Network Attached Storage

D. Leeuw

24 januari 2024
v.0.2.0

Over dit Document

0.1 Leerdoelen

0.2 Voorkennis

Inhoudsopgave

Hoofdstuk 1

Inleiding

Hoofdstuk 2

Bestanden delen over het netwerk

2.1 FTP

2.2 HTTP/HTTPs

Hoofdstuk 3

Network Filesystems

3.1 NFS - Network File System

NFS stamt uit 1984 en is bedacht door Sun Microsystems. NFS is een netwerk bestandssysteem dat het mogelijk maakt om bestanden over het netwerk op een andere server te benaderen alsof het lokale bestanden zijn. Dit in tegenstelling met het tot dan toe veel gebruikte FTP waarbij je de een server actief moet benaderen om een bestand te downloaden voordat je het gebruiken kan.

NFS heeft twee protocollen, een om het bestandssysteem te koppelen (mount) en een om het gemounte bestandssysteem te benaderen (nfs). NFS gebruikt een aantal Remote Procedure Calls (RPCs) voor de toegang tot het bestandssysteem zoals het lezen en schrijven van bestanden.

NFS server houdt geen status bij van de clients, alleen welke client welke share gemount heeft. De client moet dus bij elke opdracht alle gegevens meesturen. Dit heeft als groot voordeel dat je een NFS server kan herstarten zonder dat clients hiervan een verstoring ondervinden. Een nadeel is dat NFS dus niet zelf bijhoud wie welk bestand benaderd en er dus twee gebruikers hetzelfde bestand kunnen schrijven. Het zogenaamde file-locking moet door een extern proces gebeuren (Network Lock Manager). Vanaf NFS 4 is locking wel een onderdeel van het protocol.

3.1.1 Security

Tot versie 4 kende NFS geen authenticatie methode. Toegang tot het bestandssysteem werd volledig overgelaten aan het onderliggende bestandssysteem zelf, dus de rechten op files en directories. NFS was dus alleen geschikt voor gebruik op een lokaal netwerk waar elke server en client

over dezelfde gebruikersdatabase beschikt. Sinds versie 4 kan er gebruik gemaakt worden van kerberos voor authenticatie.

3.1.2 Interoperability

NFS is vanaf versie 2 (RFC1094) beschreven in RFCs en dus een open protocol dat door iedereen geimplementeerd kan worden. Daarnaast is het een heel simpel protocol waardoor er voor bijna elk operating system een NFS client is en er is vaak ook een NFS server beschikbaar.

3.2 SMB - Server Message Block

SMB maakt het mogelijk om bestanden en printers te delen met het netwerk en heeft tevens een interproces communicatie mogelijkheid in de vorm van een door Microsoft ontwikkelde versie van RPC genaamd MS-RPC.

Het SMB-protocol stamt uit 1983 en was oorspronkelijk ontwikkeld bij IBM. Microsoft gebruikte een verder door ontwikkelde versie in zijn Windows Operating System. SMB1 die in 2003 ook bekend was onder de naam CIFS (Common Internet File System) was een erg chatty protocol, waardoor het veel bandbreedte in gebruik nam en dus niet erg geschikt was voor gebruik over Internet links. Voor het delen van lokale resources werd het echter veel gebruikt. SMB1 zou inmiddels niet meer gebruikt moeten worden omdat het vele kwetsbaarheden bevat. SMB2 verminderde de chattyness van het protocol

3.2.1 Security

Al vroeg in de geschiedenis van SMB1 (Windows NT4 service pack 3) kreeg SMB de mogelijkheid van SMB signing. Dit betekende dat van een packet de authenticiteit vastgesteld kon worden. Hiermee werd een man-in-the-middle attack bijna onmogelijk.

SMB 3.0 bracht end-to-end encryption en SMB 3.0.2 geeft je de mogelijkheid om SMB1 niet meer te ondersteunen.

3.2.2 Interoperability

Buiten de Microsoft wereld wordt SMB gebruikt in het SAMBA-project, wat een Open Source project is om Unix-achtige systemen zoals Linux

te verbinden met de Windows-wereld. Ook Apple heeft zijn eigen SMB-protocol implementatie. Eerste werd SAMBA gebruikt, maar sinds OS X 10.7 gebruiken ze een eigen implementatie met de naam SMBX.

3.2.3 Roaming profiles

Hoofdstuk 4

Network Fileserver

4.1 NAS - Network Attached Storage

Een NAS of een Network Attached Storage is eigenlijk een fileserver in een doosje. In de juiste termen heet het een appliance. Een appliance is hardware en software in één systeem, meestal met een webinterface voor de configuratie. Je koopt een functionaliteit en niet hardware, een os en applicaties. Een NAS appliance heeft een ethernet interface en kan je direct aan je netwerk verbinden en via dat netwerk stelt een NAS dan bijvoorbeeld een SMB Netwerk share ter beschikking.

4.1.1 Opdracht - TrueNAS

Vanaf juli 2021 bestaat de FreeNAS website niet meer en heet het product TrueNAS Core. TrueNAS core is beschikbaar vanaf de https://www.truenas.com website. Om te kunnen downloaden kan je inloggen met een social media account, je kan een e-mail adres achter laten of op de link 'No Thanks. Take me to the Download Page' klikken. Lees de hardware requirments en zoek daarbij de juiste hardware of maak een VM met de juiste specificaties Hou er rekening mee dat je een disk hebt waarop TrueNAS komt te draaien en tenminsten één disk voor data opslag.

Download de ISO, koppel deze aan je VM of maak er een opstartbare USB-stick van als je eigen hardware gebruikt en installeer TrueNAS op je hardware danwel je VM. Op de download pagina staat een handleiding en een filmpje om je opweg te helpen.

Voeg tenminste één (virtuele) harddisk toe en benader je nieuwe NAS met een client computer zodat je data kan lezen en schrijven naar je TrueNAS machine. Desktop machines hebben een beperkte opslagcapaciteit

en hebben weinig tot geen back-up mogelijkheden. Door de data te centraliseren op een fileserver kunnen we de capaciteit van de desktop uitbreiden zonder daar fysiek toegang tot te hebben. Daarnaast kunnen we de opgeslagen data op de fileserver eenvoudig centraal backuppen.

Een fileserver is een computer, server, in het netwerk die via een bepaald protocol bestanden aanbiedt aan gebruikers. In Microsoft Windows netwerken kan dit een SMB, Server Message Block, zijn en in een Linux omgeving bijvoorbeeld een NFS, Network File System, server zijn.

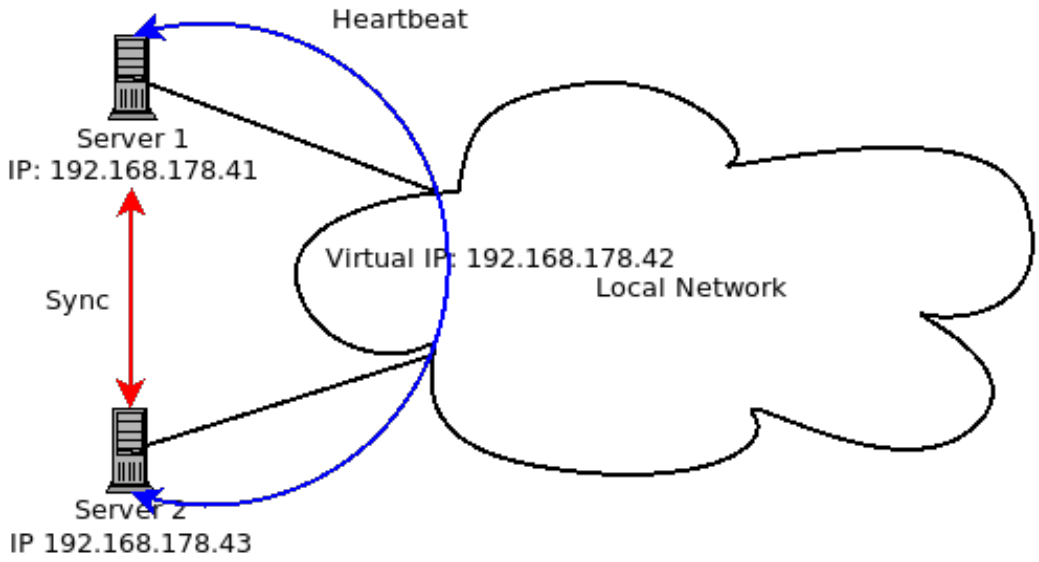
Het kenmerk van een fileserver is dat er een bepaald protocol (SMB, NFS) nodig is dat een bestandssysteem aanbiedt. Een gebruiker mount of mapped het filesysteem naar zijn lokale systeem en kan erop werken alsof het onderdeel is van zijn eigen machine.

4.2 Failover in a Heartbeat

Om te zorgen dat bestanden ook beschikbaar zijn als een server uit valt is het noodzakelijk om een systeem reduntant (dubbel) uit te voeren. Als dan de ene file server uit valt kan de andere zijn functie overnemen. De vraag is hoe de gebruiker, of beter hoe het systeem van de gebruiker, weet dat hij de server die hij gebruikte er niet meer is en hij gebruik moet maken van het vervangende systeem.

Binnen IP netwerken wordt dat gedaan door gebruik te maken van een zogenaamd virtueel IP adres. Je kan meer dan één IP adres toewijzen aan een netwerkkaart en dat is precies wat we gebruiken om failover te bereiken. We hebben 2 file servers die ieder hun eigen IP adres hebben en er is een IP adres dat ze "delen". Zolang er niets aan de hand is heeft één van de file servers ook het virtuele IP adres op zijn netwerk interface, dus deze server heeft op één netwerkkaart 2 IP adressen. Het virtuele IP adres is was wat de gebruikers gebruiken om naar de file server te verbinden. Als er een storing optreedt in server 1 dan merkt server 2 dat. Server 2 zet dan het virtuele IP adres op zijn netwerkkaart en alle vragen om bestanden komen bij server 2 uit (zie 4.1).

Twee dingen zijn hierbij van belang, namelijk dat de data op server 1 gelijk is aan die op server 2. In figuur 4.1 synchroniseren server 1 en 2 over een (rode) ethernet cross-cable, op deze manier verstoort de syncronisatie niet de bandbreedte naar de gebruikers. Het andere dat van belang is is dat server 2 moet weten wanneer server 1 niet meer bereikbaar is. Dat laatste wordt bereikt met een heartbeat, zeg maar een soort ping waarbij de servers elkaar in de gaten houden. Als server 2 de heartbeat van server 1 niet meer hoort dan neemt het virtuele IP adres over. De heartbeat kan



Figuur 4.1: File Server Failover

over een speciale kabel gaan of over de al bestaande netwerken tussen de twee servers.

Hoofdstuk 5

Web storage

Een oplossing voor het object storage probleem is door gebruik te maken van het web of zoals dat tegenwoordig heet de cloud om je data in op te slaan. De webbrowser dient als frontend voor het opslagsysteem en het maakt dan niet meer uit wat we gebruiken als storage.

5.1 webdav

Het oorspronkelijke idee van het World Wide Web (WWW) was om bestanden met elkaar te delen en er gezamenlijk aan te kunnen werken. Het werd echter een alleen lezen systeem tot in 1996 Jim Whitehead het W3C overtuigde om een paar sessies te houden over de mogelijkheid van het schrijven van documenten. Daaruit kwam WebDAV voort wat staat voor Web-based Distributed Authoring and Versioning. Kort gezegd komt het erop neer dat WebDAV een protocol is om documenten te schrijven en de voorzien van een versie. Dat versioning bleek echter voor de eerste versie te ingewikkeld en men concentreerde zich op het gezamenlijk werken aan documenten. Later is er alsnog een versioning standaard gekomen.

WebDAV voegt een aantal request methods toe aan HTTP zodat het gebruikt kan worden als bestandssysteem voor lezen en schrijven. De "methodes"die WebDAV toevoegt zijn:

1. PROPFIND haal de XML properties van een object op. Dit kan ook een complete filesystem tree zijn.

2. PROPPATCH wijzig 1 of meer properties van een object in 1 actie

3. MKCOL maakt een collectie aan, een collectie kan een map of directory zijn

4. COPY kopieert een object van een URI naar een ander
5. MOVE hernoemt een object van een URI naar een ander
6. LOCK vergrendel een object
7. UNLOCK ontgrendel een object

De functie om een bestand te schrijven naar een webserver bestond al in de PUT-methode.

5.2 Amazon S3 bucket

5.3 Google Drive

5.4 One Drive & Microsoft Azure BLOB

Hoofdstuk 6

Object Storage

Met het groeien van filesystemen wordt het steeds moeilijker om data aan te bieden. Op het moment dat we honderden terabytes of zelfs petabytes aan opslag hebben hebben we ook enorm veel bestanden. Mensen verdelen de data die zij opslaan niet evenredig over een bestandssysteem, sommige mappen bevatten veel bestanden en andere heel weinig. Het indexeren van al deze bestanden, als we bijvoorbeeld een listing opvragen van het bestandssysteem, duurt met de groei van het systeem steeds langer. De oplossing voor dit probleem is het niet meer gebruik maken van een bestandssysteem, maar van andere technieken. Een van die technieken is het gebruik van object storage.

Object storage wordt voornamelijk gebruikt als backend voor applicaties. Als je een bestand in een object storage systeem opslaat krijg je van het systeem een ID terug. Meestal in de vorm van een URL, bijvoorbeeld: `https://my.nextcloud.local/index.php/s/xLiiLba2gximHBt` Dit is een verwijzing naar het object op het systeem. Zoals je kan zien is dit niet lekker makkelijk te onthouden, vandaar dat dit vaak door applicaties gebruikt wordt, die dan een database gebruiken om een relatie te leggen tussen het object en het door de gebruiker opgeslagen bestand.

Index