

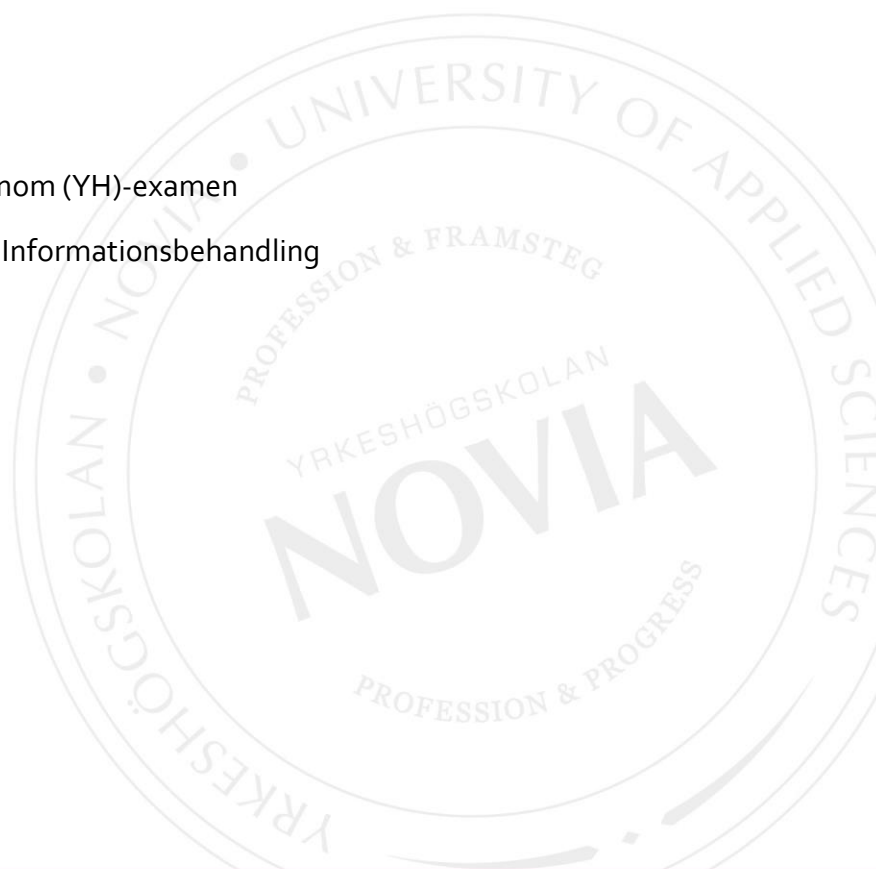
LAN-baserad Raspberry Pi server

Niklas Sandström

Examensarbete för Tradenom (YH)-examen

Utbildningsprogrammet i Informationsbehandling

Raseborg 2018



EXAMENSARBETE

Författare: Niklas Sandström

Utbildning och ort: Informationsbehandling, Raseborg

Handledare: Klaus Hansen

Titel: LAN-baserad Raspberry Pi server

Datum 30.11.2018

Sidantal 22

Abstrakt

Detta arbete beskriver arbetsprocessen för skapandet av en webbserver och NAS på en Raspberry Pi server för lokalnätverks användning. Webbservern är en Apache 2 installation med en PHP-läsare installerad, webbservern skall även ha tillgång till en databashanterare. NAS servern kommer att skötas av tjänsten SAMBA. Applikationerna kommer att konfigureras via SSH när man är kopplad till det lokala nätverket och via VNC om man inte har möjligheten att ansluta sig till det.

Syftet med detta projekt är att ha möjlighet att kunna lagra projekt och dokument på NAS servern för att ha tillgång till dem så gått som hela tiden. NAS servern innehåller en privat lagringsplats, samt en offentlig lagringsplats för lagring av material som är tillgängligt för andra användare på det lokala nätet att ha tillgång till. Syftet med webbservern är att ha möjligheten att utveckla websidor och webapplikationer på en "live"-server som man har tillgång till på plats.

Servern skall i främsta grad fungera som en Lokal nätverks server men det skall finnas möjlighet att ansluta sig till servern via internet om det krävs.

Språk: Svenska

Nyckelord: Raspberry, Server, Webbserver, NAS

BACHELOR'S THESIS

Author: Niklas Sandström

Degree Programme: Business Information Technology, Raasepori

Supervisor(s): Klaus Hansen

Title: LAN-based Raspberry Pi Server

Date 30.11.2018

Number of pages 22

Abstract

This project describes the work process in the creation of a webserver and NAS on an Raspberry Pi for usage on local networks, the webserver will be managed by Apache 2 with an PHP-interpreter installed, the webserver will also have a database-manager installed. The NAS will be managed by the service SAMBA, it will be possible to manage and configure the whole server via SSH when connected to the LAN or via VNC when not possible to connect to the LAN.

The purpose of this server is to have the possibility to store projects and documents on the NAS to have access to them at almost any time, the NAS contains two different shares one public for everyone connected to the LAN and one private. The webserver's purpose is to be able to test webpages and web applications on a "live"-server that you have physical access to.

The server is primarily meant to be used as an LAN server, but with the possibility of connecting it to the internet if necessary.

Language: English

Key words: Raspberry, Server, Webserver, NAS

Innehållsförteckning

1	Inledning.....	1
1.1	Syfte	1
1.2	Metodik	1
1.3	Produktens användningsområde.....	2
1.4	Hårdvaran	2
1.5	Installation av Operativsystemsinstalleren NOOBS	4
1.5.1	Vidare installation samt konfigurering av operativsystemet Raspbian.....	6
1.6	Beskrivning av nätverket	7
1.7	Mjukvaran	7
2	Apache 2.....	8
2.1	PHP.....	8
2.2	MySQL.....	9
2.2.1	Felsökning av MySQL.....	10
2.2.2	Raspbian's byte från MySQL till MariaDB	12
2.3	Drupal	12
2.4	SFTP.....	13
3	SAMBA.....	14
4	Arbetsförberedelser	19
4.1	Program som användes	19
4.1.1	Google Chrome.....	19
4.1.2	PuTTY.....	20
4.1.3	RealVNC.....	20
5	Testning av NAS tjänsten	22
6	Kritisk granskning.....	23
7	Avslutning	23
	Källförteckning	25

1 Inledning

I detta arbete skapar jag en server som jag skall kunna använda som lagringsplats. Där kan jag kan öka på hur mycket jag vill utan extra kostnader och testmiljö utan att den tar upp massor med utrymme eller kostar i månaden att hyra av ett företag. Jag valde detta projekt för att utveckla mina kunskaper om skapandet av webbservrar och NAS vilket jag märkte att var någonting som jag faktiskt tyckte om att göra under skoltiden under våra kurser.

Men huvudsakligen är detta ett projekt som fått mig att lära mig mera om hur mycket eller lite prestanda en Raspberry Pi egentligen har och att testa gränserna vad den klarar av inom ett annorlunda användningsområde än vad Raspberry Pi datorer vanligtvis används till.

1.1 Syfte

Detta examensarbete går ut på utvecklandet av diverse olika serverfunktioner som installeras på en Raspberry Pi dator. Raspberryn skall fungera som en webbserver på det lokala nätet för användning som testmiljö för webbsidor och webbapplikationer. Den skall även fungera som en Network Attached Storage (NAS, nätverksansluten lagringsplats.) med åtminstone en (1) lagringsplats för privat bruk samt en publik lagringsplats som skall fungera som lagringsutrymme för alla maskiner som ansluter till det lokala nätverket.

Hela helheten kommer kunna administreras via SSH och VNC på det lokala nätverket och endast med tjänsten VNC via internet ifall det krävs. VNC tjänsten kan i detta fall även fungera som en möjlighet att koppla sig vidare till andra datorer på Raspberryns lokala nätverk.

1.2 Metodik

För att göra detta examensarbete så bestämde jag mig för att först förverkliga servern, medan jag skrev anteckningar om diverse skeden i skapandet av helheten och till sist att skriva detta examensarbete. Detta har lett till att jag flera gånger har hamnat starta om på en del moduler som har installerats och som inte fungerat som jag hade förväntat mig. Men i största grad har detta examensarbete varit ett praktiskt utförande av kunskaper jag lärt mig under skolåren.

1.3 Produktens användningsområde

Produkten kommer i de flesta fallen användas som en NAS i en lägenhet. Den kommer även ibland att användas i ett egnahemshus var mängden anslutningar kan variera från en till fyra personer. Där kommer den att användas som en backup och som ett media server för filmer och musik. Webbservern kommer användas som en test-miljö för de som är anslutna till det lokala nätverket och har inloggningsuppgifterna för att ansluta till servern via SSH för att göra konfigurationer på filerna i webbservern.

1.4 Hårdvaran

Hårdvaran som användes i detta examensarbete är ytterst liten. Som grund för hela servermiljön används en Raspberry Pi model 2b (se bild 1). Detta är en enkorts dator som är utvecklad av Raspberry Pi Foundation. Datorn är ungefär samma storlek som ett kreditkort och inte värst kraftfull. Modell 2b har en 900MHz fyra-kärnig processor och 1Gb med arbetsminne. Portarna som kommer användas, inom detta arbete när det är slutfört, är internet-porten, två av de fyra USBportarna som finns tillgängliga och micro-usb porten som fungerar som ingång för strömkällan. Men utöver det så har Raspberryn även en HDMI port och en AUX-port. Detta är Raspberryns vanliga portar, sedan har den även en port för att ansluta en kamera till och en för en display och även 40 stycken GPIO kontakter.

Om inte diverse begränsningar skulle ha funnits så skulle projektet högst troligt blivit byggt på den nyare och kraftfullare Raspberry Pi Model 3B som har en kraftfullare processor på 1.4Ghz och inbyggt stöd för Wi-Fi och Bluetooth till skillnaden från den tidigare modellen.



Bild 1: Raspberry Pi Model 2b

För att få operativsystemet installerat så användes det ett Micro SD kort på 16Gb då Raspberryn inte har en hårddisk eller anslutning för en hårddisk. För att underlätta installationen av operativsystemet så användes en SD-korts adapter (se bild 2) för att få Micro SD-kortet kopplat till en dator varifrån operativsystemsinstalleringen kan överföras till Micro SD-kortet.



Bild 2: Micro SD korts adapter

Utöver Raspberryn så användes även en kort tid ett extra tangentbord och en skärm. För att konfigurera Raspberry Pi kortet så att man kan ansluta till det via SSH. Sedan har det kopplats till en externhårddisk på 1Tb för att få extra utrymme på enheten och för att göra den lämplig för användning som NAS-server. Till detta behövdes en USB hubb med en extern kraftkälla (se bild 3), då den externa hårddisken krävde för mycket energi för att Raspberryn skulle klara av att kunna använda hårddisken för att skriva och läsa ifrån den.



Bild 3: Hårdvara (inte med på bild HDMI kabel & skärm)

1.5 Installation av Operativsystemsinstalleringen NOOBS

NOOBS, står för "New Out Of the Box Software, är en installationsfil för en samling olika operativsystem som fungerar på Raspberry Pi, (Raspberry Pi, u.å), det fungerar på det viset att när du har formaterat ditt SD-kort och kört in NOOBS på kortet och lagt in det på Raspberryn så visar det en meny med olika operativsystem som du kan installera (se bild 4). Beroende på om du valt att ladda ner den fulla versionen av NOOBS eller NOOBS Lite så har du olika val. Med NOOBS Lite versionen så måste du vara ansluten till internet under installationsskedet oavsett vilket operativsystem som valts. Däremot så har den fulla versionen av NOOBS färdigt nerladdat alla filer som behövs för installationen av operativsystemet Raspbian, alla andra operativsystem fungerar lika som på Lite versionen.



Bild 4: Operativsystemen som NOOBS stöder.

För att installera NOOBS/NOOBS Lite på ett SD-kort så skall du formatera det först, för att göra detta kommer man enklast undan med att installera ett formateringsprogram som t.ex. SD Memory Card Formatter av sdcard.org. Detta program hjälper dig formatera ditt SD-kort och finns till Windows och Mac datorer (se bild 5), använder man sig av ett Linux operativsystem så kan man även med olika terminal kommandon formatera SD-kort.



Bild 5: Nedladdningsalternativ

När man har programmet installerat på exempelvis sin Windows maskin, så är det bra att kolla instruktionerna hur man skall formatera kortet för installationen av NOOBS och operativsystemet man tänker använda, i detta fall Raspbian. I detta fall så körs programmet med inställningarna FULL (Erase) och ger programmet tillstånd att ändra storleken på formateringen (se bilder 6 & 7). Efter att kortet har formaterats så extraherar man Operativsystemsinstallerares filer från det nedladdade NOOBS arkivet till det nyformaterade SD-kortet, efter det är det bara att sätta SD-kortet i Raspberry Pi datorn och starta den.

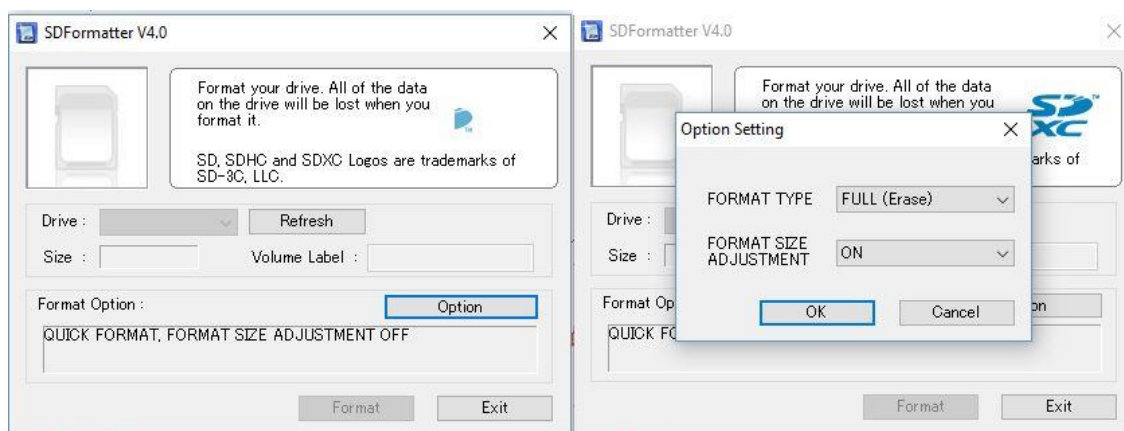


Bild 6 & 7: SDFormatter V4.0 & formaterings inställningar.

1.5.1 Vidare installation samt konfiguration av operativsystemet Raspbian

Jag valde Raspbian i detta fall för att det kräver lite prestanda av Raspberryn och det är den distributionen av Linux jag känner igen bäst efter Ubuntu som kräver lite mera av Raspberryn. Under installationen valde jag att installera Lite versionen av Raspbian för att minimera allt onödigt som kan komma med i fulla installationer, detta i sin tur ledde till att jag inte fick en Graphical User Interface (GUI, användargränssnitt) att arbeta med. För att underlätta konfigurationen av operativsystemet och för att inte hela tiden hamna springa av och an via en dator med en webbläsare installerad så installerade jag en GUI på systemet med hjälp av lite kod (se bild 8).

```
pi@zpi:~$ sudo apt install raspberrypi-ui-mods
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following packages were automatically installed and are no longer required:
  default-mysql-server galera-3 libaio1 libcgi-fast-perl libcgi-pm-perl libdbd-mysql-perl libdbi-p
  mariadb-client-core-10.1 mariadb-common mariadb-server-10.1 mariadb-server-core-10.1 mysql-commo
Use 'sudo apt autoremove' to remove them.
The following packages will be upgraded:
  raspberrypi-ui-mods
1 upgraded, 0 newly installed, 0 to remove and 65 not upgraded.
Need to get 317 kB of archives.
After this operation, 1,024 B of additional disk space will be used.
Get:1 http://archive.raspberrypi.org/debian stretch/ui armhf raspberrypi-ui-mods all 1.20181118 [3
Fetched 317 kB in 0s (765 kB/s)
Reading changelogs... Done
(Reading database ... 73633 files and directories currently installed.)
Preparing to unpack .../raspberrypi-ui-mods_1.20181118_all.deb ...
Unpacking raspberrypi-ui-mods (1.20181118) over (1.20180706) ...
```

Bild 8: Installation av grafiskt gränssnitt.

Efter att det grafiska gränssnittet var installerat så var man tvungen att starta om systemet. Detta gjordes snabbt med koden ”sudo reboot”. Efter att systemet hade startat om så installerade jag webbläsaren Chromium detta gjordes med terminal kommandot ”sudo apt-get install chromium-browser” (se bild 9) för att få tillgång till det bokmärkta arbetsmaterial och de guider som användes till projektet. Installationen skulle även gått att göra via det nya grafiska gränssnittet, men jag beslöt mig för att använda mig av terminalen när jag var van med den.

```
pi@zpi:~$ sudo apt-get install chromium-browser --yes
Reading package lists... Done
Building dependency tree
Reading state information... Done
chromium-browser is already the newest version (65.0.3325.181-0+rpt4).
The following packages were automatically installed and are no longer
  default-mysql-server galera-3 libaio1 libcgi-fast-perl libcgi-pm-perl
  mariadb-client-core-10.1 mariadb-common mariadb-server-10.1 mariadb-
Use 'sudo apt autoremove' to remove them.
0 upgraded, 0 newly installed, 0 to remove and 65 not upgraded.
```

Bild 9: Installation av Chromium via terminalen i Raspbian.

1.6 Beskrivning av nätverket

Servern i fråga kommer emellanåt flyttas från en lägenhet till ett egnahemshus och tillbaka, så den kommer vara kopplad till två olika nätverk av olika storlekar, i detta fall är det inget problem när nätverket i lägenheten har konfigurerats att ha samma interna IP-adresser som IP-adresserna i egnahemshuset och Raspberryn är konfigurerad till en statisk IP-adress som är exkluderad från DHCP funktionen i de båda nätverken.

För att konfigurera Raspberryn till att använda en statisk IP-adress så använde jag mig av kommandot `”sudo nano /etc/dhcpd.conf”` för att konfigurera DHCP funktionen till att inte dela ut någon annan IP än den jag specificerat med ändringarna i filen till Raspberryn. Efter att ändringarna på Raspberryn är gjorda så ändrade jag inställningarna på routern så att den inte skulle dela ut Raspberryns IP till någon annan maskin i nätverket genom att begränsa vilka IP-adresser den kunde dela ut. Samtidigt så konfigurerade jag routern och Raspberryn till att använda samma interna IP-rymd som finns i båda nätverken som Raspberryn skall kopplas fast till nu och då.

Skillnaden på dessa två nätverk är storleken och mängden med hårdvara mellan internetanslutningen och datorerna. I lägenheten, som Raspberryn huvudsakligen kommer användas i, så är det bara en router mellan internetanslutningen och datorerna. I det andra nätverket som är i ett egnahemshus så finns det en router, en firewall och två olika WLAN anslutningar vilket kan leda till problem. I detta fall var det inget större problem när nätverket är konfigurerat för att låta alla vara på samma IP-rymd. Detta gjorde det enkelt att konfigurera nätverken till att inte dela ut IP adressen som Raspberryn använder till någon annan maskin så att den alltid kommer vara ansluten med samma IP adress i de båda nätverken. Detta gjorde det möjligt att konfigurerar de andra maskinerna i nätverket så att de hittar de delade mapparna från NAS servern enklare. När servern har en statisk IP på båda nätverken så kunde nätverks mapparna konfigureras, utan att vare sig jag eller Raspberryn fanns på plats eller var ansluten till nätverket i egnahemshuset, via telefonsamtal.

1.7 Mjukvaran

För att få en Raspberry att fungera som en webserver eller NAS så måste man installera olika program och tjänster för att få den att agera på det viset man vill. Ifall man vill visa upp en webbsida via sin Raspberry så att den syns på det lokala nätet måste man installera en webservertjänst vilket i detta fall är Apache 2. Om man vill köra en lagringsserver, så

vill man använda sig av en tjänst som gör det säkrare, i detta fall användes tjänsten SAMBA. Om man inte använder sig av detta program och inte vet exakt vad man håller på med men ändå får det att fungera så kan datorn i fråga vara osäker med tanke på att folk kan ansluta till de delade filerna och därefter söka sig in till andra filer i datorn.

2 Apache 2

Som webbserver i detta fall användes Apache 2 som tjänst. Apache är en av de populäraste Open-Source webbservrarna som finns och lanserades år 1995 (http server project. (u.å)), detta är en av de få orsakerna som denna tjänst valdes, stor roll i valet spelade även skolningen vi har fått under våra datanätverks kurser och projekt.

För att installera Apache så använde jag mig av koden ”apt install apache2” (Raspbian France. 2018a) vilket körde in filerna som Apache behöver för att fungera och aktiverade tjänsten. Efter detta gjorde jag ett snabbt test genom att testa om den fungerade på Raspberryn med att besöka adressen 127.0.0.1 eller zpi.local/index.html via Chromium (se bild 10).



Bild 10: index.html på Chromium.

2.1 PHP

För att utöka funktionerna på webbservern så installerades även funktionen att läsa PHP filer vilket gör det möjligt att köra dynamiska skript och funktioner på webbservern. Detta gjordes enkelt med koden ”apt install php php-mbstring” (Raspbian France. 2018b), vilket på samma sätt som apachen kör in filerna och aktiverar funktionen.

För att testa ifall PHP-läsaren hade installerats så som den skulle så skapade jag en index.php fil som visar phpinfo(). Detta gjorde jag med att navigera till /var/www/html/ var jag använde mig av kommandot ”sudo nano” vilket öppnar en textfil, i textfilen skrevs

koden "<?php phpinfo ();?>" och sparades som index.php. Efter detta var navigerade jag via Chromium till adressen zpi.local/index.php (se bild 11) för att se att den fungerade som den skulle.

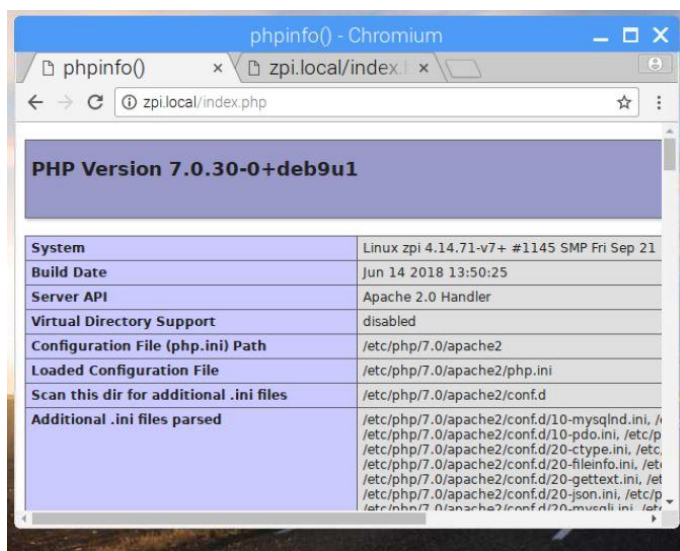


Bild 11: index.php på Chromium.

2.2 MySQL

MySQL är en databashanterare som krävs för att lagra information och för att installera Content Management Systems (CMS, innehållshanteringssystem.). I detta fall används CMS'et Drupal. För att MySQL skall fungera ordentligt så måste man även installera en funktion som länkar PHP och MySQL. För att göra detta snabbare så kördes koden "sudo apt install mysql-server php-mysql" (Raspbian France 2018c) vilket då installerade MySQL servern och PHP-MySQL. Efter installationen så testades det om MySQL fungerar med koden "sudo mysql -user=root" vilket skall logga in en på MySQL databasen som administratör. Inloggningen skulle göra databasen säkrare och för att original användarkontot inte skulle gå att användas med webbservern och PHP skript, så skulle ROOT användaren raderas för att sedan skapa en ny användare med samma namn. Detta skulle göras med SQL koderna

```
" DROP USER 'root'@'localhost';
```

```
CREATE USER 'root'@'localhost' IDENTIFIED BY 'password';
```

```
GRANT ALL PRIVILEGES ON *.* TO 'root'@'localhost';"
```

På grund av att databashanteraren inte låter mig ansluta som root för att göra konfigurationerna som står ovan så hamnade jag på grund av tidsbrist överge den funktionen tills när arbetet är färdigt och går att jobba på när det finns tid. Tyvärr så betyder det även att inget CMS system kommer vara installerat på servern tills den fungerar som den skall.

2.2.1 Felsökning av MySQL

När jag märkte att MySQL tjänsten inte fungerade efter att jag installerat tjänsten och försökte logga in via databashanteraren så var jag tvungen att börja felsöka tjänsten för att hitta vad som skulle kunna vara problemet. Efter att jag googlat en tid och varit inne på diverse forum så kom jag till slutsatsen att det inte är ovanligt det här felet. Massor med människor hade samma fel som mig. De inte kunde ställa in något root lösenord till databashanteraren. Vanligtvis så skulle man borda kunna logga in utan lösenord men i detta fall så var det inte möjligt av någon orsak. Först trodde jag att tjänsten eventuellt skulle kunnat använda lösenordet som Raspberry Pi'n hade så jag testade det med root access på terminalen. Snabbt märkte jag att det inte var så och att det var andra människor som testat på samma sak. I ett senare skede när jag återkom till problemet efter en liten paus så hittade jag en forum post på ett support forum som talade om hur man kan ställa in ett eget lösenord på root. För att göra detta så stängde jag av databashanteraren genom terminalen via kommandot "sudo systemctl stop mysql" vilket säger åt systemet att det skall stänga av tjänsten. Sedan startade jag tjänsten 'mysqld_safe' som root användare i terminalen. Jag lade till modifierarna '--skip-grant-tables' och '--skip-networking' efter kommandot för att starta databasen utan att köra in användar rättigheterna och låta mig få tillgång till mysql databasen som root användare. Detta gjorde jag genom att köra login kommandot 'mysql -u root'. När jag väl anslutit till databasen så måste jag köra in användarrättigheterna för att kunna nolla dem. Det gjorde jag med mysql kommandot 'FLUSH PRIVILEGES' som laddar in dem igen. Efter att de var laddade skulle jag försöka modifiera root användaren.

För att modifiera en användare på mysql använder man sig av ett kommando som heter 'ALTER USER' vilket man skriver på detta vis för att modifiera root i detta fall: 'ALTER USER 'root'@'localhost' IDENTIFIED BY 'nytt_losenord'.' Om detta kommando inte fungerar av någon orsak så kan det tyda på att man har ett större problem någonstans i installationen, men man kan testa ett annat kommando ifall det skulle fungera. I mitt fall så

verkade det som om det fungerade men när jag senare försökte använda det nya lösenordet så lät inte tjänsten mig ansluta igen med samma orsak: Fel lösenord.

Så jag använde mig av samma kommandon för att gå in i databasen som root. Jag testade det andra kommandot som är 'UPDATE ... SET' vilket ger en möjligheten att editera en större del användare samtidigt om man så vill. Nu använde jag mig av kommandot för att bara editera root användaren, så jag skrev kommandot som 'UPDATE mysql.user SET authentication_string =PASSWORD('nytt_lösenord') WHERE User = 'root' AND Host = 'localhost';' När jag körde kommandot fick jag svaret att kommandot var rätt skrivet och att databashanteraren hade kört det. Så jag loggade ut och försökte starta tjänsten igen, nu tog tjänsten så lång tid att starta att jag trodde det var något fel. När terminalen hade arbetat färdigt kollade jag statusen på databashanteraren och märkte att den sade att tjänsten inte kördes utan att den hade något fel. Så jag startade om hela servern och kollade om databashanteraren kördes efter det, och det gjorde den.

Sedan försökte jag logga in igen på databashanteraren med root användaren och med det nya lösenordet jag just hade ställt in och samma fel igen. Då den sade att jag inte fick tillgång till databasen den här gången så testade jag med att installera om hela tjänsten för att se om det hade gått något fel i själva installationskedet eller om jag hade missat tillfället att skriva in ett root lösenord. Detta hamnade jag göra på det viset att jag först avinstallerade hela tjänsten och sedan tog bort alla filer som inte mera var i bruk, så att det inte fanns filer som skulle störa den nya installationen av databashanteraren. Till detta använde jag mig av koden 'sudo apt-get -purge remove mysql-server php-mysql'. Efter att kommandot hade körts så testade jag mig av att installera mysql servern och php mysql igen. Den här gången höll jag mera koll på vad maskinen gör för att se om den talade något om att ställa in ett root lösenord, vilket den inte gjorde, så jag testade igen med att logga in som root användare på databashanteraren. Jag fick tyvärr exakt samma fel igen, så jag kollade statusen på tjänsten vilket visade att den nog kördes och inga fel var märkta.

Så det verkade som om den hade installerat helt som den borde men utav någon orsak så lät den mig inte logga in som root ännu heller. Efter att jag ett par gånger försökt att göra ominstallationer och att ställa in ett root lösenord med funktionerna jag tidigare beskrivit så övergav jag databashanteraren en stund medan jag arbetade med resten av servern så att jag skulle kunna se på problemet med ny energi senare. Efter ett par timmar återgick jag igen till problemet och sökte på olika supportforum för Debian Linux distributioner och på Raspberrys egna forum efter lösningar. Största delen av forumens skribenter har lyckats få

det att fungera med att använda sättet man kommer in i databashanteraren utan att ladda in användarrättigheterna. De som inte fick det att fungera så har, till största del, bytt till en annan databashanterare eller fått det att fungera med en ominstallation på en äldre version av Raspbian.

2.2.2 Raspbians byte från MySQL till MariaDB

Tidigare så använde sig Raspbian av Oracles version av MySQL. Det var stött av Raspbians egna 'repositories' men med den nyaste versionen av Raspbian så har utvecklarna bytt till MariaDB som databashanterare i sina 'repositories'. Dock innehåller paketen man installerar, när man skall ha en databashanterare, inte mera all programvara man behöver för att köra databashanteraren. Med den nya versionen av Raspbian kan man inte mera använda sig av mysql-server paketet vilket ännu går att använda sig av, på äldre versioner som, Raspbian Jessie. På Raspbian Stretch är det meningen att man skall använda sig av MariaDB's egna installationskommando för att få de filer som behövs på den nya versionen av operativsystemet. Det betyder att istället för att använda sig av kommandot 'sudo apt-get install mysql-server' så borde kommandot 'sudo apt-get install mariadb-server' användas enligt MariaDB's egna hemsida. Detta kommer vara nästa steg i testningen för att få databasen att fungera.

2.3 Drupal

Jag valde Drupal som mitt CMS mestadels för att det är ett CMS jag är van med att jobba med vilket gjorde det lätt att installera på webbservern så att jag kunde testa att det fungerar som den skall. Före projektet hade jag en Wordpress installation på servern för att se hur Wordpress fungerar och ifall jag skulle kunna tänka mig byta till det, men efter lite testande av hur det kändes att använda så kom jag snabbt fram till att Wordpress inte känns rätt för mig och att jag skulle måsta lära mig massor för att kunna göra samma saker som jag redan kan på Drupal.

Oavsett så är Drupal eller Wordpress inte nödvändiga just nu på servern men fungerade mera som ett test att allting hade gått rätt till vid installationen av Webbservern och dess tillägg. Tack vare att databashanteraren inte gick att konfigurera när jag blev ut låst från root användaren, så kunde jag tyvärr inte installera Drupal. Webbservern fungerar men det går inte att installera ett CMS på server just nu.

2.4 SFTP

SSH File Transfer Protocol (SFTP) är ett protokoll som gör det möjligt att överföra och hantera filer över SSH protokollet. Detta sköts av Raspberryn själv när SSH protokollet redan blivit aktiverat på servern. För att ha störst nytta av SFTP däremot så installeras en klient på datorn med vilken vi kan hantera och överföra filerna med. Till detta använde vi oss av klienten FileZilla vilket är en gratis FTP klient som även har stöd för SFTP (FileZilla. (u.å)), för att installera klienten på datorn så besökte vi FileZillas hemsida (se bild 12) var vi hämtade ner den nyaste FileZilla klienten.



Bild 12: FileZilla hemsida.

Efter att klienten var installerad så öppnade vi programmet och anslöt till Raspberryn med dess IP-adress eller dess namn på det lokala nätverket i fältet var det står 'Värd'. Vi fyllde i användarnamnet och lösenordet som används på raspberryn vilket, när Raspbian är nyinstallerat och du inte gjort några ändringar till användaren, är 'pi' och 'raspberrypi'. Sedan specificerade vi även porten som skall användas när vi ansluter till raspberryn, i detta fall använder vi oss av SSH porten vilken är numrerad 22 (se bild 13).

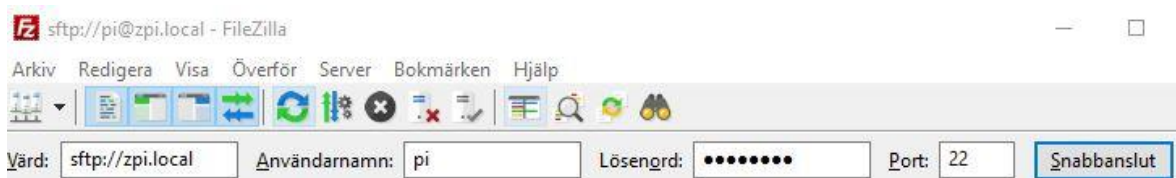


Bild 13: FileZilla anslutnings inställningar.

Användningen för FileZilla i detta projekt är för överföringen av filer till webbservern. Det skulle även varit möjligt att konfigurera NAS servern till att dela ut webbserverns mapp som en delad folder för att göra det möjligt att lägga till filer till servern. Men för att få lite

mera variation till funktionerna på servern så drogs den slutsatsen att det nog är bättre att använda sig av FileZilla istället.

3 SAMBA

SAMBA är en tjänst som gör det möjligt för datorer som använder sig av olika operativsystem att dela printrar, lagringsutrymme och andra tjänster utan operativsystems restriktioner (SAMBA, (u.å)). SAMBA används i detta fall för att dela ut en externhårdskiva till det lokala nätet så att andra användare kan använda sig av den.

SAMBA har även funktionen att kunna skapa användare för tjänsten så att endast de som har lösenord och användarnamn har tillgång till tjänsterna. Detta kommer att användas för min egna privata mapp på den externa hårdskivan, medan en annan mapp skapades för alla andra på nätverket som behöver mappen som lagringsplats och mediabibliotek i det lokala nätverket.

Skapandet av de delade lagringsplatserna gjordes med koden "mkdir /home/shares" vilket skapade det privata lagringsutrymmet. Efter att det var skapat så användes samma kod men med "../public" lagt till i slutet av koden för att skapa det publika lagringsutrymmet åt de andra användarna på det lokala nätverket. (se bild 14)

```
pi@zpi: ~$ sudo mkdir /home/shares
pi@zpi: ~$ sudo mkdir /home/shares/public
pi@zpi: ~$ sudo chown -R root:users /home/shares/public
```

Bild 14: Skapande av mappar

Efter att de var skapade så konfigurerades inställningarna för mapparna med kommandona "chown" och "chmod" detta gjorde att användaren på det lokala nätverket har möjlighet till att använda mapparna till lagring efter att SAMBA tjänsten installerades. För att installera SAMBA så användes terminalen för att skriva in koden "sudo apt install samba samba-common-bin" (se bild 15).

```

pi@zpi:~$ sudo apt install samba samba-common-bin
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  attr libaiol libgpgmell python-crypto python-dnspython python-ldb python-pysmb
Suggested packages:
  gpgsm python-crypto-dbg python-crypto-doc python-gpgme bind9 bind9utils
The following NEW packages will be installed:
  attr libaiol libgpgmell python-crypto python-dnspython python-ldb python-pysmb
0 upgraded, 13 newly installed, 0 to remove and 59 not upgraded.
Need to get 3,815 kB/3,825 kB of archives.
After this operation, 25.2 MB of additional disk space will be used.
Do you want to continue? [Y/n] y

```

Bild 15: Installation av SAMBA

Vilket körde in tjänsten i maskinen. När tjänsten väl var installerad så var man tvungen att editera tjänstens konfigurationsfil vilken man kom åt via att använda kommandot “sudo nano /etc/samba/smb.conf”. Det i sin tur öppnade konfigurationsfilen i text editeringsprogrammet nano. För att begränsa användningen på tjänsten så lades det till en rad med texten “security = user” under rubriken Authentication i konfigurationsfilen. När den raden läggs till så kommer det krävas att man har ett användarkonto i tjänsten för att kunna använda den. Efter att säkerheten var hanterad så editerades raden med texten “read only = yes” till “read only = no” under rubriken [homes] vilket editerar rättigheterna på den privata mappen till att man kan lägga till filer i den istället för att bara läsa dem.

Efter att den privata delen var konfigurerad så var det bara kvar att konfigurera den publika delen. Detta gjordes på samma sätt som den privata, men med lite mera editering i konfigureringsfilen. För att editera den publika delen av det delade lagringsutrymmet så skrevs det in en ny rubrik med namnet [public] i konfigurationsfilen, när väl den var skapad så definierades namnet på vad den skulle heta med texten “comment = public storage” vilket döpte den delade mappen till public storage. Efter det så definierades sökvägen till mappen som skapades före tjänsten installerades med texten “path = /home/shares/public”. När tjänsten gjordes säkrare så betydde det att vi även måste definiera vilka som har rätt till att ha tillgång till det delade utrymmet vilket ledde till att texten “valid users = @users” lades till, därefter så tvingade vi alla som ansluter till den publika delen att agera under gruppen Users med texten “force group = users”. För att hålla koll på rättigheterna så gjorde vi det så att ägaren och användargruppen har rätt till att lägga till filer och läsa filerna med texten “create mask = 0660” efter det så ställer vi även in själva mappens inställningar till att låta människor spara på den med att lägga till “read only = no”.

När tjänsten var konfigurerad sparades filen och tjänsten startades om för att ta de nya inställningarna i bruk med att skriva koden “sudo /etc/init.d/samba restart” i terminalen. Eftersom vi gjorde tjänsten lite säkrare och skrev in att man måste ha en användare på tjänsten så måste vi skapa en användare. Detta gör vi i terminalen med koden “sudo smbpasswd -a pi” (Raspbian France, 2018d.) vilket skapar en användare med namnet pi utan lösenord. För att i fortsättningen kunna anslutas till de delade filerna så behöver man bara i detta fall veta om användarnamnet och lämna lösenordsrutan tom när datorn frågar en efter inloggningsuppgifterna.

För att få tillgång till de olika mapparna som blev delade via SAMBA tjänsten så kan man gå in via nätverks fliken som finns i utforskaren på en Windows 10 maskin och därifrån klicka in sig till Raspberryn. Den heter i detta fall ZPI. När man har klickat in sig i ZPI mappen så ser man de två delade mapparna (se bild 16).

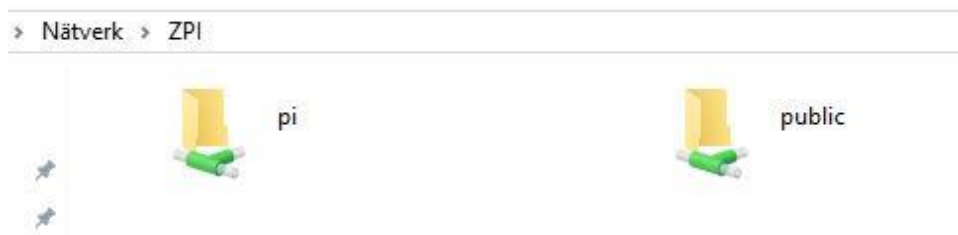


Bild 16: Delade mappar på Windows 10

Nu syns mapparna på nätverket. För att sedan göra dem mera tillgängliga på datorn så gör vi så att Windows ser mapparna som om de skulle vara lagringsutrymme som till exempel hårddiskivor genom att använda oss av en av Windows inbyggda funktioner som heter 'anslut nätverksenhet'. Den hittar man via utforskaren när man har navigerat till 'den här datorn', när man har navigerat dit trycker man på dator i balken ovanför och sedan anslut nätverksenhet (se bild 17).

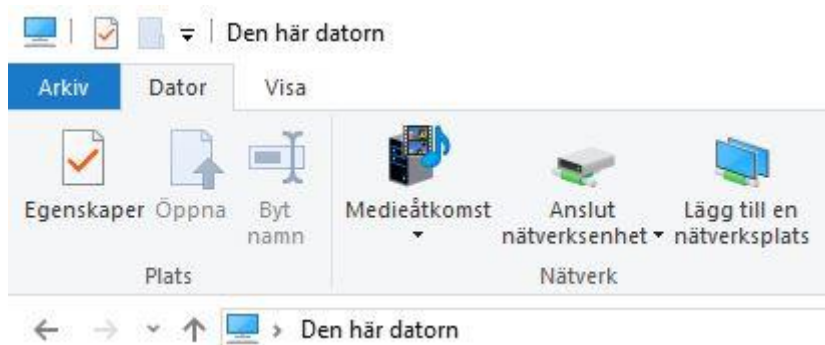


Bild 17: Anslut nätverksenhet.

När man klickat på den så öppnas det en ruta var man kan välja vilken bokstav nätverksenheten skall få, på samma sätt som hårddiskivorna har sina egna bokstäver på datorerna, exempelvis "C:/". I detta fall gav fönstret automatiskt Z: som benämning åt den här enheten, vilket lämnades som det var. Efter det skall man även fylla i sökvägen eller IP-adressen till den nätverksanslutna enheten, om man inte använder sig av ett centralt inloggningssystem som delar ut alla användarnamn och lösenord till datorerna anslutna till det. Det använde jag mig inte av i detta fall, därför markerades rutan som har texten 'Anslut med andra autentiseringsuppgifter' vilket gör att operativsystemet kräver användarnamnet och lösenordet för att ansluta till de delade mapparna. (se bild 18)

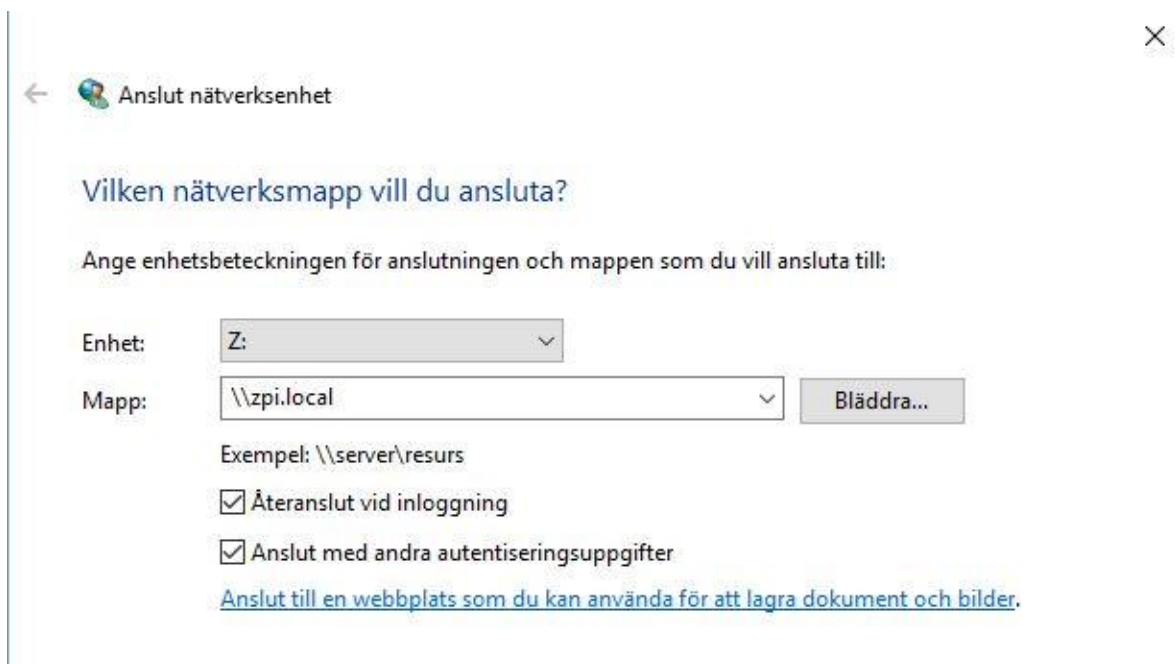


Bild 18: Anslutnings inställningar.

När allt detta var klart så anslöts den externa lagringsplatsen, exempelvis en extern hårddiskiva eller en usb-minnessticka. Detta görs genom att först ansluta den till Rasperryn. Efter att den är ansluten så kollas vad den får för benämning i systemet. Detta görs med kommandot som "dmesg", vilket ger ut en massa information angående vad som har blivit anslutet och bortkopplat från systemet. I detta fall vill vi uppmärksamma /dev/sda texter som är usb anslutningar. Om man inte har något annat anslutet till maskinen så brukar minnesstickan eller den externa hårddiskivan bli tilldelad sda1. Efter detta så formateras den om till att använda sig av ett Linux filsystem. Det görs genom att först använda oss av Linux systemets version av "säker borttagnings"-kommandot, så den inte är i användning av systemet mera, vilket är "umount /dev/sda1". Sedan formateras den med kommandot "mkfs.ext4 /dev/sda1". (se bild 19)

```

root@zpi:/home/pi# umount /dev/sdal
root@zpi:/home/pi# mkfs.ext4 /dev/sdal
mke2fs 1.43.4 (31-Jan-2017)
/dev/sdal contains a vfat file system labelled 'NAS'
Proceed anyway? (y,N) y
Creating filesystem with 1825792 4k blocks and 456960 inodes

```

Bild 19: Formatering av NAS utrymme.

När vi formaterat lagringsplatsen så skapar vi en plats var den kommer synas när den är ansluten, detta görs med kommandot `mkdir` vilket skapar en mapp. I detta fall så vill vi att mappen hamnar på samma ställe som de mappar vi tidigare har skapat för NAS tjänsten. Detta görs med att använda kommandot på detta vis ”`mkdir /home/shares/public/NASStick`”. I detta fall är `../NASStick` mappen vi skapar (se bild 20). För att sedan kunna skriva och läsa till den använder vi samma kommandon som när vi skapade SAMBA mapparna.

```

root@zpi:/home/pi# mkdir /home/shares/public/NASStick
root@zpi:/home/pi# chown -R root:users /home/shares/public/NASStick
root@zpi:/home/pi# chmod -R ug=rwx,o=rx /home/shares/public/NASStick
root@zpi:/home/pi# mount /dev/sdal /home/shares/public/NASStick
root@zpi:/home/pi# █

```

Bild 20: Skapande av mapp och behörigheter.

När vi gjort på detta vis så måste vi även meddela åt systemet att det är dit lagringsplatsen skall läggas varje gång som servern startar. Detta gör vi genom att modifiera filen ’`fstab`’ med kommandot ”`nano /etc/fstab`” vilket öppnar filen. När vi väl är inne i filen så lägger vi till texten ”`/dev/sdal /home/shares/public/NASStick auto noatime,nofail 0 0`” i slutet av filen och sparar den (se bild 21).

```

GNU nano 2.7.4 File: /etc/fstab
proc /proc proc defaults 0 0
/dev/mmcbk0p6 /boot vfat defaults 0 2
/dev/mmcbk0p7 / ext4 defaults,noatime 0 1
# a swapfile is not a swap partition, no line here
# use dphys-swapfile swap[on|off] for that
/dev/sdal /home/shares/public/NASStick auto noatime,nofail 0 0

```

Bild 21: Konfiguration av autostart.

4 Arbetsförberedelser

I förberedelser för detta arbete så spenderade jag ett flertal timmar med att söka reda på information ifall det ens är möjligt för en Raspberry att användas på det viset jag tänkt eller om den inte har tillräckligt bra prestanda. Samtidigt sparade jag sidor jag tänkte att kan innehålla information som jag har nytta av när jag börjar förverkliga idén.

Efter det så införskaffade jag en Raspberry Pi 2b och ett 16Gb SD-kort som användes som server hårdvara.

Den externa hårddisken har jag haft i flera år och plockade endast fram den och formaterade den före jag började söka efter information via nätet. Efter lite sökandet efter information så hittade jag en forumpost som talade om hur mycket ström som går åt till externa hårddiskor, vilket i sin tur ledde till att jag testade min hårddiska med Raspberryn och märkte att jag behöver något som kan ge ström åt hårddisken annat än Raspberryn. Lösningen till detta var en USB hubb som hade egen strömkälla.

4.1 Program som användes

4.1.1 Google Chrome

Med Chrome så sökte jag efter information angående hur jag skulle förverkliga mitt projekt, så jag sökte efter om det var möjligt att göra det ens och vilka tjänster som vanligtvis användes, vilka problem som förekom vanligtvis och efter lösningar om till problemen. Det slutade med att jag hade allt för många sidor att hålla i gång på Chrome hela tiden (se bild 22) så jag skapade en bokmärkes mapp för examensarbetes relaterade webbsidor.

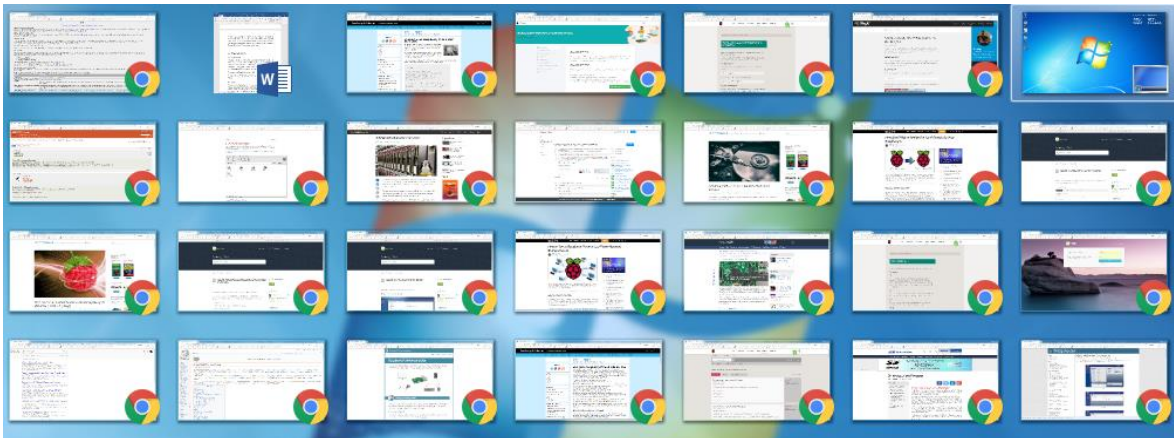


Bild 22: Alla flikar som slutligen blev använda.

4.1.2 PuTTY

PuTTY är en SSH och telnet klient (PuTTY, (u.å)) (se bild 23) vilket ger en möjligheten till att ansluta till en server som har SSH protokollet aktiverat. Detta gör att du inte behöver vara sig ha en skärm eller tangentbord direkt kopplat till servern, utan du kan använda de du har kopplat till din primära dator. Efter att du har använt PuTTY för att ansluta dig till servern via SSH så öppnas det ett fönster som har samma funktioner som om du skulle öppna terminalen på själva servern (se bild 24). PuTTY kommer användas för snabb konfiguration av olika filer i systemet och för att starta om systemet när man är kopplad till det lokala nätverket. Ifall man har behov av det grafiska skrivbordet på Raspberryn så har man möjlighet till att använda sig av RealVNC för att ansluta sig till maskinen istället för att behöva ansluta en skärm och tangentbord direkt till servern.

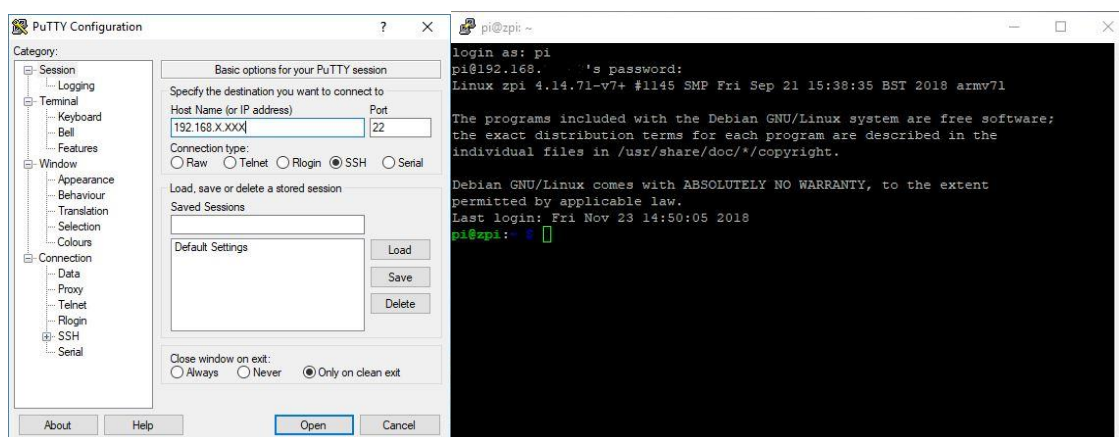


Bild 23 & 24: Putty & Putty ansluten till Raspberry.

4.1.3 RealVNC

RealVNC är ett program som ger dig möjligheten till att ansluta till en annan dator/server och styra den som om du satte vid den (RealVNC. (u.å)). Programmet kräver att VNC Viewer installeras på maskinen du vill styra ifrån och VNC Server installeras på maskinen du vill styra. Med användning av VNC så behöver man inte göra stora ändringar i ens brandvägg eller ändringar alls i sin router för att ha tillgång till sin maskin var som helst ifrån i världen så länge den är påslagen och är kopplad till Internet. Tjänsten gör det även lite säkrare när bara de som är i ens grupp har tillgång till att ansluta till maskinerna i gruppen. VNC är gratis ända upp till 5 maskiner och 3 personer som kan ansluta till dem om det inte används för kommersiellt bruk.

Under utvecklande av detta arbete så användes VNC för funktionen som gör att det är som om man skulle sitta vid datorn själv. På grund av att det finns vissa kommandon som inte kan utföras om man kör dem via SSH-klienten, måste man göra ändringar i inställningarna, det var inte något som var viktigt för detta projekt.

För att använda VNC så behöver man inte göra något annat än att logga in på sitt personliga användarkonto på serverklienten som färdigt är installerad på Raspbian, efter att man har loggat in på server klienten kan man ladda ner VNC viewer på datorn man vill styra från. (se bild 25)

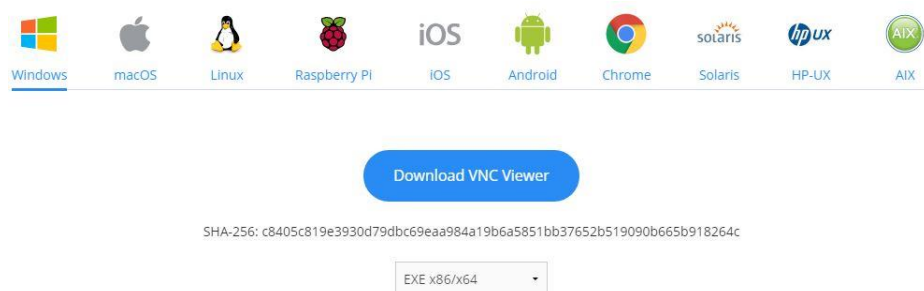


Bild 25: VNC Viewer alternativ.

Det kan man göra från VNC's hemsida, var man även hittar server klienten, om man vill installera den på någon annan maskin. Efter att man har laddat ner viewer klienten så behöver man bara logga in på den med samma konto som man har loggat in på server klienten med så visar viewer klienten de maskiner som kör server klienten just då (se bild 26).

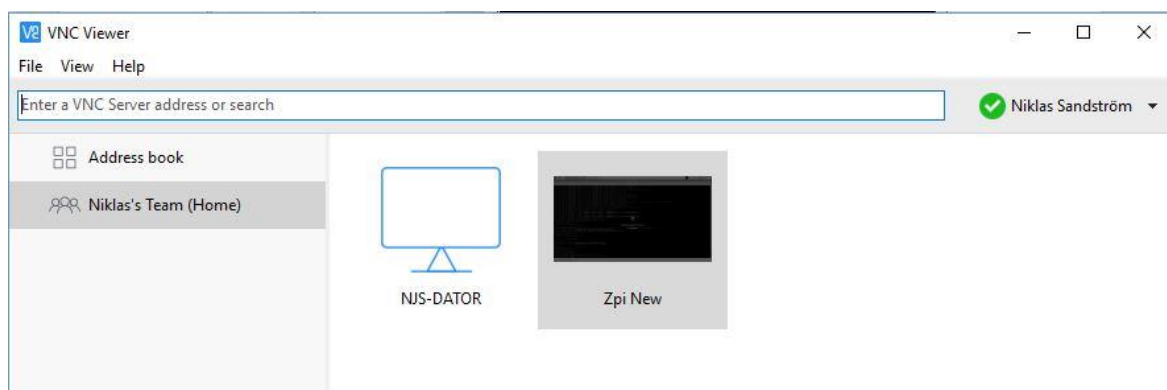


Bild 26: VNC Viewer med datorer.

Efter det är det bara att dubbel klicka på den server man vill kontrollera och slå in lösenord och användarnamnet för maskinen. Vill man inte hamna skriva in dem varje gång så kan man spara dem i viewer klienten så att man i fortsättningen ansluter direkt till servern. När

man ansluter till servern så öppnas en ny ruta var man ser skrivbordet på datorn man styr (se bild 27), man kan ändra storlek på fönstret och kan även använda sig av kortkommandon som ctrl+c och ctrl+v mellan de olika datorerna om man så vill.

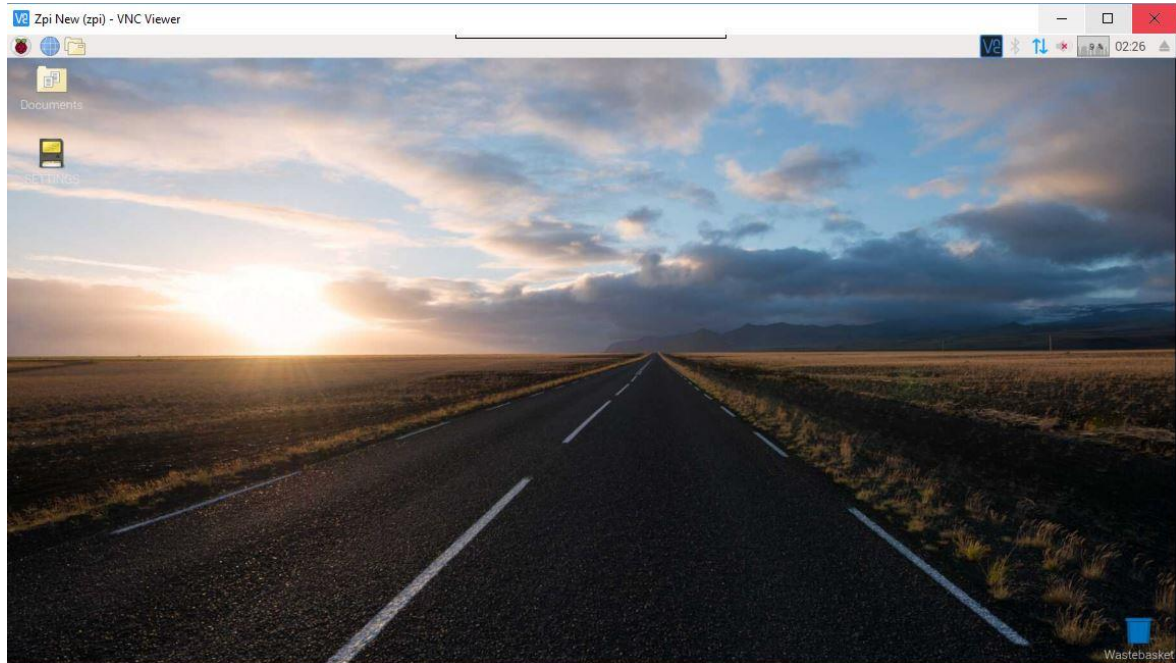


Bild 27: Ansluten till Raspberry via VNC.

5 Testning av NAS tjänsten

Efter att allting var klart så med NAS tjänsten så gjorde jag ett par olika test, jag testade överföringshastigheten på den först detta gjordes med en testfil på 182Mb. I medeltal så gick överföringen med hastigheten ~10Mb/s båda vägarna, vilket ledde mig till att helt stå över testandet av musikstreaming från Raspberryn. Istället för att testa musikstreaming så gjorde jag ett test med en HD-film för att kräva lite mera av Raspberry servern. Efter att jag överfört filmen till den vilket inte tog mera än ungefär 5 minuter så kunde jag testa det. Min slutsats var att det fungerade smärtfritt att köra åtminstone en video stream från Raspberryn, vilket är den största mängden som jag kan tänka mig att komma streamas från servern på en gång. Under testet så prövades också på att hoppa till helt olika tider i videoklippen. Den största väntetiden som man märkte av var inte längre än ungefär en sekund.

6 Kritisk granskning

Ifall jag skulle börja göra om detta projekt igen så skulle jag söka efter andra alternativ till tjänsterna jag just nu har i användning, under projektets gång så har jag hamnat läsa många olika forum poster och artiklar angående om problemen som folk har haft med just de tjänster jag har i användning just nu och vad de har hittat för lösningar till de problemen, om de inte har bytt tjänst helt, sedan har jag hittat många artiklar som har talat om hur mycket bättre tjänster det finns att använda som inte är så krävande på systemet som de som jag just nu har i användning, fastän de som jag nu använder inte är alltför prestandakrävande enligt mig, men med en så billig och svag maskin som de körs på så räknas allting som är lite lättare på systemet som ett stort plus.

Själva arbetet skulle jag göra mera systematiskt, den här gången gjorde jag det på det viset att jag hoppade från en tjänst till en annan konstant istället för att göra en klar i taget, vilket ledde till en stor mängd med lösa trådar så att säga. Det krävdes ett flertal timmar till att hitta alla saker som har blivit på hälft och inte ens i skrivande stund så är jag säker på om allt är exakt så som det skall vara med alla tjänster, de fungerar, men fungerar de som de borde egentligen är frågan.

7 Avslutning

I detta arbete har jag byggt en LAN-baserad server på en Raspberry Pi Model 2b dator och behandlat dess användningsområde. På servern har det installerats en webbserver med en PHP läsare och påbörjats konfigureringen av en MySQL server som skulle installeras men aldrig blev färdig på grund av att problemen med den var för tidskrävande för min tidsram att bli färdig med arbetet.

Jag har även installerat en NAS server som har en publik mapp som alla på det lokala nätverket har tillgång till om de vet användarnamnet till mappen. Till NAS servern har kopplats en externhårddisk som blev utbytt till en minnessticka i senare skede. Server helheten konfigurerades till att acceptera SSH anslutningar för konfigurering och service, utöver det har det även installerats en tjänst för fjärrstyrning med VNC.

NAS delen av servern har testats med överföring av en över 100Mb fil och teststreaming av en Full HD film, vilket har visat att Raspberryn är mera än kapabel till att användas som hemma server eller som mediaserver.

Slutligen var detta även ett projekt som skulle ge mig möjligheten till att se vad Raspberryn egentligen är kapabel till. Det lyckades kanske inte så bra när servern byggdes med tanke på att få den att fungera. Det ledde till att Raspberryn inte belastades prestandamässigt till den grad som den borde blivit för att se gränserna på den.

Källförteckning

Apache, (u.å). *http server project* [online]

[Hämtat:30.11.2018]

PuTTY, (u.å). *Download PuTTY* [online]

<https://www.putty.org/> [Hämtat:30.11.2018]

Raspberry Pi, (u.å). *Downloads* [online]

<https://www.raspberrypi.org/downloads/> [Hämtat:30.11.2018]

Raspbian France 2018a. *Apache installation* [online]

<https://howtoraspberrypi.com/how-to-install-web-server-raspberry-pi-lamp/>

[Hämtat:30.11.2018]

Raspbian France 2018b. *Php installation* [online]

<https://howtoraspberrypi.com/how-to-install-web-server-raspberry-pi-lamp/>

[Hämtat:30.11.2018]

Raspbian France 2018c. *MySQL installation* [online]

<https://howtoraspberrypi.com/how-to-install-web-server-raspberry-pi-lamp/>

[Hämtat:30.11.2018]

Raspbian France 2018d. *Create a NAS* [online]

<https://howtoraspberrypi.com/create-a-nas-with-your-raspberry-pi-and-samba/>

[Hämtat:30.11.18]

RealVNC, (u.å). *Secure, reliable screen sharing* [online]

<https://www.realvnc.com/en/> [Hämtat:30.11.2018]

SAMBA, (u.å). *What is SAMBA* [online]

https://www.samba.org/samba/what_is_samba.html [Hämtat:30.11.2018]

FileZilla, (u.å). *Overview* [online]

<https://filezilla-project.org/> [Hämtat:30.11.2018]

Figurförteckning

Bild 1: Raspberry Pi model 2B

Bild 2: Micro SD korts adapter

Bild 3: Hårdvara (inte med på bild HDMI kabel & skärm)

Bild 4: Operativsystemen som NOOBS stöder.

Bild 5: Nedladdningsalternativ

Bild 6: SDFormatter V4.0

Bild 7: SDFormatter formaterings inställningar

Bild 8: Installation av grafiskt gränssnitt.

Bild 9: Installation av Chromium via terminalen i Raspbian.

Bild 10: index.html på Chromium.

Bild 11: index.php på Chromium.

Bild 12: FileZilla hemsida.

Bild 13: FileZilla anslutnings inställningar.

Bild 14: Skapande av mappar

Bild 15: Installation av SAMBA

Bild 16: Delade mappar på Windows 10

Bild 17: Anslut nätverksenhet.

Bild 18: Anslutnings inställningar.

Bild 19: Formatering av NAS utrymme.

Bild 20: Skapande av mapp och behörigheter.

Bild 21: Konfiguration av autostart.

Bild 22: Alla flikar som slutligen blev använda.

Bild 23: PuTTY

Bild 24: PuTTY ansluten till Raspberry.

Bild 25: VNC Viewer alternativ.

Bild 26: VNC Viewer med datorer.

Bild 27: Ansluten till Raspberry via VNC.