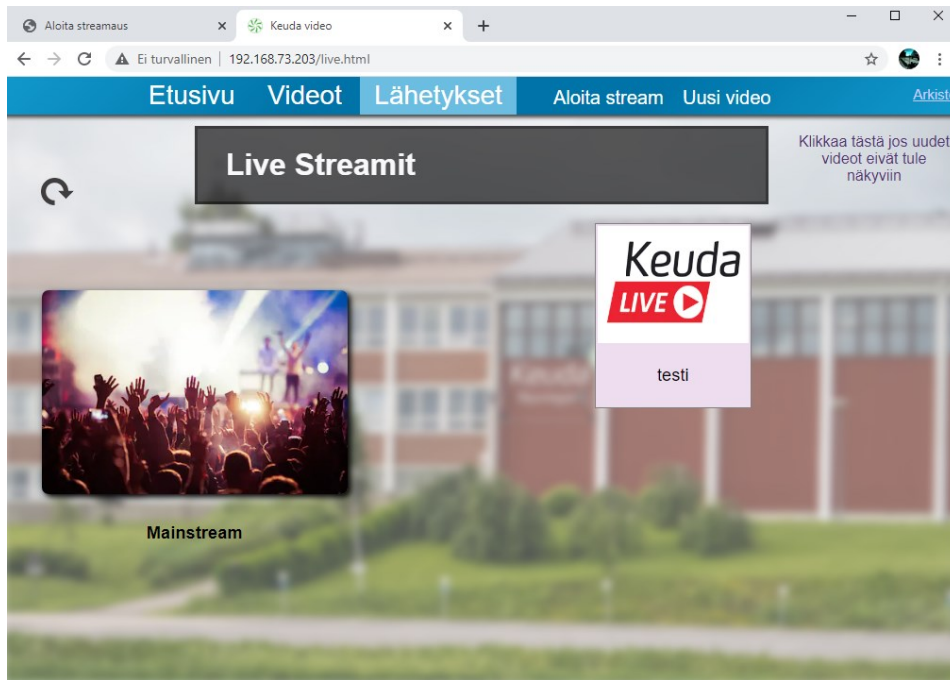
olevana linkkinä, joka avaa meille toisen iframen, joka esittää tekemäni ohjesivut (kuva 35). Ohjeet kattavat toistaiseksi vain PC ja Mac käyttöjärjestelmien yleisimmät selaimet. Selaimen nimeä klikkaamalla tulee näkyviin ohje vaihekohtaisesti.

Kuva 35 Välimuistin poisto-ohjeet



Live Streamit sivut toteutin samalla periaatteella, missä parhaillaan käynnissä olevat lähetykset esitetään iframella näytettävässä erillisessä sivussa. Streamsivu eroaa käytännössä ainoastaan siitä, sivun sisältöalue on jaettu kahtia. Toisella puolella on mediastudion streamilaitteelle omistettu linkki, joka on aina näkyvissä, lähetyksen tilasta huolimatta. Toisella puolella oli sitten edellä mainitsemani iframe, joka päivittyi käynnissä olevien aktiivihakemiston käyttäjien streameihin. Kuvassa 36 näkyy esimerkki, kun yksi aktiivihakemiston käyttäjä lähettää parhaillaan streamia.

Kuva 36 Lähetykset - sivu alias streamisivu



Tyylittelyssä olisi voinut parantaa, sillä se näyttää omasta mielestäni hieman hassulta, jos aktiivihakemiston käyttäjillä ei ole lähetyksiä käynnissä montaa, taikka lainkaan. Monen lähetyksen kanssa sivu näytti hyvältä.

### 5.3.3 Streamin aloitus

Streamin aloituksen toteutin siten, että käyttäjän painaessa *Aloita stream* linkkiä, webkäyttäjä ohjataan PHP sivulle, joka tarkastaa IP osoitetta vertaamalla, kuuluuko käyttäjä aktiivihakemiston käyttäjiin (liite 3). Mikäli kuuluu, ohjaa PHP sivu käyttäjän toiselle sivulle, missä käyttäjän on mahdollista aloittaa stream. Mikäli IP osoite ei kuulunut aktiivihakemiston käyttäjien IP osoitteisiin, ohjasi sivu käyttäjän takaisin sivuston etusivulle.

Tämä oli huono ratkaisu, ja sen tuli ollakin vain väliaikainen, siitä syystä että vaikka IP osoitteet olivat varattu aktiivihakemiston käyttäjille, voisi taitava ihminen helposti hyödyntää asettamalla IP osoitteen itse, jos hänellä olisi pääsy sisäverkkoon. Käytännössä kuitenkin kaikki kenellä olisi mahdollisuus päästä sisäverkkoon, saisi myös käyttäjän aktiivihakemistoon. Oikea tyyli silti suojata sivu ulkopuolisilta olisi ollut IIS:n yhteydessä aktiivihakemiston käyttäjätunnuksilla vaadittu tunnistautuminen, *basic authentication*.

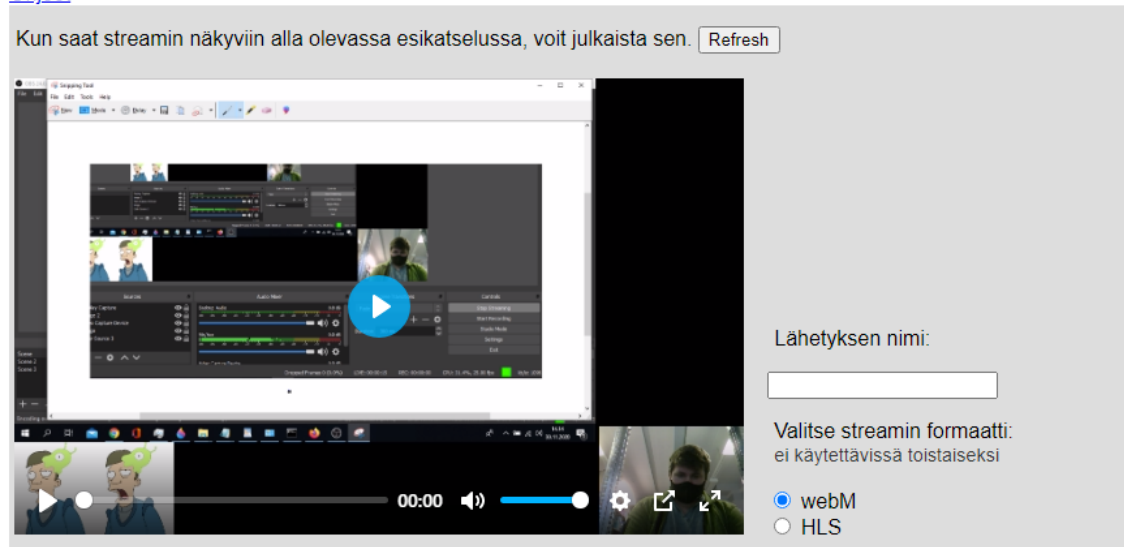
Sivu, jolle aktiivihakemiston käyttäjät ohjataan, kertoo käyttäjälle streamin aloitusta varten tarvittavat tiedot (kuva 36). Tiedot olivat streamimoottorissa asetettu streamin rtmp osoite, sekä stream key, jolla vielä varmennetaan tuleva streamdata (Ozer, 2019).

Kuva 37 Streamin aloitus 1

**Server:** <rtmp://192.168.73.206/live/stream09>

**Stream key:** [stream09](#)

[Ohjeet](#)



Kun käyttäjä saa valitsemallaan ohjelmistolla lähetysnimen käyntiin, se tulee esikatseltavaksi sivulla olevaan Plyr-toistimeen. Plyr-toistimelle on tehty päivitysnappi, jolla toistimen lähteen saa ladattua uudelleen, jos käyttäjällä on siihen tarve.

Sivulla on aluksi harmaa väri-indikaattori, joka viestii käyttäjälle että lähetys ei ole vielä päällä. Jotta todetaan lähetysnimen olevan päällä, tarkistetaan JavaScript koodilla, saako Plyr-soittimella katseltua lähetystä itse (liite 7).

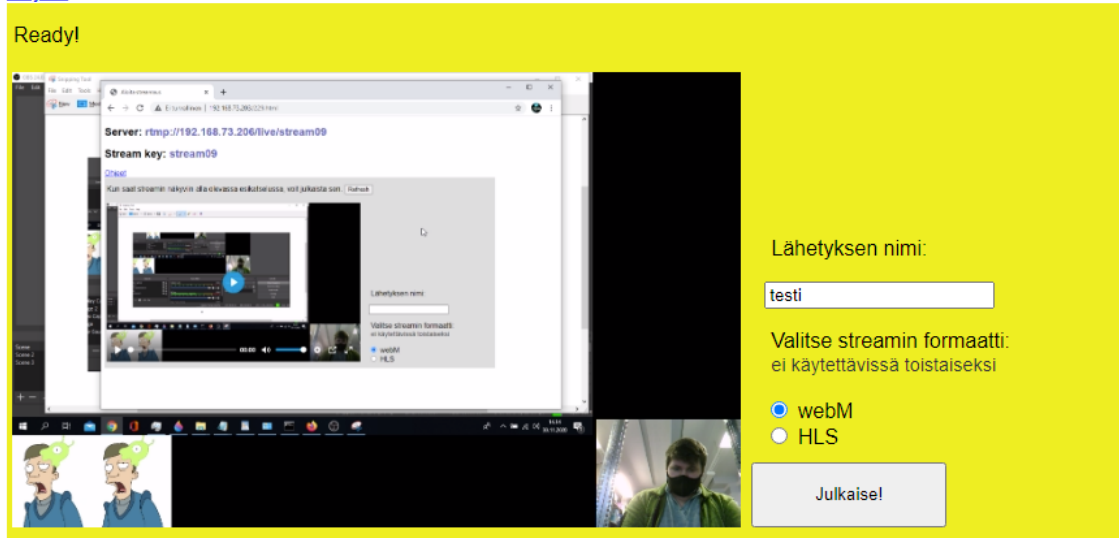
Kuvissa näkyvä *streamin formaatti* oli suunnitteilla ollut lisäominaisuus, mistä käyttäjä pystyisi valitsemaan mieleisensä transkoodaus menetelmän. Sitä varten minun olisi täytynyt asentaa moottorille lisäosa, mutta jätin sen tekemättä ollessani valmiiksi tyytymätön moottorin ominaisuuksiin.

Jos lähetyksen saa toistimessa päälle, muuttuu sivu niin sanottuun 'valmiina lähetykseen' tilaan, josta teksti ja keltainen väri-indikaattori kertovat (kuva 38). Samalla käyttäjälle ilmestyy myös painike, millä hän voi julkaista lähetyksen.

Kuva 38 Streamin aloitus 2 - valmiustila

**Stream key:** stream09

[Ohjeet](#)



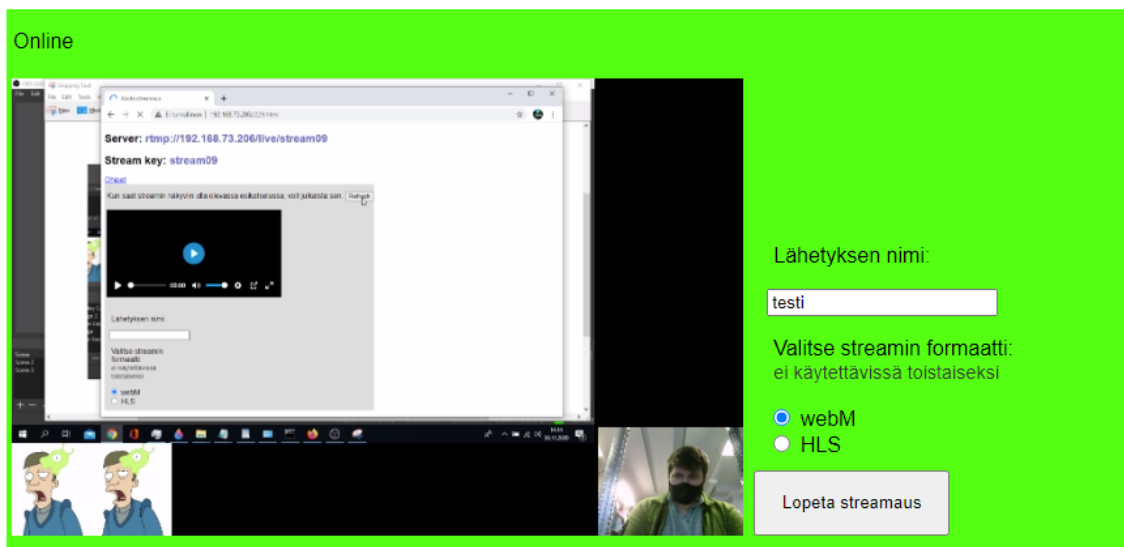
Ennen lähetyksen aloitusta tulee käyttäjän valita lähetyksen nimi. Nimi tulee näkyviin lähetyksen yhteyteen muille käyttäjille. Kun käyttäjä painaa Julkaise nappia, lähetyksen julkaistaan, joka tarkoittaa, että lähetyksen sivulle ohjaava linkki asetetaan käyttäjille näkyviin (kuva 36). Se toteutetaan PHP-koodilla, joka muokkaa lähetyksien sisältävän HTML-sivun kyseisen kohdan näkyviin (liite 7). Oletusarvona HTML-sivulla lähetykset ovat ns. kommentoitu pois, kommenttirivien sisältäessä indikaattorin, josta PHP-koodi tietää, mikä kommentti tulee poistaa. Vastaavasti lähetyksen loputtua, kommentit lisätään PHP-koodilla takaisin.

Lähetyksen alettua Streamin aloitussivun väri-indikaattori muuttuu vihreäksi, ja lähetyksen nimi korvaa streamin tiedot (kuva 39). Samanaikaisesti Julkaise painikkeen korvaa Lopeta painike, josta painamalla stream poistuu julkisesta näkymästä, vaikka stream itsessään ei sulkeudu. Tästä ilmoitetaan käyttäjälle muistutuksella sulkea stream käyttäjän streamiohjelmiston puolelta.

Käyttäjän ei tule poistua sivulta niin kauan kuin lähetys pidetään julkisena. Jos käyttäjä poistuu sivulta, tulkitsee koodi sen valintana lopettaa lähetys (liite 7). lähetysten nimeäkään ei voi vaihtaa, vaan tulee ensin lopettaa edellinen lähetys sivulla ja julkaista uusi lähetys uudella nimellä. Se on nopeaa, eikä katkaise katsojienkaan yhteyttä, mutta uuden lähetysten nimi ei tule käyttäjille ja katsojille näkyviin ennen kuin päivittävät sivut.

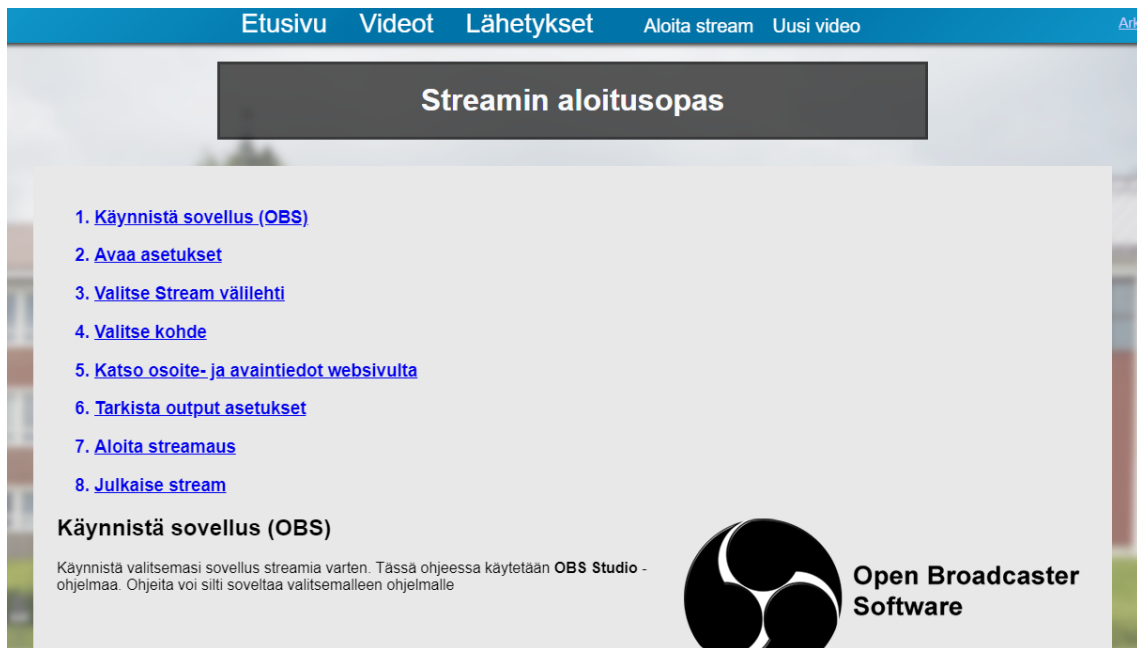
Kuva 39 Aloita stream 3 - lähetys käynnissä

## testi



Tuotin käyttäjille, joilla ei ole kokemusta entuudestaan streamauksesta, kattavat ohjeet streamauksen aloitukseen (kuva 40). Ohjeet ovat tehty olettaen, että käyttäjän streamaus ohjelmisto on OBS studio. Tämä sen takia, että OBS on todella suosittu ohjelmisto, ja se on kirjoittamaton linjaus Keudalla ohjelmiston valintaan. Ohjeet ovat pitkät, joten sivun ylälaudassa on pikavalinnat ohjeen eri vaiheisiin numeroituna.

Kuva 40 Streamin aloitusohjeet - sivu



### 5.3.4 Videon lähetys

Uuden VOD tyyppisen videon lähetys tapahtuu sivuston sivulla *Uusi video*. Videon lähetys toteutetaan PHP:n tiedostosiirron käsittelijöillä. PHP:n käsittelijöitä varten piti muodostaa yksinkertainen HTML-sivu, jossa käyttäjä ensin valitsee haluamansa videotiedoston (kuva 41), painamalla valitse tiedosto nappia sivun ylälaidassa. Tiedoston valinta toteutetaan HTML5:n sisäänrakennetulla tiedostovalintakäsittelijällä, joka avaa käyttöjärjestelmän tiedostohakemiston. Hakemistosta käyttäjä etsii videotiedostonsa. Rajoitin valinnat yhteen kappaleeseen käsittelyn helpottamiseksi.

Kuva 41 Uuden videon lähetyks

The screenshot shows a web interface for uploading a video. At the top, there are navigation tabs: 'Etusivu', 'Videot', 'Lähetykset', 'Aloita stream', and 'Uusi video'. Below the navigation, there is a 'Valitse tiedosto' button and a message 'Ei valittua tiedostoa'. A text input field for 'Videon nimi' is present, with a character limit of 50. Below the name field, there is a 'Käytä oletusarvoja ↑' button. A settings modal is open, showing options for resolution and codec. The resolution options are 240p, 360p, 480p, 576p, 720p (selected), 1080p, 1440p, and 2160p. The codec options are H.264 (.mp4) (selected), VP9 (.webm), and Xvid (.avi). Under H.264, there are profile options: high, main (selected), and baseline.

Seuraavaan valintaan käyttäjä syöttää tekstinä videolleen nimen. Asetin nimen maksimipituudeksi 50 merkkiä, että liian pitkiä nimiä ei voisi syöttää, jotka veisivät turhaan tilaa websivuilla ja palvelimelle tallennettavien tiedostojen nimissä.

Tämän jälkeen käyttäjällä on valinta avata lisäasetukset, jotka ovat tarkoitettu edistyneemmille käyttäjille, joten ne ovat oletusarvoisesti piilotettu, mutta painamalla valintaa aukeaa tulevaa enkoodausta varten valittavat asetukset. Tästä Käyttäjällä on mahdollisuus useaan valintaan, jotka sisältävät haluttujen resoluutioiden valinnat, koodekin valinnan ja muut lisävalinnat, kuten oman kuvakkeen, eli thumbnailin, lisäämisen.

Kun halutut valinnat ovat asetettu, valitsee käyttäjä sivun alhaalta Aloita vaihtoehdon. Se käynnistää lähetyksen käsittelevän PHP-koodin (liite 2). Koodi ensimmäisenä suorittaa vertailun, onko valinnat sallittuja. Vertailu tarkastaa ensin onko tiedosto edes videotiedosto hyödyntämällä MIME-tyyppejä. MIME-tyyppi on webstandardi, joka tulkitsee tiedoton todellisen muodon (MDN web docs, n.d -c). Muotoja voivat esimerkiksi olla video, audio, teksti ja ohjelma. Teen tämän siksi, että maailmassa on hyvin monenlaisia tiedostotyyppejä videoille. Lähes kaikilla laitevalmistajillakin löytyy jokin omanlaisensa videotiedosto, joita yleisesti ei osaa ulkopuoliset laitteet käsitellä. MIME-tyyppi silti kertoo tuntemattomista

tiedostoista, minkä luontoinen tiedosto on kyseessä. Saatoinkin tehdä tämän siksi, että FFmpeg osaa käsitellä niin laajasti erimuotoisia videotiedostoja, että luultavasti kukaan ei koskaan palvelimelle tule lataamaan tiedostoa, jota FFmpeg ei ymmärtäisi.

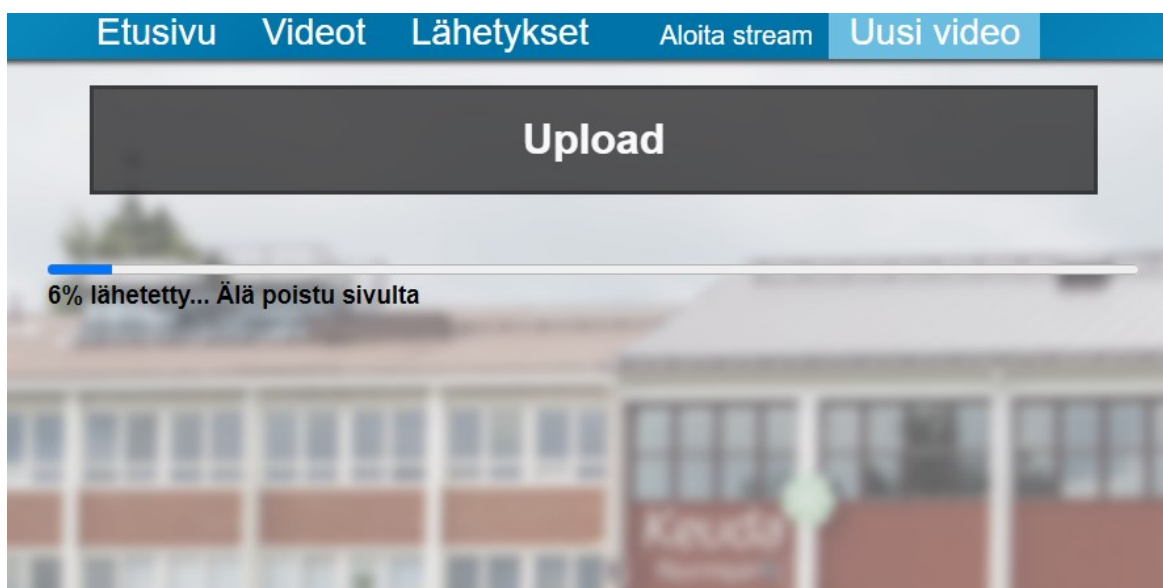
Seuraavaksi koodi tarkastaa tiedostokoon. Koska PHP ei pysty siirtämään suurempaa tiedostoa, kuin noin 2 Gigatavua, hylkään sitä suuremmat tiedostot heti. Viimeinen vertailu on tuo kirjainmerkkien määrä, minkä käyttäjä asetti Videon nimeksi. Tietysti vertailuihin kuuluu myös tyhjien arvojen hylkääminen.

Jokaiselle vertailijalle on toiminto, joka toteutuu, jos vertailu ei ole hyväksyttävä.

Hylkäystapauksissa PHP kirjaa websivulle tekstinä ilmoituksen, ettei tiedostoa voi hyväksyä, sekä syyn miksi se hylättiin. Tämän jälkeen käyttäjällä on mahdollisuus muokata valintojaan, jonka jälkeen hän voi koettaa uudelleen, jolloin koodi suorittaa vertailun uudelleen.

Jos asetukset hyväksytään, aloitetaan tiedoston siirto (kuva 42). Sivun muu sisältö piilotetaan, ja käyttäjälle esitetään etenemismittarilla sekä tekstinä tiedoston lähetyksen eteneminen. Samalla pyydetään, ettei käyttäjä poistuisi sivulta, koska PHP siirtoprosessi on sidottu kyseiselle sivulle. Jos yhteys sivulle katkeaa, katkeaa myös siirtoprosessi, mikä johtaa epätäydelliseen videotiedostoon palvelimella.

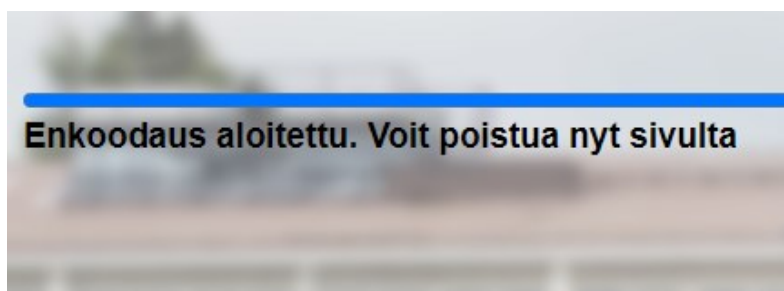
Kuva 42 Uuden videon lähetyksen aloitus onnistuneesti



Kun lähetys on valmistunut, PHP-koodi jatkaa seuraavaan kohtaansa (liite 1). Seuraavaa kohtaa varten teemme jälleen vertailun, lähetettiinkö video oletusarvoja käyttäen, vai tekikö lähettäjä edistyneitä valintoja. Vertailu ohjaa koodin jatkamaan kohdasta, jossa enkooderin prosessit käynnistetään ja videolle generoidaan uusi HTML-sivu.

PHP:n avulla voimme antaa palvelimen käyttöjärjestelmälle Windowsin komentorivikomentoja suoraan. Koodi käynnistää FFmpegin prosessin, jolla videosta muodostetaan ensin yhden videon sisältävän ruudun mukaan kuva, jota käytetään videon thumbnailina. Tämän jälkeen käynnistetään itse videoiden enkoodausprosessit jokaiselle videon valitulle laadulle. Prosessit käynnistetään Windowsin taustaprosessina, ettei IIS Application poolia sidota niihin ja pakoteta käyttäjää odottamaan sulkematta sivua ennen enkoodausten valmistumista. Enkoodausprosessien käynnistyttyä kerromme käyttäjälle, että on turvallista poistua sivulta (kuva 43).

Kuva 43 Videon lähetys onnistunut



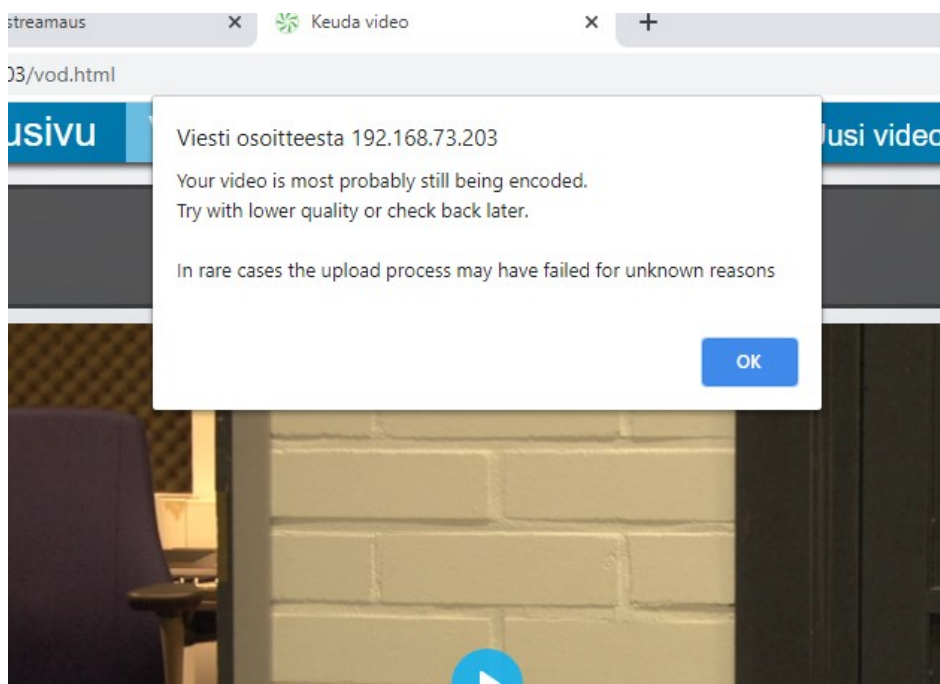
Sitä ennen koodissa kuitenkin generoitiin myös uusi sivu uudelle VODille (liite 1). Generoitu HTML-sivu (liite 5) sisältää uuden Plyr-toistimen instanssin luomisen JavaScriptilla, jossa asetetaan videon lähteiksi kaikki enkooderin luomat videotiedostot ja merkataan ne kyseisen resoluution mukaan soittimelle.

Kaiken tämän lisäksi sama koodi generoi vielä sisältöä olemassa olevaan videolista HTML tiedostoon, joka esitetään websivuston Videot sivulla iframessa. Käyttömukavuuden lisäämiseksi halusin, että uudet videot tulevat latauspäivämäärän alle ja ensimmäisenä ylhäältä ja vasemmalta katsottuna. Tätä varten PHP avaa koodissa videolista tiedoston ja navigoi tiedostossa riville, joka on sivun alussa, mutta kuitenkin oikean HTML elementin, eli bodyn, sisällä. Jotta tiedämme, tarvitseeko uutta päivämääräerottajaa lisätä, koodi tarkastaa

toisesta PHP tiedostosta päivämäärän ja sen, vastaako se tämän hetken päivämäärää. Jos ei vastaa, koodi tekee kaksi asiaa. Ensimmäisenä lisää videolistalle päivämäärän ja erottajan, ja sen jälkeen kirjaa uuden päivämäärän sen sisältämään PHP tiedostoon. (liite 1)

Nyt uudelle videolle on uusi sivu, sekä videolistalle on lisätty linkki siihen (kuva 34). Se tulee heti näkyviin videon lähetyksen onnistuttua, joten käyttäjällä oli mahdollisuus mennä sivulle, vaikkei video ollut vielä valmiina katseltavaksi. Tätä varten lisäsin ominaisuuden generoiduille sivuille, joka Plyr soittimen virheen yhteydessä ilmoittaa käyttäjälle HTML5 alert ominaisuudella, että video ei ole valmis, sekä miten tulisi toimia. Pienemmät tiedostot valmistuvat nopeasti ja siksi virheilmoitus kehottaa kokeilemaan toistoa heikommalla laadulla (kuva 44).

Kuva 44 Soittimen virheilmoitus



Tämänkin olisi voinut toteuttaa niin, ettei video tule näkyviin ennen kuin se on täysin valmis. Päädyin kuitenkin tähän ratkaisuun, että FFmpeg prosesseja seuraavan kuuntelijan ohjelmoiminen olisi ollut näin pieneen ongelmaan todella vaivalloinen ratkaisu. Toisaalta myös pidin siitä, että käyttäjä pääsee heti näkemään lähettämänsä videon sivustolla.

### 5.3.5 Videoiden arkistointi ja poisto

On kaksi syytä, miksi videoita tulee poistaa palvelimelta ja websivulta. Ensimmäinen syy on kyseenalainen materiaali. Palvelua tulevat harjoittelumielessä käyttämään paljon noin 16 – 18 vuotiaat oppilaat, joten on sanomattakin selvää, että kyseenalaisia videoita tulitisiin palvelimelle lähettämään. Websivulla tulee siis olla keino manuaalisesti poistaa sopimattomat videot.

Toinen syy on se, että palvelimen tallennustila on rajallinen. Itse asiassa se on hyvinkin rajallinen pidempää käyttöä ajatellen. Joudumme siis kehittämään automatisoidun tavan poistaa vanhat videot, sekä videoiden alkuperäiset tiedostot ennen niiden enkoodausta uusiin tiedostoihin. Koska kyseessä on harjoitteluympäristö, teemme käyttäjille myös web sivun ohjeistuksessa ilmi, että videot tullaan poistamaan, kun tarpeeksi aikaa on kulunut.

Manuaalisen poiston toteutin lisäämällä videon yhteyteen linkin, mikä siis on generoitu HTML pätkä, sivulle, joka käynnistää PHP-koodin. Tämä koodi sijaitsee erillisessä tiedostossa ja sillä suoritetaan videon poisto. PHP-koodille kerrotaan videon tunniste, jonka avulla koodi etsii videohakemistosta sitä vastaavan tiedoston ja poistaa sen. Tämän lisäksi koodi etsii myös videolista tiedostosta HTML pätkän, ja poistaa kyseiset tekstirivit. Puuttellisena ominaisuutena koodi ei toistaiseksi osaa poistaa päivämääräeroa. (liite 6)

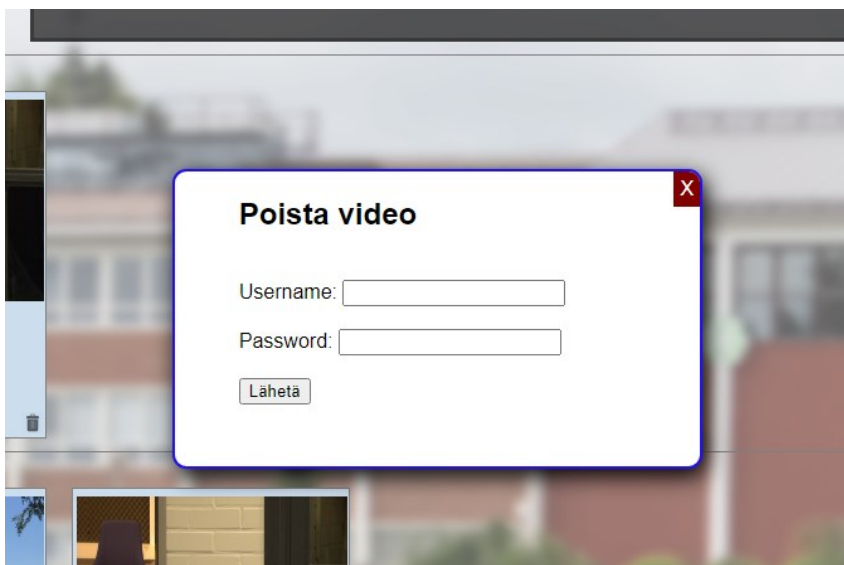
Tätä varten tarvitsimme myös oikeustason, jotta oppilaat eivät myöskään voisi ominaisuutta käyttää kiusallaan poistamaan toisten oppilaiden videoita. Tein päätöksen, että vain opettajilla on oikeus poistaa videoita. Tämä olisi ollut helppo ratkaista aktiivihakemiston basic autentichation tunnistautumisen avulla. Olisi vain asetettu PHP-koodi polkuun mikä vaatii tunnistautumisen, ja vain opettajilla olisi aktiivihakemiston käyttöoikeudet suorittaa tunnistautuminen hyväksytyksi.

Sain tässä vaiheessa kuitenkin medialinjan opettajalta pyynnön, että poistaminen voitaisiin suorittaa domainin ulkopuolelta, jota en täysin ymmärtänyt, sillä palveluihin ei pääse kuitenkaan käsiksi ulkoverkosta. Päätin silti toteuttaa toiveen.

Tätä varten tein siis PHP tiedoston, johon tein opettajille yleisen käyttäjän, tallentamalla tiedostoon vain kaksi muuttujaa, käyttäjänimi ja salasana. Pelkällä salasanallakin olisi sama

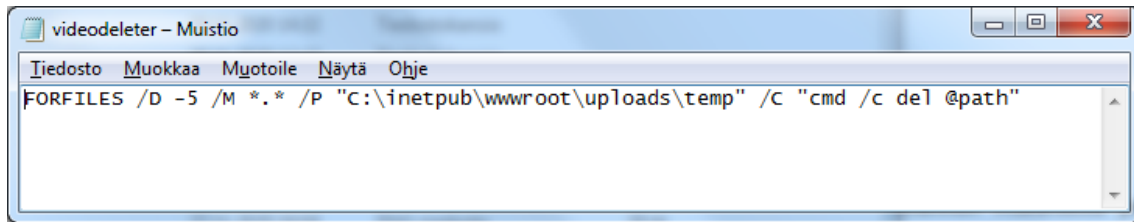
asia toteutunut, mutta toteutin asian silti näin. Videon poistolinkistä siis aukeaa ensin sivulla ikkuna, joka vaatii tunnistautumistiedot, ja hakee vertailukohteen syötetyille arvoille PHP tiedostosta, enne kuin jatkaa poistokoodin käynnistykseen (kuva 45). Käyttäjätunnuksen ja salasanan omaavaan PHP tiedostoa ei pääse tarkastelemaan webkäyttäjänä, sillä asetin IIS Managerissa siihen URL eston, ja lisäksi estin pääsyn tiedostoon Web user käyttäjäryhmältä. Palvelin siis pääsee sen avaamaan, mutta webkäyttäjä ei. Tämä ei todennäköisesti kovin tietoturvallinen ratkaisu siltikään ole.

Kuva 45 Videon poisto - tunnistautuminen



Vanhojen videoiden automaattisen poiston päätin toteuttaa vain Windowsin sisäisesti. Päätin ensin luoda Batch tiedoston, jossa suoritan vanhojen videoiden poiston annetuin parametrin. Batch tiedostot ovat Windowsin omia scriptitiedostoja, jossa selkotehtävillä annetaan Windowsin komentorivi- tai shell komentoja (Microsoft, 2009). Batch tiedostot ovat jääne Microsoft DOS käyttöjärjestelmästä, mutta koska windows on rakennettu DOS käyttöjärjestelmän päälle, ovat batch tiedostot täysin toimintakelpoisia nykyäänkin (Microsoft, 2009). Batch tiedostoja voi luoda itse helposti niiden yksinkertaisuuden vuoksi kaikilla tekstinkäsittelytyökaluilla, joten päätin tehdä omani windowsin omalla Notepad ohjelmalla. Pystyn yhdellä komennolla suorittamaan vanhojen videoide poiston (kuva 46). Tämän jälkeen tiedosto tallennetaan, ja vaihdetaan tiedostotyyppiä *.txt* päätteeksi *.bat* tiedostopäätteeksi.

Kuva 46 Ladattujen videoiden alkuperäiskappaleiden poistoskripti



Vanhojen videoiden poisto komennon mukaisesta polusta onnistuu siis ajamalla batchtiedosto, ikään kuin se olisi ohjelma, joka käynnistettäisiin. Se piti vain saada käynnistymään automaattisesti, säännöllisin väliajoin. Sen toteutukseen käytin hyödynni Windows Task Scheduler ominaisuutta. Task schedulerilla voi nimensä mukaisesti ajoittaa tapahtumia, jotka tapahtuvat annettujen ehtojen mukaisesti itsestään. Asetin siis Task schedulerilla joka sunnuntai toistuvan tapahtuman, mikä käynnistää tekemäni batch tiedoston. Valitsin sunnuntain, koska silloin ei luulisi olevan muita prosesseja tietokoneilla käynnissä, joten videoiden poisto ei tulisi haittaamaan suorituskykyä, kun palvelimella on käyttäjiä aktiivisena.

Tein siis erillisen batch skriptin ja ajoituksen sekä vanhoille videotiedostoille, että ladattujen videoiden alkuperäiskappaleelle. Syy miksi en poista alkuperäistiedostoja heti enkooderin valmistuttua suoraan PHP:lla on, että enkoodaus ajetaan taustaprosessina, tarkoittaen että prosessi pyörii taustalla omia aikojaan, valmistuen kun valmistuu. Siksi emme voi poistaa videoita suoraan websivujen PHP-komennoilla, tai joutuisimme edellyttämään videon lataajaa odottamaan sulkematta lataussivua kaikkien enkoodattavien versioiden ajan, joka oli todella huono käyttökokemus. Voisimme tietysti asettaa jonkin kuuntelijan tietokoneelle seuraamaan FFmpegin prosesseja, mutta ei se toiminnaltaan juurikaan parempi olisi manuaalisesti luodun ajastuksen kanssa. Teemme siis muutaman päivän vanhoille videoiden alkuperäiskappaleille poiston komennolla, jonka sijoitamme ajastetusti käynnistettävään batch tiedostoon (kuva 46).

Videoiden arkistoinnin toteutin edelleen samalla tavalla. Teen ajoitetun toiminnon, joka käynnistää skriptin, jolla tarpeeksi vanhat videot tulee siirrettyä toiselle kiintolevylle. Arkistokiintolevy oli tarkoitettu alkuperäisesti Streamien tallennustiedostoille, mutta koska emme voineet sitä toteuttaa, rakensin viime hetkillä kiintolevyn hyödynnykseksi arkisto-

ominaisuuden. Silti tallennustila oli rajallinen, joten videoiden poisto li tarpeellinen. kohteeksi asetettiin vain arkistoinnin polku.

Arkistoja itsessään pääsee selaamaan websivuilla olevan navigointipalkin oikeassa laidassa näkyvässä *arkisto* linkistä (kuva 35). Arkisto rakennetaan PHP-koodin lukiessa kansion sisältö ja generoimalla Arkiston HTML-sivulle listaelementillä listan kansion sisällöstä. Listausta voisi parantaa, sillä listalle merkitään kohteet niiden tiedostonimen mukaan, ja tiedostonimet eivät ole käyttäjäystävällisimmät. Ne kuitenkin sisältävät Videon nimen, joten käyttäjät saavat niistä halutessaan silti selvää (kuva 47).

Kuva 47 Arkisto



### Vanhat videotiedostot:

- [1080p\\_1583478497jymy.webm](#)
- [1080p\\_1583479133Paavoseikkailee.webm](#)
- [1080p\\_1583482782Karate.mp4](#)
- [1080p\\_1583489240aasi.mp4](#)
- [1080p\\_1583498074kink.mp4](#)
- [1440p\\_1583478497jymy.webm](#)
- [1440p\\_1583483156killeded.webm](#)
- [2160p\\_1583478497jymy.webm](#)
- [240p\\_1583479133Paavoseikkailee.webm](#)
- [240p\\_1583482782Karate.mp4](#)
- [240p\\_1583489240aasi.mp4](#)
- [240p\\_1583497206Yllätys.mp4](#)
- [240p\\_1583498074kink.mp4](#)

Arkiston sijaitessa toisella kiintolevyllä, missä itse websivu sijaitsee, tulee IIS:n asetuksista osoittaa polku arkistoon ja sallia käyttöoikeudet IIS:lle sekä webkäyttäjille.

## 6 LOPPUTULOS JA PARANNETTAVAA

Tosiasia on, että tämänkaltaisen palvelinkokonaisuuden kehittämistä voisi jatkaa niin pitkään kuin haluaa. Yhden miehen voimin kehittäminen vain olisi hidasta, ja aina joutuisi päivittämään ohjeistusta ylläpidosta vastaaville opiskelijoille. En siis oman mielipiteeni mukaan lähtisi huimasti enää harjoitteluympäristöön tarkoitetun palvelinkokonaisuuden palveluja lisäämään.

Jätän kappaleesta pois pienemmät kehityskohteet, jotka mainitsin niiden koskettavien kohtien yhteydessä aiemmin tekstissä. Nostan kuitenkin esille neljä seikkaa, jotka todella vaikuttavat työssä negatiivisesti, ja jotka muuttaisin, mikäli vain se olisi mahdollista.

Koko projektin viimeinen lopputulos jäi hieman hämärän peittoon sillä, poikkeusolot koronaviruksen vuoksi estivät palvelun oikean käyttöönoton, joten en tiedä miten oppilaat olisivat sen ottaneet vastaan. Olin suunnitellut myös kerääväni oppilaiden kokemuksen pohjalta palautetta siitä, mitä sivulla olisi voinut käyttökokemuksen parantamiseksi tehdä enemmän, taikka toisin. Tämä iso osa opinnäytetyöstä jäi nyt itsestä riippumattomista syistä toteuttamatta, joka jäi harmittamaan.

Kaikki työn tekniset tavoitteet kuitenkin saavutettiin, vaikka osa saavutettiin mielestäni hieman kyseenalaisin ratkaisuin, kuten streamausominaisuuksien karsiminen moottorin takia. Työn tilaaja kertoi olevansa tyytyväinen, vaikka pääsi vain pintapuolisesti kokeilemaan palveluita, joten työntilajankin palaute jäi vain muutamaa puhuttuun lauseeseen.

### 6.1 Palvelinkoneet ja SAN

Totuus yksinkertaisesti on, että annettu laitteisto ei sovellu mediapalvelimeksi. Tietokoneet tarvitsisivat paljon enemmän laskentatehoja usean videon yhtäaikaiseen käsittelyyn. Vaikka muodostin klusterin helpottamaan työtaakkaa, yksittäisten koneiden tehot ovat niin matalat, että liian alas jäämme silti. Jokainen klusterin node pystyi siedettävällä suorituskyvyllä käsittelemään yhtä videota kerrallaan, joka ei riitä, vaikka nodeja olisi monta.

Myöskään laitteistoa SAN-verkon muodostamiseen ei ollut lainkaan. Tämän tapainen mediapalvelin klustereineen vaatisi ehdottomasti keskitetyn tallennusalueen, joka olisi erillään muusta liikenteestä. Laitteistonkin olisi hyvä olla huippuluokkaa, sillä nopea, luotettava ja suuri sekä helposti kasvatettavissa oleva tallennustila olisi mediapalvelimelle todella tärkeää.

Kumpikaan asia ei siis ilman laitehankintaa onnistu, joten tätä työnkuvaa nämä seikat eivät edes varsinaisesti koske. Tunnen silti oleelliselta nostaa ne esille, mikäli uutta mediapalvelinta joskus kehittäisin.

## 6.2 Streamimoottori

Valitsisin Streamimoottorin paremmin, tai ostaisin Mistserverin tarjoamat palvelut. Työssä sovelletussa muodossaan se oli hyvin epäkäytännöllinen tarkoituksiinsa nähden. Harkitsisin myös streamausvaihtoehtoja hyödynyttäen FFmpegia, joka on käytössäni toiminut moitteetta sekä Linux-ympäristössä, että nyt Windows-ympäristössä.

Työ tuli toteutettua nyt kolmella vajavuudella, joista yksi oli dynaamisten streamien luonnin puute, jonka vuoksi palvelimia on epäkäytännöllistä ylläpitää. Toinen vajavuus oli adaptiivisen rakenteen puuttuminen, joka olisi lähes ehdoton käyttäjäystävällisyyden edistämiseksi. Viimeinen vajavuus oli se, ettei moottorilla voinut suorittaa lähetysten tallennusta, vaan se piti jättää käyttäjän vastuulle.

Muutoin ilman maksumuuria Mistserver tuntui kelvolliselta palvelulta, etenkin jos vertaan sitä samankaltaiseen, täysin maksulliseen Wowza Streaming engineen. En voi tuomita Mistserveriä täysin, sillä ominaisuudet palvelulla vaikuttavat täysin kilpailukykyiseltä Wowzan kanssa.

## 6.3 Videon lataus palvelimeen

Videon lataukseen tulisi ehdottomasti käyttää jotain muuta keinoa kuin PHP:ta. PHP:lla on aivan hyvä siirtää pieniä tiedostoja, mutta Videotiedostojen ollessa kyseessä, se ei sovellu lainkaan. 2GB tiedostoraja on aivan liian pieni, sillä palvelimelle voitaisiin haluta ladata

esimerkiksi esipakkaamatonta materiaalia kuvattuna suurella resoluutiolla, esimerkiksi 360 kameralla tai 4K kameralla. Lyhytkin pätkä tämänlaista materiaalia nostaa tiedostokoon äkkiä yli tuon rajan. Puhumattakaan pidemmistä, esimerkiksi puolen tunnin, tai sen yli, kestävästä videoista pienemmälläkin resoluutiolla.

Se tulisi ratkaista esimerkiksi FTP-palvelimella, mutta FTP-palvelimen käyttöönotto websivulla tuo mukanaan omia ongelmiaan. Ne olisivat silti vaivan arvoiset, että tuo sallittu tiedostokoko saataisiin ylös.

## Lähteet

Barrie, S. (2009). *Networking Bible*. John Wiley & Sons.

Bauer, R. (27.11.2018). *Back Blaze*. Noudettu osoitteesta backbalze:

<https://www.backblaze.com/blog/whats-the-diff-nas-vs-san/>

Bhagat, V. (8.8.2019). *PixelCrayons*. Noudettu osoitteesta

<https://www.pixelcrayons.com/blog/php-vs-asp-net-how-to-choose-the-right-one/>

Brown, L. (27.11.2020). *filmora*. Noudettu osoitteesta

<https://filmora.wondershare.com/video-editing-tips/what-is-video-bitrate.html>

DDVTech. (n.d). *Mistserver*. Noudettu osoitteesta about us: <https://mistserver.com/aboutus>

expert-advice. (14.12.2017). Noudettu osoitteesta <https://expert-advice.org/windows-server/how-to-set-up-auto-login-windows-server-2012-and-2016/>

FFmpeg. (n.d). *Ffmpeg downloads*. Noudettu osoitteesta

<https://ffmpeg.org/download.html#build-windows>

Hewlett Packard. (2010). *Server Blade G7 User guide*. Noudettu osoitteesta

<https://psnow.ext.hpe.com/doc/c02498009>

Hodge, K. (25.7.2017). *medium*. Noudettu osoitteesta How to use javascrip library:

<https://medium.com/front-end-weekly/how-to-use-a-javascript-library-jquery-75d0b6d50b5a>

HPESupport. (2011). *Intallation instructions*. Noudettu osoitteesta

<https://cdn.cnetcontent.com/74/67/74675541-5d2b-488c-b9f4-92f37f0b9218.pdf>

Jenkins, R. (2020). *dacast.com*. Noudettu osoitteesta <https://www.dacast.com/blog/what-is-video-encoding/>

Kameshwaran. (27.2.2017). *Vembu*. Noudettu osoitteesta

<https://www.vembu.com/blog/installing-the-failover-cluster-feature-in-windows-server-2016/>

- Keuda. (n.d). *Tietoa Keudasta*. Noudettu osoitteesta <https://www.keuda.fi/tietoa-keudasta/tietoa-keudasta/>
- Kim, B. (4.2.2016). *The survival mom*. Noudettu osoitteesta <https://thesurvivalmom.com/understanding-two-is-one-one-none-concept/>
- Loges. (27.9.2017). *Assistanz*. Noudettu osoitteesta <https://www.assistanz.com/steps-to-install-php-manually-on-windows-2016-server/>
- Mistserver Manual. (7.10.2020). Noudettu osoitteesta [https://mistserver.org/guides/MistServer\\_Manual\\_2.18.pdf](https://mistserver.org/guides/MistServer_Manual_2.18.pdf)
- MDN docs. (n.d -a). *Developer Mozilla*. Noudettu osoitteesta html5: <https://developer.mozilla.org/en-US/docs/Web/Guide/HTML/HTML5>
- MDN docs. (n.d -b). *mozilla developer*. Noudettu osoitteesta javascript: [https://developer.mozilla.org/en-US/docs/Learn/JavaScript/First\\_steps/What\\_is\\_JavaScript](https://developer.mozilla.org/en-US/docs/Learn/JavaScript/First_steps/What_is_JavaScript)
- MDN docs. (n.d -c). *Mozilla developer*. Noudettu osoitteesta [https://developer.mozilla.org/en-US/docs/Web/HTTP/Basics\\_of\\_HTTP/MIME\\_types](https://developer.mozilla.org/en-US/docs/Web/HTTP/Basics_of_HTTP/MIME_types)
- Microsoft. (8.8.2009). *Microsoft docs*. Noudettu osoitteesta [https://docs.microsoft.com/en-us/previous-versions/windows/it-pro/windows-server-2003/cc758944\(v=ws.10\)?redirectedfrom=MSDN](https://docs.microsoft.com/en-us/previous-versions/windows/it-pro/windows-server-2003/cc758944(v=ws.10)?redirectedfrom=MSDN)
- Microsoft. (2012). *Microsoft docs*. Noudettu osoitteesta [https://docs.microsoft.com/en-us/previous-versions/windows/it-pro/windows-server-2008-r2-and-2008/cc770946\(v=ws.10\)](https://docs.microsoft.com/en-us/previous-versions/windows/it-pro/windows-server-2008-r2-and-2008/cc770946(v=ws.10))
- Microsoft. (26.9.2016). *Microsoft docs*. Noudettu osoitteesta <https://docs.microsoft.com/en-us/iis/configuration/system.applicationhost/applicationpools/>
- Microsoft. (2018). *Docs*. Noudettu osoitteesta <https://docs.microsoft.com/en-us/windows-server/failover-clustering/whats-new-in-failover-clustering>
- Nelson. (2016). *IT tutorials*. Noudettu osoitteesta <https://ittutorials.net/microsoft/windows-server-2016/active-directory/>

Ozer, J. (12.2.2019). *Streaming media*. Noudettu osoitteesta

<https://www.streamingmedia.com/Articles/Editorial/What-Is-.../What-Is-Streaming-74052.aspx>

Perkins, C. (2016). *Devolutions forum*. Noudettu osoitteesta

<https://forum.devolutions.net/topics/24744/hp-ilo-web-remote-console-net-fails-to-open>

perspectives. (27.1.2012). *Hil Te*. Noudettu osoitteesta

<https://hilite.org/19749/perspectives/pros-and-cons-of-a-new-program-student-guided-education-is-optimal-but-only-under-the-right-conditions/>

Petters, J. (17.6.2020). *Varonis*. Noudettu osoitteesta varonis blog:

<https://www.varonis.com/blog/active-directory-forest/>

Pfister, G. F. (1998). *In search of clusters*. Prentice Hall PTR.

PHP Manual. (n.d). *PHP documentation*. Noudettu osoitteesta

<https://www.php.net/manual/en/features.file-upload.php>

Potts, S. (n.d). *github*. Noudettu osoitteesta plyr: <https://github.com/sampotts/plyr>

Rodrigues, S. (2015). *Mirazon*. Noudettu osoitteesta <https://www.mirazon.com/how-does-windows-10-licensing-and-activation-work/>

Rouse, M. (2012). *Techtarget*. Noudettu osoitteesta

<https://searchdatacenter.techtarget.com/definition/Single-point-of-failure-SPOF>

Ruether, T. (21.7.2020). *Wowza media systems*. Noudettu osoitteesta

<https://www.wowza.com/blog/rtmp-streaming-real-time-messaging-protocol>

Tonias, D. (14.1.2018). *DTonias*. Noudettu osoitteesta <https://www.dtonias.com/add-another-domain-controller-active-directory/>

Vincent. (2019). *Techbast*. Noudettu osoitteesta <https://techbast.com/2019/07/windows-server-installation-and-configuration-of-two-domain-controllers-running-in-parallel.html>

*w3schools*. (n.d). Noudettu osoitteesta html elements:

[https://www.w3schools.com/html/html\\_classes.asp](https://www.w3schools.com/html/html_classes.asp)

Wikipedia. (n.d). *ffmpeg*. Noudettu osoitteesta <https://en.wikipedia.org/wiki/FFmpeg>

## Liite 1: VIDEON LATAUS PALVELIMEEN, SEKÄ VIDEON ENKODERIN KÄYNNISTYS JA HTML-SIVUN GENEROIMINEN – PHP-KOODI

```
<?php
error_reporting(E_ALL);
ini_set('display_errors','on');
$fileType = $_FILES["file"]["type"]; // The type of file it is
$fileSize = $_FILES["file"]["size"]; // File size in bytes
$fileErrorMsg = $_FILES["file"]["error"]; // 0 for false... and 1 for true
$h = $_POST["pn"];
$check240 = $_POST["php240"];
$check360 = $_POST["php360"];
$check480 = $_POST["php480"];
$check576 = $_POST["php576"];
$check720 = $_POST["php720"];
$check1080 = $_POST["php1080"];
$check1440 = $_POST["php1440"];
$check2160 = $_POST["php2160"];
$customposter = $_POST["customposter"];
$codec = $_POST["codec"];
$profile = $_POST["profile"];
$profilestring = "";
if($profile == "high"){ $profilestring = "-profile:v high ";}
else if($profile == "main"){ $profilestring = "-profile:v main ";}
else if($profile == "baseline"){ $profilestring = "-profile:v baseline ";}
else { $profilestring = ""; }
$paate = ".mp4";
if($codec == "libx264"){ $paate = ".mp4";}
if($codec == "libvpx-vp9"){ $paate = ".webm";}
if($codec == "libxvid"){ $paate = ".avi";}
echo $h;
$storename = time() . "_" . basename( $_FILES['file']['name']);
$storename = str_replace(' ', '', $storename);
$encoded = time() . $h . $paate;
$encoded = str_replace(' ', '', $encoded);
$custompostername = time() . "_" . basename( $_FILES['poster']['name']);
$custompostername = str_replace(' ', '', $custompostername);
$path = "";
$posterpath = "uploads/" . $custompostername;
$today = date("d.m.y");
include "date.php";
```

```

if($check240 == "true" || $check360 == "true" || $check480 == "true" || $check576 == "true" || $check720 == "true" ||
$check1080 == "true" || $check1440 == "true" || $check2160 == "true"){
    if(!empty($_FILES['file'])) {
        if(preg_match('#^video/.*$#',$FILES["file"]["type"]) && ($FILES["file"]["size"] < 99900000000)) {
            $path = "uploads/temp/" . $storename;
            if(move_uploaded_file($_FILES['file']['tmp_name'], $path)) {
                $htmlstring = "<!DOCTYPE html> \r\n" .
                    "<html> \r\n" .
                    "<head> \r\n" .
                    "<meta charset='utf-8'> \r\n" .
                    "<style>html, body{margin:0;min-height:100%;font-family: Arial;}body{padding:0px 40px;text-align:center;}.container
{max-width: 72%;margin: 0 auto;}.plyr {border-radius: 7px;margin-bottom: 15px;}#otsikko{display:block;color: white;background-
color: #222;margin-left: 20%;margin-right: 20%;margin-bottom: 10px;margin-top: 0;padding: 20px 30px;overflow:hidden;border:
3px solid black;opacity:0.75;position:relative;z-index:2;}</style> \r\n" .
                    "<link rel='stylesheet' href='plyr.css'> \r\n" .
                    "<script src='plyr.js'></script> \r\n" .
                    "</head> \r\n" .
                    "<body> \r\n" .
                    "<h1 id='otsikko'>" . $h . "</h1> \r\n" .
                    "<div class='container'> \r\n";
                if($customposter == "true"){
                    $htmlstring = $htmlstring . "<video id='player' controls playsinline poster='/uploads/" . $custompostername . "'> \r\n";
                    move_uploaded_file($_FILES['poster']['tmp_name'], $posterpath);
                }else {
                    $htmlstring = $htmlstring . "<video id='player' controls playsinline poster='/uploads/" . $encoded . ".png'> \r\n";
                }
                if($codec == "libx264"){
                    if($check240 == "true"){ $htmlstring = $htmlstring . "<source src='/uploads/240p_". $encoded . "' type='video/mp4'
size='240'> \r\n"; }
                    if($check360 == "true"){ $htmlstring = $htmlstring . "<source src='/uploads/360p_". $encoded . "' type='video/mp4'
size='360'> \r\n"; }
                    if($check480 == "true"){ $htmlstring = $htmlstring . "<source src='/uploads/480p_". $encoded . "' type='video/mp4'
size='480'> \r\n"; }
                    if($check576 == "true"){ $htmlstring = $htmlstring . "<source src='/uploads/576p_". $encoded . "' type='video/mp4'
size='576'> \r\n"; }
                    if($check720 == "true"){ $htmlstring = $htmlstring . "<source src='/uploads/720p_". $encoded . "' type='video/mp4'
size='720'> \r\n"; }
                    if($check1080 == "true"){ $htmlstring = $htmlstring . "<source src='/uploads/1080p_". $encoded . "' type='video/mp4'
size='1080'> \r\n"; }
                    if($check1440 == "true"){ $htmlstring = $htmlstring . "<source src='/uploads/1440p_". $encoded . "' type='video/mp4'
size='1440'> \r\n"; }

```

```

        if($check2160 == "true"){ $htmlstring = $htmlstring . "<source src='/uploads/2160p_". $encoded . "' type='video/mp4'
size='2160'> \r\n"; }
        }if($codec == "libvpx-vp9"){
            if($check240 == "true"){ $htmlstring = $htmlstring . "<source src='/uploads/240p_". $encoded . "' type='video/webm'
size='240'> \r\n"; }
            if($check360 == "true"){ $htmlstring = $htmlstring . "<source src='/uploads/360p_". $encoded . "' type='video/webm'
size='360'> \r\n"; }
            if($check480 == "true"){ $htmlstring = $htmlstring . "<source src='/uploads/480p_". $encoded . "' type='video/webm'
size='480'> \r\n"; }
            if($check576 == "true"){ $htmlstring = $htmlstring . "<source src='/uploads/576p_". $encoded . "' type='video/webm'
size='576'> \r\n"; }
            if($check720 == "true"){ $htmlstring = $htmlstring . "<source src='/uploads/720p_". $encoded . "' type='video/webm'
size='720'> \r\n"; }
            if($check1080 == "true"){ $htmlstring = $htmlstring . "<source src='/uploads/1080p_". $encoded . "' type='video/webm'
size='1080'> \r\n"; }
            if($check1440 == "true"){ $htmlstring = $htmlstring . "<source src='/uploads/1440p_". $encoded . "' type='video/webm'
size='1440'> \r\n"; }
            if($check2160 == "true"){ $htmlstring = $htmlstring . "<source src='/uploads/2160p_". $encoded . "' type='video/webm'
size='2160'> \r\n"; }
        }if($codec == "libxvid"){
            if($check240 == "true"){ $htmlstring = $htmlstring . "<source src='/uploads/240p_". $encoded . "' type='video/avi'
size='240'> \r\n"; }
            if($check360 == "true"){ $htmlstring = $htmlstring . "<source src='/uploads/360p_". $encoded . "' type='video/avi'
size='360'> \r\n"; }
            if($check480 == "true"){ $htmlstring = $htmlstring . "<source src='/uploads/480p_". $encoded . "' type='video/avi'
size='480'> \r\n"; }
            if($check576 == "true"){ $htmlstring = $htmlstring . "<source src='/uploads/576p_". $encoded . "' type='video/avi'
size='576'> \r\n"; }
            if($check720 == "true"){ $htmlstring = $htmlstring . "<source src='/uploads/720p_". $encoded . "' type='video/avi'
size='720'> \r\n"; }
            if($check1080 == "true"){ $htmlstring = $htmlstring . "<source src='/uploads/1080p_". $encoded . "' type='video/avi'
size='1080'> \r\n"; }
            if($check1440 == "true"){ $htmlstring = $htmlstring . "<source src='/uploads/1440p_". $encoded . "' type='video/avi'
size='1440'> \r\n"; }
            if($check2160 == "true"){ $htmlstring = $htmlstring . "<source src='/uploads/2160p_". $encoded . "' type='video/avi'
size='2160'> \r\n"; }
        }
        $htmlstring = $htmlstring . "</video> \r\n" .
        "<a href='videolista.html' style='background-color: blue; border-radius: 12px 2px 2px 12px; text-decoration:none;
padding: 5px; color: white; display: block; float:right;'>Takaisin</a> \r\n" .
        "</div> \r\n" .
        "<script> \r\n" .
        "document.addEventListener('DOMContentLoaded', () => { \r\n" .

```

```

"const player = new Plyr('#player'); \r\n" .
"window.player = player; \r\n" .
"function on(selector, type, callback) {document.querySelector(selector).addEventListener(type, callback, false)};};
\r\n" .
"player.addEventListener('error', function() {alert(\"Your video is most probably still being encoded. \nTry with lower
quality or check back later.\n\nIn rare cases the upload process may have failed for unknown reasons\");}); \r\n" .
"</script> \r\n" .
"</body> \r\n" .
"</html> \r\n";
fwrite(fopen('uploads/' . $encoded . '.html','w'), $htmlstring);
$lista = "uploads/videolista.html";
if($customposter == "true"){
    $htmllinkki = "<span id=\"" . $encoded . "_1'></span>\n<div class='vidi'><img src='delete.png' alt='delete video'
width='18px' height='19px' onclick='del(\"" . $encoded . "\")' style='position: relative; top: 250px; left: 99px; opacity:0.6; z-index:
8;'><a style='text-decoration: none;' href='/uploads/" . $encoded . ".html'><div style='position: relative; top: -22px; z-index:
7;'><img class='t' src=\"" . $custompostername . "' alt='thumbnail'><span id=\"" . $encoded . "'>\" . $h .
"</span></div></a></div>\n<span id=\"" . $encoded . "_2'></span>";
}
else{
    $htmllinkki = "<span id=\"" . $encoded . "_1'></span>\n<div class='vidi'><img src='delete.png' alt='delete video'
width='18px' height='19px' onclick='del(\"" . $encoded . "\")' style='position: relative; top: 250px; left: 99px; opacity:0.6; z-index:
8;'><a style='text-decoration: none;' href='/uploads/" . $encoded . ".html'><div style='position: relative; top: -22px; z-index:
7;'><img class='t' src=\"" . $encoded . ".png' alt='thumbnail'><span id=\"" . $encoded . "'>\" . $h . "</span></div></a></div>\n<span
id=\"" . $encoded . "_2'></span>";
}
$tuonne = file($lista, FILE_IGNORE_NEW_LINES);
if($today != $date){
    $newdate = "<?php \$date = \"\$today\"; ?>";
    file_put_contents("date.php",$newdate);
    array_splice($tuonne, 15, 0, "<div class='pvm'>" . $today . "</div>");
    file_put_contents($lista, join("\r\n",$tuonne));
}
array_splice($tuonne, 16, 0, $htmllinkki);
file_put_contents($lista, join("\r\n",$tuonne));

echo $storename;
shell_exec("ffmpeg -i c:\\inetpub\\wwwroot\\uploads\\temp\\" . $storename . " -s 858x480 -ss 00:00:01 -vframes 1
c:\\inetpub\\wwwroot\\uploads\\" . $encoded . ".png");
if($codec == "libx264"){
    $c1 = "ffmpeg -i c:\\inetpub\\wwwroot\\uploads\\temp\\" . $storename . " -s 352x240 -c:v libx264 " . $profilestring . "-
b:v 350k c:\\inetpub\\wwwroot\\uploads\\240p_" . $encoded . " > NUL 2>&1 < NUL";
    $c2 = "ffmpeg -i c:\\inetpub\\wwwroot\\uploads\\temp\\" . $storename . " -s 858x480 -c:v libx264 " . $profilestring . "-
b:v 1200k c:\\inetpub\\wwwroot\\uploads\\480p_" . $encoded . " > NUL 2>&1 < NUL";
}

```

```

    $c3 = "ffmpeg -i c:\\inetpub\\wwwroot\\uploads\\temp\\" . $storename . " -s 1280x720 -c:v libx264 " . $profilestring .
"-b:v 3000k c:\\inetpub\\wwwroot\\uploads\\720p_" . $encoded . " > NUL 2>&1 < NUL";
    $c4 = "ffmpeg -i c:\\inetpub\\wwwroot\\uploads\\temp\\" . $storename . " -s 1920x1080 -c:v libx264 " . $profilestring .
"-b:v 6000k c:\\inetpub\\wwwroot\\uploads\\1080p_" . $encoded . " > NUL 2>&1 < NUL";
    $c5 = "ffmpeg -i c:\\inetpub\\wwwroot\\uploads\\temp\\" . $storename . " -s 640x360 -c:v libx264 " . $profilestring . "-
b:v 750k c:\\inetpub\\wwwroot\\uploads\\360p_" . $encoded . " > NUL 2>&1 < NUL";
    $c6 = "ffmpeg -i c:\\inetpub\\wwwroot\\uploads\\temp\\" . $storename . " -s 768x576 -c:v libx264 " . $profilestring . "-
b:v 2200k c:\\inetpub\\wwwroot\\uploads\\576p_" . $encoded . " > NUL 2>&1 < NUL";
    $c7 = "ffmpeg -i c:\\inetpub\\wwwroot\\uploads\\temp\\" . $storename . " -s 2560x1440 -c:v libx264 " . $profilestring .
"-b:v 12000k c:\\inetpub\\wwwroot\\uploads\\1440p_" . $encoded . " > NUL 2>&1 < NUL";
    $c8 = "ffmpeg -i c:\\inetpub\\wwwroot\\uploads\\temp\\" . $storename . " -s 3840x2160 -c:v libx264 " . $profilestring .
"-b:v 25000k c:\\inetpub\\wwwroot\\uploads\\2160p_" . $encoded . " > NUL 2>&1 < NUL";
}
else if($codec == "libvpx-vp9"){
    $c1 = "ffmpeg -i c:\\inetpub\\wwwroot\\uploads\\temp\\" . $storename . " -s 352x240 -c:v " . $codec . " " .
$profilestring . "-b:v 200k c:\\inetpub\\wwwroot\\uploads\\240p_" . $encoded . " > NUL 2>&1 < NUL";
    $c2 = "ffmpeg -i c:\\inetpub\\wwwroot\\uploads\\temp\\" . $storename . " -s 858x480 -c:v " . $codec . " " .
$profilestring . "-b:v 750k c:\\inetpub\\wwwroot\\uploads\\480p_" . $encoded . " > NUL 2>&1 < NUL";
    $c3 = "ffmpeg -i c:\\inetpub\\wwwroot\\uploads\\temp\\" . $storename . " -s 1280x720 -c:v " . $codec . " " .
$profilestring . "-b:v 2000k c:\\inetpub\\wwwroot\\uploads\\720p_" . $encoded . " > NUL 2>&1 < NUL";
    $c4 = "ffmpeg -i c:\\inetpub\\wwwroot\\uploads\\temp\\" . $storename . " -s 1920x1080 -c:v " . $codec . " " .
$profilestring . "-b:v 3200k c:\\inetpub\\wwwroot\\uploads\\1080p_" . $encoded . " > NUL 2>&1 < NUL";
    $c5 = "ffmpeg -i c:\\inetpub\\wwwroot\\uploads\\temp\\" . $storename . " -s 640x360 -c:v " . $codec . " " .
$profilestring . "-b:v 750k c:\\inetpub\\wwwroot\\uploads\\360p_" . $encoded . " > NUL 2>&1 < NUL";
    $c6 = "ffmpeg -i c:\\inetpub\\wwwroot\\uploads\\temp\\" . $storename . " -s 768x576 -c:v " . $codec . " " .
$profilestring . "-b:v 2200k c:\\inetpub\\wwwroot\\uploads\\576p_" . $encoded . " > NUL 2>&1 < NUL";
    $c7 = "ffmpeg -i c:\\inetpub\\wwwroot\\uploads\\temp\\" . $storename . " -s 2560x1440 -c:v " . $codec . " " .
$profilestring . "-b:v 9500k c:\\inetpub\\wwwroot\\uploads\\1440p_" . $encoded . " > NUL 2>&1 < NUL";
    $c8 = "ffmpeg -i c:\\inetpub\\wwwroot\\uploads\\temp\\" . $storename . " -s 3840x2160 -c:v " . $codec . " " .
$profilestring . "-b:v 20000k c:\\inetpub\\wwwroot\\uploads\\2160p_" . $encoded . " > NUL 2>&1 < NUL";
}
else if($codec == "libxvid"){
    $c1 = "ffmpeg -i c:\\inetpub\\wwwroot\\uploads\\temp\\" . $storename . " -s 352x240 -c:v " . $codec . " " .
$profilestring . "-qscale:v 1 c:\\inetpub\\wwwroot\\uploads\\240p_" . $encoded . " > NUL 2>&1 < NUL";
    $c2 = "ffmpeg -i c:\\inetpub\\wwwroot\\uploads\\temp\\" . $storename . " -s 858x480 -c:v " . $codec . " " .
$profilestring . "-qscale:v 15 c:\\inetpub\\wwwroot\\uploads\\480p_" . $encoded . " > NUL 2>&1 < NUL";
    $c3 = "ffmpeg -i c:\\inetpub\\wwwroot\\uploads\\temp\\" . $storename . " -s 1280x720 -c:v " . $codec . " " .
$profilestring . "-qscale:v 15 c:\\inetpub\\wwwroot\\uploads\\720p_" . $encoded . " > NUL 2>&1 < NUL";
    $c4 = "ffmpeg -i c:\\inetpub\\wwwroot\\uploads\\temp\\" . $storename . " -s 1920x1080 -c:v " . $codec . " " .
$profilestring . "-qscale:v 6 c:\\inetpub\\wwwroot\\uploads\\1080p_" . $encoded . " > NUL 2>&1 < NUL";
    $c5 = "ffmpeg -i c:\\inetpub\\wwwroot\\uploads\\temp\\" . $storename . " -s 640x360 -c:v " . $codec . " " .
$profilestring . "-qscale:v 15 c:\\inetpub\\wwwroot\\uploads\\360p_" . $encoded . " > NUL 2>&1 < NUL";
    $c6 = "ffmpeg -i c:\\inetpub\\wwwroot\\uploads\\temp\\" . $storename . " -s 768x576 -c:v " . $codec . " " .
$profilestring . "-qscale:v 15 c:\\inetpub\\wwwroot\\uploads\\576p_" . $encoded . " > NUL 2>&1 < NUL";
}

```

```

    $c7 = "ffmpeg -i c:\\inetpub\\wwwroot\\uploads\\temp\\" . $storename . " -s 2560x1440 -c:v " . $codec . " " .
$profilestring . "-qscale:v 6 c:\\inetpub\\wwwroot\\uploads\\1440p_" . $encoded . " > NUL 2>&1 < NUL";
    $c8 = "ffmpeg -i c:\\inetpub\\wwwroot\\uploads\\temp\\" . $storename . " -s 3840x2160 -c:v " . $codec . " " .
$profilestring . "-qscale:v 6 c:\\inetpub\\wwwroot\\uploads\\2160p_" . $encoded . " > NUL 2>&1 < NUL";
}
if($check240 == "true"){pclose(popen("START /B $c1", 'r'));}
if($check480 == "true"){pclose(popen("START /B $c2", 'r'));}
if($check720 == "true"){pclose(popen("START /B $c3", 'r'));}
if($check1080 == "true"){pclose(popen("START /B $c4", 'r'));}
if($check360 == "true"){pclose(popen("START /B $c5", 'r'));}
if($check576 == "true"){pclose(popen("START /B $c6", 'r'));}
if($check1440 == "true"){pclose(popen("START /B $c7", 'r'));}
if($check2160 == "true"){pclose(popen("START /B $c8", 'r'));}
}
} else {
    echo "please select a video file";
}
} else {
    echo "Something went wrong, please try again later";
}
} else {
    if(!empty($_FILES['file'])) {
        if(preg_match('#^video/. *$#', $_FILES['file']['type']) && ($_FILES['file']['size'] < 99900000000)) {
            $path = "uploads/temp/" . $storename;
            if(move_uploaded_file($_FILES['file']['tmp_name'], $path)) {
                $htmlstring = "<!DOCTYPE html> \r\n" .
                    "<html> \r\n" .
                    "<head> \r\n" .
                    "<meta charset='utf-8'> \r\n" .
                    "<style>html, body{margin:0;min-height:100%;font-family: Arial;}body{padding:0px 40px;text-align:center;}.container
{max-width: 72%;margin: 0 auto;}.plyr {border-radius: 7px;margin-bottom: 15px;}#otsikko{display:block;color: white;background-
color: #222;margin-left: 20%;margin-right: 20%;margin-bottom: 10px;margin-top: 0;padding: 20px 30px;overflow:hidden;border:
3px solid black;opacity:0.75;position:relative;z-index:2;}</style> \r\n" .
                    "<link rel='stylesheet' href='plyr.css'> \r\n" .
                    "<script src='plyr.js'></script> \r\n" .
                    "</head> \r\n" .
                    "<body> \r\n" .
                    "<h1 id='otsikko'>" . $h . "</h1> \r\n" .
                    "<div class='container'> \r\n" .
                    "<video id='player' controls playsinline poster='/uploads/" . $encoded . ".png'> \r\n" .
                    "<source src='/uploads/240p_" . $encoded . "' type='video/mp4' size='240'> \r\n" .
                    "<source src='/uploads/480p_" . $encoded . "' type='video/mp4' size='480'> \r\n" .

```

```

"<source src='/uploads/720p_". $encoded . "' type='video/mp4' size='720'> \r\n" .
"<source src='/uploads/1080p_". $encoded . "' type='video/mp4' size='1080'> \r\n" .
"</video> \r\n" .
"<a href='videolista.html' style='background-color: blue; border-radius: 12px 2px 2px 12px; text-decoration:none;
padding: 5px; color: white; display: block; float:right;'>Takaisin</a> \r\n" .
"</div> \r\n" .
"<script> \r\n" .
"document.addEventListener('DOMContentLoaded', () => { \r\n" .
"const player = new Plyr('#player'); \r\n" .
"window.player = player; \r\n" .
"function on(selector, type, callback) {document.querySelector(selector).addEventListener(type, callback, false);}};
\r\n" .
"player.addEventListener('error', function() {alert(\"Your video is most probably still being encoded. \n\nTry with lower
quality or check back later.\n\nIn rare cases the upload process may have failed for unknown reasons\");}); \r\n" .
"</script> \r\n" .
"</body> \r\n" .
"</html> \r\n";
fwrite(fopen('uploads/' . $encoded . '.html','w'), $htmlstring);
$lista = "uploads/videolista.html";
$htmllinkki = "<span id='\" . $encoded . \"_1'></span>\n<div class='vidi'><img src='delete.png' alt='delete video'
width='18px' height='19px' onclick='del(\"\" . $encoded . \"\")' style='position: relative; top: 250px; left: 99px; opacity:0.6; z-index:
8;'><a style='text-decoration: none;' href='/uploads/\" . $encoded . \"_1.html'><div style='position: relative; top: -22px; z-index:
7;'><img class='t' src='\" . $encoded . \"_1.png' alt='thumbnail'><span id='\" . $encoded . \"_1'>\" . $h . \"</span></div></a></div>\n<span
id='\" . $encoded . \"_2'></span>";
$tuonne = file($lista, FILE_IGNORE_NEW_LINES);
if($today != $date){
    $newdate = "<?php \$date = \"\$today\"; ?>";
    file_put_contents("date.php",$newdate);
    array_splice($tuonne, 15, 0, "<div class='pvm'>\" . $today . \"</div>");
    file_put_contents($lista, join("\r\n",$tuonne));
}
array_splice($tuonne, 16, 0, $htmllinkki);
file_put_contents($lista, join("\r\n",$tuonne));

echo $storename;
shell_exec("ffmpeg -i c:\\inetpub\\wwwroot\\uploads\\temp\\" . $storename . " -s 858x480 -ss 00:00:01 -vframes 1
c:\\inetpub\\wwwroot\\uploads\\" . $encoded . ".png");
$c1 = "ffmpeg -i c:\\inetpub\\wwwroot\\uploads\\temp\\" . $storename . " -s 352x240 -c:v libx264 -b:v 200k
c:\\inetpub\\wwwroot\\uploads\\240p_". $encoded . " > NUL 2>&1 < NUL";
$c2 = "ffmpeg -i c:\\inetpub\\wwwroot\\uploads\\temp\\" . $storename . " -s 858x480 -c:v libx264 -b:v 1000k
c:\\inetpub\\wwwroot\\uploads\\480p_". $encoded . " > NUL 2>&1 < NUL";
$c3 = "ffmpeg -i c:\\inetpub\\wwwroot\\uploads\\temp\\" . $storename . " -s 1280x720 -c:v libx264 -b:v 2500k
c:\\inetpub\\wwwroot\\uploads\\720p_". $encoded . " > NUL 2>&1 < NUL";

```

```
    $c4 = "ffmpeg -i c:\\inetpub\\wwwroot\\uploads\\temp\\" . $storename . " -s 1920x1080 -c:v libx264 -b:v 6000k
c:\\inetpub\\wwwroot\\uploads\\1080p_" . $encoded . " > NUL 2>&1 < NUL";
    pclose(popen("START /B $c1", 'r'));
    pclose(popen("START /B $c2", 'r'));
    pclose(popen("START /B $c3", 'r'));
    pclose(popen("START /B $c4", 'r'));
}
} else {
    echo "please select a video file";
}
} else {
    echo "Something went wrong, please try again later";
}
}
?>
```

## Liite 2: Videon lähetys -HTML

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8">
  <title>Keuda video</title>
  <link rel="stylesheet" type="text/css" href="style.css">
  <link rel="icon" href="media/favicon.ico" type="image/x-icon">
  <script>
    var m240 = false;
    var m360 = false;
    var m480 = false;
    var m576 = false;
    var m720 = false;
    var m1080 = false;
    var m1440 = false;
    var m2160 = false;
    var mh264 = false;
    var mvp9 = false;
    var mxvid = false;
    var mhigh = false;
    var mmain = false;
    var mbase = false;
    function uploadFile(){
      var file = document.getElementById("file").files[0];
      var chk = document.getElementById("pn").value;
      console.log(chk);
      if (file != null && chk != null || chk != ""){
        if (chk.length > 3){
          if (file.type.match('video.*')) {
            var chkres = chk.match(/:/g) + chk.match(/;/g) + chk.match(/#/g) + chk.match(/\?/g) + chk.match(/\*/g) + chk.match(/\\/g) +
chk.match(/\\/g) + chk.match(/&/g) + chk.match(/"/g) + chk.match(/'/g) + chk.match(/%/g);
            if (chkres == 0 || chkres == null || chkres == "null"){
              var formdata = new FormData();
              formdata.append("file", file);
              formdata.append("pn", document.getElementById("pn").value);

              if(adv){
                if (document.getElementById("c240").checked){
                  formdata.append("php240", "true");
                } else {formdata.append("php240", "false");}
                if (document.getElementById("c360").checked){
```

```

        formdata.append("php360", "true");
    } else {formdata.append("php360", "false");}
    if (document.getElementById("c480").checked){
        formdata.append("php480", "true");
    } else {formdata.append("php480", "false");}
    if (document.getElementById("c576").checked){
        formdata.append("php576", "true");
    } else {formdata.append("php576", "false");}
    if (document.getElementById("c720").checked){
        formdata.append("php720", "true");
    } else {formdata.append("php720", "false");}
    if (document.getElementById("c1080").checked){
        formdata.append("php1080", "true");
    } else {formdata.append("php1080", "false");}
    if (document.getElementById("c1440").checked){
        formdata.append("php1440", "true");
    } else {formdata.append("php1440", "false");}
    if (document.getElementById("c2160").checked){
        formdata.append("php2160", "true");
    } else {formdata.append("php2160", "false");}
    if (document.getElementById("h264").checked){
        formdata.append("codec", "libx264");
        if(document.getElementById("high").checked){formdata.append("profile", "high");}
        if(document.getElementById("main").checked){formdata.append("profile", "main");}
        if(document.getElementById("baseline").checked){formdata.append("profile", "baseline");}
    } else if(document.getElementById("vp9").checked){
        formdata.append("codec", "libvpx-vp9");
    } else if(document.getElementById("xvid").checked){
        formdata.append("codec", "libxvid");
    } else {formdata.append("codec", "notSelected");}
    if(document.getElementById("poster").checked){
        var posfile = document.getElementById("posterfile").files[0];
        if (posfile.type.match('image.*')) {
            formdata.append("customposter", "true");
            formdata.append("poster", posfile);
        }
    }
}

var a = new XMLHttpRequest();
a.upload.addEventListener("progress", progressHandler, false);
a.addEventListener("load", completeHandler, false);
a.addEventListener("error", errorHandler, false);

```

```

a.addEventListener("abort", abortHandler, false);
a.open("POST", "upload.php");
a.send(formdata);

document.getElementById("progressBar").style.display = "inline";
document.getElementById("progressBar").style.width = "100%";
document.getElementById("status").style.display = "inline";
document.getElementById("inputs").style.display = "none";
}else {document.getElementById("errorbox").innerHTML = "Älä käytä merkkejä \ ' : ; % & # / \ / ? ";
}
}else {document.getElementById("errorbox").innerHTML = "Valitse videotiedosto";
}
}else {document.getElementById("errorbox").innerHTML = "Nimi ei voi olla alle kolme merkkiä pitkä";
}
}else {document.getElementById("errorbox").innerHTML = "Muista valita tiedosto ja asettaa videon nimi";
}
}
var ajoitus;
var hmm = 0;
var salli = true;
function progressHandler(event){
//document.getElementById("loaded_n_total").innerHTML = "Uploaded "+event.loaded+" bytes of "+event.total;
var percent = (event.loaded / event.total) * 100;
document.getElementById("progressBar").value = Math.round(percent);
document.getElementById("status").innerHTML = Math.round(percent)+"% lähetetty... Älä poistu sivulta";

if (percent >= 99.999 && salli) {ajoitus = setInterval(tarkastus, 60);salli=false;}
}
function tarkastus() {
hmm++;
}
var adv = false;
function advanced() {
adv = !adv;
if (adv){
document.getElementById("advanced").style = "display:block; background-color:#eef1ee; padding: 10px;";
document.getElementById("advlabel").innerHTML = "Käytä oletusarvoja &uarr;";
} else {
document.getElementById("advanced").style = "display:none;";
document.getElementById("advlabel").innerHTML = "Edistyneet asetukset &darr;";
}
}
}
function completeHandler(event){

```

```
if (hmm < 4) {document.getElementById("status").innerHTML = "<span style='color: orange;background-color:#222;'>Hmm,
se kävi liian nopeasti, saattoi siis tapahtua virhe.</span>"
}
else {document.getElementById("status").innerHTML = "Enkoodaus aloitettu. Voit poistua nyt sivulta";
}
//document.getElementById("progressBar").value = 0;
}
function errorHandler(event){
    document.getElementById("status").innerHTML = "Upload Failed";
}
function abortHandler(event){
    document.getElementById("status").innerHTML = "Upload Aborted";
}
function resdisplay() {
if (document.getElementById("original").checked){
    if (document.getElementById("c240").checked){
        document.getElementById("c240").checked = false;
        m240 = true;
    }
    if (document.getElementById("c360").checked){
        document.getElementById("c360").checked = false;
        m360 = true;
    }
    if (document.getElementById("c480").checked){
        document.getElementById("c480").checked = false;
        m480 = true;
    }
    if (document.getElementById("c576").checked){
        document.getElementById("c576").checked = false;
        m576 = true;
    }
    if (document.getElementById("c720").checked){
        document.getElementById("c720").checked = false;
        m720 = true;
    }
    if (document.getElementById("c1080").checked){
        document.getElementById("c1080").checked = false;
        m1080 = true;
    }
    if (document.getElementById("c1440").checked){
        document.getElementById("c1440").checked = false;
        m1440 = true;
    }
}
```

```
if (document.getElementById("c2160").checked){
    document.getElementById("c2160").checked = false;
    m2160 = true;
}
if (document.getElementById("h264").checked){
    document.getElementById("h264").checked = false;
    mh264 = true;
}
if (document.getElementById("vp9").checked){
    document.getElementById("vp9").checked = false;
   .mvp9 = true;
}
if (document.getElementById("xvid").checked){
    document.getElementById("xvid").checked = false;
    mxvid = true;
}
document.getElementById("resolutions").style = "opacity:0.7";
document.getElementById("codecs").style = "opacity:0.7";
}else{
    document.getElementById("resolutions").style = "";
    document.getElementById("codecs").style = "";
    if(m240){document.getElementById("c240").checked = true;}
    if(m360){document.getElementById("c360").checked = true;}
    if(m480){document.getElementById("c480").checked = true;}
    if(m576){document.getElementById("c576").checked = true;}
    if(m720){document.getElementById("c720").checked = true;}
    if(m1080){document.getElementById("c1080").checked = true;}
    if(m1440){document.getElementById("c1440").checked = true;}
    if(m2160){document.getElementById("c2160").checked = true;}
    if(mh264){document.getElementById("h264").checked = true;
        if(mhigh){document.getElementById("c2160").checked = true;}
        else if(mbase){document.getElementById("c2160").checked = true;}
        else{document.getElementById("c2160").checked = true;}
    }
    else if(mvp9){document.getElementById("vp9").checked = true;}
    else{document.getElementById("xvid").checked = true;}

    m240 = false;
    m360 = false;
    m480 = false;
    m576 = false;
    m720 = false;
    m1080 = false;
```



```

<h1 id="otsikko">Upload</h1>
<br>
<div id="uploadhandler">
  <form id="upload_form" enctype="multipart/form-data" method="post">
    <div id="inputs">
      <input type="file" name="file" id="file" required><br><br>
      <label>Videon nimi</label><br>
      <input type="text" name="pn" id="pn" pattern="^[^;'\x22]*" title="Ei kaksoispisteitä, puolipilkkuja eikä heittomerkkejä
kiitos" required>
      <!-- <label>Lyhyt kuvaus:</label>
      <input type="text" name="description" rows="3" cols="30" placeholder="Minun kiva video"><br> -->
      <p onclick="advanced()" id="advlabel" style="color:#363;">Edistyneet asetukset &darr;</p>
      <div id="advanced" style="display:none;">
        <input id="original" onclick="resdisplay()" type="checkbox" name="original" value="original"><label
for="original">Käytä alkuperäistä videotiedostoa (ei enkoodausta)<span style="color:#533; font-size:80%;"> ei
suositella</span></label><br><br>
        <label><b>Valitse enkoodattavat resoluutiot</b></label><br>
        <div id="resolutions" style="">
          <input onclick="resume()" id="c240" class="rescheck" type="checkbox" name="240" value="240"><label
for="240">240p</label><br>
          <input onclick="resume()" id="c360" class="rescheck" type="checkbox" name="360" value="360"><label
for="360">360p</label><br>
          <input onclick="resume()" id="c480" class="rescheck" type="checkbox" name="480" value="480"><label
for="480">480p</label><br>
          <input onclick="resume()" id="c576" class="rescheck" type="checkbox" name="576" value="576"><label
for="576">576p</label><br>
          <input onclick="resume()" id="c720" class="rescheck" type="checkbox" name="720" value="720" checked><label
for="720">720p</label><br>
          <input onclick="resume()" id="c1080" class="rescheck" type="checkbox" name="1080" value="1080"><label
for="1080">1080p</label><br>
          <input onclick="resume()" id="c1440" class="rescheck" type="checkbox" name="1440" value="1440"><label
for="1440">1440p</label><br>
          <input onclick="resume()" id="c2160" class="rescheck" type="checkbox" name="2160" value="2160"><label
for="2160">2160p</label><br>
        </div>
      <br>
      <div style="border-top: 1px solid gray; width 100%; height: 1px;"></div>
      <br><label><b>Koodekki</b></label><br>
      <div id="codecs">
        <input onclick="resume()" id="h264" class="codecradio" type="radio" name="codec" value="h.264" checked><label
for="h.264">H.264 <span calss="sulut">(.mp4)</span></label><br>
        <div id="profiles264" style="position:relative; left:40px; font-size:90%">

```

```

        <label>Profili</label><br>
        <input onclick="resume()" id="high" class="prof264" type="radio" name="profile264" value="high"><label
for="high">high</label>
        <input onclick="resume()" id="main" class="prof264" type="radio" name="profile264" value="main"
checked><label for="main">main</label>
        <input onclick="resume()" id="baseline" class="prof264" type="radio" name="profile264" value="baseline"><label
for="baseline">baseline</label>
    </div>
    <input onclick="resume()" id="vp9" class="codecradio" type="radio" name="codec" value="vp9"><label
for="vp9">VP9 <span calss="sulut">(.webm)</span></label><br>
    <input onclick="resume()" id="xvid" class="codecradio" type="radio" name="codec" value="xvid"><label
for="xvid">Xvid <span calss="sulut">(.avi)</span></label><br>
    </div><br>
    <div style="border-top: 1px solid gray; width 100%; height: 1px;"></div>
    <br>
    <div id="customposter">
        <input onclick="showpoterinput()" id="poster" class="poscheck" type="checkbox" name="poster"><label
for="poster">Käytä omaa poster/thumbnaail kuvaa</label><br>
        <div id="posterinput" style="display:none">
            <input type="file" name="posterfile" id="posterfile">
        </div>
    </div>
    <br>
    <br>
    <input type="button" value="Upload File" onclick="uploadFile()"><br>
</div>

<div id="outputs">
    <progress style="display: none;" id="progressBar" value="0" max="100" style="width:100%;"></progress>
    <h3 style="display: none;" id="status"></h3>
</div>
</form>
</div>
<p id="errorbox"></p>

</body>
</html>

```

**Liite 3: IP OSOITEEN TARKASTUS JA UUELLEENOAJAUS – PHP-KOODI**

```
<?php

if ($_SERVER['REMOTE_ADDR'] == "192.168.73.221") {
    header('location: /221.html');
} else if ($_SERVER['REMOTE_ADDR'] == "192.168.73.222") {
    header('location: /222.html');
} else if ($_SERVER['REMOTE_ADDR'] == "192.168.73.223") {
    header('location: /223.html');
} else if ($_SERVER['REMOTE_ADDR'] == "192.168.73.224") {
    header('location: /224.html');
} else if ($_SERVER['REMOTE_ADDR'] == "192.168.73.225") {
    header('location: /225.html');
} else if ($_SERVER['REMOTE_ADDR'] == "192.168.73.226") {
    header('location: /226.html');
} else if ($_SERVER['REMOTE_ADDR'] == "192.168.73.227") {
    header('location: /227.html');
} else if ($_SERVER['REMOTE_ADDR'] == "192.168.73.228") {
    header('location: /228.html');
} else if ($_SERVER['REMOTE_ADDR'] == "192.168.73.229") {
    header('location: /229.html');
} else if ($_SERVER['REMOTE_ADDR'] == "192.168.73.230") {
    header('location: /230.html');
} else if ($_SERVER['REMOTE_ADDR'] == "192.168.73.231") {
    header('location: /231.html');
} else if ($_SERVER['REMOTE_ADDR'] == "192.168.73.232") {
    header('location: /232.html');
} else if ($_SERVER['REMOTE_ADDR'] == "192.168.73.233") {
    header('location: /233.html');
} else if ($_SERVER['REMOTE_ADDR'] == "192.168.73.234") {
    header('location: /234.html');
} else if ($_SERVER['REMOTE_ADDR'] == "192.168.73.235") {
    header('location: /235.html');
} else {
    echo "<h1 style='margin:10%'>unauthorized ip-
address</h1><script>setTimeout(function(){location.replace('index.html');},2000);</script>";
}

?>
```

## Liite 4: VIDEOLISTA - HTML

```

<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8">
  <meta http-equiv="Cache-control" content="no-cache, no-store, must-revalidate">
  <style> .vidi {width: 215px; height: 270px; background-color: #cde; text-align:center; border: 1px solid gray; display: inline-block; margin:10px;
padding: 2px; text-overflow: ellipsis; overflow: hidden; font-size: 22px; text-decoration: none;}
  .t{width: 100%; height: 160px; margin: 0px 1px 6px 1px;}.closer{color:white;background-color:maroon;text-
size:33px;padding:5px;float:right;}.closer:hover{cursor:pointer;}
  .container{display:flex; align-items: start; justify-content: left; flex-wrap: wrap; box-sizing: border-box; width: 100%; padding:0px 35px;}
  .pvm {width:100%; border-top: 1px solid gray; margin-top:0; padding:0;}
  html, body{margin:0;min-height:100%; max-width:100%;font-family:Arial;} #ref{opacity:0.85} #ref:hover{animation: spin 0.2s linear
forwards;}@keyframes spin{100%{-webkit-transform:rotate(90deg);transform:rotate(90deg);}}
  #otsikko{display:block;color: white;background-color: #222;margin-left: 20%;margin-right: 20%;margin-bottom: 10px;margin-top: 0;padding:
20px 30px;border: 3px solid black;opacity:0.75;position:relative;z-index:2;} </style>
</head>
<div id="formi" style="display:none;"><div class="closer" onclick="closeform()">X</div><h2 id="formioticsikko">Poista video</h2><br><form
method="post" action="/delevid.php">
  Username: <input type="text" name="user"><br><br>Password: <input type="password" name="password" ><input type="hidden"
name="suffix" id="suffix" value="no value"><input type="hidden" name="title" id="videotitle" value="no value"><br><br><input type='submit'
name='submit'></form></div>
<body><h1 id="otsikko">Videot</h1>
<div class="container">
<div class='pvm'>17.03.20</div>
<span id='15844507024Ktesti.mp4_1'></span>
<div class='vidi'><img src='delete.png' alt='delete video' width='18px' height='19px' onclick='del("15844507024Ktesti.mp4")' style='position:
relative; top: 250px; left: 99px; opacity:0.6; z-index: 8;'><a style='text-decoration: none;' href='/uploads/15844507024Ktesti.mp4.html'><div
style='position: relative; top: -22px; z-index: 7;'><img class='t' src='15844507024Ktesti.mp4.png' alt='thumbnail'><span
id='15844507024Ktesti.mp4'>4K testi</span></div></a></div>
<span id='15844507024Ktesti.mp4_2'></span>
<span id='1584448773Testi.mp4_1'></span>
<div class='vidi'><img src='delete.png' alt='delete video' width='18px' height='19px' onclick='del("1584448773Testi.mp4")' style='position: relative;
top: 250px; left: 99px; opacity:0.6; z-index: 8;'><a style='text-decoration: none;' href='/uploads/1584448773Testi.mp4.html'><div style='position:
relative; top: -22px; z-index: 7;'><img class='t' src='1584448773Testi.mp4.png' alt='thumbnail'><span
id='1584448773Testi.mp4'>Testi</span></div></a></div>
<span id='1584448773Testi.mp4_2'></span>


<!-- <button onclick="location.reload()" style="position:fixed; top:55px; left: 40px;">refresh</button>
-->
</div>
<script>
  function del(viteo){
    if(window.innerWidth > 1000){

```

```
document.getElementById('formi').style = "display: block; position: fixed; padding: 0% 0% 4% 4%; top:29%; left:29%; width:29%;
height:29%; background-color:white; border: solid 2px #21e; border-radius:11px; box-shadow: 3px 6px 18px; z-index: 9; overflow:hidden;";
}else { document.getElementById('formi').style = "display: block; position: fixed; padding: 0% 0% 4% 4%; top:15%; left:15%; width:60%;
height:60%; background-color:white; border: solid 2px #21e; border-radius:11px; box-shadow: 3px 6px 18px; z-index: 9; overflow:hidden;"; }
console.log(document.getElementById('suffix').value);
document.getElementById('suffix').value = viteo;
console.log(document.getElementById('suffix').value);
console.log(document.getElementById('videotitlex').value);
document.getElementById('videotitlex').value = document.getElementById('huuhaa').innerHTML;
console.log(document.getElementById('videotitlex').value);
document.getElementById('formioticsikko').innerHTML = "Poista video " + document.getElementById('suffix').value;
}
function closeform(){
document.getElementById('formi').style = "display: none";
}
</script>
</body>
</html>
```

**Liite 5: GENEROITU HTML-SIVU ESIMERKKI JA PLYR INSTANSSIN RAKENNUS**

```
<!DOCTYPE html>
<html>
<head>
<meta charset='utf-8'>
<style>html, body{margin:0;min-height:100%;font-family: Arial;}body{padding:0px 40px;text-align:center;}.container {max-width: 72%;margin: 0
auto;}.plyr {border-radius: 7px;margin-bottom: 15px;}#otsikko{display:block;color: white;background-color: #222;margin-left: 20%;margin-right:
20%;margin-bottom: 10px;margin-top: 0;padding: 20px 30px;overflow:hidden;border: 3px solid black;opacity:0.75;position:relative;z-
index:2;}</style>
<link rel='stylesheet' href='plyr.css'>
<script src='plyr.js'></script>
</head>
<body>
<h1 id='otsikko'>4K testi</h1>
<div class='container'>
<video id='player' controls playsinline poster='/uploads/15844507024Ktesti.mp4.png'>
<source src='/uploads/360p_15844507024Ktesti.mp4' type='video/mp4' size='360'>
<source src='/uploads/576p_15844507024Ktesti.mp4' type='video/mp4' size='576'>
<source src='/uploads/720p_15844507024Ktesti.mp4' type='video/mp4' size='720'>
<source src='/uploads/1080p_15844507024Ktesti.mp4' type='video/mp4' size='1080'>
<source src='/uploads/1440p_15844507024Ktesti.mp4' type='video/mp4' size='1440'>
<source src='/uploads/2160p_15844507024Ktesti.mp4' type='video/mp4' size='2160'>
</video>
<a href='videolista.html' style='background-color: blue; border-radius: 12px 2px 2px 12px; text-decoration:none; padding: 5px; color: white; display:
block; float:right;'>Takaisin</a>
</div>
<script>
document.addEventListener('DOMContentLoaded', () => {
const player = new Plyr('#player');
window.player = player;
function on(selector, type, callback) {document.querySelector(selector).addEventListener(type, callback, false)};
player.addEventListener('error', function() {alert("Your video is most probably still being encoded. \nTry with lower quality or check back later.\n
\nIn rare cases the upload process may have failed for unknown reasons");});
</script>
</body>

</html>
```

**Liite 6: VIDEON POISTO MANUAALISESTI – PHP-KOODI**

```

<?php
$directed = true;
include "nuuts.php";
if( $_POST['user'] == $id && $_POST['password'] == $ss ) {
    $videofile = $_POST['suffix'];
    $videotitle = $_POST['title'];
    $v1 = "240p_" . $videofile;
    $v2 = "480p_" . $videofile;
    $v3 = "720p_" . $videofile;
    $v4 = "1080p_" . $videofile;
    $thumb = $videofile . ".png";
    if (file_exists("uploads/" . $v1)){
        echo "deleting " . $v1;
        if (!unlink("uploads/" . $v1)){
            echo "..failure";
        }
        else { echo " <span style='font-size: 110%; color: green;'>...OK! </span><br>"; }
    }
    if (file_exists("uploads/" . $v2)){
        echo "deleting " . $v2;
        if (!unlink("uploads/" . $v2)){
            echo "..failure";
        }
        else { echo " <span style='font-size: 110%; color: green;'>...OK! </span><br>"; }
    }
    if (file_exists("uploads/" . $v3)){
        echo "deleting " . $v3;
        if (!unlink("uploads/" . $v3)){
            echo "..failure";
        }
        else { echo " <span style='font-size: 110%; color: green;'>...OK! </span><br>"; }
    }
    if (file_exists("uploads/" . $v4)){
        echo "deleting " . $v4;
        if (!unlink("uploads/" . $v4)){
            echo "..failure";
        }
        else { echo " <span style='font-size: 110%; color: green;'>...OK! </span><br>"; }
    }
    if (file_exists("uploads/" . $thumb)){
        echo "deleting thumbnail";
    }
}

```

```
if (!unlink("uploads/" . $thumb)){
    echo "..failure";
}
else { echo " <span style='font-size: 110%; color: green;'>...OK! </span><br>"; }
}
echo "unlinking video";
$lista = file_get_contents("uploads/videolista.html");
$lista = str_replace("<span id=\"" . $videofile . "_1'></span>", '<!-- Video Poistettu', $lista);
$lista = str_replace("<span id=\"" . $videofile . "_2'></span>", '->', $lista);
file_put_contents("uploads/videolista.html", $lista);
echo " <span style='font-size: 110%; color: green;'>...OK! </span><br>";
echo "<h2 style='margin:50px'>Video
poistettu!<h2><script>setTimeout(function(){location.replace('/uploads/videolista.html');},2500);</script>";
}
else { echo "<h2 style='margin:100px 50px 10px'>Virheellinen tunnus tai
salasana</h2><script>setTimeout(function(){location.replace('/uploads/videolista.html');},2000);</script>"; }
```

## Liite 7: STREAMIN ALOITUS JA JULKAISU, SEKÄ LOPETUS

```

<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8">
  <title>Aloita streamaus</title>
  <link rel="stylesheet" href="plyr.css">
  <script src="plyr.js"></script>
  <style> p {padding-left:5px;}body{font-family: Arial;}</style>
</head>
<body>
<div id="header">
<!-- <h4>IP todennus onnistui (< ?php echo $_SERVER['REMOTE_ADDR'] ?>)</h4> -->
<h2>Server: <span style="color: #66b;"> rtmp://192.168.73.206/live/stream09 </span> </h2>
<h2>Stream key: <span style="color: #66b;"> stream09 </span> </h2>
<a href="setup.html" target="_blank">Ohjeet</a>
</div>
<div id="vidi" style="width: 100%; display:flex; flex-wrap: stretch; align-items: flex-start;">
<div id="status" style="background-color:#ddd;">
<p id="info">Kun saat streamin näkyviin alla olevassa esikatselussa, voit julkaista sen. <button onclick="reload()">Refresh</button></p>
<div id="videocontainer" style="margin: 3px 4px 6px 4px; padding-bottom: 7px; display: inline-block; width: 65%;">
<video id="player" controls playsinline>
  <source src="http://192.168.73.206:8080/stream09.webm">
<!-- <source src="prfile3.mpg" type="video/mpg"> -->
</video>
</div>
<div id="formcontainer" style="display: inline-block; width:33%;">
<form id="markku" style="padding: 10px;"> <!--action="221status.php" method="post" -->
<p>Lähettyksen nimi:</p>
<input id="title" type="text" name="title">

<p>Valitse streamin formaatti: <br><span style="color:#373737; font-size:90%;">ei käytettävissä toistaiseksi</span></p>
  <input id="radiowebm" type="radio" name="format" value="webm" checked> webM <br>
  <input id="radiohls" type="radio" name="format" value="hls"> HLS <br>
</form>
<div id="streamStart" style="display: none;">
  <button onclick="start()" style="display: inline-block; position: relative; float:right; min-width: 150px; min-height:50px;">Julkaise!</button>
</div>
<div id="streamStop" style="display: none; margin: 0px 0px 10px 5px">
  <button onclick="stop()" style="display: inline-block; position: relative; float:right; min-width: 150px; min-height:50px;">Lopeta
streamaus</button>
</div>
</div>

</div>

</div>

</div>

```

```
<script>
var working = false;
var nimi = "En osannut laittaa streamilleni nimeä";
var q = new XMLHttpRequest();
function ready() {
    document.getElementById("status").style = "background-color: #ee2";
}
function start() {
    if (document.forms["markku"]["title"].value !== "") {
        nimi = document.forms["markku"]["title"].value;
    }
    var formdata = new FormData();
    formdata.append("title", nimi);
    console.log(nimi);
    if (document.getElementById("radiowebm").checked) {
        formdata.append("format", document.getElementById("radiowebm").value);
    }
    if (document.getElementById("radiohls").checked) {
        formdata.append("format", document.getElementById("radiohls").value);
    }
    formdata.append("host", "229"); // Vaihda
    formdata.append("status", "online");

    q.open("POST", "publish.php");
    q.send(formdata);

    console.log("Taavi on paras!");
    document.getElementById("status").style = "background-color: #5f1";
    document.getElementById("streamStart").style = "display: none";
    document.getElementById("streamStop").style = "display: inline-block";
    document.getElementById("header").innerHTML = "<h1>" + nimi + "</h1>";
    document.getElementById("info").innerHTML = "Online";
}
function stop() {
    var formdata2 = new FormData();
    formdata2.append("title", nimi);
    if (document.getElementById("radiowebm").checked) {
        formdata2.append("format", document.getElementById("radiowebm").value);
    }
    if (document.getElementById("radiohls").checked) {
        formdata2.append("format", document.getElementById("radiohls").value);
    }
    formdata2.append("host", "229"); // Vaihda
    formdata2.append("status", "offline");

    q.open("POST", "publish.php");
    q.send(formdata2);

    document.getElementById("status").style = "background-color: #ddd";
}
```

```

document.getElementById("streamStart").style = "display: none";
document.getElementById("streamStop").style = "display: none";
document.getElementById("info").innerHTML = "Stopped &nbsp; &nbsp; &nbsp; <b>Remember to shut down the stream from your streaming
software";
    working = false;
    player.pause();
    console.log("PHP stopping variables here");
}
function reload() {
window.location.reload(true);
}
window.onbeforeunload = function() {
    stop();
}
document.addEventListener('DOMContentLoaded', () => {
    const player = new Plyr('#player');
    window.player = player;
    function on(selector, type, callback) {document.querySelector(selector).addEventListener(type, callback, false);}
});

player.addEventListener('playing', function() {
    if (!working) {
        ready();
        document.getElementById("info").innerHTML = "Ready!";
        document.getElementById("streamStart").style = "display: inline-block";
        working = true;
    }
});
function publish(){
    if (document.forms["markku"]["title"].value !== null) {
        nimi = document.forms["markku"]["title"].value;
    }
    var formdata = new FormData();
    formdata.append("title", nimi);
    if (document.getElementById("radiowebm").checked) {
        formdata.append("format", document.getElementById("radiowebm").value);
        console.log(document.getElementById("radiowebm").value);
    }
}

</script>
</body>
</html>

```

**Liite 8: WEBSIVUN TYYLITTELY - CSS**

```
#bg-img {
  background-image: url('media/background.jpg');
  background-size: cover;
  filter: blur(4px);
  -webkit-filter: blur(4px);
  opacity: 0.7;
  position:fixed;
  top:0;
  left:0;
  height:100%;
  width:100%;
  z-index:1;
}

#menu {
  position: fixed;
  top: 0;
  left: 0;
  width: 100%;
  min-width: 490px;

  background-image: linear-gradient(135deg, #19c, #0073a8, #0073a8, #19c);
  box-shadow: 0px 3px 4px #444;
  color: white;
  z-index:100;
}

#menulink {
  display: inline-block;
  font-size: 28px;
  padding: 4px 15px;
  text-decoration: none;
  color: inherit;
}

#menulink2 {
  display: inline-block;
  font-size: 20px;
  padding: 5px 8px;
  text-decoration: none;
  color: inherit;
}

#menulink:hover, #selected{
  background-color: #6cbde2;
  display: inline-block;
  font-size: 28px;
  padding: 4px 15px;
  text-decoration: none;
  color: inherit;
}
```

```
}  
#menulink2:hover{  
  background-color: #6cbde2;  
}  
  
#language {  
  padding: 10px 4px;  
  font-size: 15px;  
  float: right;  
  color: #ccf;  
}  
  
#video {  
  display: block;  
  width: 70%;  
  max-width: 70%;  
  left: 15%;  
  margin-top: 20px;  
  z-index: 10;  
}  
#bodycenter {  
  position: fixed;  
  top:60px;  
  bottom:0;  
  left: 10%;  
  right: 10%;  
  border-radius: 5% 5% 0 0;  
  background-color: white;  
  box-shadow: -3px -3px 4px #444, 3px 3px 4px #444;  
  z-index:1;  
}  
#otsikko {  
  display:block;  
  color: white;  
  background-color: #222;  
  margin-top: 60px;  
  margin-left: 20%;  
  margin-right: 20%;  
  margin-bottom: 10px;  
  padding: 20px 30px;  
  border: 3px solid black;  
  opacity:0.75;  
  position:relative;  
  z-index:2;  
}  
#blur:hover {  
  filter: blur(2px);  
  -webkit-filter: blur(2px);  
  opacity: 0.8;
```

```
}  
#homecontainer {  
  width: 100%;  
  height: 70%;  
  position: absolute;  
  bottom: 0;  
  min-height:330px;  
  text-align:center;  
}  
#homeblock1, #homeblock2, #homeblock3, #homeblock4{  
  display: inline-block;  
  color: white;  
  background-color: #222;  
  border: 3px solid black;  
  opacity:0.82;  
  width: 22%;  
  height: 91%;  
  margin: 5px 10px;  
  padding: 0;  
  position:relative;  
  top: -30px;  
  z-index:2;  
  transition: 0.3s;  
}  
#homeblock1:hover, #homeblock2:hover, #homeblock3:hover, #homeblock4:hover{  
  margin: 0;  
  padding: 5px 10px;  
}  
#homeblock1 p, #homeblock2 p, #homeblock3 p, #homeblock4 p{  
  height: 100px;  
  position: absolute;  
  padding: 5px;  
}  
#blockimg1, #blockimg2, #blockimg3, #blockimg4{  
  margin: 20px 0px;  
  opacity: 1;  
}  
#bar {  
  margin: 25px 11% 25px 11%;  
  border-bottom: 1px solid #aaa;  
}  
#navbutton {  
  display: inline-block;  
  padding: 15px 40px;  
  font-size: 33px;  
}  
#navbutton a{  
  text-decoration: none;  
  color: #000;
```

```
}
#navbutton a:hover {
  color: #777;
}
#advlabel {
  cursor:pointer;
  margin: 1px 3px 8px 3px;
}
#chat {
  text-align: center;
  position: fixed;
  right: 2%;
  bottom: 0;
  width: 460px;
  height: 640px;
  z-index:11;
  display: none;
}
#uploadhandler {
  display: inline-block;
  width: 65%;
  text-align: left;
  padding: 20px;
  position:relative;
  z-index:2;
}
#uploadhandler input, label{
  font-size: 20px;
  padding: 4px;
}
#my-awesome-dropzone {
  display: block;
  width: 80%;
  height: 55%;
  background-image: radial-gradient(#e1f3f5, #e1f3f5, #add);
  border: 3px dotted #9cc;
  border-radius: 16px;
  position:absolute;
  left:10%;
  z-index:2;
  cursor: pointer;
  transition: 1s;
}
#my-awesome-dropzone:hover {
  background-image: radial-gradient(#fff, #fff, #add);
  border-color: #fff;
  border-radius: 22px;
}
html, body {
```

```
height: 100%;
margin:0;
padding:0;
text-align: center;
}
body {
min-height: 100%;
min-width: 100%;
position: absolute;
top: 0;
font-family: Arial;
}
iframe {
border-style: none;
width: 100%;
max-width: 100%;
height:92%;
margin: 0;
position:absolute;
bottom:0;
left:0;
z-index:3;
}
.sulut {
font-size: 90%;
}
```