



HØGSKOLEN I SØR-TRØNDELAG

Avdeling for informatikk og e-læring

Bachelorprosjekt 2010

Tittel - norsk: Virtualisering med Hyper-V. Installasjon, konfigurering og drift av Exchange 2010, SCVMM og PowerShell.

Oppgave nr.: 6E

Tittel - engelsk: Virtualizing and Hyper-V management. Installation, configuration and management of Exchange 2010, SCVMM and PowerShell.

Oppgavestiller:

Kontaktperson: Stein Meisingseth

Telefon: 73 55 95 62

E-postadresse: stein.meisingseth@hist.no

Postadresse: E. C. Dahls gt. 2

Studenter (navn og klasse): Bjørn Inge Vikhammermo Berg (3BADR), Marius Foss (3BADR)

Veileder ved HiST:

<i>Stikkord som beskriver innholdet i oppgaven:</i>	<i>Windows Server 2008R2</i>	<i>Active Directory</i>	<i>Hyper-V</i>
	<i>SCVMM 2010</i>	<i>Failover Clustering</i>	<i>Exchange 2010</i>
	<i>PowerShell</i>		

Sammendrag: Installasjon av Server 2008 R2 med Hyper-V som virtualiseringsgrunnlag. Oppsett av cluster for Hyper-V. Installasjon og bruk av SCVMM mot Hyper-V og clustering. Oppsett og bruk av Exchange 2010 med Edge og Forefront Security. Oppsett og generelt bruk av PowerShell. PowerShell for styring av Active Directory, SCVMM og Exchange 2010.

Produserte resultater er tilgjengelig på følgende måte (sett kryss):

☒ *fins i arkiv på avdelingsbiblioteket, åpent tilgjengelig*

☐ *fins i arkiv på avdelingsbiblioteket. Tilgjengelig etter avtale med oppgavestiller.*

☒ *publisert på www - oppgi URL: <http://bachelor2010.mafo.net/>
Kun URL'er som er rimelig permanente er av interesse.*

Forstudierapport

Virtualisering og
bruk av software

Gruppe 6E – Bachelor 2010

Marius Foss, Bjørn Inge V. Berg

Innholdsfortegnelse

1. Hensikten med dokumentet.....	4
2. Bakgrunn for prosjektet	5
2.1 Beskrivelse av problemer og behov	5
2.2 Kort om dagens systemer og rutiner.....	6
3. Prosjektmål.....	7
3.1 Effektmål	7
3.2 Resultatmål.....	7
3.3 Prosessmål.....	7
3.4 Prosjektets omfang	7
3.5 Milepæler	7
4. Rammebetingelser og interessenter.....	8
4.1 Rammebetingelser	8
4.2 Interessenter	8
5. Kritiske suksessfaktorer	9
6. Risikoanalyse	10
6.1 Hensikt med risikoanalysen.....	10
6.2 Uønskede hendelser.....	10
7. Kost/nytte analyse	12
7.1 Kvantifiserbar og ikke-kvantifiserbar nytte.....	12
7.2 Estimerte kostnader	12
7.2.1 Kostnader for prosjektet.....	12
7.2.2 Nytteanalyse	13
8. Prosjektorganisering	14
8.1 Utstyr som trengs.....	15
9. Retningslinjer og standarder	16
9.1 Krav til dokumentasjon	16
9.2 Krav til kvalitetsgjennomganger.....	17
9.3 Krav til standarder og metoder	17
10. Anbefaling om videre arbeid	18

Figurer

Figur 1 - Organisasjonskart	6
Figur 2 - Utsnitt av nettverk	6
Figur 3 - Risiko og sannsynlighet	11
Figur 4 - Prosjektorganisasjon	14

Tabeller

Tabell 1 – Dokumentasjon av prosjektet	16
--	----

Tillegg

[Tillegg 1 – Gantt-diagram](#)

1. Hensikten med dokumentet

Forstudierapporten inneholder en beskrivelse av en tenkt bedrift vi skal arbeide med for å sette opp og drifte systemer for. Bedriften simuleres slik en bedrift vil være til dags dato.

Hensikten med forstudierapporten er å

- Komme fram til prosjektets mål.
- Komme fram til lønnsomhet av prosjektet.
- Få oversikt over strategi og overordnede planer.
- Tid og ressursbehov for gjennomføring av prosjektet.

Dette dokumentet skal beskrive grunnlag og bakgrunn for prosjektet, hva vi vil oppnå, rammebetingelser for prosjektet, organisering og beregninger av prosjektet for at dette skal kunne lykkes. Vi vil ut i fra dette gi grunnlag for anbefalinger av videre arbeid med dette prosjektet.

- **Bakgrunn for prosjektet** – Beskrivelse av bedriften, hensikten med prosjektet og hvorfor vi har tatt for oss akkurat dette prosjektet.
- **Prosjekt mål** – Beskrivelse av de mål vi ønsker å oppnå, hva som må gjøres for å oppnå denne effekten og effekten på de som arbeider med prosjektet.
- **Rammebetingelser** – Beskrivelse av absolutte krav som stilles til prosjektet og krav til gjennomføringen og resultatets omfang som vil ha innflytelse på planene for prosjektet.
- **Kritiske suksessfaktorer** – Beskrivelse av faktorer som er avgjørende for prosjektet.
- **Risikoanalyse** – En analyse for risikofaktorer som kan hindre at prosjektet lykkes.
- **Prosjektorganisering** – Beskrivelse av arbeidsfordeling og roller i prosjektet.
- **Retningslinjer og standarder** – Beskrivelse av krav og retningslinjer til dokumentasjon og kvalitetsgjennomgang.
- **Anbefaling av videre arbeid** – En vurdering av anbefalinger om prosjektet skal føres videre.

2. Bakgrunn for prosjektet

Prosjektet dreier seg om å oppgradere serversoftware til nyere versjoner. De nye systemene skal samtidig bli beregnet for fremtidige besparelser. Det dreier seg om å flytte servere over til virtuelle maskiner. Bakgrunn for hele prosjektet en total oppgradering og ny tilrettelegging for selve bedriften siden de hittil bare har lagt til ved hjelp av fysiske tjenere og det som har vært tilgjengelig og med lave kostnader. Dette skal også være et skall for utrulling av liknende system i andre avdelinger.

Det skjer stadig oppgradering og utvikling av ny programvare. Servere blir mer avanserte og får flere muligheter. Oppgradering og skifting av programvare på tjenere blir derfor utfordrende og kritiske for store organisasjoner med tanke på mengden data som skal være sikret og ha minst mulig utilgjengelighet. Effektivisering av dette kan gjøres ved script og kode som utfører store operasjoner på en logisk måte.

Bedriften **MafoBerg Systems AS** er en mellomstor bedrift med 444 ansatte og lokalisert i Trondheim. De arbeider med prosjekter for utvikling av datasystemer for alle typer bedrifter. Dette inneholder også oppgradering og andre konsulenttjenester.

Den har en ledelse på fire, en administrasjon og lønnsavdeling, Selgere, IT-avdeling for intern og ekstern drift, spesialister i prosjektgrupper, 1. linjesupport for kunder, 2. og 3. linje support og en stor konsulentgruppe dedikert til en god del bedrifter både som utleide IT-drifere og snarlig tilkallingshjelp.

2.1 Beskrivelse av problemer og behov

I tiden som bedriften har vokst, har man ansatt folk til avdelinger der det har vært behov for folk. Databasene og dataene har vokst deretter. Grupperingene av organisasjonen har utviklet seg og blitt mer spesialiserte.

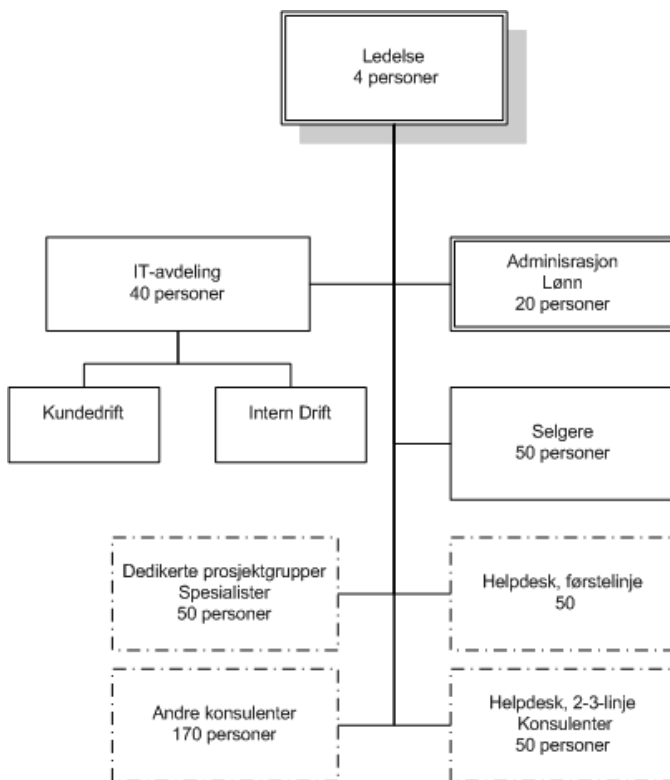
De har hele tiden hatt fokuset på å opprettholde tjenester for sine kunder og å opprettholde effektiviteten av konsulenttjenester. Derfor har ikke MafoBerg Systems AS satset fullt ut på oppgradering av sine egne tjenere.

Vi har fått i oppdrag å fornye organisasjonens servere og epost-system og automatisere overgangsfaser. Dette innebærer å ta en del av organisasjonens system å oppgradere for å se at dette går smertefritt og blir vellykket. Når prosjektet er fullført, vil automatiserte rutiner gjøre oppgraderingen av andre avdelinger enklere. Disse skal kunne styres fra it-avdelingen ved hjelp av programmer og script.

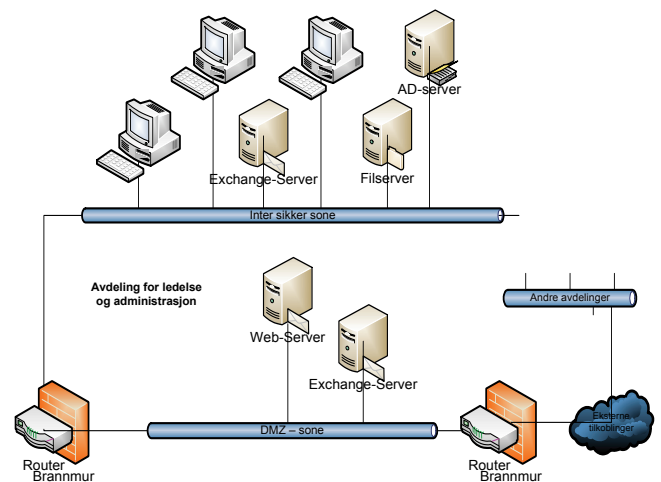
2.2 Kort om dagens systemer og rutiner

Dagens systemer består av 1 hoveddomenekontroller i hovedavdelingen, 1 read-only domenekontroller i hver avdeling, 1 exchange-server per avdeling, men 2 i hovedavdelingen, webtjenere i hovedavdelingen, filtjenere i hver avdeling og en rekke klienter både trådløse og kablede.

Rutiner og sikkerhet har utviklet seg med tiden. Hver avdeling har sine policyer og regler. Disse er utviklet i forhold til hvordan hver avdeling arbeider og hva som gjøres i hver avdeling.



Figur 1 - Organisasjonskart



Figur 2 - Utsnitt av nettverk

3. Prosjektmål

Målet med dette prosjektet er at vi skal få en oversikt over hvordan en sikrer Windows-servere, med hovedvekt på Exchange-Server 2010, og finne en måte å automatisere opprettelse av mange brukere i et domene ved hjelp av Powershell-teknologien til Microsoft.

3.1 Effektmål

- Effektivisering av innlegging av brukere.
- Økt sikkerhet, og bedre filtrering av spammail.
- Redusere strømbruk for servere, når det er liten aktivitet.
- Redusere tid det tar for administrasjon og ledelse og benytte epost-systemet.
- Effektiv utrulling av liknende systemer ved bruk av klargjorte script og oppsett.

3.2 Resultatmål

Våre resultatmål går ut på å finne regler til konkrete tiltak for bedring av sikkerheten, slik som regler for Exchange-servere, hvordan vi kan migrere virtuelle maskiner mellom fysiske hoster. Og det skal lages script som kan kjøres av administrasjonen for å legge inn og endre brukerkontoer. Dette skal fullføres innen 25. Mai 2010, som er prosjektets sluttdato.

3.3 Prosessmål

Vi har som mål at vi skal jobbe som en helhet der vi deler arbeidet slik at begge gruppedeltakere får gjort omtrent like mye innenfor hver av de ulike delene prosjektet består av. Vi setter oss som mål å oppnå et sikkert windows-miljø og bli mer effektiv for å rulle ut denne programvaren slik at videre prosjekt for andre avdelinger kan blir effektive.

3.4 Prosjektets omfang

Vi skal kun ta for oss en liten del av bedriftens infrastruktur med sikte på å rulle ut våre endringer hos hele bedriften, hvis det skulle vise seg at prosjektet blir en suksess. Vi skal rulle ut forbedringer på epost-systemer og administrasjon av brukerkontoer og automatiseringsrutiner ved hjelp av Powershellscript. Og administrering av virtuelle servere og administrering av disse.

3.5 Milepæler

Gjennom prosjektperioden er det en del hovedoppgaver som skal gjennomføres. Noen av de går parallelt, mens noen krever at andre oppgave er fullført før de kan påbegynnes. Se nærmere på Gantt-diagrammet i Vedlegg 1.

4. Rammebetingelser og interesser

Vi vil ta for oss forskningsomgivelser, krav til datoer, dokumentasjonsomgivelser og andre nødvendige rammer.

4.1 Rammebetingelser

Prosjektet er beregnet til å vare frem til sluttrapporten skrives. Dette er 14. mai 2010. Da har vi tid til å klargjøre og bearbeide all informasjon til et komplett prosjekresultat. Prosjektets sluttdato er den 25.mai 2010. Da skal all dokumentasjon, data og tilhørende for prosjektet være levert.

Dokumentering vil foregå med Microsoft Office 2007 programvare og enkle ikkeformaterbare editorer.

Kostnadsrammene for prosjektet vil bli overslagsmessig, og den er beregnet kun på det utstyr og programvare for utførelse på dette tidspunkt. All bearbeidelse til utrullinger i andre avdelinger og liknende institusjoner kan bli forskjellig.

Dette prosjektet vil ikke ta for seg myndighetskrav og/eller krav fra datatilsynet. Dette er allerede tatt hånd om siden rammer for typen kunder allerede er lagt tidlig i denne bedriften.

Rammer for driftsmiljøet vårt vil bli kun det som er nevnt i systemkravdokument og maskinvaredokument. Dette er til bruk kun for utførelse i den bestemte seksjonen i bedriften. Prosjektet skal ikke omfatte hele bedriften både av sikkerhetsmessige grunner og forsøksgrunner. Det vil være lurt å se at denne delen blir godkjent før man ruller videre ut. Vi holder oss derfor til to domenekontrollere, tre exchange-servere, server for SCVMM og noen klienter til test.

4.2 Interesser

Oppdragsgiver: MafoBerg Systems AS

Veileder:

Stein Meisingseth
Tlf: 926 06 092
Epost: stein.meisingseth@hist.no

Kvalitetskontrollør:

Stein Meisingseth
Tlf: 926 06 092
Epost: stein.meisingseth@hist.no

Prosjektleder og prosjektdeltakere:

Marius Foss
Tlf: 416 77 706
Epost: mariusfo@stud.hist.no
Bjørn Inge Vikhammermo Berg
Tlf: 970 98 045
Epost: bivb@stud.hist.no

5. Kritiske suksessfaktorer

For at vi skal lykkes med gjennomføringen av prosjektet er det viktig at vi får tilgang til maskinvare og lisenser, slik som definert i hovedoverskrift 8.1 i dette dokumentet. Det er også viktig at de tiltakene som vi innfører med oppdateringer følges opp slik at serverne ikke blir stående usikret.

Når systemene er fullt implementert, må de tas i bruk av de ansatte. Da bedriften er IT-sentrert, regner vi med at dette ikke blir noe problem. Det kan likevel være slik at de trenger noe opplæring for å kunne ta i bruk de mest avanserte funksjonene i Powershell, og være mest mulig effektiv med tanke hyper-v og strømsparing.

Systemet må dessuten være skalerbart, slik at det ikke oppstår tregheter, når brukerantallet utvides.

6. Risikoanalyse

6.1 Hensikt med risikoanalysen

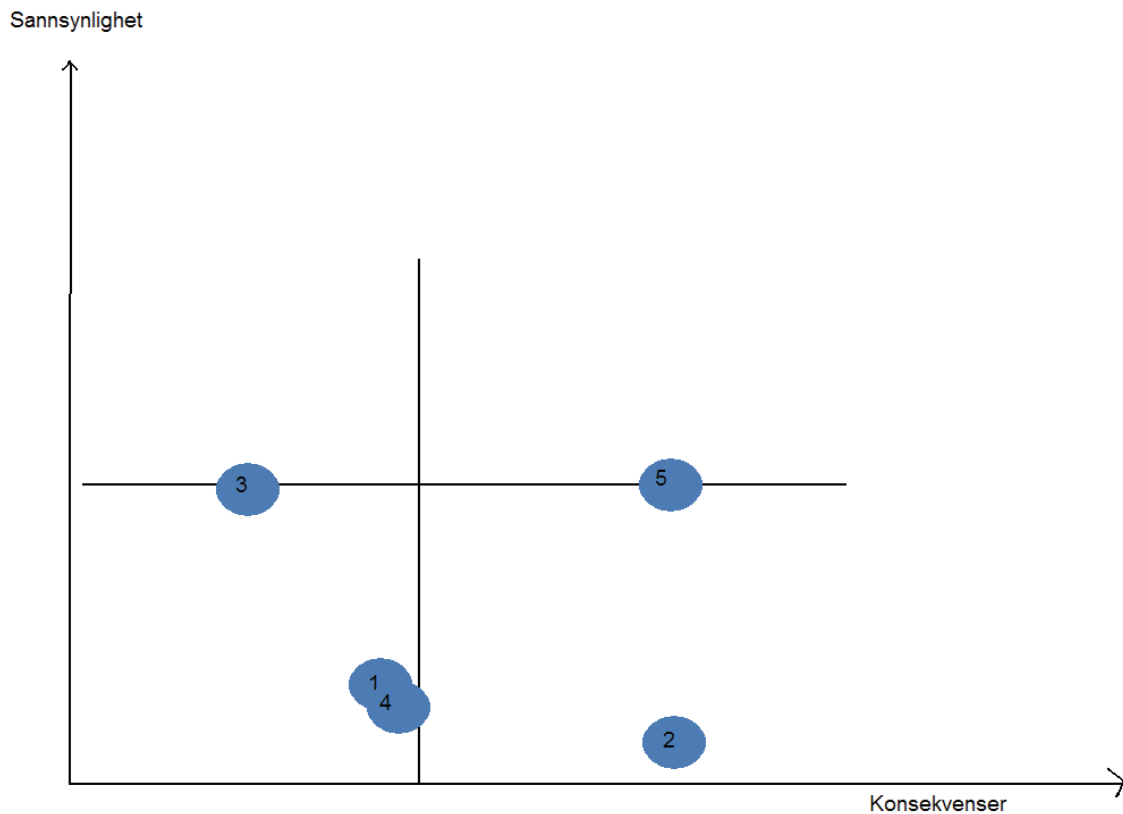
Dette dokumentet er en risikoanalyse utført for prosjektet Mafoberg Systems IT upgrade. En risiko er sannsynligheten for at en uønsket hendelse oppstår og konsekvensene den medfører seg. Risikoen blir beregnet på grunnlag av sannsynligheten for at hendelsen inntreffer og hvor stor konsekvensen av hendelsen er. Av dette vil få en prioritert oversikt over hvilke trusler som det må tas tak i. De trusler som har både høy sannsynlighet og høy konsekvens, må prioriteres først.

6.2 Uønskede hendelser

1. Fraværende gruppe-medlemmer
 - På grunn av sykdom eller andre faktorer, kan ett eller begge gruppe-medlemmer hindres fra i å møte til fastsatte tider.
 - **Konsekvens:** vi blir forhindret fra i å gjennomføre enkelte oppgaver til oppsatte tider.
 - **Sannsynlighet:** Lav.
 - **Tiltak:** Ingen konkrete tiltak, men ved lengre og/eller gjentakende forsinkelser, vil en god dialog med veileder vil i slike tilfeller være å satse på.
2. Tap av data
 - På grunn av feilende hardware eller feil sletting fra vår side, kan data og dokumenter gå tapt.
 - **Konsekvens:** Mye tid vil gå med på feilretting, eller vi må begynne en oppgave på nytt.
 - **Sannsynlighet:** Lav.
 - **Tiltak:** Vi bruker Dropbox-systemet for all lagring, inkludert nettside. Dette sørger for at vi har dobbeltlagring av data, med revisjonskontroll.
3. Dårlig kommunikasjon internt
 - Det kan oppstå uenigheter internt i gruppa om hvordan en deloppgave skal gjennomføres.
 - **Konsekvens:** Større tidsbruk på den aktuelle deloppgaven.
 - **Sannsynlighet:** Middels.
 - **Tiltak:** Vi må revidere hvordan den aktuelle deloppgaven skal gjennomføres.
4. Tap av nettilinje
 - Nettilinje ved skolen går ned, eller vi mister tilgangen til serverne hjemmefra.
 - **Konsekvens:** Ingen jobbing kan gjøres for den perioden linja er nede.
 - **Sannsynlighet:** Lav.
 - **Tiltak:** Ingen mot bortfall av linje mellom skole og hjem; det vil være enklere å fikse det interne nettverket mot serverne.

5. Påkrevd hardware for prosjektet, blir ikke gitt oss i tide

- Påkrevd hardware for å gjennomføre prosjektet, blir ikke gitt oss innen vi starter med installasjon og konfigurasjon.
- **Konsekvens:** Vi får ikke påbegynt prosjektet og dokumentasjon.
- **Sannsynlighet:** Middels.
- **Tiltak:** Oppdragsgiver må informeres oftere for å forstå alvoret i denne situasjonen. Alternativet blir å gå ned på omfanget i prosjektet.



Figur 3 - Risiko og sannsynlighet

7. Kost/nytte analyse

Dette er først og fremst et prosjekt som skal øke erfaring og kunnskap innen tilpasning av Windows Server 2008 R2, Exchange og styring av disse ved hjelp av Powershell og SCVMM. Dette gjelder altså den ikke-kvantifiserbare nytten. Derfor vil en beregning av den kvantifiserbare delen ikke være for reell. Den er med her for å vise omfanget av eventuelle lisenskostnader og maskinvare fra MafoBerg Systems AS sitt utgangspunkt.

Siden beregning av kostnader for drift av systemene ikke er veldig relevant til dette prosjektet og dets mål, vil vi ikke beregne selve bortfallet av kostnadene grunnet bedre administrering. Vi sitter ikke med noen form for økonomisk oversikt og vil derfor bare beregne dette anslagsvis ved prosent i kvantifiserbar nytte.

Vi beregner kostnadene for prosjektet for den ene avdelingen med administrasjon og driftspersonell. De utgjør en gruppe på 60 ansatte og maskinvareutstyr av servere og noen få oppgraderte klientmaskiner. Da beregner vi også besparinger og kostnader kun denne gruppen.

Ved utrulling for flere avdelinger vil det kunne bli lettere å utføre prosjektet basert på nåværende beregninger. Det vil blant annet kunne minske prosjektutgiftene og i tillegg gjøres for større grupper som vil utgjøre en enda større nytte.

7.1 Kvantifiserbar og ikke-kvantifiserbar nytte

Ut i fra effektmålene skal vi kunne effektivisere bruken av maskinvare uten å øke mengden av maskinvare bedriften allerede har. Dette vil redusere strømforbruk, arealforbruk og vedlikehold.

Kvantifiserbar nytte:

- Besparelse av maskinvareutstyr med 50 % (ca. kr 50000 per år).
- Redusert vedlikeholdsarbeid med 20 % grunnet enklere administrasjon av tjenere og backup av disse (ca. kr 750000 per år).

Bortfall av kostnader:

- Energiforbruk ned med 30 % (ca. kr 50000 per år).

Ikkekvantifiserbar nytte:

- Nye lettere måter å administrere et større system på.
- Lettere å øke antall tjenester og virtuelle maskiner.
- Nytt maskinvareutstyr som kan utvide levetiden for maskinparken.

7.2 Estimerte kostnader

Kostnader som har kun med prosjektet å gjøre beregner vi ut i fra tenkt lønning og kostnader i dag. Prisene på lisenser er funnet ut i fra gjennomsnittlig priser fra Microsofts sider på internett.

7.2.1 Kostnader for prosjektet

Timelønn per prosjektleder (kr)	1 000
Antall timer per person	450
Antall deltakere	2
Sum lønn (kr)	900 000

Pris lisens Windows 2008R2 Ent Single Open (kr)	65 876
Antall lisenser Windows 2008R2 Ent Single Open CAL15	2
Sum lisenser Windows 2008R2 Ent CAL15 (kr)	131 752
Pris brukerlisens Exchange 2010 (kr)	635
Antall brukerlisenser Exchange 2010	60
Sum lisenser Exchange 2010 (kr)	38 100
Pris lisens SCVMM (kr)	3 000
Antall lisenser SCVMM	1
Sum lisenser SCVMM (kr)	3 000
Sum lisenser per år	172 852
Totale utviklingskostnader	1 072 852

7.2.2 Nytteanalyse

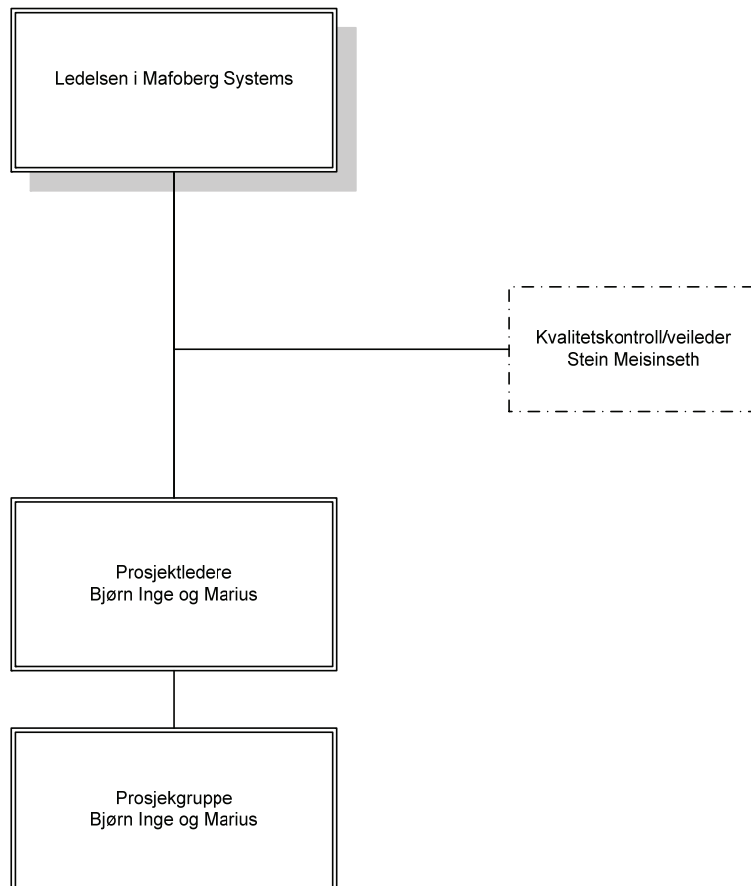
	År 1	År 2	År 3	År 4	Sum
Kvantifiserbar nytte	400 000	800 000	800 000	800 000	2 800 000
Bortfall kostnader	25 000	50 000	50 000	50 000	175 000
Sum nytte	425 000	850 000	850 000	850 000	2 975 000
Utviklingskostnader	1 072 852				1 072 852
Drift og forvaltningskostnader		172 852	172 852	172 852	518 556
Sum kostnader	1 072 852	172 852	172 852	172 852	1 591 408
Beregnet nytte (nytte-kostnad)	-647 852	677 148	677 148	677 148	1 383 592

Den totale nytten av dette prosjektet er beregnet til kr 1 383 592 i overskudd på en drift av avdelingen over 3 år. Nyten blir med sikkerhet større for videre utrulling av prosjektet for andre avdelinger.

8. Prosjektorganisering

Til dette prosjektet har vi blitt enige om en lik fordeling av ansvar blant de gruppedeltakerne. Skrivning av møteinnkallelser og referat vil gå på rotasjon for hvert møte. Det samme gjelder loggføring av timebruk, samt skrivning av rapporter, men her vil det nødvendigvis bli slik at én gruppedeltaker tar seg av en større del av rapporten enn den andre.

Oppdragsgiver for denne prosjektoppgaven er vår bedrift Mafoberg Systems.



Figur 4 - Prosjektorganisasjon

8.1 Utstyr som trengs

Vi skal i denne prosjektperioden virtualisere flere Exchange-servere, og vi skal se på Migrering av virtuelle maskiner mellom flere instanser av Hyper-V. Det blir derfor nødvendig med to fysiske maskiner.

Dette trengs av utstyr:

- To kraftige Servere med minimum 4GB minne hver
- Begge servere må støtte virtualiseringsteknologiene til enten AMD eller Intel.

Dette trengs av programvare og lisenser:

- Windows Server 2008r2
- Windows 7
- Hyper-V Server
- Scvmm
- Forefront Security
- Exchange Server 2010

IP-adresser:

- 2 til Hyper-V hosts
- 2 til Exchange-servere
- 5 til ulike testklienter

Det blir også nødvendig med tilgang til SAN-et, slik at vi kan migrere virtuelle maskiner ved hjelp av Scvmm.

9. Retningslinjer og standarder

9.1 Krav til dokumentasjon

Dokumentasjon skal utarbeides skriftlig og med opptak av oppsett vi gjør. Dokumentasjonen skal legges ut på et område hvor den blir tilgjengelig. <http://bachelor.bjorninge.net/>. Dokumentasjonen skal være til hjelp for å se hvor mye tid som er brukt på hva, og eventuelle feil som blir gjort kan rettes opp. Den skal gi oversikt over hvordan systemer er satt opp slik at man kan forandre om nødvendig etter behov hvis andre faktorer utenfra skulle virke negativt på systemet.

Dokumentasjonen skal til slutt samles på en CD som leveres sammen med all dokumentasjon i papirutgave. Dette skal til slutt leveres til veileder og avdelingskontor i 3. etg. I HiST lokaler på Kalvskinnet.

Type aktivitet	Distribusjonstype	Frist
Arbeidskontrakt	Papirutgave for godkjenning og doc-format på hjemmeside.	8.1.2010
Timelister inkludert rapport	xls-format på hjemmeside og link som sendes til Stein.	Slutten av hver uke. Klar til hvert møte.
Møteinnkallinger	Papirutgave for godkjenning og doc-format på hjemmeside og outlook kalender.	Hver fredag innen kl. 12:00
Møtereferater	Papirutgave for godkjenning og doc-format på hjemmeside og outlook kalender.	Innen samme dag som møte.
Forstudierapport	Papirutgave for godkjenning og doc-format på hjemmeside.	18.1.2010
Systemkravdokument	Papirutgave for godkjenning og doc-format på hjemmeside.	22.1.2010
Prosjektpresentasjon	Powerpointpresentasjon. Papirnotater for presentasjon.	5.5.2010
Prosjektdokumentasjon	Papirutgave i 2 eksemplarer og på hjemmeside.	25.5.2010
Sluttrapport	Papirutgave inkludert i prosjektdokumentasjon og på hjemmeside.	Ca 20.5.2010

Tabell 1 – Dokumentasjon av prosjektet

9.2 Krav til kvalitetsgjennomganger

For at resultatet skal bli bra må kvaliteten på arbeidet gjennomgås til bestemte tidspunkt. Dette vil som oftest foregå ved møter med veileder. Det vil også foregå ved interne små møter og diskusjoner for å se hva som er reelt i prosjektets utførelse. Ideer blir notert ned når de oppstår og vil bli gjennomgått for hver arbeidsdag. Endringer skal dokumenteres og testes.

Veileder kommer med kommentarer til utførelsen av prosjektet. Dette kan føre til kvalitetsmessige forandringer underveis.

- Dokumentasjonsmessig forhold
- Krav til omfang av styring
- Scriptenes kvalitet
- Organisasjonsmessig oppbygging på datalagre i forhold til bedrift

9.3 Krav til standarder og metoder

For å unngå problemer med kompatibilitet, bestemmer vi oss for standarder av programvare vi vil bruke under utførelsen av prosjektet når det gjelder skriving av dokumentasjon og script og lignende. Dette gjør at vi lett kan samarbeide med dokumentasjon og data.

- Dokumentasjon skal foregå med MS Office 2007 og tilhørende programvare utgitt av Microsoft i 2007-serien. Dette innbefatter Word, Excel, Powerpoint, Project, Outlook (webbasert) og enkle formatløse editorer for script. Eventuell Powershell-editor vil bli introdusert for bruk senere.
- Hjemmesider bruker vi enkel formatløse editorer for redigering.
- Kun Microsofts programvare som standarder.
- Utførelse av tester for script i utviklingsmiljøet underveis og testing at det holder i nytt oppsatt miljø.

10. Anbefaling om videre arbeid

Det er et godt grunnlag for å fortsette med videre arbeid av dette. Både fordi kostnadsbesparelser for bedriften vil være gode, men at dette først og fremst baserer seg på kunnskap om fremtidig prosjekt og drift av datasystemer.

Bedriften MafoBerg Systems AS har tydelig forstått at å utvikle dette prosjektet er til stor fordel siden firmaet bare har vokst og kostet på kostnader for tillagt utstyr uten å tenke stort på effektiviteten. Det har vært kundene og tjenestene for dem som har vært hovedfokuset.

Derfor vil vi starte med utviklingen av prosjektet.

Gruppe 6E

Systemkravdokument



Marius Foss, Bjørn Inge V. Berg
MafoBerg Systems AS

Innhold

1 Hensikten med dokumentet	3
2 Overordnet prosjektbeskrivelse	4
2.1 Kort overordnet beskrivelse av det nye systemet	4
2.2 Organisatoriske og personlige konsekvenser	4
3 Spesifikasjon av systemets funksjonelle egenskaper	6
4 Krav til systemkonstruksjon	7
4.1 Windows Server 2008 R2	7
4.2 Windows 7 – klient	7
4.3 Hyper-V Server eller VMware ESX.....	7
4.4 System Center Virtual Machine Manager (SCVMM).....	7
4.5 Microsoft Forefront Security for Exchange Server	7
4.6 Microsoft Exchange Server 2010.....	8
5 Krav til dokumentasjon	9
6 Reviderte resultater fra forstudiet	10

Figurer

Figur 1 – Nytt design av serveroppsett	4
--	---

1 Hensikten med dokumentet

Dette dokumentet skal gjøre klart hva krav som stilles for kjøring av planlagte tjenester og utvikling av selve prosjektet. Det skal gi et bilde av hvordan prosjektet ser ut og hvordan det ser ut etter at det er satt til liv.

1. Overordnet beskrivelse

En beskrivelse av hvordan systemet er når prosjektet er gjennomført og hva dette innebærer for brukere og driftere.

2. Spesifikasjon av systemets funksjonelle egenskaper

En kort beskrivelse av de største funksjonelle egenskapene av prosjektet, hvilke nye funksjoner som kan brukes.

3. Krav til systemkonstruksjon

Krav til maskinvare og software for at prosjektet skal nå sitt mål. Her forklares også hva den forskjellige programvaren skal benyttes til.

4. Krav til dokumentasjon

Dokumentasjon som skal følge med prosjektet når det leveres.

5. Revisjon av forstudierapporten

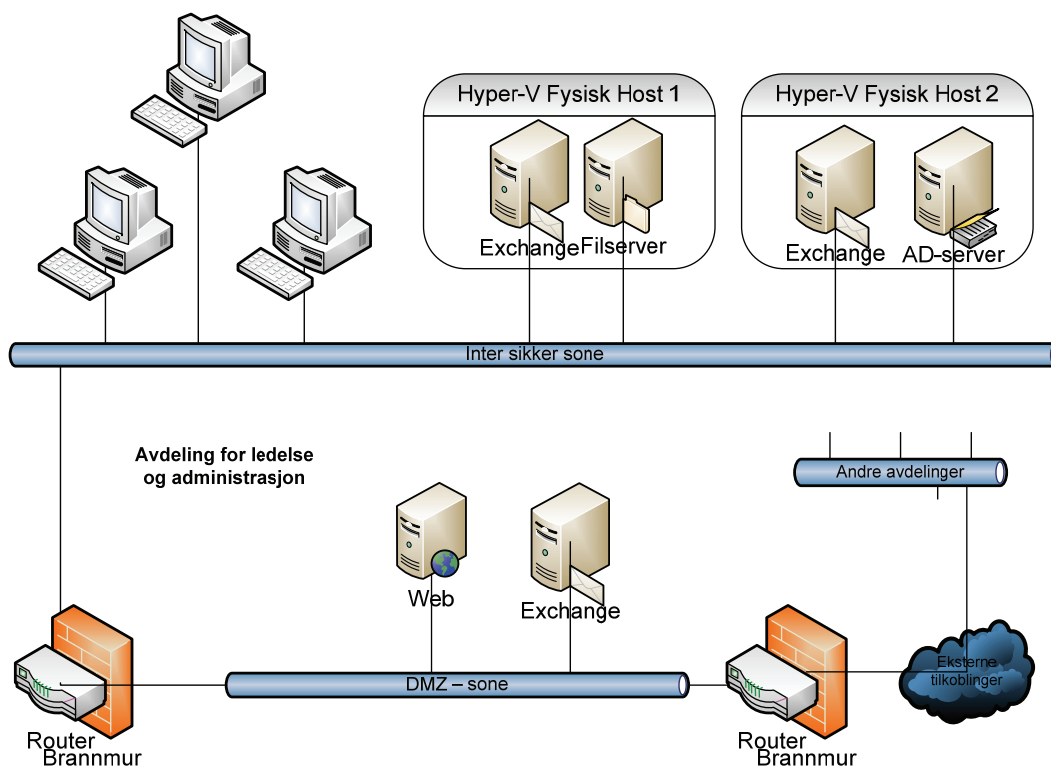
Forandringer som er blitt gjort i forstudierapporten.

2 Overordnet prosjektbeskrivelse

Dette prosjektet er en del av et større prosjekt for implementering av virtuelle servere for å redusere maskinvare, energiforbruk og plass (spesielt plass i fremtiden for bedriften). Prosjektet går ut på å bytte ut maskinvare for kraftigere maskinvare til kjøring av 64-bits maskinvare, og oppgradere software for de tjenestene som skal kjøres på disse. De gamle serverne består av Windows 2003 med domenekontrollere og Exchange 2003. De nye virtuelle serverne skal styres fra System Center Virtual Machine Manager (SCVMM) og med tilpassede script i Powershell. Prosjektet skal også gi mulighet for effektiv flytting av servere mellom Hyper-v serverne som er grunnlaget for eventuell energisparing; se del 3 for beskrivelse av disse softwareproduktene.

2.1 Kort overordnet beskrivelse av det nye systemet

Fra det gamle systemet prosjektet tar for seg skal det nye systemet ha de samme typene servere, men disse skal oppgraderes. I tillegg skal disse serverne flyttes inn på virtuelle servere. Dette betyr ny installasjon av disse serverne. Det nye systemet for den ene avdelingen vil kunne se ut som i figur 1. E-postserveren Exchange som står i DMZ-sonen (sone med forhøyede krav til sikkerheten) vil bli emulert som en egen virtuell maskin i fysisk host 1 under utføring av prosjektet. Denne vil stå som egen server senere når prosjektet skal ruller ut.



Figur 5 – Nytt design av serveroppsett

2.2 Organisatoriske og personlige konsekvenser

Det vil ikke ha noen spesiell innvirkning på vanlige brukere av systemet. Brukere vil kunne oppleve hurtigere tilgang, bedre filtrering av epost og liknende. Oppgraderte servere vil gi flere funksjonelle muligheter mot ny klientprogramvare. Dette vil bli testet, men ikke organisert for brukere.

IT-drifts avdeling vil merke de største forandringene, som blir av måten å drifte systemene på. Tjenesten System Center Virtual Machine Manager (SCVMM) vil bli en hovedkomponent for

drifting, og analysing av de virtuelle maskinene, og vil også sørge for at bedriften vil kunne spare betydelig med strøm ved lav last på maskinene. Powershell-script blir en del av styringskomponenter for disse serverne.

Det vil bli svært få personlige konsekvenser. Det vil for det meste bli fordeler. Konsekvenser som kan inntreffe er større sikkerhet som gjør at man får større passordkrav og andre tilkoblingskrav.

Reorganisering av enkelte mapper vil kunne medføre at man må endre enkelte vaner.

For organisasjonen vil det heller ikke bli noen økte konsekvenser, men heller fordeler. Eventuelt vil problemer med ny kunnskap føre til litt treghet i starten. Men dette vil føre til fordeler. Endrede rutiner bør følges nøye så vaner blir tilvent. Siden maskiner blir samlet på en fysisk maskin, vil fysisk sikring kunne få høyere konsekvens.

3 Spesifikasjon av systemets funksjonelle egenskaper

De funksjonelle egenskapene er slik systemet allerede opererer. Vi skal utvide allerede eksisterende funksjonalitet, slik at vi kan kontrollere de virtuelle Exchange-serverne og tilhørende Active Directory-informasjon fra en interaktiv Powershell-sesjon. Det skal være mulig å endre slike egenskaper som mailboxstørrelse til én eller flere brukere. Vi skal kunne kjøre alle brukere over fra én Exchange-server til en annen Exchange-server som kjører på forskjellige fysiske Hyper-V-servere, uten at det skal oppleves målbar nedetid.

Det tas også sikte på å utvikle et administrasjonsgrensesnitt for dette, i form av et enkelt og sikkert webinterface.

Administrasjonsfunksjonalitet som skal dekkes av Powershell:

- Opprette én bruker, eller flere brukere samtidig ved hjelp av brukermal.
- Opprette epostkonto til én eller flere brukere.
- Flytte epostkontoer mellom forskjellige Exchange-servere.
- Styre konfigurasjon av systeminnstillinger på virtuelle maskiner.
- Vise informasjon om systembelastning med tilpasset framvisning

4 Krav til systemkonstruksjon

Dette prosjektet skal virtualisere tidligere fysiske servere for administrasjonsavdelingen i Mafoberg Systems AS. Arbeidsstasjoner og de andre avdelingenes maskiner skal ikke inngå i prosjektet.

Eksisterende utstyr som skal benyttes:

- Alle switcher og routere.
- Web server.
- Eksisterende arbeidsstasjoner.

Maskinvareutstyr som skal kjøpes inn:

- 2x1u rackmonterte servere. Disse må støtte virtualiseringsteknologien til henholdsvis AMD eller Intel, og må ha minimum 8GB RAM hver. For dette firmaet anbefales det å ha 64-128GB RAM per fysiske server. Men i dette prosjektet holder vi oss til 8GB RAM for testmiljø.

4.1 Windows Server 2008 R2

Den nye versjonen av windows server sørger for redusert strømbruk og en del nye sikkerhetstiltak, i tillegg til en vesentlig forbedret virtualiseringsteknologi. Brukes både som gjesteoperativsystem og på de fysiske hostene som kjører Hyper-V. Krever 1. 4 GHz 64-bit prosessor. Minimum 1GB RAM, 32 GB lagringsplass.

4.2 Windows 7 – klient

Operativsystem for klienter. Støtter ipv6, og har bedre støtte for PC-er med flere prosessorkjerner. Har også redusert minneforbruk enn tidligere versjoner, og brukeropplevelsen blir derfor langt bedre. Krever 1GHz 32-bit prosessor. Minimum 1GB RAM, 16GB lagringsplass. Dette kan man lese mer om på Microsofts sine hjemmesider.

4.3 Hyper-V Server eller VMware ESX

Virtualiseringsprogramvare som lar oss kjøre gjesteoperativsystem. Det vil si at operativsystemene slipper å kjøre på egne fysiske maskiner, men at de kjører på en emulert maskin. Dette gjør at man får utnyttet maskinvaren mye bedre. 64-bit prosessor med Intel VT eller AMD-V virtualiseringsteknologi aktivert kreves. Minimum 1,4 GHz prosessor, 1 GB RAM og 8GB lagringsplass. Vi anbefaler minimum 2.8 GHz 64-bit prosessor med 8GB minne og 400 GB lagringsplass for de virtuelle maskinene.

4.4 System Center Virtual Machine Manager (SCVMM)

Programvare for å styre virtuelle gjesteoperativsystem. Man får oversikt over alle maskiner og ressurser innenfor det virtuelle miljøet. Den kjøres på Hyper-V-servern og har derfor samme krav som Hyper-V. SCVMM er designet for å håndtere flere fysiske maskiner som kjører Hyper-V Server, og vil la oss flytte virtuelle maskiner fra én fysisk server til en annen. Dermed kan man slå av fysiske servere når det ikke er behov for de ekstra ressursene. De tjenestene som kjøres vil fortsette å kjøre uten at brukerne opplever noen nedetid. SCVMM kan sammenliknes med vmware sin vSphere.

4.5 Microsoft Forefront Security for Exchange Server

En sikkerhetspakke fra Microsoft som inneholder beskyttelse mot virusangrep og inntrengning. Pakken som kalles Business Ready Security inneholder en rekk farer. Den tar både for seg identifisering, angrepsbeskyttelse og styring for både Exchange 2010, Sharepoint og flere av Microsofts server applikasjoner.

4.6 Microsoft Exchange Server 2010

Epostserver som støtter kontaktlister, distribusjonslister og kalender. Denne nye versjonen har større muligheter for replikering, og kan behandle større datamengder fordi den krever et 64-bit operativsystem. Vi velger å kjøre Exchange virtielt, og dermed krever den minimum 4GB minne på gjesteoperativsystem, 1.5 GB lagringsplass for installasjon og det bør være 2GB lagringsplass eller mer per epostkonto.

Følgende programvare skal benyttes:

5 Krav til dokumentasjon

Krav til dokumentasjon er **hovedrapporten**. Den inneholder

- Installasjon av software.
- Brukerveiledning av systemene.
- Videodokumentasjon, digitalt.

Denne skal leveres inn sammen med systemet som utvikles. Den skal beskrive installasjon og drift av systemet, samt brukerveiledning i separate deler. Brukerveiledning skal skrives slik at it-personal og administrasjon som ikke nødvendigvis har inngående teknisk innsikt skal kunne benytte seg av produktet.

Dokumentasjonen skal beskrive hvilke endringer som blir gjort, hvilke nye systemer som blir innført og hvordan implementeringen er gjennomført. Ved bruk av skjermopptak med lyd skal vi forklare installasjon og bruk av de aktuelle programmene steg for steg. I tillegg skal alle Powershell-script kommenteres tilstrekkelig slik at nødvendige endringer i ettertid skal være lett å gjennomføre.

6 Reviderte resultater fra forstudiet

Fra forstudierapporten har vi gjort klarere hva prosjektet faktisk dreier seg om. Det er behandlet i kapittel 2. Vi beskriver litt mer om hvorfor prosjektet blir utført i MafoBerg Systems AS. Dette har vært litt uklart.

Beskrivelse om hvem som har nytte av prosjektet og hvem det berører har blitt redigert. Dette har gjort at forstudiet blir mer spesifikt rettet mot de som skal drifte systemet, de som skal ha systemet og de som skal ta systemet i bruk.

Kostanalysen har blitt redigert med større rammebetingelse. Det vil blant annet ha innvirkning på enkelte økte kostnader ikke blir overraskende.

*Våren
2010*

MAFOBERG
SYSTEMS AS

SYSTEMDRIFT OG OPPGRADERING



Gruppe 6E – Bachelor 2010
Marius Foss, Bjørn Inge V. Berg

Innhold

1. Driftsdokumentasjon	16
1.1. Programvare som brukes.	16
1.2. Maskinvare som brukes.	16
1.3. Nettverk.....	16
1.4. Fremgangsmåter.	16
1.5. Script.....	17
2. Installasjon og konfigurasjon	18
2.1. Windows 2008 R2.....	18
2.1.1. Hva er nytt og forbedret i Windows Server 2008 R2?	18
2.1.2. Bruk av serverne	19
2.1.2. Installasjon og konfigurering av servere	20
2.1.2.1. Fysisk maskin 1 – Hyp-1	21
2.1.2.2. Fysisk Maskin 2 – Hyp-3	21
2.1.2.3. Fysisk maskin 3 – Hyp-2	22
2.2 Hva er virtualisering?	23
2.2.1 Fordeler med virtualisering.....	23
2.2.2 Hvordan Hyper-V fungerer.....	23
2.2.3 Installasjon av Hyper-V	25
2.2.4 Installasjon av virtuelle maskiner.....	29
2.2.4.1 Installasjon av nye operativsystem.....	29
2.3. Active Directory (AD), Domenekontroller	33
2.3.1. Active Directory Domain Services	34
2.3.2. Installasjon og konfigurering av AD DS	34
2.3.3 Gjennomføring, tillegg og igangsetting av AD DS	36
2.3.4. Installering av Read Only Domain Controller.....	47
2.3.4.1. Forhåndsoperasjoner før vi kommer i gang.....	47
2.3.4.2. Installering av RODC.....	47
2.3.4.3. DNSoppslag	51
2.4 Windows Failover Clustering og Hyper-v Failover	52
2.4.1 Hva er Windows Failover Clustering?	52
2.4.2 Forberedelse før installasjon	52
2.4.3 Clusterinstallasjon.....	54

2.4.4 Validering av Clusteret	57
2.4.5 Opprettelse av Clusteret	60
2.4.5.1 Aktivere delte volum	61
2.4.6 Flytte over maskiner til clusteret	63
2.5 System Center Virtual Machine Manager (SCVMM).....	67
2.5.1.1. VMM Server	69
2.5.1.2. VMM Administration Console.....	70
2.5.1 Analyse og validering av scvmm-server	71
2.5.1.1 Installasjon av IIS.....	74
2.5.1.2 Minnestørrelse.....	75
2.5.2 Installasjon av SCVMM-server	76
2.5.3 Installasjon av administratorkonsollet.....	80
2.5.4 Legge til ny host	82
2.6. Exchange 2010	86
2.6.1. Nytt med Exchange 2010	86
2.6.2. Installasjon	87
2.6.2.1. Installere .NET Framework 3.5 SP1.....	87
2.6.2.2. Installere Powershell v2.....	89
2.6.2.3. Velge Exchange språkmuligheter	90
2.6.2.4. Installere Exchange	91
2.6.2.5. Forberedelser for Exchange	95
2.6.2.6. Kritiske oppdateringer for Exchange	98
2.7. Installere Edge og Forefront Security for Exchange 2010.....	99
2.7.1. Hva er Edge?	99
2.7.2. Forberedelse og planlegging	99
2.7.3. Installasjon av Edge.....	100
2.7.4. Konfigurering av Edge for Exchange	103
2.7.4.1. Synkronisere Edge1 og Exchange1.....	104
2.7.5. Edge og filtere	106
2.7.5.1. Edge Transport Toolbox.....	106
2.7.6. Forefront Protection Systems.....	107
2.7.6.1. Forefront Protection installasjon	108
2.7.6.2. Oppstart og bruk av Forefront	111
2.8 Webgrensesnitt for Powershell.....	114

2.8.1 Prerequisites	114
2.8.2 Hvordan systemet fungerer	114
2.8.3 Installasjon	115
2.8.3.1 Oppsett av Auto oppstart	116
2.8.3.2 Oppsett av Selve grensesnittet.	116
2.8.3.3 Oppsett av DNS	116
2.8.4 Lett bruk av grensesnittet.	118
2.8.4.1 Epostutlisting	118
2.8.4.1 Serverstatistikk.....	119
3.1. Exchange 2010 - oppsett og bruk.....	120
3.1.1. Recipient Configuration	120
3.1.1.1. Oppretting av postkasse (mailboks) til eksisterende bruker i AD (GUI)	120
3.1.1.2. Opprette en postmasterkonto	123
3.1.1.3. Opprette en distribusjonsgruppe.....	125
3.1.1.4. Andre Recipient Configurations	127
3.1.2. Mailbokser.	128
3.1.2.1. Mailboks kvoter	129
3.1.3. Distribusjonsgrupper oppsett.	130
3.2. Bruk av SCVMM.....	131
3.2.1. Lage en ny maskin	133
3.2.1.1. Hardwareprofil – BIOS (Basic Input Output System)	134
3.2.1.2. Hardwareprofil – Processor	134
3.2.1.3. Hardwareprofil – Minne.....	135
3.2.1.4. Hardwareprofil – Floppydrive	135
3.2.1.5. Hardwareprofil – COM.....	136
3.2.1.6. Hardwareprofil – IDE Disk	136
3.2.1.7. Hardwareprofil – CD/DVD rom	137
3.2.1.8. Hardwareprofil – CD/DVD rom	138
3.2.1.9. Hardwareprofil – Priority	138
3.2.1.10. Hardwareprofil – Tilgjengelighet (Availability)	139
3.2.2. Å klonе en virtuell maskin.....	141
3.2.3. Lage template av en virtuell maskin	145
3.2.4. Konvertere en fysisk maskin til en virtuell maskin.....	147
3.2.5. Legge til en host fra et annet virtualiseringsmiljø	153

3.2.6. SCVMM Oppgavelinjen	155
3.2.7. SCVMM Library server	157
3.2.7.1. Oppsett av Library Server	157
3.2.7.2. Innhold i en Library Server	157
3.2.7.3. Ny maskin fra SCVMM Library Server	160
3.3. Bruk av Failover Clustering.....	163
3.3.1. Flytte maskiner mellom noder.....	164
3.3.1.1. Move Virtual Machine	164
3.3.1.2. Quick Migrate.....	165
3.3.1.3. Live Migration	166
3.3.1.4. Teste automatisk (unplanned) failovering	167
3.3.1.5. Detaljstyre failovering.....	168
4. Script i Powershell	169
4.1 Installasjon av Powershell	169
4.2 Komme i gang med Powershell.....	169
4.2.1 Execution Policy	169
4.2.2 Powershell-profiler	171
4.2.2.1 Opprette Powershell-profiler.....	171
4.3 syntaks og sammenlining	174
4.3.1 Get-childItem – en kommando med mange navn	174
4.3.2 Funksjoner, kontrollstrukturer og operatører	177
4.3.2.1 Switch som alternativ kontrollstruktur	179
4.3.2.2 Arrays og hashtables	180
4.3.2.3 Løkker.....	181
4.3.2.4 Pipeline.....	181
4.3.2.5 Where-Object Filtre	183
4.3.2.6 Sort-object	184
4.3.2.6 Begrense resultatmengde med Select-object.....	184
4.3.2.7 Foreach I en pipe.....	184
4.4 Administrering og fjernstyring med Powershell og remoting.....	186
4.4.1 Aktivere Powershell remoting	186
4.4.1.1 Aktivere Powershell remoting med GPO	187
4.4.2 Forskjellige typer remoting	189

4.4.2.1 Klassisk remoting og Windows Management Infrastructure (VMI)	189
4.4.2.2 Powershell remoting	189
4.4.2.3 Bruk av lokale .ps1-filer i eksterne sesjoner	191
4.4.2.4 Implisitt remoting	192
4.4.3 Generell administrasjon av servere	193
4.4.3.1 Installasjon av oppdateringer	193
4.4.3.2 Belastning av servere	195
4.4.3.3 Sammenkjøring av all lastinformasjon.....	195
4.4.3.4 Behandling av prosesser	197
4.4.3.5 Diagnostikk ved hjelp av Eventlog og Get-WindowsFeature	198
4.5 Powershelladministrasjon av AD	199
4.5.1 Uthenting av brukere og OUer.....	200
4.5.2 Utlisting og redigering av brukerinformasjon.....	202
4.5.3 Opprette OUer og grupper	204
4.5.4 Innlegging av brukere	206
4.5.4.1 ConvertTo-SecureString	206
4.5.5 Automatisere utlesing og innlegging av flere brukere.....	208
4.5.5.1 Innlegging av brukerne	208
4.5.6 Sjeldent innloggede brukere	210
4.6 Powershelladministrasjon av Exchange	211
4.6.1. Mailboks og brukerkommandoer	212
4.6.2.1. Mailbox brukere	213
4.6.2.2. Mailboks kvoter	214
4.6.2. Distribusjonsgrupper.....	216
4.6.2.1. Basert på Gruppeobjekt – Nytt opprettet	216
4.6.2.2. Basert på gruppeobjekter	216
4.7 PowerShelladministrasjon av SCVMM	218
4.7.1. Listing av virtuelle maskiner og informasjon	219
4.7.2. Listing av maskiner med ip-adresser fra DNS	220
4.7.2. Skifte maskinnavn på en maskin.....	221
4.7.3. Lage en ny virtuell maskin.....	221
4.7.4. Lage maskin ut ifra template.	222
4.7.4.1. Template forhåndsoppsett.	223
4.7.4.2. Enkelt script for en maskin.....	223

4.7.4.3. Opprette mange maskiner	224
4.7.4.4. Opprette maskiner i kø.	226
4.7.5. Avbryte en jobb og slette en maskin	226
4.7.5.1. Slette maskiner	227
4.7.6. Andre kommandoer	228
4.8 Nyttige Scripts i forbindelse med rapportskrivning	230
4.8.1 Sammenfletting av Word-dokumenter	230

Tabeller

Tabell 1 - Operativsystemkrav	18
Tabell 2 - Sammenlikning Windows 2008 R2 server-typer	20
Tabell 3 - VMM Serverkrav	69
Tabell 4 - VMM Serverkrav	69
Tabell 5 - VMM Konsollkrav	70
Tabell 6 - VMM Konsollkrav	70
Tabell 7 - Forefront maskinkrav	108
Tabell 8 - Oversikt over programkrav	114
Tabell 9 - Miljøvariabler	172
Tabell 10 - Mye brukte kommandoer og alias	175
Tabell 11 - Sammenlikning av operatører	178

Figurer

Figur 1 Serversoftwareoppsett før og etter Hyper-V-installasjon	23
Figur 2 - Rolleinstallasjon med Server Manager	25
Figur 4 - Hyper-V-rollen installeres	26
Figur 3 - Ting som må leses før installasjon	26
Figur 5 - Valg av ett eller flere nettverkskort som skal brukes av Hyper-V sine virtuelle maskiner. Her fravelger vi det adapteret som går mot SANet	27
Figur 6 - Installasjon utføres med disse innstillingene	27
Figur 7 - Rollen Hyper-V krever omstart for å legge inn hypervisor	28
Figur 8 - Installasjon av Hyper-V ferdiggjort	28
Figur 9 - Oversiktsbilde i Hyper-V Manager	29
Figur 10 - Opprettelse av en ny virtuell maskin	29
Figur 11 - Navngivelse av virtuell maskin	30
Figur 12 - Valg av virtuell nettverksswitch	30
Figur 13 - Valg av virtuell lagringsmedium for den nye virtuelle maskinen	31
Figur 14 - Flere valgmuligheter for installasjon av OS	32
Figur 15 - Hyper-V Manager med virtuelle maskiner aktivert	32
Figur 16 - AD DS Rollevalg	34
Figur 17 - .NET Framework 3.5.1 tilleg påkrevd	34

Figur 18 - AD DS Installasjonsnotater	35
Figur 19 – Resultater	36
Figur 20 - AD DS - DCPROMO.EXE	36
Figur 21 - Advanced Installation	37
Figur 22 - NT 4.0 støtte	38
Figur 23 - Nytt domene i ny skog	39
Figur 24 - Nytt domenenavn	39
Figur 25 - Skogsfunksjonalitet	40
Figur 26 – Domenefunksjonalitetsnivå	41
Figur 27 - DNS-tillegg	42
Figur 28 - DNS fortsettelse	42
Figur 29 - Database og script plasseringer	43
Figur 30 - DS Restore Mode Administrator Password	44
Figur 32 - Eksportere installasjonsinformasjon	45
Figur 31 - Eksporteringsveiviser og restart	45
Figur 33 - DNS-oppsett i nettverkskort	46
Figur 34 - Installasjon RODC – 1	47
Figur 35 - Installasjon RODC – 2	47
Figur 36 - Installasjon RODC - 3	48
Figur 37 - Installasjon RODC eksisterende domene	48
Figur 38 - Installasjon RODC - Skog og Site-name	49
Figur 39 - Installasjon RODC - valg for RODC	49
Figur 40 - Installasjon RODC - Annet	50
Figur 41 - RODC i funksjon. Ingen opprettelsestilgang.	50
Figur 42 - DNS oppslag utvidelse	51
Figur 43 - DNS oppslag laget	51
Figur 44 - Oversikt over nettverksadaptere. Failover Clustering krever minimum ett dedikert adapter til lagringsmediumet	52
Figur 45 - SAN-tilkoling for hyp-1.mafoberg.net (venstre) og hyp-2.mafoberg.net (høyre)	53
Figur 46 - Installasjon av Failover Cluster-rollen	54
Figur 47 - Konfirmering av installasjon. Dette må gjennomføres på hver server(node) i clusteret	55
Figur 48 - Oppsett av delt lagringsmedium	55
Figur 49 - Koble til lagringsmediumets logiske disk	56
Figur 50 - Koble til volumer på nettverksdisk	56
Figur 51 - Oppstart av cluster-validering	57
Figur 52 - Innlegging av clusternoder	57
Figur 53 - Ved hjelp av domenekontroller, finner validatoren alle Hyper-V-maskiner i nettet	58
Figur 54 - Kjøring av clustertester	58
Figur 55 - Valideringsrapport	59
Figur 56 - Clusteret vil få sin egen administrasjonsadresse	60
Figur 57 - Clusteret er klart til å opprettes	60
Figur 58 - Cluster opprettet uten feil	61
Figur 59 - Aktivere delte volum	61
Figur 60 - Legge til nettverksdisker i clusteret	62
Figur 61 – Visning av Clusterlagringsmappe. Denne mappa vil være synlig på alle noder i clusteret	62

Figur 63 - Informasjon om high availability i clustersammenheng.....	63
Figur 62 - Flytte maskiner over i clusteret	63
Figur 64 - Valg av virtuell maskin	64
Figur 65 - Kjørende virtuelle maskiner kan ikke flyttes.....	64
Figur 66 - Flytte avslått maskin	65
Figur 67 - Flytteprosessen starter	65
Figur 68 - Flytteprosessen er utført	66
Figur 69 - VMMmedium for installasjon	67
Figur 70 - SCVMM installasjonsvalg	68
Figur 71 - VMM Server Analyse og valideringssoftware for nedlasting.....	71
Figur 72 - VMM Server Baseline Configuration Configuration	72
Figur 73 - VMM Server Configuration Analyzer	72
Figur 74 - VMM Server Configuration Analyzer plassering.....	72
Figur 75 - VMM Server Configuration Analyzer kjøring	73
Figur 76 - Domene og dnskrav	73
Figur 77 - SCVMM Server Configuration Analyzer rapport.....	74
Figur 78 - SCVMM Server Installasjon av rollen IIS	74
Figur 79 - SCVMM Server installasjon av IIS.....	75
Figur 80 - SCVMM Server installasjon av IIS ferdig	75
Figur 81 - SCVMM Server Configuration Analyzer andre gang	76
Figur 82 - SCVMM Server installasjon	76
Figur 83 - SCVMM Server installasjon	76
Figur 85 - SCVMM Server sjekk og prosess	77
Figur 84 - SCVMM Server registrering	77
Figur 86 - SCVMM Server plassering av programvare	78
Figur 87 - SCVMM Server Libraryfunksjonens oppsett	78
Figur 88 - SCVMM Server oppsummering.....	79
Figur 89 - SCVMM Server installasjon fullført.....	79
Figur 90 - SCVMM Konsoll installasjon.....	80
Figur 91 - SCVMM Konsoll installasjon plassering	80
Figur 92 - VMM Konsoll installering portnummer	81
Figur 93 - Oppstart og tilkobling	81
Figur 94 - SCVMM Konsoll Add hosts.....	82
Figur 95 - SCVMM - Legge til host.....	83
Figur 96 - SCVMM - Legge til cluster	83
Figur 97 - SCVMM - vårt cluster	84
Figur 98 - SCVMM hostgruppe	84
Figur 99 - SCVMM hostegenskaper.....	84
Figur 100 - SCVMM importert virtuelle maskiner.....	85
Figur 101 - SCVMM - Clusterlagring av virtuelle maskiner	85
Figur 102 - Startskjerm.....	87
Figur 104 - Installasjon Feature .NET Framework 3.5.1	88
Figur 103 - .NET Framework error	87
Figur 105 - Install IIS 7.5 standard - 1.....	89
Figur 106 - Install IIS 7.5 standard - 2.....	89

Figur 107 - Install IIS 7.5 standard - 3.....	89
Figur 109 - Språkpakkenedlasting	90
Figur 110 - installering og klargjøring	90
Figur 108 - Språkvalg	90
Figur 111 – Introduksjon og lisens	91
Figur 112 - Feilmeldingsrapportering.....	91
Figur 113 - Installasjonstype	92
Figur 114 - Rollevalg.....	92
Figur 115 - Exchange organisasjonsnavn	93
Figur 116 - Klientoppsett for Outlook 2003	94
Figur 117 - Klient-tilgkobling	94
Figur 118 - Microsofts informasjonsinnsamling.....	95
Figur 119 - Readiness check	95
Figur 120 - Servermanagercmd -i RSAT-ADDS	96
Figur 121 - ServerManagerCmd -i Web-Metabase	97
Figur 122 - ServerManagerCmd -i Web-Lgcy-Mgmt-Console	97
Figur 123 - Exchange installasjon og ferdiggjøring	97
Figur 125 - Edge installasjonsstart	100
Figur 124 - Edge DNS Suffix.....	100
Figur 126 - Edge Custom valg	101
Figur 127 - Edge installasjon valg.....	101
Figur 128 - Edge installasjonsklart og oppsummering	102
Figur 129 - Exchange Edge etterkonfigurering.....	102
Figur 130 - Edge konfigurasjon, Exchange Management Console.....	103
Figur 131 - Edge DNSoppsett 1	103
Figur 132 - Edge1 subscription eksportering fra Edge1	104
Figur 133 - Edge - Subscription tillegg og synkronisering	104
Figur 134 - Edge - Test av synkronisering.....	105
Figur 135 - Edge - Starte synkroniseringen	106
Figur 136 - Forefront installasjon oppstart 1	108
Figur 137 - Forefront nedlasting	109
Figur 138 - Lisens og services	109
Figur 139 - Program og dataplassering	109
Figur 140 - Proxyinstillinger	110
Figur 141 - Spamaktivering	110
Figur 142 - Erfaringsmedelelse til Microsoft.....	110
Figur 143 - Oppsummering av installasjon	110
Figur 144 - Forefront Protection fullførelse.....	111
Figur 145 - Aktivering	111
Figur 146 - Forefront Dashboard - Monitoring	111
Figur 147 - Forefront Dashboard – Tasks.....	112
Figur 148 - Forefront Dashboard – Policymanagement.....	113
Figur 149 - Antispam filter aktivering (enable)	113
Figur 150 - Installasjon av XAMPP.....	115
Figur 151 - Oppsett av XAMPP	115

Figur 152 - XAMPP kontrollpanel.....	116
Figur 153 - Finne IP-adresse til Webserver	116
Figur 154 - Legge til en A-record i DNS	117
Figur 155 - Åpningsside for webgrensesnitt	118
Figur 156 – Epostutlisting	118
Figur 157 - Utlisting av serverstatistikk.....	119
Figur 158 - Ny postkasse	120
Figur 159 - Ny epostbruker - Usertype.....	120
Figur 160 - Brukerinformasjon til AD	121
Figur 161 - Mailbox plassering og policyer	121
Figur 162 - Mailboks Archive settings	121
Figur 163 - Mailboks og brukerresyme	122
Figur 164 - Mailboks for ny bruker, PowerShell.....	122
Figur 165 - Mailbokser opprettet.....	122
Figur 166 - Exchange postmasterkonto administrator	123
Figur 167 - Exchange postmaster epostadresser.....	124
Figur 168 - opprette postmasteradresse	124
Figur 170 - Postmastermailboks verifisert	125
Figur 169 - Lagt til epostadresse til brukeren Administrator.....	124
Figur 171 - Distribusjonsgrupper	125
Figur 172 - Bruk eksisterende gruppe	126
Figur 173 - Ferdigfylt groupdata	126
Figur 174 - Ny distribusjonsgruppe	127
Figur 175 - Recipient Configuration	127
Figur 176 - Kontakt eller brukere.....	127
Figur 177 – Mailboksegenskaper	128
Figur 178 - Mailboks oppsett	129
Figur 179 - Storage Quotas	129
Figur 180 - Distribusjonsgruppeegenskaper	130
Figur 181 - SCVMM hovedvindu	131
Figur 182 – SCVMM Host Groups.....	133
Figur 183 – SCVMM New Virtual Machine.....	133
Figur 184 - SCVMM Lage helt ny maskin.....	133
Figur 185 - SCVMM Ny maskin informasjon	133
Figur 186 - SCVMM Ny maskin - Hardware configuration	134
Figur 187 - HWprofile – Prosessor	134
Figur 188 - HWprofile – Minne	135
Figur 189 - HWprofile – Floppy	135
Figur 190 - HWprofile – COM.....	136
Figur 191 - HWprofile – IDE Disk.....	136
Figur 192 - HWprofile – IDE CDdrive	137
Figur 193 - HWprofile – Network Adapter	138
Figur 194 - HWprofile – Kjøreegenskaper.....	138
Figur 195 - HWprofile – Highly availability.....	139
Figur 196 - SCVMM maskinplassering.....	139

Figur 197 - SCVMM hostvalg	139
Figur 198 - SCVMM Plassering på host	139
Figur 199 - SCVMM Nettverksadapter	140
Figur 200 - SCVMM ekstra egenskaper	140
Figur 201 - SCVMM Ny maskin oppsummering	140
Figur 202 - Clone maskin for template	141
Figur 203 - Eier av virtuell maskin	142
Figur 204 - Hardware konfigurasjon	142
Figur 205 - Plassering av klonen.....	143
Figur 206 - Velge host for kloner og sted for plassering	143
Figur 207 - Status for klonen.....	144
Figur 208 - Kloneoppsummering.....	144
Figur 209 - Valg av maskin for template	145
Figur 210 - Template navn og hardware.....	145
Figur 211 - Template systemprofiloppsett.....	146
Figur 212 - Server og maskinpath	146
Figur 213 - Template summary	147
Figur 214 - Konvertering av fysisk maskin (1)	147
Figur 215 - Konverteringsveiviser (1)	148
Figur 216 - Konverteringsveiviser (2) - destinasjon	148
Figur 217 - Konverteringsveiviser (3) – informasjonssjekk	149
Figur 218 - Konverteringsveiviser (4) – HW-oppsett.....	149
Figur 219 - Konverteringsveiviser (5) – Hostplassering.....	150
Figur 220 - Konverteringsveiviser (6) - Highly available.....	150
Figur 221 - Konverteringsveiviser (7) – clusterpath.....	150
Figur 222 - Konverteringsveiviser (8) – NICs	151
Figur 223 - Konverteringsveiviser (9) – Tilleggsoppsett.....	151
Figur 224 - Konverteringsveiviser (10) – Sjekk av hinder	151
Figur 225 - Konverteringsveiviser (11) – Oppsummering	152
Figur 226 - Konverteringsveiviser (12) – Konvertering	152
Figur 227 - Konverteringsveiviser (13) – ferdig.....	153
Figur 229 - SCVMM ESX-tilkobling	154
Figur 228 - VMware kobling	153
Figur 230 - SCVMM VCenter Host.....	154
Figur 231 - SCVMM ESXhost	154
Figur 232 - SCVMM ESX maskin maskinprofil	154
Figur 235 - Oppgavelinja (Taskpane).....	155
Figur 233 - SCVMM ESXhost meldinger	155
Figur 234 - SCVMM ESXmaskin	155
Figur 236 - SCVMM PRO Tips	156
Figur 237 - SCVMM Networking Configuration	156
Figur 238 - SCVMM Library server	157
Figur 240 - SCVMM Library - Hardware profil.....	158
Figur 239 - SCVMM Library - ISOs og filters	158
Figur 241 - Gjøre bruk av hardwareprofil	159

Figur 242 - New Guest OS Profile.....	159
Figur 243 - New Guest OS Profile - Product key	160
Figur 245 - Ny Guest OS Profile.....	160
Figur 244 - New Guest OS Profile domene	160
Figur 246 - Ny maskin ut fra template	161
Figur 247 - Ny server fra SCVMM Library – hardwareprofil	161
Figur 248 - Ny server fra SCVMM Library - Guest OS Profile	161
Figur 249 - Ny server fra SCVMM – hostplassering	161
Figur 250 - Ny server fra SCVMM – hostvalg	161
Figur 251 - Administrere et cluster	163
Figur 252 - Standardflytting av virtuelle maskiner.....	164
Figur 253 - En Quick migrate gjennomføres	165
Figur 254 En Live migration utføres.....	166
Figur 255 - Simulering av automatisk failovering	167
Figur 256 - Failoveregenskaper for VMen DCserver1	168
Figur 257 - Failoveregenskaper 2 for DCserver1.....	168
Figur 258 - Utlisting av Powershell-versjon	169
Figur 259 - Feilmelding ved eksekvering av script	170
Figur 260 - Endre Execution Policy.....	170
Figur 261 - Standard profilsti	171
Figur 262 - Profilsti kan forandres av andre program.....	171
Figur 263 - Eksempel på -whatif.....	172
Figur 264 - Profilsript autokjørt ved oppstart	172
Figur 265 - Innlegging av innloggingsinformasjon i profilen.....	173
Figur 266 - Enkel utskrift med Powershell	174
Figur 267 - fargelegging av uttekst	174
Figur 268 - Eksempel på Get-Help.....	174
Figur 269 - Get-ChildItem kan skrives på hele tre forskjellige måter	175
Figur 271 - Gangeoperasjon på tall kan gi tilsynelatende feil resultat	176
Figur 272 - Tvinge fram datatype.....	176
Figur 270 - Opprettelse og utlisting av mapper	176
Figur 273 - Bruk av if-elseif-else.....	177
Figur 274 - Switch når en skal gjøre potensielt mange sjekker mot samme variabel	179
Figur 275 - Enkel bruk av arrays.....	180
Figur 276 - Bruke løkker til å gjøre gjentakende handlinger.....	181
Figur 277 - for-løkke for å liste ut enkelte element	181
Figur 278 - foreach-løkke går igjennom alle elementer.....	181
Figur 279 - Linux-skall: Utlisting av mapper som inneholder ordet "dev"	182
Figur 280 - Powershell lister ut alle filer som objekter istedet for tekst	182
Figur 281 - Filtrering med where	183
Figur 282 - Utlisting av frister i denne måneden og seinere.....	183
Figur 283 - Sortering basert på høyest minnebruk.....	184
Figur 284 - Select kan velge ut enkeltobjekter	184
Figur 285 - Utvelging av prosessnavn	184
Figur 286 - Alternativ bruk av foreach-løkke	185

Figur 287 - Aktivering av en remote-listener	186
Figur 288 - Aktivering av Powershell Remoting	187
Figur 290 - Aktivere Listeners	187
Figur 289 - Ny GPO for servere	187
Figur 291 - Tillate innkommende forbindelser	188
Figur 292 - Koble til eksterne maskiner med "-computername"	189
Figur 293 - Remoting vha. Enter-PSSession	190
Figur 294 - Invoke-command (icm) kan koble til flere maskiner	190
Figur 295 Invoke-command (icm) vha. lagrede sesjoner	190
Figur 296 - Utlisting av IP-adresser for hver server	190
Figur 297 – Annotert eksempel på bruk av lokale filer i eksterne sesjoner.....	191
Figur 298 - Implisitt installasjon av Servereprogramvare	192
Figur 299 - Utlisting av oppdateringer	193
Figur 300 - Installasjon av oppdateringer begynner	193
Figur 301 - Installasjon av oppdateringer fullført	194
Figur 302 - Utlisting av brukt plass.....	195
Figur 303 - Utlisting av prosessorlast	195
Figur 304 - Utlisting av ledig og tilgjengelig ram	195
Figur 305 - Kode for lastberegning på flere servere	196
Figur 306 - Utdrag fra lastberegningsutskrift.....	196
Figur 307 - Utlisting av prosesser sorter på minnebruk.....	197
Figur 308 - utlisting av scriptet list-non-responding-apps.ps1	197
Figur 309 - Avslutning av ikke-responerende programmer.....	197
Figur 310 - Installasjon av Roller eller features.....	198
Figur 311 - Liste ut installerte roller of features	198
Figur 312 - Finne tilgjengelige logger	198
Figur 313 - Advarselslogger fra SCVMM	198
Figur 314 - Import og utforsking av kommandoene i AD-modulen	199
Figur 315 - Bruk av wildcard-filtre.....	200
Figur 316 – To AD-søk basert på fornavn.....	200
Figur 317 - Begrensing av søkebase	201
Figur 318 - Brukere i en forhåndslagret OU	201
Figur 319 - Lagre brukerkontoer	202
Figur 320 - Utlisting av egennavn	202
Figur 321 - AD-utlisting	202
Figur 322 - Setting av brukerinformasjon	203
Figur 323 - Setting av mer brukerinformasjon.....	203
Figur 324 - Utlisting av personinformasjon i AD	203
Figur 325 - OU-en teknikere.....	204
Figur 327 - Brukere flyttet.....	205
Figur 326 - Opprettet OU i AD.....	205
Figur 328 - Innlegging av enkeltbruker	206
Figur 329 - AD-utlisting av innlagt bruker	206
Figur 330 - Innlegging av bruker vha. forhåndslagret passord	207
Figur 331 - Brukerlistestruktur i Excel, fra eksempelfila "brukere.xls"	208

Figur 332 - Bruk av excel-get-user-infos.ps1.....	208
Figur 333 - Innlegging av brukere fra Excel-arket	209
Figur 334 - Resultat av masseinnlegging.....	209
Figur 335 - Resultat av siste kommando.....	210
Figur 336 - Exchange Management Shell oppstart.....	211
Figur 337 - Exchange Management Shell PSoppstart	211
Figur 338 - Exchange 2010 PSSnapins.....	212
Figur 339 – Get-Mailbox	212
Figur 340 - Get-MailUser.....	212
Figur 341 - Get-User	213
Figur 342 - Mailboks for en bruker	213
Figur 343 - Mailbokser for alle brukere	214
Figur 344 – Mailboksinformasjon	214
Figur 345 - Mailboks kvotelisting	215
Figur 346 - Set-Mailbox.....	215
Figur 347 - Oppsett av kvoter i konsoll	215
Figur 348 - Ny distribusjonsgruppe	216
Figur 349 - Dynamisk distribusjonsgruppe	217
Figur 350 - SCVMM PowerShell oppstart med snapin.....	218
Figur 351 - SCVMM Server tilkoblet.....	218
Figur 352 - SCVMM Powershell kommandoer.....	219
Figur 353 - PS - Get-VM navn og info.....	219
Figur 354 - SCVMM PowerShell adresselisting	220
Figur 355 - SCVMM Hostadresseliste.....	220
Figur 356 - scvmm get-ip-from-host	220
Figur 357 – IP-adresse listing	221
Figur 358 - Ny virtuell maskin	222
Figur 359 - Template OS-konfigurasjon	223
Figur 360 - PowerShell ny maskin fra template	224
Figur 361 - \$TemplateJobGroup	225
Figur 362 - SCVMM New-vm-from-template.....	225
Figur 363 - SCVMM jobblisting.....	226
Figur 364 - Status på maskiner.....	227
Figur 365 - SCVMM Funksjon RemVM-by-status.....	228
Figur 366 - Eksempel på bruk.....	230
Figur 367 - Scriptet samler inn filer.....	230

1. Driftsdokumentasjon

Denne driftsdokumentasjonen er rettet mot konsulenter og IT-personell som skal drifte nye datasystemer basert på Microsoft programvare for bedriften Mafoberg Systems AS. Dokumentasjonen er en veiledning for installasjon, oppsett konfigurering av tjenester og bruk av tjenestene. Hovedsakelig er dokumentasjonen beregnet på bedriften Mafoberg Systems AS, men innholdet er skrevet for at også andre liknende bedrifter eller prosjekter skal kunne benytte dokumentasjonen for sine forhold.

Før påbegynt driftsdokumentasjon er det utarbeidet både en forstudierapport og et systemkravdokument.

Forstudierapporten inneholder en analyse av systemene for bedriften Mafoberg Systems AS før prosjektets start. Det gir grunnlag for hvilke systemer som skal fjernes, oppgraderes eller nyinstalleres. Det kommer også frem hva som kan gjøres for bedriften for å bedre situasjoner.

Systemkravdokumentet inneholder maskinvare som er minimum for prosjektet og hva bedriften bør basere seg på når de skal ta systemene i bruk. Minimumskrav vil kunne gi begrensninger på hva som kan utføres og utvikles.

1.1. Programvare som brukes.

Microsoft programvare krever lisens for bruk. Noen tidligere utgivelser av programvare kan kreve lisens for i det hele tatt å få lov å installeres. Programvaren vi bruker her har evalueringsperiode på opptil 120 dager.

- Microsoft Windows Server 2008 R2 Datacenter Edition 64-bit
- Microsoft Windows Server 2008 Datacenter Edition 32-bit
- Microsoft Exchange 2010.
- Microsoft System Center Virtual Machine Manager 2010 R2.

1.2. Maskinvare som brukes.

I dette prosjektet blir vi tildelt maskinvare Høgskolen I Sør-Trøndelag har tilgjengelig. Maskinvaren er testutstyr som kan være litt forskjellig i forhold til hva som er blitt satt som krav i systemkravdokumentet. Det gjør at enkelte oppsett eller utføring av handlinger kan gi forskjellig utslag i forhold til hva andre vil få med annen type maskinvare.

1.3. Nettverk.

Utviklingsstedet for dette prosjektet skjer på Høgskolen i Sør-Trøndelag. Her har vi tilgang til et adresseområde for IPv4. Adresseområde er 158.38.56.244 til 158.38.56.251. Eventuelle midlertidige adresser kan variere rundt dette området.

Andre adresser kan benyttes for et liknende prosjekt, men vi tar her forbehold om eventuelle forandringer i konfigurasjon som må til for andre nettverk.

1.4. Fremgangsmåter.

Vi bruker mye standard fremgangsmåter under installasjoner av programvare og konfigurering av disse. Vi vil trekke frem mulige andre måter å gå frem på så vi får mulighet til å gi rom for alternative løsninger.

Problemer og feilmeldinger vi kommer over underveis vil vi avdekke og forklare. Når vi skal forklare disse, kan det ofte hende vi beskriver disse før de oppstår i vår installasjon. Da legger vi disse til i

krav på forhånd til installasjon (prerequisites). Enkelte tilfeller vil vi også vise til som ofte kan oppstå, så det blir enklere å kjenne seg igjen under veiledningen.

Det er også en del ulemper og problemer vi kanskje ikke oppdager i denne driftsdokumentasjonen, men som andre som vil gå frem på samme måte kanskje oppdager. Dette kan oppstå på grunn av små forskjeller som kan være vanskelig å oppdage. Oppsett av roller, tillegg og bruk av nettverket kan være årsaker til slike problemer.

1.5. Script.

Mye av dette prosjektet behandler bruk av PowerShell. Vi tar da for oss bruk av PowerShell versjon 2 som er inkludert i Windows Server 2008 R2. Vi vil da forklare de grunnleggende deler av PowerShell på i tillegg til eksempler.

I tillegg til PowerShell blir det lagt til snapins og moduler for styring av programvaren til Microsoft. Vi vil forklare en del bruk av disse for å kunne styre programvaren på en alternativ måte til det grafiske brukergrensesnittet.

Alle script som blir utviklet for dette prosjektet vil bli lagt som vedlegg til dokumentasjonen.

2. Installasjon og konfigurasjon

2.1. Windows 2008 R2

Windows 2008R2 serversoftware er kun utgitt i 64-bits versjon. Det vil si at maskinvaren dette operativsystemet kjører på må fullt ut støtte 64-bits operasjonsmodus. Minimum maskinvarekrav er satt opp i tabell 1 når det gjelder maskinvare.

Komponenter	Krav
Prosesor	1.4GHz X64 CPU
Minne	Min: 512MB RAM Max: 8GB (Foundation), 32GB (Standard), 2TB (Enterprise, Datasenter og høyere)
Diskkapasitet	Min: 32GB eller mer Foundation: 10GB eller mer Notat: Minnevekslingsfil må legges til i størrelse, alt etter hvor mye minne maskinen har.

Tabell 2 - Operativsystemkrav

Vi vil holde oss til Datasenter-versjonen i dette prosjektet. Dette er med på å utgjøre at kapasiteten vil være stor.

2.1.1. Hva er nytt og forbedret i Windows Server 2008 R2?

Siden Windows Server 2008 er det en rekke oppdateringer som er gjort. Dette kom ut i en ny utgivelse, Windows Server 2008 R2. Det er gjort en rekke fremskritt for å øke effektivitet av tjenestene, utnytte ressursbruken av maskinvaren, effektivisere arbeidet for IT-administratorer og brukere av serverene. Vi skal blant annet se litt på hva som generelt er nytt og hva som er nytt for IT-administratorer.

- **AD Rights Management Services**

Et sikkerhetsverktøy for å administrere rettigheter til data og filer på tvers over nettverk. Verktøyet behandler informasjonen kryptert og gjør tilgangen enklere.

- **BranchCache™ Content and Hosted Server**

Dette er en stor nyhet for å frigjøre båndbredde for spredt kommunikasjon. Ved å midlertidig lagre mer data lokalt og på lokale tjenere unngår man unødvendig bruk av WAN-båndbredde.

- **Direct Access**

Microsofts nye tilkoblingsmetode til et domene når man befinner seg utenfor nettverket. Dette gjør at man slipper andre VPN-løsninger eller å måtte koble seg til direkte til nettverket i fysisk forstand. Moduler er nedlastbare fra Microsofts internettsider.

- **Hyper-V™ 2.0**

Maskin virtualiseringssoftware som er konkurrent mot VMware og Citrix. Denne er gjort betydelig bedre siden utgivelsen i Windows Server 2008. I R2 har man mulighet for å ha flere Highly Available Virtual Machines per host. Det vil si at man under samme host kan gjøre tilgjengelig flere maskiner som blir overtatt av andre hosts ved nedgang av server. Støtten for høy feiltoleranse er bedre gjennomført. Flere virtuelle cpuer er tilgjengelig samtidig. Failover Clustering er et forbedret tillegg som gjør de høytilgjengelige funksjonene for virtuelle maskinene mer effektivt og i flere antall.

- **Internet Information Services 7.5**

En ny og forbedret WEBsoftware versjon. Dette innebærer at også FTP-tjenesten og bruke av PowerShell mot dette er langt forbedret.

- **Remote Desktop Services**

Ny utviklet Remote Desktop funksjon hvor man kan holde oversikt i et konsoll over flere Remote Desktop sesjoner. Denne er ny og forbedret også i forbindelse med støtte og sikkerhet mot Windows 7.

- **Core Parking**

Støtte for strømsparing når prosessorer ikke er i bruk. Dermed kan prosessorene parkeres da arbeidsmengden er kortet nok ned til å kunne kjøres på færre prosessorer.

- **Server Management**

Fjernstyring med Server Manager er nå tilgjengelig. Dette er ytterlig fordel for IT-drifere.

- **Windows PowerShell™**

PowerShell versjon 2.0 er integrert med Windows Server 2008 R2. Forbedret siden sist utgivelse versjon 1.0.

2.1.2. Bruk av serverne

Vi vil holde oss til Datasenter-versjonen av Windows 2008 R2. Grunnlaget for dette er virtualisering av operativsystemet med Hyper-V. Vi skal kunne bruke failover-clustering funksjon som gir oss muligheten til å flytte virtuelle maskiner til andre maskiner om en fysisk maskin skulle svikte eller settes ut av drift av andre grunner. Både standard versjon og enterprise-versjon støtter ikke funksjoner som vi ønsker i slike situasjoner. Vi stiller opp en sammenlikningstabell i tabell 2¹.

Da ser vi blant annet at forskjellen fra en Enterprise-server og en Datacenter-server gir mulighet til direkte forandring av prosessor og minne. De gir også muligheten til å kjøre flere enn 4 virtuelle maskiner på en host.

¹ Tabell hentet fra Microsoft - <http://www.microsoft.com/windowsserver2008/en/us/r2-compare-specs.aspx>

Specification	Web	Standard	Enterprise	Datacenter	Itanium	Foundation	HPC
Cross-File Replication (DFS-R)							
Failover Cluster Nodes (Nodes)			16	16	8		
Fault Tolerant Memory Sync							
Hot Add Memory							
Hot Add Processors							
Hot Replace Memory							
Hot Replace Processors							
IA64 RAM					2 TB		
IA64 Sockets					64		
Network Access Connections (IAS)		50	Unlimited	Unlimited	2	10	
Network Access Connections (RRAS)		250	Unlimited	Unlimited		50	250
Remote Desktop Admin Connections	2	2	2	2	2	2	2
Remote Desktop Services Gateway		250	Unlimited	Unlimited		50	
Virtual Image Use Rights	Guest	Host + 1 VM	Host + 4 VM	Unlimited	Unlimited		Host + 1 VM
X64 RAM	32 GB	32 GB	2 TB	2 TB		8 GB	128 GB
X64 Sockets	4	4	8	64		1	4

Tabell 3 - Sammenlikning Windows 2008 R2 server-typer

Vi har valgt å ha 2 fysiske maskiner som kjører Windows 2008 R2 med rollen Hyper-V. Disse maskinene skal igjen kjøre virtuelle Windows 2008 R2 under seg.

Disse maskinene skal i hovedsak ligge på et NAS. Serverne vil kjøre disse maskinene direkte derifra som gir mulighet til failoverclustering hvor en fysisk maskin kan ta over kjøringen av maskiner som blir kjørt av en maskin som går ned.

2.1.2. Installasjon og konfigurering av servere

Vi har to fysiske servere som skal kjøre virtuelle maskiner. Både de virtuelle og fysiske maskinene skal kjøre Windows Server 2008 R2. Alle maskiner skal bli meldt inn i ett domene. Dette har ingen

betydning for om maskinene er fysiske eller virtuelle. Domenekontrolleren kjøres på en virtuell maskin med mulighet for failover-funksjon. Det gjør det mulig å opprettholde serveren i live selv om en fysisk maskin skulle gå ned. Det samme vil være med maskinen som kjører MS Exchange 2010.

Forskjellen på de fysiske maskinene og de virtuelle maskinene er hovedsakelig at rollen Hyper-V ikke kan kjøre virtuelt. Det vil si det er ikke mulig med virtuelle maskiner i virtuelle maskiner. Rollen Hyper-V vil derfor kjøres på de fysiske maskinene.

Vi oppretter Virtuelle servere med formål dedikert for hver av dem. Vi kjører ikke flere forskjellige tjenester på samme server som ikke har behov for å være på samme server.

2.1.2.1. Fysisk maskin 1 – Hyp-1

Kort oppsummering av spesifikasjoner på fysisk maskin 1 vi bruker i dette prosjektet.

- CPU: Intel Core 2 Duo 6600 @ 2.40GHz – X64
- RAM: 4GB
- Disks: 1x120GB SATA HDD, 1x500GB SATA HDD
- NICS: 1x1Gbit Integrert NIC, 1x1Gbit PCI NIC

Installasjon av Windows 2008 R2 på Hyp-1

Vi installerer Windows Server 2008 R2 på disk 1 (120GB) som fyller hele disken. Vi gir tilgjengelig disk 2 for lagring av store filer (iso-bildefiler) av installasjonscd'er vi trenger. Her er det også plass til lokalt kjørende virtuelle maskiner om vi skal opprette flere enn det er plass til på tildelt NAS.

Roller og tillegg (features)

Serveren skal være dedikert kun til å kjøre virtuelle maskiner. Derfor har vi bruk for roller og tillegg som har med virtualisering å gjøre på denne maskinen.

- Hyper-V – Virtualiseringsrollen
- Failover Cluster Manager

Montering av failover cluster stasjoner, monteres disse under *ClusterStorage* på stasjon C. Heartbeat-disken monteres som egen stasjon.

2.1.2.2. Fysisk Maskin 2 – Hyp-3

Kort oppsummering av spesifikasjoner på fysisk maskin 2 vi bruker i dette prosjektet.

- CPU: Intel Core 2 Duo 6600 @ 2.40GHz – X64
- RAM: 4GB
- Disks: 1x120GB SATA HDD, 1x500GB SATA HDD
- NICS: 1x1Gbit Integrert NIC, 1x1Gbit PCI NIC

Installasjon av Windows 2008 R2 på Hyp-3

Vi installerer Windows Server 2008 R2 på disk 1 som fyller hele disken. ISO-filer og kopier lagres også lokalt på stasjon C under en egen mappe.

Roller og tillegg (features)

Serveren skal være dedikert kun til å kjøre virtuelle maskiner. Derfor har vi bruk for roller og tillegg som har med virtualisering å gjøre på denne maskinen.

- Hyper-V – Virtualiseringsrollen
- Failover Cluster Manager

Montering av failover cluster stasjoner, moteres disse under *ClusterStorage* på stasjon C. Heartbeat-disken monteres som egen stasjon.

2.1.2.3. Fysisk maskin 3 – Hyp-2

Kort oppsummering av spesifikasjoner på fysisk maskin 3 vi bruker i dette prosjektet.

- CPU: AMD Athlon Dual core processor 4850 2.5GHz – x64
- RAM: 4GB RAM
- Disk: 1x600GB SATA HDD
- NICS: 1x1Gbit Integrert NIC, 1x100Mbit USB NIC

Denne maskinen blir brukt som stand-alone maskin utenfor clusteret. Denne vil bli brukt til å kjøre maskiner som vi må ha oppe for testing av mindre tjenester og klienter. Servere som ikke trenger failoveregenskaper legger vi også her.

Installasjon av Windows 2008 R2 på Hyp-2

Vi installerer Windows Server 2008 R2 på disk 1 som fyller hele disken. Disk 2 brukes til lagring av ISO-filer og kopier av maskiner og virtuelle harddisker.

Roller og tillegg (features)

Serveren skal være dedikert kun til å kjøre virtuelle maskiner. Derfor har vi bruk for roller og tillegg som har med virtualisering å gjøre på denne maskinen.

- Hyper-V – Virtualiseringsrollen
- Failover Cluster Manager

Montering av failover cluster stasjoner, moteres disse under *ClusterStorage* på stasjon C. Heartbeat-disken monteres som egen stasjon.

Selv om maskinen ikke skal være med i et cluster, er den satt opp på forhånd til dette. Den kan da innlemmes i clusteret om behovet skulle melde seg.

2.2 Hva er virtualisering?

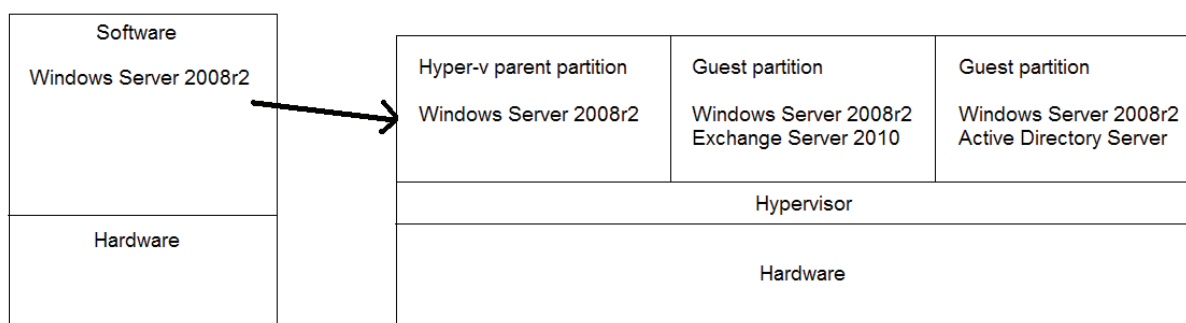
Virtualisering er en teknikk som har eksistert i én eller flere former helt siden dataalderens begynnelse. Tanken er at man ved å bruke tilgjengelig - og ellers ubrukt kapasitet og ressurser på én eller flere fysiske hosts - kan kjøre flere separate operativsystemer på samme hardware samtidig. Disse blir ofte kalt for "Guest Operating Systems" fordi de kjører på hostmaskinen. Dette gjør at man kan redusere behovet for fysisk hardware, og dermed spare store kostnader på serverfarmen vår sammenliknet med om vi måtte kjøre hvert serveroperativsystem på sin egen fysiske hardware.

2.2.1 Fordeler med virtualisering

Vi har allerede sett at vi kan spare penger på å utnytte virtualisering til det fulle. Det som ellers er fordelene med virtualisering, i alle fall i en oppsettsfase av et nettverk, er at Hyper-V og andre tilsvarende virtualiseringsteknologier støtter snapshots. Når et snapshot tas, blir et bilde av operativsystems nåværende miljø – inkludert RAMen som er i bruk - lagret til ei diskfil. Når så endringer i den virtuelle maskinen fører til feilsituasjoner, er det ved hjelp av snapshots mulig å gjenopprette det virtuelle miljøet til en tidligere tilstand.

2.2.2 Hvordan Hyper-V fungerer

Etter en installasjon av Windows Server 2008r2, vil vi som vist på figurene nedenfor, sitte igjen med ett operativsystem som har full tilgang til hardware. Skulle vi kjørt en workstation-versjon av kjente virtualiseringsteknologier slik som Windows Virtual Server 2005 eller VMWare workstation, som begge er program som kjører som et program inne i Windows, ville det bety at all bruk av hardwareressurser først måtte gå igjennom operativsystemet, og dermed vil ikke de virtuelle maskinene fungere med topp ytelse. Denne ytelsesreduksjonen er grunnen til at virtualisering inntil ca. 2006 ikke var vesentlig brukt i datasentre som har høye krav til oppetid eller ytelse, som definert i SLA-avtalene² som er inngås. For i 2006 fikk nemlig den mest bruke prosessorarkitekturen x86 - som også kjøres av de fleste prosessorer på arbeidstasjoner – støtte for maskinvareakselerert virtualisering. Dette betyr at vi kan kutte ut mellomledet som er nødvendig ved tradisjonell virtualisering, og kjøre de virtuelle maskinene direkte på hardwaren, med bare et tynt mellomlag imellom – kalt en hypervisor – som kontrollerer tilgang til hardwaren.



Figur 6 Serversoftwareoppsett før og etter Hyper-V-installasjon

Overfor ser vi at et før og etterbilde som illustrerer hvordan hardwaren og softwaren blir konfigurert etter Hyper-V er installert, samt hvilke tjenester som skal kjøres på vår fysiske host. Det har blitt

² SLA eller Service Level Agreement er en avtale mellom leverandør og brukere av tjenester som definerer hvilke krav som settes for tjenestene som leveres.

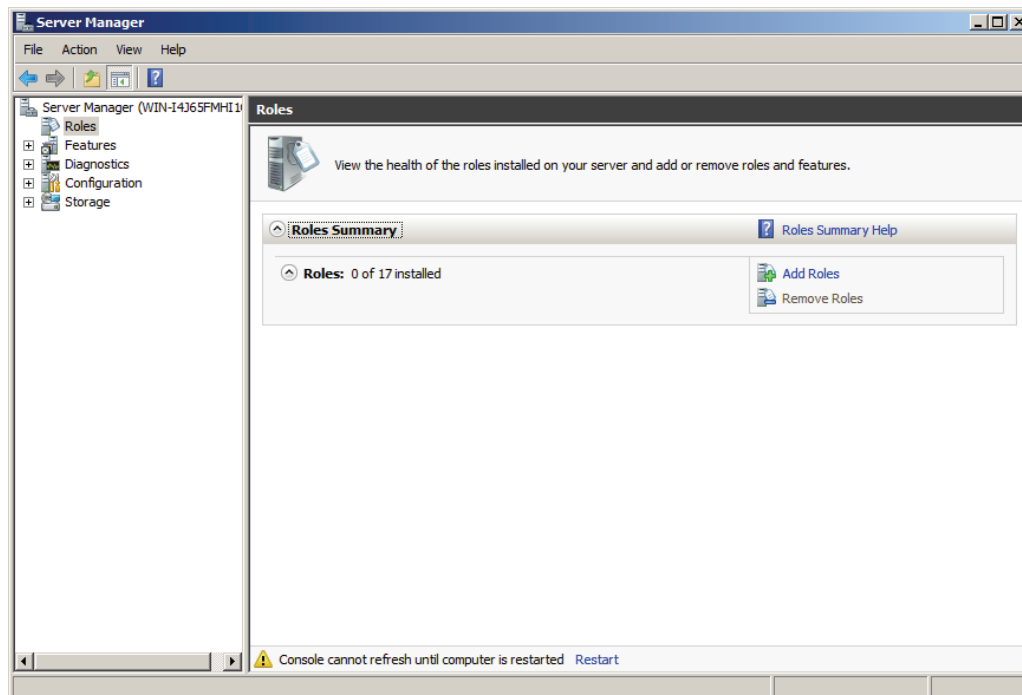
introdusert en såkalt "Parent partition" og flere "Guest partitions". En partition inneholder ett operativsystem. Forskjellen er at en parent partition inneholder det operativsystemet vi hadde før installasjonen av Hyper-V, mens alle guest partitions kjører andre virtuelle maskiner. Vi ser videre at alle partitions nå vil benytte seg av hypervisoren for å kommunisere med hardwaren, men en guest vil kun ha tilgang til å se de delene av hardwaren som den har blitt tildelt av en parent. Parent eksisterer derfor for å kunne administrere de andre virtuelle maskinene som kjører på Hyper-V-serveren. Vi skal seinere vise hvordan installasjonen av guest partitions foregår, og hvordan tjenestene Active Directory og Exchange Server isntalleres.

2.2.3 Installasjon av Hyper-V

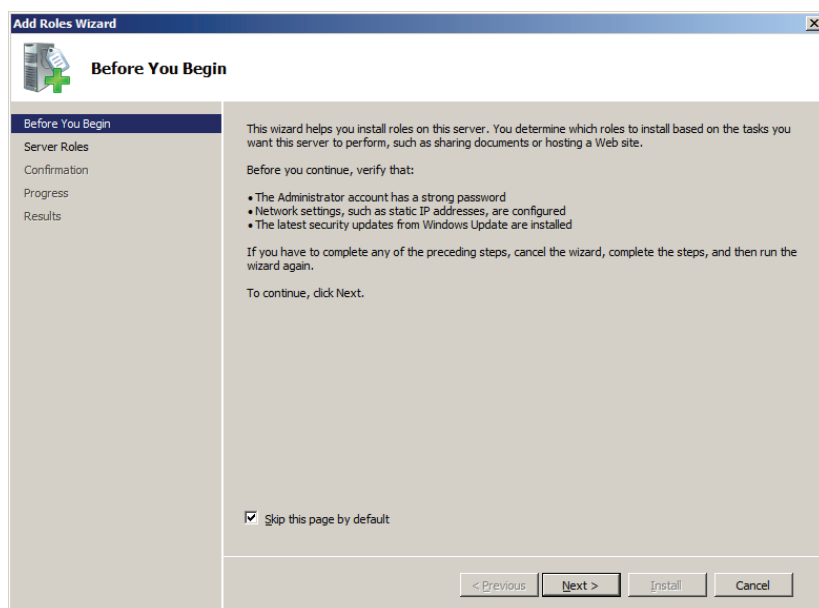
Hyper-V kommer i mange forskjellige utgaver, og har fått en kraftig ytelsesforbedring i forbindelse med slipp av Windows Server 2008r2. I Windows Server 2008 var det ikke mulig å installere Hyper-V med mindre en kjørte 64-bit-versjonen av operativsystemet. I forbindelse med Server 2008r2, er støtten for 32bit operativsystem også faset ut for Windows Server-operativsystemet. Dette betyr at om maskinvaren støtter Server 2008r2, kan en også installere Hyper-V-rollen.

Microsoft tilbyr en standalone Hyper-V server gratis fra sine hjemmesider. Denne er derimot så nedstrippet og mangler grafisk grensesnitt, at vi har satset på den fulle installasjonen av Server 2008r2 pluss Hyper-V som en egen rolle i stedet, som vi skal vise hvordan installeres nedenfor.

Server manager er det programmet som brukes for å installere roller og tjenester i Windows Server, det finnes under administrative tools-menyen på startmenyen. Nedenfor ser vi at vår serverboks ikke har noen roller installert. I forbindelse med en Hyper-V -server som skal ha høy oppetid er det viktig å ha en så liten angrepsflate som mulig.



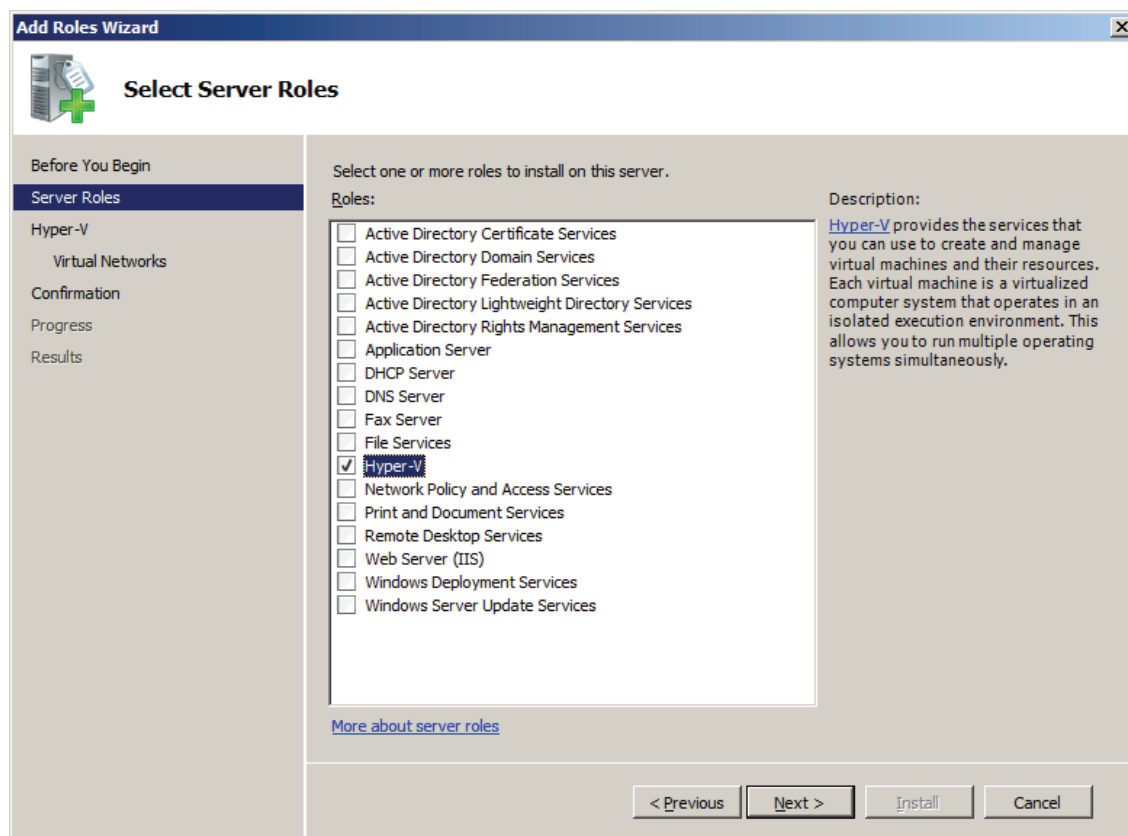
Figur 7 - Rolleinstallasjon med Server Manager



Her ser vi generell sikkerhetsinformasjon som dukker opp når vi trykker "Add roles" i forrige skjermbilde. Etter informasjonen er lest, har vi valgt å krysse av for å ikke vise denne informasjonen

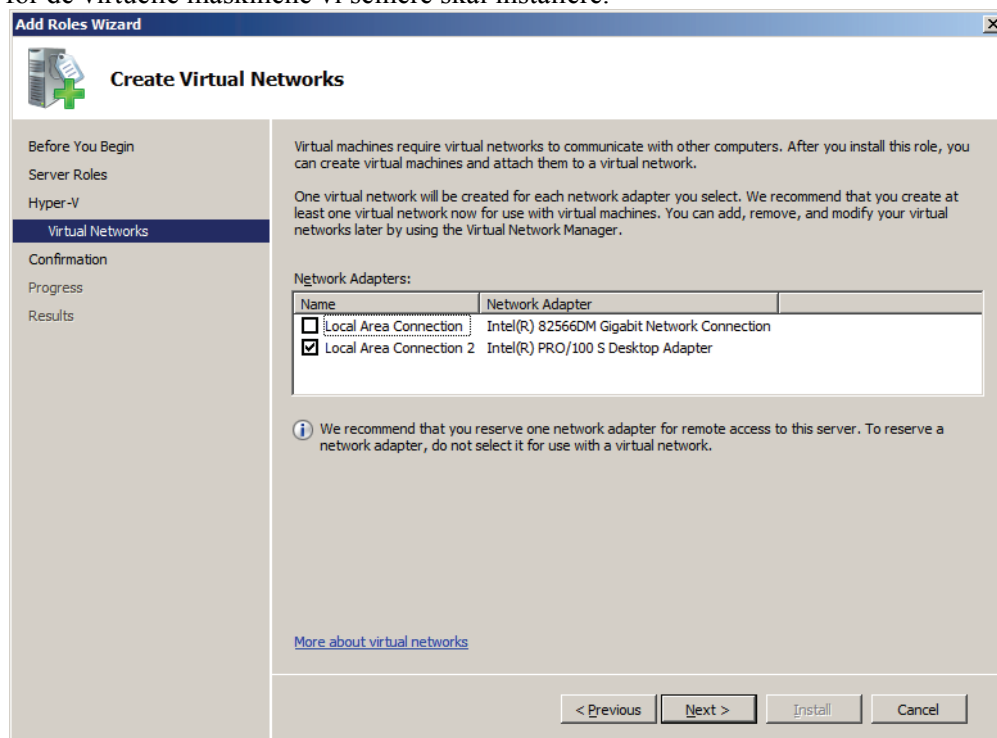
Figur 8 - Ting som må leses før installasjon

Vi velger å installere rollen Hyper-V som vist under.



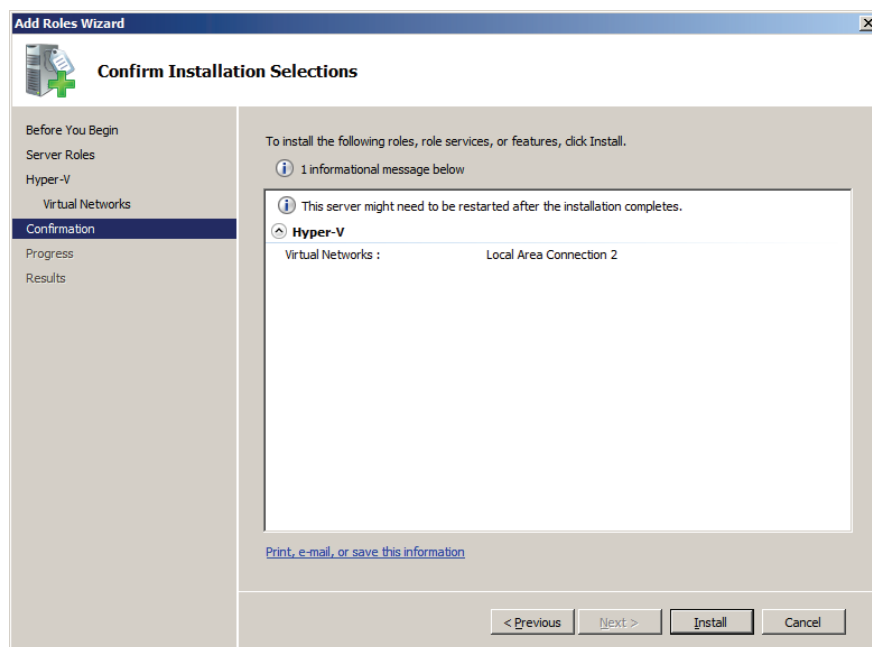
Figur 9 - Hyper-V-rollen installeres

Ettersom vår Hyper-V boks har flere nettverksadaptere, velger vi nedenunder kun én av disse som skal knyttes opp mot et virtuelt adapter. Dette virtuelle adapteret vil da framstå som et nettverkskort for de virtuelle maskinene vi seinere skal installere.



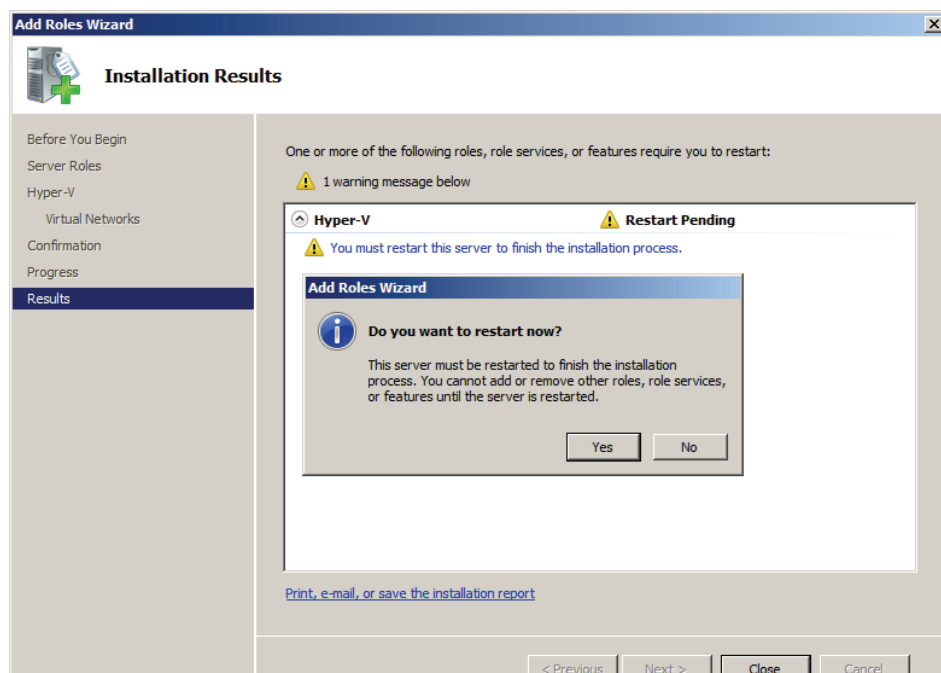
Figur 10 - Valg av ett eller flere nettverkskort som skal brukes av Hyper-V sine virtuelle maskiner. Her fravelger vi det adapteret som går mot SANet

Vi får nå detaljoversikt over hvilke innstillinger som skal gjelde for vår installasjon av Hyper-V



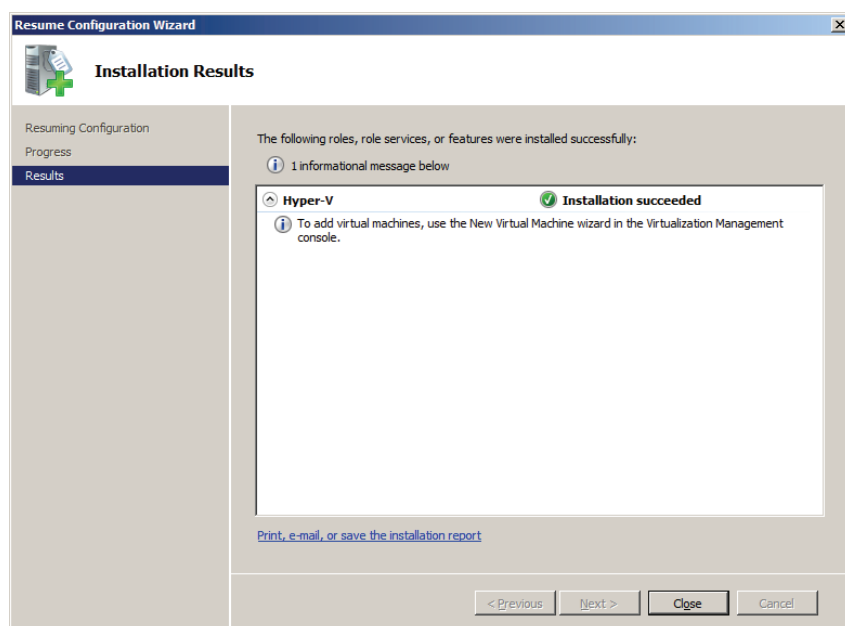
Figur 11 - Installasjon utføres med disse innstillingene

For at Hyper-V-rollen skal aktiveres, kreves det en omstart. Dette skyldes at operativsystemet skal ta i bruk hypervisoren i bunn for å kommunisere med hardware og styre tilgang for de virtuelle maskinene som skal kjøres.



Figur 12 - Rollen Hyper-V krever omstart for å legge inn hypervisor

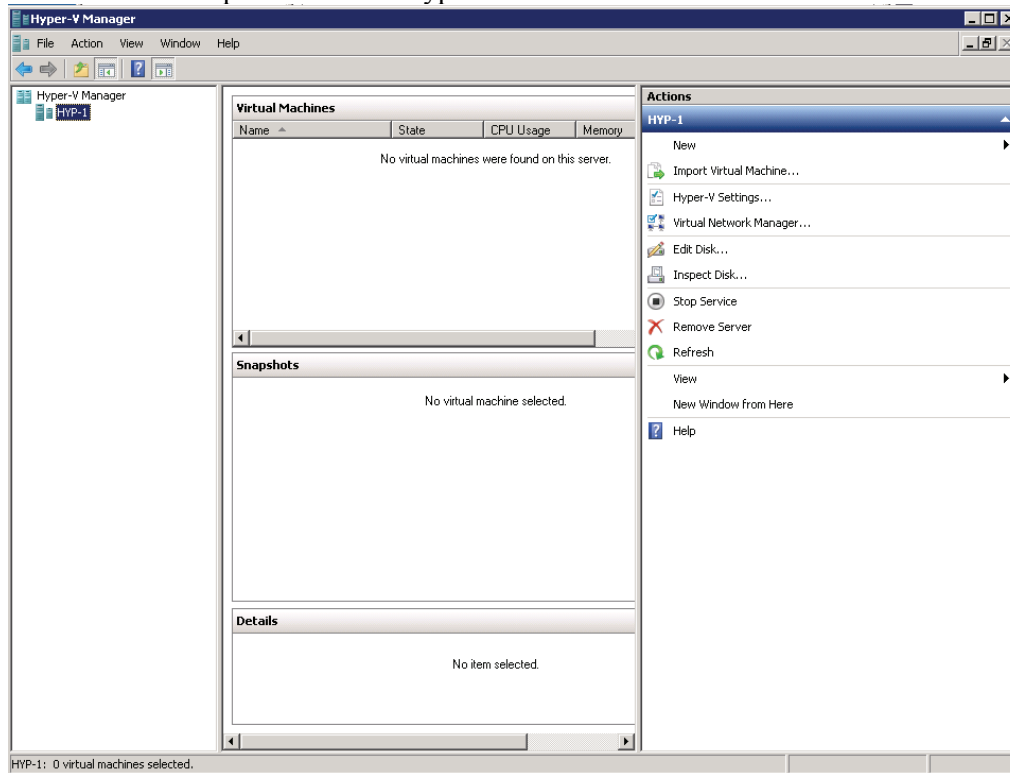
Etter en restart av systemet, vil Configuration Wizard dukke opp igjen. Vi ser at installasjonen av Hyper-V er ferdig.



Figur 13 - Installasjon av Hyper-V ferdiggjort

2.2.4 Installasjon av virtuelle maskiner

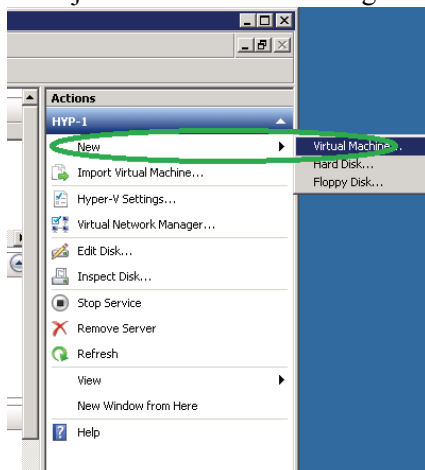
Etter installasjon, legger vi merke til at Hyper-V Manager blir tilgjengelig som i startmenyen. Programmet kan også startes gjennom Server Manager. I lista til venstre ser vi alle Hyper-V-servere som kan styres. Her vil vi seinere legge til enda en Hyper-V-server, og vi har derfor også valgt å skifte datamaskinnavnet på serveren til "hyp-1"



Figur 14 - Oversiktsbilde i Hyper-V Manager

2.2.4.1 Installasjon av nye operativsystem

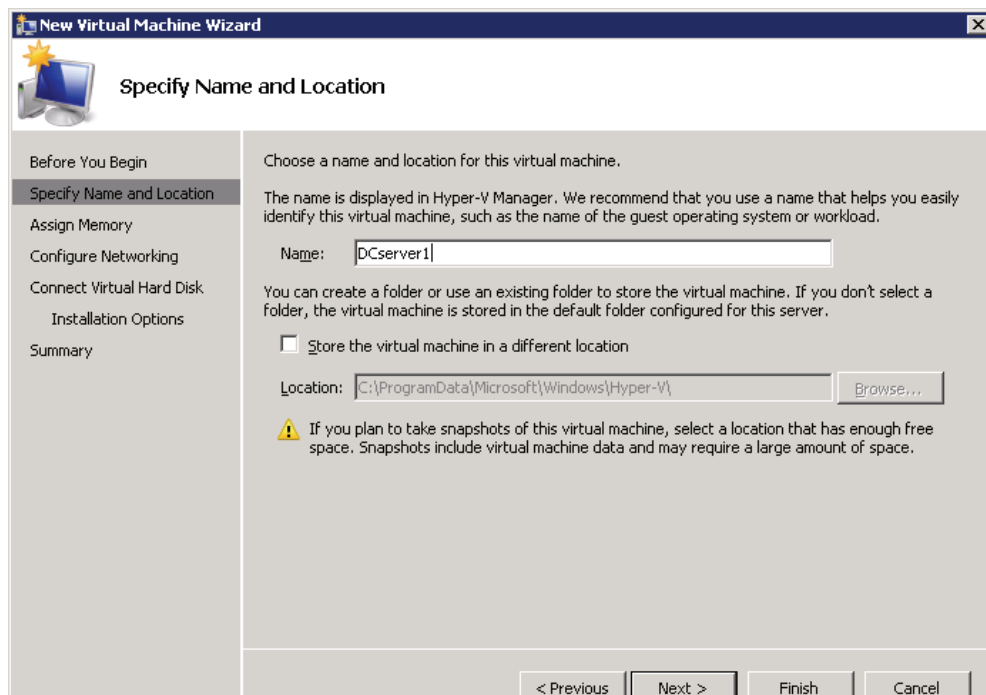
Her ser vi høyrepanel i Hyper-V Manager. Installasjon av et nytt operativsystem, startes ved å velge "new" og deretter "Virtual Machine". Dette skal vi bruke for å installere to virtuelle maskiner, som skal kjøre henholdsvis Exchange Server og Active Directory.



Figur 15 - Opprettelse av en ny virtuell maskin

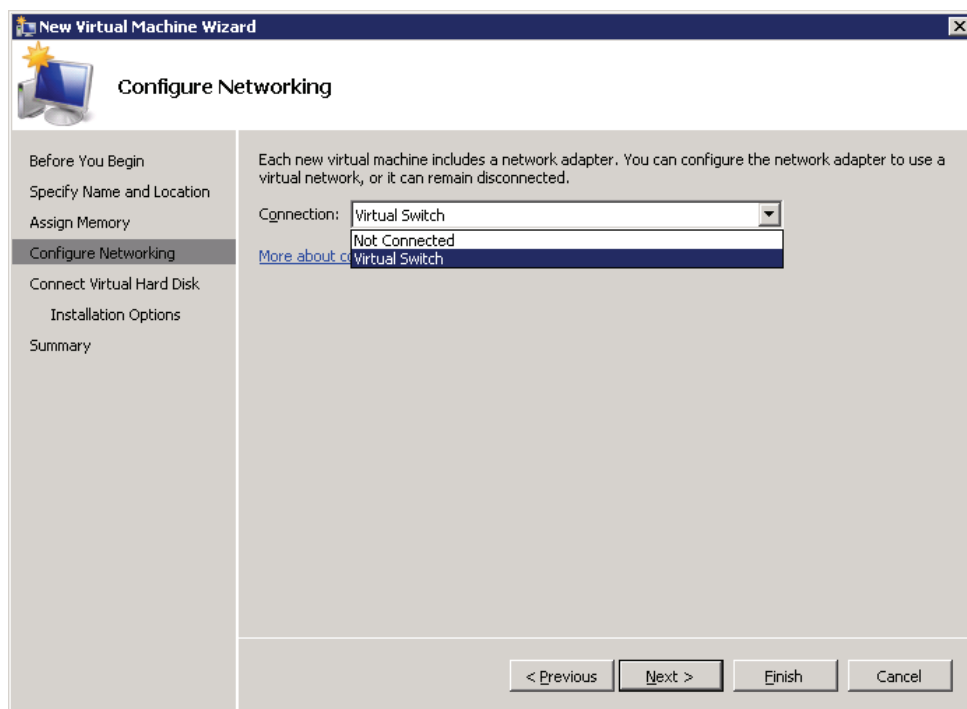
Her må vi være veldig obs på at de virtuelle maskinene vil bruke av hyp-1 sin totale harddiskplass og andre ressurser slik som RAM. I tilfellet med Exchange Server som må ha høy oppetid, vil dette ofte ikke være tilstrekkelig. Vi skal seinere se på hvordan vi kan bruke System Center Virtual Machine Manager og Failover Clustere til å la Exchange bruke et SCSI-basert nettverksdisk til å holde dataene våre, slik at en Hyper-V host kan feile uten at de virtuelle maskinene som kjører får nedetid.

Her ser vi installasjonen av en Active Directory Server, der den navngis og plasseres på den lokale disken.



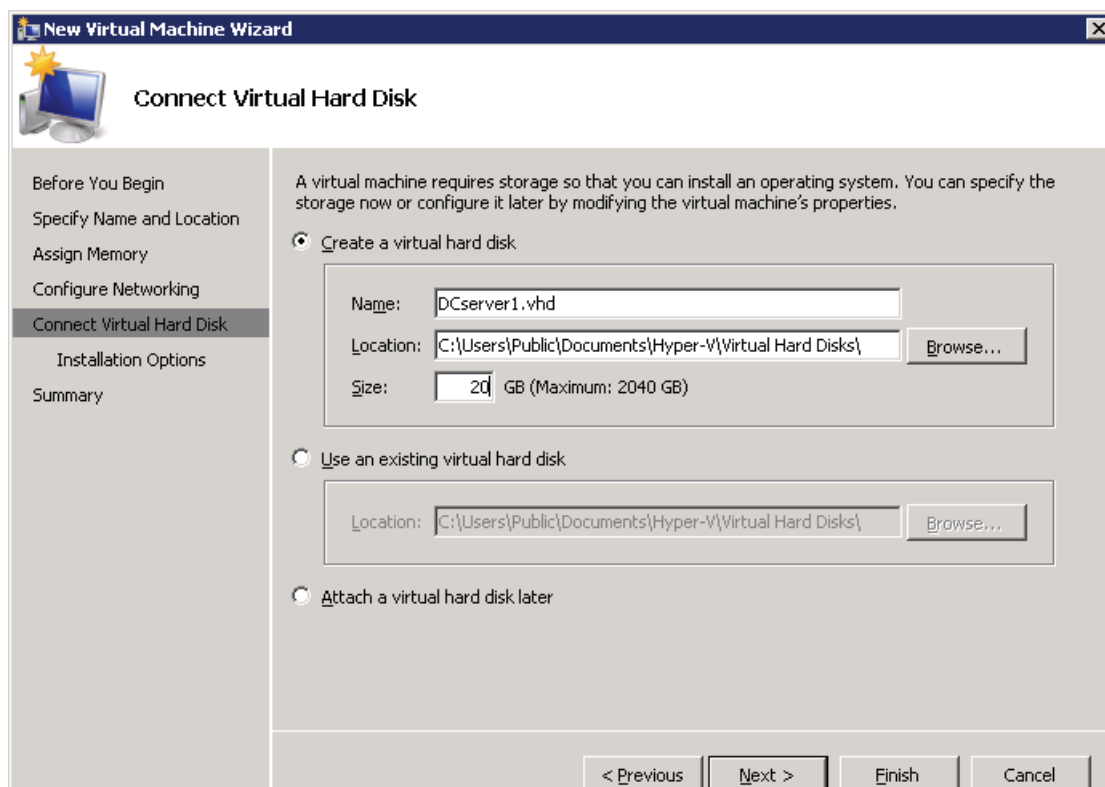
Figur 16 - Navngivelse av virtuell maskin

Minnemengden bør stå på et minimum 1024 MB. Når en kommer til nettverksvalg, må en virtuell switch tilkoblet ett fysisk nettverksadapter på Hyp-1 velges. Dette vil Hyper-V sette opp for deg.



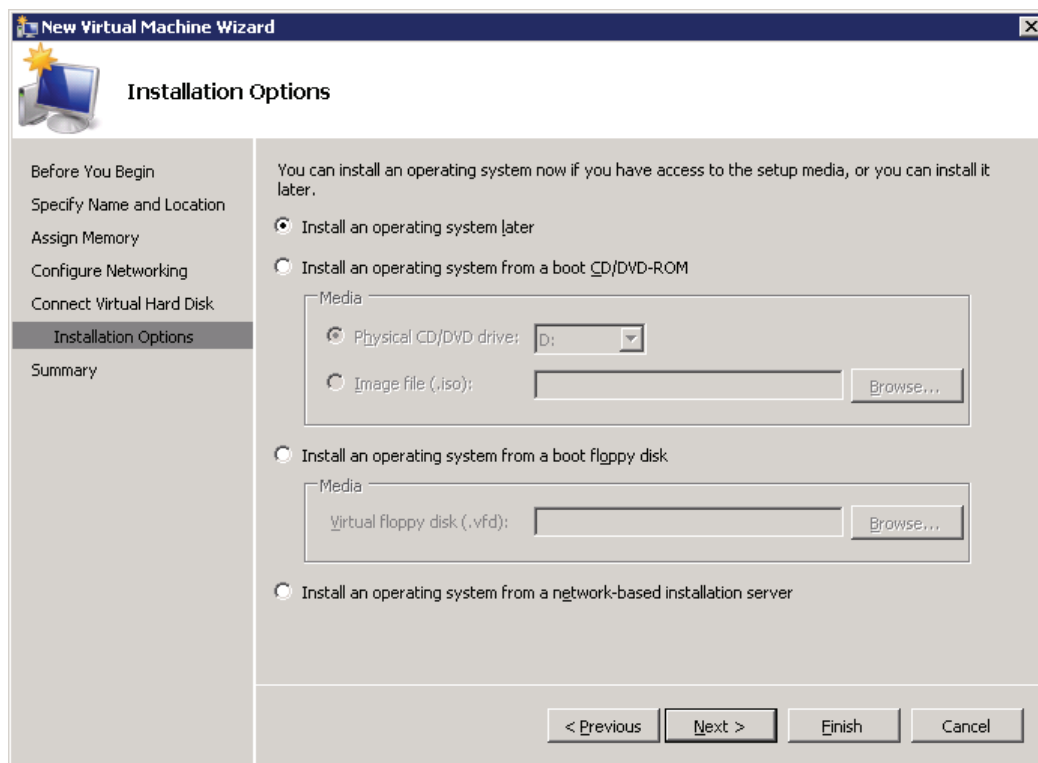
Figur 17 - Valg av virtuell nettverksswitch

Her ser vi at Hyper-V lar oss koble til en virtuell harddisk til den nye gjesten. Disse virtuelle harddiskene lagres på hyp-1 som en enkelt fil framfor som en partisjon på disken. Dette sørger for at vi seinere kan flytte virtuelle maskiner ved å bare flytte en enkelt vhd-fil pluss konfigurasjonen av selve maskinen.



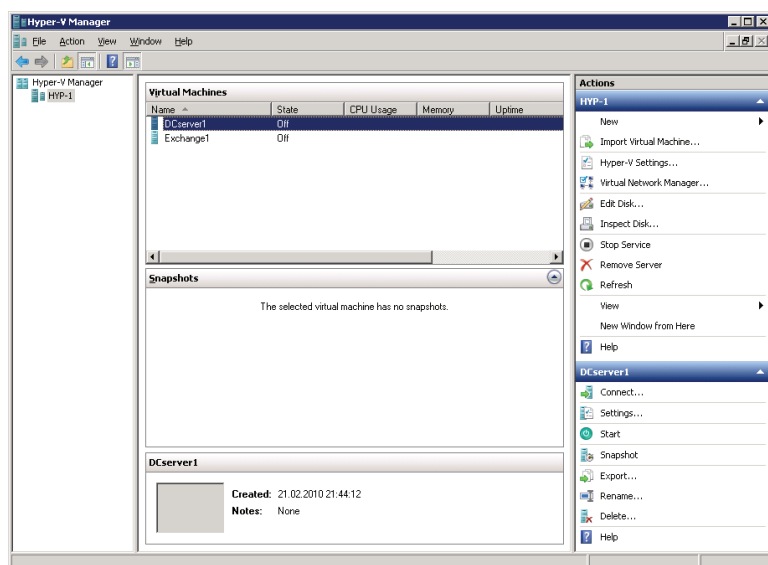
Figur 18 - Valg av virtuell lagringsmedium for den nye virtuelle maskinen

Installasjonen kan foregå på flere måter, blant annet fra DVD eller fra en virtuell floppy disk. Dersom operativsystemet som skal installeres, finnes i form av speilbildefiler(iso-filer), vil installasjonen gå en del forttere enn fra dvd-media. I seksjonen 3.2.3 vil vi også gå igjennom opprettelse av OS fra templates, da ved hjelp av System Center Virtual Machine Manager (SCVMM).



Figur 19 - Flere valgmuligheter for installasjon av OS

Her ser vi at to virtuelle maskiner er klargjort for installering.



Figur 20 - Hyper-V Manager med virtuelle maskiner aktivert

2.3. Active Directory (AD), Domenekontroller

Active Directory (AD) er en samling med flere roller som lar deg katalogisere og samle inn informasjon i et nettverk. Ved hjelp av AD kan en administrere datamaskiner, brukere, brukergrupper og rettigheter på filer, så vel som å integrere disse med andre tjenester, slik som for eksempel epost-tjenester og sørge for å gi en tilpasset opplevelse til hver bruker ved å sentralisere innloggingstjenester, hjemmemapper og fellesområder. AD blir derfor mest aktuelt å benytte i bedriftsnettverk eller hjemmenettverk hvor en vil ha sentralisert kontroll over datamaskiner.

Active Directory består av flere roller kan benyttes. De har alle sine bruksområder, men det er hovedsakelig AD DS og AD LDS som vi skal benytte oss av i dette prosjektet.

- **Active Directory Certificate Services (AD CS)**

Benyttes når en ønsker å utstede og validere sertifikater. Sertifikater benyttes i mange sammenhenger for å validere at en fil eller tjeneste er mottatt uten at det er gjort endringer av en tredjepart. Dette sikrer at en ikke kjører skadelig kode. Denne tjenesten benyttes i rollen Network Access Protection og kan også benyttes for å sikre webkommunikasjon.

- **Active Directory Domain Services (AD DS)**

AD DS (tidligere kalt bare Active Directory, AD) er databasen som inneholder all sentralisert data. AD DS fungerer som en domenekontroller, der et hver bruker, datamaskin og grupperer representert av et objekt i et hierarkisk tre. Hver grein i dette treet er representert av en organisasjonsenhet eller OU. Ved hjelp av policyer (Group Policy Objects, GPO) kan en administrator så velge å begrense eller åpne opp rettighetene til et sett med objekter, basert på hvor de ligger i dette treet. Alle objekter vil arve reglene satt i overordnede OUs GPOer

- **Active Directory Federation Services (AD FS)**

AD FS er en rolle som kan installeres for å ha en løsning som kan skalere til veldig mange brukere. Den har også muligheter for å tillate sikker innlogging fra både windows-og andre miljøer, dvs både fra andre operativsystemer og over internett.

- **Active Directory Lightweight Directory Services (AD LDS)**

AD LDS er som navnet sier en lettvektversjon av AD DS. Den kan lagre brukernavn og grupper, men den kan ikke lagre GPOer eller passord. Fordelen med denne rollen, er at den kan kjøres på en server i et annet domene, eller på en server som ikke er meldt inn i et domene. Vi skal seinere se på oppsett av Microsoft Exchange Edge Server, som aktivt bruker AD LDS replikert fra vårt domene, for å holde informasjon om brukere og epostkontoer. Dette blir et ekstra sikkerhetstiltak, fordi en hacker som greier å komme seg inn på en Edge Server, ikke vil ha mulighet til å forårsake skade på andre servere på domenet.

- **Active Directory Rights Management Services (AD RMS)**

AD RMS er en rolle som sørger for at rettigheter bevares når ei fil blir lest og kopiert. Bare opphavsmannen bak fila vil ha mulighet til å endre disse rettighetene.

Et nytt rammeverk (.NET framework 3.5.1) er nå påkrevd for å installere noen av disse rollene. Dette for å supportere funksjonene i webtjenesten. Nye sett med verktøy er også laget for å administrere Active Directory. Dette er bygd over PowerShell og gjør det enklere å integrere egne løsninger for å

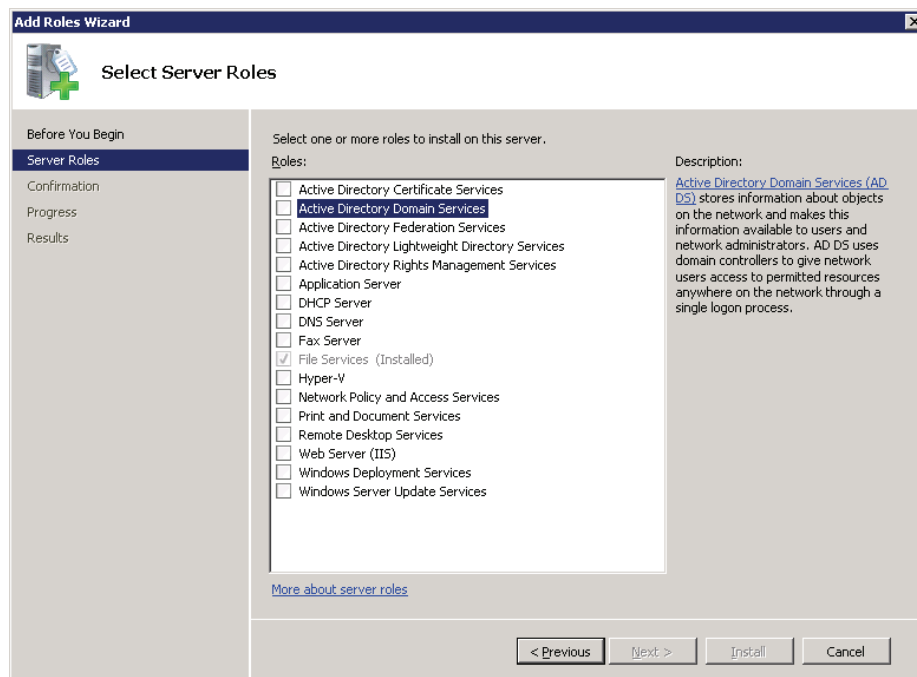
administrere rollene. Det er også mulig for arbeidsstasjoner som ikke har tilgang til domenekontrolleren å få tilgang til deler av nettverket. Dette brukes Best Practices Analyzer for.

2.3.1. Active Directory Domain Services

Vi skal i hovedsak konsentrere oss om Active Directory Domain Services (AD DS) i dette prosjektet for å ta dette i bruk og senere lage script og administrere andre script for å styre Active Directory.

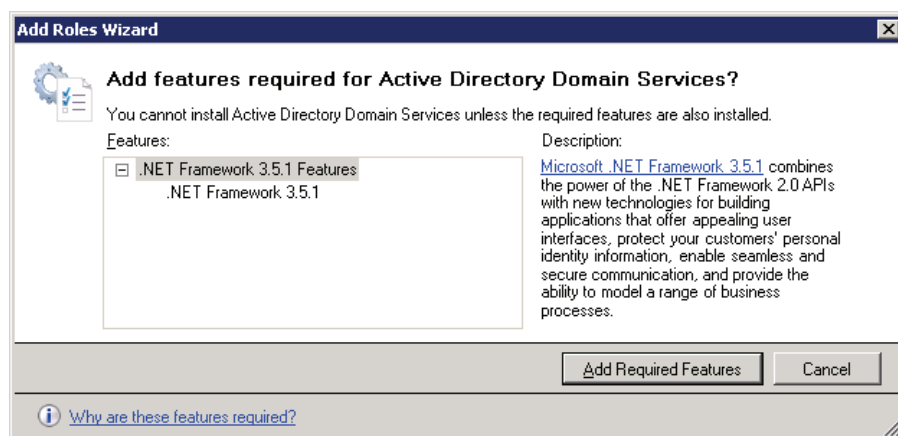
2.3.2. Installasjon og konfigurering av AD DS

Vi åpner Server Manager velger roller. Her er vi at vi har fem forskjellige Active Directory Services. Vi skal legge til Domain Services som tar seg av objekter i nettverket og behandler disse.



Figur 21 - AD DS Rollevalg

Dermed får vi beskjed om å installere .NET Framework. Dette er en oppdatering fra forrige rammeverk som er påkrevd i samarbeid med IIS v7.5. Selv om vi ikke skal installere IIS v7.5 er denne enheten påkrevd for eventuell fremtidig installasjon av IIS.



Figur 22 - .NET Framework 3.5.1 tillegg påkrevd

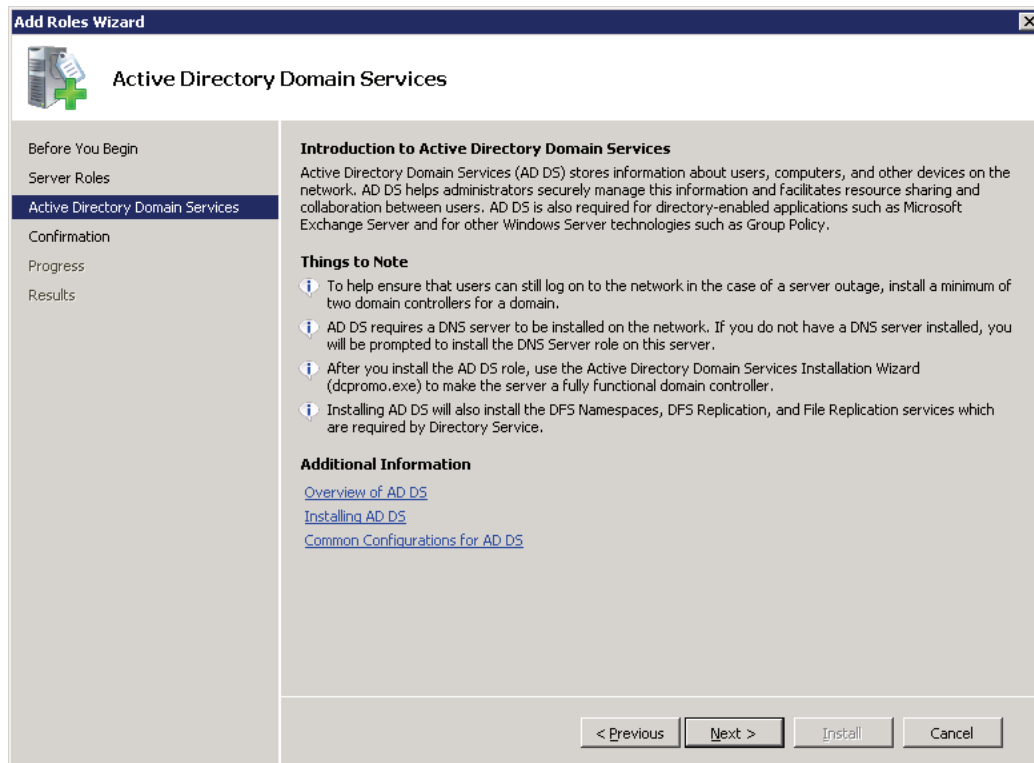
Videre får vi en kort introduksjon om hva AD DS er. Det er gitt forslag om at 2 domenekontrollere skal installeres for å sikre at brukere skal få tilgang til å logge på om de ikke skulle få tilgang til en av

domenekontrollerene. Dette vil vi innarbeide i en løsning i Hyper-V ved å virtualisere serverne og gjøre dem Highly Available. Det vil si at kjøring av maskinene vil bli overlatt til en annen maskin om den ene skulle falle ned.

Vi blir også påmint om at en DNS-tjeneste må knyttes til denne. Det er da mulighet etter installasjon av AD DS å installere denne sammen med domenekontrolleren.

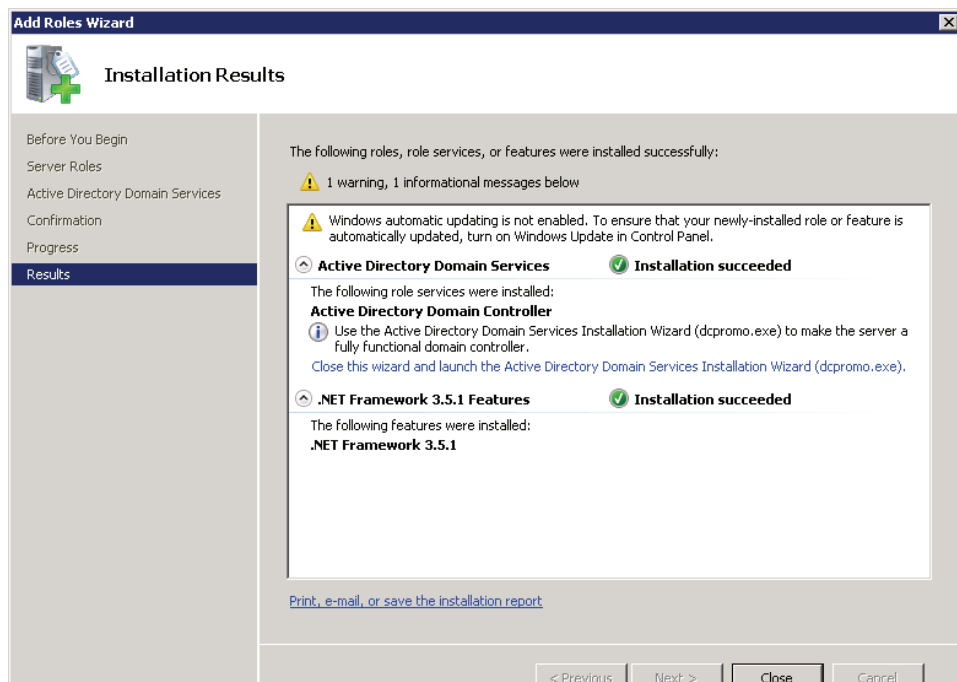
Et tilleggsverktøy må kjøres etterpå for å sette domenekontrolleren i gang. Dette vil vi få en snarvei til som peker mot å kjøre DCPROMO.EXE. Eller den kan kjøres fra kommandolinjen.

Videre får vi beskjed om hva videre det blir bruk for i AD og hva som blir lagt til. Se Figur 23 - AD DS Installasjonsnotater.



Figur 23 - AD DS Installasjonsnotater

Deretter får vi beskjed om hva som blir gjennomført og deretter resultatene. Vi vil få advarsler om eventuelle tillegg som ikke er kjørt, eller sikkerhet som bør gjennomføres i tillegg. Se Figur 24 – Resultater.

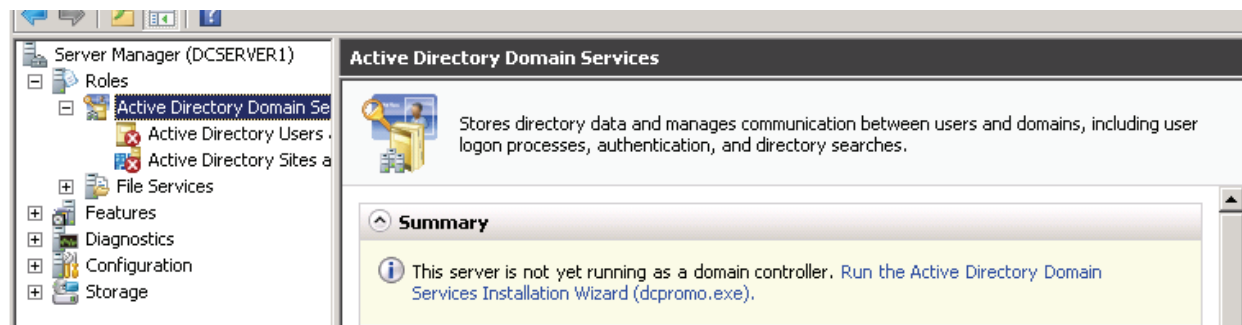


Figur 24 – Resultater

2.3.3 Gjennomføring, tillegg og igangsetting av AD DS

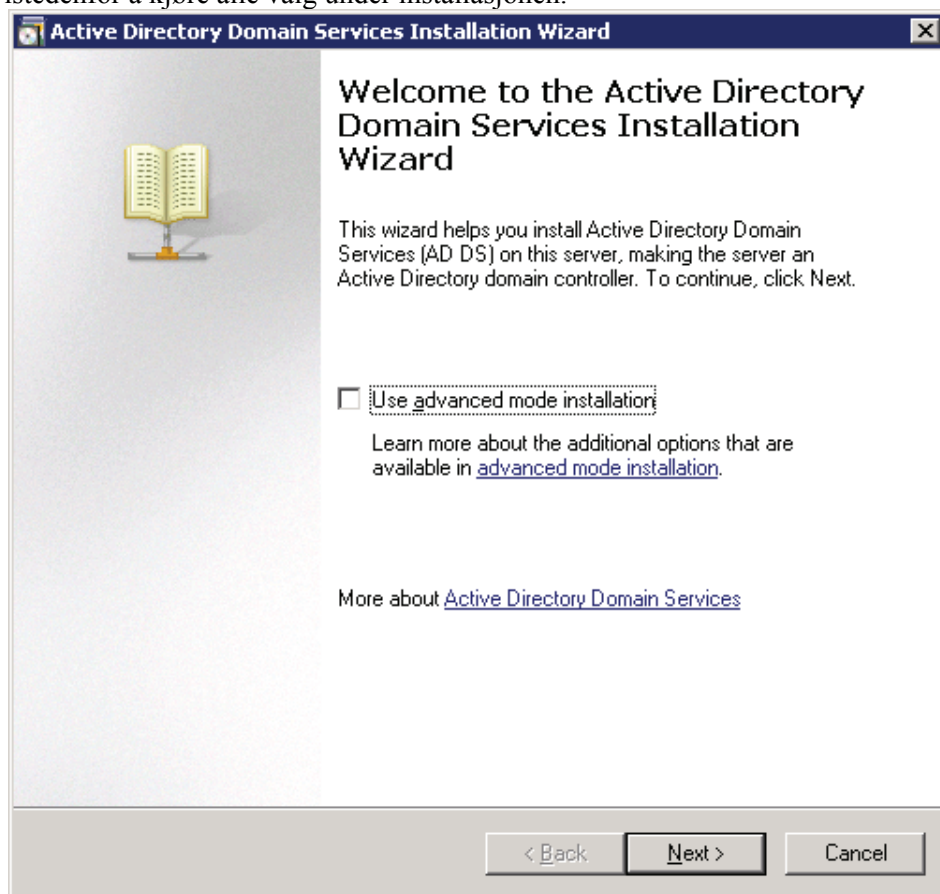
Installeringen av AD DS er fullført, men den er fortsatt ikke aktiv. Dette må gjøres ved å kjøre DCPROMO.EXE som tidligere nevnt. Dette er forklart i rollens management-vindu. Det sies ifra om at aktivering må gjøres.

For å aktivere AD DS er det en blå link til DCPROMO.EXE. Vi setter i gang denne. Se Figur 25 - AD DS - DCPROMO.EXE.



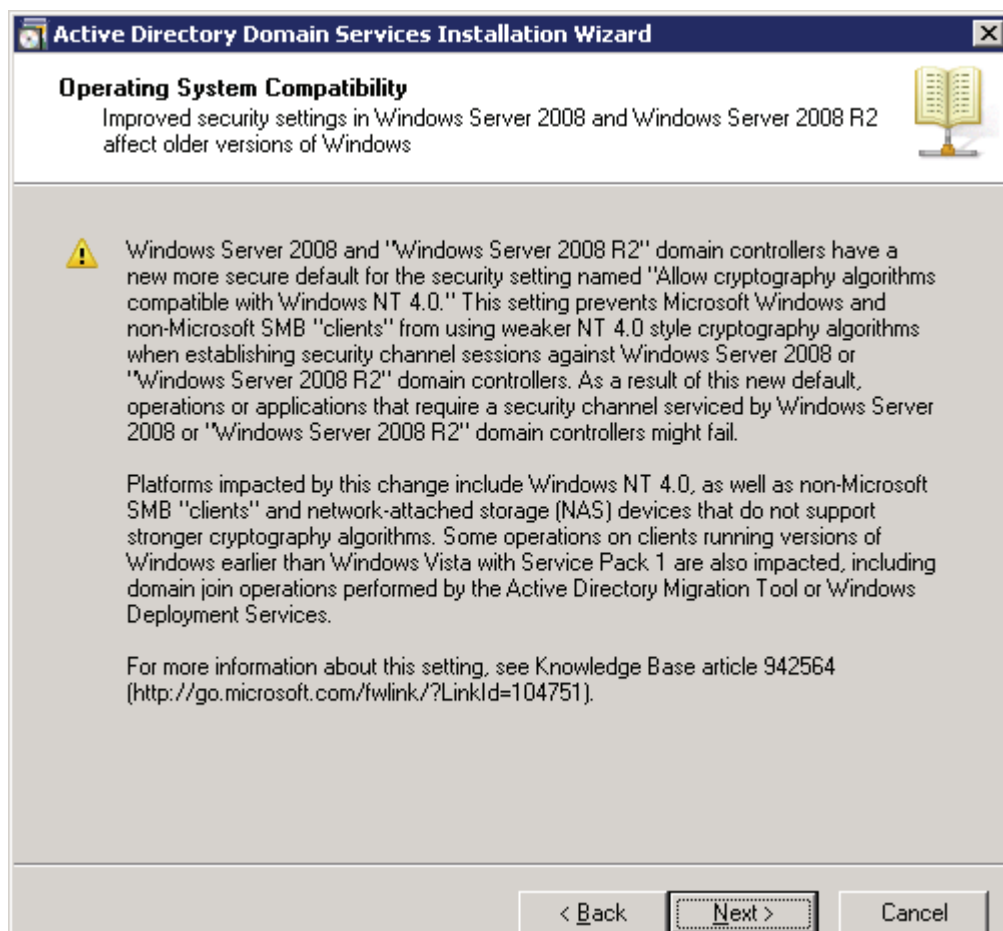
Figur 25 - AD DS - DCPROMO.EXE

For å få denne i gang velger vi å ikke kjøre Advanced Installation. Alle forandringer kan gjøres senere istedenfor å kjøre alle valg under installasjonen.



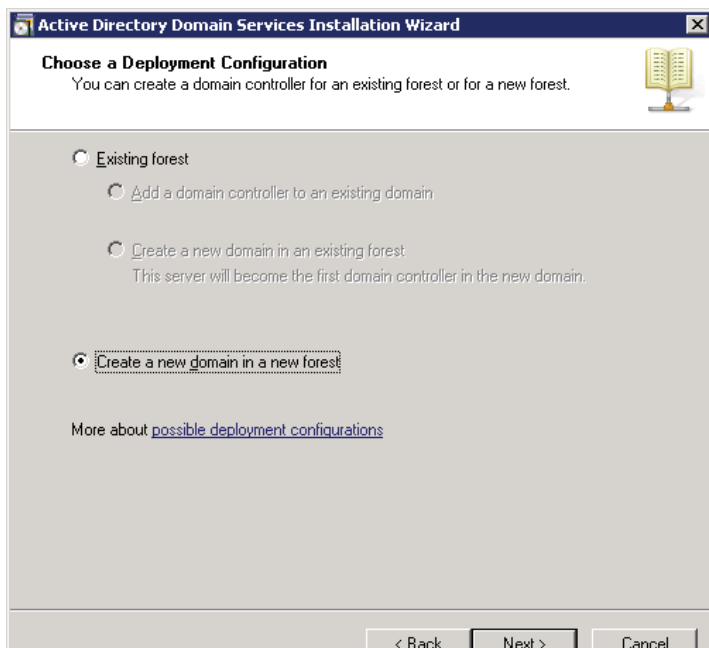
Figur 26 - Advanced Installation

Videre får vi beskjed om eksisterende Windows NT 4.0 servere ikke har den kryptografien som Windows Server 2008 har. Dette kreves. Derfor vil disse gamle NT 4.0 serverne feile. Alle de gamle serverne bør derfor oppgraderes eller fjernes fra nettverket. Dette skal ikke være noe problem i det nåværende nettverket som denne driftsdokumentasjonen er beregnet på siden det ikke finnes noen NT 4.0 servere der fra før.



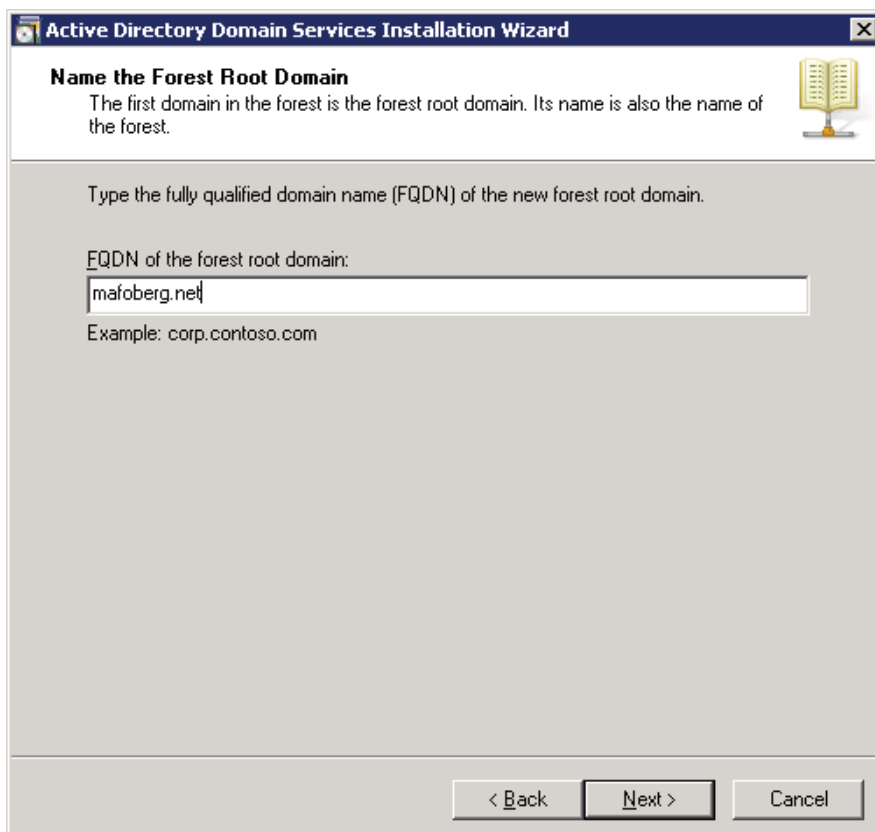
Figur 27 - NT 4.0 støtte

Vi vil i denne igangsettingen opprette en ny skog for bedriften. Dette for at hele domenet og strukturen skal gjennomføres på nytt i lengre periode. Derfor går vi frem med å opprette et nytt domene her. Planen er å migrere alle eksisterende brukere fra gamle servere inn i denne nye skogen og oppsettet.



Figur 28 - Nytt domene i ny skog

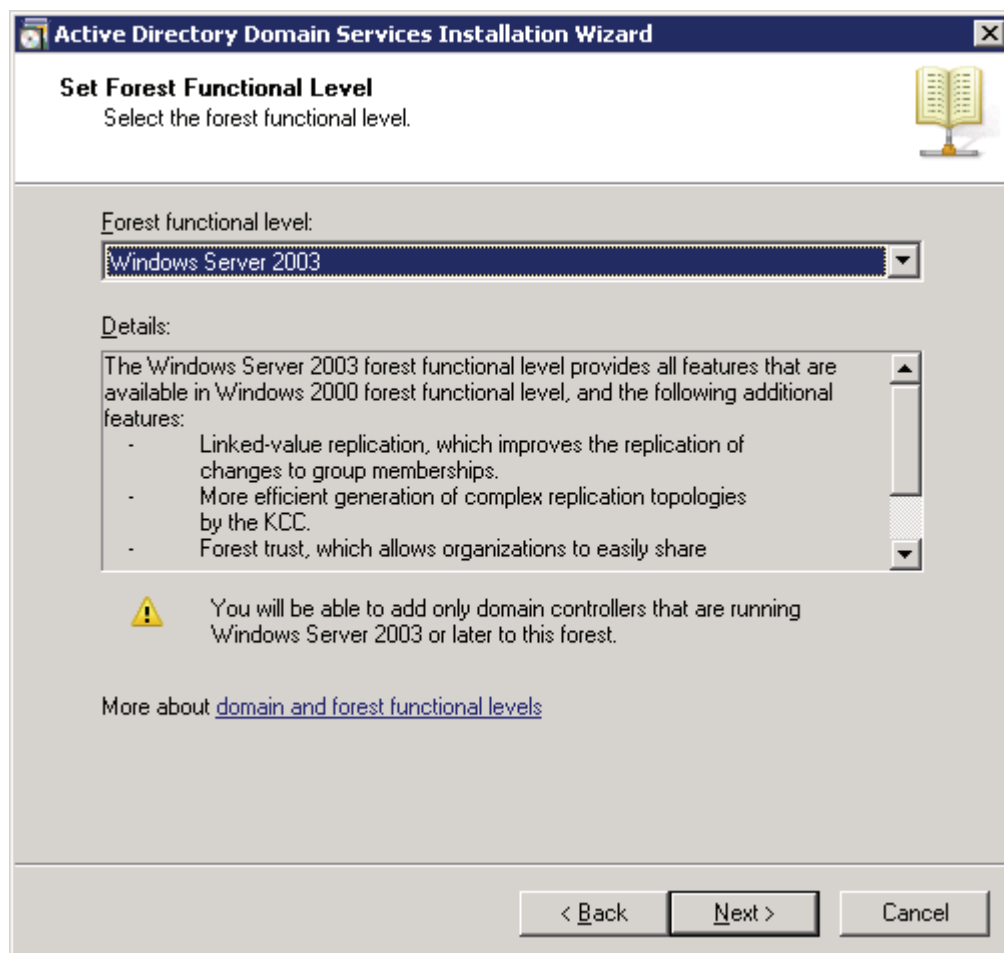
Deretter får vi beskjed om å navngi domenet. Vi har valgt mafoberg.net som nytt domene.



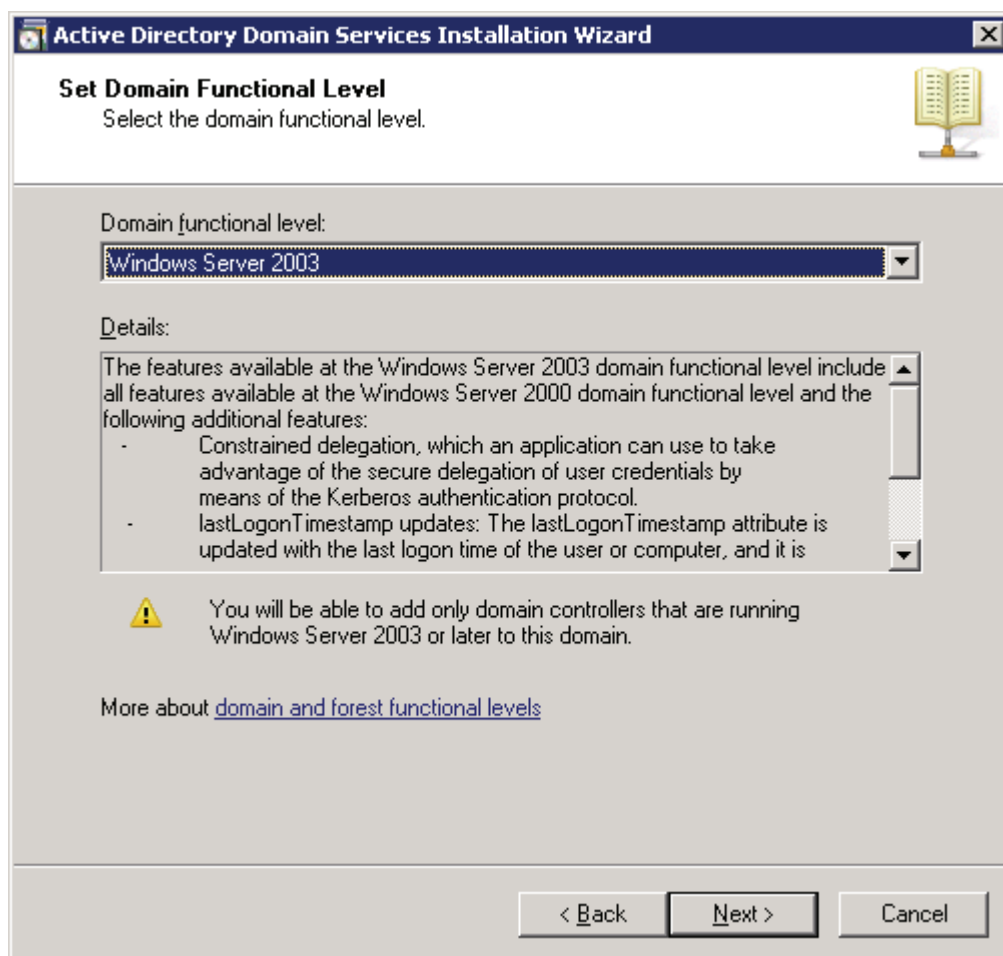
Figur 29 - Nytt domenenaavn

Siden vi ikke tar med alle domenekontrollere med en gang, kan de gamle domenekontrollerne settes i funksjon sammen med de nye. Derfor tar vi med funksjonaliteten i skogen opp til Windows Server 2003 servere. Vi får informasjon om den viktigste funksjonaliteten i skogen vi får med. Deretter følger informasjon for domenekontrollerens funksjonalitet. Denne har vi også satt til nivå for Windows Server 2003. Da har vi funksjoner som på de samme gamle Windows Server 2003 som vi

har i nettverket fra før. Disse kan etterkonfigureres når de gamle serverne settes ut av funksjon eller blir tatt vekk for godt.



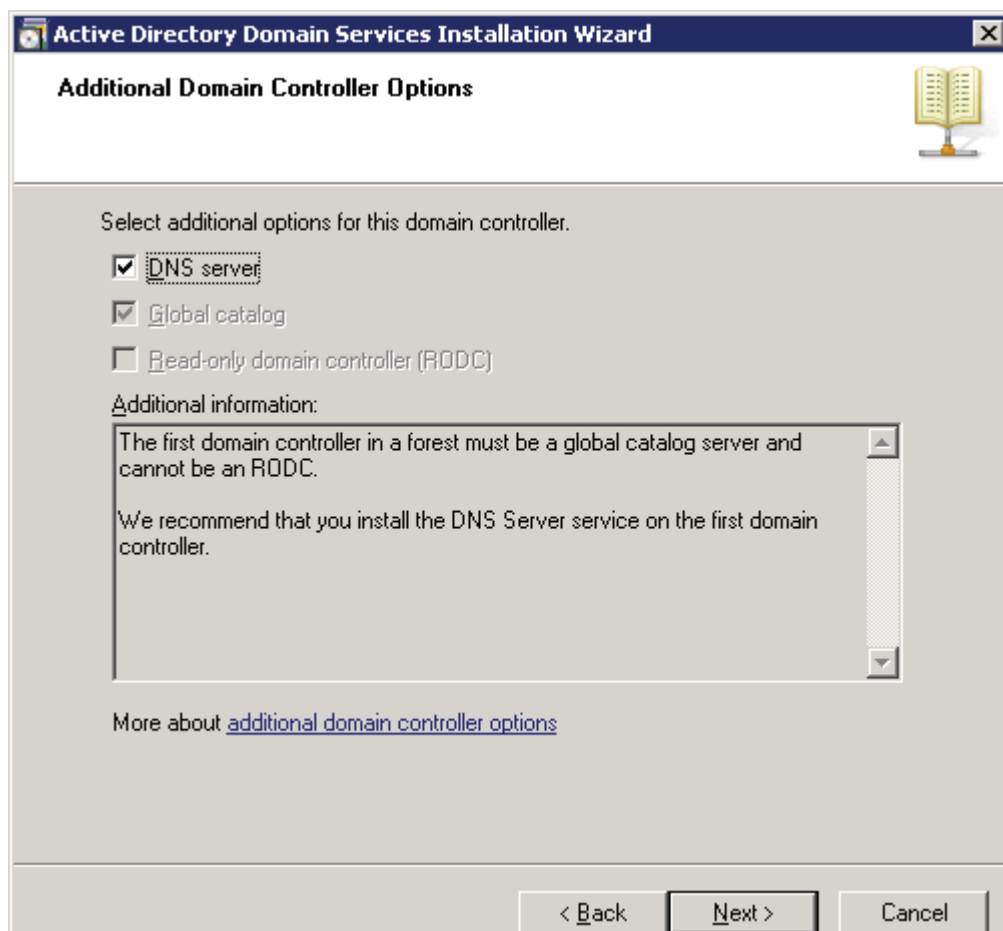
Figur 30 - Skogsfunksjonalitet



Figur 31 – Domenefunksjonalitetsnivå

Neste skritt er å gjennomføre oppsett av DNS-tjenesten sammen med domenekontrolleren. Siden dette er den første domenekontrolleren vi setter opp, kjører vi DNS-tjenesten direkte på denne. Senere kan vi flytte DNS-tjenesten over til en annen maskin om ønskelig.

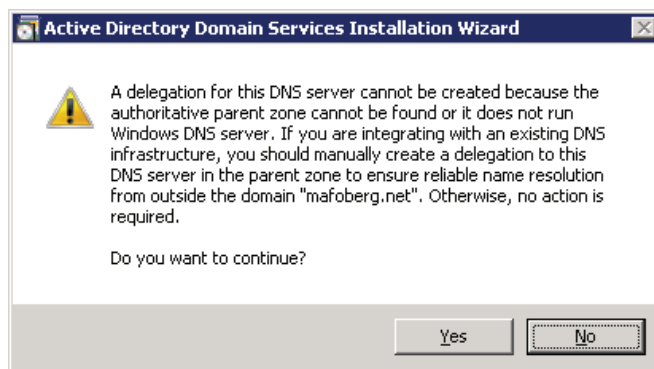
Vi kan ikke installere DNS-tjenesten på en Read-only domenekontroller. Denne kontrolleren installerer vi senere på en ny virtuell maskin.



Figur 32 - DNS-tillegg

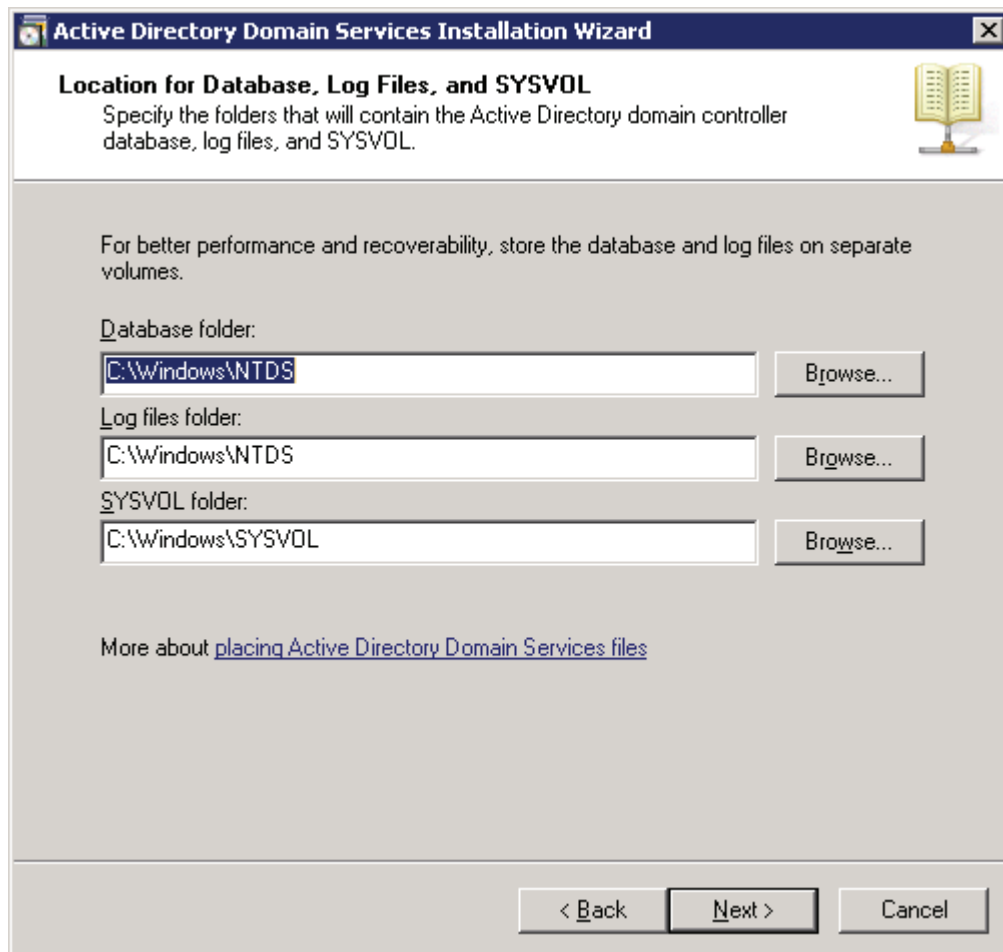
DNS-tjenesten sjekker med en gang om det finnes en delegasjon til denne serveren for domenet. Dette har vi ikke gitt tilgang til enda, men domenet er registrert og satt i verk. Så lenge vi har domenet kan vi bare svare Yes på ønske å fortsette oppsettet.

Vi har registrert domeneoppsett til domeneshop.no som vi styrer selv. Vi har derfor angitt en delegasjon (A-record) bakover til en ipadresse denne serveren skal ha. Til dette må vi sette opp nettverksforbindelsen til at det er denne DNS-tjeneren som skal brukes som standard oppslag. Siden vi nå sitter i et lukket nettverk, vil ikke oppslag mot DNS-serveren kunne gjøres siden brannmur er satt opp for å blokkere all innkomne forespørsler.



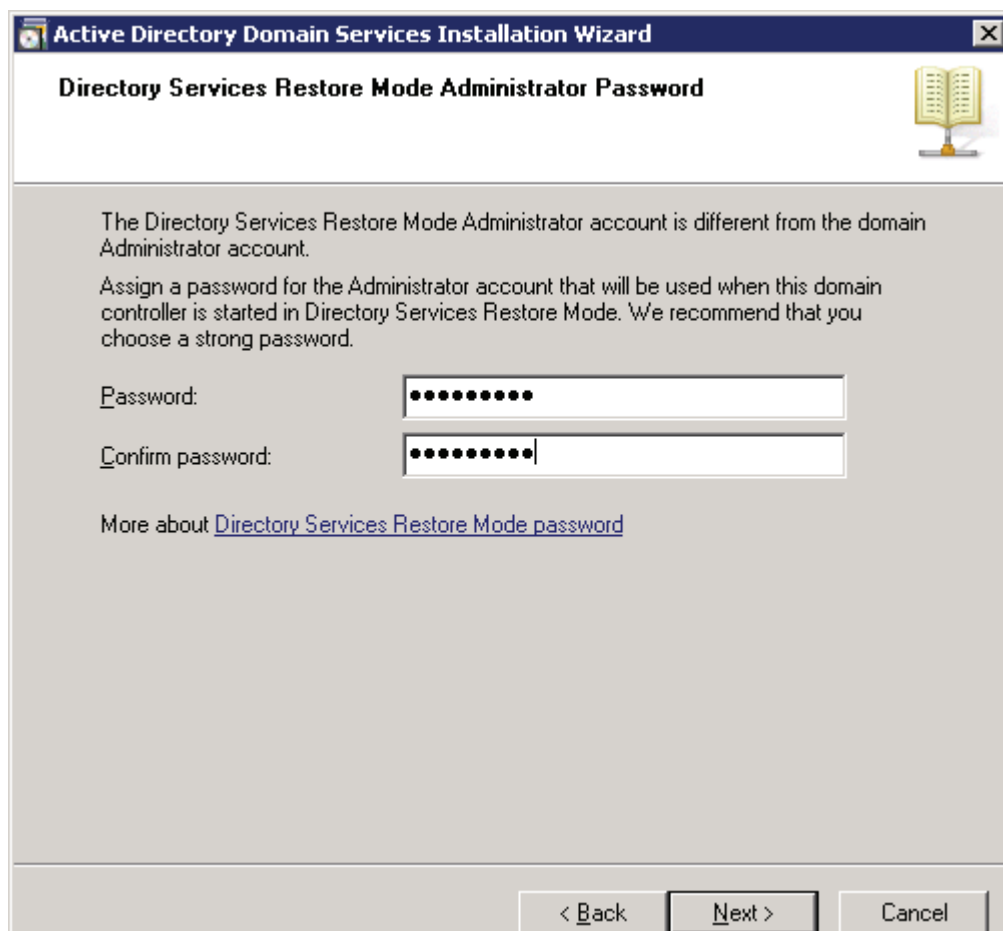
Figur 33 - DNS fortsettelse

Vi oppretter også nye volum og stier for hvor plassering av log og script skal ligge. Vi velger standard plassering da dette kan være ganske hensiktsmessig i forhold til hva som er gjort før. Eventuelle forandringer og nye plasseringer kan forandres på senere.



Figur 34 - Database og script plasseringer

Deretter får vi beskjed om å opprette passord for Remote Services Restore Mode Administrator. Dette må ikke forveksles med den vanlige domeneadministratoren. Denne brukeren har andre tilganger for å gjenopprette tidligere lagringspunkt for domenekontrolleren. Denne startes opp i egen modus.



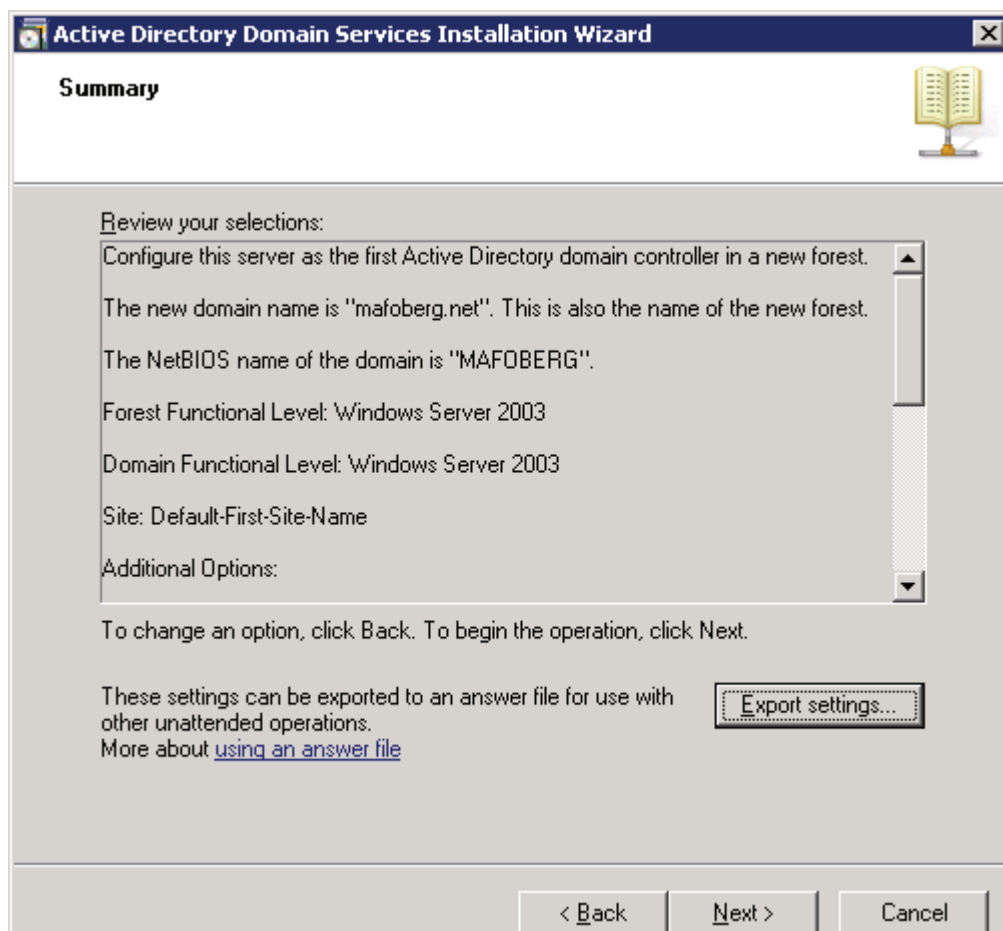
Figur 35 - DS Restore Mode Administrator Password

Etter dette er all konfigurasjon om det nye domenet, skogen og DNS-tjenesten satt opp. All denne informasjonen kan bli lagret for sikkerhets skyld. Skal noe gå galt etter installasjonen i forbindelse med andre ting, kan denne informasjonsfilen brukes til raskt å hente inn installasjonsinformasjon så vi får gjennomført den på nytt, raskt og enkelt. Derfor velger vi å eksportere denne informasjonen til egen fil.

Mulighet for å reinstallere med samme informasjon gjøres i kommandolinje. For eksempel når en ren installasjon av Windows Server er gjort på nytt skal vi installere domenetjenesten på nytt. Vi kjører cmd som er det vanlige kommandogrensesnittet og får opp et konsollvindu. Her kjører vi kommandoen `dcpromo`.

```
dcpromo /answer:adds-svarfil.txt
```

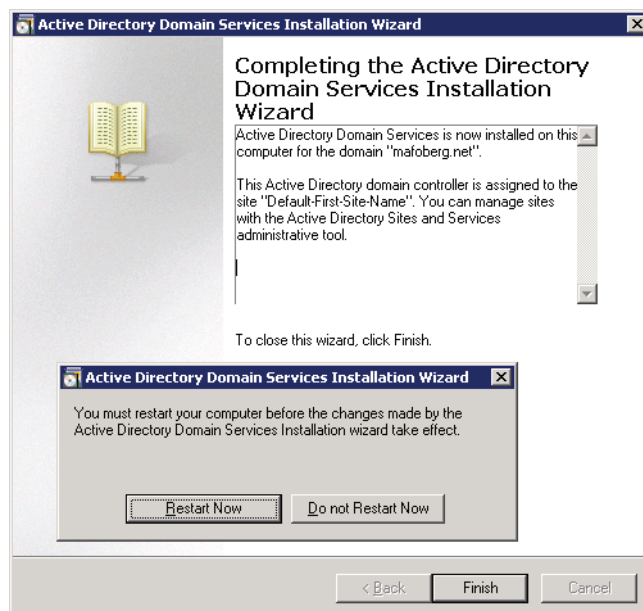
Da kjøres oppsettet automatisk slik som man har klargjort på forhånd.



Figur 37 - Eksportere installasjonsinformasjon

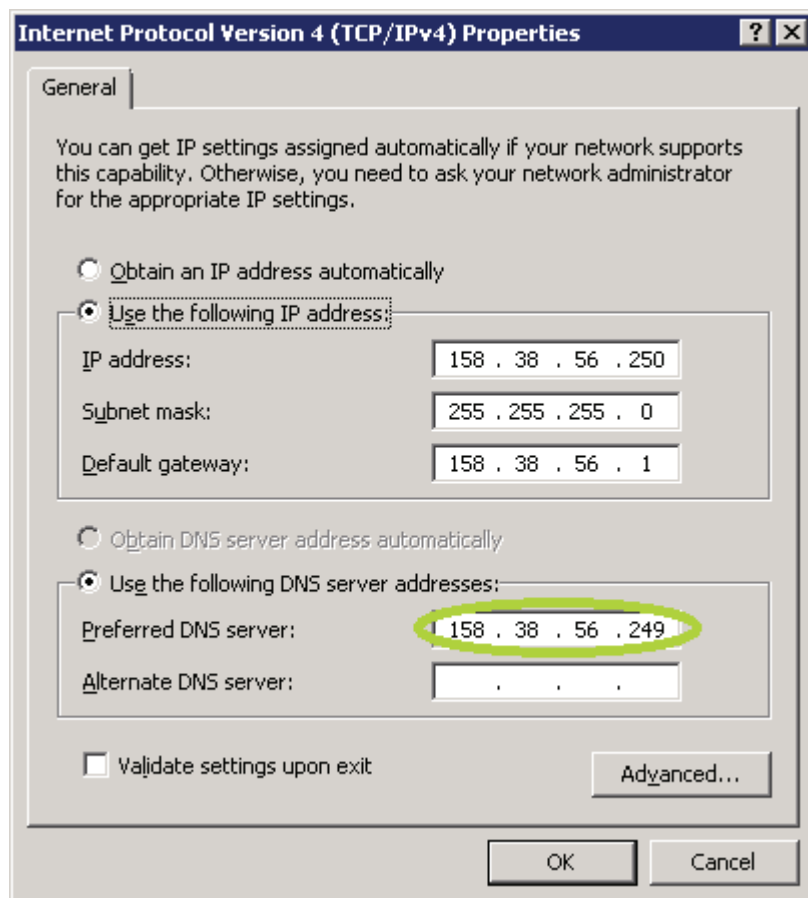
Når oppsettet av domenekontrolleren er gjennomført, bes det om restart for at det hele skal settes i verk. Da vil domeneadministratoren kunne logge inn som administrator i domenet.

For at domenet skal være tilgjengelig i nettverket må DNS-serveren være adressert i nettverkskortinnstillingene.



Figur 36 - Eksporteringsveiviser og restart

Etter at maskinen er restartet, setter vi DNS-adressen i denne maskinen. Dette må også gjøres i alle andre nettverkskorts innstillinger akkurat nå inntil en dhcp-tjeneste³ blir konfigurert til å gi denne adressen ut, eller at det blir satt opp en ny DHCP-tjeneste.



Figur 38 - DNS-oppsett i nettverkskort

³ DHCP er en tjeneste som kjører på en server som sender ut nettverksinnstillinger til klientmaskiner som kobler seg på nettverket, slik at en bruker som flytter seg mellom flere fysiske plasseringer slipper å konfigurere disse innstillingene manuelt.

2.3.4. Installering av Read Only Domain Controller

En Read Only Domain Controller (RODC) er en domenekontroller som fungerer som en kopi av en allerede eksisterende domenekontroller. Men denne domenekontrolleren kan ikke skrives direkte til. Den synkroniseres kun med en domenekontroller som har hovedansvaret for et domene. Vi kan ha mange slike kopier, men vi skal opprette en for å vise hvordan det gjøres, og denne skal fungere som en oppslagsserver når hosten med den domenekontrolleren går ned.

2.3.4.1. Forhåndsoperasjoner før vi kommer i gang.

Vi har opprettet en server med navn "DC-RO" på en host utenfor clusteret med Windows 2008 R2. Denne kan opprettes fra en template om vi har en tilgjengelig. Når serveren er installert, legger vi til oppdateringer som er kommet ut om disse ikke allerede er installert.

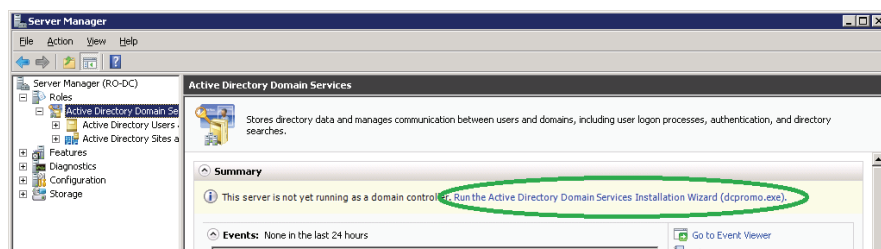
Vi sørger for at maskinen er meldt inn i domenet før vi starter installering. Deretter legger vi til rollene

- Active Directory Domain Services – Selve domenerollen.
- .NET Framework 3.5.1 Features – Tillegg som kreves av de fleste roller.

I en skog med en eller flere domenekontrollere på 2003 servernivå, må man først kjøre "adprep.exe /rodcprep" for å klargjøre 2003-serveren for en Windows 2008server. Eventuelle skjema fra installasjonsmediumet fra 2008 serveren må legges til 2003-serveren.

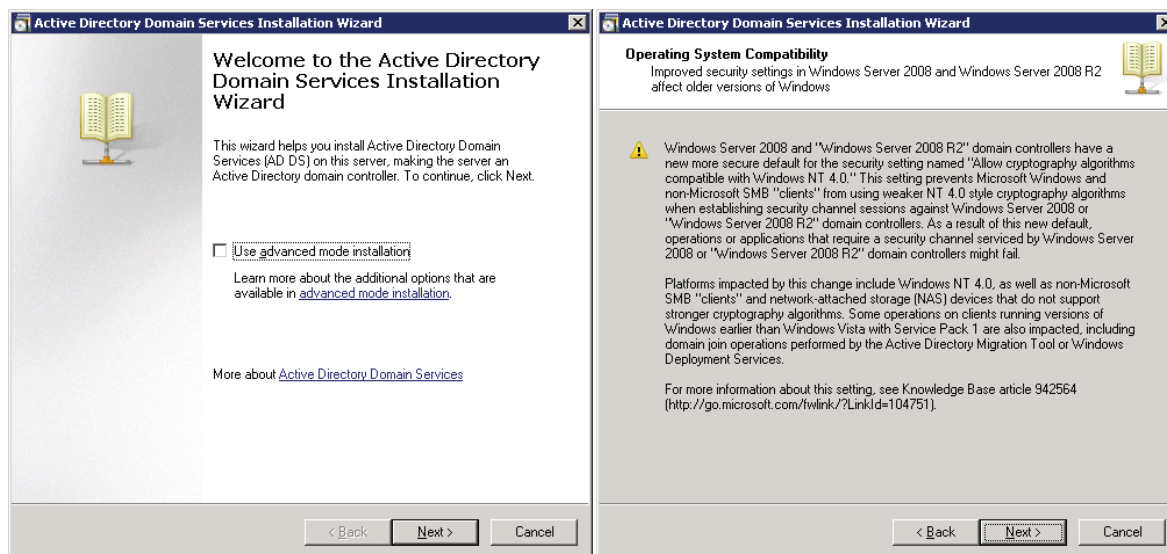
Vi kjører her kun en RODC på domenet hvor vi allerede har en domenekontroller på server 2008.

2.3.4.2. Installering av RODC.

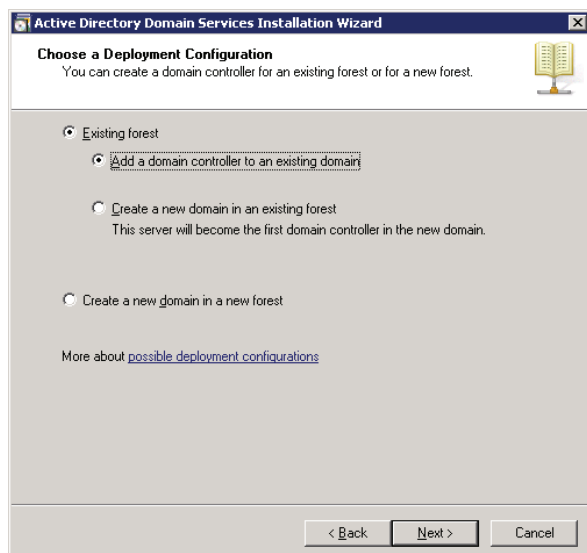


Figur 39 - Installasjon RODC – 1

Vi starter opp med dcpromo.exe som da vi la inn DC første gang, og fortsetter til vi får skogvalg.



Figur 40 - Installasjon RODC – 2

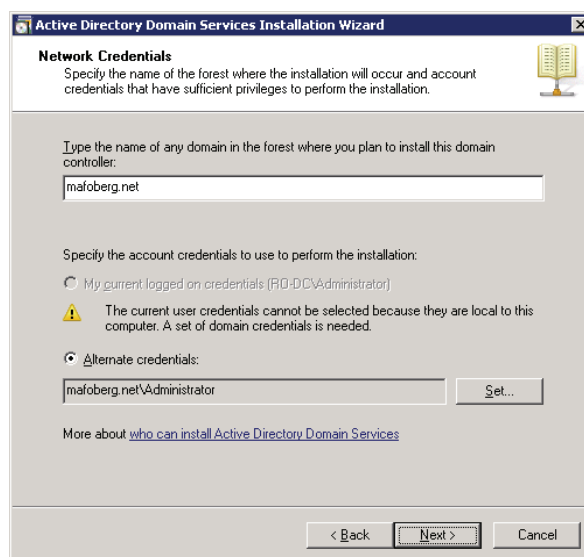


Figur 41 - Installasjon RODC - 3

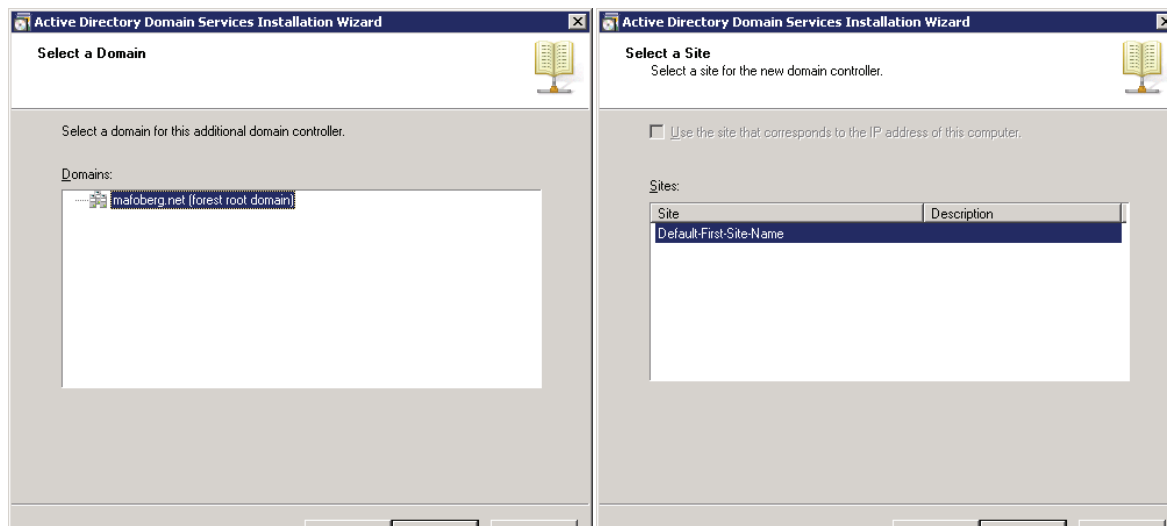
Deretter angir vi domenet som denne serveren skal tilhøre.

Vi må også angi administratorkonto for domenet i *Alternate credentials*. Vi må sørge for at administratorbrukeren er med i gruppen *Enterprise Admins* eller *Enterprise Read-only Domain Controllers*.

Vi har allerede et domene med navnet *mafoberg.net*. Når vi skal legge til en domenecontroller, velger vi å legge den til i skogen som allerede eksisterer med eksisterende domene.

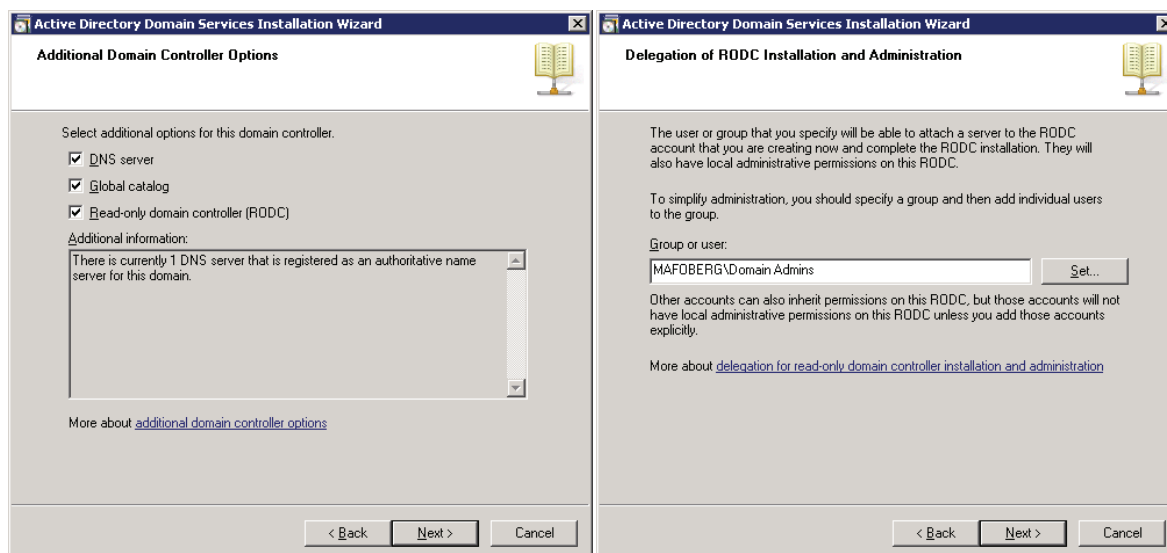


Figur 42 - Installasjon RODC eksisterende domene



Figur 43 - Installasjon RODC - Skog og Site-name

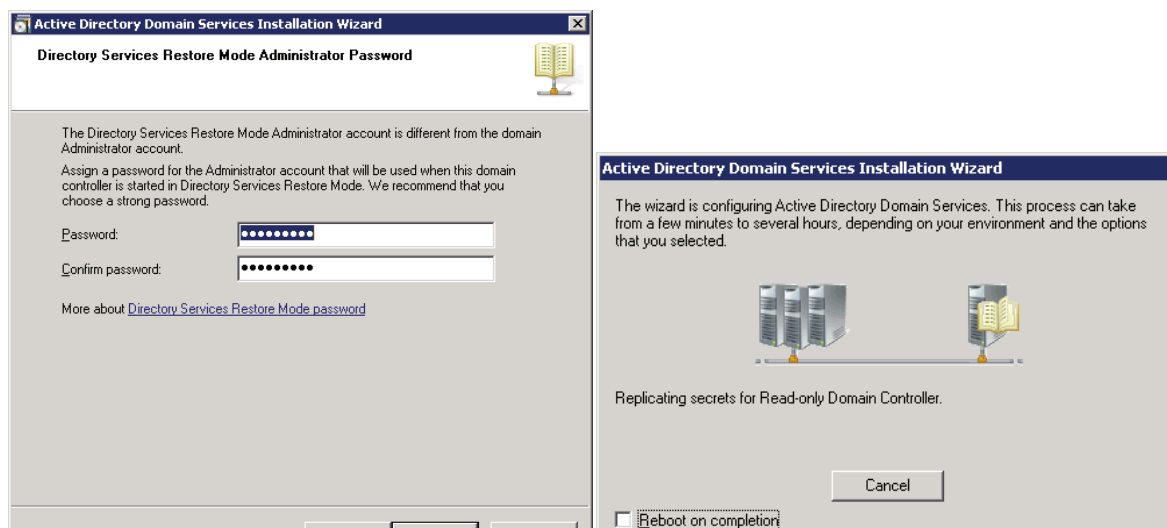
Vi velger så domene som allerede eksisterer til å ta hånd om. Deretter *område* område for domenekontrolleren å befinne seg i.



Figur 44 - Installasjon RODC - valg for RODC

Vi får så velge hva domenekontrolleren skal ha som status. DNS server og global katalogisering skal den ta hånd om. Men her må vi velge *Read-only domain controller (RODC)* i tillegg for at den skal fungere som en read-only fra hoveddomenekontrolleren vår.

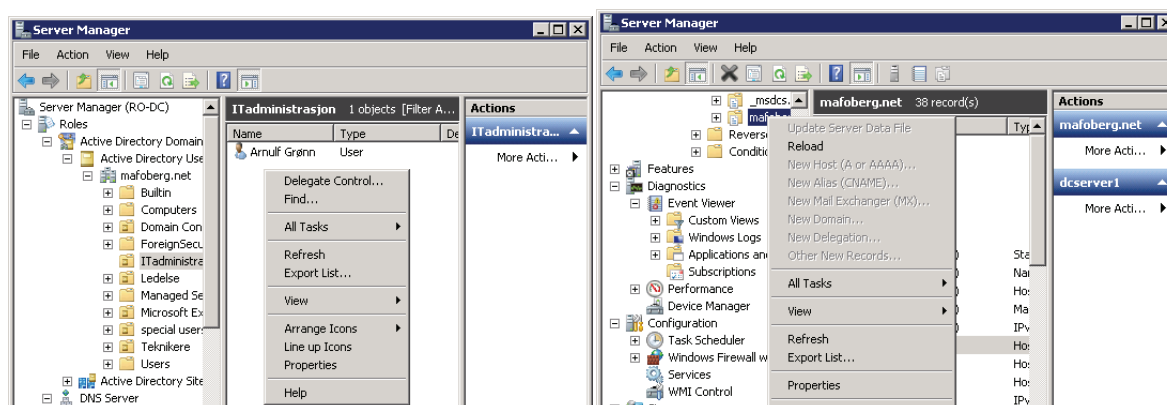
Deretter kan vi velge hvilken gruppe eller bruker som skal kunne administrere vår RODC. Dette kan vi velge ut ifra eksisterende tre.



Figur 45 - Installasjon RODC - Annet

Deretter passord for *Restore Mode*. Som vi har gjort før.

Så starter synkroniseringen mot domenekontrolleren vi allerede har. Etter får vi beskjed om å restarte domenekontrolleren for at den skal tre i kraft.



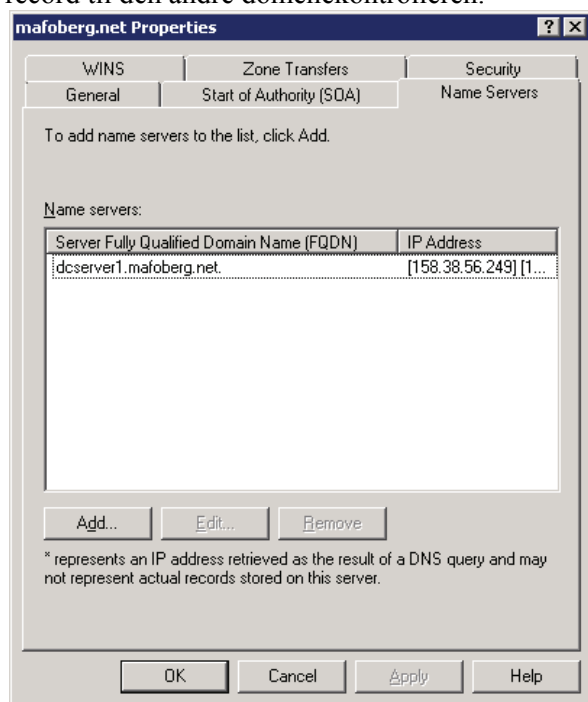
Figur 46 - RODC i funksjon. Ingen opprettelsestilgang.

I domenekontrollerbehandlingen ser vi at vi ikke har tilgang til å opprette nye objekter. Heller ikke under DNS-oppsettet.

2.3.4.3. DNSoppslag

Det er viktig å sette opp DNS-oppslag for domenekontrolleren. Hvis ikke vil vi ikke kunne nå frem til denne om DNS-tjeneren skulle gå ned sammen med den.

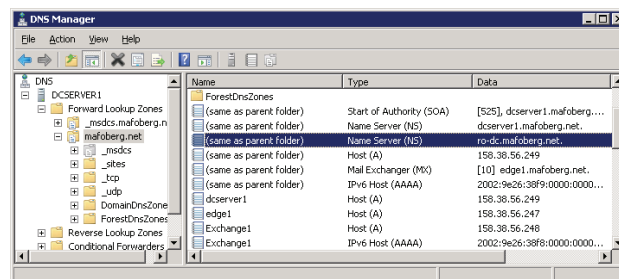
På domenekontrolleren vår går vi på *Forward lookup zones* og vårt domene. Vi har der en NS-record som heter (*same as parent folder*) som peker til den første dnsserveren. Her må vi legge til en NS-record til den andre domenekontrolleren.



Figur 47 - DNS oppslag utvidelse

DNS-oppslag peker her mot den første serveren. Vi må legge den andre dnsserveren her ved å trykke *Add*.

Vi får da beskjed om å oppgi fullt DNS-navn til serveren og deretter klikke *Resolve* for å finne serveren. Dermed finner den nettverksadressen selv.



Figur 48 - DNS oppslag laget

Når DNS oppslaget er laget, vil det ta litt tid før eventuelle cachinger vil bli oppdatert. Dette gjelder klienter og andre servere.

Når dette er gjort, vil nedetid av hoeddomenekontrolleren ikke være så alt for risikabel lengere. Man vil da fortsatt kunne logge inn i domenet. Dette er viktig når man har maskiner på cluster. Man kan da også sette domenekontrolleren på clusteret.

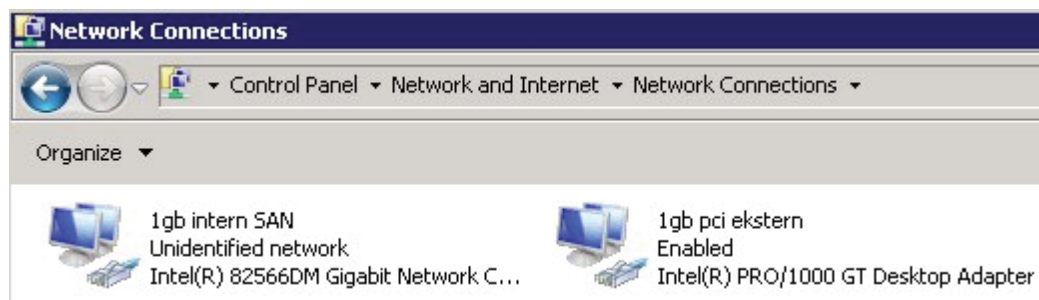
2.4 Windows Failover Clustering og Hyper-v Failover

2.4.1 Hva er Windows Failover Clustering?

Windows Failover Clustering er et Microsoftprodukt som bygger på Microsoft Cluster Server fra 2003. Det følger med som en standard feature i Windows Server, men må installeres og aktiveres. Den gjør det mulig for en serie av uavhengige servere kan dele et høyhastighets nettverksbasert lagringsmedium og på den måten kunne dele på arbeidsoppgaver som krever disklagring.

Failover Clustering har fått betydelige forbedringer i Windows 2008r2-versjonen, der det er lagt spesielt vekt på ytelsesforbedringer samt muligheten til å gjøre flere enn én virtuell maskin høyt tilgjengelig per fysiske Hyper-V host. Mens Failover Clustering også kan brukes til andre lagringsformål, er det spesielt denne muligheten til å kjøre flere virtuelle operativsystem per Hyper-V host vi skal benytte oss av for å gjøre Exchange Mail Server og ADDS Server alltid tilgjengelig, uavhengig om én host går ned.

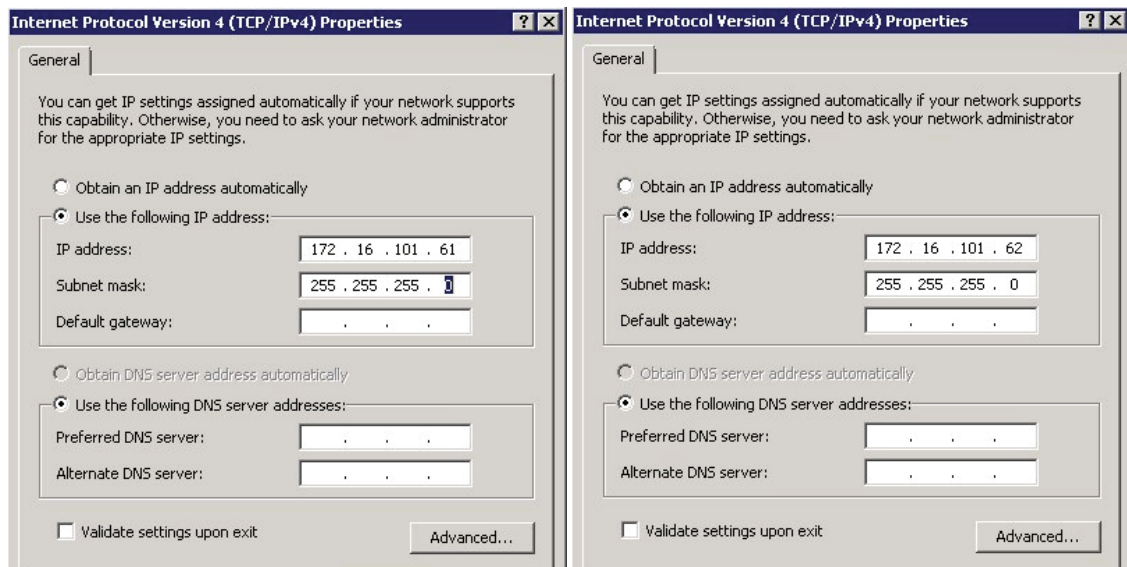
2.4.2 Forberedelse før installasjon



Figur 49 - Oversikt over nettverksadapere. Failover Clustering krever minimum ett dedikert adapter til lagringsmediumet

Det er viktig å legge merke til at Failover Clustering for hver host krever en dedikert tilkobling mot det felles lagringsmediet, og helst én nettverksadapter per logiske lagringsenhet på nettverket. Vi skal se at Windows Clustering krever to logiske lagringsenheter for kjøring av virtuelle maskiner, nemlig en heartbeat-forbindelse i tillegg til lagringsplass for de virtuelle maskinene. En heartbeatforbindelse (også kalt quorum) er en konfigurasjonsdatabase som holder oden på hvilke hostmaskiner (eller noder) som har kontrollen over en spesifikk høytliggjengelig virtuell maskin. Dette er nødvendig fordi ved en failoversituasjon hvor alle hoster mister sin nettverksforbindelse, ville de uten tilgang til en slik konfigurasjonsbase prøvd å gjenoppta kjøring av alle høytliggjengelige virtuelle maskiner. Dette unngår vi altså når vi har en slik heartbeat, hvor alle hosts kontinuerlig sjekker opp konfigurasjonsdatabasen for å finne ut hvilke andre hostmaskiner som er oppe eller har gått ned. Det finnes også flere typer forskjellige konfigurasjonsdatabaser som har forskjellige krav til antall noder som må være oppe til en hver tid. For vårt bruk med to noder i clusteret, er det bare mulig å velge "Node and disk majority", hvor konfigurasjonsdatabasen dobbeltlagres på hver node i tillegg til det delte lagringsmediet.

Her ser vi hver nodes nettverkstilkobling mot det delte lagringsmediumet. Begge har altså et dedikert gigabit nettverkskort og hver har en egen IP gitt av nettverksadministratoren. Lagringsløsningen valgt er et Storage Attached Network (SAN) og benytter seg av iscsi-protokollen for overføring av data.

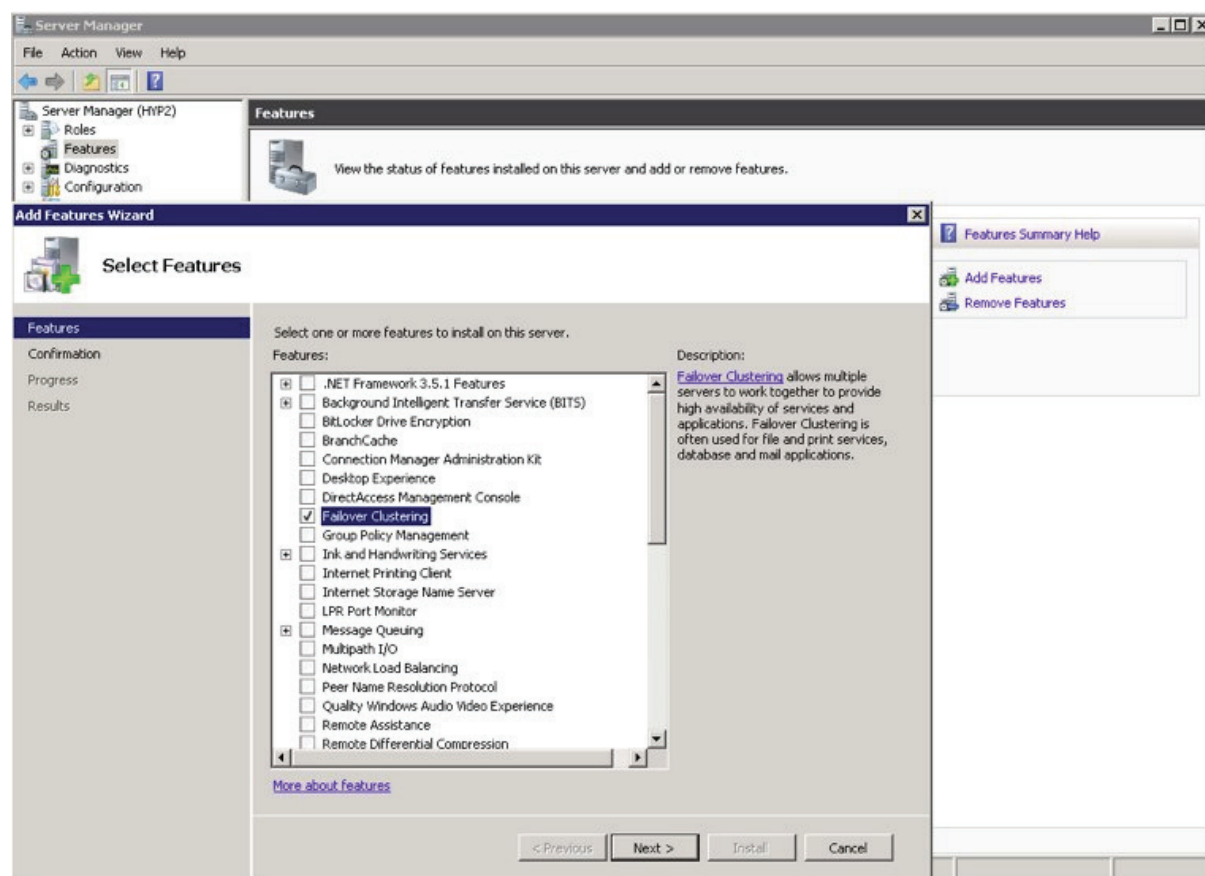


Figur 50 - SAN-tilkøling for hyp-1.mafoberg.net (venstre) og hyp-2.mafoberg.net (høyre)

Det er verdt å merke seg at disse nettverksadapterne bare skal kommunisere internt med lagringsmediet og hverandre. Det er derfor ikke satt noen standard gateway på disse nettverksadapterne. Disse konfigurasjonsinnstillingene blir som regel gitt av nettverksansvarlig.

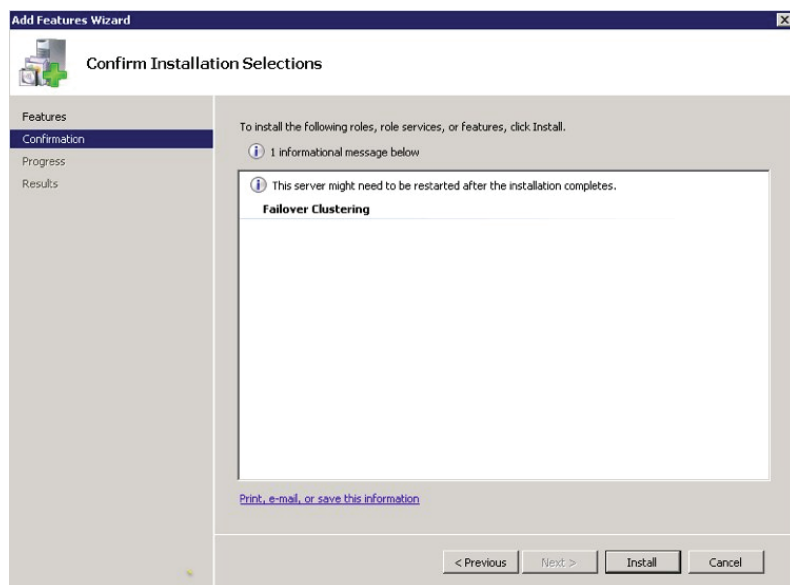
2.4.3 Clusterinstallasjon

Installasjonen av Windows Failover Cluster er veldig rett fram, og gjøres fra Windows Server Manager, under "Features", deretter "Add features" og "Failover Clustering".



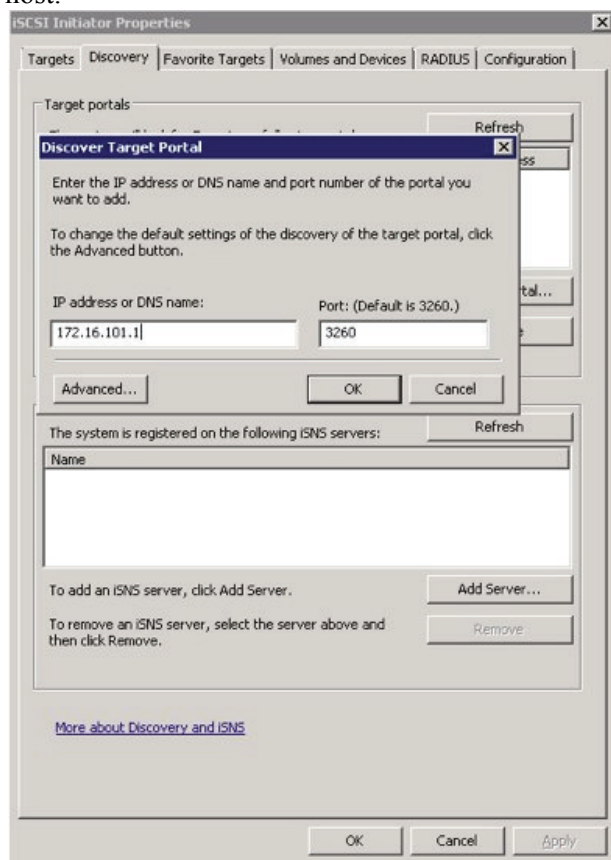
Figur 51 - Installasjon av Failover Cluster-rolle

Her må vi også være klar over at serveren kan måtte restartes etter installasjon. Samme prosedyre må så gjennomføres på de resterende Hyper-V maskinene.



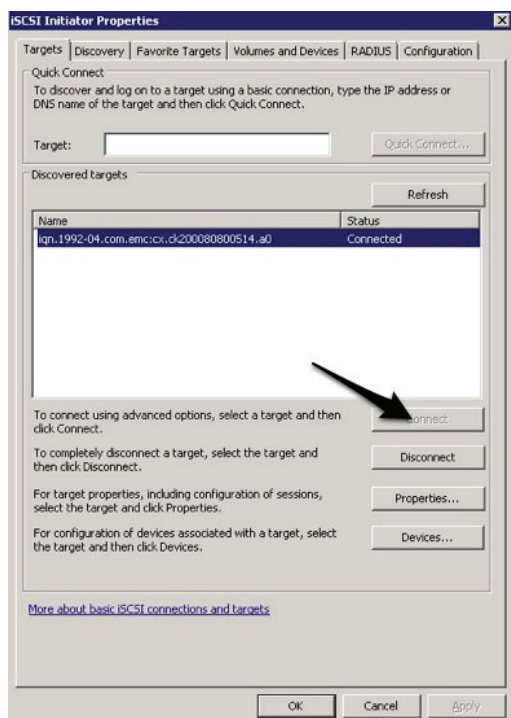
Figur 52 - Konfirmering av installasjon. Dette må gjennomføres på hver server(node) i clusteret

Når Windows Clustering er installert, må også diskene kobles til og linkes opp mot hver Hyper-V host.



Figur 53 - Oppsett av delt lagringsmedium

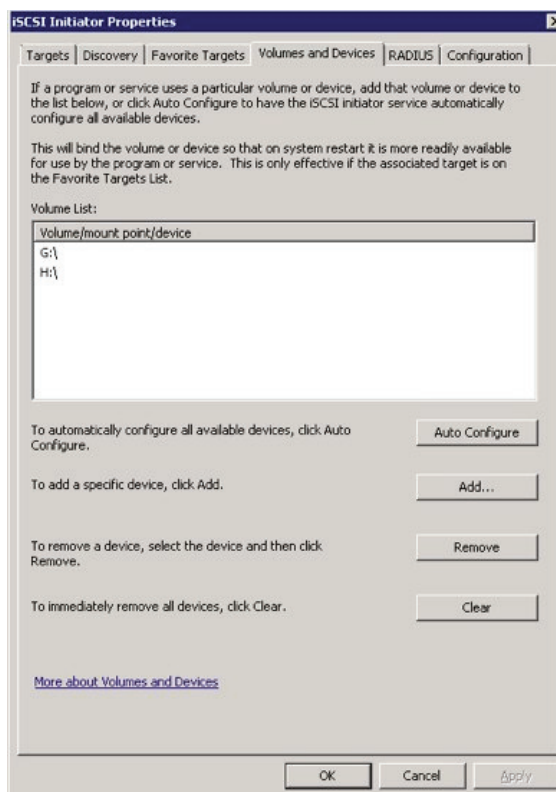
Her ser vi programmet "iSCSI Initiator" som brukes for å koble opp våre SAN-disker. Vi har tidligere satt opp en nettverksadapter på hver host til å bruke adressene i området 172.16.101/24. Under Fanen "Discovery" og deretter "add portal", kobler vi til nettverksressursen til en IP-adresse gitt av oss av nettverksadministrator. Da denne adressen er i samme subnett, brukes nettverksadapteret automatisk for all kommunikasjon mot SANet.



Figur 54 - Koble til lagringsmediumets logiske disk

Under fanen "Volumes and Devices" har vi to partisjoner tilgjengelig. Disse har vi seinere valgt å navngi som "g:" og "h:" og vil da dukke opp med stasjonsbokstaver i stedet for lang bokstavrekke, som signaliserer den fysiske plasseringen til volumet på disken.

På fanen "Targets" vil nå en eller flere måldisker være tilgjengelig. Første gang kobles de ikke til automatisk, slik at vi her må trykke "Connect" før vi går videre.



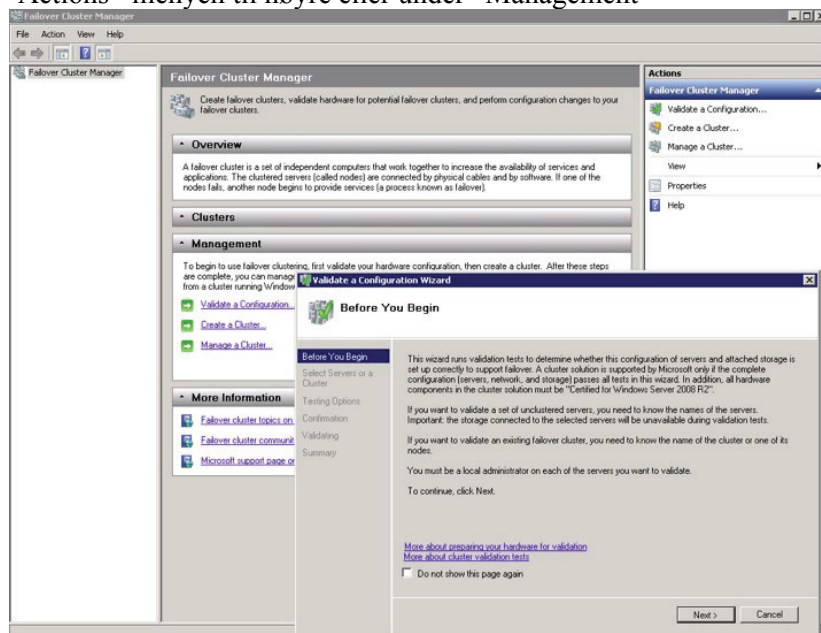
Figur 55 - Koble til volumer på nettverksdisk

en

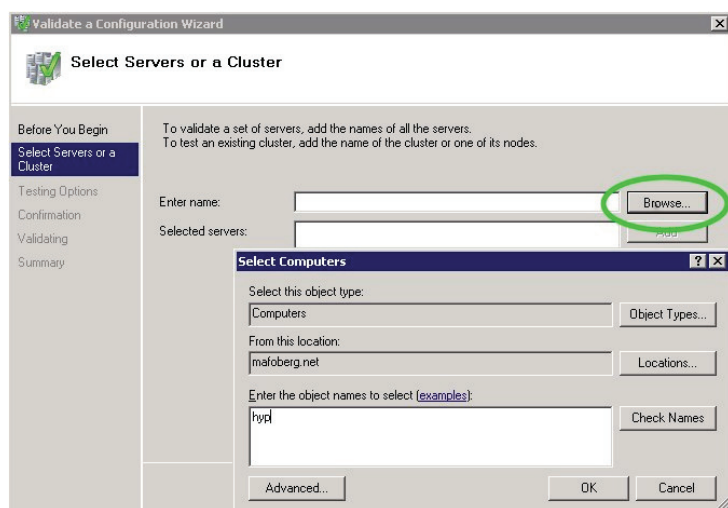
2.4.4 Validering av Clusteret

Etter diskene også er lagt til på de resterende hyper-V hostene, må vi validere at clusteret er satt opp korrekt, samt at det støtter alle nødvendigheter for at clusteret skal fungere med Hyper-V. Det er for eksempel slik at Hyper-V krever SCSI-3 Persistent Reservation⁴, noe som ofte ikke støttes av de billigste SAN-løsningene.

Fra den nylig installerte Failover Cluster Manager, velger vi "Validate a Configuration" fra enten "Actions"-menyen til høyre eller under "Management"



Figur 56 - Oppstart av cluster-validering

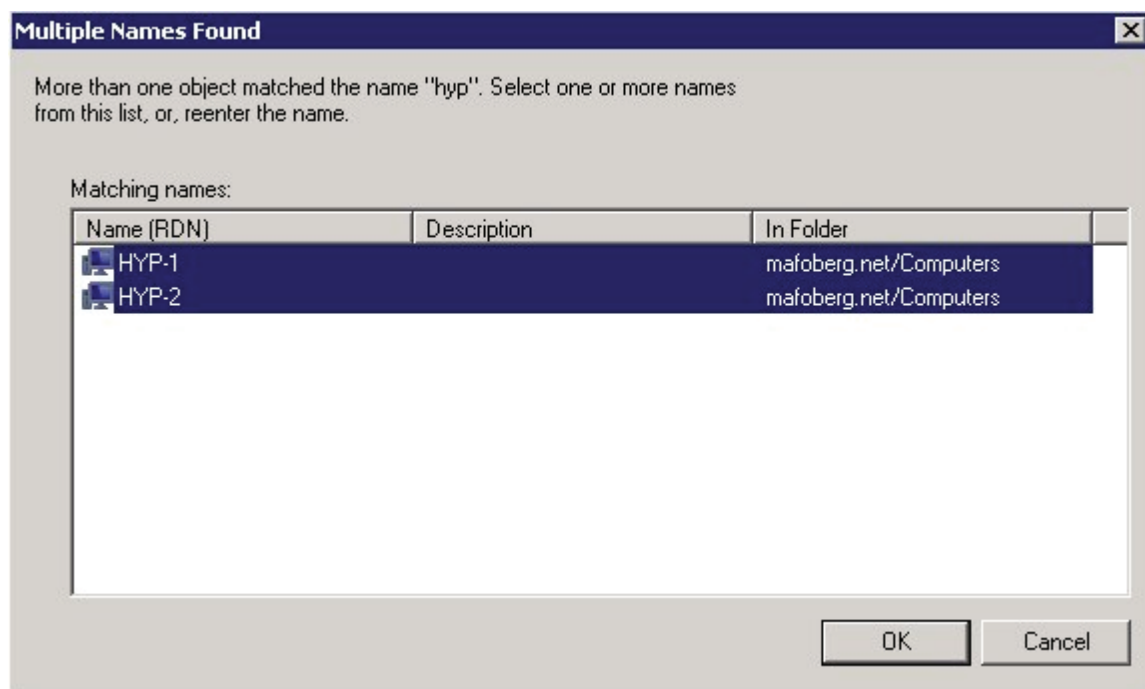


Figur 57 - Innlegging av clusternoder

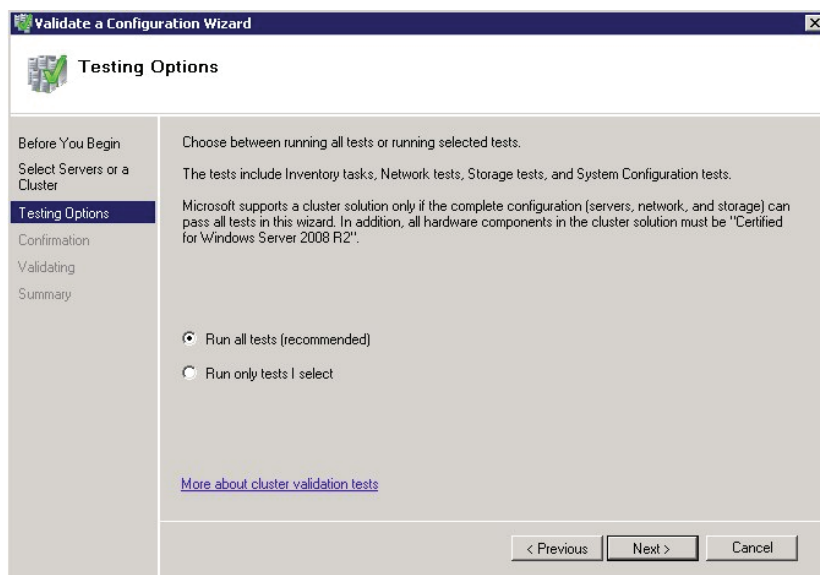
Her kommer vi til et punkt hvor vi må velge alle noder i clusteret. Vi skriver bare inn "hyp" her ettersom vi har navngitt Hyper-H hostene slik at de alle starter med "hyp".

⁴ SCSI-3 Persistent Reservation er en utvidelse av SCSI-protokollen som ikke er bakoverkompatibel med gamle protokoller. Det gir en såkalt initiator –dvs serverene - større tilgang til måldisken på lagringsmediet

Her vil vi få ei liste over alle servere som kan kjøre Hyper-V og som matcher filteret innskrevet i forrige steg. Denne lista over datamaskiner hentes ut fra domenekontrollerene på domenet, som alltid vil ha oversikt over både datamaskiner og brukere som eksisterer.



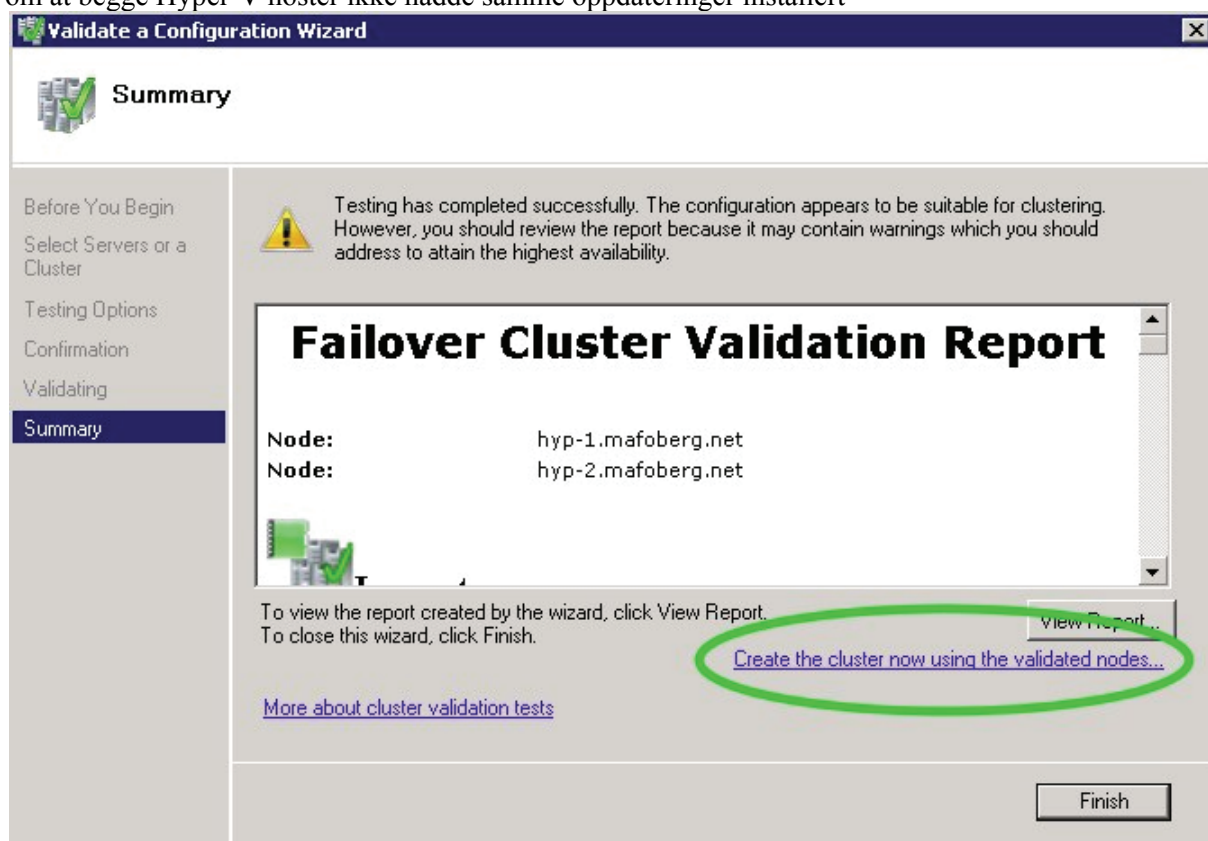
Figur 58 - Ved hjelp av domenekontroller, finner validatoren alle Hyper-V-maskiner i nettet



Figur 59 - Kjøring av clustertester

Her anbefales det å kjøre alle tester, for å forsikre oss om at det ikke er noen feil. Dette vil i tillegg til å validere selve clusterkonfigurasjonen, sjekke om hver node har oppdatert programvare og samme konfigurasjon.

Vi ser at clusteret vårt er korrekt satt opp, men at validatoren fant noen feil. I vårt tilfelle handlet det om at begge Hyper-V hoster ikke hadde samme oppdateringer installert



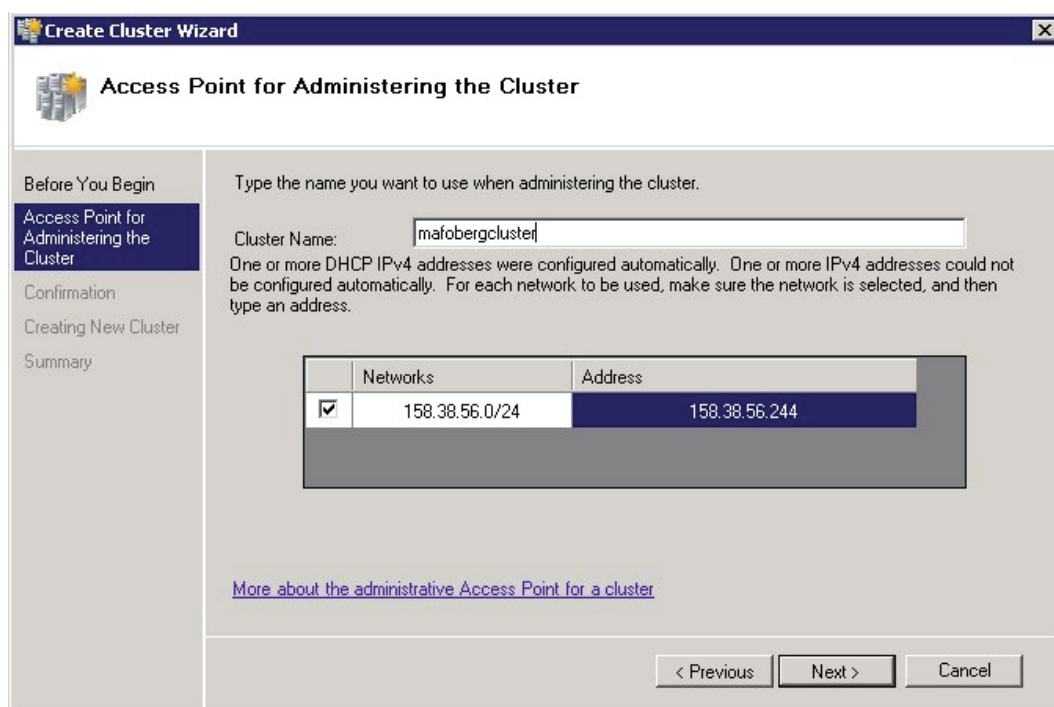
Figur 60 - Valideringsrapport

Etter installasjon av alle oppdateringer og en ny gjennomkjøring av denne validatoren, vil feilmeldingen forsvinne fra rapporten. Det er likevel ikke slik at clusternodene må ha alle oppdateringer installert for at vi skal få lov til å fortsette oppsettet av clusteret; windows vil likevel tillate dette.

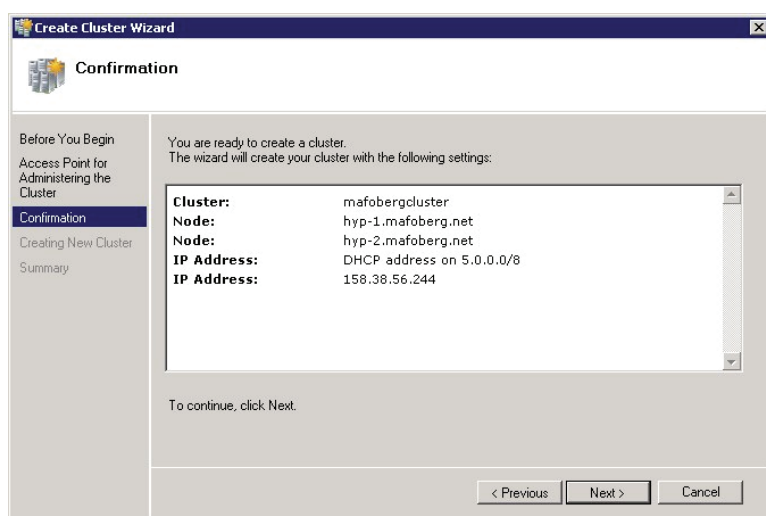
2.4.5 Opprettelse av Clusteret

Når vi har satt opp innstillingene som kreves, og validert clusteret, valgte vi i forrige steg å følge lenka "Create the cluster now using the validated nodes". Vi kunne også valgt "Create a cluster" direkte fra Failover Cluster Manager, men da måtte vi valgt nodene som skal være med i clusteret på nytt.

Her ser vi at at clusteradministrasjonen må gis en egen ip-adresse. Ideelt burde en dedikert nettverksadapter velges til administrasjonen, men det fungerer fint å bruke det adapteret som kobles mot nettverket også. Clusternavnet vil bli en del av domenet, slik at det ville kunne aksessereres ved dnsnavnet mafbörgcluster.mafbörg.net.



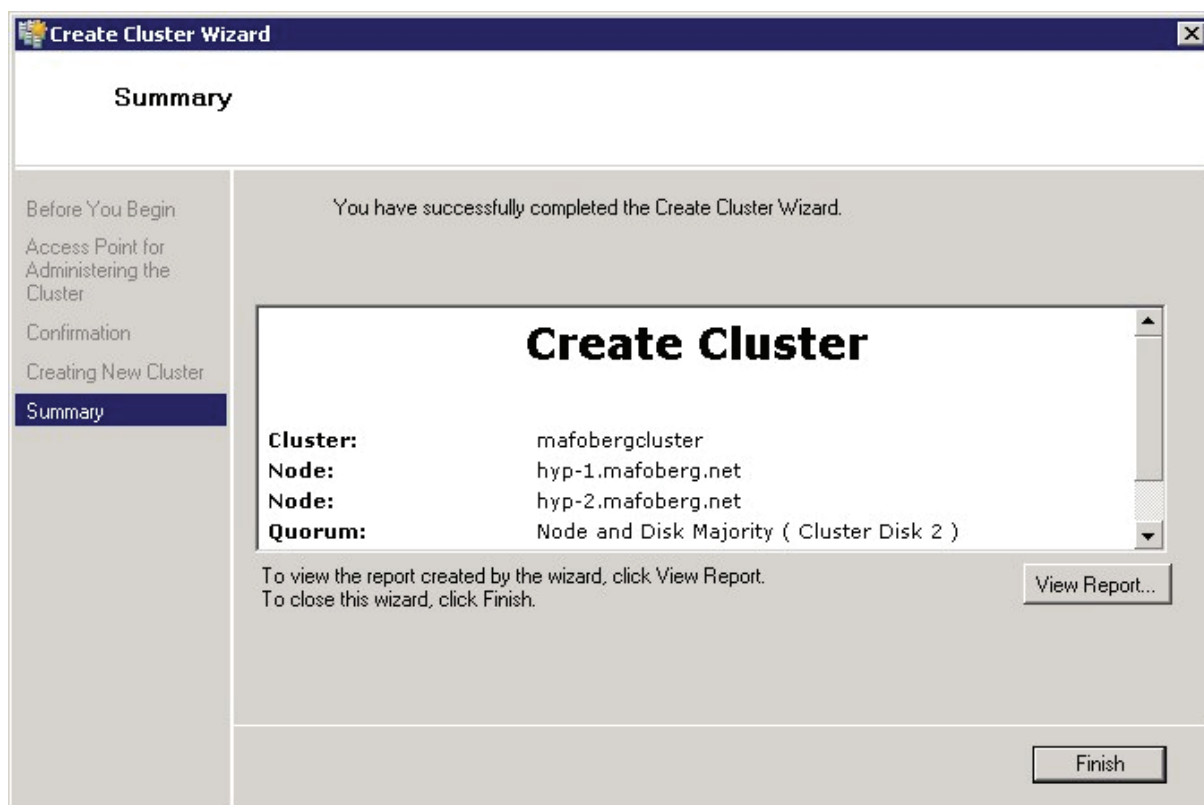
Figur 61 - Clusteret vil få sin egen administrasjonsadresse



Figur 62 - Clusteret er klart til å opprettes

Clusteret er klart til å opprettes, men vi ser også at vi har fått en ekstra ip-adresse i området 5.0.0.0/8 . Dette skyldes et program for fjernadministrasjon som vi tidligere har brukt. I et enterprise-nettverk, anbefales det å fjerne dette ekstra IP-nettet, men for vårt testformål gjør det ingen skade.

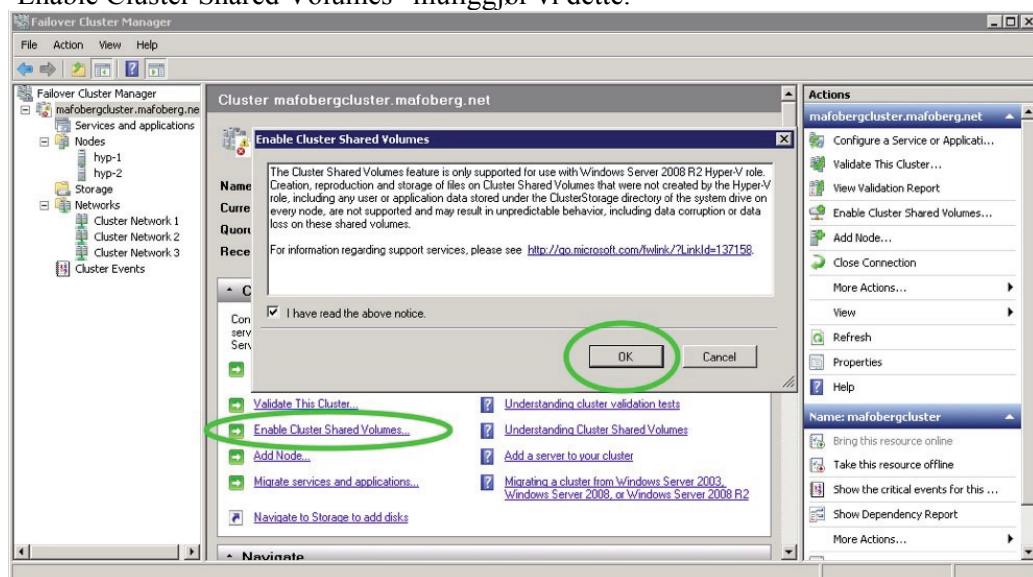
Vi ser her at clusteret ble opprettet uten problemer.



Figur 63 - Cluster opprettet uten feil

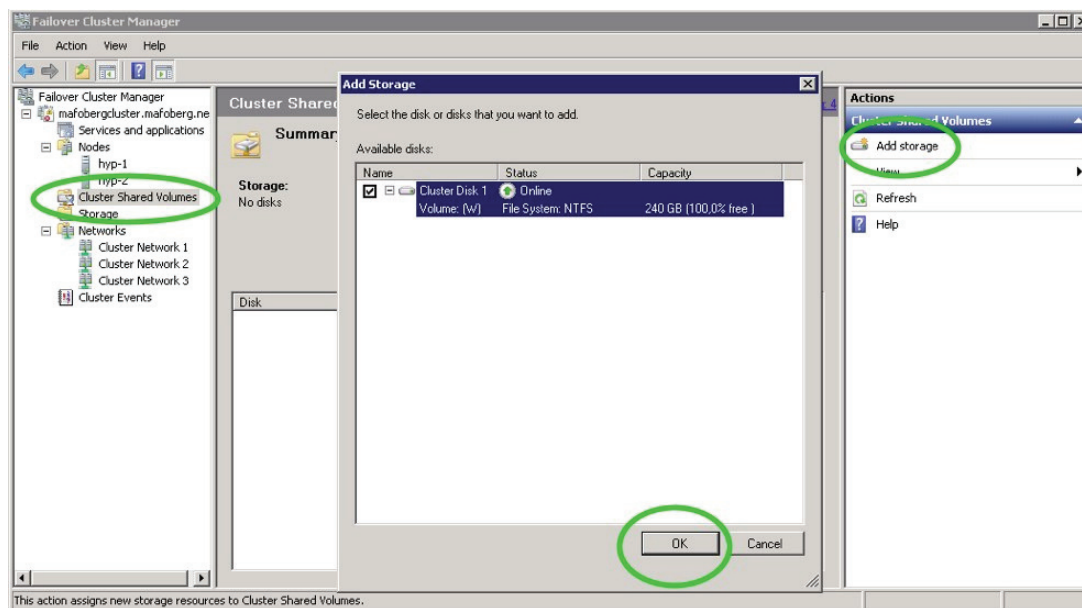
2.4.5.1 Aktivere delte volum

Clusteret er opprettet og begge Hyper-hoster er lagt til som noder i clusteret. Vi har oppretta forbindelse til SAN-diskene og validert clusteret. Men de virtuelle maskinene til nodene kjører fortsatt på lokale disker. Nå må de få tilgang til clusterlagringa. Gjennom Failover Cluster Manager og "Enable Cluster Shared Volumes" muliggjør vi dette.



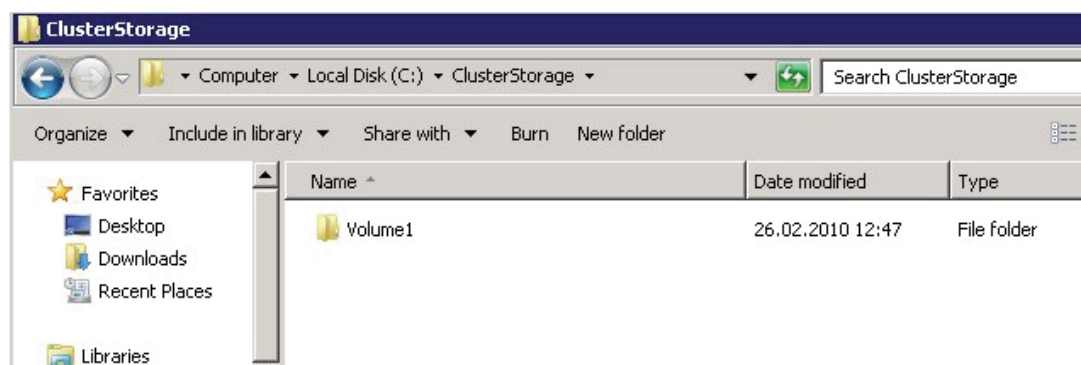
Figur 64 - Aktivere delte volum

Vi velger så ”Add storage” under det nye menypunktet ”Cluster Shared Volumes”. Bare én disk vil dukke opp her, den mindre vil automatisk bli tatt i bruk som en heartbeat-forbindelse.



Figur 65 - Legge til nettværksdisker i clusteret

Ser vi på Hyper-V-nodene nå, vil ei mappe kalt ClusterStorage opprettes automatisk. Det er her de virtuelle maskinene nå må flyttes eller opprettes for å bli ”Highly available”.



Figur 66 – Visning av Clusterlagringsmappe. Denne mappa vil være synlig på alle noder i clusteret.

2.4.6 Flytte over maskiner til clusteret

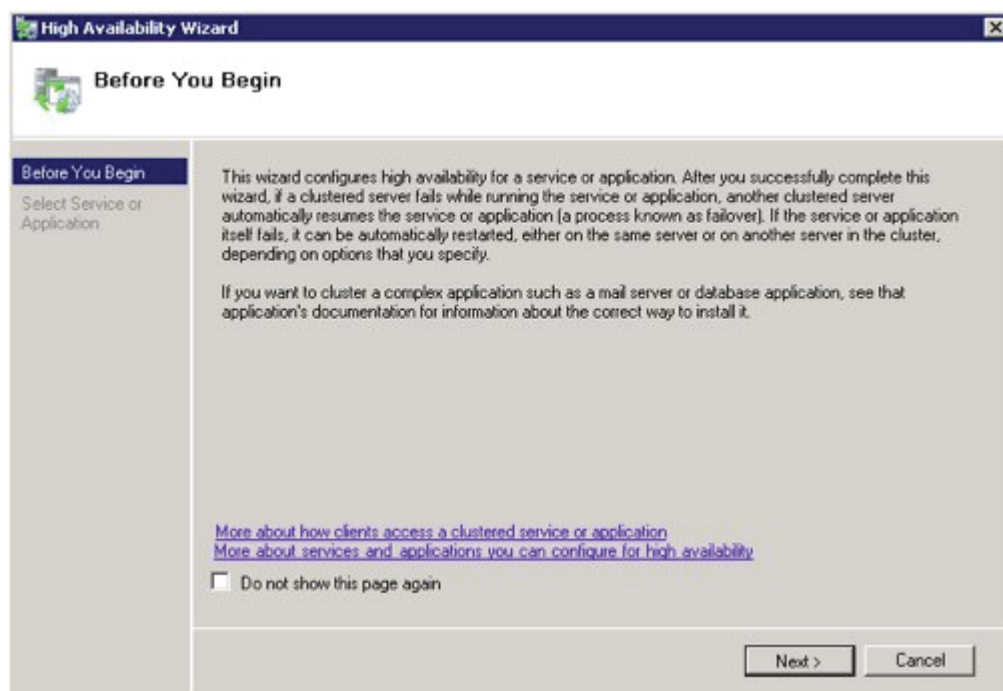
Å flytte over virtuelle maskiner til clusteret er i utgangspunktet enkelt, men må gjøres fra Failover Cluster Manager, da Hyper-V manager ikke støtter flytting mellom noder ellers.



For å starte en flytteprosessen, høyreklikker vi services and applications i Failover Cluster Manager og velger Configure a Service or application.

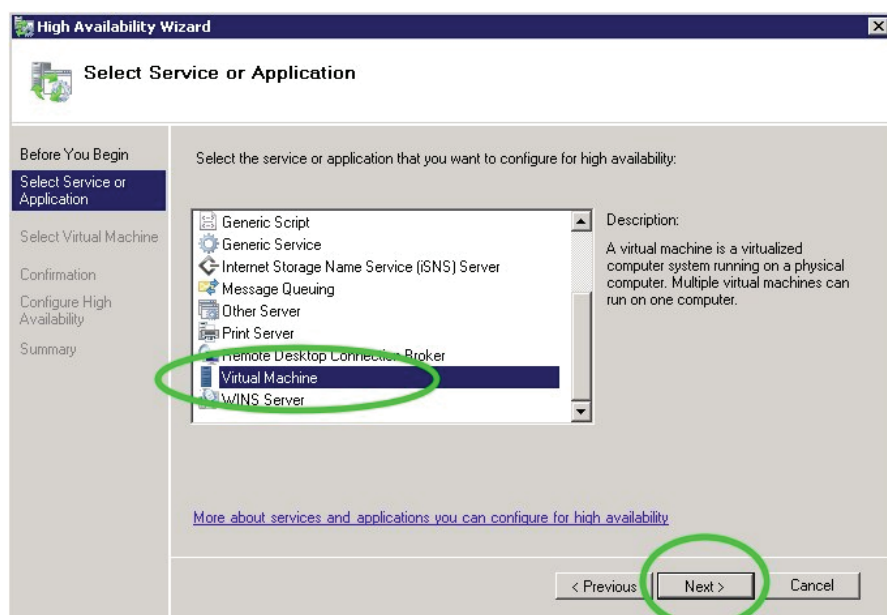
Figur 67 - Flytte maskiner over i clusteret

Her får vi beskjed om hvordan en highly available applikasjon fungerer i praksis. Når en node går ned, vil en annen automatisk ta over. Dette går altså bare om de har delt lagring, og det er derfor vi må flytte vekk de virtuelle maskinene fra det lokale lagringsmediumet.



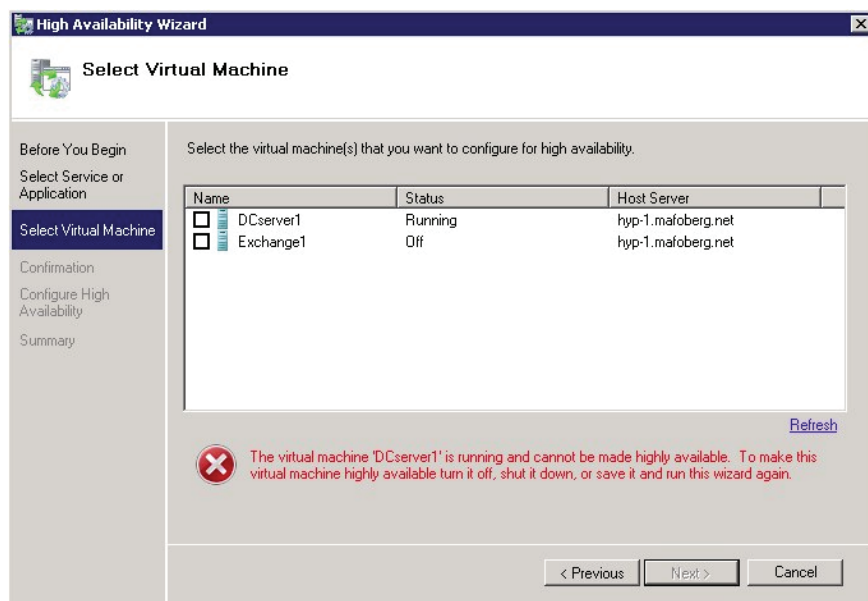
Figur 68 - Informasjon om high availability i clustersammenheng

Her er et flere forskjellige tjenester som kan gjøres høyt tilgjengelig; vi velger ”Virtual Machine”.



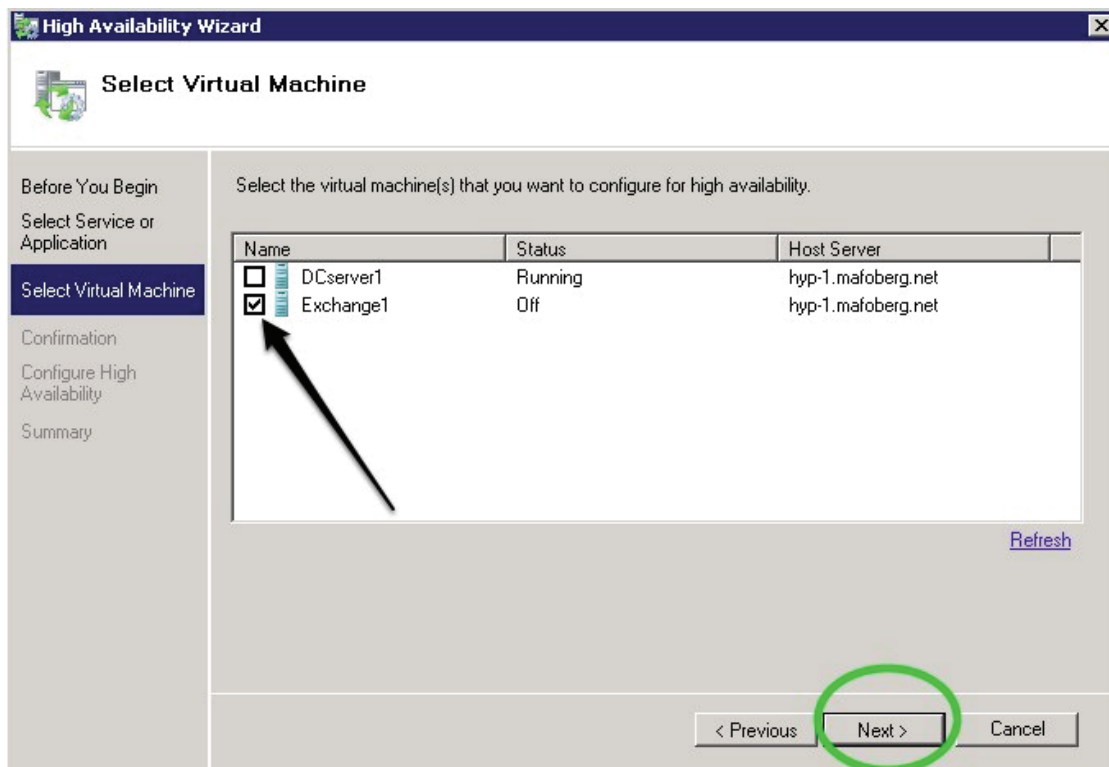
Figur 69 - Valg av virtuell maskin. Det er også mulig å sette opp andre tjenester som highly available.

Vi ser at vår Active Directory Server kjører, og kan derfor ikke flyttes over til delt lagring.

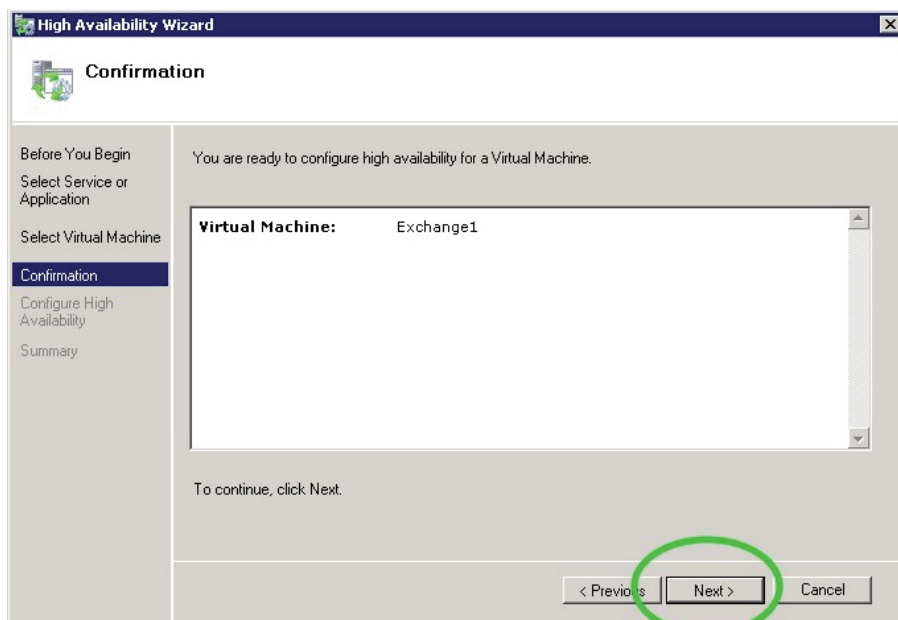


Figur 70 - Kjørende virtuelle maskiner kan ikke flyttes

Her velger vi derfor bare å flytte over exchange, som allerede er skrudd av.

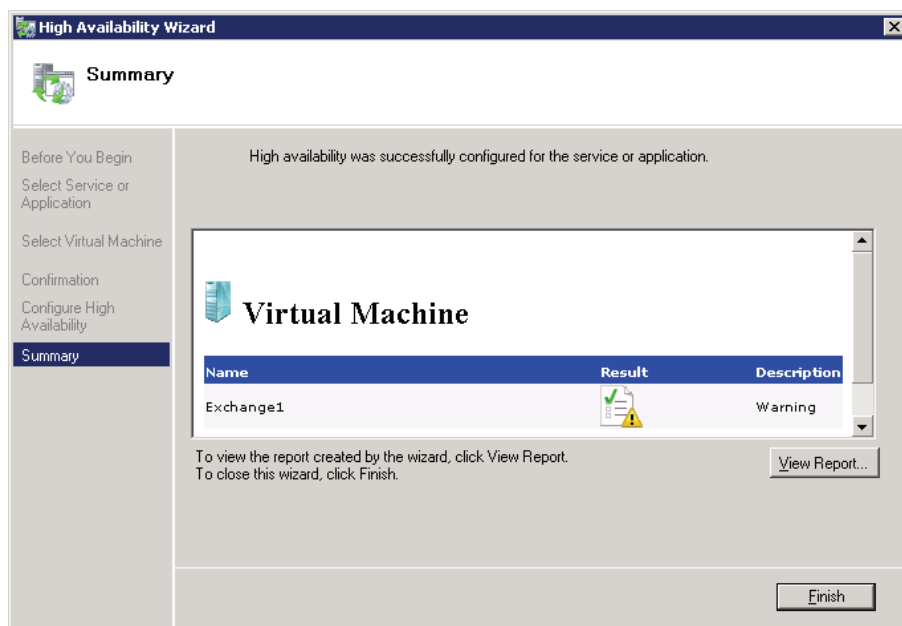


Figur 71 - Flytte avslått maskin



Her ser vi at flytteprosessen starter.

Figur 72 - Flytteprosessen starter



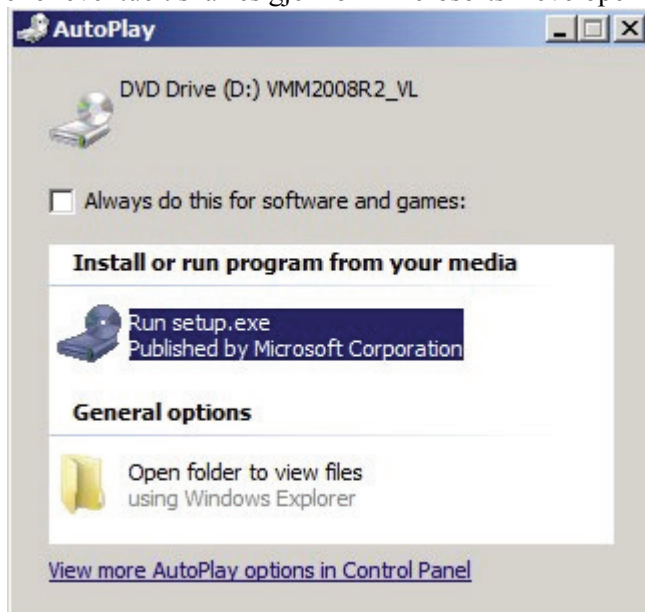
Noen småfeil eller advarsler kan forekomme, men den virtuelle maskinen vil nå være flytta over. Merk at dette vil ta noe tid, fordi hele konfigurasjonen og harddiskene må flyttes over til det delte lagringsmediet.

Figur 73 - Flytteprosessen er utført

2.5 System Center Virtual Machine Manager (SCVMM)

System Center Virtual Machine manager er en del av en større programsamling for rapportering og behandling av Microsoftprogramvare. Den er spesiallagd for å gjøre det lettere å administrere og overvåke enkeltmaskiner og clustrede noder som kjører Microsoft Hyper-V virtualiseringstjenester, samt VMWare ESX Server, og muligheten for å styre disse gjennom både et administrativt grensesnitt samt fra gjennom powershell. Vi skal se på begge disse mulighetene under kapittelet ”Drift Av Systemet”, og flere av scriptene som ligger som vedlegg til kapittel 4, ”Script av Powershell”, benytter de powershell commandletene som Scvmm tilbyr, for å liste ut informasjon om de virtuelle maskiner som er tilgjengelig. Scvmm tilbyr også enklere administrering av migrering av fysiske maskiner til virtuelle maskiner.

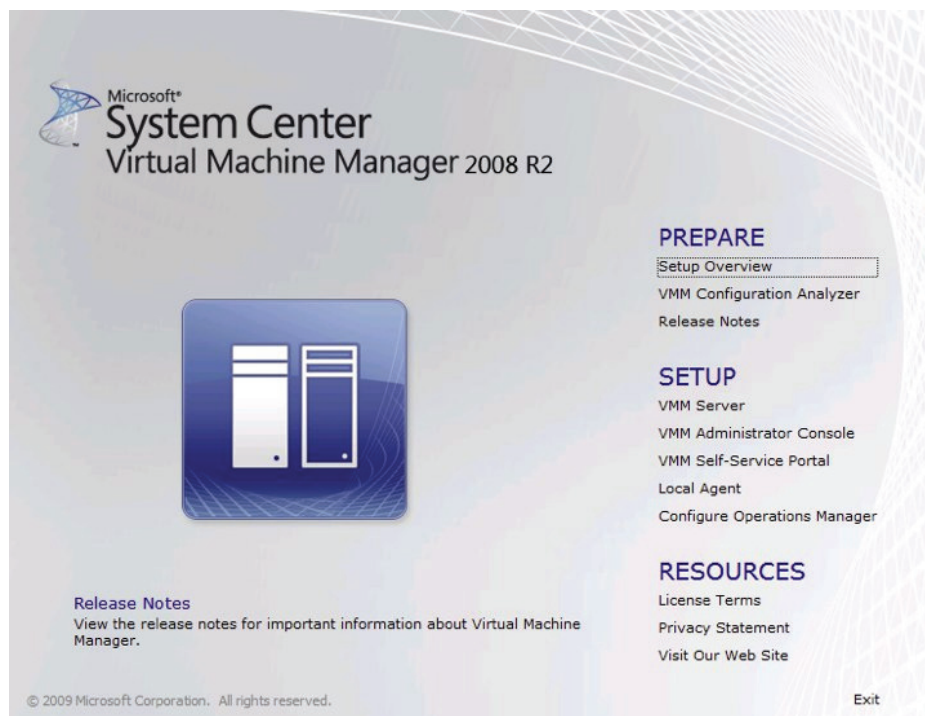
Scvmm er ikke tilgjengelig som en rolle i Windows Server, og må normalt sett kjøpes av en reseller, eller eventuelt skaffes gjennom Microsofts Developer Network.



Vi starter derfor opp fra installasjonsplata.

Figur 74 - VMMmedium for installasjon

Oppstartsbildet viser at en har flere valg en kan gjennomgå før og etter installasjon. Vi ser også at Microsoft har valgt å separere ut Scvmm som en serverversjon og en administratorkonsoll. Forskjellen er at serveren kun kan kjøres på en enkelt maskin, mens administratorkonsollet kobler til denne serveren for å administrere virtuelle maskiner. Det er ingenting i veien for at både scvmm-server og konsoll kan kjøre på samme maskin



Figur 75 - SCVMM installasjonsvalg

- **VMM Server** er selve serverprogramvaren med databaser og koblinger for styringen av de virtuelle maskinene.
- **VMM Administrator Console** er en klientprogramvare som kan brukes mot SCVMM-serveren for å styre den. Da man slipper å sitte direkte på serveren som kjører serverprogramvaren.
- **VMM Self-Service Portal** er en WEB-portal som ofte er etterspurt for styringsenheter. Dette gir mulighet for å styre serveren fra webgrensesnitt som kan gi bra nok oversikt så man slipper å installere klientprogramvaren på for mange maskiner.
- **Local Agent** er en agent på serveren som hjelper til med å kommunisere med andre scvmm-servere. Da kan man samle flere maskiner og utstyr i et bibliotek. Tilganger til andre virtuelle maskiner fra andre vmm's.
- **Configure Operations Manager** er et tillegg for System Center Operation Manager (SCOM) 2007 for konfigurering på tvers av scvmm-er for ytelse og ressurs-optimalisering med rapportering.

2.5.1.1. VMM Server

VMM server er programvare som kjøres på en egen maskin. Denne kan godt kjøres på en virtuell maskin, men ikke en virtuell maskin som ligger i et cluster som den styrer selv. Det vil det være lite hensiktsmessig å gjøre.

Programvaren er kun tjenester og databaser som opprettholder oversikt over maskiner. Det ingen konsoll som følger med denne. Det brukes *WMM Administrator Console* til.

WMM Server krav til maskinvare opp til 150 hosts⁵

Hardwarekomponenter	Minimum	Foretrukket
Prossessor	Pentium 4, 2 GHz (x64)	Dual-Processor, Dual-Core, 2.8 GHz (x64) eller bedre
Minne	2 GB	4 GB
Harddiskplass (uten lokal VMM database)	2 GB	40 GB
Harddiskplass (med lokal SQL Server 2005 Express Edition database)	10 GB	50 GB
Harddiskplass (med lokal, full versjon SQL Server 2005 database)	80 GB	150 GB

Tabell 4 - VMM Serverkrav

WMM Server krav til maskinvare mer enn 150 hosts⁶

Hardwarekomponenter	Minimum	Foretrukket
Prossessor	Pentium 4, 2.8 GHz (x64)	Dual-Processor, Dual-Core, 3.6 GHz eller bedre (x64)
Minne	4 GB	8 GB
Harddiskplass	10 GB	50 GB

Tabell 5 - VMM Serverkrav

I tillegg krever den noe software på forhånd.

- Windows Server 2008 x64
- Windows Remote Management (WinRM) 1.1 eller senere. (Inkludert i Windows Server 2008 R2.)
- Microsoft .NET Framework 3.0
- Windows Automated Installation Kit (WAIK) 1.1
- Supportert MS SQL Server (Standard eller bedre er foretrukket)

⁵ Tabell hentet fra <http://technet.microsoft.com/en-us/library/cc764224.aspx> og oversatt.

⁶ Tabell hentet fra <http://technet.microsoft.com/en-us/library/cc764224.aspx> og oversatt.

2.5.1.2. VMM Administration Console

Administratorkonsollet er et konsoll som ikke trenger å bli installert sammen med serveren. Men noen velger å gjøre dette likevel. Det er ikke noen direkte ulemper med dette. Men administratorer foretrekker å ha administratorkonsollet på sin egen maskin når de kobler til VMM serveren. Maskinen som VMMkonsollet installeres på må være medlem av domenet den skal styre VMM serveren på.

Konsollet finner man på samme installasjonsmedium som serveren (*VMM Administrator Console* som vist i Figur 75).

WMM Server krav for å administrere opp til 150 hosts⁷

Hardwarekomponenter	Minimum	Foretrukket
Prosesor	Pentium 4, 550 MHz	Pentium 4, 1 GHz eller bedre
Minne	512 MB	1 GB
Harddiskplass	512 MB	2 GB

Tabell 6 - VMM Konsollkrav

WMM Server krav for å administrere over 150 hosts⁸

Hardwarekomponenter	Minimum	Foretrukket
Prosesor	Pentium 4, 1 GHz	Pentium 4, 2 GHz eller bedre
Minne	1 GB	2 GB
Harddiskplass	512 MB	4 GB

Tabell 7 - VMM Konsollkrav

I tillegg krever den noe software på forhånd.

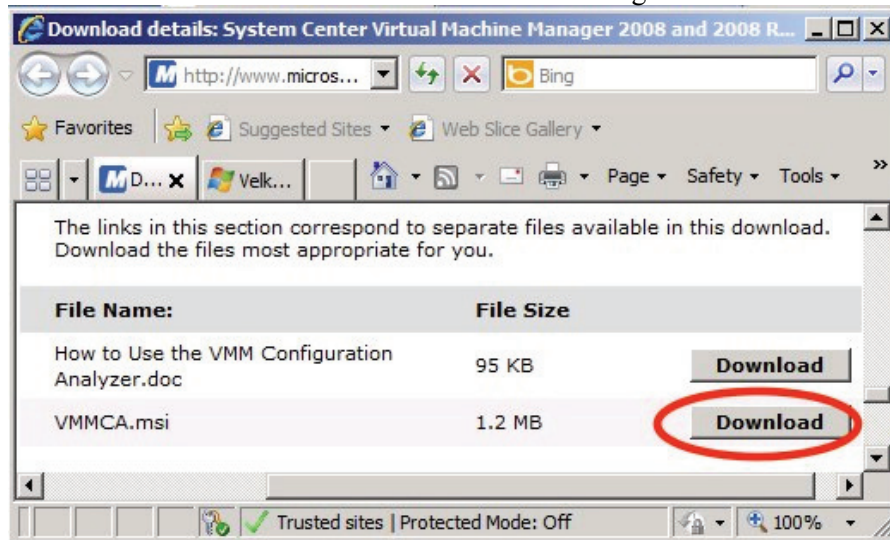
- OS for kjøring av konsollet (Windows XP Pro SP2 eller senere).
- Windows PowerShell 1 eller 2
- Microsoft .NET Framework 2.0

⁷ Tabell hentet fra <http://technet.microsoft.com/en-us/library/cc764321.aspx> og oversatt.

⁸ Tabell hentet fra <http://technet.microsoft.com/en-us/library/cc764321.aspx> og oversatt.

2.5.1 Analyse og validering av scvmm-server

Før en går i gang med installasjon av softwaren, er det meget lurt å kjøre en analyse av serveren for å finne eventuelle feil som kan hindre Scvmm fra å fungere.



Her har vi blitt videreført til Microsoft sine nettsider etter at vi trykte "VMM Configuration Analyzer" i skjermbildet overfor.

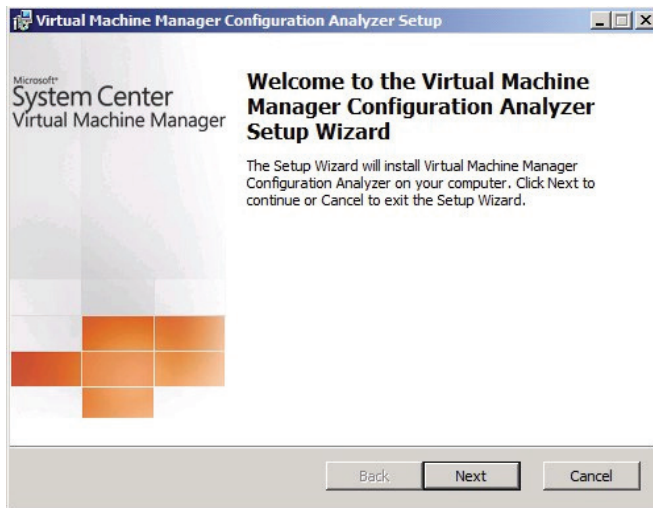
Figur 76 - VMM Server Analyse og valideringssoftware for nedlasting

Dersom ikke allerede installert, vil Configuration Analyzer be deg installere Baseline Configuration Analyzer først. Her velges naturlig nok 64bit-versjonen, fordi vår windowsinstallasjon også er 64bit.

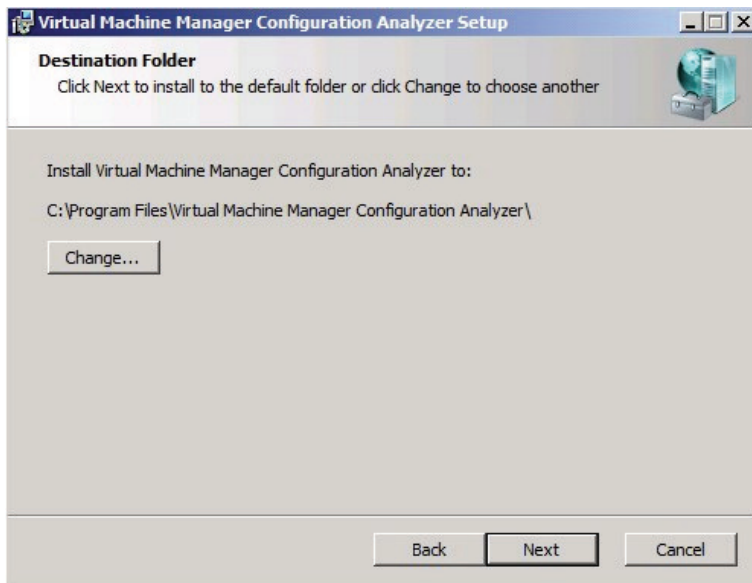
File Name:	File Size	
MBCASetup32.msi	1.3 MB	Download
MBCASetup64.msi	1.3 MB	Download

Figur 77 - VMM Server Baseline Configuration Configuration

Installasjonen av Configuration Analyzer er veldig enkelt



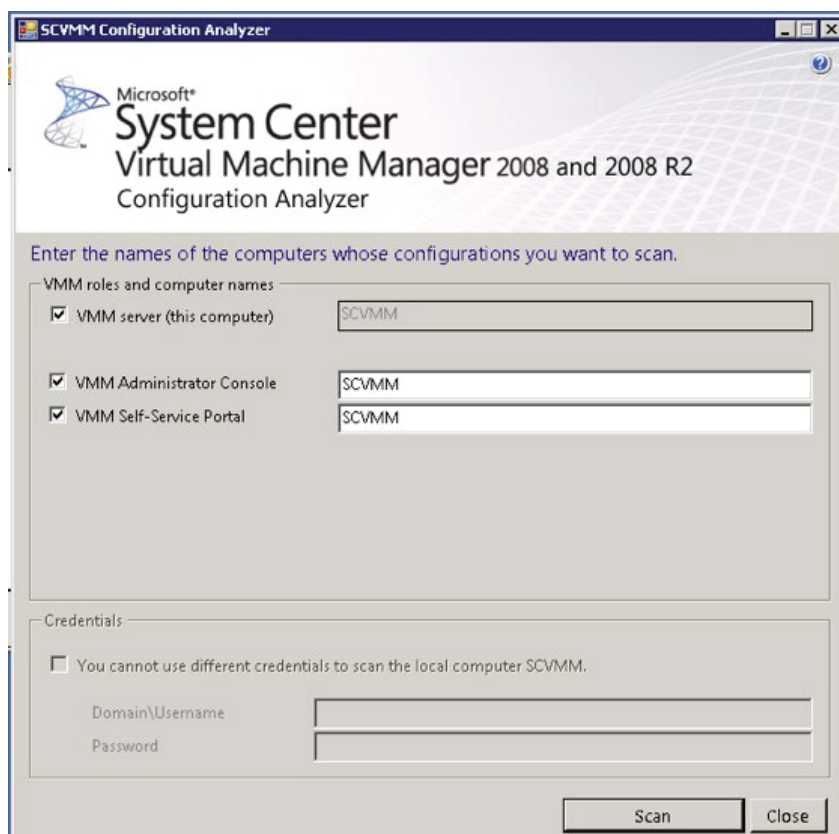
Figur 78 - VMM Server Configuration Analyzer



Vi installerer her bare programmet på den plassen som blir foreslått.

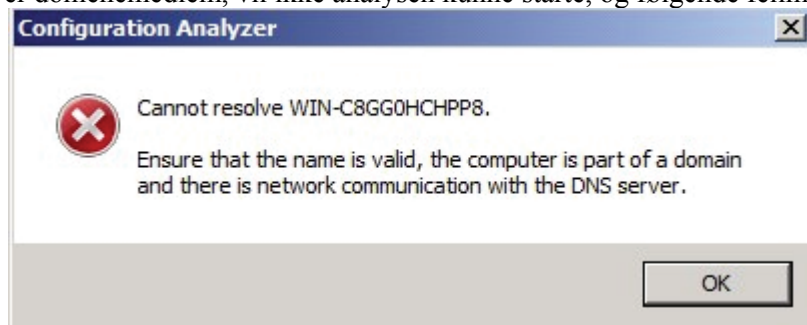
Figur 79 - VMM Server Configuration Analyzer plassering

Configuration analyzer er i gang. Vi ser her at maskinen vår blir hetende scvmm.mafoberg.net



Figur 80 - VMM Server Configuration Analyzer kjøring

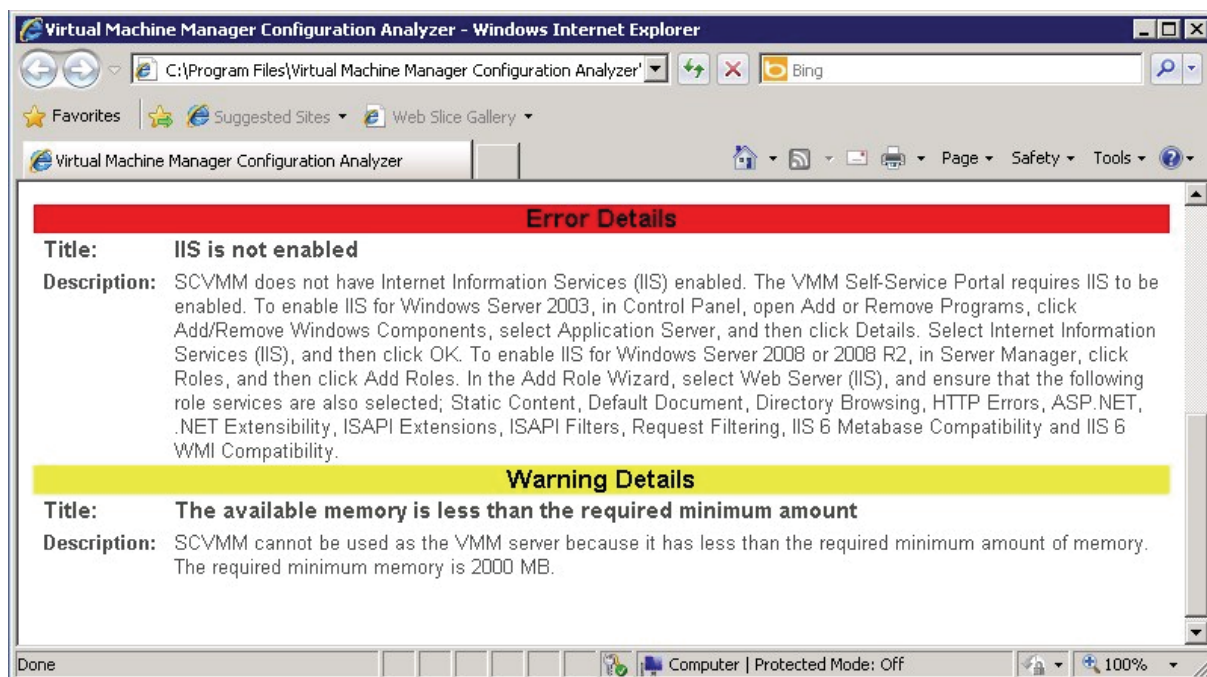
Det er viktig at maskinen som skal kjøre scvmm er en del av domenet og at installasjonen kjøres av en bruker med administratorrettigheter på både domenet og den lokale maskinen. Dersom maskinen ikke er domenedlem, vil ikke analysen kunne starte, og følgende feilmelding vil komme fram:



Figur 81 - Domene og dnskrav

Det er også verdt å merke seg at vi skal administrere virtuelle maskiner som er medlem av domenet vårt. Scvmm vil heller ikke la seg installere dersom den ikke er medlem av et domene.

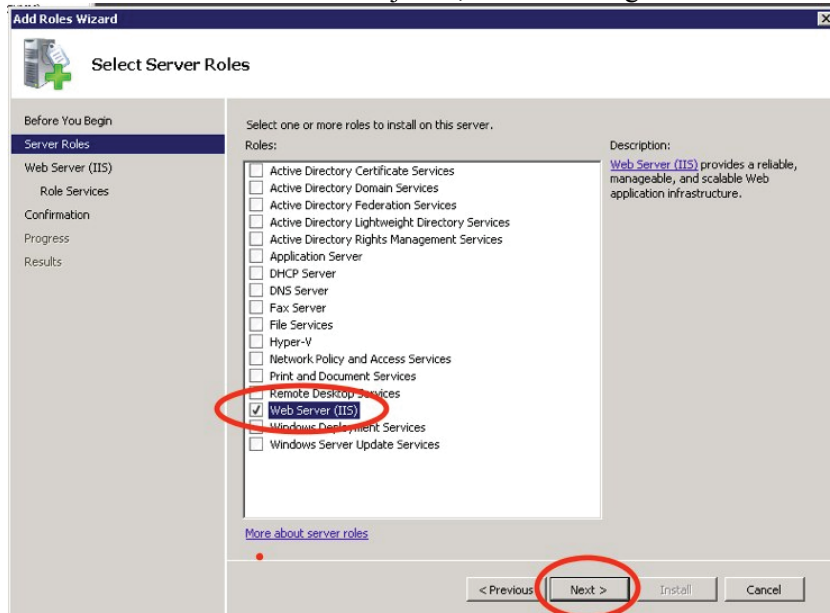
Etter scanning (trykk "Scan") vil en rapport åpnes i Internett Explorer. Her ser vi to feilmeldinger, hvor den øverste indikerer at installasjonen ikke kan fortsette før IIS er installert.



Figur 82 - SCVMM Server Configuration Analyzer rapport

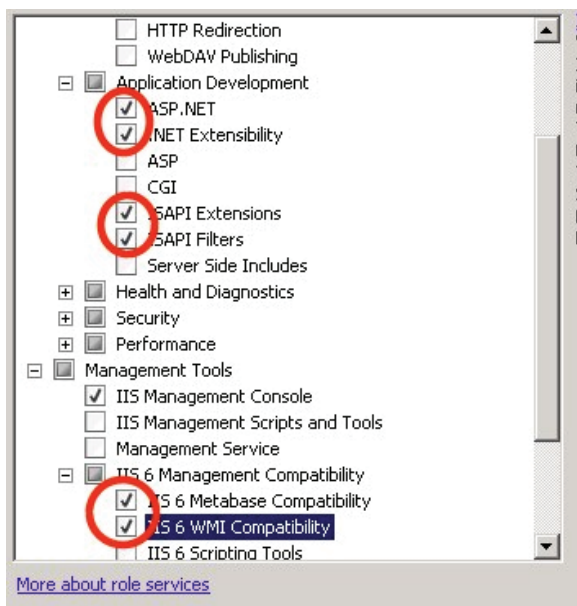
2.5.1.1 Installasjon av IIS

Før vi altså kan fortsette installasjonen, må vi derfor gå til Server Manager og velge "Add Role"



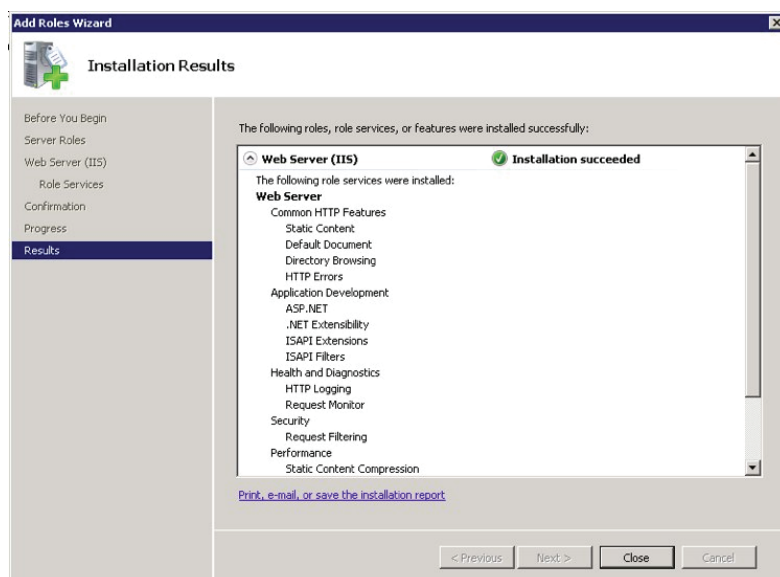
Rollen Web Server (IIS) er her valgt for installasjon.

Figur 83 - SCVMM Server Installasjon av rollen IIS



I tillegg til standardvalgene, må altså de punktene som var nevnt i rapporten installeres. Dette inkluderer rammeverket ASP.NET, filtre og kompatibilitetslagre for IIS6 og installeres under "Role Services"

Figur 84 - SCVMM Server installasjon av IIS



Installasjonen av IIS er komplett.

Figur 85 - SCVMM Server installasjon av IIS ferdig

2.5.1.2 Minnestørrelse

Vi så også i analyserapporten at Scvmm krever minst 2GB minne. Vår server som kjørte med 768MB minne var derfor ikke tilstrekkelig. Vi valgte derfor å oppgradere serveren til å ha 2.1GB med minne. Merk likevel at feilmeldingen bare var kategorisert som en advarsel ("warning"), og at installasjonen kunne fortsette med mindre enn 2GB Minne, selv om det ikke er anbefalt.

Når vi nå kjører analysen en gang til, vil den ikke finne noen feil, og vi kan gå i gang med installasjonen av Scvmm.

Virtual Machine Manager Configuration Analyzer Report

[Privacy Statement](#)

Run Date: mars 11, 2010 12:04:23

For information about how to use and troubleshoot problems with the Virtual Machine Manager Configuration Analyzer, please visit the [VMMCA download site](#).

To scan a remote computer, make sure that a firewall is not blocking Remote Service Management or Management Instrumentation (WMI) communication.

To scan an installed VMM server with a remote VMM database, ensure that remote connections are enabled on the SQL Server database instance.

No configuration issues found.

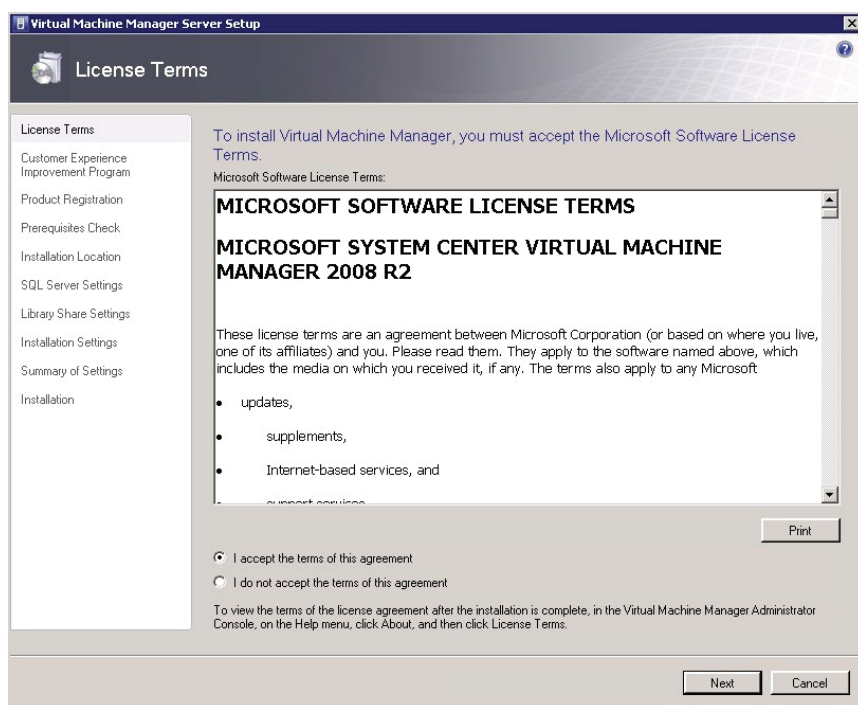
Figur 86 - SCVMM Server Configuration Analyzer andre gang

2.5.2 Installasjon av SCVMM-server

Vi starter installasjonen ved å trykke på "VMM Server" i åpningsbildet.



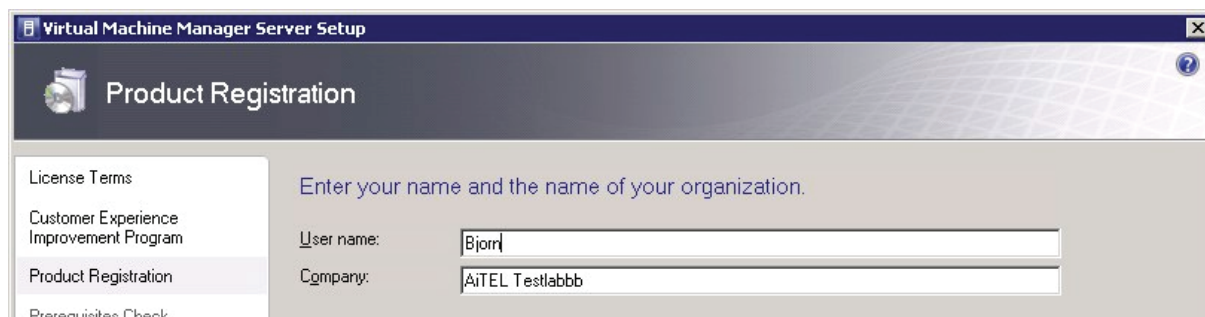
Figur 87 - SCVMM Server installasjon



Det første bildet som møter oss ved installasjonen, er en generell lisensavtale som må aksepteres. Deretter følger et skjermbilde som spør om vi vil delta i en kundeundersøkelse. Dette unngår vi.

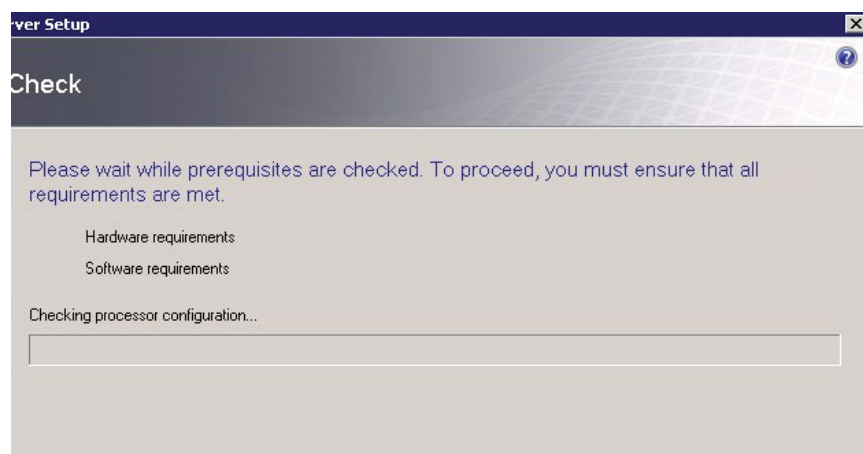
Under *Product Registration* nevner vi at vi er på AiTEL sin testlab.

Figur 88 - SCVMM Server installasjon



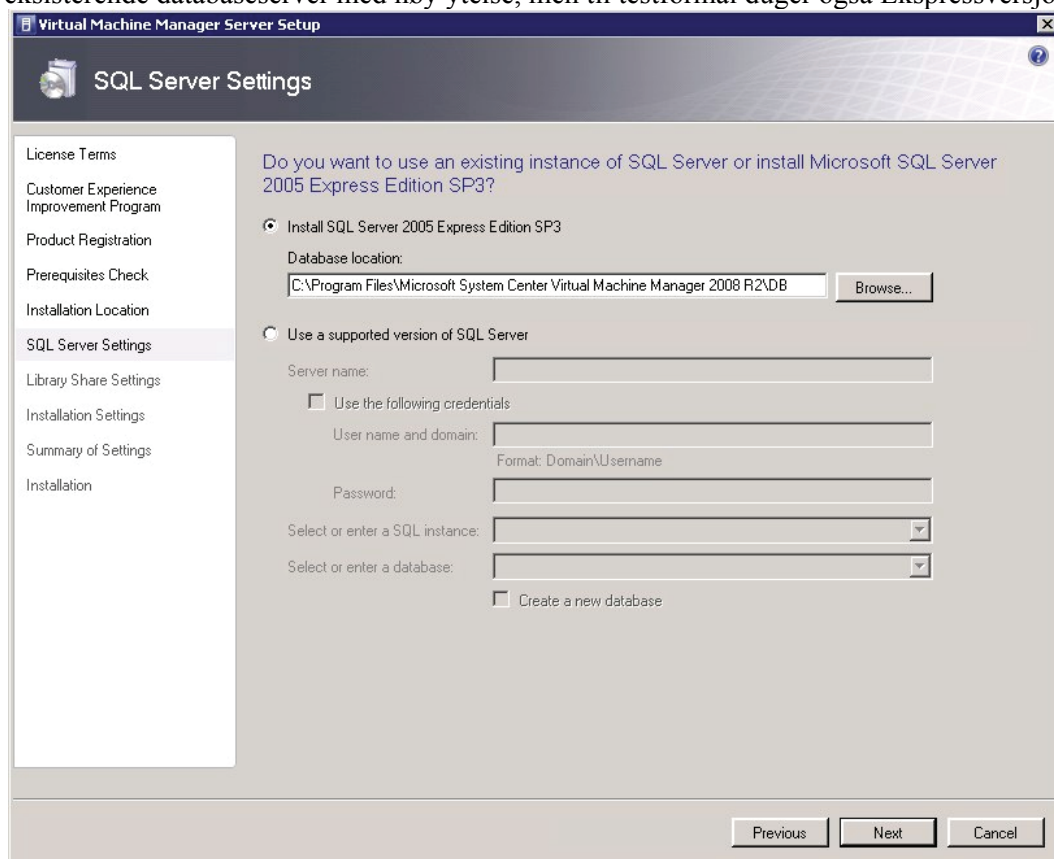
Figur 89 - SCVMM Server registrering

Her gjør installasjonen sin egen validering av hardwaren og softwaren den kjører på. Ettersom vi tidligere i analysen luket vekk alle feilmeldinger, vil denne sjekken også passere uten problemer.

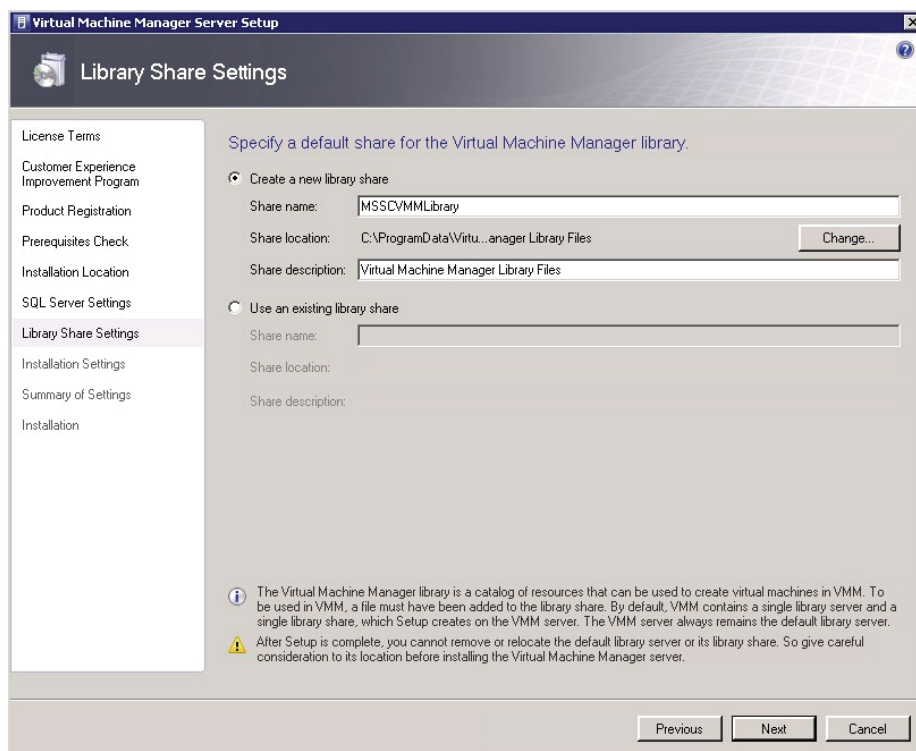


Figur 90 - SCVMM Server sjekk og prosess

For å lagre konfigurasjonsdata, bruker Scvmm en SQL-database. Her anbefales det sterkt å velge en eksisterende databaseserver med høy ytelse, men til testformål duger også Ekspressversjonen.



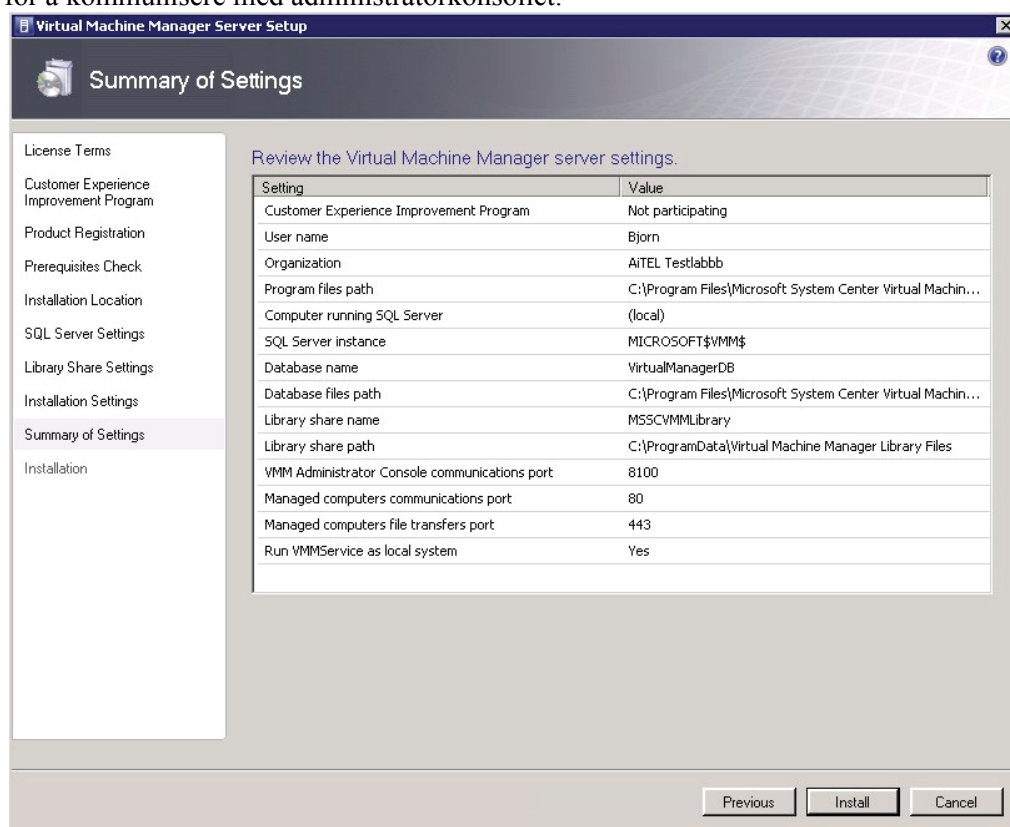
Figur 91 - SCVMM Server plassering av programvare



Figur 92 - SCVMM Server Libraryfunksjonens oppsett

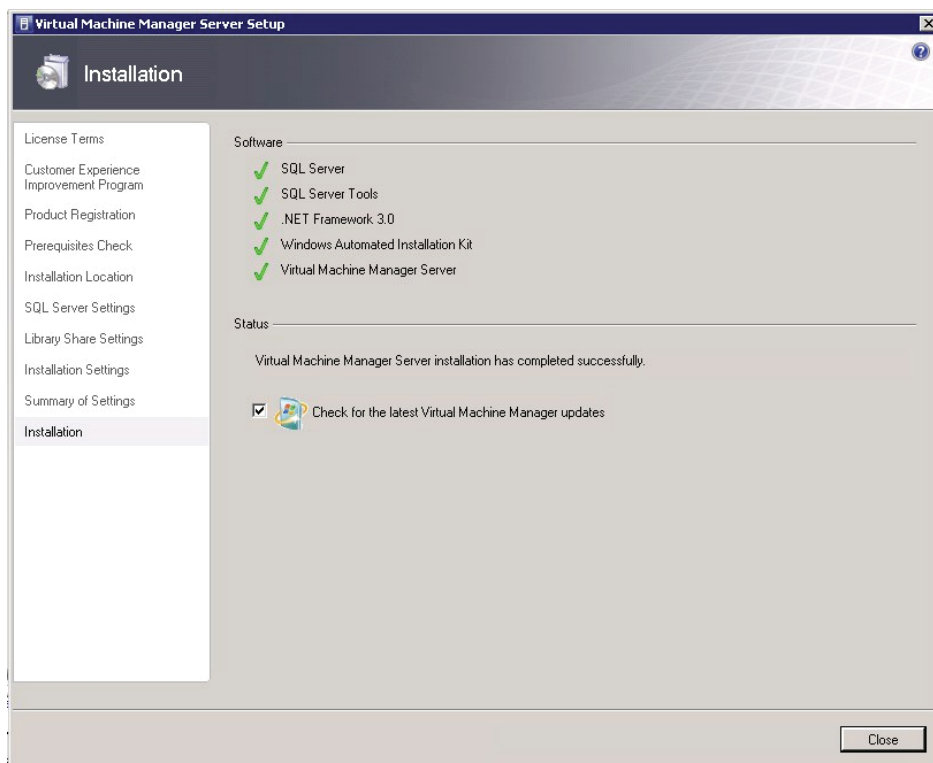
Biblioteket vil inneholde store tilleggsressurser som lagrede iso-bilder of profiler av hardware og operativsystem. Det er en sentral lagringsplass for ting som ikke skal brukes umiddelbart.

I summary ser vi hvorfor Scvmm krever IIS; den bruker http og https-standardportene samt port 8100 for å kommunisere med administratorkonsollet.



Figur 93 - SCVMM Server oppsummering

Installasjonen vil i tillegg til Scmm og SQL Server installere .NET Framework og Windows Automated Installation kit. Scvmm vil nå også oppdatere seg gjennom Windows Update.



asdohadsldlshdlsj

Figur 94 - SCVMM Server installasjon fullført

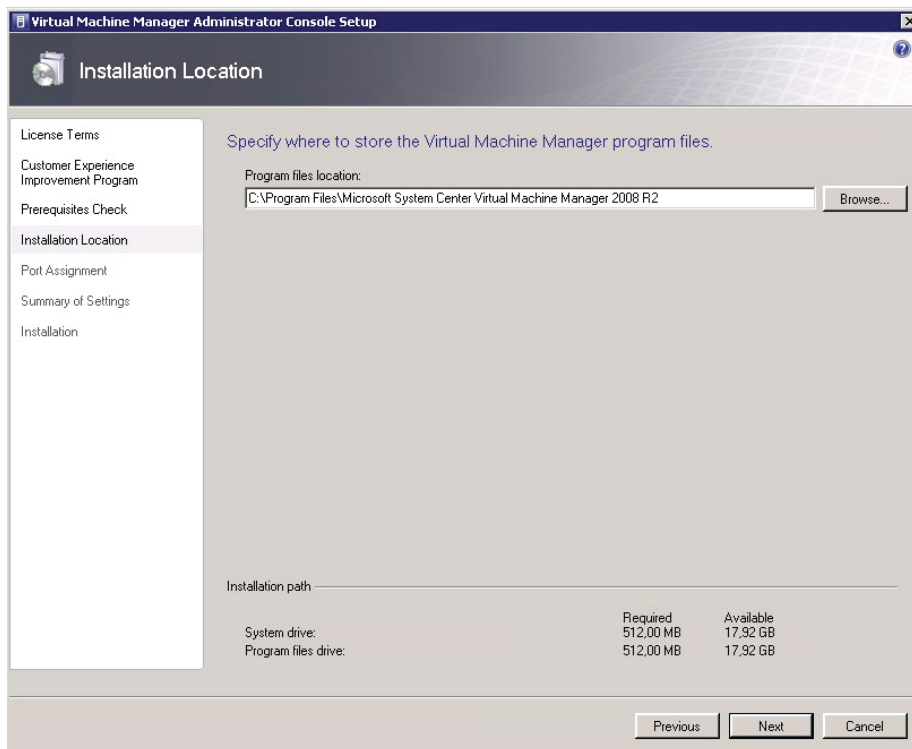
2.5.3 Installasjon av administratorkonsollet

Administratorkonsollet skal altså koble til scvmm-serveren. Vi velger å installere disse på samme maskin, men det er ingenting i veien for å installere administratorkonsollet også på andre maskiner i tillegg.



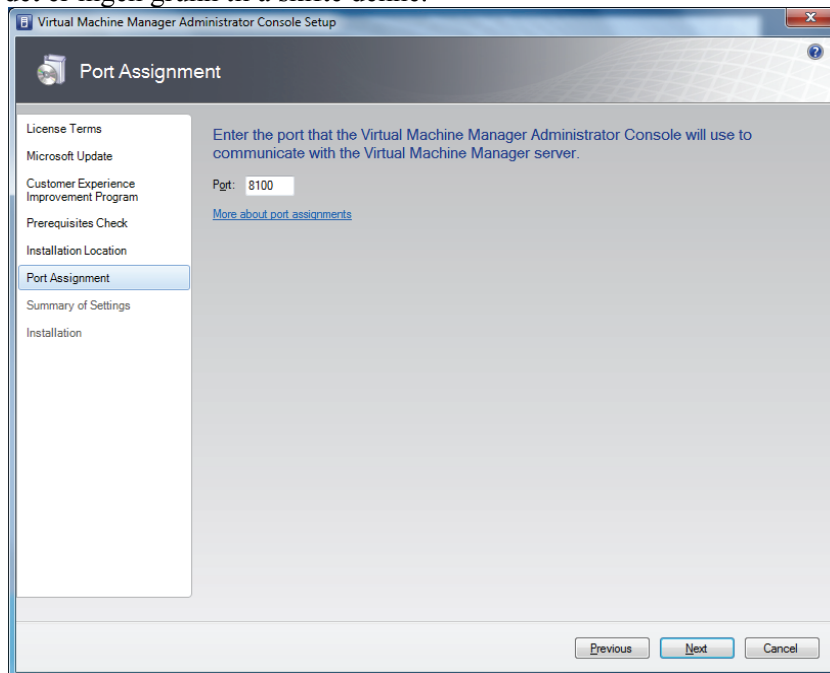
Figur 95 - SCVMM Konsoll installasjon

Plassering endrer vi heller ikke her på.



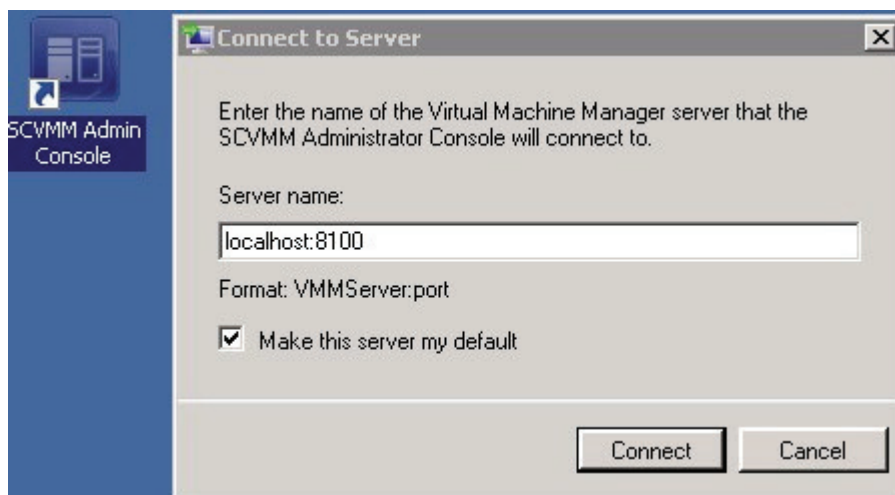
Figur 96 - SCVMM Konsoll installasjon plassering

Under Port Assignment må det velges hvilken port administratorkonsollet skal bruke for å koble til Scvmm-serveren. Dette er som standard satt til samme port som scvmm-serveren installerer seg på, så det er ingen grunn til å skifte denne.



Figur 97 - VMM Konsoll installering portnummer

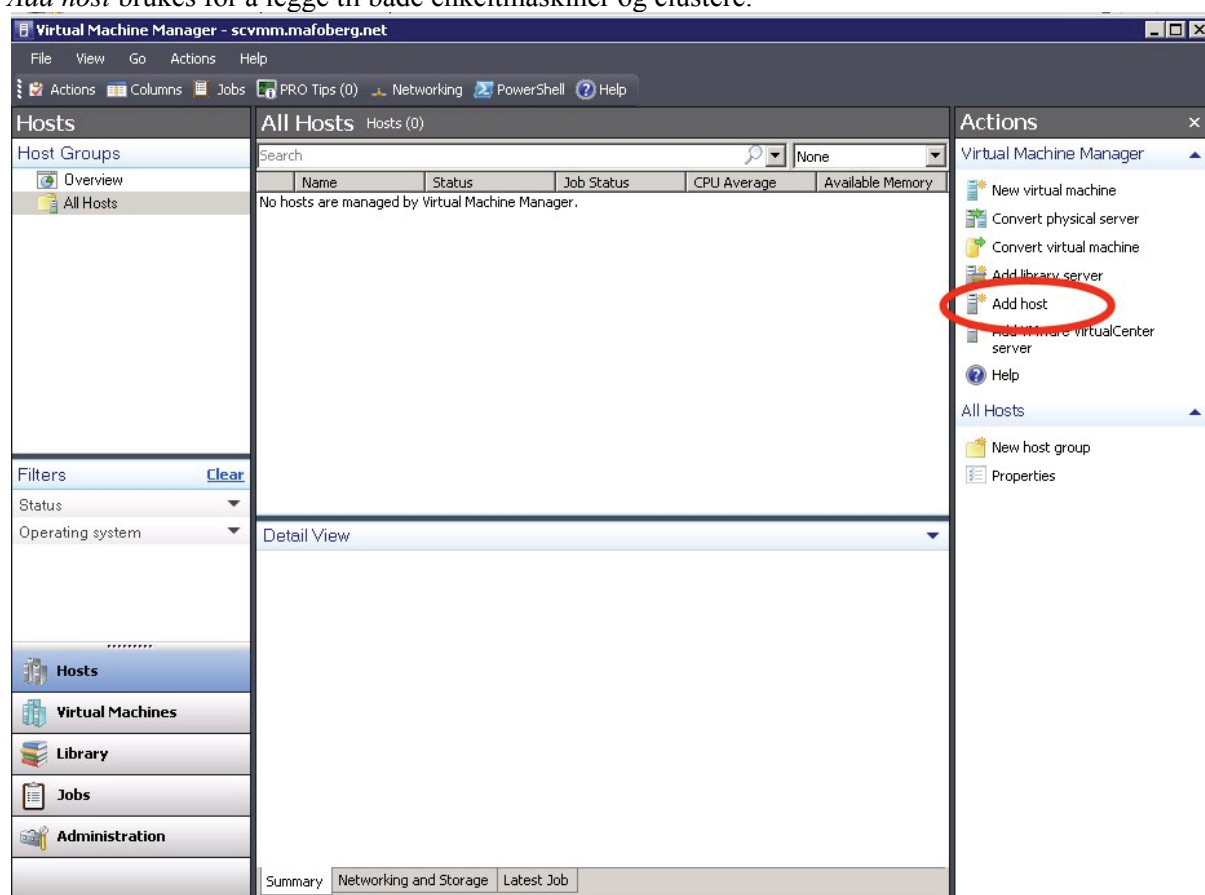
Etter fullføring av installasjonen ser vi at et nytt ikon har dukket opp på skrivebordet. *SCVMM Admin Console* åpnes, og vi velger *Connect*.



Figur 98 - Oppstart og tilkobling

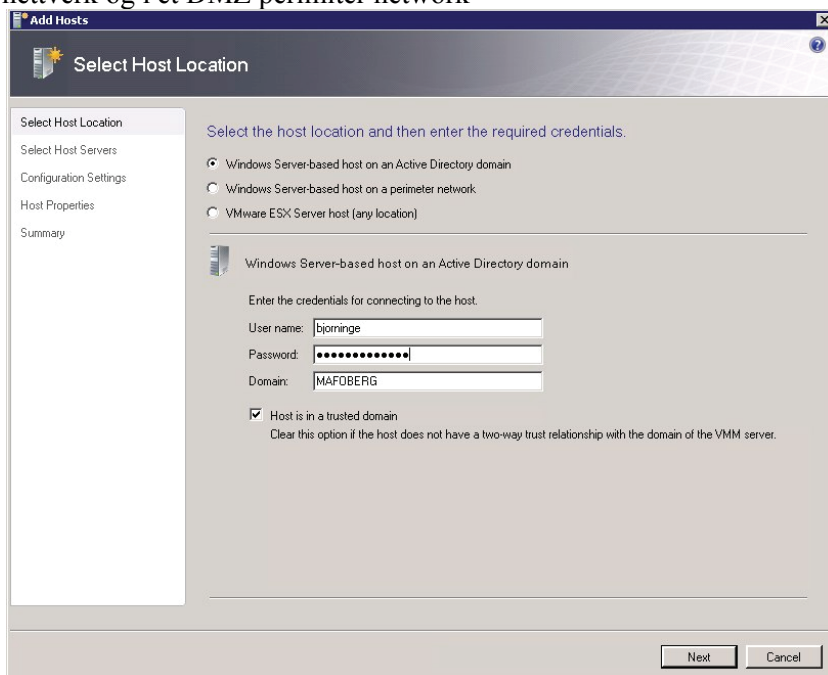
2.5.4 Legge til ny host

Vi ser her et tomt grensesnitt for scvmm. Vi har tidligere lagd et Failover Cluster bestående av to Hyper-V noder. Og skal nå vise hvordan dette clusteret legges til for management gjennom scvmm. *Add host* brukes for å legge til både enkeltmaskiner og clusterer.



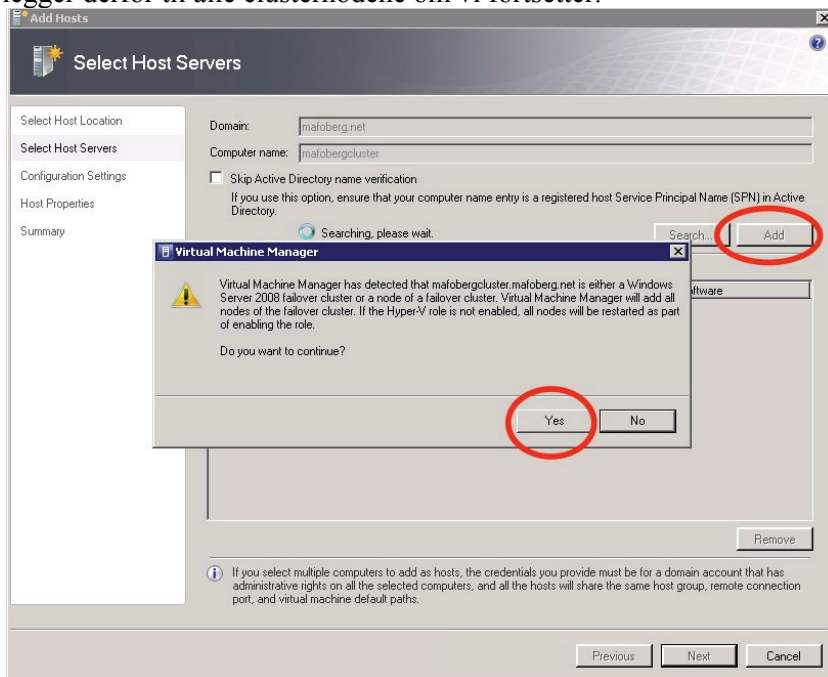
Figur 99 - SCVMM Konsoll Add hosts

Her legger vi merke til at Scvmm kan styre både ESX og Windows Server-maskiner på både et internt nettverk og i et DMZ perimeter network⁹



Figur 100 - SCVMM - Legge til host

Her legger vi til clusteret vårt. Vi ser at Scvmm oppdager at denne adressen peker mot et cluster, og legger derfor til alle clusternodene om vi fortsetter.



Figur 101 - SCVMM - Legge til cluster

Her ser vi at den fant nodene hyp-1 og hyp-2 i clusteret vårt.

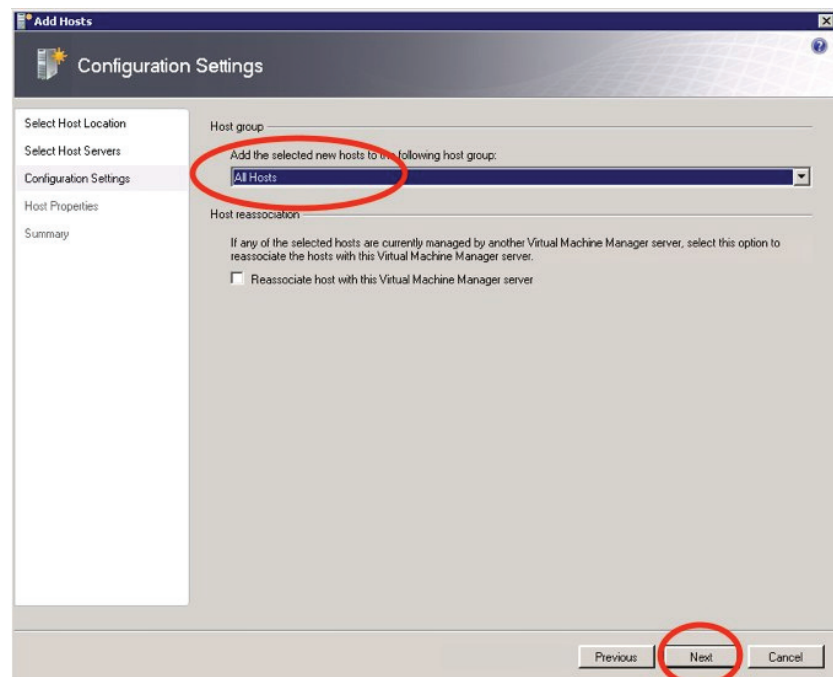
⁹ DMZ – demilitærisert sone eller også kalt et primiternettverk er en sikker sone mellom lokalnettet og internett som kan gi tjenester til begge disse netdsfsdf

Selected servers:

Computer Name	Operating System	Virtualization Software
 mafeobergcluster (hyp-1, hyp-2)	Windows Server 2008 R2 Datace...	Microsoft Hyper-V

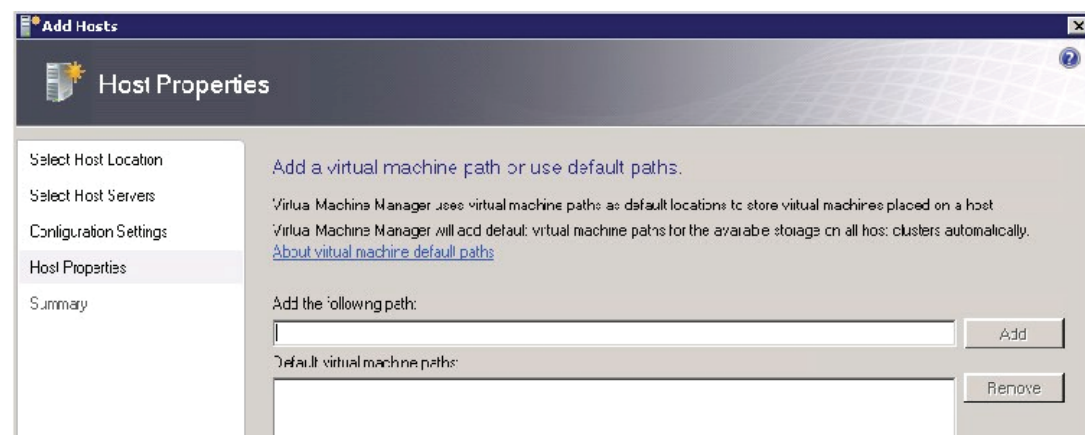
Figur 102 - SCVMM - vårt cluster

Da oppsettet vårt ikke har noen grupper satt opp, legges nodene til i standardgruppa "All Hosts". Vi får deretter videre



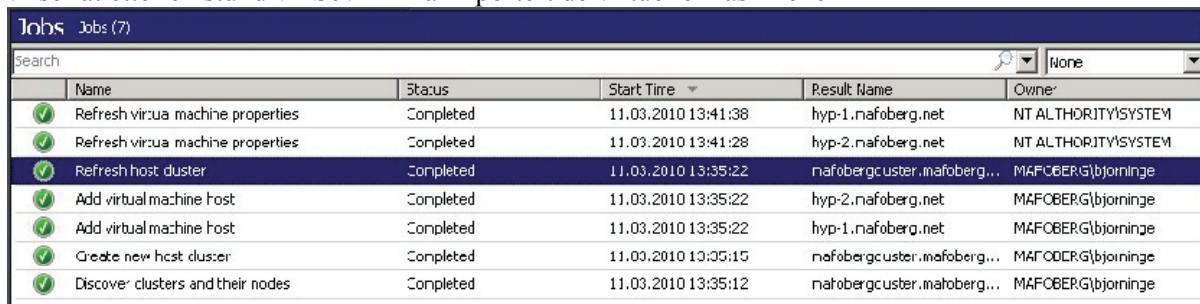
Figur 103 - SCVMM hostgruppe

Under Host properties kan vi legge til plasseringen for hvor Scvmm skal legge virtuelle maskiner. I og med at vi nå benytter et cluster, skal alt lagres der, og vi legger derfor ikke til noen plasseringer i dette punktet.



Figur 104 - SCVMM hostegenskaper

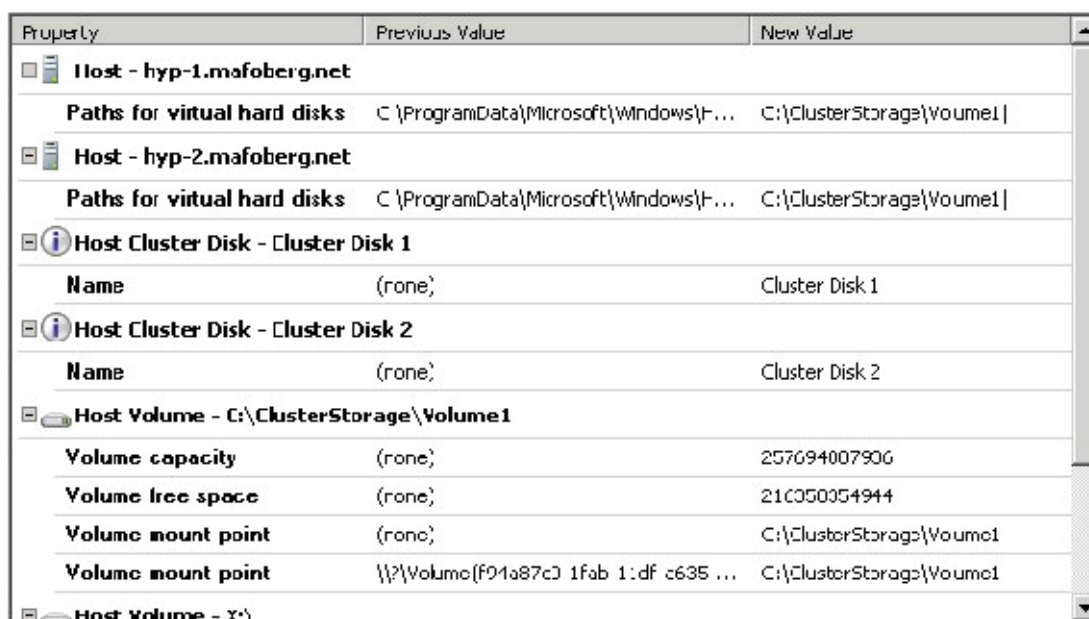
Vi ser at etter en stund vil Scvmm ha importert de virtuelle maskinene



Name	Status	Start Time	Result Name	Owner
Refresh virtual machine properties	Completed	11.03.2010 13:41:38	hyp-1.nafoberg.net	NT AUTHORITY\SYSTEM
Refresh virtual machine properties	Completed	11.03.2010 13:41:28	hyp-2.nafoberg.net	NT AUTHORITY\SYSTEM
Refresh host cluster	Completed	11.03.2010 13:35:22	mafobergcluster.nafoberg...	MAFOBERG\bjorninge
Add virtual machine host	Completed	11.03.2010 13:35:22	hyp-2.nafoberg.net	MAFOBERG\bjorninge
Add virtual machine host	Completed	11.03.2010 13:35:22	hyp-1.nafoberg.net	MAFOBERG\bjorninge
Create new host cluster	Completed	11.03.2010 13:35:15	mafobergcluster.nafoberg...	MAFOBERG\bjorninge
Discover clusters and their nodes	Completed	11.03.2010 13:35:12	mafobergcluster.nafoberg...	MAFOBERG\bjorninge

Figur 105 - SCVMM importert virtuelle maskiner

Vi ser også her at Scvmm har tatt i bruk clusterlagring for alle virtuelle maskiner og deres harddisker.



Property	Previous Value	New Value
Host - hyp-1.nafoberg.net		
Paths for virtual hard disks	C:\ProgramData\Microsoft\Windows\F...	C:\ClusterStorage\Volume1
Host - hyp-2.nafoberg.net		
Paths for virtual hard disks	C:\ProgramData\Microsoft\Windows\F...	C:\ClusterStorage\Volume1
Host Cluster Disk - Cluster Disk 1		
Name	(none)	Cluster Disk 1
Host Cluster Disk - Cluster Disk 2		
Name	(none)	Cluster Disk 2
Host Volume - C:\ClusterStorage\Volume1		
Volume capacity	(none)	257694007906
Volume free space	(none)	210350354944
Volume mount point	(none)	C:\ClusterStorage\Volume1
Volume mount point	\\?\Volume{F01a87c3-1fab-11df-c635-...}	C:\ClusterStorage\Volume1
Host Volume - X:\		

Figur 106 - SCVMM - Clusterlagring av virtuelle maskiner

2.6. Exchange 2010

Dette kapitlet skal ta for seg installasjon og oppsett av Exchange 2010. Andre tilhørende roller som kreves for installasjon vil vi også ta for oss. Andre roller som vi selv velger å kjøre i forbindelse med Exchange 2010 tar vi for oss i egne kapitler. Dette gjelder blant annet Edge Transport Role og Management Role.

Vi skal i hovedsak ta for oss å installere Exchange fra scratch. Altså ingen direkte oppgradering av tjenesten. Mailbokser og brukere skal vi importere ved hjelp av script i Powershell fra gamle servere.

2.6.1. Nytt med Exchange 2010

Exchange 2010 er en god oppgradering fra Exchange 2007. PowerShell var allerede brukt i Exchange 2007. Exchange Management Console (EMC) var bygd på dette. Det ble et brukergrensesnitt som ble meget effektivt i bruk. Siden da har Exchange 2010 blitt videre utviklet på PowerShell versjon 2. Det har kommet flere cmdlets som kan brukes og flere funksjoner.

Exchange 2010 har nye spesielle funksjoner.

- **High Availability**
En av de viktige nye funksjonene med Exchange 2010. Slik som med virtuelle maskiner gir denne funksjonen mulighet til å unngå problemer med isolasjon fra databaser, servere og nettverksfeil. Database Availability Group (DAG) er en logisk kontainer som kan inneholde sett med mailboksservere. DAG og DAG nettverk settes opp ved hjelp av EMC.
- **Archiving**
En funksjon for arkivering av mailbokser slik at de ikke blir lagret vekk i andre former av backup.
- **Federation and Sharing**
Gir mulighet for en utvidet samarbeidsmulighet med *Instant Private Messaging* som kalendere, kontakter osv med kontakter som befinner seg utenfor Exchangemiljøet for organisasjonen.
- **Certificate Managing**
Behandling av sertifikater i EMC. Dette gir muligheter for utvidelse og bedre styring av sertifikater.
- **Managing Multiple Forests**
Styring av flere skoger i samme EMC. Før var administratorer nødt til å gjøre dette via terminal-tjenester.
- **Moving Mailboxes**
Flytting av mailbokser mellom databaser og skoger. Det finnes nå cmdlets for PowerShell som kan gjøre dette.
- **Unified Messaging**
Administratorer kan nå Administrere typer fra Unified Messaging lettere. Slik som å forhåndsviser Voice Mail, konfigurasjoner og liknende hendelser.
- **Diagnostic Logging Management**
Styring for logging er med nå. Administratorer kan overvåke ved hjelp av logger og utveksle dem seg imellom. Diagnostisering er gjort lettere og alt dette styres fra EMC.
- **EMC og RBAC**
Role-Based Access Control (RBAC) styrer tilgangen til Exchange Server 2010. Styrbar fra

PowerShell. For større organisasjoner er dette en stor fordel da det blir skalerbart for administratorer og lettere få oversikt og gjøre jobben.

- **Organizational Health**

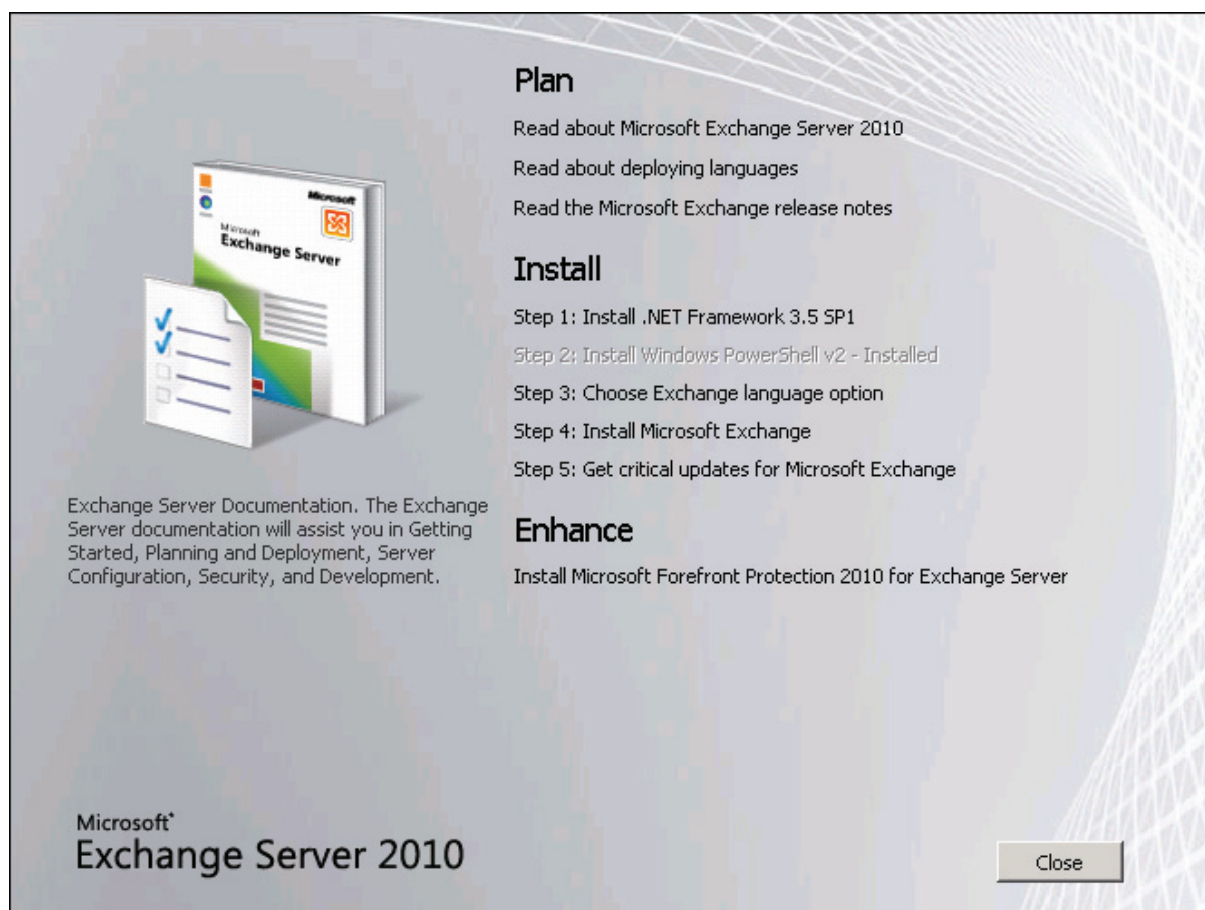
Oversikt over helsenivå for Exchange infrastrukturen rett i EMC. Oppsummeringer over Client Access Licenses (CLAs), databaser og epostmottakere.

- **Bulk Editing**

Det er mulighet til å editere flere brukere og mailbokser i samtid. Ved å merke flere og velge egenskaper og oppsett på dem. Før var dette bare mulig under PowerShell.

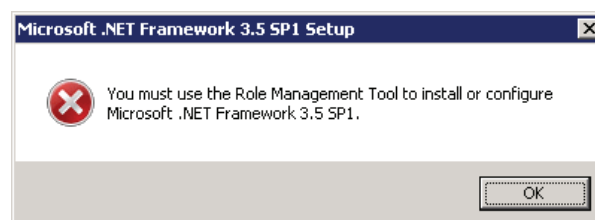
2.6.2. Installasjon

Vi installerer med et iso-image koblet til den virtuelle dvd-stasjonen i den virtuelle maskinen (mu_exchange_server_2010_x64_dvd_x15-82068.iso). Dette gjøres i den overordnede maskinen som kjører Hyper-V. Når denne monteres får vi beskjed på den virtuelle maskin om at en plate er montert. Autorun er satt til å kjøre Setup.exe. Kjør denne. Man får opp en velkomst hvor man får det meste man trenger for exchange av informasjon og installasjonsmulighet for å få denne opp å kjøre.



Figur 107 - Startskjerm

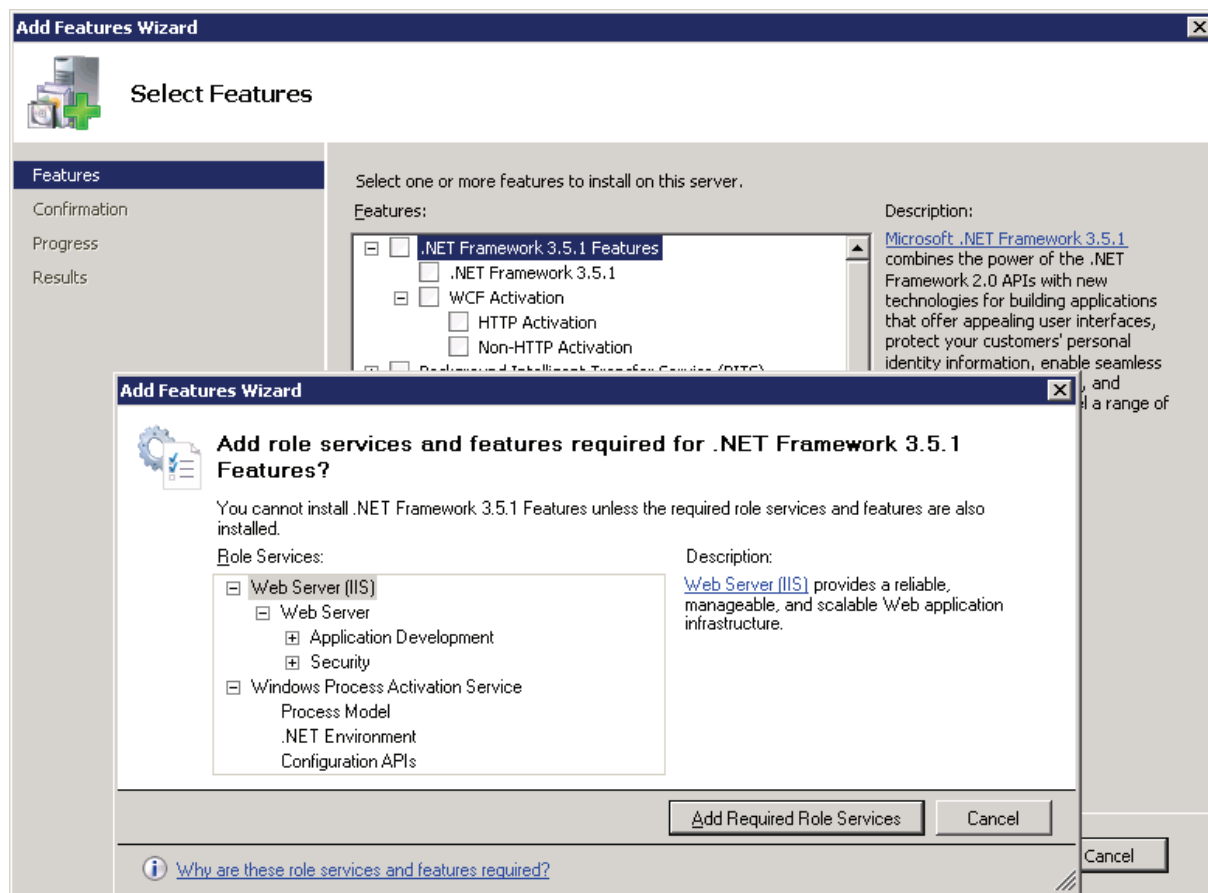
Installasjonen av Exchange 2010 x64 går igjennom 5 steg som man må fullføre. Vi skal ta for oss disse stegene her.



Figur 108 - .NET Framework error

2.6.2.1. Installere .NET Framework 3.5 SP1

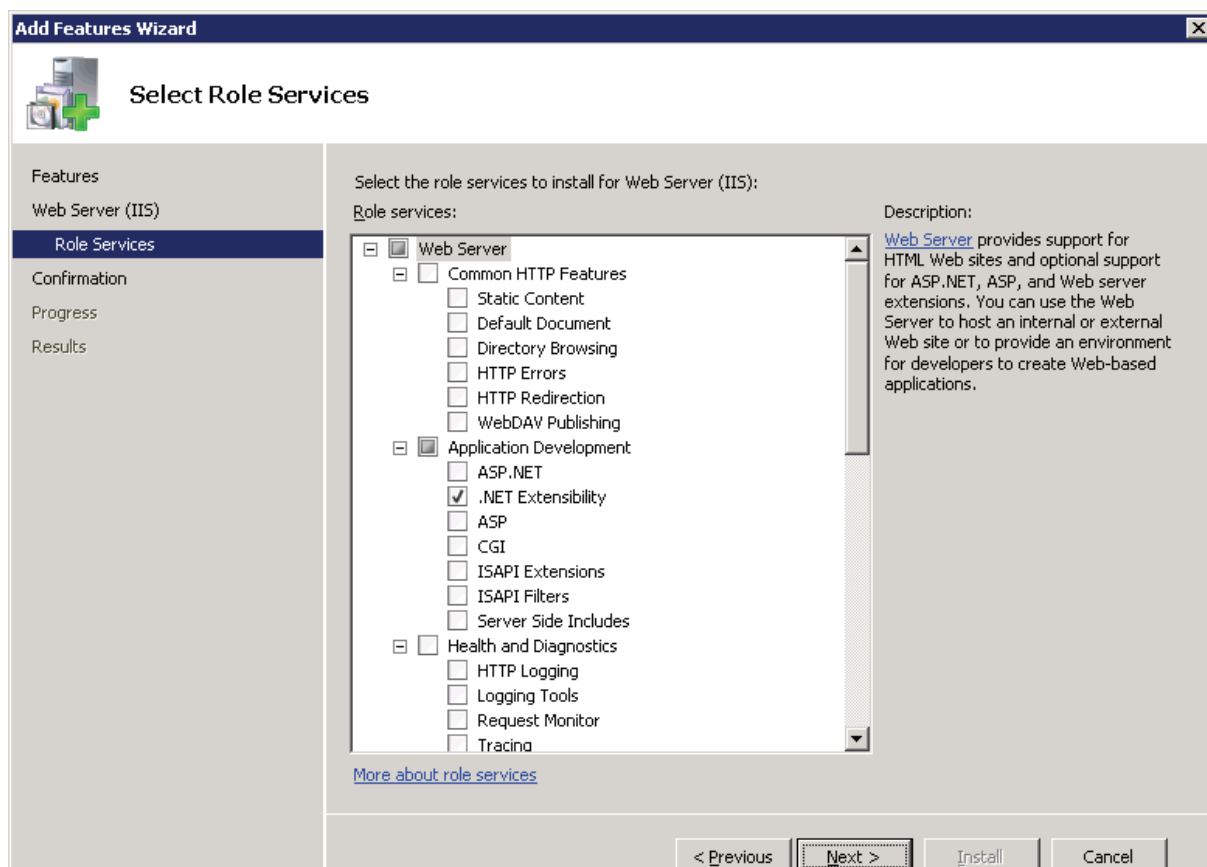
Linken ”Step 1” fører til Microsofts nettsider for å laste ned dotnetfx35setup.exe. Windows Server 2008 R2 kommer allerede med .NET Framework 3.5.1 som et tillegg. Derfor er denne installasjonen overflødig. Rammeverket må likevel settes opp. Dette gjør vi i Features fra Server Management.



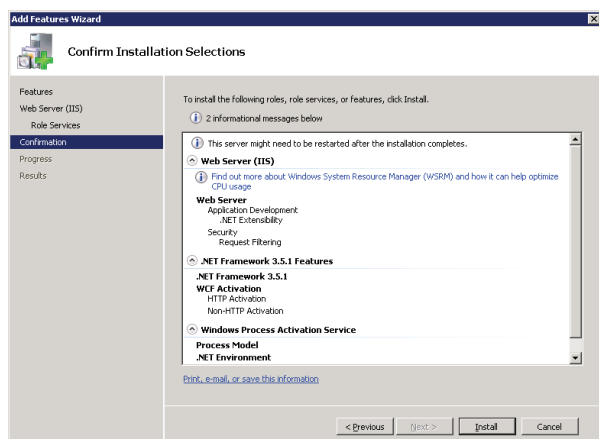
Figur 109 - Installasjon Feature .NET Framework 3.5.1

Klikke på ”.NET Framework 3.5.1 Features” starteroppsettet automatisk. Her får vi en gjennomgang hva som følger med rammeverket, og disse kreves også for at hele rammeverket skal installeres.

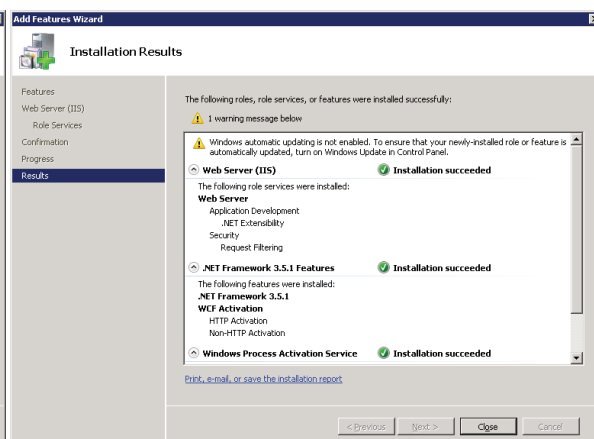
Web Server (IIS) er web-tjenesten fra Microsoft. Denne må installeres på grunn av rammeverket, men også på grunn av selve Exchange. Exchange gir som kjent en mulighet for webgrensesnitt for epost. Vi vil komme til å se i flere forskjellige roller som installeres krever webtjenesten for å kjøre. *Windows Process Activation Service (WPAS)* er en tjeneste som gjør det mulig å unngå bruk av http-protokollen for å kjøre administrasjonsverktøy. Tjenestene som før var tilgjengelige over http-protokollen er nå tilgjengelig via Windows Communication Foundation (WCF). WPAS gjør det mulig med tilgang til aktiveringsbeskjeder over http-protokollen støttet fra IIS 7.0.



Figur 110 - Install IIS 7.5 standard - 1



Figur 111 - Install IIS 7.5 standard - 2



Figur 112 - Install IIS 7.5 standard - 3

Her gjør vi ingen forandring siden det ikke er IIS 7.5 vi hovedsakelig skal installere i denne veiledningen. Ekstra tillegg i IIS 7.5 vil bli lagt til de applikasjonene som har behov for det senere, slik som selve Exchange 2010. Vi kjører videre og installerer denne og får resultatskjermen som i Figur 107.

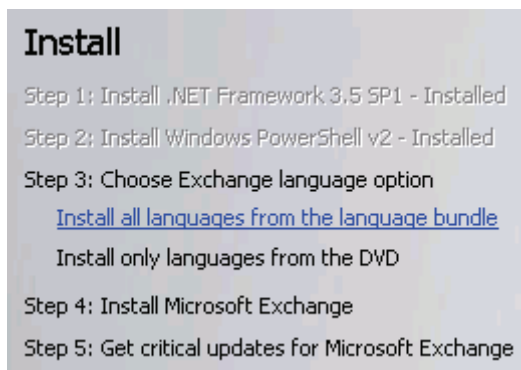
2.6.2.2. Installere Powershell v2

Windows Server 2008 R2 inneholder allerede Powershell versjon 2. Denne er aktiv om man installerer Exchange 2010 på Windows Server 2008 eller tidligere serverversjoner.

2.6.2.3. Velge Exchange språkmuligheter

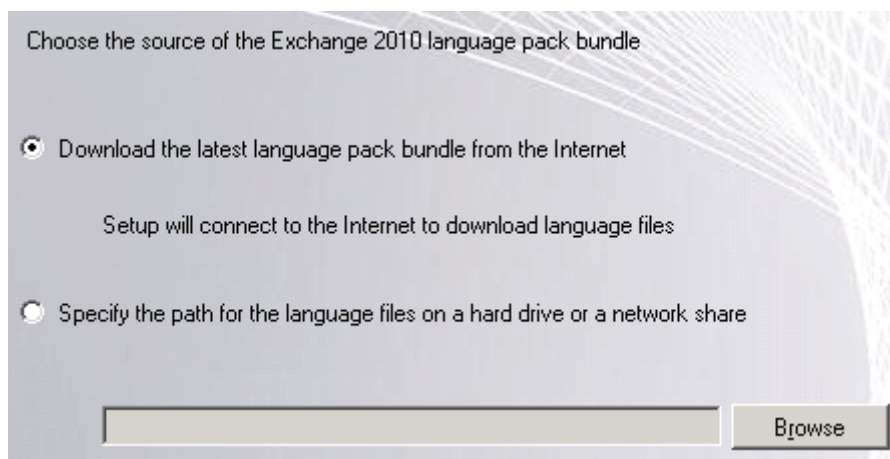
Vi får to valg av språkmuligheter. Dette innebærer språkstøtte for Exchange generelt. Siden vi også kan tenke oss støtte for norsk, går vi for valg som gir dette med under installasjon med en gang isteden for å gå for dette senere.

- **Install all languages from the language bundle**
Dette valget tar med nødvendige språk under installasjon med en gang. Eventuelt kan man bla igjennom egne bundle-pakker med språk om man har dette fra før.
- **Install only languages from the DVD**
Her får man kun installert engelsk (U.S.) språkversjon. Etter installasjon kan man senere legge til andre språkpakker. Velger vi denne, blir steget ferdig med en gang.

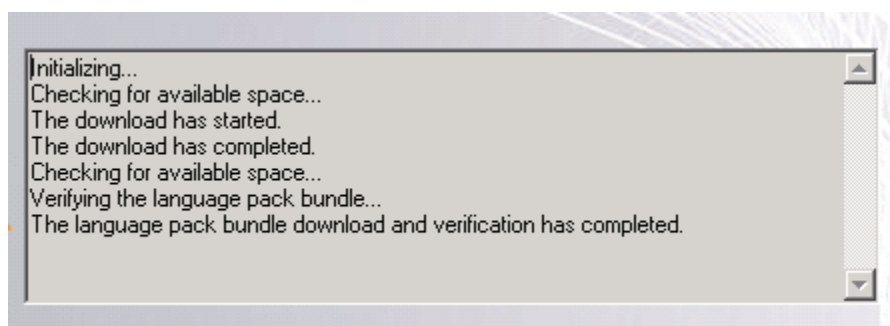


Figur 113 - Språkvalg

Vi velger å installere alle språk fra språkpakkene som vist i Figur 108 merket blått.



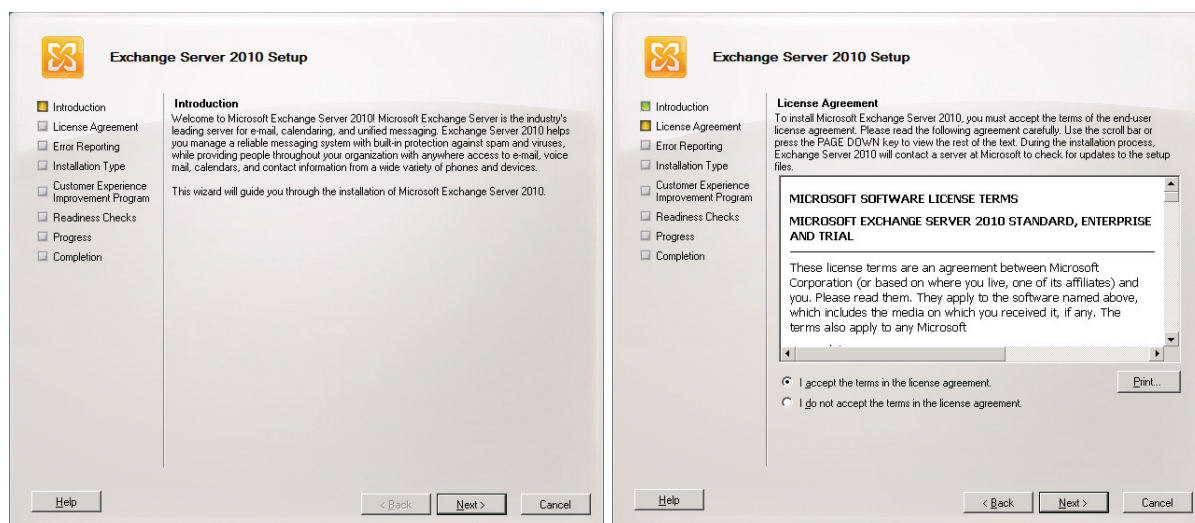
Figur 114 - Språkpakkenedlasting



Figur 115 - installering og klargjøring

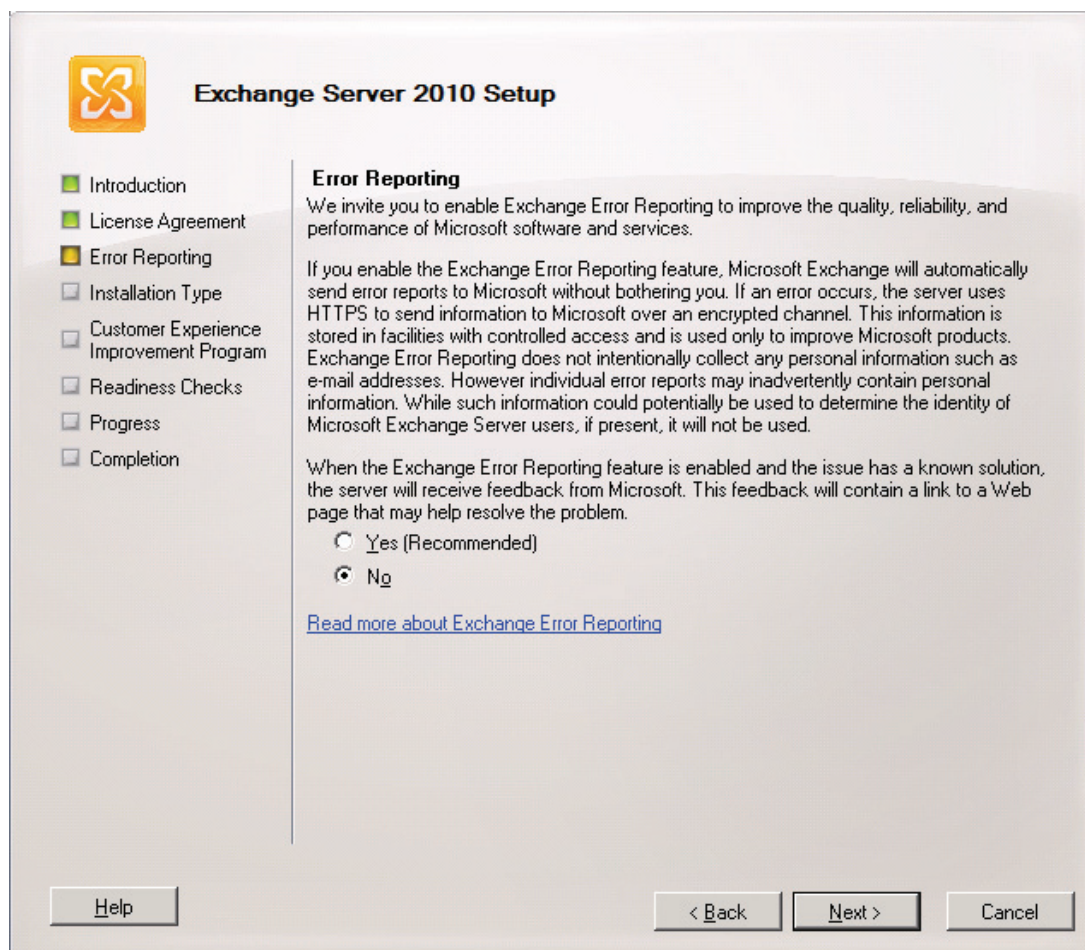
Vi får da valg om å laste ned eller å velge språkpakken som nevnt tidligere. Vi velger her å laste ned seneste språkpakke og deretter ferdiggjøre dette.

2.6.2.4. Installere Exchange



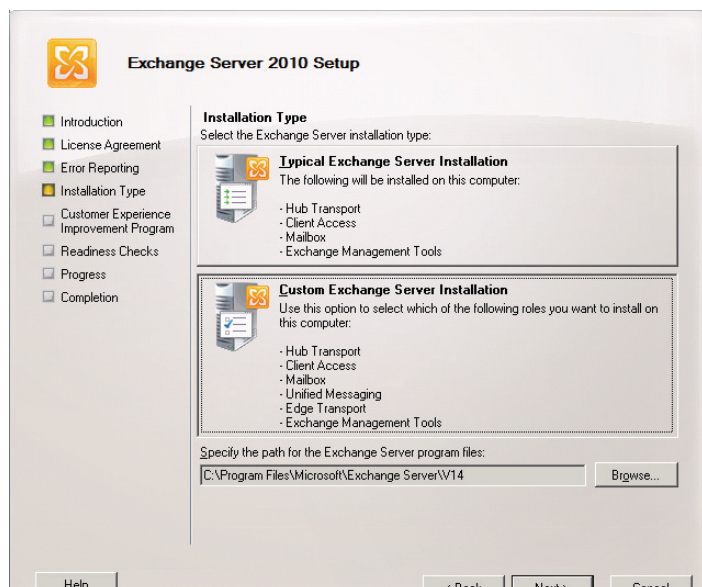
Figur 116 – Introduksjon og lisens

Installasjon startes fra åpningsskjermen og vi kommer til en introduksjon og lisens for programvaren.



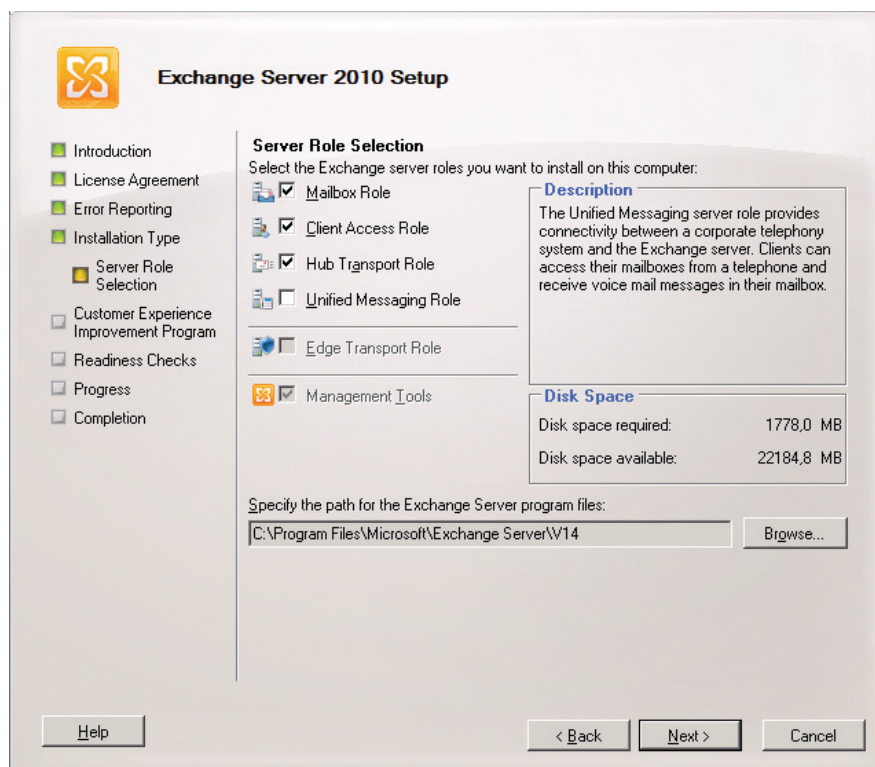
Figur 117 - Feilmeldingsrapportering

Error Reporting er en funksjon for å hjelpe Microsoft med å gi informasjon som skal brukes til å forbedre programvaren for Microsoft. Det er ”anbefalt” å hjelpe Microsoft, men vi ser ingen grunn her til å legge til denne funksjonen. Vi velger derfor nei.



Figur 118 - Installasjonstype

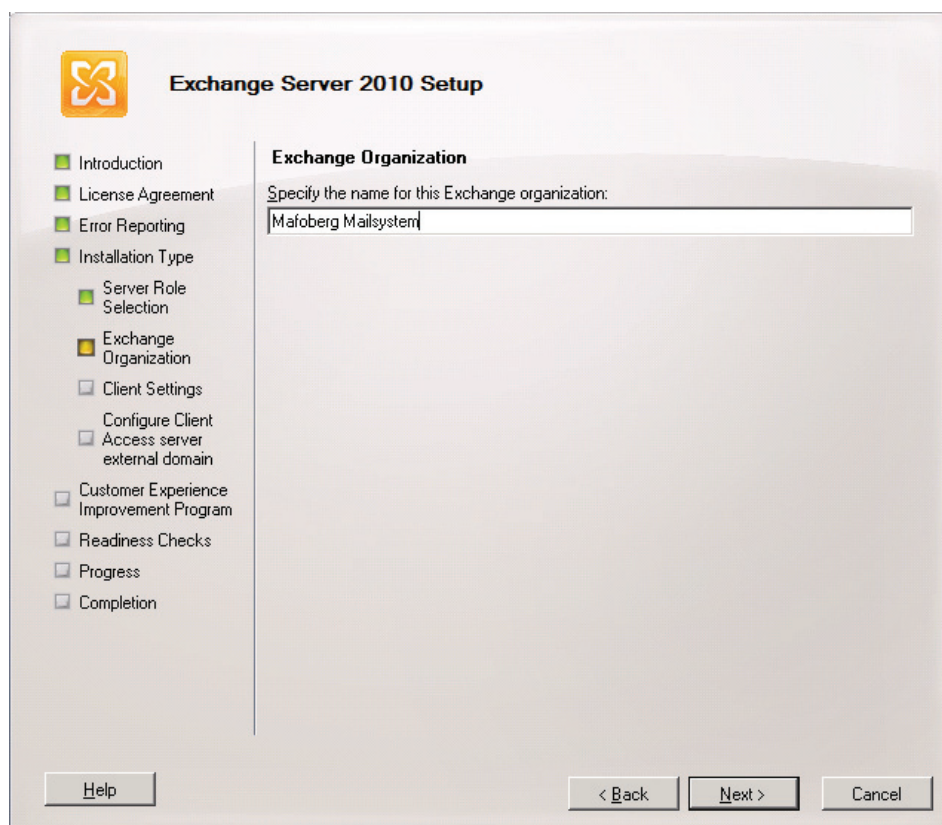
Vi velger manuell installasjon for å få mulighet til å få informasjon om de forskjellige delene, og for å kunne ha mulighet til å velge komponenter vi vil installere. Vi ser blant annet at vi her har mulighet til å velge Edge Transport rollen og Unified Messaging rollen.



Figur 119 - Rollevalg

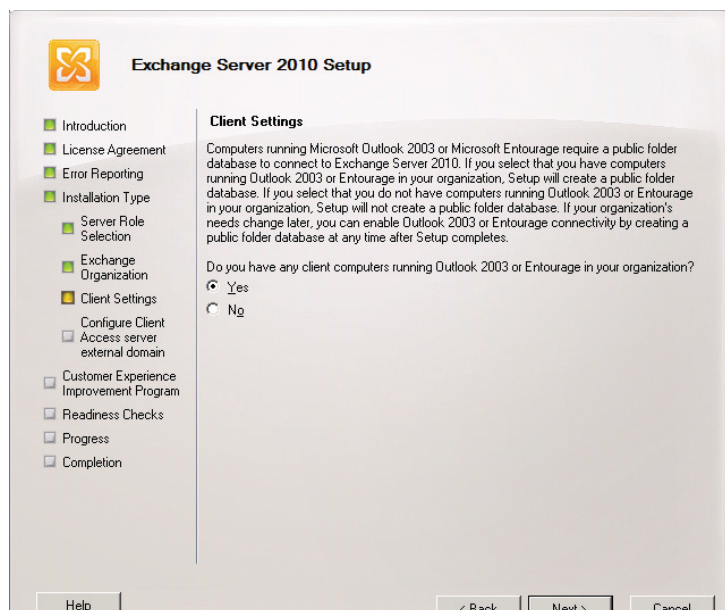
I Figur 119 ser vi at samme hva slags rolle vi velger, får vi med Management Tools uansett. Men vi ikke lov å kjøre Edge Transport rollen sammen med selve Exchange-serveren. Denne må vi installere på egen server senere. Dette blant annet fordi den ikke skal ha tilgang til AD og at funksjonen med Edge er beregnet på datatransporten mot Exchange for styring av filtrering med mer.

- **Mailbox Role**
Rollen er selve Exchange programdel for lagring og styring av datalagring. Dette er selve Exchange mailsystem.
- **Client Access Role**
Styringsenhet for oppkobling fra klienter. Web-grensesnitt, POP, IMAP eller mobile enheter.
- **Hub Transport Role**
Rollen gir mulighet til ruting innen Active Directory. Rollen kan også brukes til policyer for sikkerhet, spam, virus og meldinger generelt.
- **Unified Messaging Role**
Rollen gir mulighet for tilkobling av telefoner som støtter meldinger og voicemail i Exchange.
- **Edge Transport Role**
Server-rolle som gir støtte for filtrering av epost for blant annet virus og spam. Denne tjenesten må legges til utenfor domenet som en gjennomgang for meldinger osv.
- **Management Tools**
Dette er verktøy for å konfigurere og styre tjenestene i Exchange Server. Denne følger med uansett hvilke andre roller som installeres. Men den kan også installeres helt alene.



Figur 120 - Exchange organisasjonsnavn

Her velger vi navn for Exchangeorganisasjonen. Dette er et navn vi kanskje ikke vil blande selve firmanavnet inn i. Dette fordi navnet ikke vil kunne forandres så enkelt. Mange firma har satt enkle navn som "Corporate Email", "Exchange" eller liknende. Dette gir en mulighet for at firma slipper gamle navn som tilhører gamle deler eller andre firma om de skulle bli kjøpt opp eller slått sammen med andre. Men vi velger her "Mafoberg Mailsystem" siden vi satser på at denne bedriften består selv.



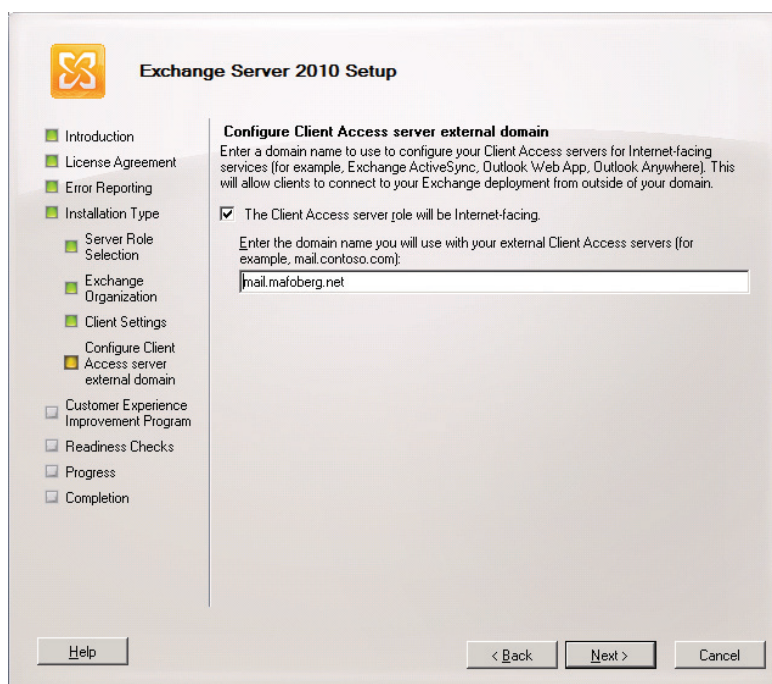
Figur 121 - Klientoppsett for Outlook 2003

I starten vil vi ta med støtte for Outlook 2003. Mafoberg Systems AS har fortsatt gamle Outlook-klienter i nettverket, og det er ikke med det første sagt at de skal renovere hele bedriftens programvare da dette vil komme til å koste penger. Det er også tatt med fordi vi har delt oppgraderingen for hele bedriften i deler.

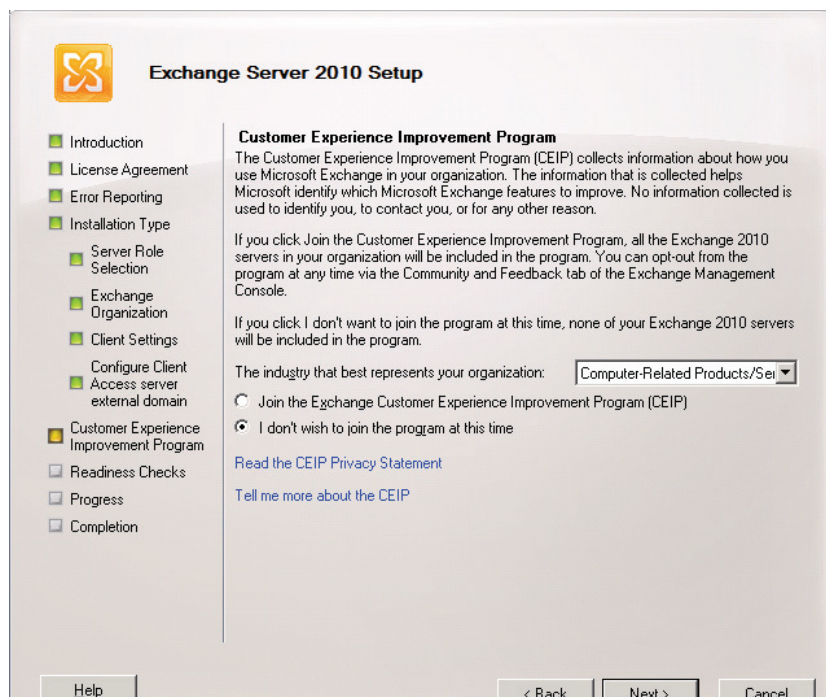
Svare "Yes" her vil opprette en offentlig mappe (public folder) for Exchangeklientene. Dette var noe de gamle Outlookklientene krevde.

I neste steg som vist i Figur 117 får vi mulighet til å legge til tjenester mot internett. For eksempel som web-mail, Outlook Anywhere-tilkobling og andre tjenester.

Siden vi har opprettet domene for dette firmaet, deligerer vi adressen til vår mailtjener som mail.mafoberg.net. Dette blir adressen man bruker utenfra internettet.



Figur 122 - Klient-tilkobling



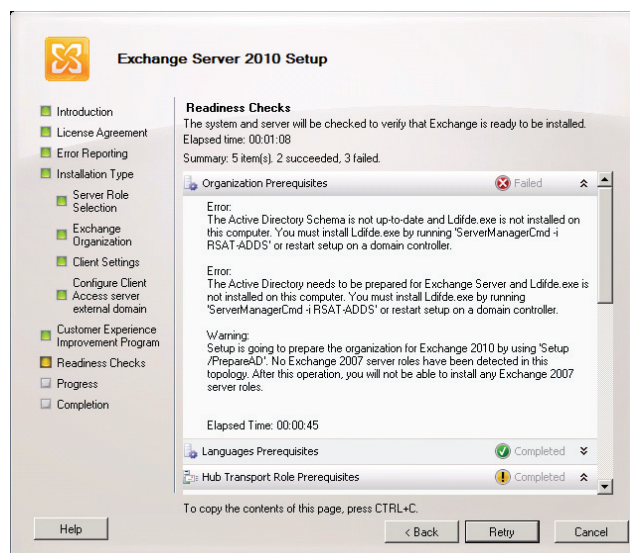
Figur 123 - Microsofts informasjonsinnsamling

Dette er kun informasjonssamling for Microsoft. Både for at de skal kunne utvikle software i retning for kundene sine, og for å forbedre applikasjoner. Vi kan velge hva type industri vi er, men vi velger å ikke delta i dette siden vi ikke ønsker å bruke tid, båndbredde eller ressurser for dette prosjektet. Dette er det mulighet til å forandre senere.

2.6.2.5. Forberedelser for Exchange

Når vi så kjører i gang "Readiness Checks", kjøres det noen tester for sjekking om hva som er forberedt og hva som eventuelt må gjøres for å kunne kjøre Exchange. Da kan vi få både feil som må rettes, krav, advarsler og om tester har gått bra g er fullverdig. Både organisasjonsmessig mot domenekontrollere, klienter, og andre servere som transport foregår igjennom.

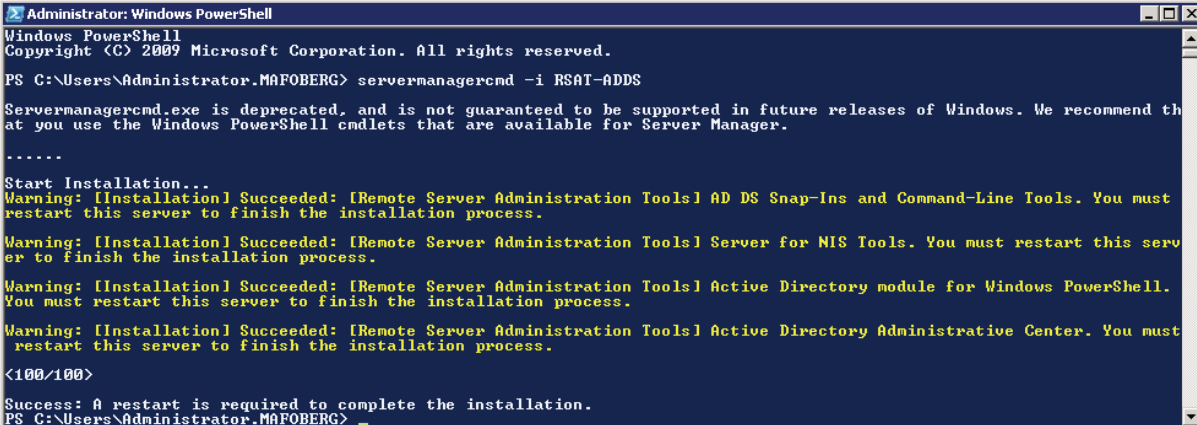
I Figur 119 ser vi at vi har fått noen feil som må rettes opp i. Dette i forhold til domenekontrolleren og schemas Exchange bruker. Det er også flere deler vi må forberede som ligger lenger ned i denne sjekken. Vi skal ta for oss delene hver for seg. Det er også forklart i feilmeldingsboksen hvordan man kan rette på dem.



Figur 124 - Readiness check

Organization Prerequisites

AD Schemas er ikke tilgjengelig på domenekontrolleren og Ldifde.exe er ikke installert på Exchange-serveren. Vi legger til remote-tjenesten på Exchange-serveren ved å kjøre "ServerManagerCmd -i RSAT-ADDS".



```
Administrator: Windows PowerShell
Windows PowerShell
Copyright (C) 2009 Microsoft Corporation. All rights reserved.

PS C:\Users\Administrator.MAFOBERG> servermanagercmd -i RSAT-ADDS

Servermanagercmd.exe is deprecated, and is not guaranteed to be supported in future releases of Windows. We recommend that you use the Windows PowerShell cmdlets that are available for Server Manager.

.....

Start Installation...
Warning: [Installation] Succeeded: [Remote Server Administration Tools] AD DS Snap-Ins and Command-Line Tools. You must restart this server to finish the installation process.
Warning: [Installation] Succeeded: [Remote Server Administration Tools] Server for NIS Tools. You must restart this server to finish the installation process.
Warning: [Installation] Succeeded: [Remote Server Administration Tools] Active Directory module for Windows PowerShell. You must restart this server to finish the installation process.
Warning: [Installation] Succeeded: [Remote Server Administration Tools] Active Directory Administrative Center. You must restart this server to finish the installation process.

<100/100>
Success: A restart is required to complete the installation.
PS C:\Users\Administrator.MAFOBERG>
```

Figur 125 - Servermanagercmd -i RSAT-ADDS

For hver gang vi får beskjed om restart av tjeneste eller server, velger vi å gjøre dette og kjører installasjonen på nytt med Exchange for å se hva videre feil som gjenstår etter dette.

Vi står fortsatt igjen med en advarsel om at domenekontrolleren må kjøre "Setup /PrepareAD" for å kunne samarbeide med Exchange 2007. Men dette er fortsatt bare et prosjekt der vi ikke har tatt med de gamle Exchange serverene som bedriften fortsatt kjører. Dette kan man gjøre på domenekontrolleren før den settes i nettverket med den gamle Exchange-serveren.

Language Prerequisites

Siden vi tok med alle språk i språkpakken lastet ned fra microsoft under installasjonen, går denne greit igjennom.

Hub Transport Role Prerequisites

Siden vi valgte å ta med støtte for gamle versjoner av Outlook, får vi beskjed om at vi bør legge til et konverteringstillegg.¹⁰ Dette tar også hånd om krav i Mailbox Role Prerequisites.

Client Access Role Prerequisites

Webtjenesten IIS7.5 har ikke fått med nødvendige tillegg som Exchange krever. Disse må legges til. Linken på feilmeldingen (Recommendation) viser at en bør installere IIS på nytt. Vi satser derimot på å legge til disse to enhetene direkte til ISS ved å kjøre kommandoene "ServerManagerCmd -i Web-Metabase" og "ServerManagerCmd -i Web-Lgcy-Mgmt-Console" på Exchange-serveren. Se Figur 126 og Figur 127.

Dette vil også ordne kravene for Mailbox Role Prerequisites.

¹⁰ Kan lastes ned fra <http://go.microsoft.com/fwlink/?LinkId=123380>.



```
Administrator: Windows PowerShell
Windows PowerShell
Copyright (C) 2009 Microsoft Corporation. All rights reserved.

PS C:\Users\Administrator.MAFOBERG> ServerManagerCmd -i Web-Metabase

Servermanagercmd.exe is deprecated, and is not guaranteed to be supported in future releases of Windows. We recommend that you use the Windows PowerShell cmdlets that are available for Server Manager.

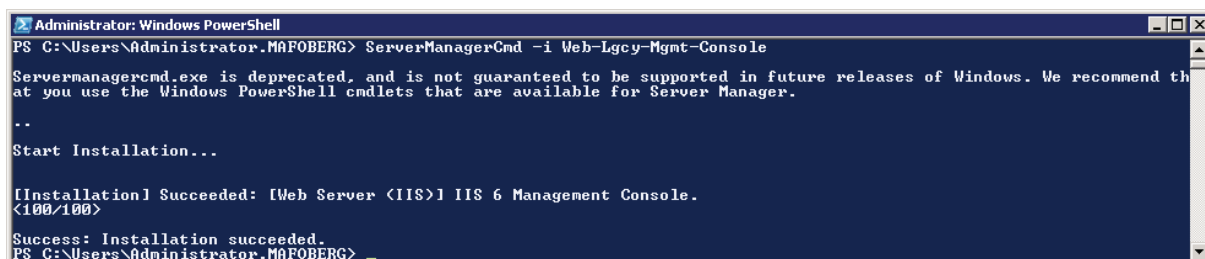
.....

Start Installation...

[Installation] Succeeded: [Web Server (IIS)] IIS 6 Metabase Compatibility.
<100/100>

Success: Installation succeeded.
```

Figur 126 - ServerManagerCmd -i Web-Metabase



```
Administrator: Windows PowerShell
Windows PowerShell
Copyright (C) 2009 Microsoft Corporation. All rights reserved.

PS C:\Users\Administrator.MAFOBERG> ServerManagerCmd -i Web-Lgcy-Mgmt-Console

Servermanagercmd.exe is deprecated, and is not guaranteed to be supported in future releases of Windows. We recommend that you use the Windows PowerShell cmdlets that are available for Server Manager.

..

Start Installation...

[Installation] Succeeded: [Web Server (IIS)] IIS 6 Management Console.
<100/100>

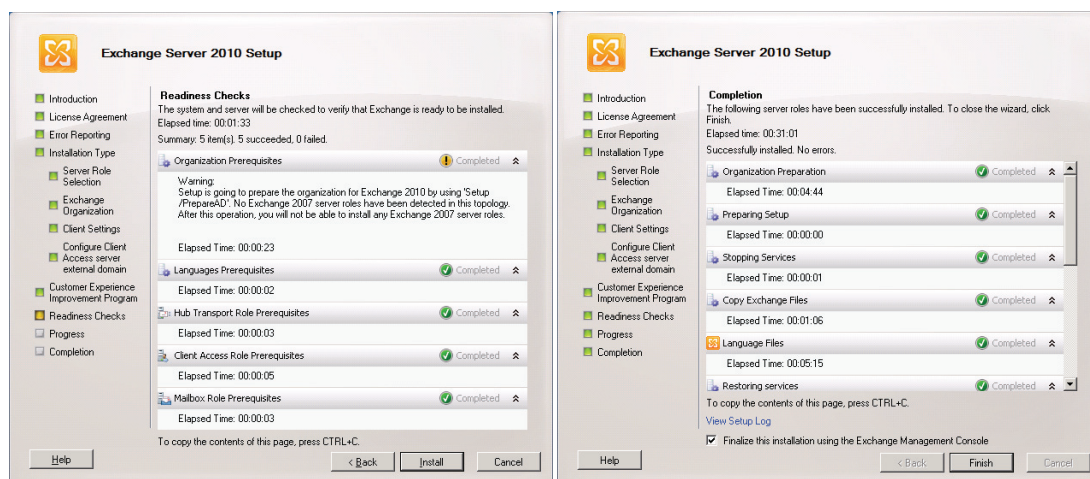
Success: Installation succeeded.
PS C:\Users\Administrator.MAFOBERG>
```

Figur 127 - ServerManagerCmd -i Web-Lgcy-Mgmt-Console

Videre får vi beskjed å starte Net. TCP Port Sharing service som må settes til automatisk oppstart. Dette gjør vi ved å kjøre kommandoene "Set-Service NetTcpPortSharing -StartupType Automatic".

Til slutt får vi samme melding som for HUB Transport Role Prerequisites om konverteringsfilter om vi ikke allerede har lagt til dette filteret fra dette rollekravet.

Når alt er installert vil vi få muligheten til å installere Exchange 2010 og ferdigstille serveren.



Figur 128 - Exchange installasjon og ferdiggjøring

2.6.2.6. Kritiske oppdateringer for Exchange

Til slutt har vi en valgfri mulighet til å oppdatere Exchangeserveren for kritiske oppdateringer. Dette er selvfølgelig noe vi må gjøre for å holde sikkerheten oppe, og for at funksjonaliteten skal være på topp. Dette gjøres fra Microsofts oppdateringsside

(<http://www.update.microsoft.com/microsoftupdate/v6/vistadefault.aspx?ln=en-us>).

Dette foregår sammen med Windows update. Denne bør som regel være satt til automatisk så man slipper å følge med til en hver tid.

2.7. Installere Edge og Forefront Security for Exchange 2010

I dette kapittelet skal vi ta for oss installasjon av Edge, beskrive litt om hva det er og hvordan vi bør benytte oss av det i vår situasjon. Installasjonen av Edge følger med installasjonen av MS Exchange 2010 som vi har lagt opp denne driftsdokumentasjonen for. Vi vil ta i bruk edgerollen i deler av prosjektet, kun for å vise hvordan den installeres og hva som er lønnsomt å sette opp for Edge for å øke sikkerheten.

2.7.1. Hva er Edge?

Edge er en sikkerhetstjeneste for en bedriftens informasjonsutveksling. Tjenesten står mellom epostserverne og utgangen eksterne nett. Den vil fungere som et filter basert på policyer og regler som alt må igjennom før den kommer til bedriftens epostservere. Den vil da kunne filtrere spam og meldinger som er uønsket. Den blokkerer også angrep mot eposttjenestene før de vil komme frem til selve epost-serverene i bedriften.

Edge er ikke pålagt å kjøres for Exchange. Derfor vil denne være en egen installasjon, men likevel vil den ligge sammen med installasjonen for Exchange. Dette fordi tjenesten er basert på Exchange og brukes kun for dette.

Tjenesten installeres på en server for seg selv hvor epost og meldinger må transporteres igjennom. Dette er en form for sikkerhet før meldinger og data kommer inn i bedriftens nettverk og skåner derfor både trafikk og data for bedriften. Edgetjenesten erstatter ikke tjenester som Forefront Security, men øker sikkerheten for bedriften ved at denne installeres ved gitt plassering.

Forefront Security kan installeres direkte på exchangetjenesten og/eller edgeserverene og hjelper til med vasking av epost. Forefront security vil da ta seg av den interne sikkerheten og meldinger ut fra organisasjonen og inn til organisasjonen.

Selv om edgetjenesten er en tjeneste utenfor domenet, vil den likevel ha med Active Directory og domenet å gjøre siden den skal samkjøres med dette for at post skal kunne transporteres. For å få tilgang til informasjon fra Active Directory brukes Active Directory Lightweight Directory service (AD LDS).

Hub-Transport-tjenesten ligger gjerne sammen med en exchangeserver. Denne rollen filtrerer meldinger og koblinger mellom brukere internt i exchangeorganisasjonen og strømmen av epost ut og inn av organisasjonen. Denne rollen kan ta seg av både innhold i meldinger og ruting av dem.

2.7.2. Forberedelse og planlegging

Hvordan vi skal bruke edgetjenesten er viktig å vite om. En edgeserver settes vanligvis ikke opp med full sikkerhet fra starten. Man kan bruke strategi som å sette den med lav filtrering og gjennomgang til å starte med. Etter hvert øker man aggressiviteten på filtrene når man har analyser av post og meldinger man vil unngå at kommer inn i organisasjonen. En annen måte er å teste ut edgetjenesten i et separat nettverk for å se hvilken data som kommer til å gå igjennom serveren. Deretter implementerer man regler og sett i en edgeserver som settes opp i bedriften.

Man må ha klargjort krav for installering av edgetjenesten. Noen krav er...

- Minimum server er Windows Server 2008 Standard eller Enterprise 64-bit med SP2
- Microsoft .NET Framework 3.5
- Windows Remote Management 2.0
- Windows PowerShell versjon 2
- Active Directory Lightweight Directory Services (AD LDS)

Maskinen skal ikke være meldt inn i domenet for bedriften. Men den skal likevel være med i oppslag fra domeneserveren. Dette gjør vi ved å sette navnet til Edge-serveren først på Edge-serveren, så på domenekontrolleren.

Vi må sette navn på Edgeserveren.

Gå inn i **kontrollpanel – System and Security – System**

Klikke på linken **Change settings**

Vi får da opp *systemproperties*.

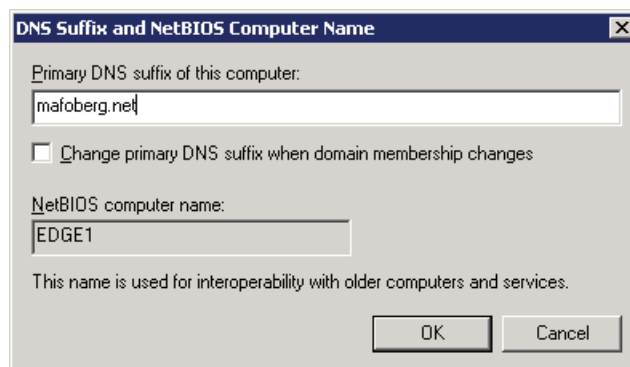
Velge fanen **Computernavn** og klikk **Change**.

Da har vi *Computer Name / Domain Changes*.

Der klikker vi **More**.

Vi har så *DNS Suffix and NetBIOS Computer Name* opp. Der gir vi navnet til domenet som maskinen skal være oppslagsbart igjennom.

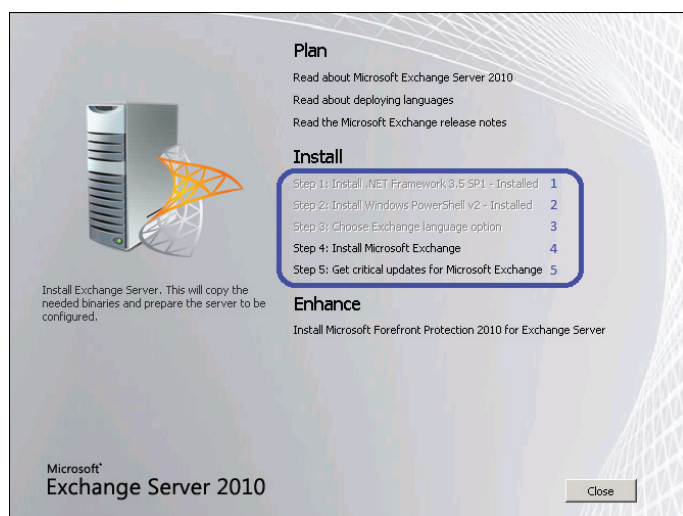
Navnet til maskinen vil derfor bli *edge1.mafoberg.net* i vårt tilfelle hvor all mail skal sendes til.



Figur 129 - Edge DNS Suffix

2.7.3. Installasjon av Edge.

Installasjonen er ganske lik installasjonen av Exchange. Fra installasjonsstarten da vi får skjermbilde av installasjon for Exchange, tar vi forbehold om at vi har gått igjennom steg 1 til 3 før vi kommer i gang med selve installasjonen av Edge. Se Figur 130.



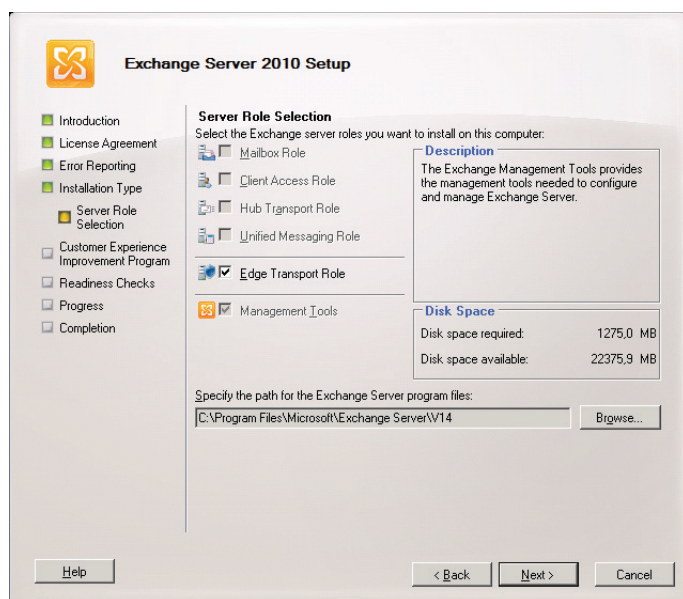
Figur 130 - Edge installasjonsstart

Videre går vi i gang med steg 4 hvor vi starter installasjonen av Exchange hvor vi må velge Edge manuelt.



Figur 131 - Edge Custom valg

Når vi har kommet frem til Installasjonstype, velger vi *Custom Exchange Server Installation*. Hvis vi ser på *Typical Exchange Server Installation* ser vi at vi kun har med de nødvendige enheter for selve Exchange. Men vi skal nå velge ut Edge sine deler.

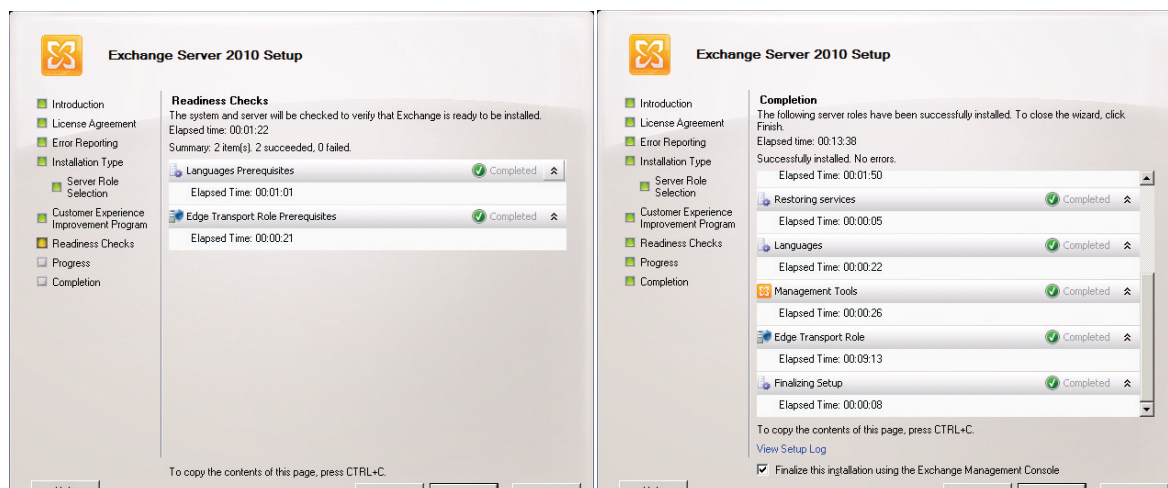


Figur 132 - Edge installasjon valg

Når vi skal velge Edge Transport Role, blir alle andre funksjoner automatisk utilgjengelige. Management Tools er obligatorisk i denne sammenheng slik som i installasjon av Exchange rollen.

Etter dette testes kravene for Edge Transport rollen. Her kommer vi til krav som må opprettholdes. I planleggingen av installasjonen har vi tatt forbehold om disse. Hvis ikke de kravene er gjennomført før installasjonen vil disse stoppe installasjonen her.

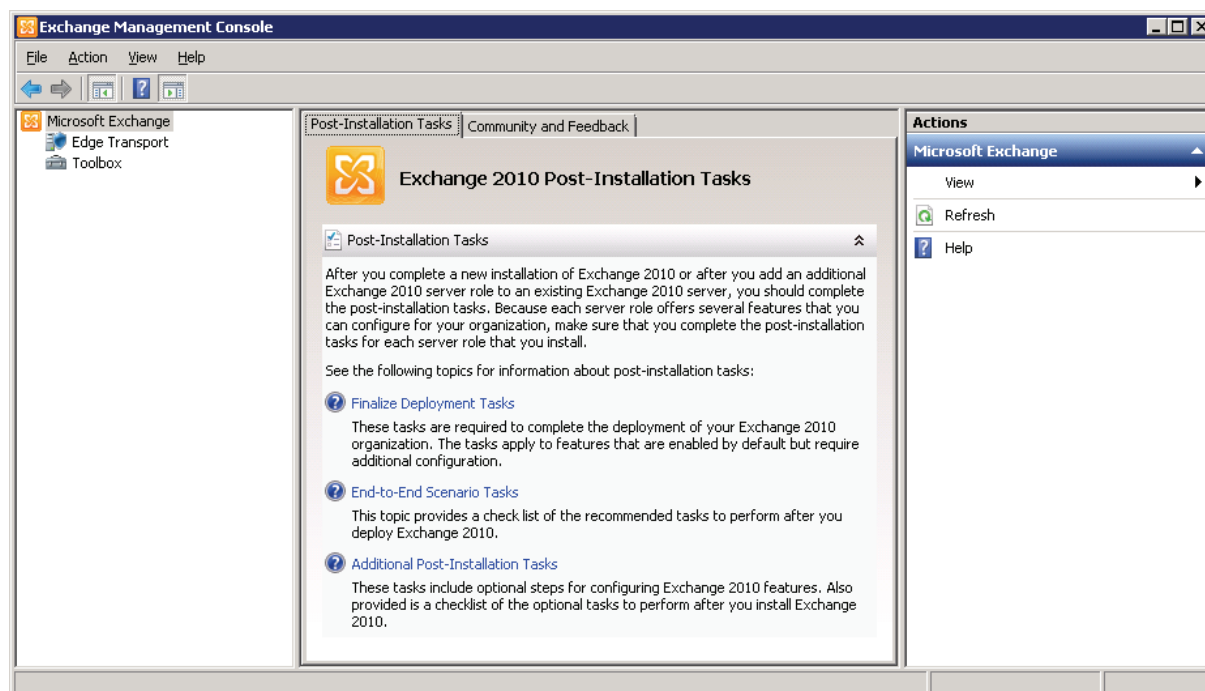
Lesesjekken mot Edge testes og kravene er opprettholdt og vi får en oppsummering før installasjon.



Figur 133 - Edge installasjonsklart og oppsummering

Vi har satt Exchange Management Console til å kjøres rett etter installasjonen. Dette er for å starte med oppsett av filteringsregler mot Exchangeserveren. Automatiske oppdateringer av Anti-spam er satt på fra starten av. Dette gjør at filteret får oppdateringer fra Microsoft når de blir tilgjengelige.

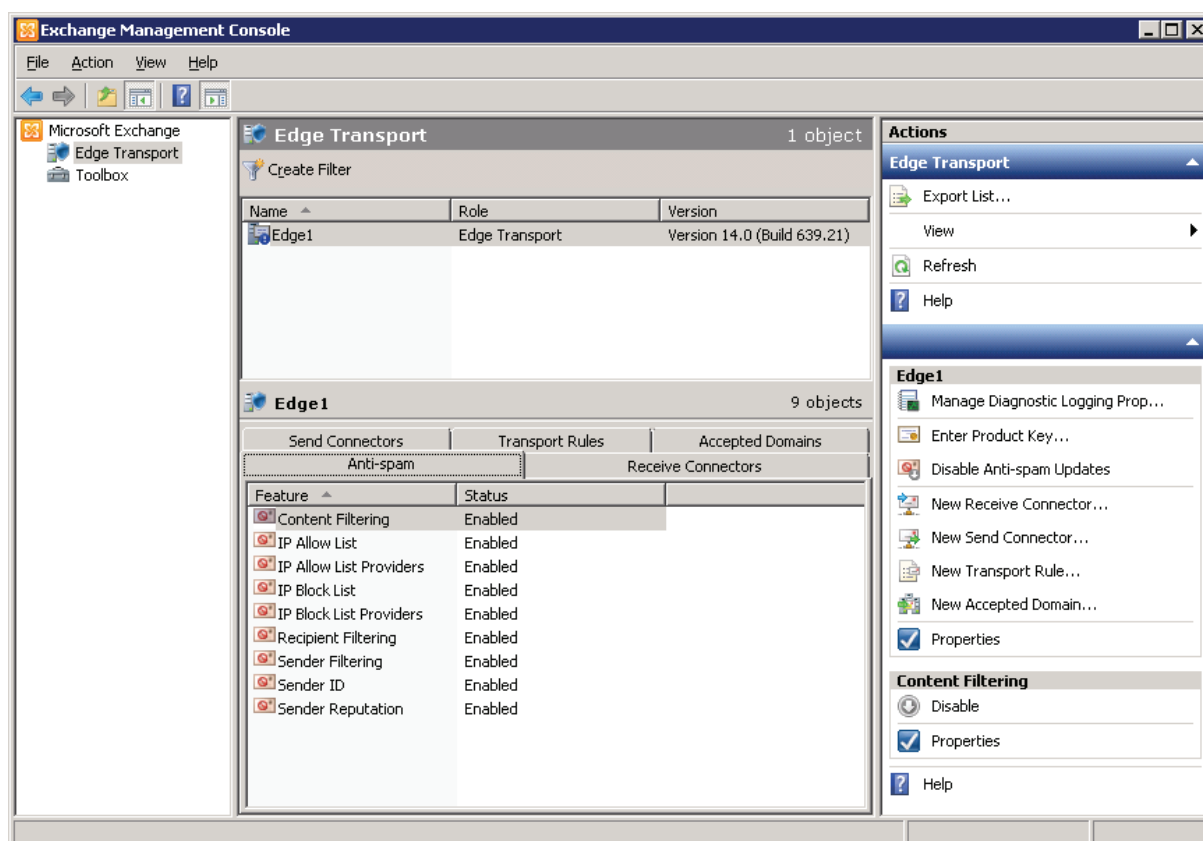
Etter fullføring av installasjonen får vi oppstartsskjerm bilde for Exchange Management Console med edge-snapins som er tilgjengelige.



Figur 134 - Exchange Edge etterkonfigurerings.

Vi vil som i exchangeinstallasjonen kunne gjøre en del konfigurering for å sette opp edge klar til bruk. Under *Finalize Deployment Tasks* får vi tips om å sette opp subscription mellom edgerollen og hubtransportrollen på exchangeserverene. Vi får også beskjed om å sette opp en postmasterkonto for edgeserveren. Dette skal vi komme tilbake til under drift – 3.2. Exchange 2010 - oppsett og bruk.

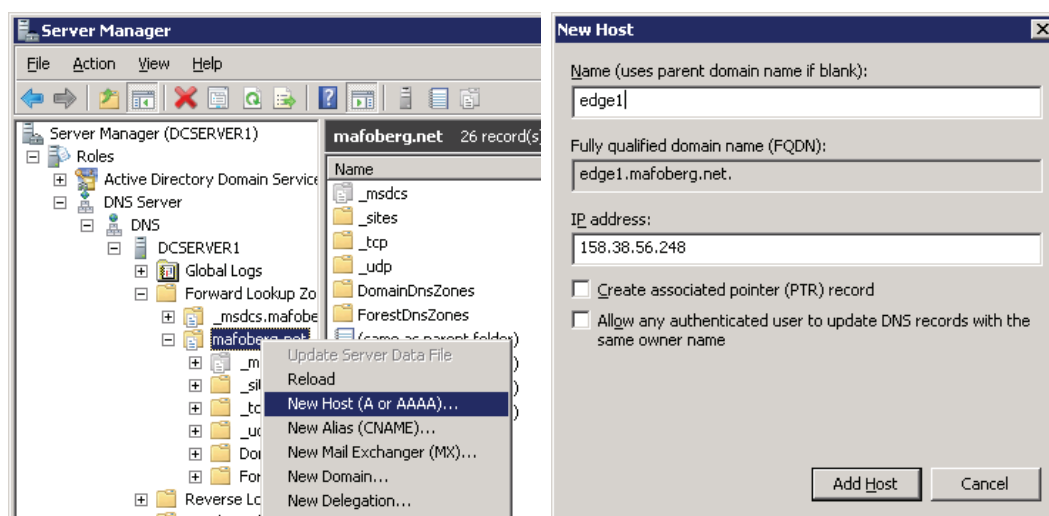
2.7.4. Konfigurering av Edge for Exchange



Figur 135 - Edge konfigurering, Exchange Management Console

Skjermbildet i Figur 135 er konfigureringsskonsollet til Edge. Vi har to underkonsoller. *Edge Transport* og *Toolbox*. *Edge Transport* er konsollet for å sette opp regler for meldingsflyt mellom serverne.

Vi må sette opp en adresse til vår Edge-server i vår DNS-server for at navn mot ipadressen skal eksistere siden vi ikke melder inn Edge-serveren i domenet.

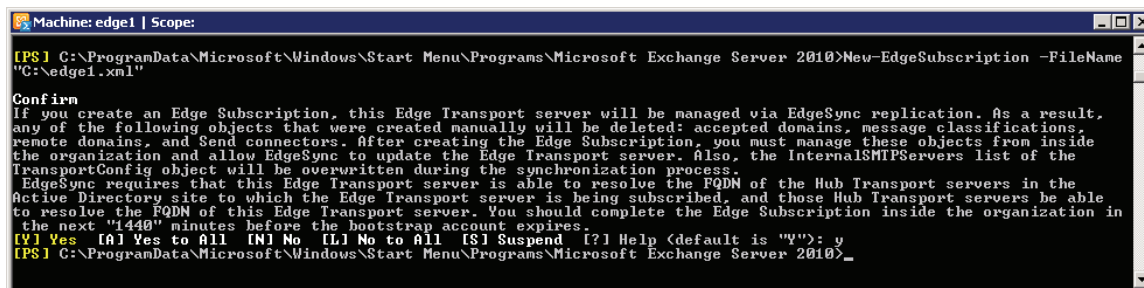


Figur 136 - Edge DNSoppsett 1

Som vist i Figur 136 lages det en ny A-host for vår Edgeserver på DNS-serveren.

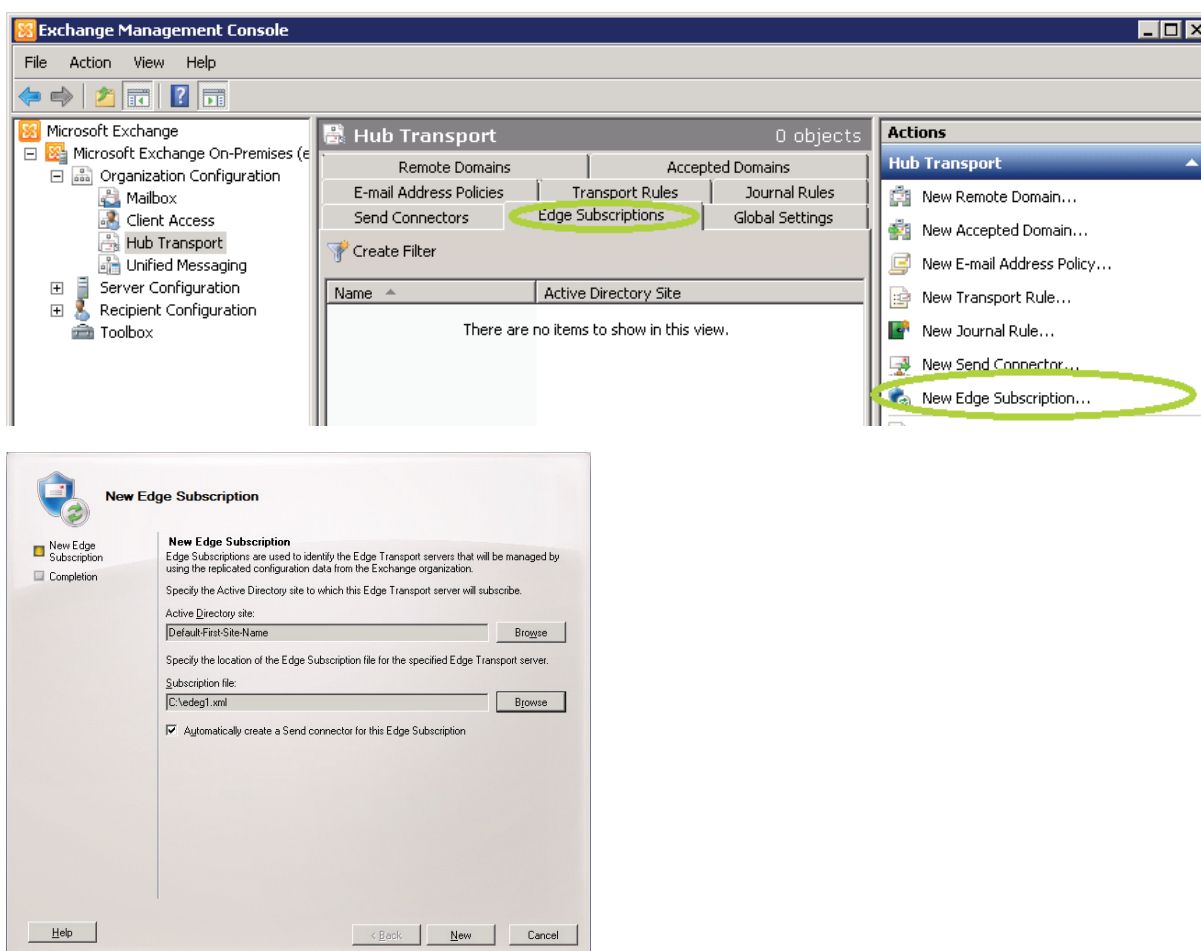
2.7.4.1. Synkronisere Edge1 og Exchange1

For at Edgeserveren skal synkroniseres med Exchangeserveren, må vi blant annet sette opp en *Subscription* som også autogenererer sertifikat slik at disse aksepterer hverandre. For å gjøre dette må vi først lage subscription-informasjon fra Edge-serveren.



Figur 137 - Edge1 subscription eksportering fra Edge1

På edgeserveren kjører vi derfor "New-EdgeSubscription -FileName "C:\edge1.xml" ". Denne kommandoen lager en ny subscription og eksporterer denne til filnavnet *edge1.xml* som vi legger på C:\. Denne subscription-fila må vi få tilgang til fra vår exchangeserver. Enten ved å legge den på en filplassering eller kopiere den direkte. Vi plasserer i vårt eksempel fila i samme lokasjon på Exchangeserveren.



Figur 138 - Edge - Subscription tillegg og synkronisering

I *Exchange Management Console* kan vi legge til en subscription ved å velge *Hub Transport* i konsollet, velge fanen *Edge Subscriptions* og deretter kjøre *New Edge Subscription....* Der velger vi

Brows for å finne frem til aktuell *site*¹¹ som her vil bli standardoppsett fordi vi befinner oss i et lite nettverk. Vi vil kun finne *Default-First-Site-Name*.

Subscriptionfilen som vi laget på Edgeserveren for synkronisering, finner vi på c:\. Vi ønsker å lage Send-connectors for denne subscriptionen automatisk. Kryss av for dette.

Alternativ for oppsett i Exchangeserveren

VI kan sette opp synkroniseringsdataene ved å kjøre akkurat det samme via Exchange Management Shell.

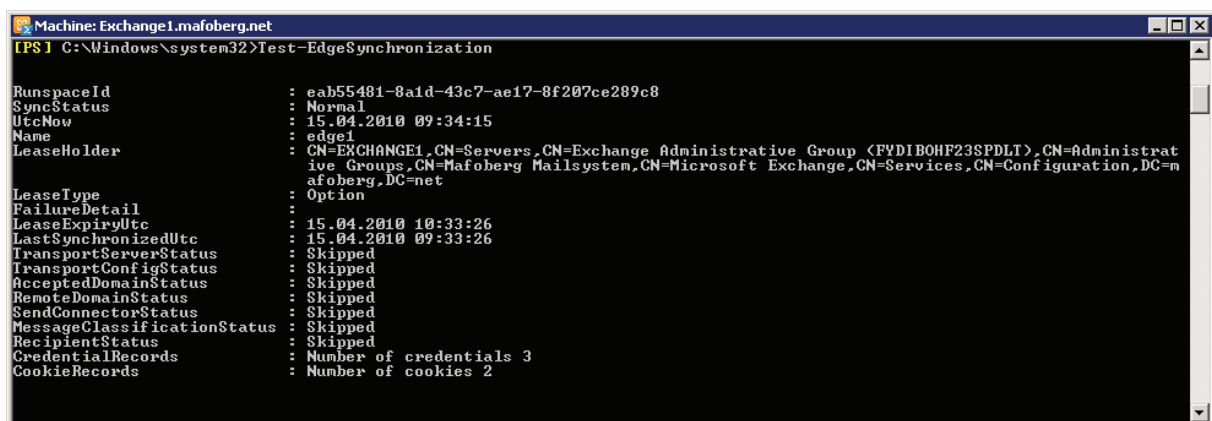
Eksempel:

```
New-EdgeSubscription -FileData "C:\Edge1.xml" -site "mafoberg.net/Configuration/Sites/Default-First-Site-Name" -CreateInternetSendConnector $true - CreateInboundSendConnector $true
```

Denne kommandoen vil lese synkroniseringsfilen edge1.xml, og samtidig sette opp site får vår lokasjon, sendconnector for eksternt nett og sendconnector for internt nett.

Test og synkronisering

Vi kan teste synkroniseringen før vi starter den. Dette gjør vi ved hjelp av Exchange Management Shell. Vi kjører kommandoen "Test-EdgeSynchronization".



```
Machine: Exchange1.mafoberg.net
[PS] C:\Windows\system32>Test-EdgeSynchronization

RunspaceId      : eab55481-8a1d-43c7-ae17-8f207ce289c8
SyncStatus      : Normal
UtcNow          : 15.04.2010 09:34:15
Name            : edge1
LeaseHolder     : CN=EXCHANGE1,CN=Servers,CN=Exchange Administrative Group (FYDIBOHF23SPDLT),CN=Administrative Groups,CN=Mafoberg Mailsystem,CN=Microsoft Exchange,CN=Services,CN=Configuration,DC=mafoberg,DC=net
LeaseType       : Option
FailureDetail    :
LeaseExpiryUtc  : 15.04.2010 10:33:26
LastSynchronizedUtc : 15.04.2010 09:33:26
TransportServerStatus : Skipped
TransportConfigStatus : Skipped
AcceptedDomainStatus : Skipped
RemoteDomainStatus : Skipped
SendConnectorStatus : Skipped
MessageClassificationStatus : Skipped
RecipientStatus  : Skipped
CredentialRecords : Number of credentials 3
CookieRecords    : Number of cookies 2
```

Figur 139 - Edge - Test av synkronisering

I Figur 139 ser vi at SyncStatus er normal. Så lenge denne ikke feiler, kan vi starte synkroniseringen. Dette gjør vi ved å kjøre "Start-EdgeSynchronization".

¹¹ En *site* er et objekt i AD som definerer et hostet nettverk geografisk. Under en site kan vi ha subnett-objekter, Group Policy objekter med mer.

```
Machine: Exchange1.mafoberg.net
C:\Windows\system32>Start-EdgeSynchronization

RunspaceId      : eab55481-8a1d-43c7-ae17-8f207ce289c8
Result          : Success
Type            : Configuration
Name            : edge1
FailureDetails  :
StartUTC        : 15.04.2010 09:34:35
EndUTC          : 15.04.2010 09:34:35
Added           : 0
Deleted         : 0
Updated         : 0
Scanned         : 0
TargetScanned   : 0

RunspaceId      : eab55481-8a1d-43c7-ae17-8f207ce289c8
Result          : Success
Type            : Recipients
Name            : edge1
FailureDetails  :
StartUTC        : 15.04.2010 09:34:35
EndUTC          : 15.04.2010 09:34:35
Added           : 0
Deleted         : 0
Updated         : 0
Scanned         : 0
TargetScanned   : 0
```

Figur 140 - Edge - Starte synkroniseringen

Figur 140 viser begge oppstartene. Både for konfigureringen og mottakerene. Vi ser da at synkroniseringsstatusene ble suksessfulle.

2.7.5. Edge og filtre

Vi tar utgangspunkt i Figur 135 hvor vi har oppstartsbildet for edgeserveren med *Exchange Management Console*. Vi har

- **Edge Transport**
Konsollen inneholder oppsettsmulighetene og oversikten over filterene i Edge-rollen. Her har vi connectors, regler, domeneaksepteringer og Anti-spam regler. Disse skal vi gå nærmere inn på.
- **Toolbox**
Dette er en samling av verktøy som kan brukes for å analysere praksisen av exchangeorganisasjonen og koblinger mellom exchangetjenerene, logger av køer, rutinger, sporinger og meldingsflyt og optimaliseringsverktøy. Disse kan man bruke om man finner exchangetjenestene trege, eller om ting skulle gå galt og liknende.

For hver edge-server er det en rekke regler og filtre som blir satt opp. Hvis det er flere edge-servere i et hierarki, er det ofte slik at de er satt opp med forskjellige regler for å filtrere forskjellig type meldinger og trafikk for neste nivå. Vi kan derfor sette opp flere edgeservere i rekke om man ønsker dette.

2.7.5.1. Edge Transport Toolbox

Det er generelt 5 tabber i transportrollen. *Anti-spam*, *Receive Connectors*, *Send Connectors*, *Transport Rules* og *Accepted Domains*. Disse inneholder oppsettet for edge-transportrollen.

- Anti-spam inneholder en rekke filtertyper. Disse kan slås på og av etter ønsket mengde filtrering.
 - Content Filtering – Inneholder direkte fraser av innhold i hele meldinger. Her vil det enten bli tillatt eller blokkert for gjennomgang.
 - IP Allow List – Liste av egne oppsett av ipadresser man gir tilgang for at skal kunne sende mail til vår organisasjon. Denne listen gir uavbrutt tilgang til våre servere, fordi vi stoler på disse.
 - IP Allow List Providers – Dette er koblinger til organisasjoner som opprettholder lister av ipadresser som tillates å sende meldinger til oss. Slike Real-time

organisasjoner kan utveksle seg imellom oppdateringer så ofte som en gang i timen. Dette gjør det enklere å holde oversikt over meldingsservere som er gode og anbefales.

- IP Block List – Liste av servere vi selv har satt opp ikke skal ha tilgang til oss. Dette kan være hvis vi selv har funnet servere som sender meldinger vi ikke ønsker.
- IP Block List Providers – Slik som Providers for *IP Allow List* hvor eksterne aktører holder oppdaterte lister over ipadresser som bør blokkeres.
- Recipient Filtering – Liste over mottakere i form av brukere eller domener som man ønsker å blokkere innkomne meldinger til.
- Senders Filtering – Blokkerer utgående meldinger fra brukere, domener eller hele domenhierarkier. Man kan velge å merke meldingene om man ønsker dette.
- Sender ID – Man kan velge handling for hva som skal skje om ID på sendere ikke stemmer. Dette er et mottiltak for spoofing¹².
- Sender Reputation – Gradering av brukere for blokkering hvis nok spam sendes fra denne brukeren.
- Recieve Connectors inneholder koblinger for ruting av meldinger inn i organisasjonen. Når en edgserver er synkronisert, vil den automatisk sette opp en standard innkommende kobling.
- Send Connectors inneholder koblinger for ruting av meldinger

2.7.6. Forefront Protection Systems

Forefront er en sikkerhetsapplikasjon fra Microsoft som tar seg av filtrering av spam, virus, malware og sikkerhet for direktemeldinger. Forefront kommer i forskjellige versjoner til beskyttelse for forskjellige typer applikasjoner.

Forefront Protection bruker eksterne leverandørers motorer og databaser for søking av malware og virus. Disse vil kunne sjekkes status på under *Engines* i kontrollpanelet. Man kan også etter hvert velge hvor mange motorer man vil bruke samtidig, alt etter hvor mye maskinressurser man ønsker å bruke.

- *Forefront Client* brukes for klienter. Den tar seg av den lokale vaskingen av inntregningene.
- *Forefront Security* er utviklet for Sharepoint og for Office Communication Server.
- *Forefront Threat Management Gateway* er rutingsoftware for Microsoftservere som filtrerer og stopper inntrengninger mot organisasjonens nettverk.
- *Forefront Unified Access Gateway* er en remoteapplikasjon for å sikre trafikken for remoting av applikasjoner i organisasjonens nettverk så vel som DirectAccess funksjoner med vpn.
- *Forefront Protection* er serverprogramvare for Exchange. Den tar seg av vasking av virus, ormer, spionprogrammer og annet innhold i epost og meldinger. Denne kalles nå *Forefront Protection for MS Exchange 2010* og er utviklet fra tidligere versjoner av Forefront Security. *Forefront Protection Online* er en tjeneste fra Microsoft i Protectionpakken som er et lag fra Microsoft hvor epost kan filtreres igjennom. Dette vil senke litt tid på leveranse, men utgjør en ekstra ekstern vask for epost og meldinger.

¹² Spoofing er uttrykk for å bruke en annens identitet for å kunne få tilgangen eieren av denne identiteten har.

Forefront Protection 2010 for Exchange Server krever ekstra ressurser av hardware. Derfor er det lønnsomt å se over maskinvare man kjører Exchange Server på for mulig oppgradering.

Vi kommer ikke til å opprettholde kravene i dette prosjektet på grunn av manglende ressurser. Men det er likevel mulig å kjøre. Kun tregere prosessering vil oppstå på grunn av dette.

Prossessor	X64 arkitekturs prosessor. Intel Xeon eller Pentiumfamilie eller AMD 64bits familier.
Serversoftware	<ul style="list-style-type: none"> • Microsoft Windows Server 2003 SP2, Microsoft Windows Server 2008, eller Microsoft Windows Server 2008 R2 • Microsoft Exchange Server 2007 SP1 eller senere, eller Microsoft Exchange Server 2010
Minne	2GB minne ekstra i tillegg til hva Exchange Server 2010 bruker. Ut over dette kreves planlegging for hvordan servere er koblet sammen.
Harddisk	2GB ledig diskplass.
Server	4-kjernes CPU med 2.0GHz klokkefrekvens på hver. Dette for at motorene skal kunne kjøre realtime uten avbrudd fra andre prosesser. Dette øker også hastigheten på gjennomstrømning av epost og beskjeder.
Andre krav	<ul style="list-style-type: none"> • Microsoft .NET Framework 3.0 SP 1 Windows Communication Foundation or Microsoft .NET Framework 3.5. • Microsoft Chart Controls for Microsoft .NET Framework 3.5. Disse vil bli installert om de ikke er installert på forhånd. • Windows PowerShell 1.0 (PS2 er installert med Server 2008 R2) • Client Access Server (CAS). Denne er krevd dersom man skal kjøre en manual scan på Exchange Server 2010.

Tabell 8 - Forefront maskinkrav

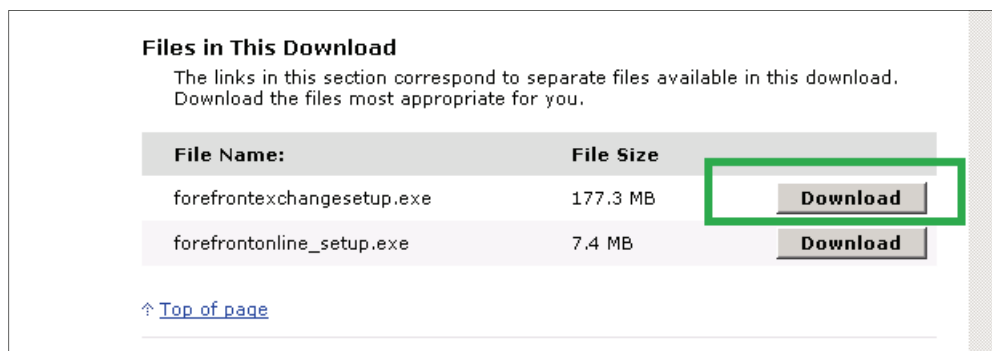
2.7.6.1. Forefront Protection installasjon

Vi kobler til Exchange Server 2010 installasjonsmedium og starter installasjonen på vanlig måte. Enten ved autorun eller kjøre setup.exe. Vi får installasjonsskjermen for Exchange Server 2010.



Figur 141 - Forefront installasjon oppstart 1

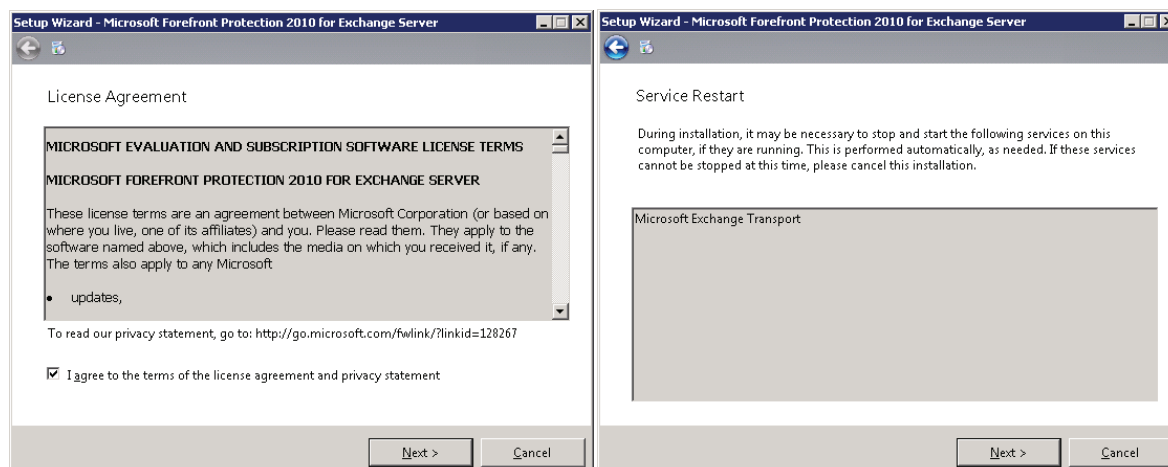
Nederst har vi *Enhancements* for Exchange Server 2010. Dette er en link mot Microsofts sider hvor vi kan laste ned installasjonen for Forefront Protection. Der må vi trykke knappen Download for så å måtte logge inn til Technet med en Windows Live ID. Etter dette får vi tilgang til å laste ned Forefront. Men her får vi to muligheter.



Figur 142 - Forefront nedlasting

Forefrontexchangesetup.exe – Hele pakken med Forefront Protection 2010 for Exchange Server 2010. Dette er den vi skal bruke. Den inneholder motorer, konsoll for styring og Powershellscript.

Forefrontonline_setup.exe – En tilkoblingsenhet for oppsett mot Microsofts onlinetjeneste av Forefront. Denne skal vi ikke ta med siden den kun er en prøve for en kort periode.

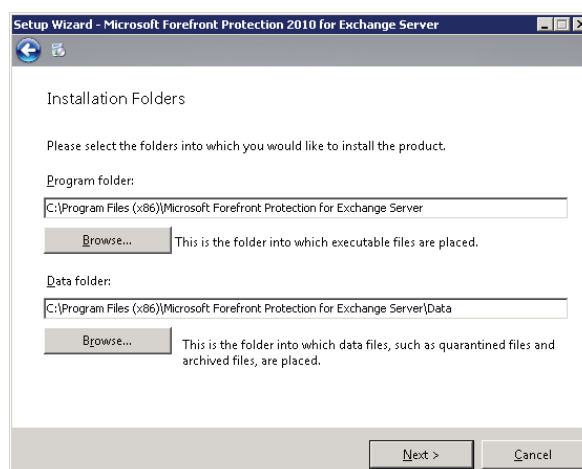


Figur 143 - Lisens og services

Vi får beskjed om lisens. Denne leser vi igjennom og aksepterer så lenge betingelsene er ok.

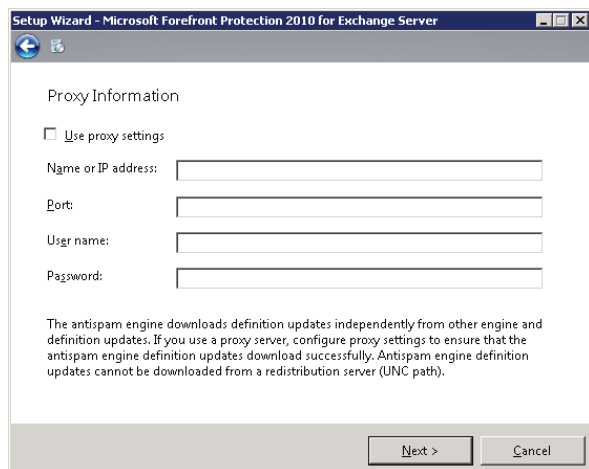
Deretter kreves det en restart av tjenester. Disse vil bli listet opp som vist i Figur 143 til høyre. Hvis vi ikke avslutter denne på forhånd, gjør installasjonen det for oss. Denne vil automatisk bli startet opp igjen etter installasjonen.

Etter dette får vi beskjed om plassering av programvaren og data til den.



Figur 144 - Program og dataplassering

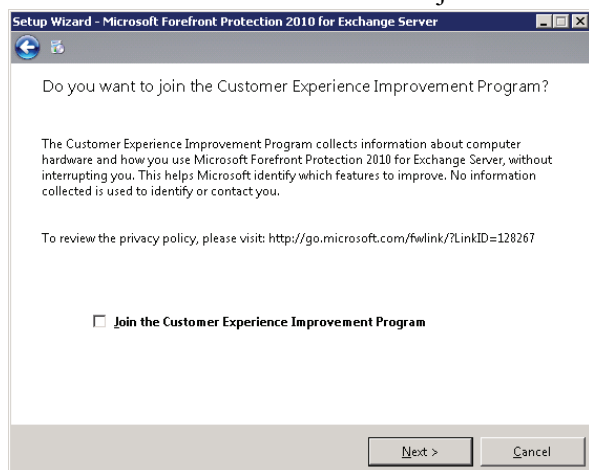
Vi kjører standard plassering. Dette gjør det enklere å kjøre installasjon og lettere å finne igjen om vi skulle ha behov for hjelp senere da det lett kan refereres til standard plassering.



Figur 145 - Proxyinnstillinger

Videre får vi spørsmål om vi vil aktivere spamfilter med en gang eller vente til senere. Hvis vi aktiverer med en gang, vil Forefront starte oppdateringer av spamfiltrene med en gang.

Hvis vi velger å sette opp spamfilter senere må vi aktivere spammotoren i programmet. Da vil det kun bli installert men ikke oppdatert noen database før vi aktiverer denne funksjonen.



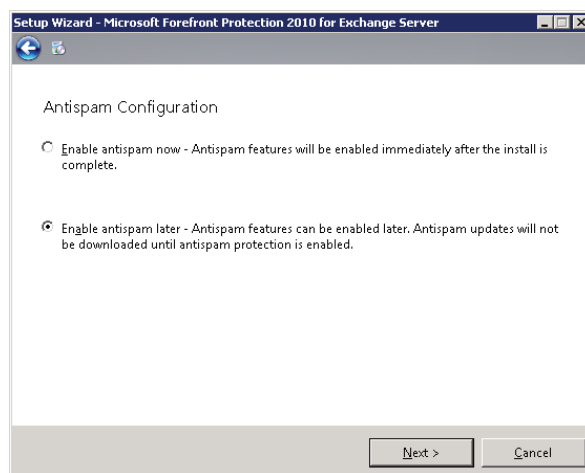
Figur 147 - Erfaringsmedelelse til Microsoft

Vi velger ikke å gjøre dette i vårt testmiljø.

Deretter får vi en kort oppsummering av installasjonen.

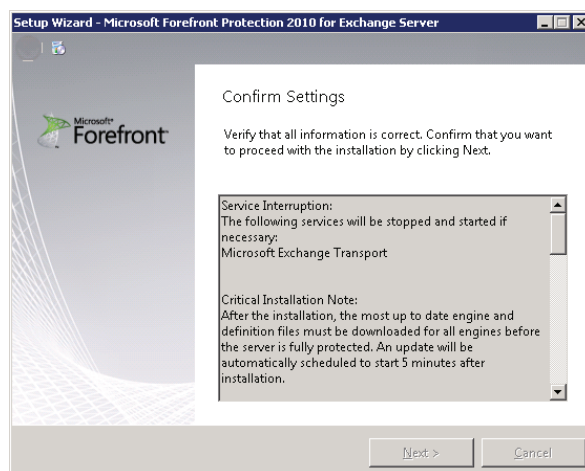
Vi får beskjed om proxyinnstillinger. Dette bruker vi ikke i dette prosjektet. Hvis det skulle være en slik caching-server i miljøet, bør man angi disse innstillingene med en gang for at Forefront Protection skal kunne koble seg til Microsofts oppdateringsbase med en gang, og tilpasse seg nettverket.

Hvis man ikke vet dette med en gang, kan man fortsette installasjonen uten å sette det opp. Det mulig å sette opp dette i programmet senere.



Figur 146 - Spamaktivering

Vi kan velge å bli med i Microsofts forbedringstjeneste. Dette vil gjøre at vi meddeler data av spam og liknende til Microsoft for forbedring av tjenestene.



Figur 148 - Oppsummering av installasjon



Figur 149 - Forefront Protection fullførelse

Etter installasjon vil vi få beskjed om aktivering. Dette kan være lurt å gjøre med en gang så lang man har anskaffet lisens for dette.

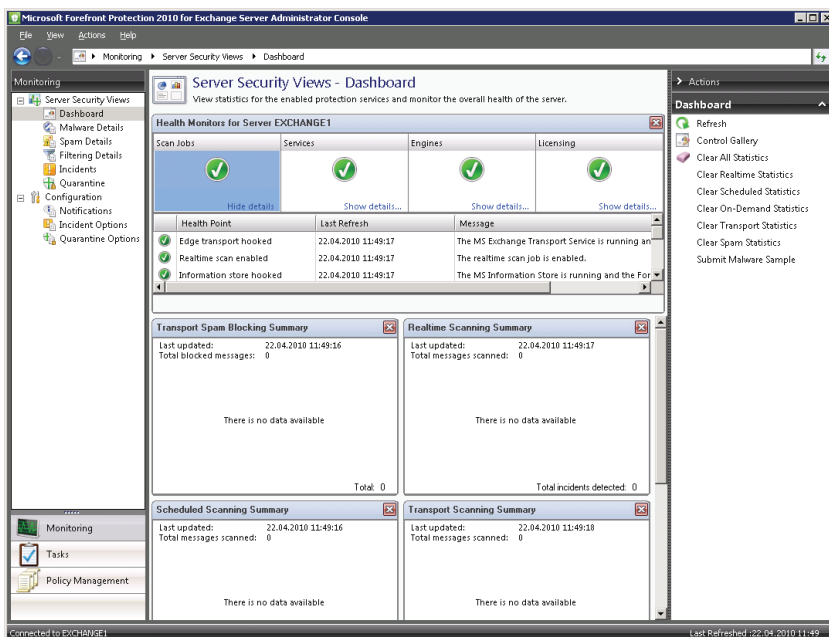
Deretter får vi opp Startbilde til Forefront Protection 2010.

2.7.6.2. Oppstart og bruk av Forefront

Siden vi under installasjonen ikke aktiverte antispamfiltreringen kan vi gå direkte til dette under *Policy Management – Online Protection*. Dette er hovedsakelig for å starte oppdatering av spamfilter med en gang.

I høyre fane er det oppgaver som kan gjennomføres for hvert verktøy som kan brukes i hver hovedfane. Man kan også finne igjen disse i vinduene.

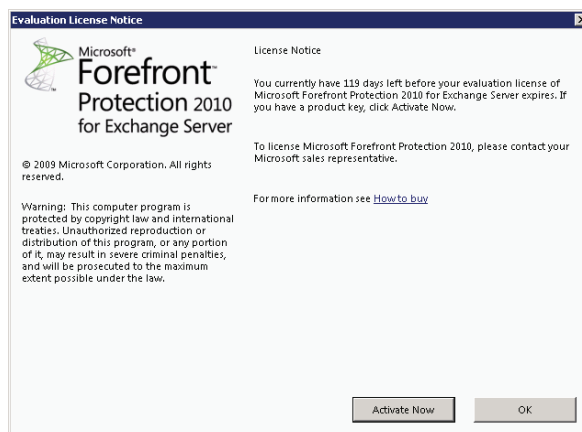
Monitoring



Figur 151 - Forefront Dashboard - Monitoring

Vi kan nå velge å kjøre Forefront Online Protection i tillegg til Forefront Protection 2010 som vi allerede har installert.

Dette gjør vi ikke i vårt prosjekt.



Figur 150 - Aktivering

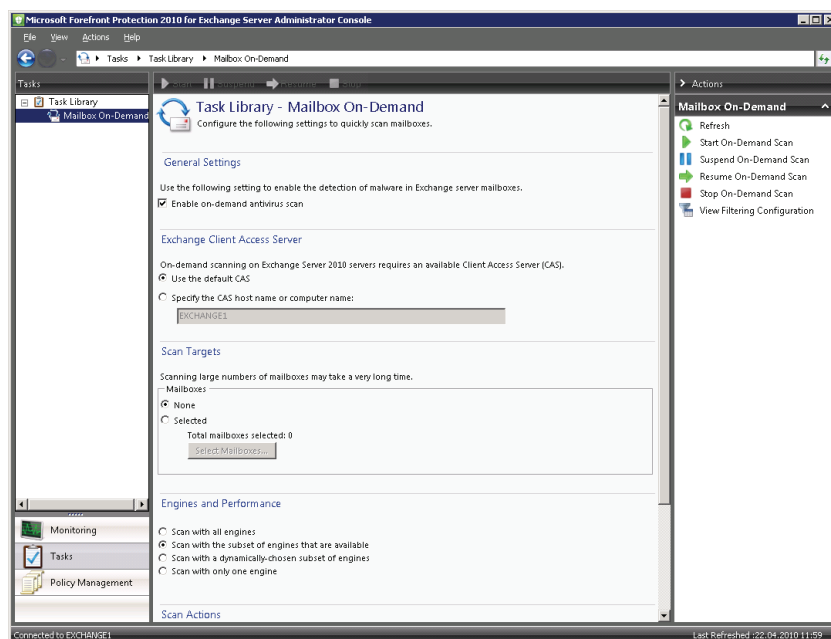
Det første vi får opp er *Dashboard* i Forefront Security. Her har vi tre hovedfaner. Disse ser vi nede i venstre hjørnet i Figur 146.

I *Monitoring*-fanen har vi oversikt over detaljene for hva som blir oppfanget av Forefront Security.

Quarantined files havner i egen kasse hvor man kan administrere om man skal slette eller gi tilgang til å hente element likevel. Dette kan være hvis noen har fått vedlegg som likevel er akseptabel. For eksempel i utviklingsmiljøer hvor programmerere tester

software og program blir filtrert som skadelige, men likevel er under kontroll. I konfigurasjonen kan man sette opp hva krav til automatisk fjerning og liknende skal gjøres.

Tasks



Figur 152 - Forefront Dashboard – Tasks

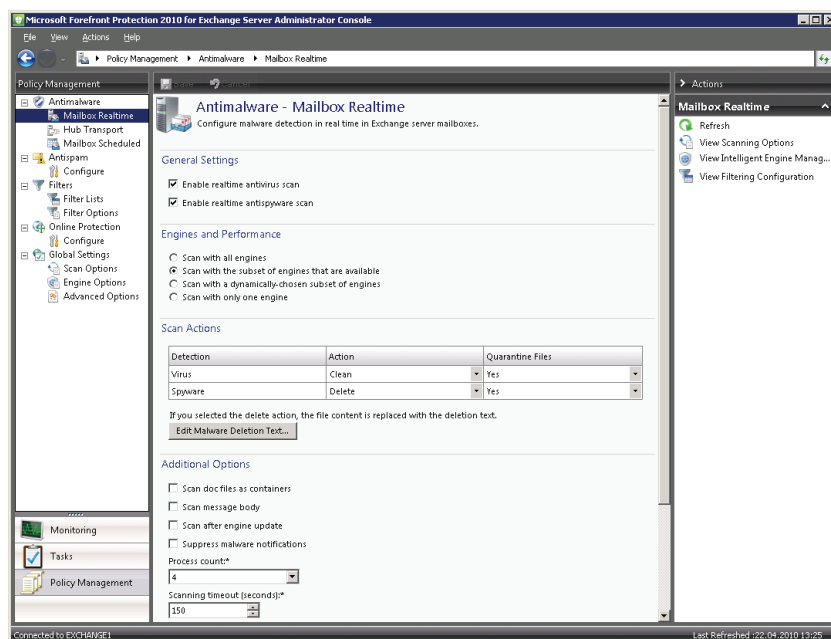
I *Tasks* har vi kun en oppgave vi kan utføre. Det er direktesøk. *Mailbox On-Demand* har flere valg.

I fanen *Tasks* har vi mulighet til å søke egne manuelle søk direkte på serveren. Vi kan velge server server vi skal scanne om vi har flere servere med mailbokser, eller den aktuelle serveren.

Under *Targets* kan vi velge ut mailbokser på serveren vi ønsker å søke i dersom vi spesifikt mistenker eller vet om en postkasse som kan være utsatt for virus og malware. Hvis det er mange store postkasser, vil det ta tid å søke igjennom mange postkasser.

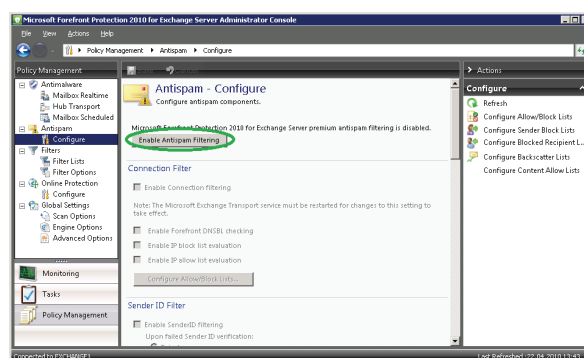
Under *Scan Actions* kan vi editere tekst som skal settes inn med filer som er slettet som vedlegg.

Policy Management



Figur 153 - Forefront Dashboard – Policymanagement

- Antimalware – Mailbox Realtime – styring av malware som skal fjernes. I *Mailbox Realtime* er det oppsett for å søke etter malware på direkten. Dette gjelder for virus og spionware.
- Antimalware – Hub Transport – oppsett av søking via Hub Transport tjenesten. Vi kan definere hvor kraftig det skal søkes. Enten delvis bruk av motorer, bruk av alle motorer eller bare én. Microsoft foreslår 250MB ledig minne for hver motor som skal kjøres. Det er lurt å samarbeide med Forefront på edgeserver. Da er det lettere å fordele bruk av maskinressurser for scanning.
- Antimalware – Mailbox Scheduled – oppsett av tidsrutiner for søking i mailbokser. Det er satt til standard tid men har mulighet for spesifisert søk.
- Antispam – Configure – som vist i Figur 149 er det en enkel ting vi kan gjøre. Aktiver Antispamfiltreringen.
- Filters – oppsett for filter av innhold i meldinger.
- Online Protection – kobling til Microsofts onlinescanning.
- Global Settings – Scan Options – oppsett for hva som skal scannes. Inngående eller utgående og hvilke mailbokser spesifisert.



Figur 154 - Antispam filter aktivering (enable)

- Global Protection – Engine Options – her setter vi proxyinstillinger om vi hoppet over dette i installasjonen. Også *Universal Naming Conversion (UNC)* kan aktiveres her.
- Global Protection – Advanced Options – her settes krav for hvordan Forefront skal oppføre seg. Oppsett for logging og koblinger gjøres her.

2.8 Webgrensesnitt for Powershell

I forbindelse med forstudiet til dette prosjektet, var det bestemt at det skulle implementeres en prototype på en webløsning for administrering av serverne i domenet. I og med at bedriften har en størrelse på 444 personer, hvor hovedfokuset er på å vedlikeholde systemer for kunder og egne interne systemer, ser bedriften det hensiktsmessig å innføre noen automatiske rutiner for overvåkning av serverparken. Dette bør implementeres på et fellesområde, og det har derfor blitt bestemt at serverne skal overvåkes fra et felles web-basert grensesnitt.

Det understrekes at dette prosjektet hovedsakelig består av oppsett og konfigurasjon av bedriftens servere. Dette grensesnittet kommer som et supplement, og er ment som en basis for videre utvikling. Dette grensesnittet er derfor ikke komplett på noen måte, men viser hva som er teknisk mulig.

2.8.1 Prerequisites

Dette systemet er bygd på et sett med opensource-programmer kjørende på Windows-plattformen. Det tar i bruk funksjonaliteten i Active Directory og powershell, samt at utlistering av informasjon benytter seg av Powershell-modulene for Exchange og System Center Virtual Machine Manager. Installasjon av disse er forklart andre steder i dette dokumentet.

Vi skal benytte oss av en ferdig pakke kalt Xampp for webserverfunksjonaliteten. Nedenfor er ei liste over all programvare som må være installert for at webgrensesnittet skal fungere

Programvare	Type	Krever
Apache	Web Server	Xampp
PHP	Scriptespråk for web	Xampp
Powershell	Shellscript for systeminteraksjon	Oppdatert Windows Server
Exchange Snappin	Scriptemodul for Powershell	Powershell, Exchange
Scvmm Snapping	Scriptemodul for Powershell	Powershell, Scvmm
jQuery	Javascript-rammeverk	Nettleter med Javascript påslått

Tabell 9 - Oversikt over programkrav

Et arkiv som inneholder all nødvendig PHP- og Javascriptkode kommer til å bli lagt ved dette dokumentet som et vedlegg.

2.8.2 Hvordan systemet fungerer

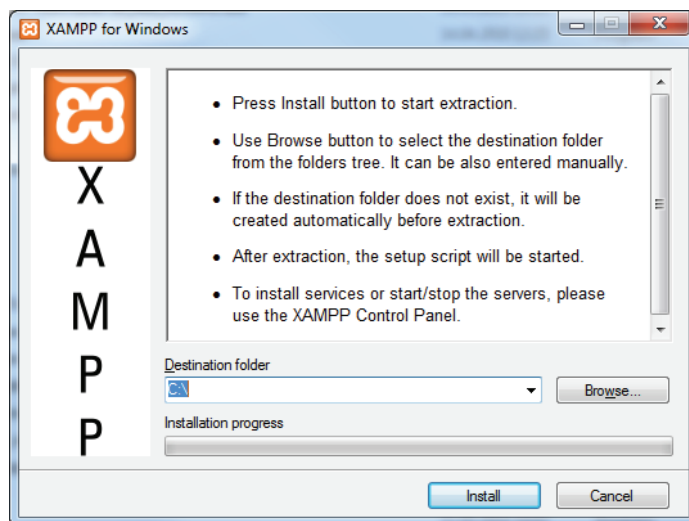
Dette systemet benytter seg av html og css for å vise grensesnittet for brukerne. Dette grensesnittet kjører vi så på Webserveren Apache gjennom Xampp. Grensesnittet er delt inn i logiske separate faner, slik at informasjon om eposter holdes separat fra statistikk om serverstatistikk. Selve utlisteringene hentes fra en cache-fil, men holdes oppdatert regelmessig av en oppdateringsprosess som går i bakgrunnen.

Denne oppdateringsprosessen er et javascript ajax-kall som kommuniserer med webserveren. Webserveren bruker så Scriptespråket PHP for å kommunisere med Powershell. Det er Powershell som står for å liste ut all nødvendig informasjon før den blir sendt tilbake til php og deretter tilbake til nettsida som oppdateres med nye verdier ved hjelp av dynamisk innlegging fra javascript.

2.8.3 Installasjon

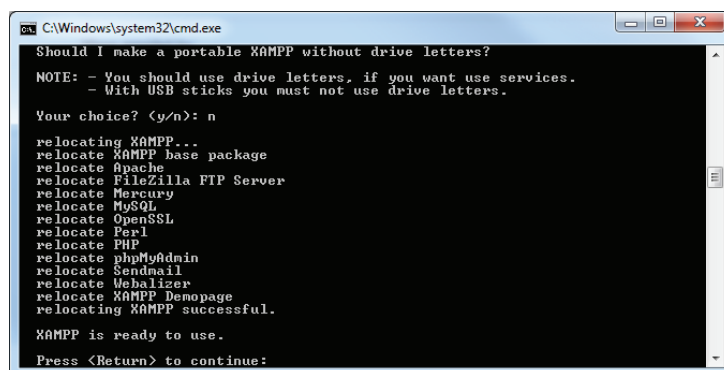
Vi skal under denne seksjonen kun fokusere på installasjon av Xampp og den nødvendige koden for å få nettsida opp og kjøre. Dette fordi det antas at leseren har gått igjennom nødvendig installasjon av Exchange og System Center Virtual Machine Manager(SCVMM) slik som dokumenter i seksjon 2.4 - 2.7. Videre kan en lese om oppsett og syntaks i Powershell i seksjon 4 - Script av Powershell, samt bruk av Snappinmodulene til henholdsvis Exchange og SCVMM i seksjon 4.6 – Powershelladministrasjon av Exchange og seksjon 4.7 -Powershelladministrasjon av SCVMM.

XAMPP er en samling av diverse serverprogramvare, slik som apache webserver og scriptespråket php. XAMPP-pakken er tilgjengelig for en rekke operativsystem, inkludert Windows. Den lastes ned fra nettsida til apachefriends: <http://www.apachefriends.org/en/xampp-windows.html> . Vi har valgt å installere xampp-win32-1.7.3.exe . Dette er en selvutpakkende installasjonsfil. Som vist i Figur 150 er destination folder satt til "c:\".



Figur 155 - Installasjon av XAMPP

Programmet vil deretter installere seg i ei undermappe "xampp", slik at den fulle dokumentstien til filene som skal vises på nett blir "C:\xampp\htdocs"

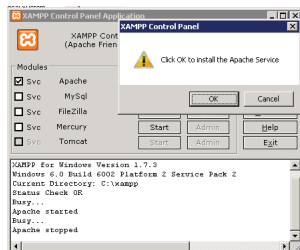


Figur 156 - Oppsett av XAMPP

Etter installasjon vil et kommandoskall dukke opp, som vil spørre deg om noen valg. Standardvalgene er for dette formålet godt nok; vi velger ikke å endre disse.

2.8.3.1 Oppsett av Auto oppstart

XAMPP vil nå installere et kontrollpanel kalt XAMPP Control Panel Application. Med dette kan en styre alle tjenestene som den tilbyr. I figur 152 har vi haket for boksen ”Svc” foran ”Apache”. Dette lar oss opprette en service som starter sammen med maskinen, uavhengig av om det finnes brukere som er innlogget.



Figur 157 - XAMPP kontrollpanel

Denne servicen blir hetende noe sånt som Apache2.2. Den er i utgangspunktet ikke satt til å starte automatisk, så dette må en gjøre fra ”services.msc”. Alternativt kan dette gjøres direkte fra Powershell ved å kjøre følgende kommando:

```
get-service "apache*" | Set-Service -StartupType Automatic | start-service
```

2.8.3.2 Oppsett av Selve grensesnittet.

Vedlagt dette dokumentet er arkivet ”webgrensesnitt_htdocs.zip”. Dette inneholder all kode som er nødvendig for å få et funksjonelt grensesnitt. Mappestien c:\xampp\htdocs vil allerede inneholde noen filer. Disse må slettes og byttes ut med innholdet i arkivet.

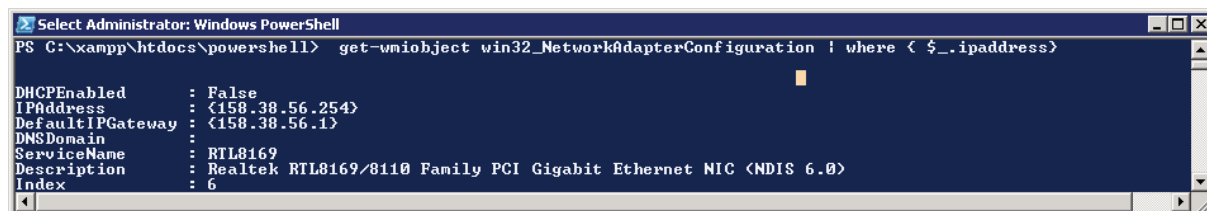
Nå vil det ligge en del php- og powershellfiler i htdocs. I tillegg må mappa c:\xampp\htdocs\powershell\cache gjøres les-og skrivbar for den brukerkontoen som webserveren kjøres som. Dette er i vårt tilfelle brukeren ”mafoberg.net\bjorninge” som er satt til å ha nødvendige administrasjonsrettigheter, slik at en kan benytte seg av powershell remoting og funksjoner for å hente ut lastkalkulasjonsinformasjon fra de lokale serverne som benyttes i prosjektet.

2.8.3.3 Oppsett av DNS

Vi ønsker nå å kunne benytte en adresse, slik som for eksempel i vårt tilfelle ”web.mafoberg.net” for å kunne nå denne webtjenesten. Da mafoberg.net er et domene vi kontrollerer, kan vi legge til denne adressen i vår dns-server som kjører på en domenekontroller.

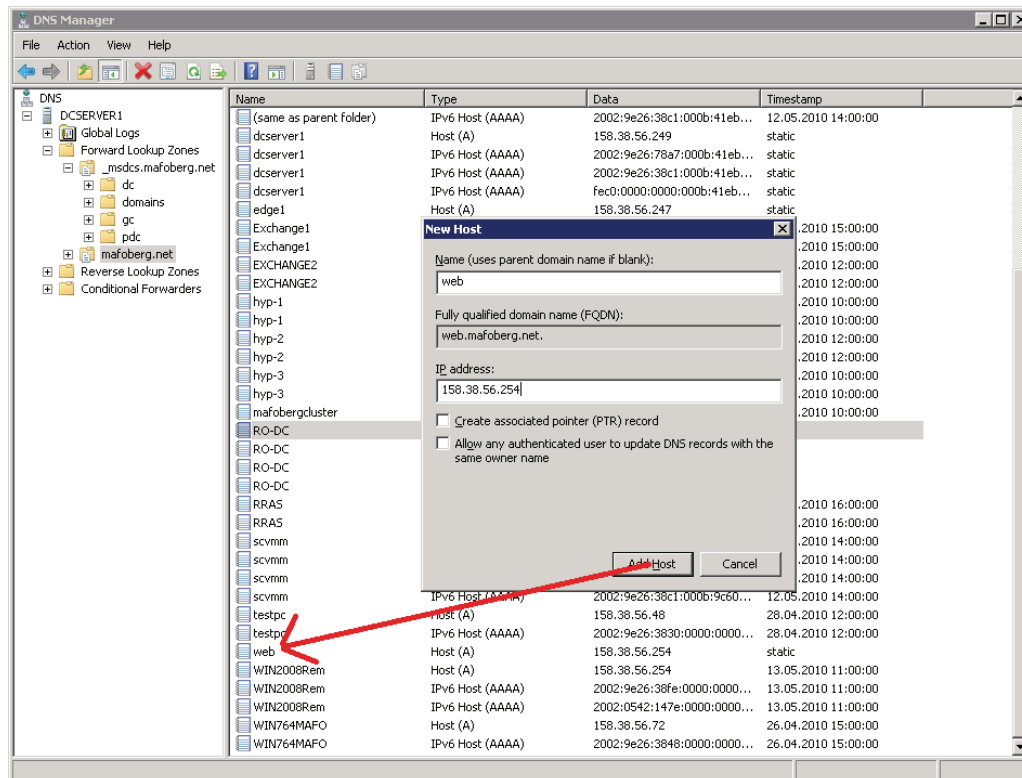
```
get-wmiobject win32_NetworkAdapterConfiguration | where { $_.ipaddress }
```

Kommandoen overfor vil hjelpe oss å identifisere IP-adressen til maskinen den kjører på, slik som vist i Figur 158.



Figur 158 - Finne IP-adresse til Webserver

Nedenfor ser vi et utdrag fra vår DNS Server. Her har vi valgt å bruke IP-adressen som vi fant overfor til å lage en ny host A-record kalt "web" under vårt domene "mafoberg.net".

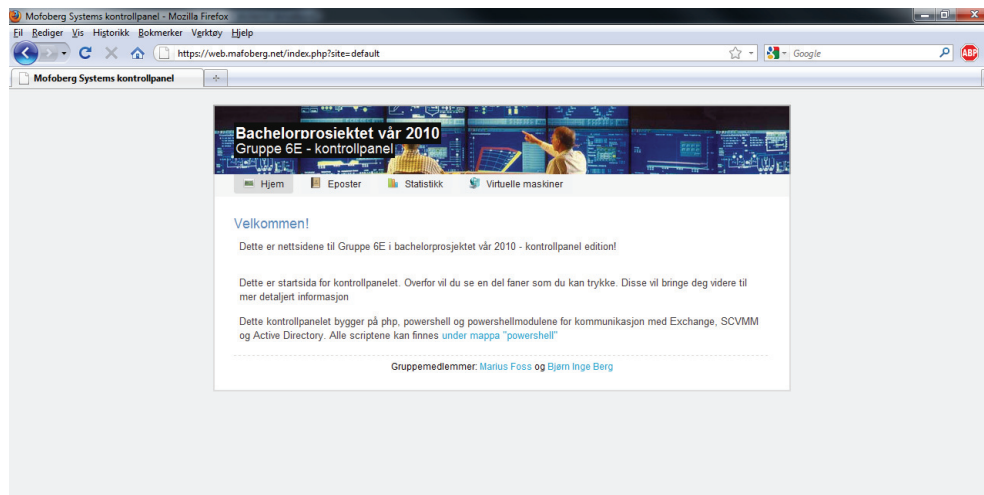


Figur 159 - Legge til en A-record i DNS

Dette resulterer i at vi kan nå webgrensesnittet ved å skrive inn <https://web.mafoberg.net>. Denne adressen vil være offentlig tilgjengelig så lenge dette prosjektet varer.

2.8.4 Lett bruk av grensesnittet.

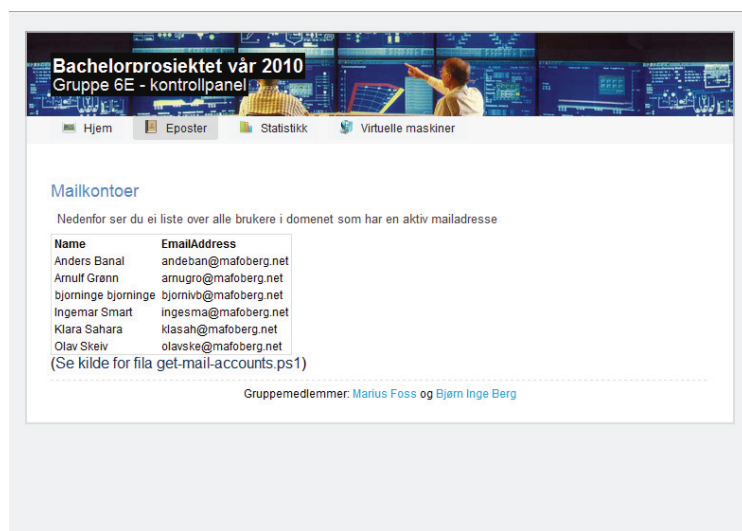
Når vi nå aksesserer <https://web.mafoberg.net>, vil vi møtes med et velkomstbilde som forteller om mulighetene i nettgrensesnittet.



Figur 160 - Åpningsside for webgrensesnitt

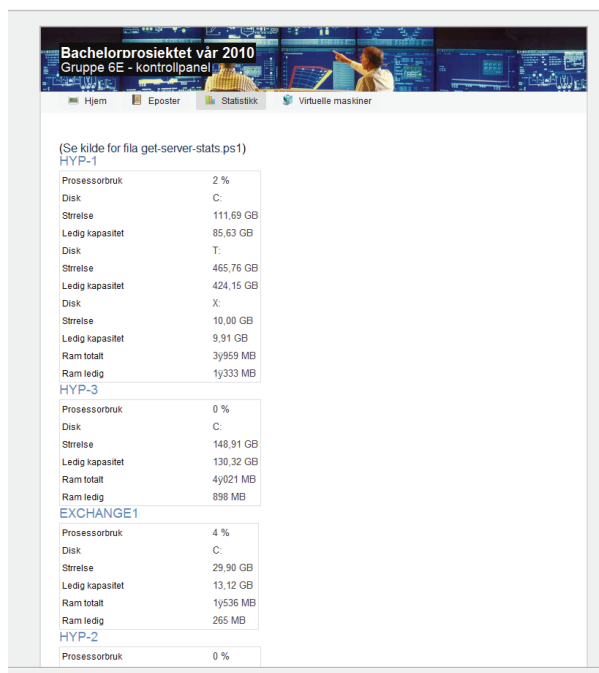
2.8.4.1 Epostutlisting

Nedenfor ser vi fanen ”Eposter”. Denne lar oss vise informasjon om mailkontoer som eksisterer på domenet.



Figur 161 – Epostutlisting

2.8.4.1 Serverstatistikk



Figur 162 - Utlisting av serverstatistikk

Figur 157 Viser en oversikt over prosessorbruk, diskplass og minnebruk på flere av serverne våre. Denne statistikken oppdaterer seg selv ca hvert 40. sekund og benytter seg av Powershell Remoting for å kontakte hver server og hente ut denne statistikken.

3. Drift av systemet

3.1. Exchange 2010 - oppsett og bruk

Exchange 2010 gir oss mulighet til å bli administrert både ved GUIen Exchange Management Console (EMC) og ved hjelp av PowerShell med Exchange integrasjon, Exchange Management Shell (EMS). EMS er bedre utviklet i Exchange 2010.

3.1.1. Recipient Configuration

Opprettelse av epostkontoer (mailboxes) kan gjøres på flere måter. Vi vil ta for oss å gjøre dette via EMS og PowerShell. Exchangebrukere er brukere vi kobler til AD, og eksterne brukere om vi hoster adresser for disse. Disse kan vi finne når vi blar igjennom ved oppretting i GUIen eller om vi lister dem ut med PowerShell.

Vi kan opprette mailbokser for allerede eksisterende brukere i AD, men vi kan også opprette brukere i det vi oppretter mailbokser. Ofte vil mailbokser for eksterne brukere bli opprettet som mailbokser med nye brukere som havner i egne OUer eller medlemmer i grupper som er koblet til andre brukere i skogen vår.

3.1.1.1. Oppretting av postkasse (mailboks) til eksisterende bruker i AD (GUI)

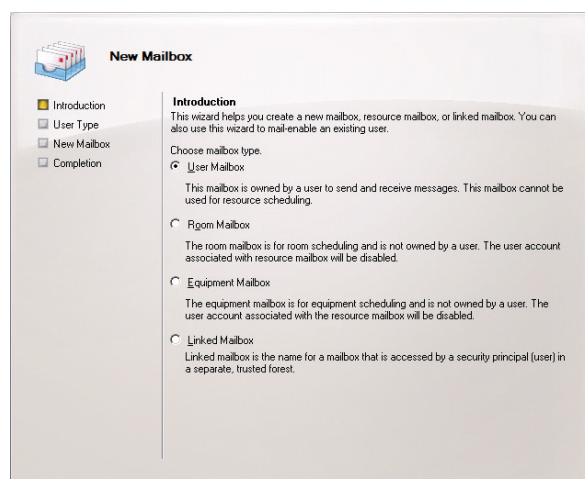
For manuelt å opprette brukere med EMC, går vi til *Recipient Configuration – Mailbox*. Høyreklikk på denne og velg *New Mailbox*.

Der får vi en veiviser for å sette opp mailboks for User, Room, Equipment og Linked Mailbox.

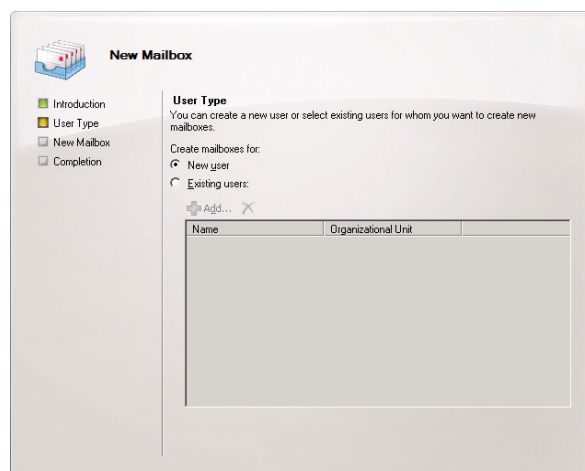
- User Mailbox er mailboks knyttet direkte til en bruker. Dette er den vanligste formen for mailboks.
- Room Mailbox er mailboks knyttet til spesifikke rom, auditorium osv. Dette deaktiverer visse funksjoner for mailboksen.
- Equipment Mailbox er postkasse for utstyr. Dette kan være datamaskiner, telefoner, biler, prosjektorer og liknende. Denne adressen vil bli listet i *Global Address List (GAL)*.
- Linked Mailbox er mailboks som kan brukes av en bruker i en annen skog i domenet. Brukeren er da linket opp mot denne istedenfor å ha mailboksen i sin skog.

Vi velger her *User Mailbox* for å opprette en postkasse for en bruker direkte.

I *User Type* har vi mulighet til å opprette en ny bruker direkte i AD fra mailboxveiviseren. Dette kan være hensiktsmessig hvis bruker med kun mailboksrettigheter er hovedsaken til at mailboksen blir opprettet.



Figur 163 - Ny postkasse



Figur 164 - Ny epostbruker - Usertype

Man kan velge *Existing users* og bla seg fram til allerede eksisterende brukere i AD man kan koble mailboksen til. Se Figur 159 hvor vi kan liste opp eksisterende bruker. Det kan også kobles flere eksisterende til samme mailboks også, men dette har svært skjeldent noe hensikt.

Figur 165 - Brukerinformasjon til AD

Vi velger å opprette med ny bruker. Vi får da informasjonsskjerm bilde opp for brukeren. Her angir vi informasjon slik vi ville gjort det i AD.

Vi kan velge i hvilken OU brukeren skal ligge ved å rysse av for dette og deretter kan vi browse oss frem til OU brukeren skal ligge i.

User logon name (User Principal Name) er brukernavnet som blir koblet til brukeren. Et eget brukernavn kan settes opp for eldre serverversjoner, men vanligvis er det lettest å bruke det samme.

Man kan tvinge bytte av passord ved neste innlogging. Dette lønner seg til nye brukere slik at de får sitt eget passord som ingen andre skal ha tilgang til.

Figur 166 - Mailbox plassering og policyer

gjøre dette kreves det klientlisens (CAL) for Exchange. Vi kan velge en Exchange ActiveSync Mailbox policy. Dette er policy som gjelder for synkronisering av mailbox mellom forskjellige enheter en har mailboxtilgangen på. Disse skal ikke vi gå nærmere inn på. Vi angir alias og går videre for en enkel mailboksopprettelse.

Archive Settings er en mulighet for å linke et arkiv til mailboksen. Da vil alle meldinger bli flyttet til hovedmailboksen som man får bestemt igjennom policyoppsett for denne.

Her har vi oppsett som har med selve mailboksen å gjøre.

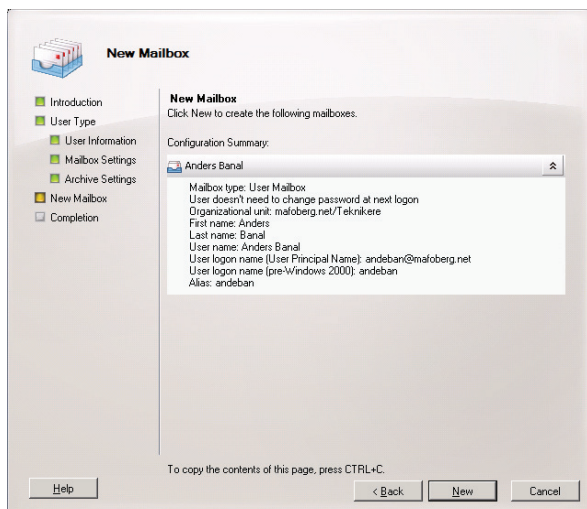
Alias er et alternativt navn til brukernavnet som kan brukes til mailkontoen i stede for brukernavnet. Dette er ikke nødvendig, men vi kan fylle inn det samme som brukernavnet om vi ønsker det.

Vi har tre valg.

Vi kan velge database hvor mailboksen skal ligge i. Vi kan velge policy som skal gjelde for *Managed folder mailbox*. For å

Figur 167 - Mailboks Archive settings

For dette kreves det også en Exchange Enerprice Client Access License (CAL).

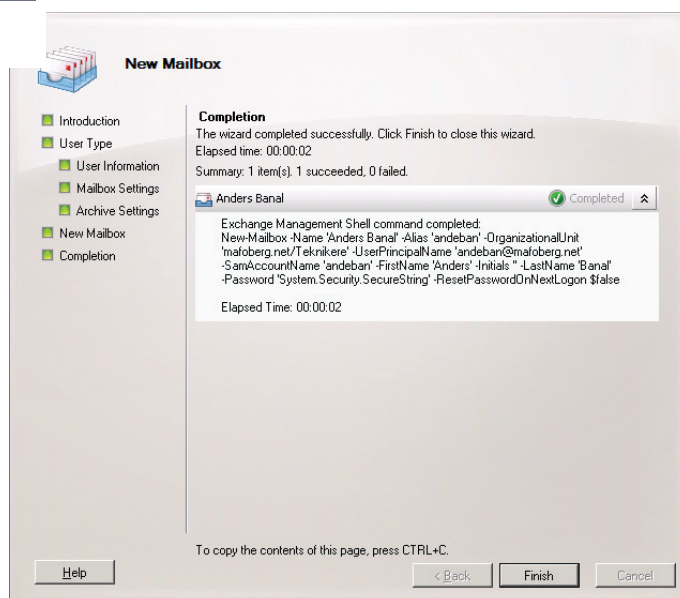


Figur 168 - Mailboks og brukerresyme

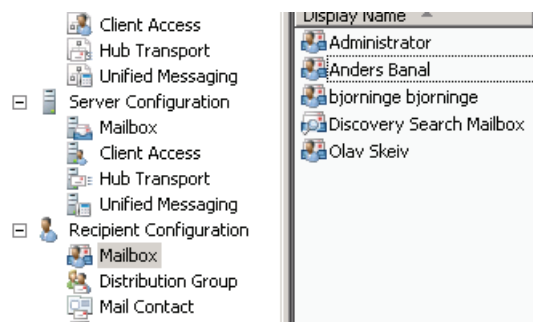
Før ferdiggjøring av mailboksen får vi også vite hvordan PowerShellscriptet vil bli for å opprette en slik mailboks med ny bruker.

Dette kan utnyttes slik at hvis man ikke har oversikt over hva som skal til for å opprette en ny bruker og mailboks ved hjelp av PowerShell, kan man gå alle skrittene frem til siste vindu, som vist i Figur 164, og deretter kopiere dette innholdet (CTRL+C).

Vi vil komme tilbake til administrering av mailbokser med brukere og oppsett ved hjelp av PowerShell i kapittel 4.



Figur 169 - Mailboks for ny bruker, PowerShell



Figur 170 - Mailbokser opprettet

Her ser vi i EMC en liste over mailbokser vi har laget ved manuelt å legge til ny bruker for mailboks, eller laget mailboks og koblet til eksisterende brukere i AD.

Administrator er brukeren som automatisk har fått tildelt en postkasse da Exchange ble installert på serveren. Brukeren som installerer Exchange får gjort dette automatisk fordi kriterier og brukerinformasjon allerede ligger tilgjengelig for Exchange og for at Exchange trenger en

hovedbruker for Exchange som kan få meldinger fra start. Det er mulig å unngå å opprette denne brukeren, men ikke anbefalt ved ny ren installasjon.

3.1.1.2. Opprette en postmasterkonto

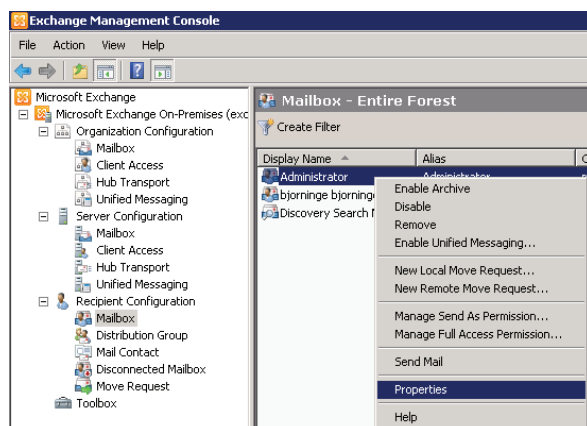
En postmasterkonto er en epostkonto som brukes administrativt i exchangeorganisasjonen. Meldinger om epost som ikke kommer frem, om noe uvanlig skjer med post, feil i eposttrafikk og hendelser som skal meldes ifra om noe bør gjøres med sikkerhet blir sendt til denne epostadressen. I Exchange 2010 med Edge Transportrollen på egen server må denne legges til på begge disse stedene. Kommandoen for å gjøre dette i EMS er i Exchange 2010 forandret.

For å se om det allerede eksisterer en slik postmasterkonto bruker man kommandoen
`Get-TransportConfig`

Da kan man på linjen hvor det står *ExternalPostmasterAddress* se om det er satt noen postmasteradresse. Hvis ikke, kan man lage denne ved hjelp av
`Set-TransportConfig -ExternalPostmasterAddress postmaster@mafoberg.net`

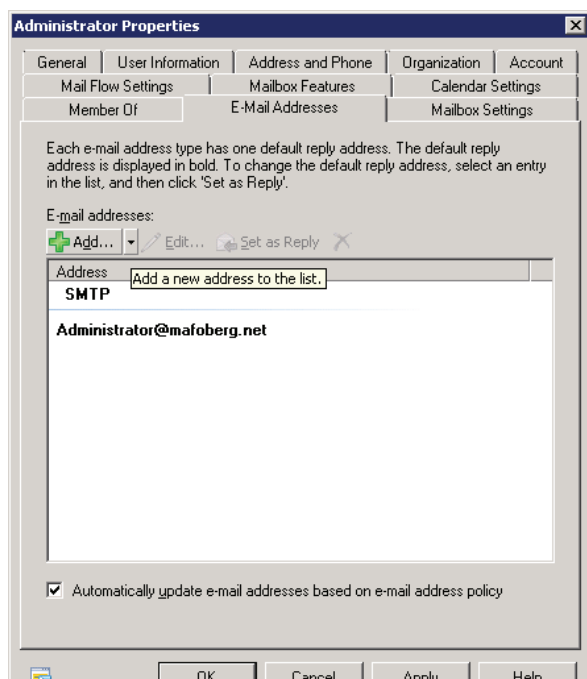
Dette kan man gjøre både på Exchangeserveren og edge-serverene om man ikke synkroniserer dem.

Postmasterkontoen bør enten opprettes til en egen bruker i AD, eller tilegne denne direkte til en administratorbruker. Dette kan man gjøre på følgende måte.



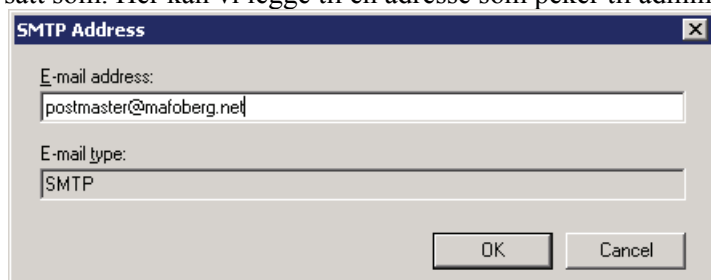
Figur 171 - Exchange postmasterkonto administrator

I *Exchange Management Console* velger man *Mailbox* i *Recipient Configuration*. Der har vi allerede en administratorpostkasse. Vi velger egenskapene på denne.



Figur 172 - Exchange postmaster epostadresser

Her ser vi altså epostadressene som peker til standardadressen (reply-adresse) som svaradressen blir satt som. Her kan vi legge til en adresse som peker til administratorkontoen ved å klikke **Add**.

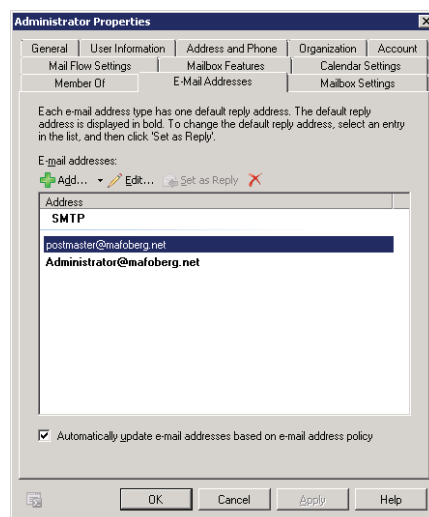


Figur 173 - opprette postmasteradresse

Skribe in adressen til postmasterkontoen og klikke **ok** som vist i Figur 173. Eposttypen vil uansett bli satt til SMTP her. Dermed er denne epostadressen opprettet. Post som vører til ”ikkelevert epost” vil bli varslet her og meldt i fra om hva som blir gjennomført før posten stoppes.

Til høyre ser vi at vi har fått to epostadresser i adresselisten til administrator. administrator@mafoberg.net er satt til å være reply-adressen (uthevet skrift). Det gjør at denne adressen vil bli brukt som reply-adresse når mail sendes ut.

Deretter klikker vi OK og vi har opprettet en postmasterkonto.



Figur 174 - Lagt til epostadresse til brukren Administrator

```

Machine: Exchange1.mafoberg.net
[PS] C:\Windows\system32>Get-TransportConfig

ClearCategories                        : True
DSNConversionMode                     : UseExchangeDSNs
ExternalDelayDsnEnabled                : True
ExternalDsnDefaultLanguage            : 
ExternalDsnLanguageDetectionEnabled   : True
ExternalDsnMaxMessageAttachSize       : 10 MB (10,485,760 bytes)
ExternalDsnReportingAuthority         : 
ExternalDsnSmtpDomain                  : True
ExternalPostmasterAddress              : postmaster@mafoberg.net
ExternalSmtpDomain                     : mafoberg.net
HygieneSuite                          : Standard
InternalDelayDsnEnabled                : True
InternalDsnDefaultLanguage            : 
InternalDsnLanguageDetectionEnabled   : True
InternalDsnMaxMessageAttachSize       : 10 MB (10,485,760 bytes)
InternalDsnReportingAuthority         : 
InternalDsnSmtpDomain                  : True
InternalSmtpServers                    : 
JournalingReportNdrTo                 : 
MaxDumpsterSizePerDatabase            : 18 MB (18,874,368 bytes)
MaxDumpsterTime                       : 7:00:00:00
MaxReceiveSize                        : 10 MB (10,485,760 bytes)
MaxRecipientEnvelopeLimit              : 5000
MaxSendSize                           : 10 MB (10,485,760 bytes)
MigrationEnabled                      : False
OpenDomainRoutingEnabled              : False
Rfc2231EncodingEnabled                : False
ShadowHeartbeatRetryCount              : 3
ShadowHeartbeatTimeoutInterval        : 00:05:00
ShadowMessageAutoDiscardInterval      : 2:00:00:00
ShadowRedundancyEnabled                : True
OrganizationRelationshipForExternalOrganizationEmail : 
SupervisionAge                        : (Reject, Allow)
TLSReceiveDomainSecureList             : 
TLSSendDomainSecureList                : 
VerifySecureSubmitEnabled              : False
VoiceMailJournalingEnabled             : True
HeaderPromotionModeSetting            : NoCreate
Xexch50Enabled                        : True
  
```

Figur 175 - Postmastermailboks verifisert

Vi kan verifisere at dette er gjort ved å gå til Exchange Management Shell. Vi kjører kommandoen ”Get-TransportConfig” og får listet ut transportkonfigurasjonen for Exchange. Her ser vi (som merket i Figur 175) at mailboksen er opprettet. Og vi har knyttet denne til administratorkontoen.

3.1.1.3. Opprette en distribusjonsgruppe

I Exchange 2010 har vi mulighet til å opprette distribusjonsgrupper som før, men vi har nå både vanlige distribusjonsgrupper og dynamiske distribusjonsgrupper.

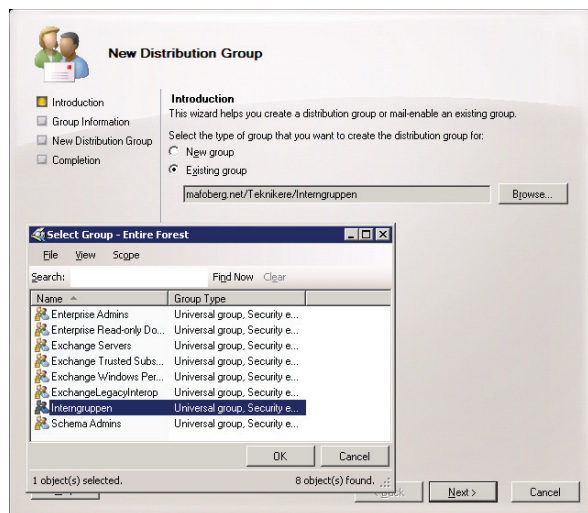


Figur 176 - Distribusjonsgrupper

- **Distribution Group** er distribusjonsgrupper i form av grupper med medlemmer med scope som er satt som Universal. Om de er security-enabled eller ikke, får de tilgang til distribusjonene likevel. Men de uten har ikke tilgang til sikkerhetsoppsett via AD. Ved å sette opp en Distribution Group blir alle medlemmene i en gruppe fra AD man velger tilgjengelig for epostdistribusjon. Brukerne i denne gruppen må ha epostkonto for å bli tilgjengelig for dette. Om noen ikke har fått epostkonto, vil resten av gruppe få tilgang på distribusjonen likevel.
- **Dynamic Distribution Group** er distribusjonsgrupper i Exchange hvor man ikke går ut i fra selve gruppene i AD. Her går gruppene på mapper og OUer. Dette gjør at gruppene blir mer dynamiske. En gruppe i form av OU vil kunne forandres enklere enn å administrere selve gruppene. Siden en OU også kan inneholde brukergrupper, vil disse bli medregnet i distribusjonen. Her gjelder også det at alle med epostkonto vil få tilsendt distribusjonsmail selv om det skulle være brukere uten epostkontoer.

I dynamiske distribusjonsgrupper vil vi få tilgang til å velge policy for hvilke typer grupper mail skal sendes til. Dette kan velges. Brukere med eksterne mailbokser, Exchange Mailbokser, kontakter med ekstern adresse ressurser og grupper med fulle mailmuligheter. Man får videre velge om man vil sette spesielle krav for innhold, hvor de sendes osv.

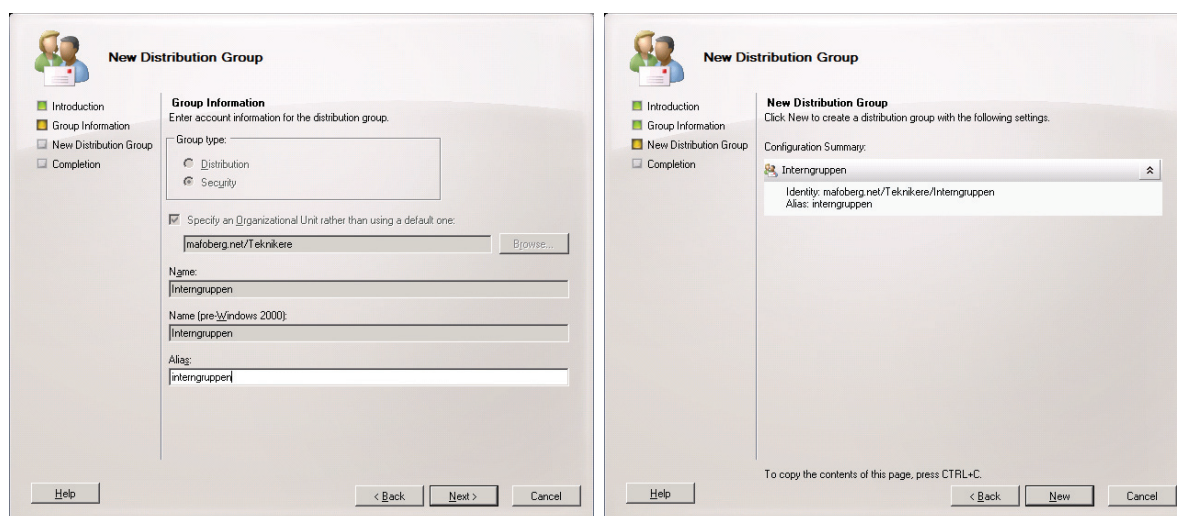
Vi tar utgangspunkt i *New Distribution Group*. Vi har på forhånd også laget en distribusjonsgruppe i AD hvor vi har plassert noen brukere med epostkontoer.



Figur 177 - Bruk eksisterende gruppe

Introduksjonen starter med å velge enten en eksisterende gruppe eller lage en ny gruppe. Hvis vi lager en ny gruppe, vil gruppen bare bli laget uten medlemmer. Disse medlemmene må vi så legge til senere.

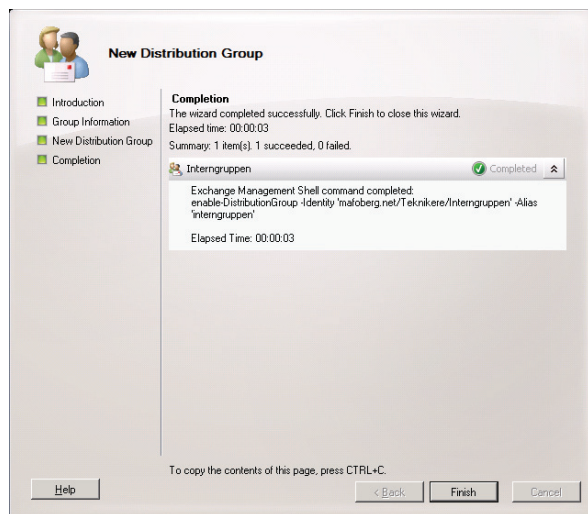
Vi velger en eksisterende gruppe hvor vi allerede har lagt til brukere i. `maloberg.net/Teknikere/Interngruppen`. Denne gruppen består av noen av teknikerne i OUen Teknikere. Vi ser gruppen fra AD er listet i browsevinduet. Her er det listet Universal groups og Security groups.



Figur 178 - Ferdigfylt groupdata

Vi får data fra AD hvor informasjonen om denne gruppen fylles ut. Hvis vi hadde opprettet en ny distribusjonsgruppe, ville vi kunne angi denne informasjonen selv.

Deretter får vi oppsummeringen som ikke er stor for denne gruppen siden gruppeinformasjonen er hentet fra AD fra før.



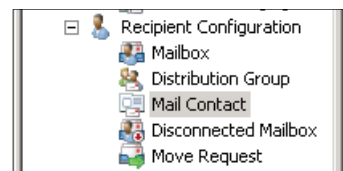
Figur 179 - Ny distribusjonsgruppe

Her ser vi den enkle PowerShellkommandoen som kjøres for å opprette denne distribusjonsgruppen. All post som sendes til intergruppen@mafoberg.net vil nå komme til alle brukerne av denne gruppen.

Hvis vi bruker remote fra en klientmaskin som har administrasjonsrettigheter for Exchange, ville vi kunne høyreklikke på gruppen i denne Remote EMC og deretter velge "send mail" for å sende en mail med en gang fra den lokale mailklienten.

3.1.1.4. Andre Recipient Configurations

Det finnes flere funksjoner for mailboxer i Exchange 2010. Disse finner vi under *Recipient Configuration* hvor vi har *Mail Contacts*, *Disconnected Mailbox* og *Move Request*. Dette gir oss oversikt over mailbokser som skal behandles eller ligger i databasene vi har til Exchange serveren.

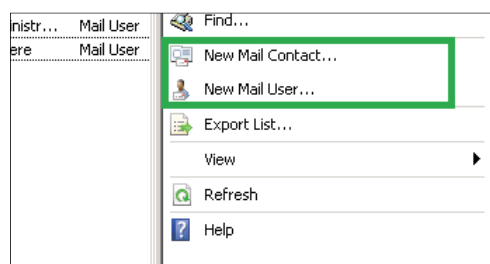


Figur 180 - Recipient Configuration

Mail Contacts

Slike kontakter gir tilgjengelighet for oppsøking av brukere. En base over alle adresser for brukere både eksternt og internt.

New Mail Contact gir oss mulighet til å opprette kontakter for eksterne brukere. Disse kontaktene gjelder for brukere utenfor vår AD. Man bruker gjerne slike mailadresser for eksterne brukere og i andre skoger i AD. Enten brukes da en *Default* bruker eller man kan plassere disse i egne Ouer.



Figur 181 - Kontakt eller brukere

New Mail User er brukere med innloggingsmulighet mot vår AD. Da kan man opprette nye eller koble til eksisterende slik vi gjorde da vi laget mailboxer for brukere. Disse er fortsatt eksterne mailadresser, men vil bli listet i *Global Address List (GAL)* i vår Exchange.

Disconnected Mailbox

Mailbokser for brukere som enten er blitt slettet fra AD, eller blitt frakoblet fra mailboksen vil havne her. Disse ligger fortsatt i databasen for Exchange, i vårt tilfelle standard database opprettet for Exchange. Mailbokser ikke koblet til noen brukere i det heletatt.

Vi har mulighet til å koble til andre Exchangeservere. Da vil vi få listet opp disconnected mailbokser fra disse og vi kan koble disse til brukere i vår AD.

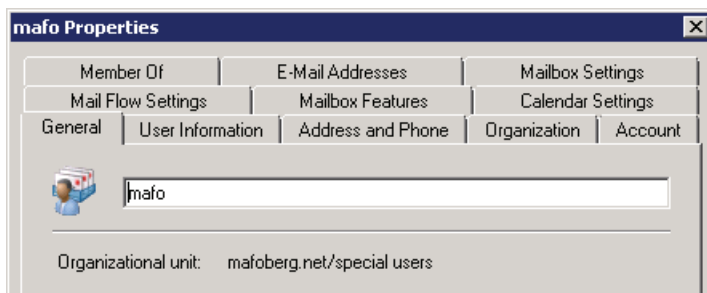
Move Request

Her havner mailbokser som er blitt forespurt for å kunne flyttes til en annen Exchangedatabase. Databasen kan være i vår Exchangeserver eller på Exchangeservere i andre skoger.

3.1.2. Mailbokser.

Vi har mange valg for hver postkasse i Exchange 2010. Vi kan sette størrelser på sendinger og mailbokser. Måter å sende på kan vi forandre på og mye mer.

Under *Recipient Configuration – Mailbox* har vi alle mailbokser. Vi velger ut brukeren *mafo* og velger egenskaper på han.



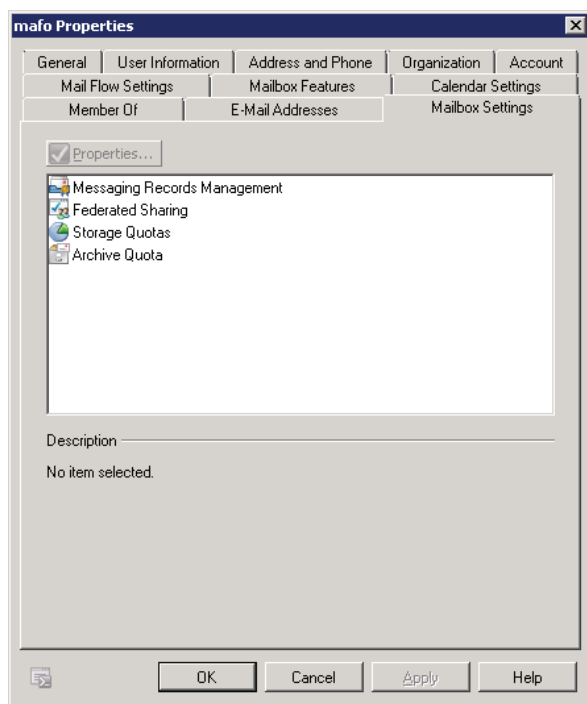
Figur 182 – Mailboksegenskaper

Her ser vi viktige egenskaper på mailboksen.

- **E-mail Addresses** – Adresser koblet til denne mailboksen. Man kan legge til og endre her.
- **Mail Flow Settings** – Vi kan definere størrelse og begrensninger på sendinger av epost. Begrensning på størrelse på mail kan være hensiktsmessig.
- **Mailbox Features** – Oppsett for hvilke protokoller man skal ha tilgang til å bruke. Man har en rekke protokoller man kan enable og disable.
- **Calendar Settings** – Oppsett for tilganger i bruk av kalenderen for brukeren.
- **Mailbox Settings** – oppsett og begrensninger på selve mailboksen.

De øvrige fanene er generelle innstillinger for og informasjon om selve brukeren.

3.1.2.1. Mailboks kvoter



Figur 183 - Mailboks oppsett

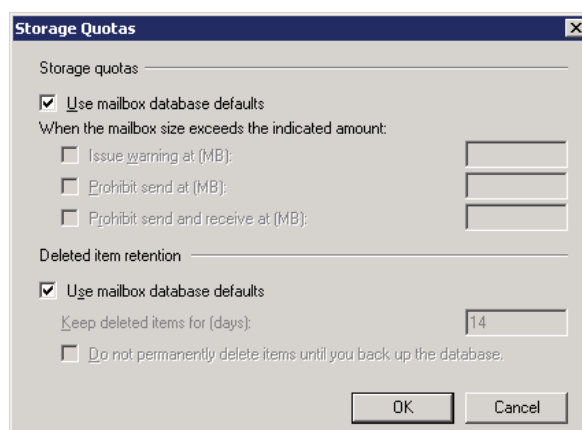
Under innstillinger for selve mailboksoppsettet har vi mulighet for å sette begrensninger på mailboksen, og vi har andre begrensninger.

- **Messaging Records Management** – Vi kan sette opp varighet for innhold i mailboksene. Fra startdato til sluttdato hvis ønskelig, og begrensninger med policies.
- **Federation Sharing** – Deling av kalenderinformasjon og kontaktinformasjon på tvers av exchangeorganisasjoner.
- **Storage Quotas** – størrelse på mailbokser og begrensning på tid for innhold.
- **Archive Quota** – Arkiveringskvote for mailbokser.

Hvis vi ser på *Storage Quotas* har vi mulighet til begrensning på størrelse. Vi kan gå etter standard størrelse på mailboksene. Fra starten er den ubegrenset. Fjerner vi denne standarden kan vi sette på steg for steg av overskridning.

Melding om overskridning, sperre tillatelse etter en viss mengde og motta etter en viss mengde.

Varighet på innhold av mailboksen kan vi bestemme varighet på.

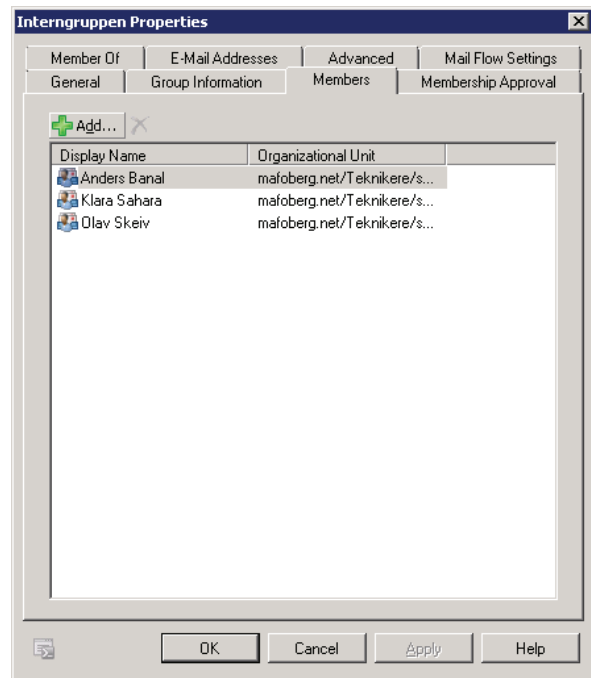


Figur 184 - Storage Quotas

3.1.3. Distribusjonsgrupper oppsett.

Vi har allerede opprettet en distribusjonsgruppe for en gruppe vi kalte Interngruppen. Vi har også her flere oppsettsmuligheter.

- **Members** – Er den samme lista som man får i gruppeobjektet i AD. Den angir hvem av medlemmene som skal få post gjennom denne gruppen.
- **Group Information** – Tillegg for hvem som skal ha tilgang til å administrere gruppen.
- **Membership Approval** – Verifisering av brukere som skal få tilgang til å bli med i eller å forlate gruppen.
- **Mail Flow Settings** – Begrensning på eposttransaksjoner.
- **Advanced** – Oppsett for utvidelsesserver. Man kan velge å skjule gruppen fra Exchange adresseliste. Dette kan gjøre det lettere å unngå å sende mail uforvarende til denne gruppen.



Figur 185 - Distribusjonsgruppeegenskaper

Justeringer kan være greit for å begrense strøm av epost til brukerne hvis man tilhører flere grupper. Dobbelsending av epost kan man unngå hvis gruppene ikke vises i adresselistene.

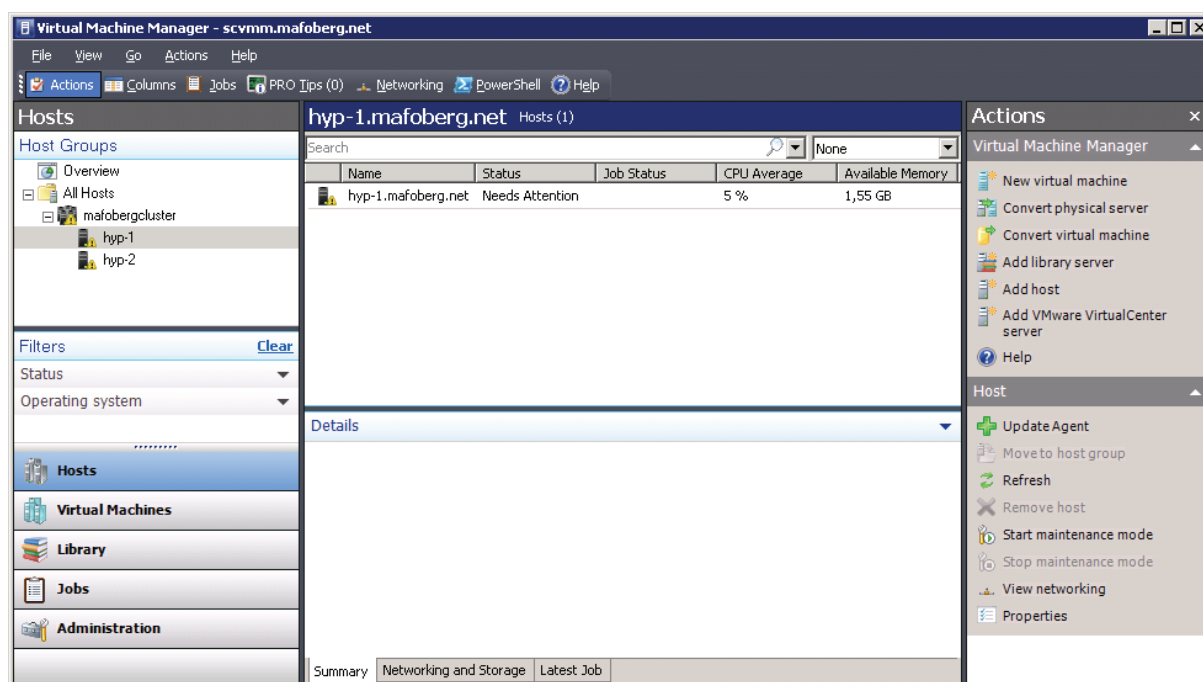
3.2. Bruk av SCVMM

Som sett på tidligere er SCVMM et styringsverktøy for drifting av virtuelle maskiner i Hyper-V. Vi skal i dette kapittelet se nærmere på SCVMM og hvordan bruke forskjellige funksjoner. Blant annet blir det lettere når man kan lage maler (templates) av maskiner så vi slipper å opprette, installere og sette opp alle maskiner helt fra start hver gang vi skal ha en ny maskin. Vi tar for oss flytting av virtuelle maskiner mellom to hoster. Det er også en hel rekke andre funksjoner SCVMM har som vi skal se nærmere på. Vi vil ikke ta for oss alle funksjoner som SCVMM kan brukes til, men de viktigste funksjonene som kommer til å bli brukt.

En SCVMM Server kan brukes på forskjellige måter. De to viktigste måtene er styringsserver for hoster, og libraryserver som lagringsserver for virtuelle maskiner og miljø.

Vi har i dette prosjektet begrensede ressurser for kjøring av alt dette. Vi kjører SCVMM som en virtuell tjeneste for å styre maskiner og håndtere dem. Ikke generelt for lagring. Men vi skal ta med eksempel som viser dette.

Vi skal ta for oss hvordan å lage en ny maskin, å klonen en maskin, å lage en template fra en maskin og konvertere fysisk maskin til en virtuell maskin. Det er også mulig å koble til flere andre servere som kjører virtuelle maskiner.



Figur 186 - SCVMM hovedvindu

Vi har fem hovedgrupper for administrering på venstre side. Hosts, Virtual Machines, Library, Jobs og Administration. Hver av disse hovedgruppene har sine handlings-faner (Action-bars) på høyre side. Disse skifter for hver hovedgruppe vi velger.

- **Hosts.**

Gruppen viser clusterene med hoster (fysiske maskiner) som denne scvmm-serveren administrerer. Vi har begge våre fysiske hoster i denne listen. Det vises status for hvordan de oppfører seg og meldinger for det som er nødvendig. Vi ser også at hyp-1 her trenger oppmerksomhet. Dette gjelder oppdateringer som vi fortsatt ikke har kjørt.

- **Virtual Machines.**

Denne viser lister over virtuelle maskiner som behandles av denne scvmm-serveren. Vi kan da se status av CPU-bruk, host den kjører på, status om den er opp eller ned osv.

- **Library.**
Et bibliotek for lagring av konfigurasjon for virtuelle maskiner som kan brukes til å opprette nye maskiner fra maler, bruke tidligere konfigurasjoner fra maskiner med mer.
- **Jobs.**
Jobber på virtuelle maskiner, status og hendelser listes her. Blant annet oppstart av maskiner, flyttinger og andre hendelser. Disse kan restarteres om noe skulle gå galt.
- **Administration.**
Administrasjon av statistikk for ressurser som brukes, roller for brukere som skal ha tilgang til å styre deler av maskinene og andre styringsenheter som får tilgang til styringsserverene.

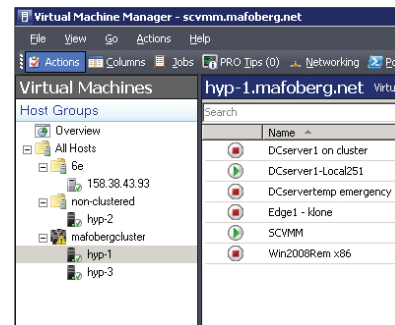
I menyen *Virtual Machine Manager* har vi ulike måter å behandle maskiner på. Vi kan konvertere maskiner fra andre maskiner, fra fysiske maskiner, legge til maskiner i libraries, koble SCVMM sammen med andre managements-servere og likennde.

- **New Virtual Machine.**
For å opprette nye maskiner i clusteret. SCVMM vil da i tillegg legge til egne ressurser for styring av serverne på clusteret.
- **Convert Physical Server.**
For å konvertere en fysisk maskin til virtuell maskin. Denne tjenesten kopierer en hel server og gjør den tilgjengelig som virtuell maskin. Alle drivere og tilhørende blir tilpasset.
- **Convert virtual machine.**
Konverterer andre virtuelle maskiner til Hyper-V maskin. Dette gjør at man kan innhente for eksempel VMware-maskiner og andre typer virtuelle maskiner som Hyper-V er kompatibel med.
- **Add library server.**
Legger til en bibliotekserver til styringskonsollet. Der kan man lagre kloner, templates og liknende.
- **Add host.**
Legger til flere hoster til cluster i SCVMM. Her er det mulig å legge til servere som kjører Hyper-V virtualisering og VMware-maskiner som kjører sine virtuelle maskiner. For at man skal kunne legge til VMware-maskiner, må man først ha lagt til en VirtualCenter server.
- **Add VMware VirtualCenter server.**
Man kan legge til VirtualCenter servere til SCVMM. Da kan man også styre maskiner fra VMware-grensesnitt. Etter å ha lagt til en VirtualCenter-maskin, må man legge til hostene fra denne.

3.2.1. Lage en ny maskin

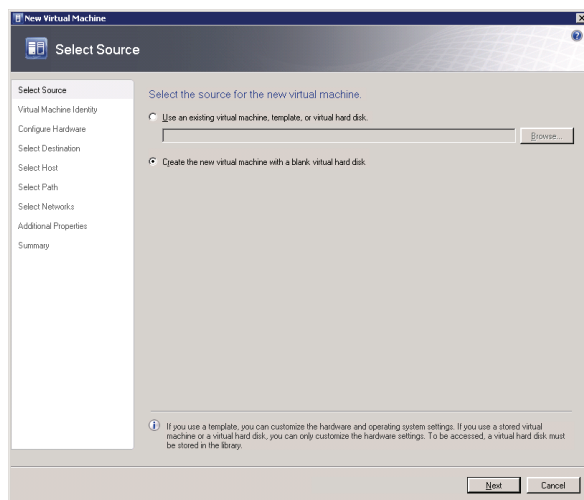
I hovedfanen *Virtual Machines* får vi oversikt over virtuelle maskiner vi har. *Host Group* er trestrukturen for hostene vi er koblet til med SCVMM-serveren. Vi gjør smart i å sette hostene i forskjellige grupper. Hosts som står alene utenfor cluster plasserer vi i egen gren. Clustrede hosts vil bli plassert under et cluster-gren.

Som vist i Figur 182 har vi tre grener. Maskinene som er vist er maskiner styrt av cluster-host Hyp-1.

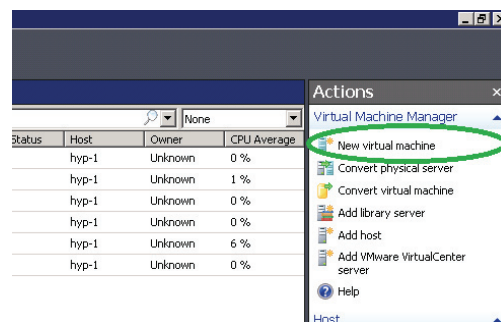


Figur 187 – SCVMM Host Groups

For å lage en ny maskin velger vi *New virtual machine* i *Virtual Machine Manager* –fanen. Vi får da opp en veiviser for å lage en ny maskin.



Figur 188 - SCVMM Lage helt ny maskin



Figur 189 – SCVMM New Virtual Machine

Her får vi velge mellom å bruke en eksisterende virtuell maskin, template eller virtuell disk.

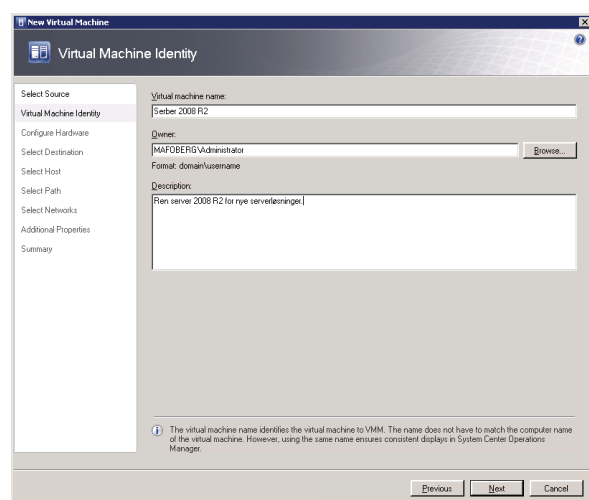
Vi kan også velge å opprette en helt ny maskin med nye maskin-ID.

Vi skal ta utgangspunkt her i å lage en helt ny maskin.

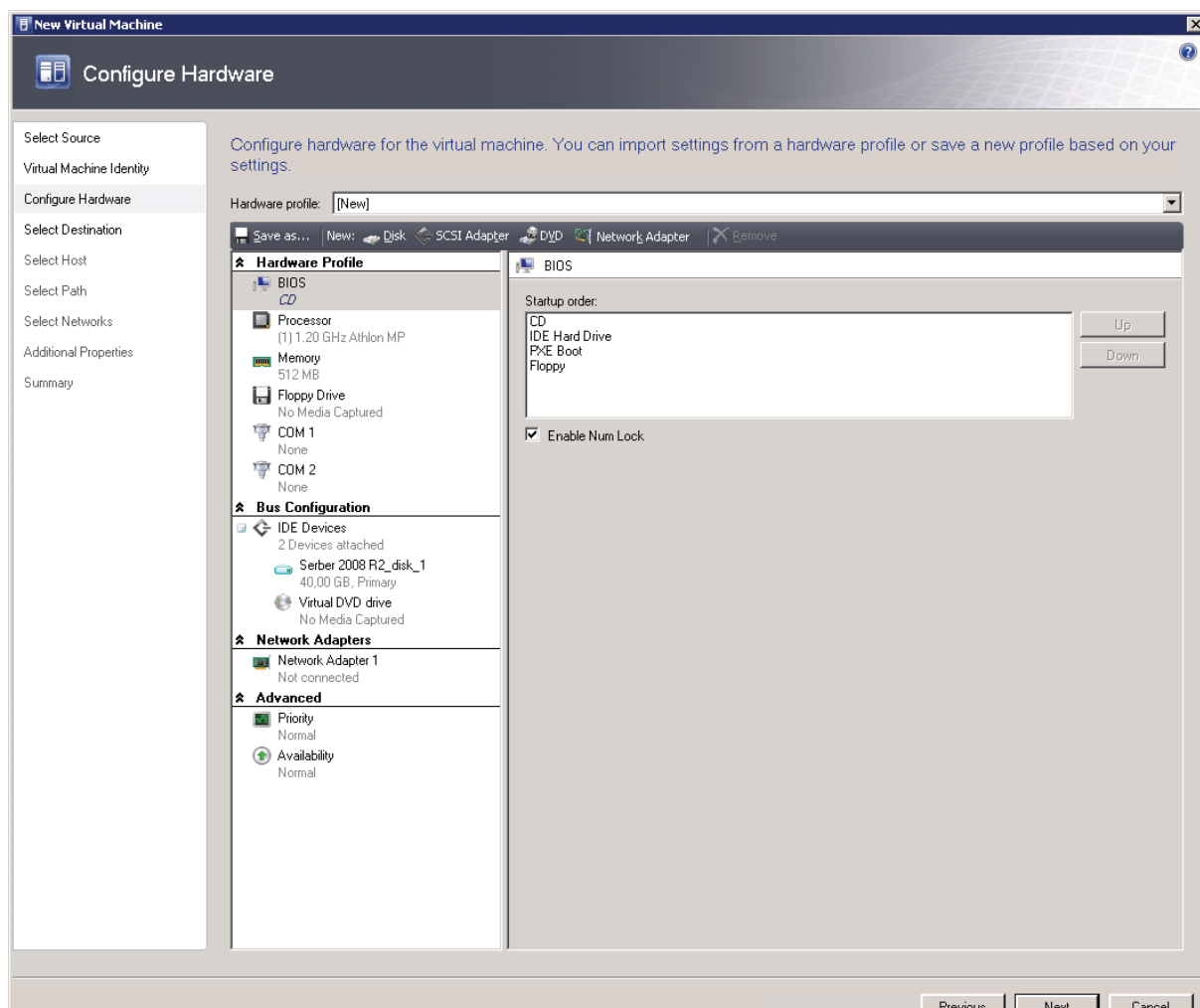
Vi angir et passende navn ut ifra det vi skal bruke den til.

Eier for maskinen setter man til den som skal ha hovedansvaret for maskinen. Hvis forskjellige brukere av SCVMM logger inn med konsollet, vil rettigheter kunne administreres for disse.

I description skriver vi inn informasjon som er relevant til den nye maskinen.



Figur 190 - SCVMM Ny maskin informasjon



Figur 191 - SCVMM Ny maskin - Hardware configuration

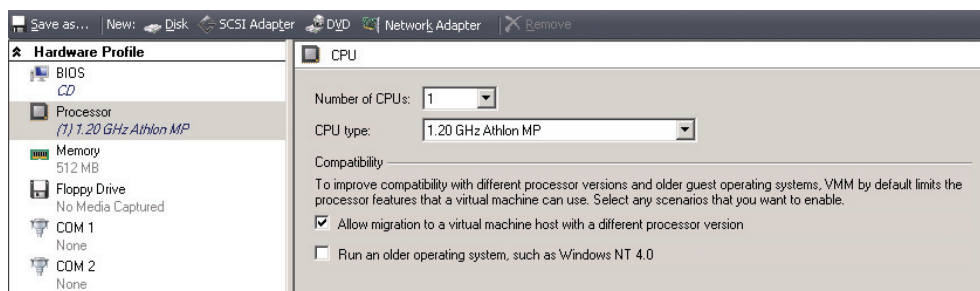
I rullegardinen øverst (*Hardware profile*) har vi en proffilliste over maskinoppsett vi kan lagre. Dette kan bli svært nyttig hvis vi skal ha maskiner til bestemte formål med maskinvareprofil som ikke differerer for mye. Vi skal opprette en Server 2008 R2 profil som kan brukes senere.

Vi skal ta for oss hver del av oppsett for maskinvareprofiler.

3.2.1.1. Hardwareprofil – BIOS (Basic Input Output System)

Som vi ser av Figur 191 har BIOS mulighet til å sette opp boot-rekkefølgen for enhetene. Standard er CD-medium først. Num Lock status er vanligvis ikke satt på.

3.2.1.2. Hardwareprofil – Processor



Figur 192 - HWprofile – Prosessor

Vi har mulighet til å velge antall cpuer som skal være tilegnet den virtuelle maskinen. Skal det være tunge maskiner som kjører Exchange med Forefront Protection, lønner det seg med mer enn 1 cpu. Eller blir arbeidsprosessen uholdbar.

CPU-typen kan settes til forskjellig emulert hastighet. En rekke emulerte CPUer fra AMD og Intel. For utviklingsmiljø kan det være krav til hva CPU som skal være på bestemte maskiner. Også for en del Programvare er det krav til hastighet for CPU, og noen vil ikke valideres ok om ikke kravet er innstilt.

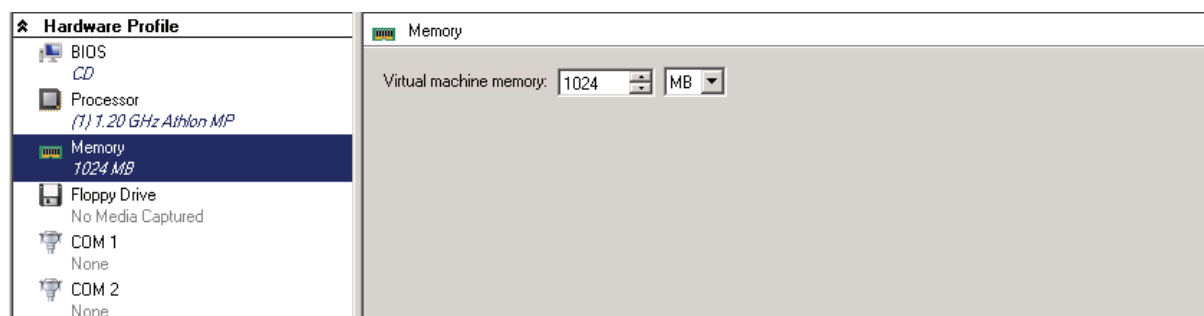
- *Allow migration...*

Dette er en opsjon som er veldig aktuell for Highly Available Clusters. To fysiske maskiner som har akkurat lik prosessor har mulighet for migrering likevel. Men hvis vi slår denne på, kan to fysiske maskiner har forskjellig prosessorarkitektur. Da kan man migrere en virtuell maskin fra en host til en annen uten å tenke på å forandre maskinvareprofil.¹³

- *Run an older operation system...*

For å være kompatibel med eldre operativsystem, bør denne boksen være på. Eldre prosessorer har en annen arbeidsmåte enn nyere prosessorer, og gamle OS kan derfor komme i kompatibilitetsproblem. Kjører man Windows 95 eller Windows NT 4 bør man krysse av for denne.

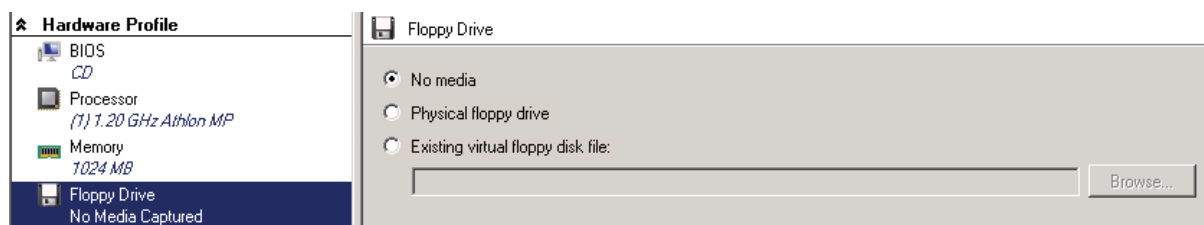
3.2.1.3. Hardwareprofil – Minne



Figur 193 - HWprofile – Minne

For tunge servere er det viktig med nok minne. For eksempel Exchange 2010. Dette kan forandres spesifikt for slike maskiner senere. Minne er en enhet som ofte forandres fra maskin til maskin alt etter typen bruk.

3.2.1.4. Hardwareprofil – Floppydrive

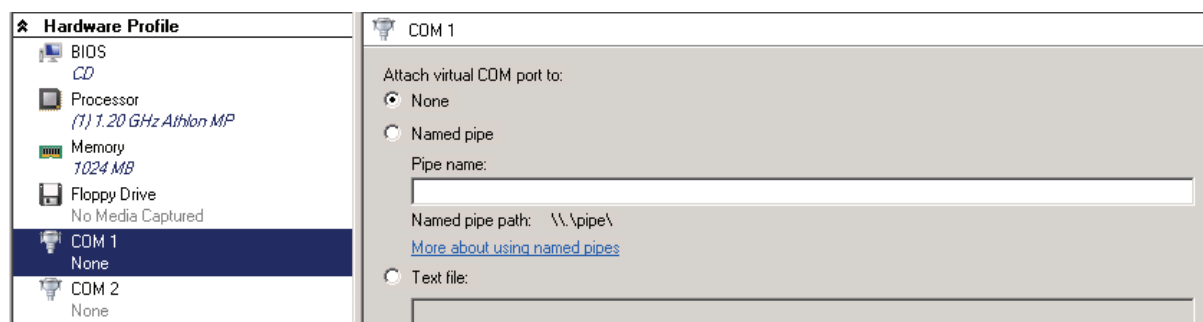


Figur 194 - HWprofile – Floppy

Floppy kan være lønnsomt i testmiljøer for å teste oppstartsmedium. Enten for å teste floppyer på en fysisk stasjon for å se på gamle bootmedia. Ofte er det mest vanlig å kjøre både floppyer og cder som monterte ISOer eller bilder.

¹³ Dette gjelder ikke for *Live Migration*. Da må prosessorene på de fysiske maskinene komme fra samme produsent av prosessorer. For eksempel AMD eller Intel.

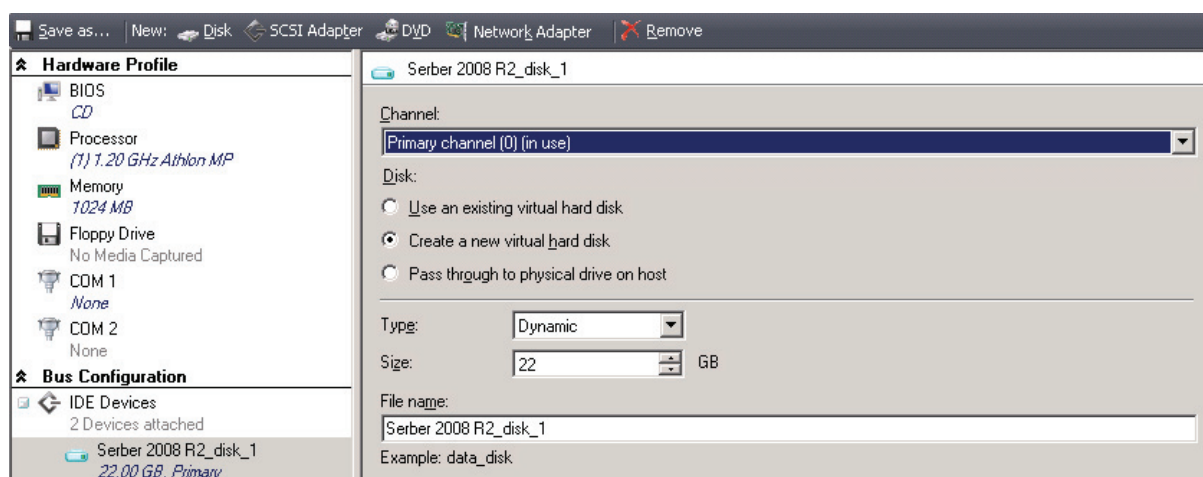
3.2.1.5. Hardwareprofil – COM



Figur 195 - HWprofile – COM

Comporter er vanligvis ikke aktuell i servertilfeller. Disse kan kobles i pipes til andre enheter eller fysisk comport. Dette kan være aktuelt som på fysiske maskiner mot for eksempel gamle UPSer. Man kan bruke en text file for fast kommunikasjon.

3.2.1.6. Hardwareprofil – IDE Disk



Figur 196 - HWprofile – IDE Disk

Vi kan velge ikke å ha noe disk, en disk eller flere. En IDE Device er standard kontroller som følger med nye virtuelle maskiner. Den er som standard med 4 kanaler, kanal 1 og 2 med master og slave.

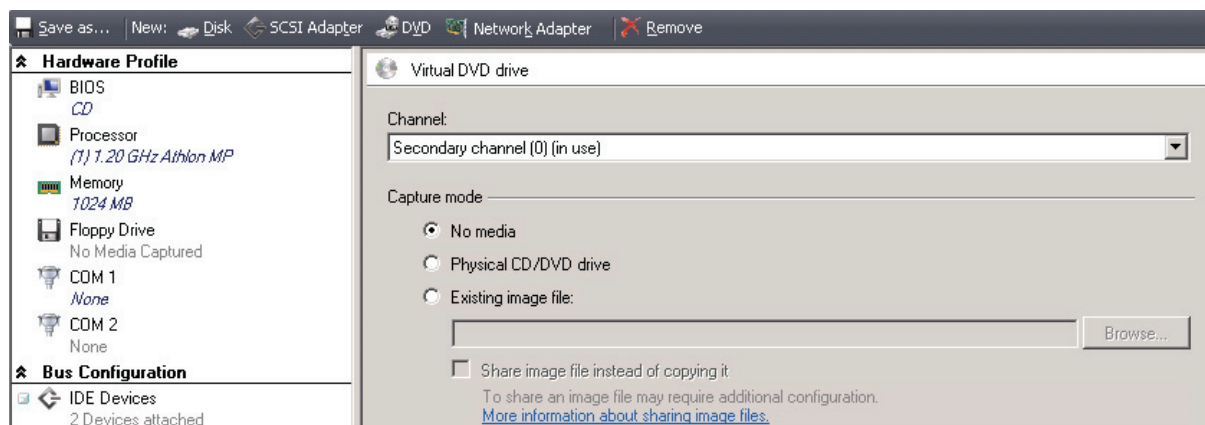
Andre typer virtuelle maskiner krever SCSI-enheter, for eksempel VMware-maskiner på andre hoster. Vi må da klikke "New: ... SCSI Adapter" for å legge til en slik kontroller. Når vi da legger til nye disker, vil de automatisk bli lagt til under IDE-kontrolleren. Vi velger da den disken og får mulighet til å velge Channel satt til en SCSI-enhet.

- Velge eksisterende harddisk – hvi vi har en eksisterende *.vhd fil liggende med os eller data allerede på, har vi mulighet til å koble til denne. Denne bør da kopieres til en valgt plass for disken, for den vil bli brukt direkte. Dette for å sikre at forandringene ikke skjer på den disken man har liggende, men lage en kopi av denne.
- Lage ny diskfil – vi velger kanal hvor den skal ligge som på en vanlig IDE-kontroller.
- Pass through to pphysical host – gir mulighet til å bruke en fysisk disk direkte på hosten. Dette er sjelden bruk. Hosten selv vil ikke ha direkte tilgang til denne disken når den er montert.
- Type disk setter oppførselen til fila. *Static* gjør at diskfila får sin størrelse etter den logiske disken med en gang. *Dynamic* gir fila mulighet til å forandre seg etter mengden innhold.

Dette er den mest fleksible muligheten. Disker blir ikke så store å kopiere. Kloner og templates blir heller ikke så vanskelig å bruke.

- Størrelse setter vi til 22GB. Det gir plass til Operativsystem og eventuell programvare den skal brukes til. Det anbefales å opprette egen disk for databaser om man ikke bruker en disk et annet sted eller en database over nettverk.
- Navn på disk – Det gis forslag, men mulighet for å forandre etter eget ønske.

3.2.1.7. Hardwareprofil – CD/DVD rom



Figur 197 - HWprofile – IDE CDdrive

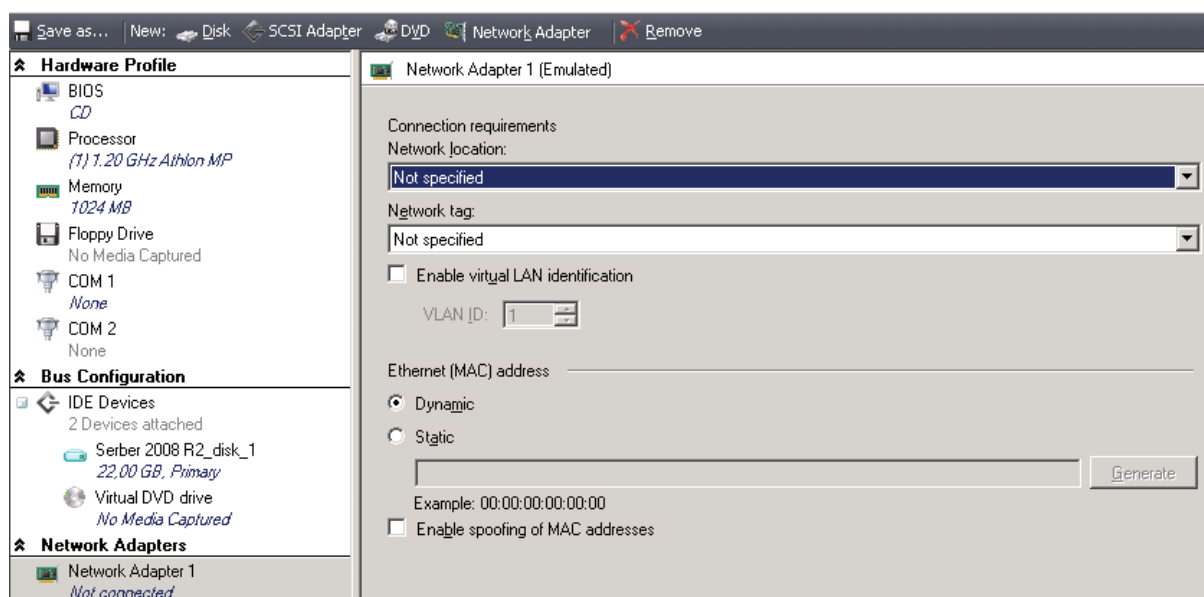
Her velger vi om vi vil ha et ISO-image tilkoblet, fysisk stasjon på hosten eller ingen innhold. Hvis man monterer en ISO, bør denne være tilgjengelig på samme sted fra begge hoster.

En felles nettverkssted for ISO-filer for CDer og DVDer lønner seg. Dette gir mulighet til å unngå dobbeltlagring og feil i montering av ISOer.

Hvis en maskin blir satt til *Live Failover Clustering*, og path til isofil ikke stemmer, vil SCVMM gi kommentar om dette og gi mulighet til direkte reparering.

Share image file lar filen ligge på originalt sted uten å kopiere det til et eget buffer. Dette kan være plassbesparende i enkelte tilfeller, men må ha tilgang til som må konfigureres for dette.

3.2.1.8. Hardwareprofil – CD/DVD rom



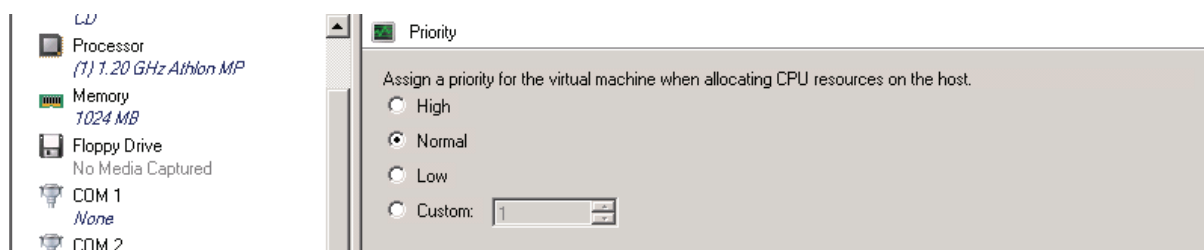
Figur 198 - HWprofil – Network Adapter

Network location og *Network Tag* gir navn for hva slags nettverkskort de virtuelle kortene er koblet til. Ofte er det satt opp en virtuell switch eller annen enhet som kan kobles til.

LAN Identification er en identifikasjon for adskilte interne linjer kortet skal kobles på. Dette fungerer som VLAN i virkeligheten.

Ethernet (MAC) Address gir mulighet til å gi de virtuelle kortene en dynamisk adresse hvor vmm selv angir adressen for kortet. Det er mulig å sette adressen selv om man ønsker dette. Det er da mulig å sette adressen lik en annen hvis man krysser av for *spoofing* nedenfor.

3.2.1.9. Hardwareprofil – Priority



Figur 199 - HWprofil – Kjøreegenskaper

Prioritering av CPU-ressurser på hosten er standard satt til Normal. Hvis maskinen skulle være svært viktig at får CPU-ressurser, kan vi øke nivået på denne. Det kan være at maskinen kjører kritiske tjenester på nettverket og derfor burde bli førprioritert.

Man kan også sette en egen skala fra 1 til 1 000 000 hvor man selv kan planlegge nivåer.

3.2.1.10. Hardwareprofil – Tilgjengelighet (Availability)

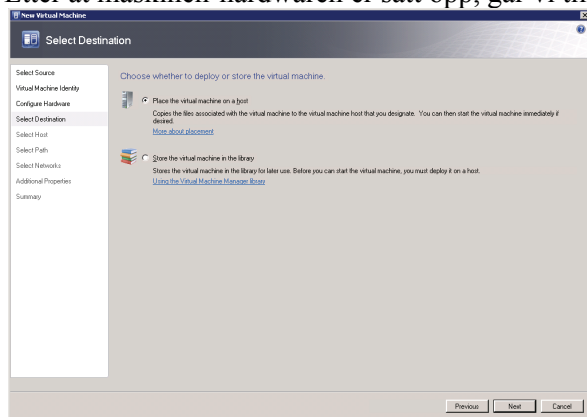


Figur 200 – HWprofile – Highly availability

Denne opsjonen gjør maskinen tilgjengelig for failovering. Maskinen blir da plassert på et cluster hvor hoster kan ta over for hverandre om andre hoster feiler, enten ved strømbrudd, krasj og andre mulige scenarier.

Vi setter den ikke på i dette tilfellet for å kunne bruke den utenfor clusteret, lage templates og spre denne maskinen.

Etter at maskinen-hardwaren er satt opp, går vi til neste skritt i veiviseren for å lage maskinen.



Figur 201 - SCVMM maskinplassering

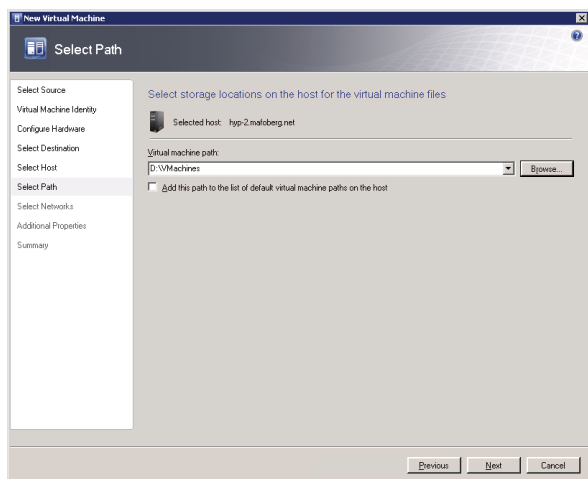
Vi må nå plassere maskindataene og disken enten på en host, eller i en Library.

Library – er en lagerplass for virtuelle maskiner og ressurser hvor man kan hente ut dette når man trenger det. Dette er en SCVMM server med lagringsplass som ikke bør kjøres som en virtuell maskin. Eventuelt blir lagringsplassen på et lagringsnettverk (NAS/SAN).

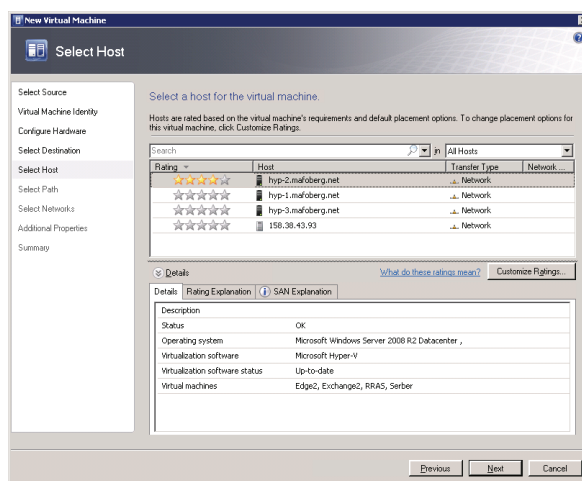
Plasserer vi maskinen på en host, kan vi kjøre den. Det kan ikke gjøres dersom maskinen ligger i en library.

I Figur 197 har vi liste over hoster tilgjengelige for lagring av maskinen. Ratingen angir hvor mye ressurser som brukes av hosten av kjørende virtuelle maskiner.

Her har vi Hyp-2 som ikke er koblet til clusteret.



Figur 203 - SCVMM Plassering på host



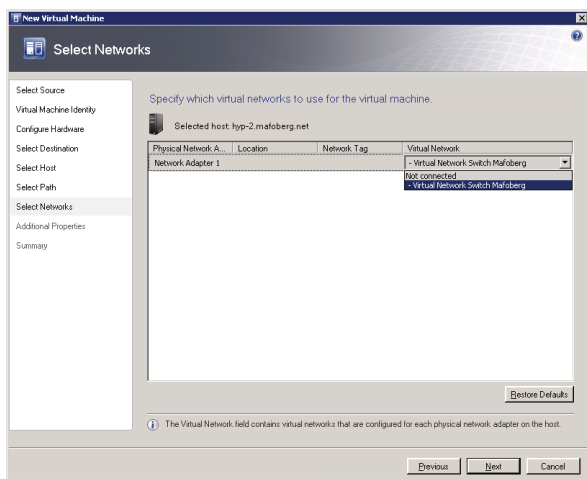
Figur 202 - SCVMM hostvalg

På clusteret kjøres maskiner i høytilgjengelighet.

Vi velger her Hyp-2 hvor vi skal bruke denne maskinen senere.

Man får også god oversikt over maskiner som kjører på disse hostene så man kan vurdere hvilke maskiner man kan legge den nye maskinen sammen med.

Velg sti på den aktuelle hosten hvor maskinen skal ligge.

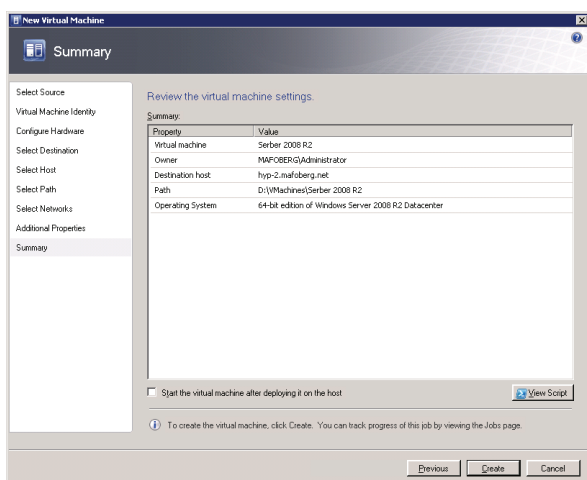


Figur 204 - SCVMM Nettverksadapter

Vi velger oppførsel til maskinen når hosten starter opp. For eksempel om den skal starte samtidig, eller om den skal holde seg av. Sett gjerne en ventetid på maskinen så hosten får tid til å starte opp tjenester og klargjøre seg før de virtuelle maskinene trer i kraft.

Måter å avslutte på er også viktig. I enkelte tilfeller kan det være kritisk å lagre maskinforholdet på hosten og bare avslutte. I dette tilfellet hvor vi skal bruke maskinen til malbruk, gjør ikke dette noe.

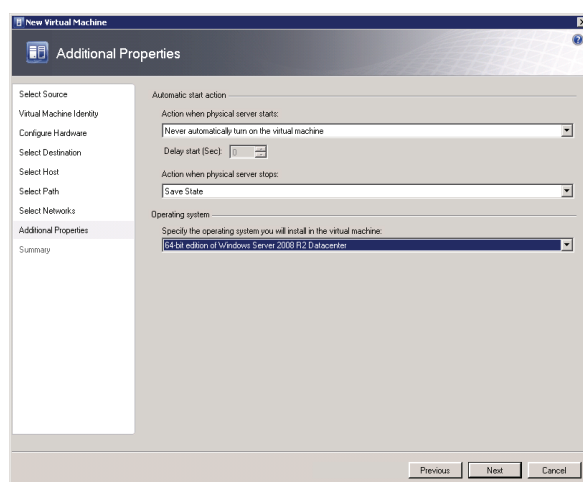
Vi setter operativsystemtype for at maskinen skal være forberedt til det gjeldende OS.



Figur 206 - SCVMM Ny maskin oppsummering

Vi velger så tilkobling for hvilket nettverk nettverkskortet skal kobles til.

Vi har en adapter som vi kobler til vår standard virtuelle switch vi har opprettet.



Figur 205 - SCVMM ekstra egenskaper

Deretter en oppsummering med PS-script generering hvis man ønsker å gå ut ifra dette.

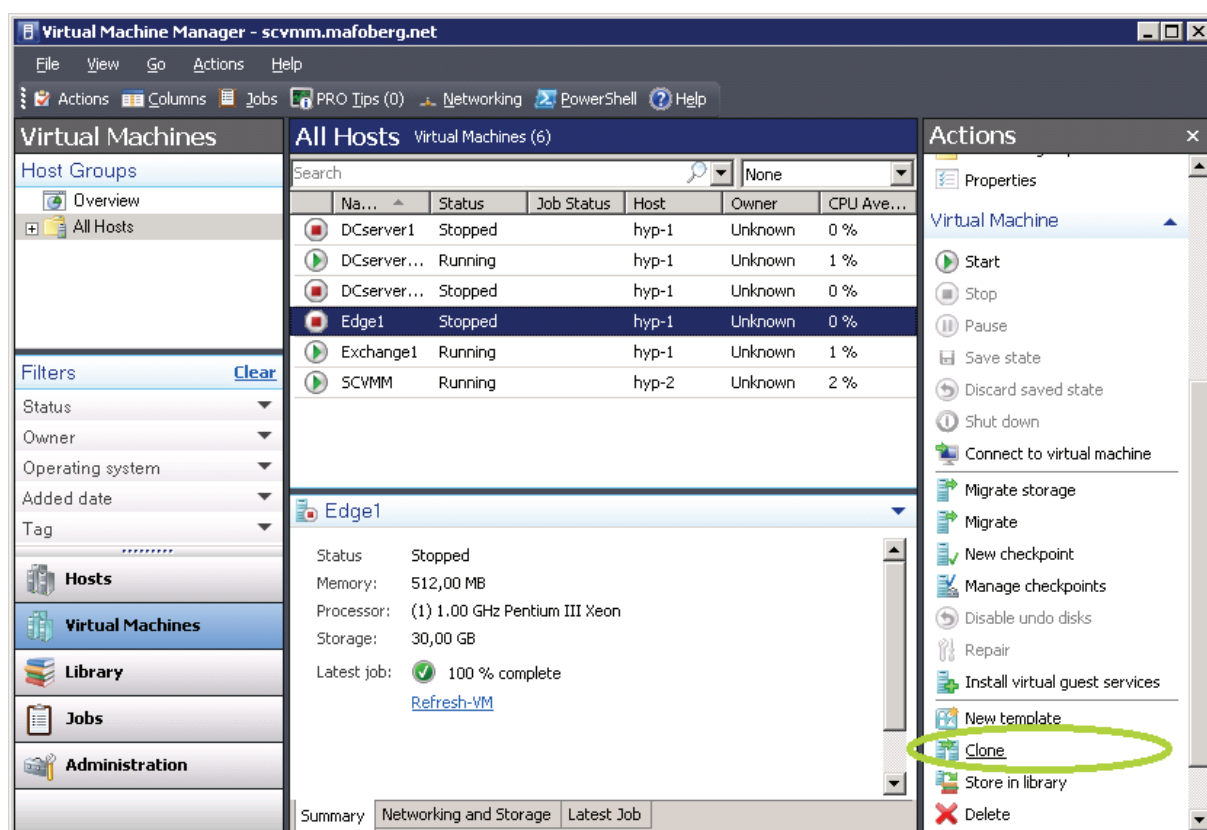
3.2.2. Å klonе en virtuell maskin

En klonе er en nøyaktig kopi av en annen maskin. Det endres ikke informasjon for maskinen. Dette kan være kjekt å ha om man vil ha et nøyaktig utgangspunkt fra en maskin med all konfigurasjon og maskinvareIDer som den originale. Ved hjelp av SCVMM gjøres dette på en enkel måte.

Maskinen som skal brukes som utgangspunkt for å lage en ny virtuell maskin, bør være nyinstallert og rent oppsatt med de nødvendige applikasjonene som skal følge med for de maskinene som skal opprettes fra disse. Maskin som skal brukes til arbeidsmaskin har applikasjoner som office, mailklient med mer. En maskin som brukes til server for databaser med SQL, vil kanskje ha MS-SQL installert og klar til å settes opp.

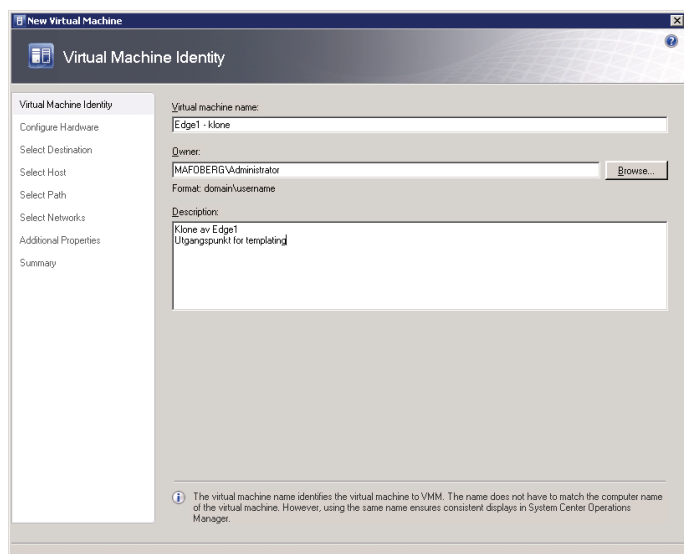
Vi går her ut ifra å ha en helt ren Windows 2008 R2 server. Vi har gitt maskinen navnet Edge1 fordi vi senere skal fortsette å bruke akkurat denne til å sette opp en maskin etter at klonen er opprettet.

Før maskinen opprettes blir vi anbefalt å lage en klonе av maskinen siden templatene blir strippet for brukerdata og konfigurasjoner.



Figur 207 - Clone maskin for template

Vi administrerer de virtuelle maskinene under *Virtual Machines*. Der merker vi den virtuelle maskinen vi skal gå ut ifra og velger *Clone* i menyen *Action*. Vi får da en veiviser for hvordan vi skal sette den opp og hvor å plassere den.

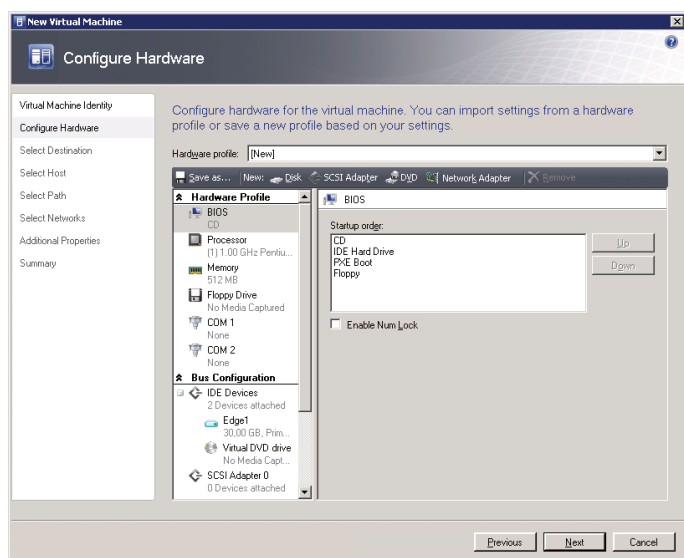


Figur 208 - Eier av virtuell maskin

Ved start av kloningen får vi beskjed om å gi klonen navn. Vi gir den navn etter hva den inneholder.

Eieren er navnet på brukeren som skal ha ansvaret for den virtuelle maskinen. Siden vi i dette domenet velger at administrator skal ta seg av maskinene, gir vi retten til denne. Det kan være at en bestemt administrator skal ha tilgang til visse kloner av maskiner. Det kan da være lurt å gi ansvaret til denne.

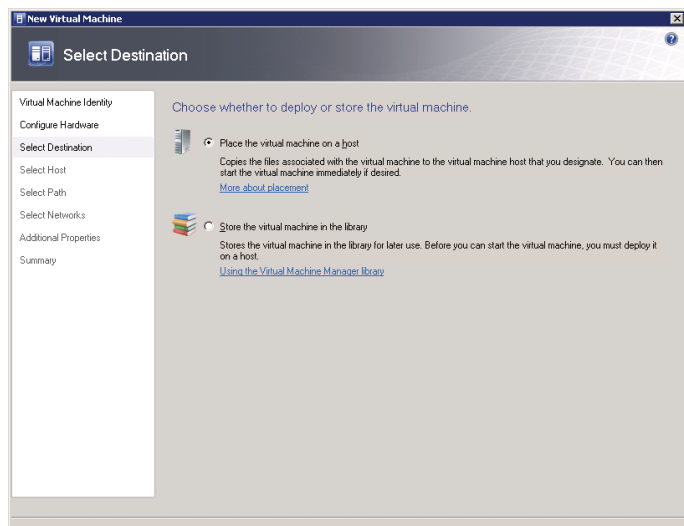
I description velger vi å beskrive maskinen for hva den er til og hva den inneholder.



Figur 209 - Hardware konfigurasjon

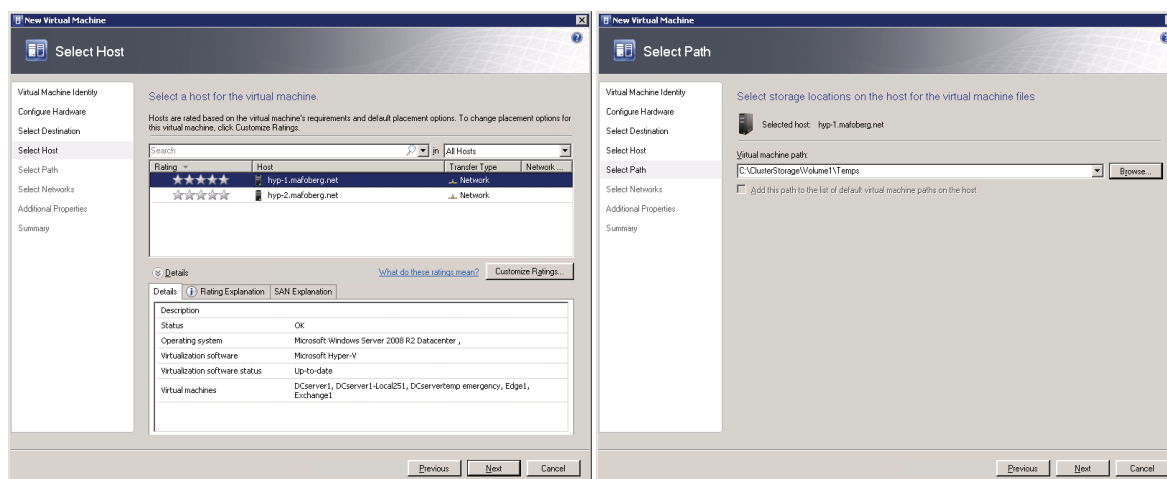
I Figur 209 Setter vi blant annet opp maskinvaren som templatene skal inneholde. Denne maskinvare kan forandres når maskiner som lages ut ifra denne er satt opp. Vi har mulighet til å legge til og fjerne adaptore av forskjellige typer og montering av image-filer mot CD/DVD-stasjoner og liknende.

Det kan også være man har satt opp maskinvareprofiler til bruk på forhånd. Da kan man lett velge dette ut ifra rullegardinen øverst.



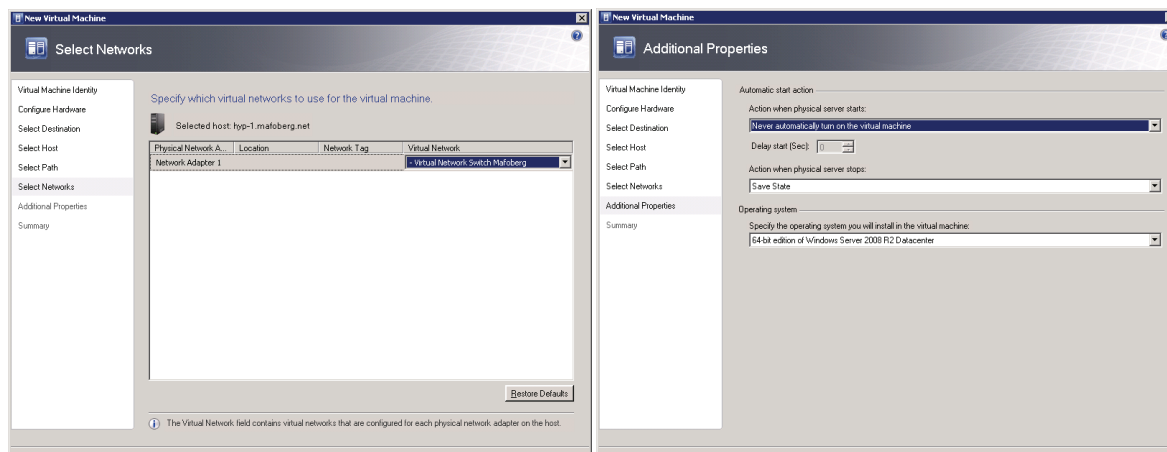
Figur 210 - Plassering av klonen

Vi plasserer filene (diskfil, maskinvareprofil med mer) på en fysisk host siden vi her kjører SCVMM som en virtuell maskin fra før. SCVMM-maskinen bør derfor ikke brukes som en library for å lagre kopier av andre virtuelle maskiner. SCVMM bør som omtalt tidligere kjøres på en egen fysisk maskin.



Figur 211 - Velge host for kloner og sted for plassering

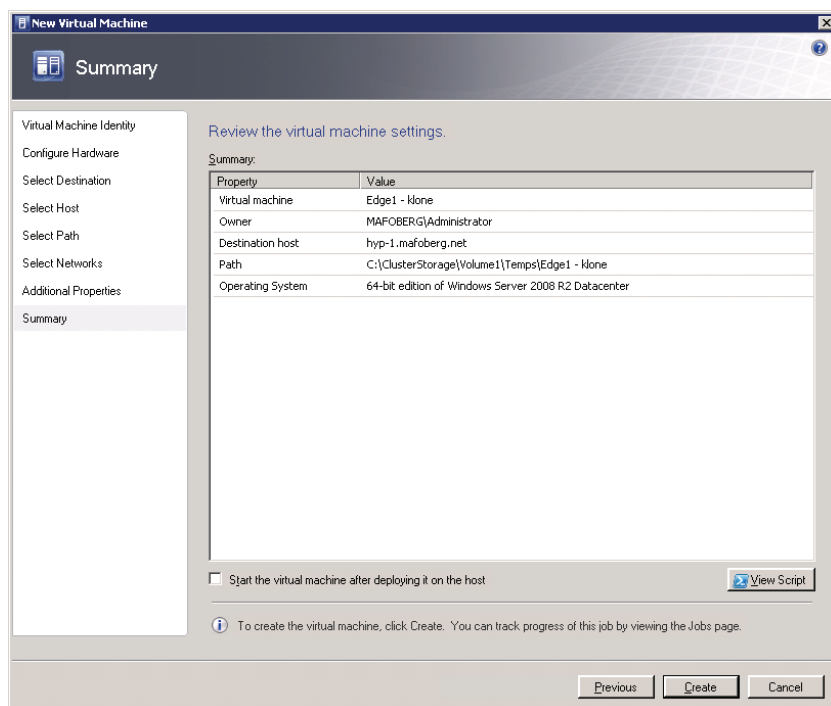
Vi velger host som klonefilene skal ligge på. Hos oss er HYP-1 maskinen vi velger å ta hånd om de virtuelle maskinene og klonene. Vi lagrer maskininformasjonen på en templatesmappe vi har opprettet på clusteret (ClusterStorage\Volume1\Temps). Vi bruker denne mappen både til kloner og templates vi oppretter i vårt tilfelle. Det kan være lurt å skille disse av hverandre i hver sin mappe om man får mange av disse.



Figur 212 - Status for klonen

Vi får mulighet til å sette status for klonen. Hvilket nettverk den skal være koblet til vil her være likegyldig siden den ikke skal skrues på.

Status på maskinen for om den skal skrues av, hvordan den skal lagres med mer, trenger vi heller ikke sette opp nå siden vi ikke skal gå ut i fra selve klonen for å lage en ny maskin, men bruk den originale maskinen å gjøre det fra.

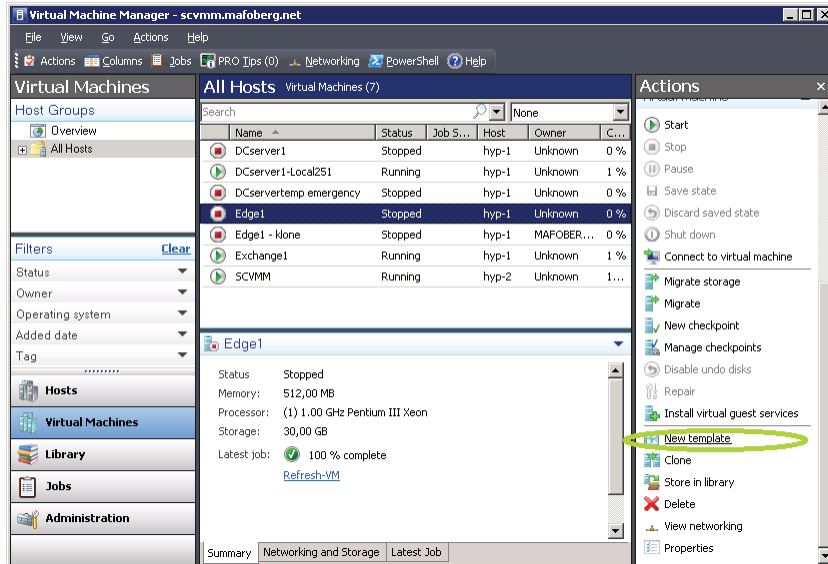


Figur 213 - Kloneoppsummering

Oppsummering av den nye klonens opprettelse. Vi har mulighet til å se scriptet som kan kjøres med PowerShell for å gjøre det vi har gått igjennom i denne veiviseren.

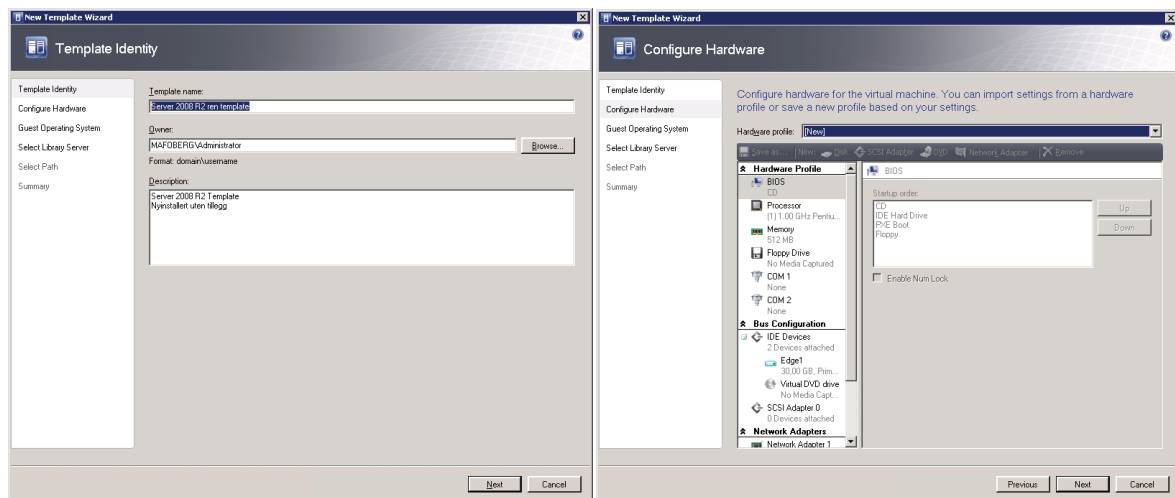
3.2.3. Lage template av en virtuell maskin

Vi har sikret oss en maskin å ta utgangspunkt i ved å lage en kloner først. Da gjør det ikke noe om det skulle skje noe skulle skje med den. Når klonen er laget kan vi ta utgangspunkt i fra Edge1-maskinen. Denne veiviseren vil være ganske lik kloningsveiviseren.



Figur 214 - Valg av maskin for template

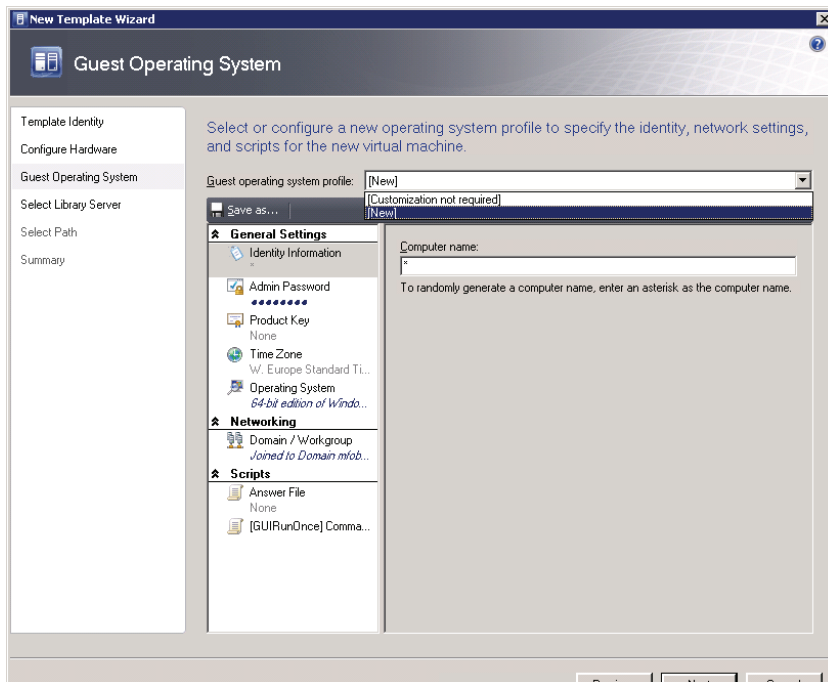
Vi velger Edge1-maskinen og velger *New Template* fra menyen *Action*.



Figur 215 - Template navn og hardware

Vi gir templatet et navn som passer til denne med passende beskrivelse for hva denne skal inneholde.

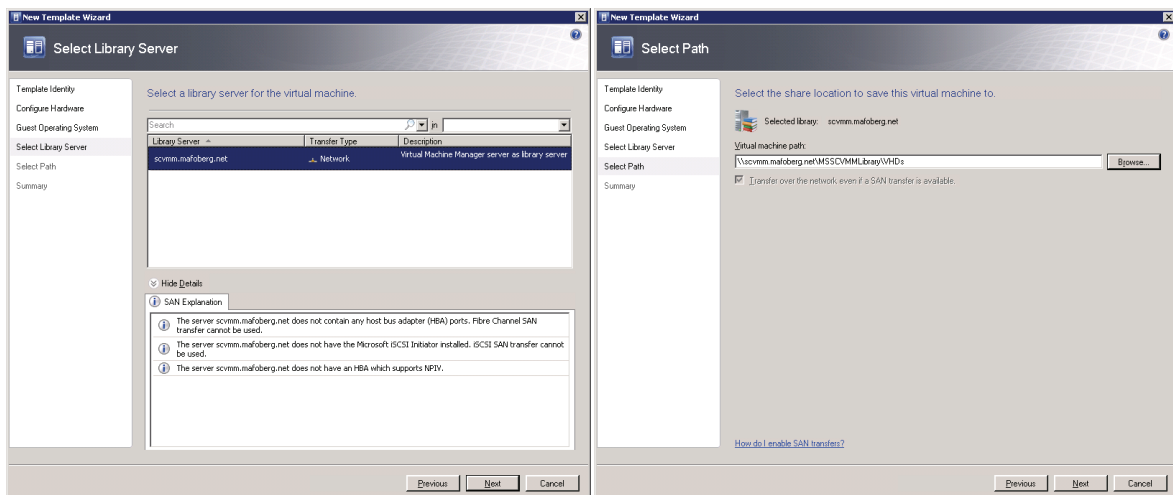
Hardwarekonfigurasjon vil vi ikke få tilgang til å redigere her. Siden maskinen vi valgte skal være utgangspunkt for templatet, må maskinen på forhånd være konfigurert hardwaremessig før den brukes som utgangspunkt.



Figur 216 - Template systemprofiloppsett

For å gjøre maskinen klar med interne data, får vi mulighet til å sette opp informasjon direkte for det interne systemet. Datamaskinnavn kan vi gjøre autogenerated ved å sette "*" som wildcard. Passord for bruker som skal opprettes med mer, kan også settes. Lisensnøkkel kan også settes opp.

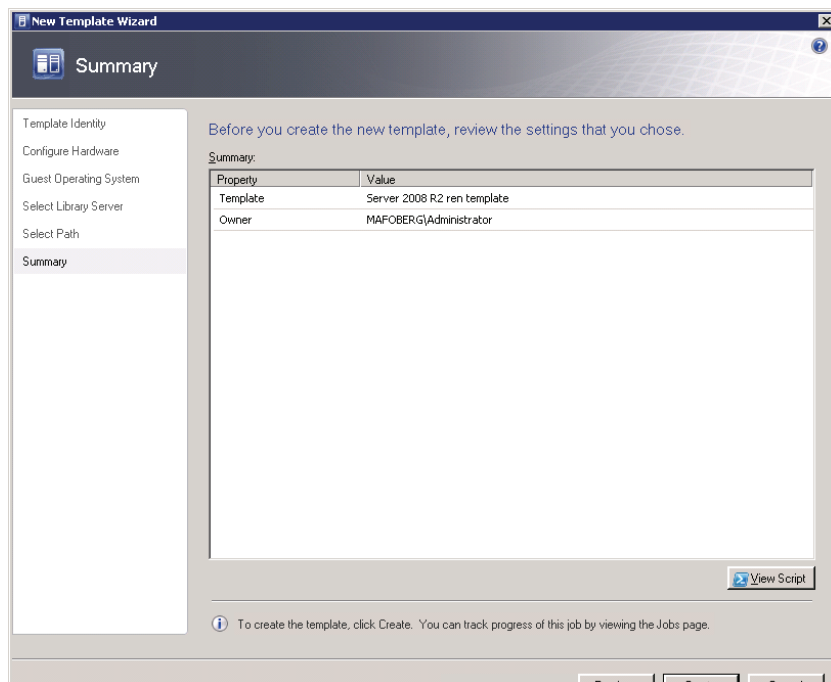
Dersom vi ikke ønsker å sette opp disse alternativene på forhånd, kan vi velge *Costumation not required* i rullegardinen over. Da slipper vi å gi informasjon for det interne systemet, men kan gi dette senere når vi skal ta den i bruk.



Figur 217 - Server og maskinpath

Vi velger hvilken libraryserver vi skal lagre templatene på. Templateen er nødt til å lagres på en library-server. Det er nok plass i vår tilfelle for å vise hvordan dette gjøres, men SCVMM maskinen bør ha egen plass for dette eller lagre disse på et SAN eller liknende.

Vi velger også sti for hvor denne skal lagres.



Figur 218 - Template summary

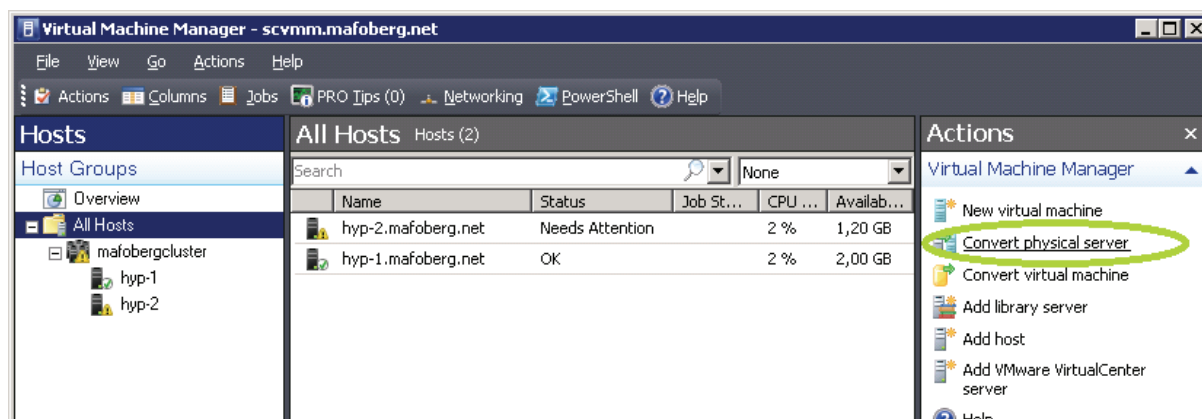
Vi vil også her få en oppsummering av gjennomføringen av å lage en template. Vi vil også få muligheten til å se et autogenerated Powershellscript for kjøring av dette.

Etter kjøring vil vi ha en maskin som heter *Edge1-clone*.

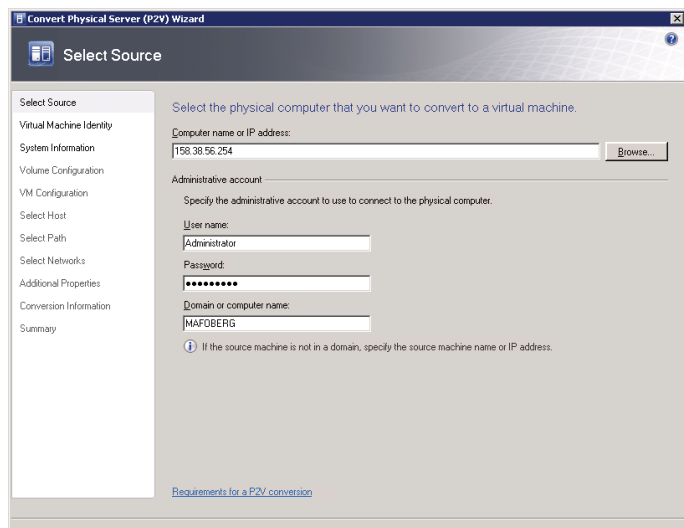
3.2.4. Konvertere en fysisk maskin til en virtuell maskin

Vi vil ta i bruk en Windows Server 2008 x86 når vi her skal konvertere over til en virtuell maskin. Denne maskinen kjører på en intelplattform fra før. Alle drivere og oversettelinger av maskinvare vil konverteringen i SCVMM ta seg av.

For å starte veiviseren, velger vi *Convert physical server* fra VM manageren.



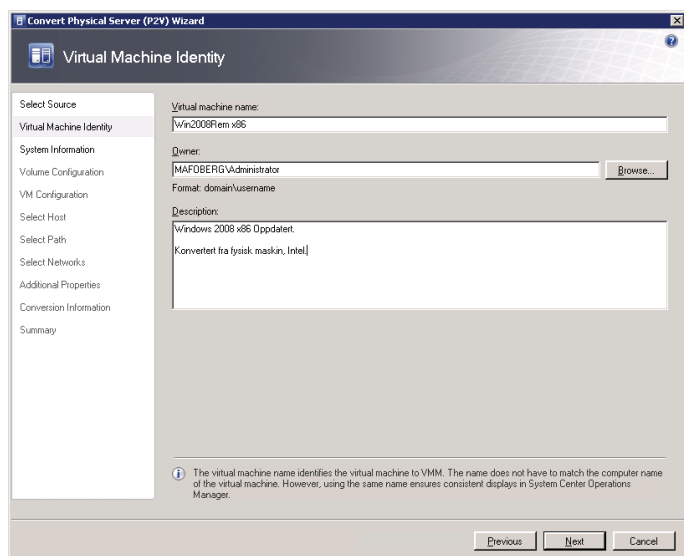
Figur 219 - Konvertering av fysisk maskin (1)



Figur 220 - Konverteringsveiviser (1)

Det første vi kommer til er å angi datamaskinnavn eller ipadresse. Vi angir ipadressen siden vi kjenner denne best.

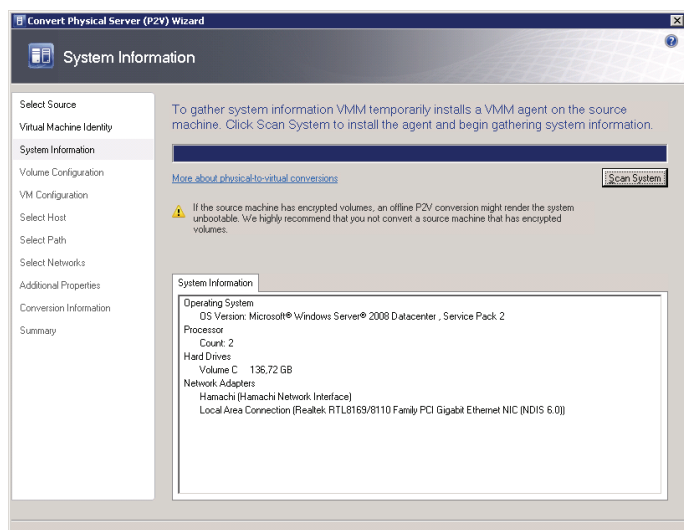
For å logge inn på den å få full tilgang til alt på den, må vi ha en bruker som har adgang til dette. Administrator. Vi må også oppgi datamaskinnavnet eller domenenavn til denne siden den er meldt inn i domenet vårt. Uten domenet må datamaskinnavnet brukes.



Figur 221 - Konverteringsveiviser (2) - destinasjon

Her må vi oppgi den virtuelle identiteten. Det er smart å legge til informasjon så man senere husker hva maskinen dreide seg om, hvor den kom i fra og liknende.

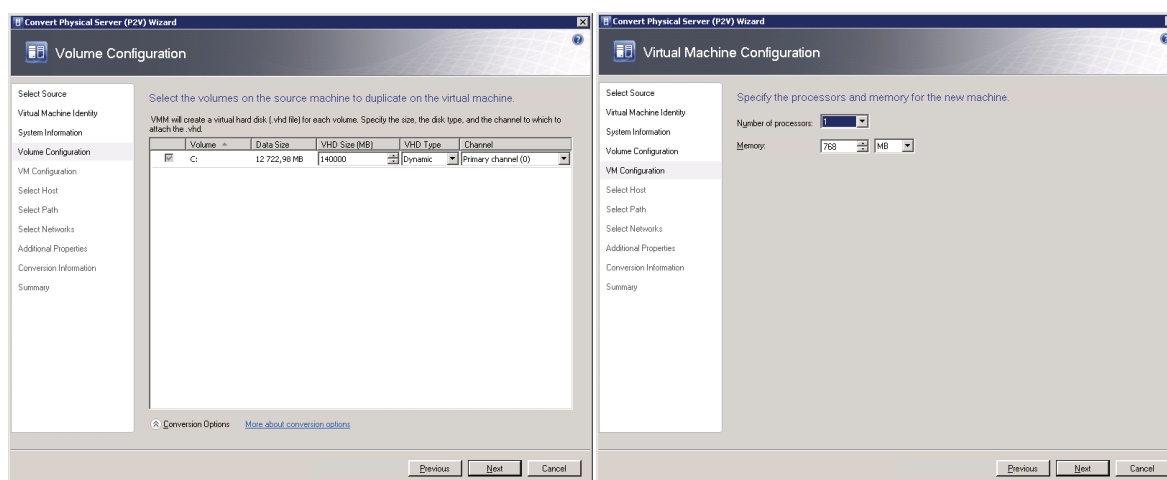
Vi lar Domeneadministratoren være eier av den virtuelle maskinen som lages.



Figur 222 - Konverteringsveiviser (3) – informasjonssjekk

Før vi kan scanne systemet på en Windows Server 2008 x86, må den være helt oppdatert med servicepack 2 og videre sikkerhetsoppdateringer. Komplikasjoner som kan oppstå om man ikke gjør dette er autentikasjonsproblemer ved fjernstyring av maskinen. Dette er det kommet ut en rekke oppdateringer for. I tillegg må brannmuren åpnes med tilgang. Dette er også lagt til med oppdateringene. Filsystemet på maskinen må heller ikke være kryptert da dette kan gi komplikasjoner for bruk av data på den nye virtuelle maskinen.

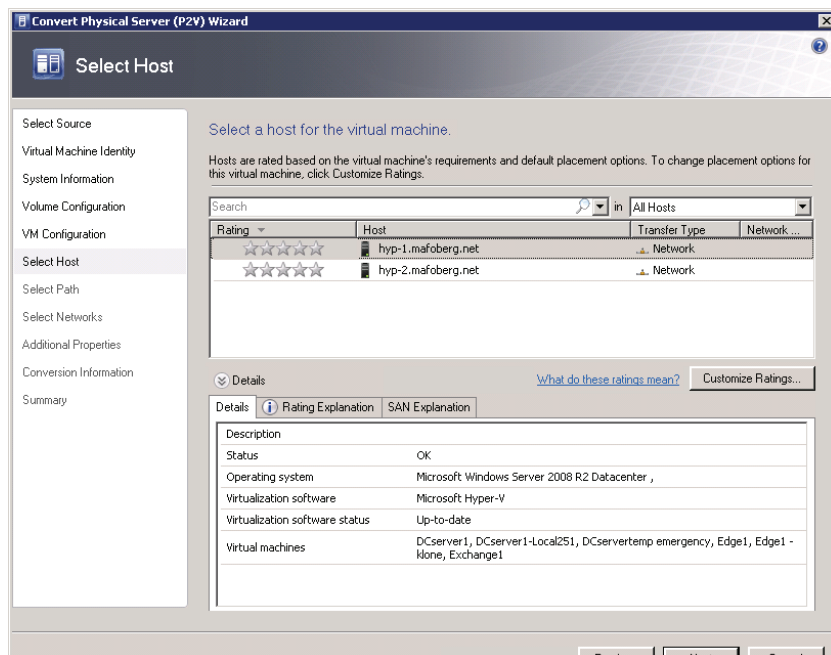
Etter scanningen vil vi få som skjermbildet viser, en oppsummering av systemdelene som er viktige.



Figur 223 - Konverteringsveiviser (4) – HW-oppsett

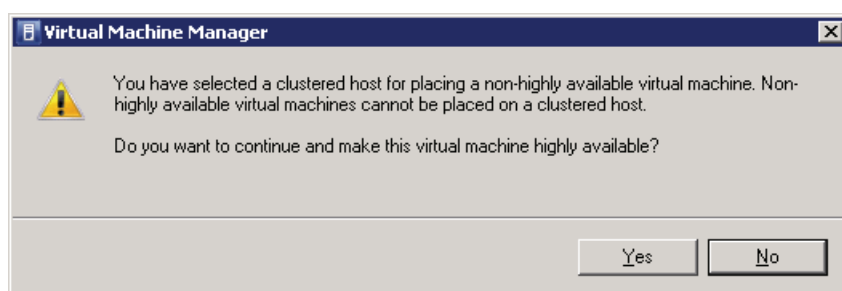
Vi får i *Volume Configuration* beskjed om størrelse på volum av partisjoner som skal konverteres. Dette er ikke konfigurertbart, siden dette kun er valg for hvilke partisjoner man vil overføre.

I *Virtual Machine Configuration* får vi derimot mulighet til å velge antall kjerner og minnestørrelse. Dette kan være greit å senke dersom man ikke har behov for mer. Dette gjør vi i vår tilfelle siden dette er en visning om hvordan man går frem.



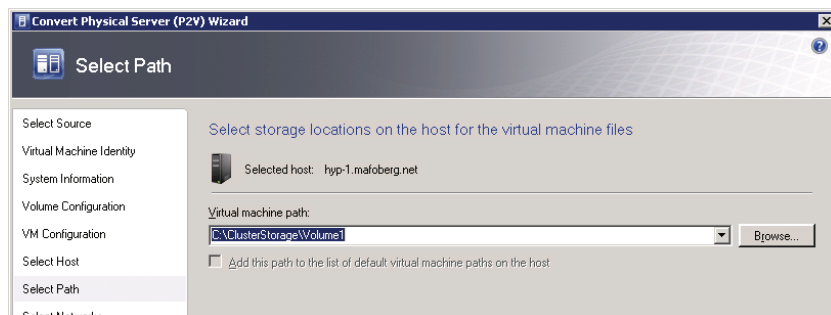
Figur 224 - Konverteringsveiviser (5) – Hostplassering

Som vi ser ut i fra Figur 224 er allerede alle hoster ranket til 0. Dette er fordi hostene allerede har for lite ressurser til å kjøre disse maskinene i vårt testsenter. Både for lite minne, diskplass på SAN med mer. Men siden en del av disse maskinene ikke er aktive, gjør det ikke noe i dette tilfelle å plassere den nye virtuelle maskinen til HYP-1.



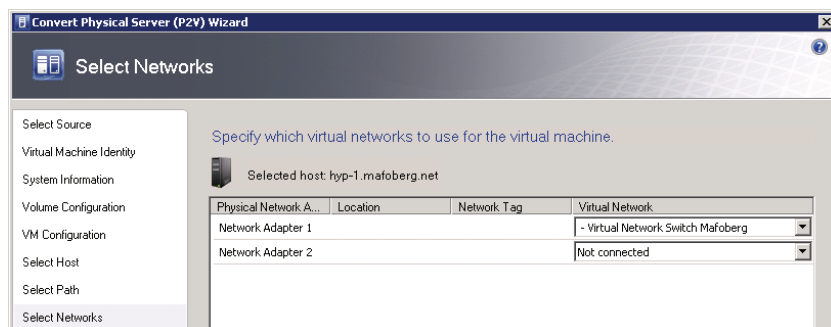
Figur 225 - Konverteringsveiviser (6) - Highly available

I vårt tilfelle har vi satt begge hostene i cluster som gjør at konvertereren krever å gjøre den highly available om den skal ligge der. Hadde vi hatt en host til som ikke var tilmeldt clusteret, kunne vi lagt denne der for å se at den var i orden før vi flyttet den til clusteret om vi ønsker dette. Men vi velger her å fortsette konverteringen direkte til clusteret.



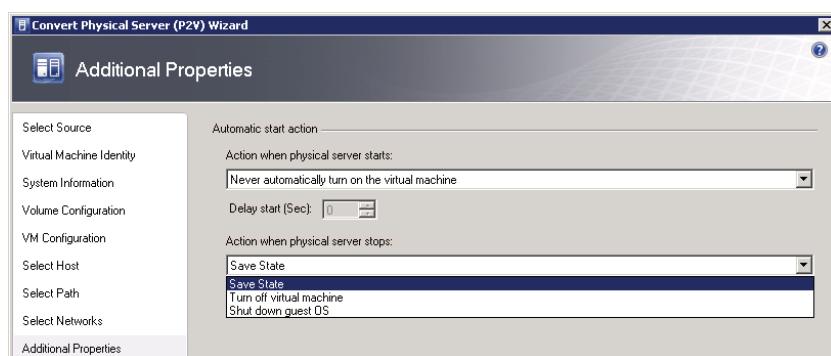
Figur 226 - Konverteringsveiviser (7) – clusterpath

Videre får vi valg om hvor på clusteret den skal ligge. Her ligger alle maskinene fra før direkte, så vi gjør ingen forskjell i vårt tilfelle. Men det kan lønne seg om det blir mange servere å kategorisere dem. Mailservere for seg selv, testservere for seg selv og liknende.



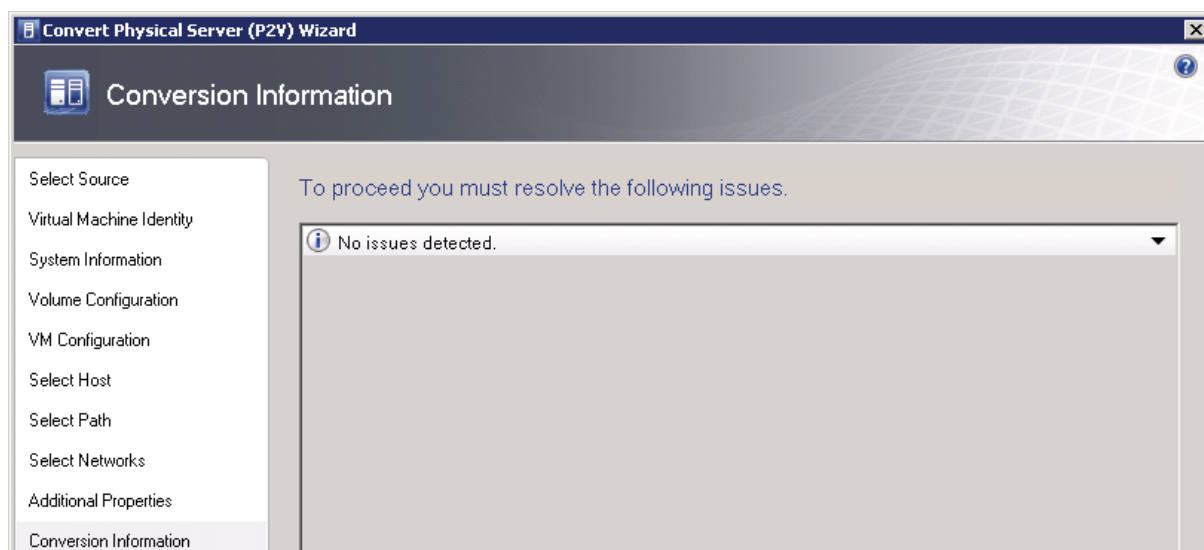
Figur 227 - Konverteringsveiviser (8) – NICs

Som vi ser kommer det opp forslag om to nettverkskort. Dette fordi den fysiske maskinen hadde 2 nettverkskort fra før. Vi velger å kople til adapter 1 til den virtuelle switchen på hosten (HYP-1). Den andre har ingen behov for tilkopling.



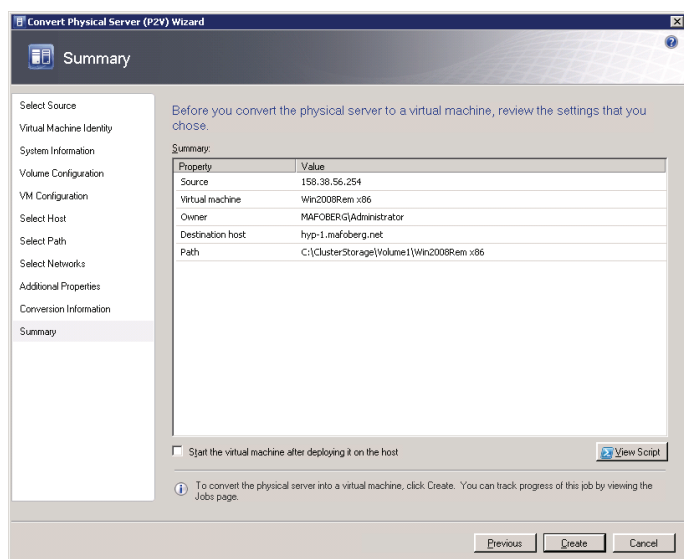
Figur 228 - Konverteringsveiviser (9) – Tilleggsoppsett

Som vi ser har vi mulighet til å sette avslutningsalternativer til maskinen. Ved opprettelse av, konvertering til virtuell maskin vil vi få valg om hvordan den skal oppføre seg i forhold til hosten. *Save state* blir som standard foreslått.



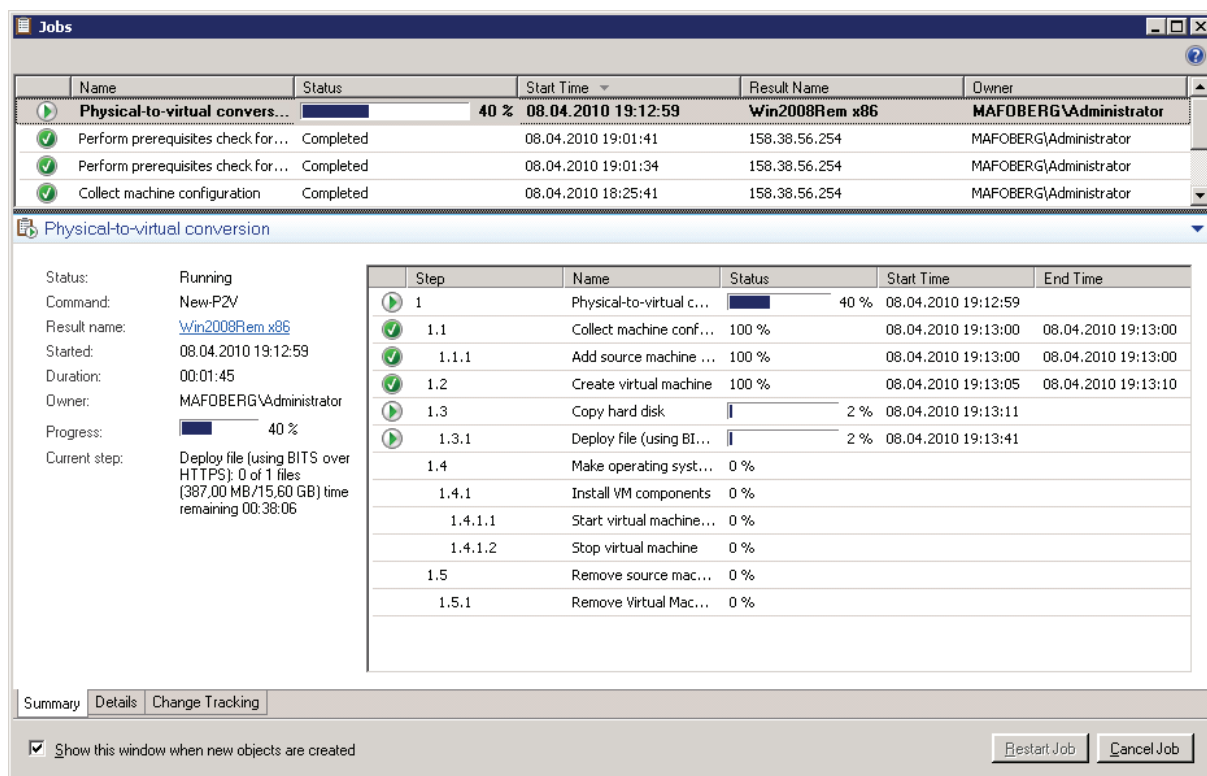
Figur 229 - Konverteringsveiviser (10) – Sjekk av hinder

Det viser seg her at vi ikke har hinder som hindrer oss i å sette i gang operasjonen.



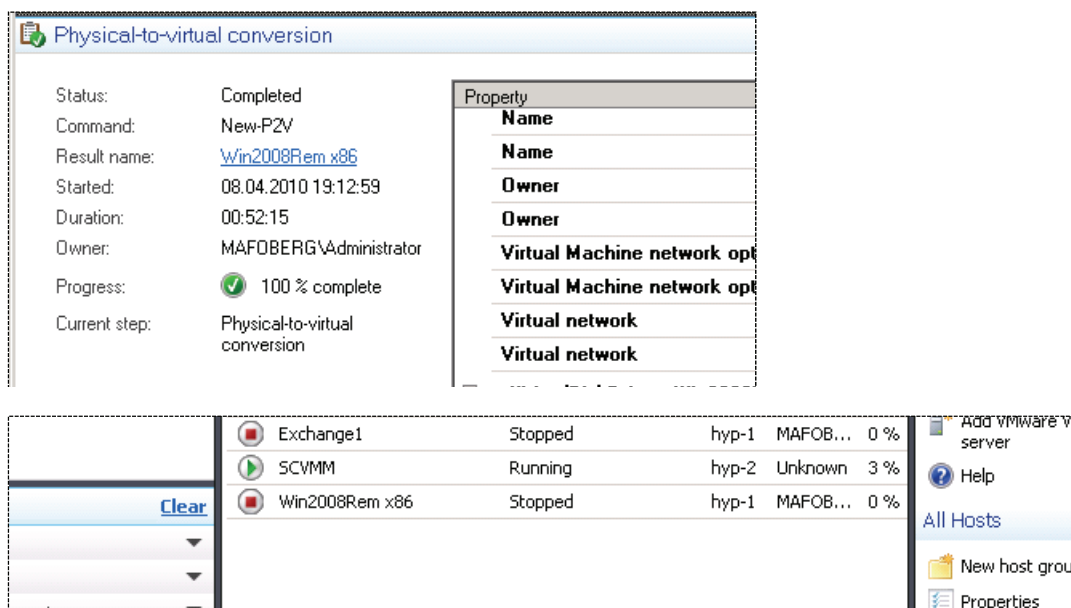
Figur 230 - Konverteringsveiviser (11) – Oppsummering

Vi får en oppsummering av maskin, plassering, adresser med mer. Se til at disse er riktige før man begynner. Operasjonen kan ta en god del tid dersom det er mye data på disken. Dette har vi her ordnet på forhånd for at dette ikke skal ta for lang tid. Skulle det ta lang tid, har vi mulighet til å autostarte maskinen etter konvertering (avkryssningsboks). Det er også lurt å ta vare på scriptet til denne maskine (knappen i høyre hjørnet) da dette kan være til stor nytte om vi skal gjøre dette for en rekke servere. Bruk av PowerShell vil være meget nyttig i massekonvertering.



Figur 231 - Konverteringsveiviser (12) – Konvertering

Det er flere steg i denne jobben. I Figur 231 ser vi fremskrittene. Det kan være en fordel å sette på dette over en tid man gjør annen jobb eller over natten når ikke backuptjenester er i gang.

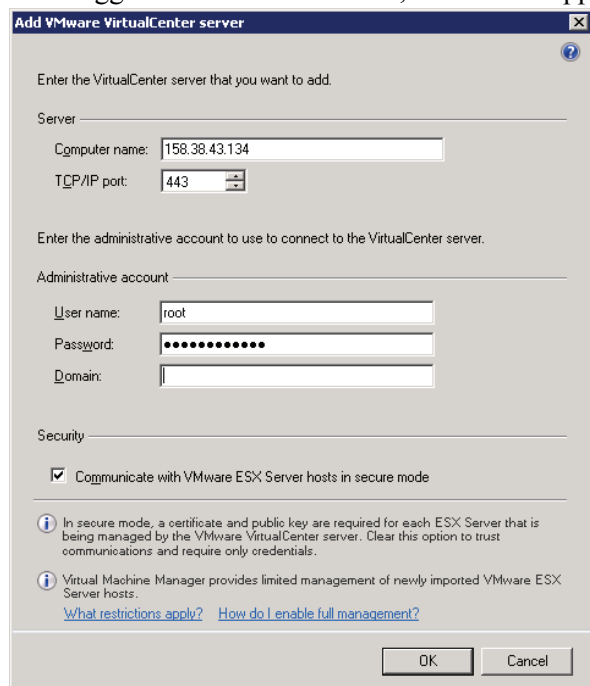


Figur 232 - Konverteringsveiviser (13) – ferdig

Vi ser at maskinen er konvertert til en virtuell maskin på clusteret. Denne er nå klar til å kjøres. Skru gjerne av den fysiske maskinen før kjøring av denne. For å være helt sikker, og om man har plass tilgjengelig, kan man gjerne kloner denne nye maskinen, eller ta snapshot (checkpoint i SCVMM) for å kunne gå tilbake til konverteringspunktet.

3.2.5. Legge til en host fra et annet virtualiseringsmiljø

For å legge til hoster fra VMware, må vi først opprette en tilkobling mot et VCenter¹⁴. Et VCenter er en styringsplattform for et virtuelt miljø i VMware.



Figur 233 - VMware kobling

Vi angir IPadressen eller datamaskinnavnet for denne serveren. Deretter er standard kommunikasjonsport 443.

Brukernavnet for en VCenter er vanligvis root fordi esx-tjenere kjører på linux-plattform.

Denne er ikke satt i et domene, og vi lar dette stå blankt.

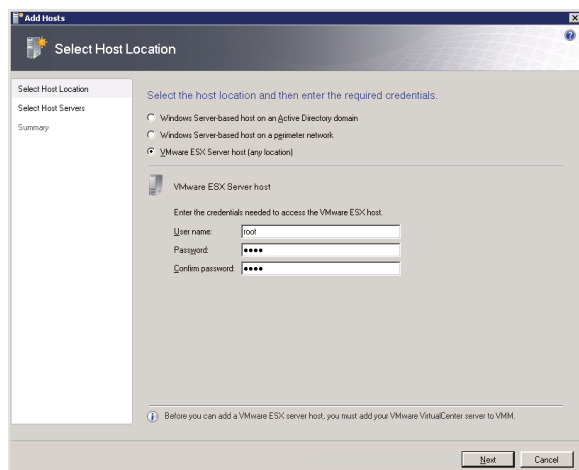
VCenter-erveren er satt opp til å kommunisere med sikker overføring.

Vi ser som kommentar at vi ikke har fulle styringsmuligheter fra VMM til VCenter, men det kan hjelpe om det kun er et lite nettverk eller oppsett av VMware-miljø. Dette fordi Hyper-V og miljøene rundt dette er i en tidlig fase for

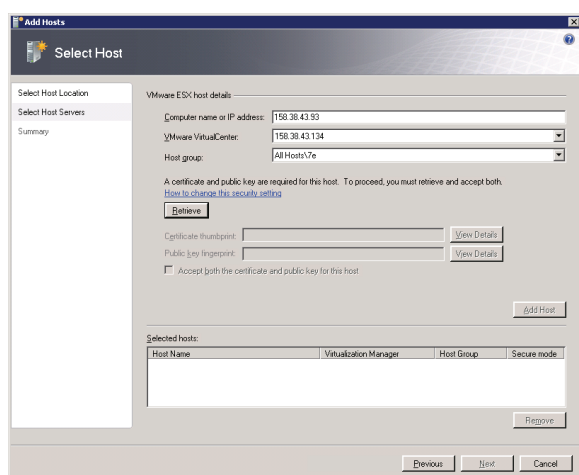
Microsoft. Det vil kunne bli utviklet mer hvis behovet og samarbeidet med VMware utvikler seg.

¹⁴ VCenter (Virtual Center) er VMware sin virtualiseringsagent for ESX-miljø.

Da vi har koblet til, kan vi opprette en tilkobling mot en ESX-server. Vi velger da *Add Host* på managementmenyen til høyre i SCVMM.



Figur 234 - SCVMM ESX-tilkobling



Figur 235 - SCVMM VCenter Host

Vi velger *VMware ESX Server host (any location)*. Vi får da angi brukernavn og passord til ESX-hosten.

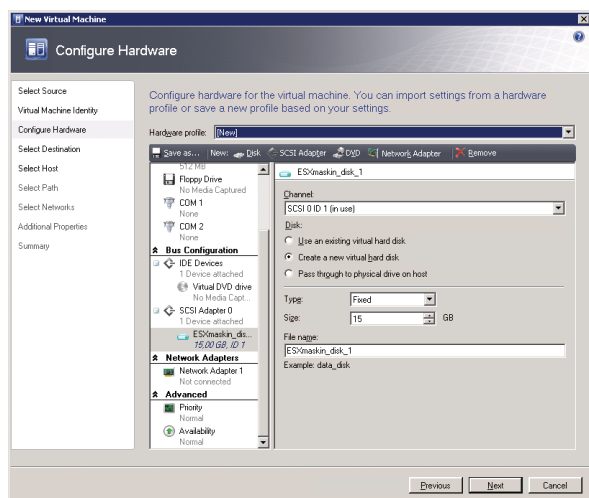
Her har vi så IPadresse eller maskinnavn mot hosten vi skal legge til. Denne går igjennom VCenter-systemet til ESX-tjeneren.

I gruppen under kan vi velge hvilken gruppe av hosts som er opprettet i VCenter.

Deretter må vi få sertifikat for sikker tilkobling mot serveren.

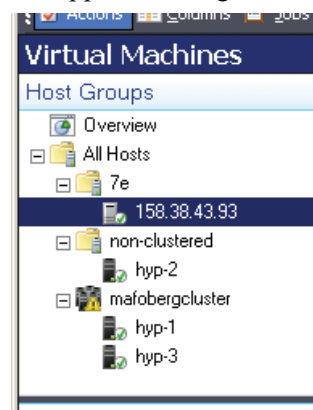
Når dette er gjort, får vi en oppsummering som

oppsummerer host, miljø, sertifikat og det vi har valgt.



Figur 237 - SCVMM ESX maskin maskinprofil

Her ser vi at en host fra VMware er lagt til i en egen hostgruppe vi har opprettet. Hosten er markert grå som skiller den ut ifra våre egne hoster i Hyper-V miljøet.



Figur 236 - SCVMM ESXhost

Da er det bare å opprette en maskin på denne ESX-hosten. Vi markerer denne hosten og velger *New virtual Machine* fra managementmenyen.

Dertter gjør vi som da vi lagde en ny virtuell maskin. Opprettet ny virtuell blank disk. Satte

navn og eier på maskinen. Deretter kommer vi til maskinvareoppsett. ESX krever SCSI¹⁵ grensesnitt for virtuelle harddisker. Harddiskene på en ESX-server støtter heller ikke dynamiske disk, slik at diskfilene kan variere i størrelse etter data på dem.

Vi klikker på snarveisknappen øverst for å legge til en SCSI-adapter. Deretter går vi på disken under *IDE-devices* og velger *Channel* satt til for eksempel 1. Disken blir da automatisk flyttet under *SCSI adapter*.

Deretter må vi velge type *Fixed* som krav for ESX.

Når vi velger ESX-hosten får vi eventuelt feilmeldinger under fanen *Rating Explanation*. Enten krav som ikke opprettholdes, eller advarsler om utilgjengelige funksjoner.

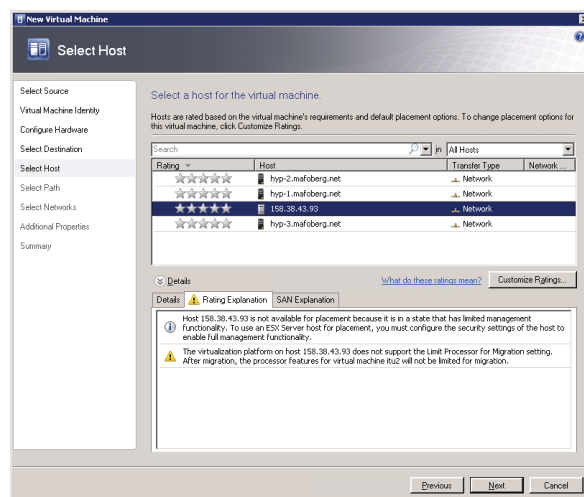
Som vi ser i Figur 233 at vi har fått melding om migreringsoppsett som ikke er kompatibel.

Dette går bra for testoppsett og vanlig kjøring uten spesielle høye krav.

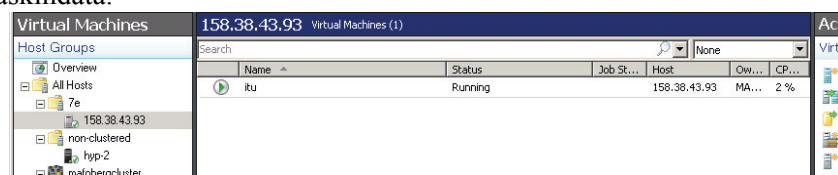
Deretter får vi velge plassering av maskindata.

Nettverksoppsett blir valg av nettverksenheter på ESX-serveren. Til slutt får vi oppstartsoppførsel og oppsummering av maskinen.

Maskinen på ESX-hosten blir så opprettet.

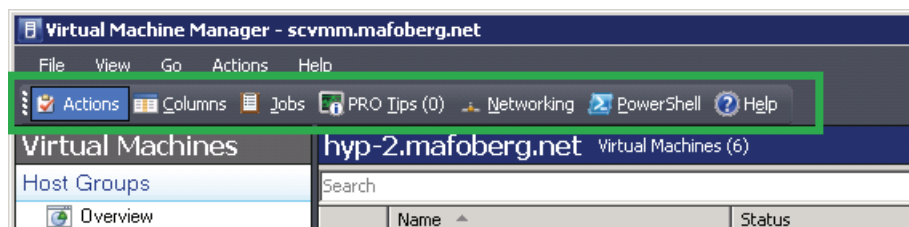


Figur 238 - SCVMM ESXhost meldinger



Figur 239 - SCVMM ESXmaskin

3.2.6. SCVMM Oppgavelinjen



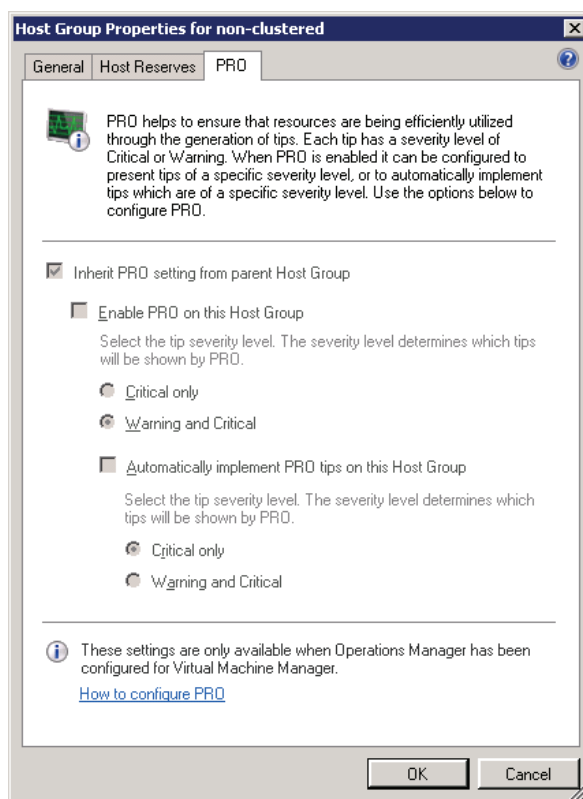
Figur 240 - Oppgavelinja (Taskpane)

Øverst i SCVMM har vi en oppgavelinje med ofte brukte snarveier.

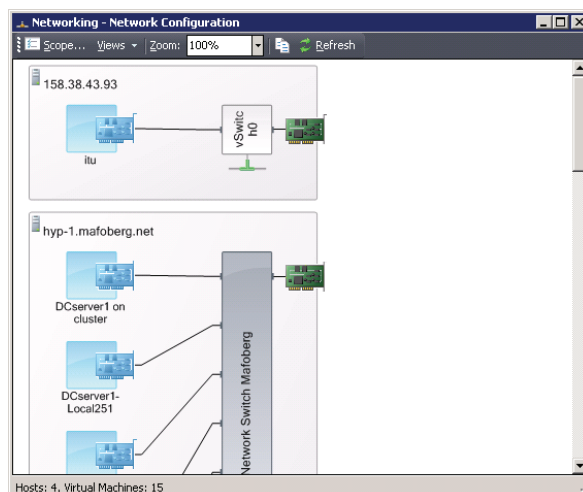
- **Actions** – Slår av og på actionsmenyen til høyre i SCVMM. Det kan gi større oversikt om man slår denne av. Handlingene i Acionmenyen nåes ved hjelp av høyreklikk på objekter i SCVMM.

¹⁵ SCSI - Small Computer System Interface.

- *Columns* – Valg av informasjon i hovedvinduet hvor maskinlista og hostlista vises. Vi kan liste flere typer bruk, som minne cpu-telling, disktilkoblinger med mer.
- *Jobs* – Lista over jobber som utføres og er blitt utført. Opprettelse av maskiner, flytting av maskiner, omjusteringer, forhold og det som kan gjøres i SCVMM.
- *PRO Tips* – Dette er et oppsett for å behandle virtuelle maskiner effektivt. For å aktivere dette går man til hostgruppen man vil aktivere dette på og velger properties. Deretter velger man PRO-fanen. Men her ser vi at alle mulighetene er deaktivert. Microsoft Operations Manager må konfigureres for SCVMM for at denne funksjonen skal kunne tre i kraft. Når denne er aktivert, får vi hint i lister over hva som kan gjøres for å forbedre effektivitet i det virtuelle miljøet under hostgruppene. Dette kan være en stor fordel når man bruker clustrede miljøer. Man kan også ekskludere enkeltmaskiner fra PRO Tips konfigureringen.
- *Networking* – Nettverkskonfigurasjon med grafisk oversikt. Her kan man få en fin oversikt over alle maskiner og nettverksenheter vi har i nettverket. I Figur 237 ser vi et utsnitt av nettverk i vår virtuelle miljø. I menyen *Scopes* har vi mulighet til å velge hosts-grupper å vise.
- *PowerShell* – Powershell med oppstart av snapins for SCVMM. Hvis man kjører PowerShell rent, kan man kjøre kommandoen
Add-PSSnapin Microsoft.SystemCenter.VirtualMachineManager
- *Help* – snarvei til hjelp for det meste i SCVMM.



Figur 241 - SCVMM PRO Tips



Figur 242 - SCVMM Networking Configuration

3.2.7. SCVMM Library server

En SCVMM server er også laget en lagringsfunksjon for maskiner, disk, floppydisk, ISOfiler, maskinmaler, Answerings-filer, PowerShell scripts, Hardwareprofiler og gjesteoperativsystem. Dette er en fin måte å holde oversikt over ressurser og utstyr man har tilgjengelig i nettverket. Men dette gjør at man helst burde ha andre kriterier for hvordan å sette opp maskinvare.

3.2.7.1. Oppsett av Library Server

En SCVMM Libraryserver krever lagringsplass og båndbredde. Dette kan settes opp på forskjellige måter.

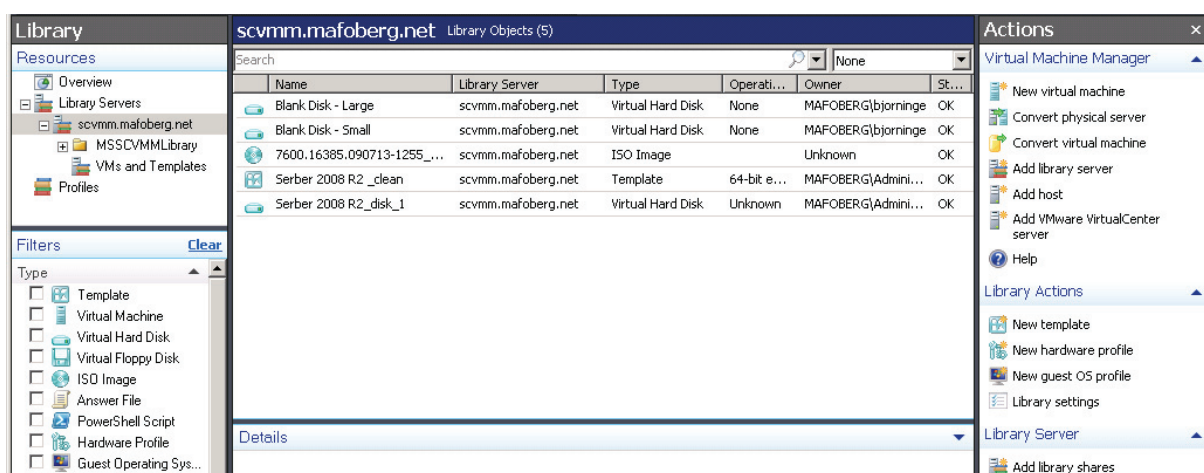
- Library server som en virtuell maskin i et cluster bør ikke lagre maskiner på sin egen server. En slik server kan brukes til å lagre hardware-profiler, Answering files, Virtuell maskin uten harddisk (eller tom sådan), floppydisker og eventuelt små isofiler.
- Library server på en fysisk maskin som er direkte knyttet til et SAN eller NAS for lagring har bedre forutsetninger for å ta vare på virtuelle maskiner, store ISO-filer, Templates og virtuelle harddisker. Denne bør være knyttet til hoster med stor båndbredde siden det er store mengder data som skal overføres når en skal behandle datamaskiner.

Scenarier

En SCVMM Library Server som ligger som en virtuell maskin i et cluster eller på en egen unclustered host kan få problemer med å hente maskiner for oppretting av templates fra maskiner som ligger på en annen host. Siden en template lagres i selve den virtuelle maskinen, vil den måtte lagre en virtuell maskin i den virtuelle maskinen. Det kan være unødvendig.

En SCVMM Library Server som ligger på en egen fysisk maskin, vil enklere kunne lagre virtuelle maskiner og disk på en fysisk disk og sende disse til de fysiske hostene uten å måtte ut igjennom en virtuell maskin, men da rett fra en fysisk maskin til eventuelt et cluster eller en host.

3.2.7.2. Innhold i en Library Server



Figur 243 - SCVMM Library server

Til venstre ser vi treet for lagring av ressurser i SCVMM. *Overview* gir oss en oversikt over forbruk av plass på serveren og hva som er lagret av typer ressurser.

Library Servers gir oss oversikt over library servere som vi har lagt til i dette konsollet. Vi har her kun scvmm.mafoberg.net som library server og som styringsserver. Dette er en virtuell maskin og vi har bestemt oss for å også bruke denne som en library server.

Katalogen *MSSCVMMLibrary* er fysisk plassert i filsystemet på SCVMMserveren

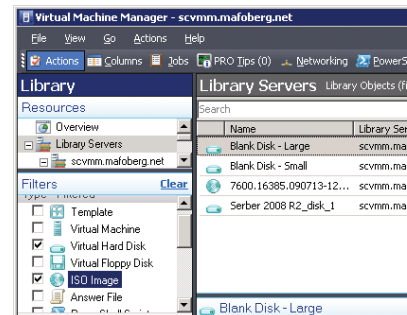
”C:\ProgramData\Virtual Machine Manager Library Files” er delt som

”\\scvmm.mafoberg.net\MSSCVMMLibrary”. Her lager vi oss en struktur over ISO-filer og diskfiler. Virtuelle maskiner som er lastet inn som templates og lagrede (stored) masiner kommer i kategorien *VMs and Templates*.

I *Profiles* nedenfor ligger maskinvareprofiler vi har laget oss og profiler fra guest operativsystemer.

I Figur 243 har vi en ISO-fil. Denne er lite beskrivende. Navnet hentes ut i fra filene, så filene bør ha beskrivende navn. Navn på maskiner blir gjerne beskrevet i det vi lager dem, eller forandrer dem. De vil derfor ofte være lett gjenkjennelig for dem som har laget dem.

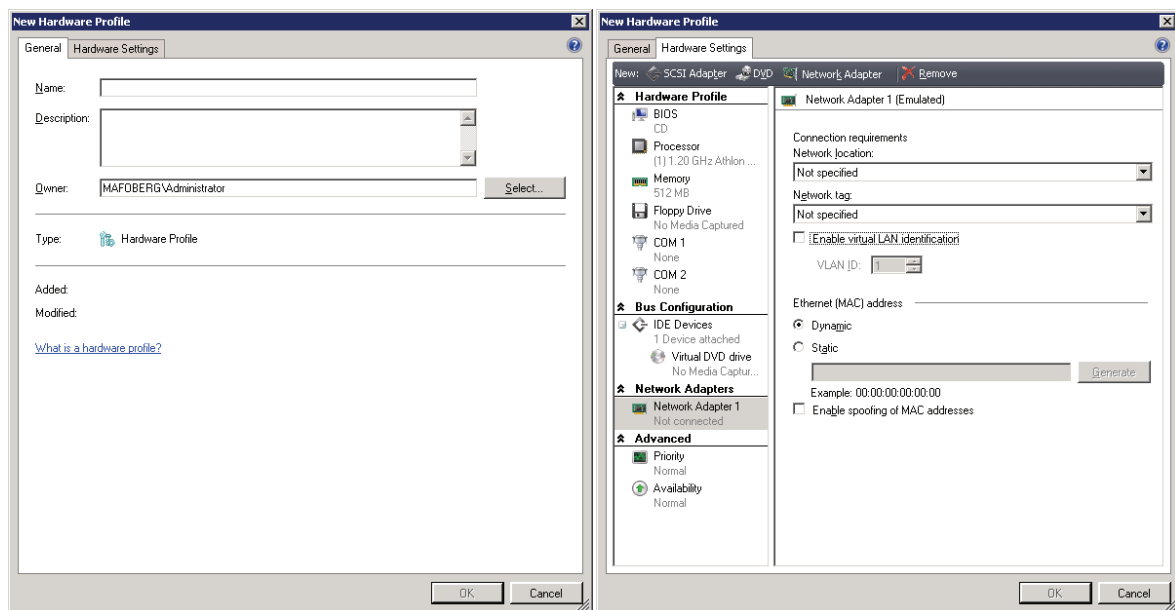
Hvis vi merker *Library Servers* vil vi få en liste over alle filer fra alle hoster. Hvis vi leter etter en bestemt fil kan vi da bruke *Filters* til å merke den type ressurser vi ønsker å få listet. Dette kan være en fordel om vi har veldig mange ressurser til sammen.



Figur 244 - SCVMM Library - ISOs og filters

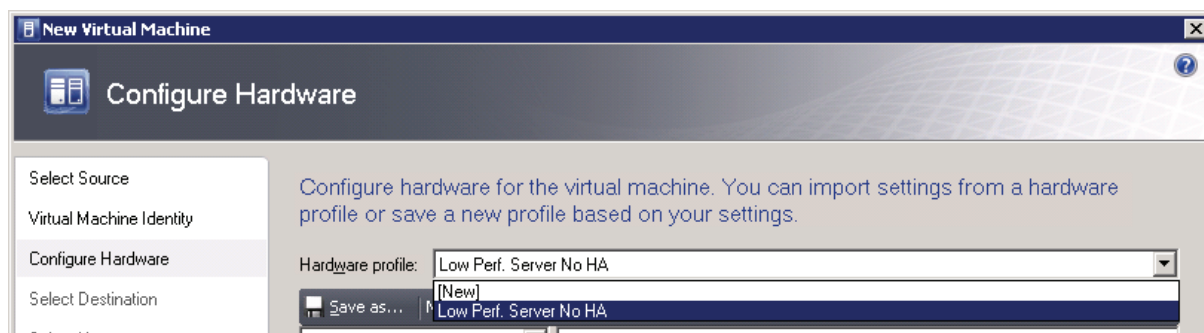
Hardware profiler

Om vi ofte oppretter maskiner av forskjellige typer, men også oppretter mange av like slag, kan det være en fordel å opprette maskinvareprofiler for enkelt å sette opp en bestemt type server, eller for en bestemt type klientmaskiner.



Figur 245 - SCVMM Library - Hardware profil

Når vi klikker på *New Hardware Profile* får opp oppsett for navn og maskinvareoppsett. I fanen *General* gir vi den beskrivende tittelen som vises i listen. I fanen *Hardware Settings* setter vi opp profilen for hardware. Her er det viktig å lage oppsett som passer til den type maskiner som skal lages ut ifra dem. Det gjør det greit å velge oppsett når vi skal lage en ny virtuell maskin. Det er mulig å forandre oppsettet når man tar denne i bruk. Dette gir hovedsakelig et utgangspunkt.

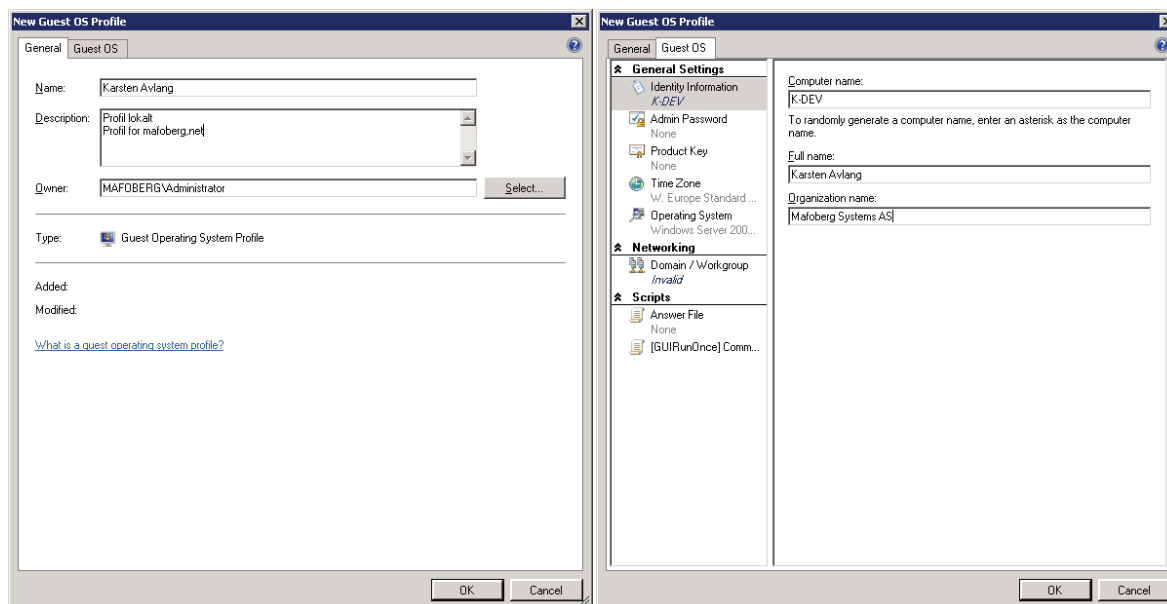


Figur 246 - Gjøre bruk av hardwareprofil

Opprettelse av ny maskin blir enklere.

New Guest OS Profile

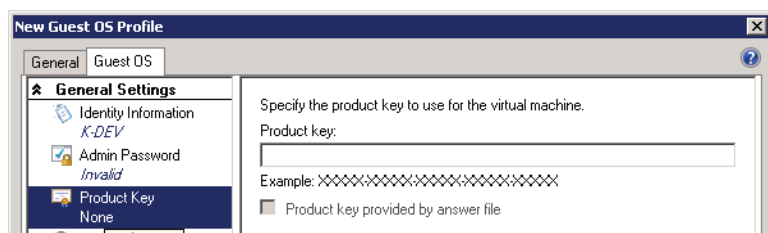
Dette er en hendig funksjon med SCVMM. Når man skal lage en template av en maskin med operativsystem installert, og hvor man har logget inn med brukere, kan man lagre brukere fra den maskinen til profiler. Vi kan også opprette nye profiler uten å være innlogget i maskinen først. Vi får samme type veiviser som for hardwareprofiler.



Figur 247 - New Guest OS Profile

I fanen *General* beskriver vi kort profilens innhold. I fanen *Guest OS* er mer omfattende. Her setter vi opp profilen mot maskinen og operativsystemet til maskinen som dataen skal inn på. Dette kan være offentlige profiler og personlige profiler. Her har vi tatt med en profil for en utvikler. Han kan trenge en resoft (reinstallering) eller opprettelse av nye maskiner for nye prosjekter. I første vindu har vi angitt maskinnavn og informasjon om bruker.

Under *Admin Password* oppgir vi passord for den lokale administratoren om vi kjenner dette.



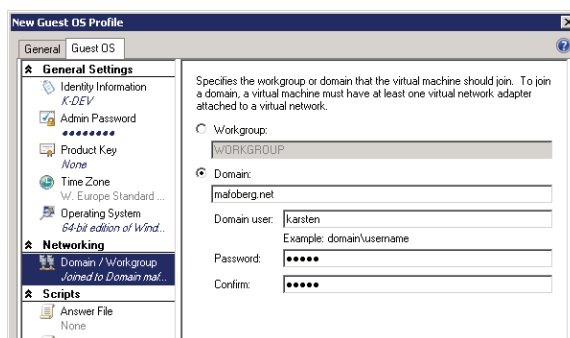
Figur 248 - New Guest OS Profile - Product key

Hvis vi har en produktnøkkel for maskinen, kan vi angi denne her. Der er da litt viktig at det er samme maskinvare vi har produktnøkkelen for, eller om vi har en volumnøkkel noe som er svært sjeldent. Vi kan velge å ha angitt denne nøkkelen i en senere svarfil for oppsettet. Hvis dette står tomt vil vi få en aktiveringsadvarsel når vi starter operativsystemet.

Time-zone angir tidsriktig informasjon.

Operating System kan vi velge hvilket operativsystem profilen skal gjelde for. Når vi velger her, kan noe av de tidligere informasjonsfeltene forandres. Enten fjernes eller det legges til andre felter. Alt etter operativsystem som velges.

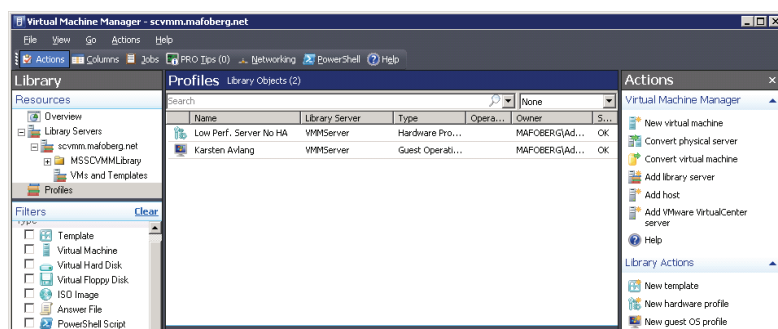
Hvis vi skal ha brukeren til automatisk å bli tilkoblet domene, har vi mulighet til å legge inn denne informasjonen direkte. Veiviseren vil allerede her teste ut om domene og profil er gyldig. Vi får da svar under *Domain / Workgroup* om informasjonen er invalid eller om den er tilkoblet domene. Hvis brukeren ikke skal være medlem av domenet, vil det ikke foretas noen sjekk, og det vil bli angitt en standard brukergruppe.



Figur 249 - New Guest OS Profile domene

I *Answer File* kan vi legge til script i form av .inf eller .xml fil som kan kjøre flere avanserte oppsett for brukeren om det skulle være flere oppsett.

Eventuelt kan kommandoer bli gitt i *[GUIRunOnce] Commands*. Dette er en registerenhet i registeret til windows som kun kjøres en gang. Her legges det inn nøkkelverdier direkte.

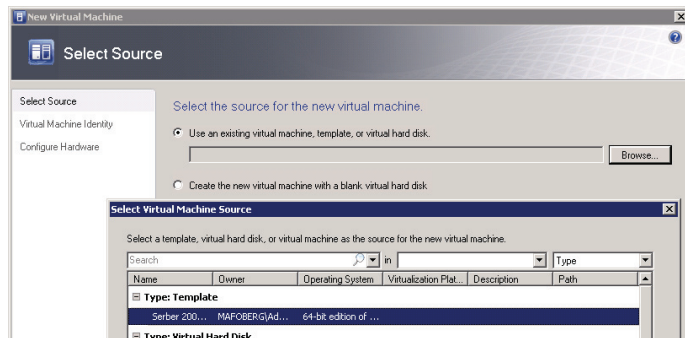


Figur 250 - Ny Guest OS Profile

Her ser vi at vi har fått opprettet en ny guestprofil som kan brukes mot en ny maskin.

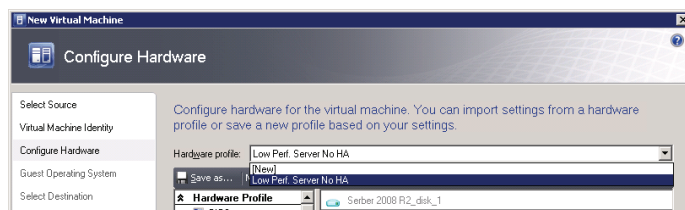
3.2.7.3. Ny maskin fra SCVMM Library Server

Når man har profiler og standardoppsett av servere og maskiner, blir det enklere å opprette maskiner. Vi har forhåndskonfigurert både maskinvare og innhold for brukere. Vi velger å lage ny maskin.



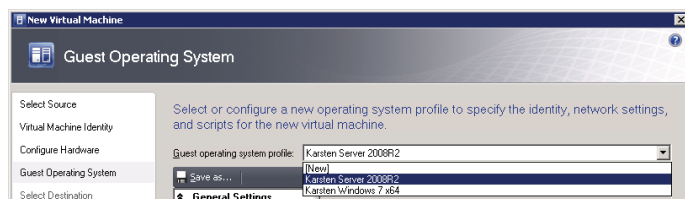
Figur 251 - Ny maskin ut fra template

Vi velger å lage ny maskin ut ifra en template. Vi har en Serber 2008 R2 som vi bruker. Deretter gir den et navn og beskrivelse.



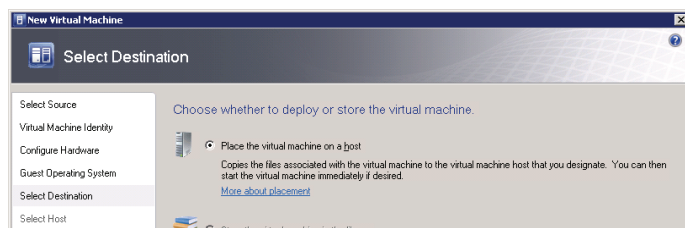
Figur 252 - Ny server fra SCVMM Library – hardwareprofil

Vi kan enkelt opprette fra hardwareprofil som er forhåndslaget.



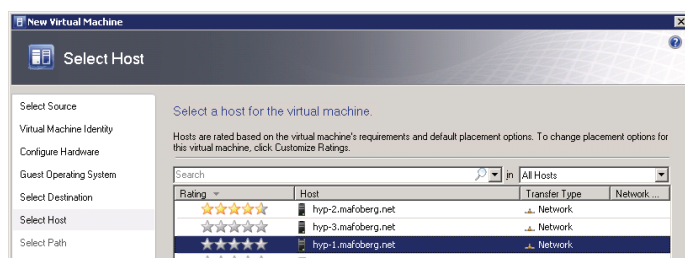
Figur 253 - Ny server fra SCVMM Library - Guest OS Profile

Og enkelt opprette innhold til operativsystem om vi ønsker dette med *Guest operating system profile*.



Figur 254 - Ny server fra SCVMM – hostplassering

Deretter må vi plassere den på en host som skal kjøre den.



Figur 255 - Ny server fra SCVMM – hostvalg

Vi velger å ta denne på en server i clusteret. Det bør sikres trafikkplass for å kopiere maskinen siden størrelsen på harddisken vil være rundt 7-8GB. Så velge plassering av maskindataene på hosten. Så

velger vi tilkoblingspunkt for nettverkskort. Til slutt velger vi hvordan maskinen skal oppføre seg, hvilken tilstand den skal ha. Så får vi en oppsummering før vi starter jobben.

For å kopiere data til hosten brukes Background Intelligence Transfer Service (BITS) over HTTP. Dette gjør at det i enkelte situasjoner over nettverk kan gå veldig tregt eller rett og slett stoppe opp om nettverkene har sperret for tilgang eller konfigurert for prioritering av trafikk av denne typen på lavt nivå.

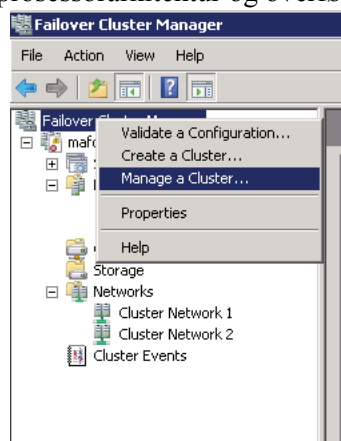
Vi valgte her å sette maskinen i clusteret hvor SCVMM so ligger på Hyp-1 har direkte tilgang fra hosten.

3.3. Bruk av Failover Clustering

I seksjon 2.4 -Windows Failover Clustering og Hyper-v Failover har vi allerede sett på hvordan vi validerer og setter opp et cluster, samt hvordan vi overfører virtuelle maskiner til et delt lagringsmedium på clusteret. Ved å bruke dette delte lagringsmediumet, har vi gjort maskinene highly available, det vil si at de kjørende virtuelle maskinene blir uavhengig av den fysiske maskinvaren for å kjøre. Dette vil si at en virtuell maskin kan gjenoppta kjøring, selv om den fysiske hostmaskinen(cluster node) skulle gå ned. Vi skal i denne seksjonen se på hvordan vi kan benytte clusteret til og manuelt flytte en virtuell maskin fra én node til en annen, samt hvordan vi kan simulere havari av én node.

En clusternode, det vil si en fysisk maskin med failover clustering og Hyper-V installert, kan kjøre flere virtuelle maskiner. Ikke alle disse virtuelle maskinene må benytte seg av clusteret. Fordelen med å bruke et lokalt lagringsmedium i stedet for clusterdisken, er at maskinen kan fortsette å kjøre i tilfelle forbindelsen mot clusterdisken går ned¹⁶. Ulempen blir at det ikke er mulig å flytte de virtuelle maskinene mellom to noder uten å først manuelt flytte diskene over til clusteret; Se kapittel 2.4.6 for instruksjoner som forteller hvordan en kan flytte over maskiner til clusterlagring.

Vi skal nå se på hvordan vi kan flytte en virtuell maskin som kjører på clusterlagring fra én node til en annen. En skulle tro at dette er en enkel prosess, men det er flere begrensninger pga overføring av prosessorarkitektur og overføring og synkronisering av minnet.



Figur 256 - Administrere et cluster

Vi skal benytte oss av programmet Failover Cluster Manager, som blir tilgjengelig på hver clusternode etter Failover Clustering er installert. Det er også mulig å installere et sett med administrasjonsverktøy fra Microsoft, slik at en kan administrere clusteret fra sin arbeidsstasjon¹⁷.

Til venstre ser vi at vi kan velge clusteret vi ønsker å administrere. Her kan en bruke DNS-navnet eller IP-adressen til clusteret, som oppgitt i seksjon 2.4.5 - Opprettelse av Clusteret

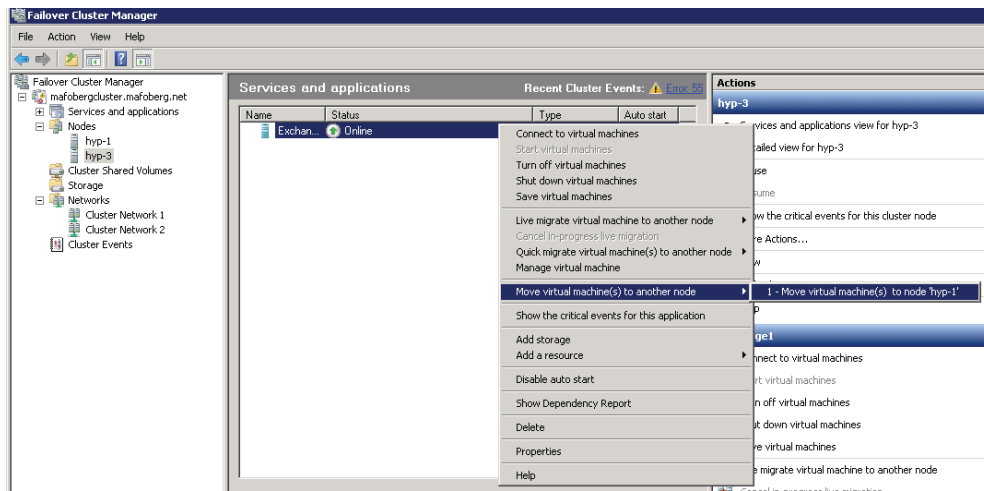
¹⁶ . I sammenheng med testing i dette prosjektet har clusterdisken faktisk gått ned én gang. Løsningen var å kontakte nettverksansvarlig som kunne konfigurere lagringsmediumet på nytt. Det var her snakk om at lagringsmediumet autovekslet mellom to prosessorer, hvorav den ene ikke fungerte med Failover Clustering

¹⁷ Se <http://www.microsoft.com/downloads/details.aspx?FamilyID=7d2f6ad7-656b-4313-a005-4e344e43997d&displaylang=en> eller <http://bit.ly/16kaQK> for nedlastingslenke for denne pakken

3.3.1. Flytte maskiner mellom noder

Failover Cluster Manager har flere metoder for å flytte maskiner mellom to noder. Alle med sine fordeler og ulemper. Kun den tradisjonelle flyttinga av maskiner ved hjelp av "Move Virtual Machine" støtter flytting mellom to noder med forskjellig prosessorarkitektur (Intel - AMD), de to andre samt automatisk failovering vil kun være tilgjengelig mellom to noder som kjører samme prosessorarkitektur, dvs Intel-Intel eller AMD-AMD.

3.3.1.1. Move Virtual Machine

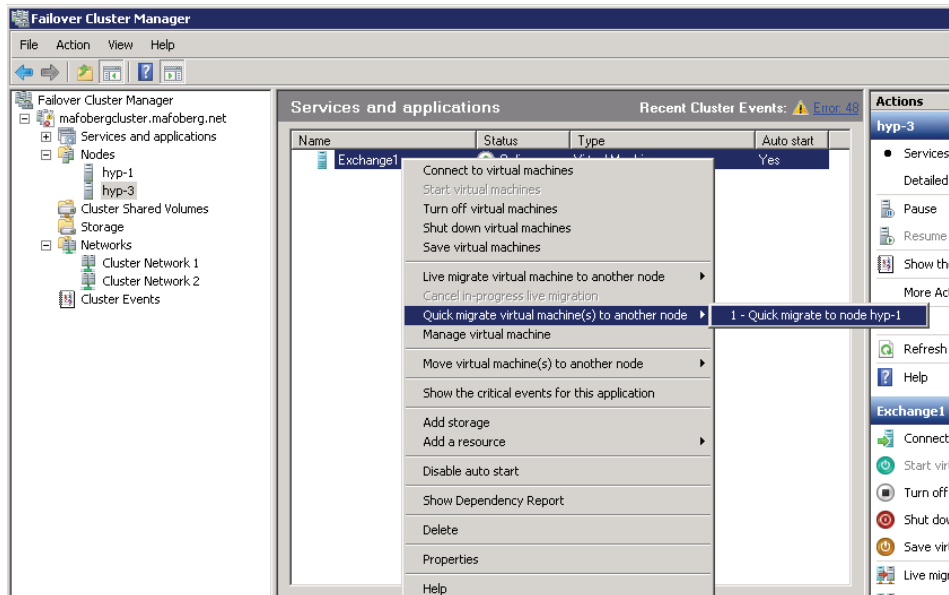


Figur 257 - Standardflytting av virtuelle maskiner

Figuren overfor viser hvordan vi i Failover Cluster Manager kan flytte en virtuell maskin mellom to noder. Her har vi bare noden "hyp-1" tilgjengelig; hadde vi hatt flere noder tilgjengelige på clusteret, ville vi hatt valget om å flytte den virtuelle maskina til disse nodene også. Denne metoden å flytte maskiner på er veldig kjapp. Den slår av den kjørende virtuelle maskina, gir den nye noden kontroll over diskbildet til VMen og deretter skruer den på igjen. Ulempen blir at kjørende tjenester må avsluttes før de flyttes, og dermed kan det skape avbrudd i arbeidsflyten til brukerne.

3.3.1.2. Quick Migrate

Nedenfor ser vi hvordan skjermbildet i Failover Cluster Manager ser ut når vi skal kjøre en Quick Migrate. En Quick Migrate vil lagre all minne til den kjørende virtuelle maskinen, for deretter å flytte over både tilgang til dette minnet og VMen til den nye noden.



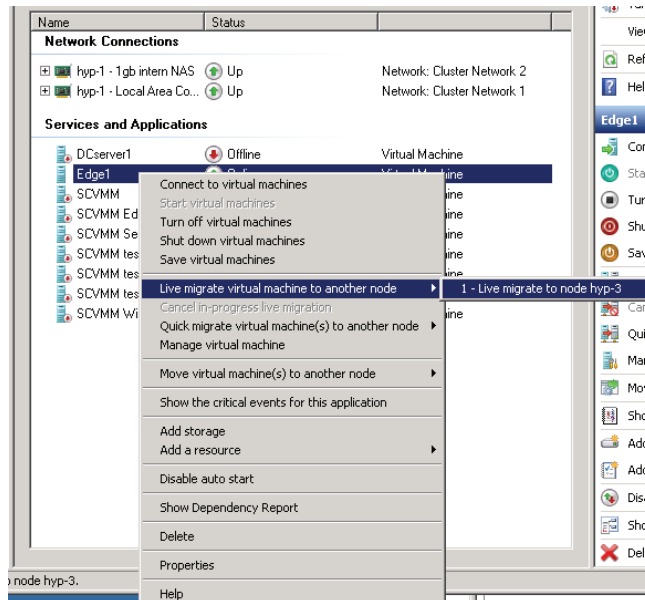
Figur 258 - En Quick migrate gjennomføres

Minnet til VMen vil deretter gjenopptas av den nye noden. Resultatet blir at brukeren kan gjenoppta bruk av tjenester på den virtuelle maskinen etter noe nedetid. Alle noder må kjøre på samme prosessorarkitektur for at minnet skal kunne flyttes over - og dermed for at Quick migration skal fungere.

En Quick migration kan feile dersom det ikke er nok ressurser til å starte den virtuelle maskinen på nytt på den nye eiernoden. Isåfall vil den virtuelle maskinen som standard flytte seg tilbake til sin tidligere eier før den gjenopptar kjøring.

3.3.1.3. Live Migration

Live migration er et nytt konsept siden Server 2008, og er en sterk forbedring av Quick Migration. Det er en direkte flytting av virtuelle maskiner mellom to noder uten at brukerne merker nedetid. I praksis vil det derimot bli et "lagg" - et lite øyeblikk (mindre enn ett sekund) der en bruker kan oppleve at tjenesten den virtuelle maskinen tilbyr henger.



Figur 259 En Live migration utføres

Denne metoden for migration fungerer ved å progressivt kopiere minnet til den kjørende virtuelle maskinen for så og overføre denne til den nye noden.

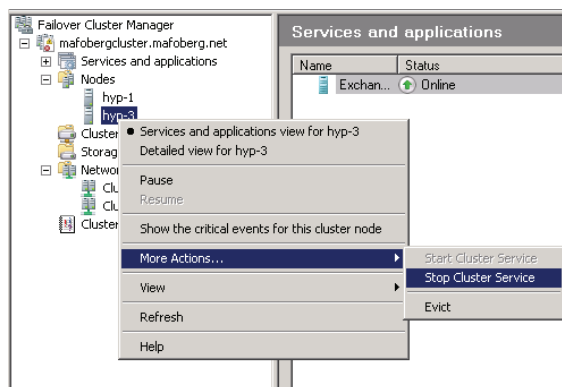
Dersom en live migration startes på en virtuell maskin med 1 GB RAM i bruk, vil denne rammen altså kopieres over til den nye eieren av VMen. Denne kopieringa vil ta noe tid, noe som resulterer i at ikke alt av den overførte RAMen skal brukes. Cluster Manager vil da se etter endringer i RAMen til den virtuelle maskinen og overføre kun disse endringene til den nye noden. Denne prosessen vil så fortsette helt til begge noder (nye og tidligere eier) har en synkronisert

RAM for den kjørende virtuelle maskinen. I dette tidspunktet vil den gamle eieren skrus av og den nye eieren blir instruert til å ta

over kjøring av maskinen med sin egen kopi av RAMen til den virtuelle maskinen. Denne prosessen kan ta opp til ett sekund å initialisere, og vil føre til et (veldig) lite lag for brukerne av tjenestene på den virtuelle maskinen.

3.3.1.4. Teste automatisk (unplanned) failovering

Å kunne flytte virtuelle maskiner fra én node til en annen manuelt er forklart overfor. Hvordan sørger vi så for at vi kan flytte maskiner automatisk til en ny node i clusteret når én av nodene uventet ramler ned? Dette kan skje ved hardwarefeil eller at forbindelsen mot cluster og nettverk ramler ut samtidig. En automatic failover vil da skje - gitt at de virtuelle maskinene ligger i clusteret og er highly available. Dette kalles automatic eller unplanned failover. Fordi lagringsmediumet er delt, vil en ny node i clusteret ta over umiddelbart. Derimot vil ikke RAMen til den virtuelle maskinen være tilgjengelig - slik at dataene som lå i RAM før failover skjedde må regnes som tapt.

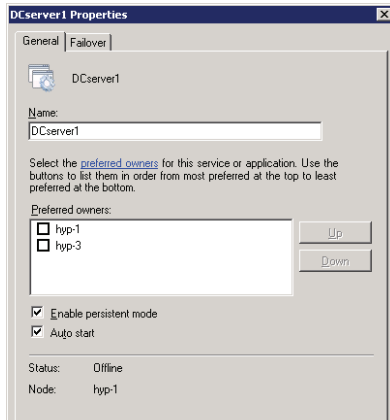


Figur 260 - Simulering av automatisk failovering

Det som nå skjer, er at noden vil opptre som ikke tilgjengelig for Failover Management. Alle virtuelle maskiner vil flyttes over til ledige noder og startes opp automatisk. Når (i vårt tilfelle) "hyp-3" blir tilgjengelig igjen, vil de virtuelle maskinene som feilet over til andre noder flyttes tilbake til "hyp-3" og startes der igjen.

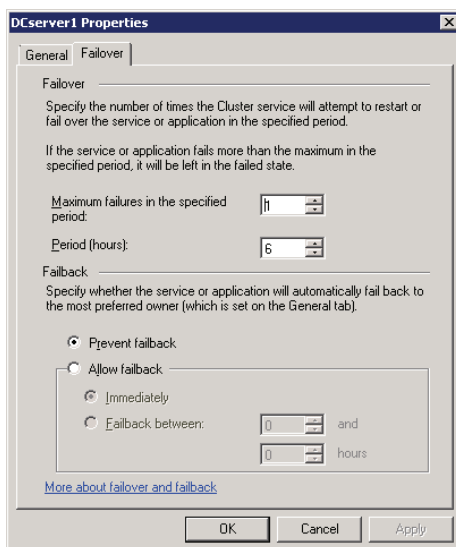
3.3.1.5. Detaljstyre failovering

Denne failoveringa skjer til de noder som har ledig kapasitet og når den noden som feilet kommer opp vil denne få tilbake kontroll over sine tidligere virtuelle maskiner. Dette er ofte - men ikke alltid ønskelig. I Failover Cluster Manager kan en derfor velge de virtuelle maskinene og endre deres egenskaper. Det er viktig å påpeke at disse egenskapene bare er tilgjengelig i Failover Cluster Manager og ikke i Hyper-V-konsollet.



Figur 261 - Failoveregenskaper for VMen DCserver1

En kan her kontrollere innstillinger direkte relatert til Failovering. I fanen "General" kan vi velge hvilke noder vi ønsker den virtuelle maskina skal kunne failovere til. "Enable persistent mode" gjør at den virtuelle maskina bestandig vil prøve å starte på samme eiernode, mens "auto start" velger om den også skal starte opp etter en failovering.



Figur 262 - Failoveregenskaper 2 for DCserver1

Fanen "Failover" detaljstyrer hvordan en failovering skal skje. Vi ser at det satt som standard slik at en virtuell maskin kan feile maks én gang per 6. time før den blir liggende som feilet. Merk forskjellen mellom failover og failback. Ved feilovering vil den flytte seg tilbake til en annen node med ledig kapasitet. Failback spesifiserer hva som skal skje når en feilet node så blir tilgjengelig.

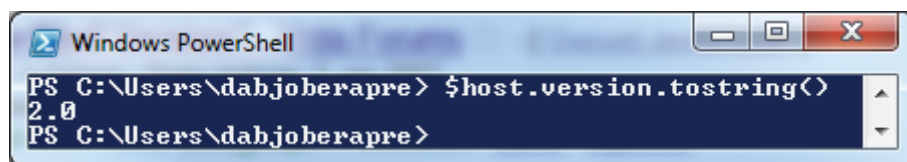
4. Script i Powershell

Powershell er et relativt nytt scriptingspråk introdusert i 2006 av Microsoft. Det var hovedsakelig designet som en smartere erstatning til de forskjellige UNIX-miljøenes kommandolinjer - hvor du har full tilgang til å endre de fleste systemkonfigurasjoner, dvs. både installere tjenester og konfigurere de, samt utføre kalkulasjoner fra et tekstlig grensesnitt. Fra og med Powershell v2.0 som følger med som standard i Windows Server 2008r2 og Windows 7, er også støtte for såkalt Powershell Remoting inkludert. Denne tjenesten sørger for at én eller flere maskiner kan aksessereres og konfigureres fra én sentralisert plassering; konfigurasjonsinnstillinger kan også sendes ut til flere maskiner samtidig. Powershell Remoting benytter en webbasert protokoll for denne kommunikasjonen. Dette gjør at også andre aktører enn Microsoft kan tilby kompatible løsninger, slik at for eksempel en Linux-basert VMWare ESX-server enkelt kan konfigureres fra en ekstern Windows Powershell-sesjon. Powershell er en sentral del av Microsofts nye produktserie med Windows Server Core, hvor all grafisk administrasjon er fjernet til fordel for administrasjon gjennom henholdsvis Powershell og grafiske grensesnitt som kjører på klientmaskiner i stedet for server, og som benytter Powershell Remoting for å kontrollere Core Servere.

4.1 Installasjon av Powershell

Det er viktig å påpeke at alle maskiner som skal kommunisere ved hjelp av Powershell, må ha samme versjonsnummer av Powershell installert. Dersom vi antar at alle parter har Windows Server 2008r2 eller Windows 7 installert, trengs det ingen videre installasjon. For Windows XP som mangler Powershell, og Windows Vista samt Server 2008, kan Powershell 2.0 lastes ned gratis fra Microsoft sine sider: <http://support.microsoft.com/kb/968929> . I dette prosjektet kjører alle servere Microsoft Server 2008r2, og vi vil derfor ikke gå videre innpå installasjon på eldre operativsystemer. Lurer du likevel på hvilken Powershellversjon du kjører, kan dette vises ved å skrive

```
$host.version.toString()
```



Figur 263 - Utlisting av Powershell-versjon

4.2 Komme i gang med Powershell

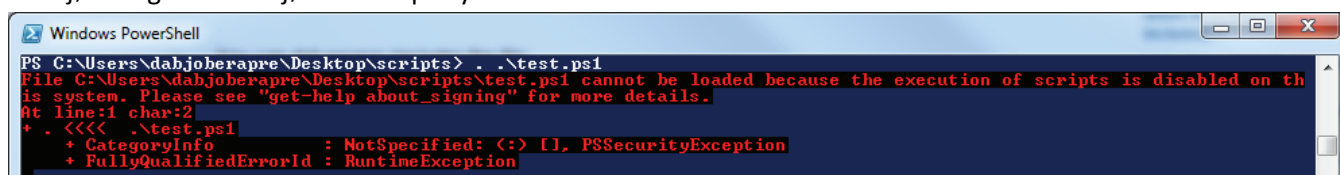
Når Powershell er installert og tilgjengelig, finnes den fra startmenyen og deretter et søk på Powershell. Enkelte kommandoer vil kreve administrasjonsrettigheter for å kunne utføres. Selv om det er mulig å bruke Powershell-kode for å endre rettighetsnivået for en kommando som krever det, er det ofte enklest å starte et nytt Powershell-skall direkte med administrasjonsrettigheter; Høyreklikk programmet og velg "Kjør som administrator".

Videre i dette dokumentet antas det at leseren har noe erfaring med kommandolinja eller programmering fra før av, da vi bare kort skal gå igjennom oppsett og Powershell-syntaks for deretter å gå mer i dybden på administrasjon av servere ved hjelp av Powershell.

4.2.1 Execution Policy

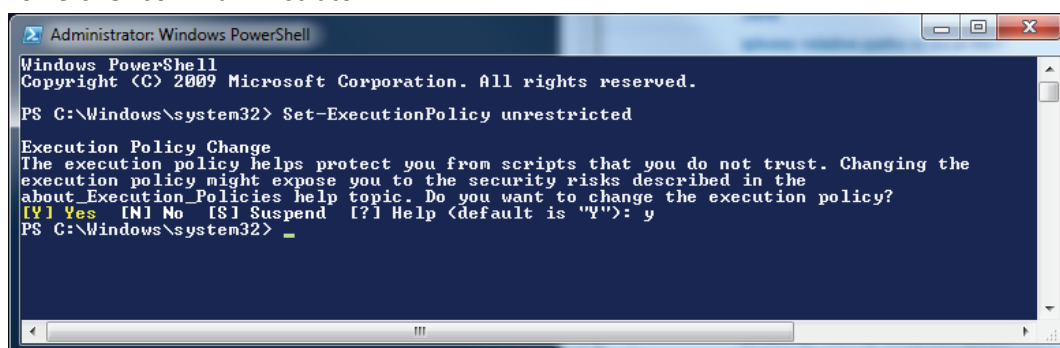
Vi skal nå benytte et Powershell administrasjonskonsoll for å aktivere en execution policy uten restriksjoner. En Execution Policy gjelder for alle framtidige Powershell-sesjoner for brukeren, og styrer om en kan importere og bruke kode fra andre Powershell-script. Under ser vi et eksempel på

at vi i en kjørende sesjon prøver å inkludere et script kalt "test" ved hjelp av såkalt dot-sourcing¹⁸. Innholdet i dette testscriptet spiller ingen rolle, for Powershell vil uansett ikke tillate oss å kjøre det. Dette har med sikkerhetsregler å gjøre, for å hindre at nedlastede script fra for eksempel Internett kan kjøres og dermed kjøre skade på systemet.



Figur 264 - Feilmelding ved eksekvering av script

Når vi likevel har kontroll på hvor våre script kommer fra, er det mulig å lempe litt på disse restriksjonene ved å kjøre "Set-ExecutionPolicy unrestricted". Dersom kommandoen feiler med beskjed om "ingen tilgang til registernøkkelen" eller tilsvarende, dobbeltsjekk at du kjører Powershell som Administrator.



Figur 265 - Endre Execution Policy

Det er også mulig å sette Execution Policy til andre verdier som "allsigned" og "remotesigned". Disse sørger for at henholdsvis alle script må være signert eller at bare script som kommer fra eksterne kilder må være signert. Signering er en form for å sikre autensiteten til den som har lagd scriptet, slik at en minsker risikoen for å kjøre vilkårlig skadelig kode. Vi tar ikke opp signering i mer detalj her, men det lønner seg altså i miljøer hvor du inkluderer mye kode fra tredjeparter. En må også merke seg at signering ikke garanterer for at et script er fri for skadelig kode, det garanterer bare at scriptet er lagd av en person du stoler på.

¹⁸ Dot-sourcing er en teknikk lånt fra linuxskallet "sh" og andre, hvor en ved å skrive et punktum og deretter stien til ønsket fil får inkludert koden i den fila som en del av nåværende kjørende prosess.

4.2.2 Powershell-profiler

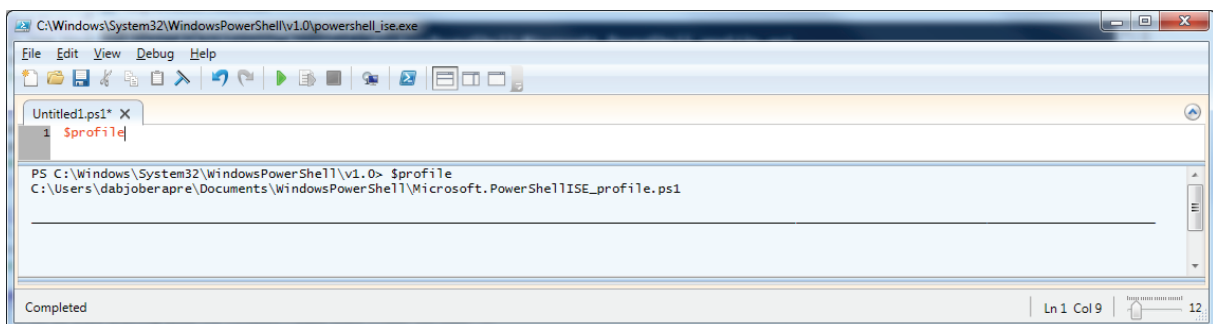
Når vi nå har gått igjennom grunnleggende oppsett, er det på tide å benytte oss av mulighetene i Powershell til å lage oss en oppstartsprofil. Oppstartsprofiler er vanlige Powershellscript som autokjøres ved oppstart av en Powershellsesjon. Typisk kan dette brukes til å vise innloggingsinformasjon, klokkeslett og annen nyttig informasjon som er ønskelig å vite.

Alle som har Powershell har en profilsti, men den kan ofte peke til en ikke-eksisterende sti. Variabelen `$profile` vil opprettes automatisk av Powershell, og peke til denne stien.



Figur 266 - Standard profilsti

Merk også at det er mulig å ha flere profiler, men at denne er standard for Powershell. Bruker du Powershell-editoren "Powershell ISE" som følger med Windows og Powershell v.2, vil denne bruke en annen profilsti, her "*Microsoft.PowerShellISE_profile.ps1*" i stedet for "*Microsoft.PowerShell_profile.ps1*"



Figur 267 - Profilsti kan forandres av andre program

En bruker som regel bare ISE for å redigere og debugge scripts, så vi kommer ikke til å opprette en egen profil for ISE.

4.2.2.1 Opprette Powershell-profiler

Nå er det tid for å opprette en Powershell-profil. Under mappa "My Documents"(Windows XP) eller "Documents" (Windows 7) Sørg for at det eksisterer ei undermappe kalt WindowsPowershell; lag den om den ikke eksisterer. Deretter opprett fila "*Microsoft.PowerShellISE_profile.ps1*" i denne mappa og rediger den. Legg til følgende kode:

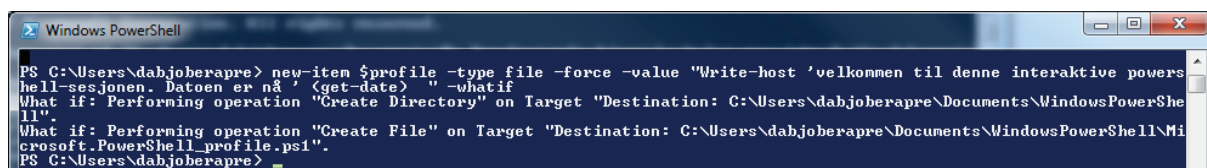
Write-host 'velkommen til denne interaktive Powershell-sesjonen. Datoen er nå ' (get-date)

Lagre så fila og åpne et nytt Powershell-vindu.

Alternativt kan du la Powershell opprette profilen for deg. Koden for det er følgende:

```
new-item $profile -type file -force -value "Write-host 'velkommen til denne  
interaktive Powershell-sesjonen. Datoen er nå ' (get-date) "
```

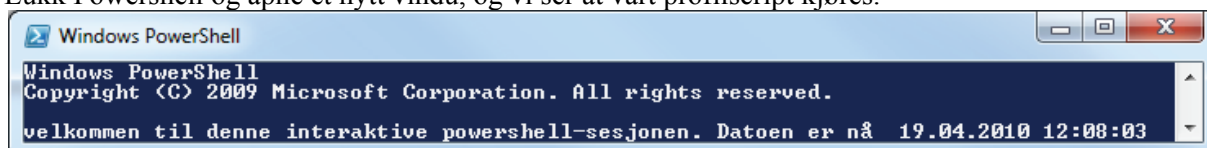
Kommandoen overfor er delt opp over to linjer, men må skrives på éi linje i Powershell. Parameteret `“-force”` sørger for at Powershell oppretter mappa WindowsPowerShell dersom den ikke eksisterer. Ved å kjøre samme kommando, men med et parameter `“-whatif”`, kan vi se hva Powershell kommer til å gjøre, uten at det faktisk gjennomføres.



Figur 268 - Eksempel på `-whatif`

Her ser vi tydelig at Powershell ville både opprette undermappa og legge profilen vår under denne mappa. Hadde vi unnlatt å bruke `“-force”`, ville Powershell gitt feilmelding om at mappa ikke eksisterer.

Lukk Powershell og åpne et nytt vindu, og vi ser at vårt profilscript kjøres.



Figur 269 - Profilscript autokjørt ved oppstart

Vi skal seinere se på Powershell Remoting, en funksjon som lar deg kontrollere andre Windows-maskiner (fortrinnsvis servere), både manuelt og automatisert. Ved manuell innlogging, kan det derfor være meget nyttig å vite hvilken bruker du er innlogget som, og hvilken maskin du er logget inn på. Dette bidrar sterkt til å redusere risikoen for at du utfører kommandoer på feil server.

Følgende kommando har vi derfor valgt å lagre som standard i våre Powershellprofiler, slik at denne informasjonen vises hver gang:

```
write-host $env:username"@">$env:computername"."$env:userdnsdomain
```

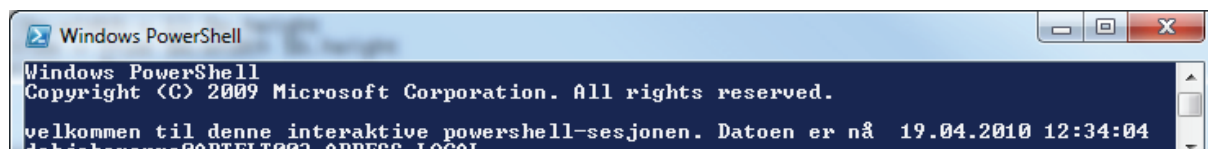
Miljøvariabler	Eksempelverdier
Allusersprofile	C:\Programdata
Computername	DCSERV1
Homedrive	C:
OS	Windows_NT
Processor_architecture	AMD64
UserDNSDomain	Mafoberg.net
Username	bjorninge

Variabelen `$env` opprettes av Powershell automatisk og inneholder miljødata og informasjon om brukeren og datamaskinen som er i bruk; slik som hvilket operativsystem som brukes, hjemmemapper og informasjon om brukerens datamaskininnstillinger.

Tabellen på venstre side inneholder eksempeldata på noen av de tilgjengelige verdiene en kan inspisere. `$env:OS` vil da i vårt tilfelle referere til teksten `"Windows_NT"`. En full utlisting kan hentes ut med kommandoen `"ls env:"`

Etter kommandoen overfor er lagt til og et nytt Powershell-vindu er åpna, vil informasjonen dukke opp ved oppstart:

Tabell 10 - Miljøvariabler



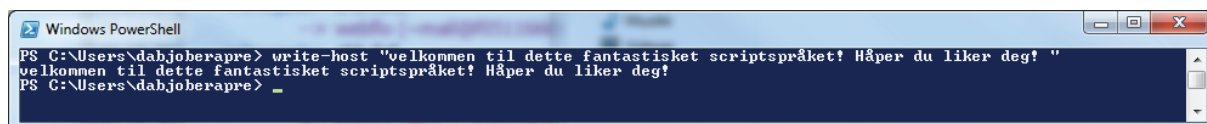
Figur 270 - Innlegging av innloggingsinformasjon i profilen

Brukernavnet finner vi før alfakrøllen, fulgt av maskinnavn og domenenavn.

4.3 syntaks og sammenlining

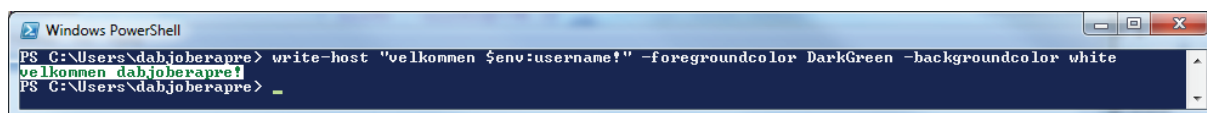
Powershell er bygd opp av en hel del kommandoer og kompabilitetslag for å gjøre hverdagen til systemadministratorer som kommer fra både DOS og UNIX-verdenen enklere. En systemkommando i Powershell har bestandig formen *verb-substantiv*, men kan ha en rekke alias.

I Figur 271 er det et eksempel på enkel bruk av Powershell, hvor vi benytter kommandoen *write-host*, som gjentar det som står som *argument*, dvs. teksten etter kommandonavnet, og skriver det ut til skjermen.



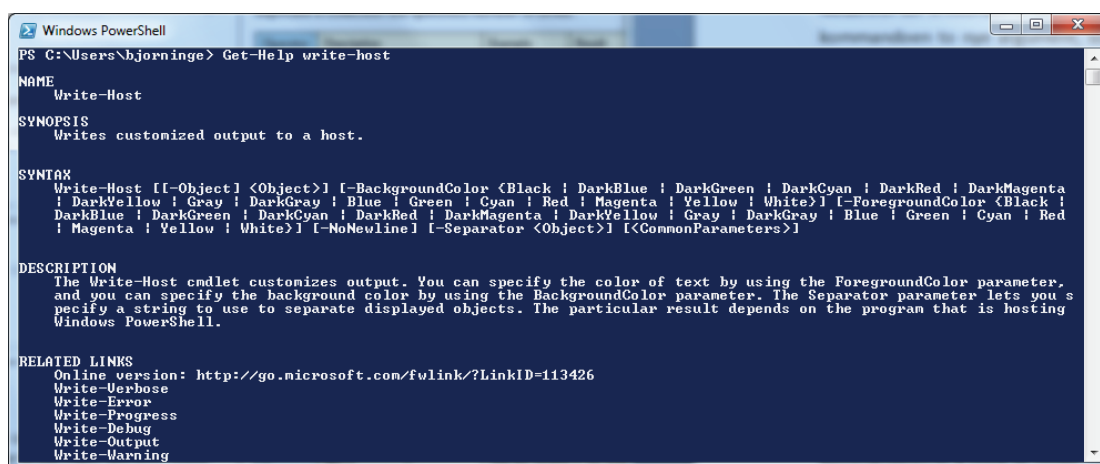
Figur 271 - Enkel utskrift med Powershell

En kommando kan ta alt fra 0 til et uendelig antall argument. Ofte kan vi bare skrive inn verdiene vi ønsker, slik som overfor, andre ganger må vi navngi argumentene for deretter å gi de en verdi. Nedenfor ser vi hvordan vi bedre kan benytte mulighetene til kommandoen *"write-host"*. Her har vi gitt kommandoen to nye argument, som henholdsvis spesifiserer tekstfarge og bakgrunnsfarge. Vi ser også et eksempel på bruk av innebygde variabler i Powershell, hvor vi henter ut brukerens brukernavn.



Figur 272 - fargelegging av uttekst

Vi kan alltid slå opp for å finne ut hvilke argumenter en kommando tar. Antar vi at vi lurer på hvordan kommandoen *"write-host"* er bygd opp, kan vi kjøre *"get-help write-host"* for å se tilgjengelige argument samt hvilke farger disse argumentene kan ta. Se Figur 273



Figur 273 - Eksempel på Get-Help

Oversikten viser også relaterte lenker, slik at du kan lære mer, også fra webben.

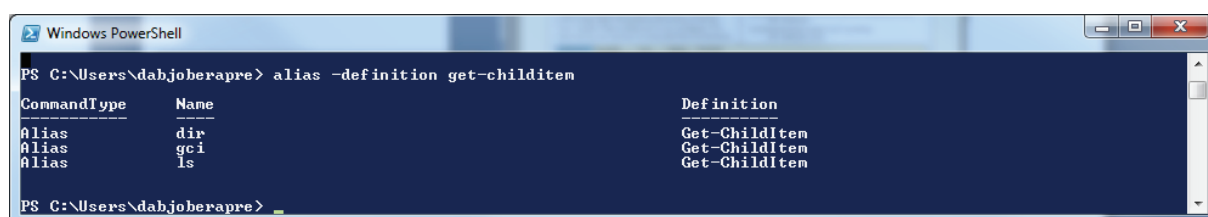
4.3.1 Get-ChildItem – en kommando med mange navn

En annen kommando som du kommer til å bruke ofte, er kommandoen for å liste ut filer i stående mappe. Til dette kan vi bruke kommandoen *"Get-ChildItem"* – som også kan liste ut informasjon om registernøkler eller miljøvariabler – i tillegg til filer i ei mappe. Da primærbruken til denne

kommandoen er å finne ut hvilke filer som eksisterer i nåværende mappe, har Microsoft valgt å legge til flere alias til denne kommandoen som kjent fra behandling av filer i tidligere kommandoskall.

Både "ls" (kjent fra "sh"-skall i linux og unix) og "dir" (dos) gjør nøyaktig det samme som "Get-ChildItem". Kommandoer i Powershell (til motsetning fra sh-skall) er case-insensitive, dvs at store og små bokstaver har samme betydning. "get-childitem" og "LS" er altså like gyldige kommandoer som "Get-ChildItem" og "ls".

Kommandoen "alias -definition enkommando" viser alle alias til enkommando som vist i Figur 274.



Figur 274 - Get-ChildItem kan skrives på hele tre forskjellige måter

Wikipedia¹⁹ har en oversikt over en del kommandoer og deres aliaser. Nedenfor er ei bearbeidet liste over de mest brukte kommandoene, og deres alias:

Powershellkommando	DOS-alias	"SH"-alias	Beskrivelse
Get-location	<i>pwd</i>	<i>pwd</i>	Lister ut stående mappe
Set-location	<i>cd</i>	<i>cd</i>	Setter ny stående mappe
Remove-item	<i>del</i>	<i>rm</i>	Fjerner mappe eller fil
Copy-item	<i>copy</i>	<i>cp</i>	Kopierer filer
Move-item	<i>move</i>	<i>mv</i>	Flytter mappe eller fil
Set-location	<i>cd</i>	<i>cd</i>	Setter ny stående mappe
New-Item	<i>mkdir</i>	<i>mkdir</i>	Lager ei ny mappe eller fil

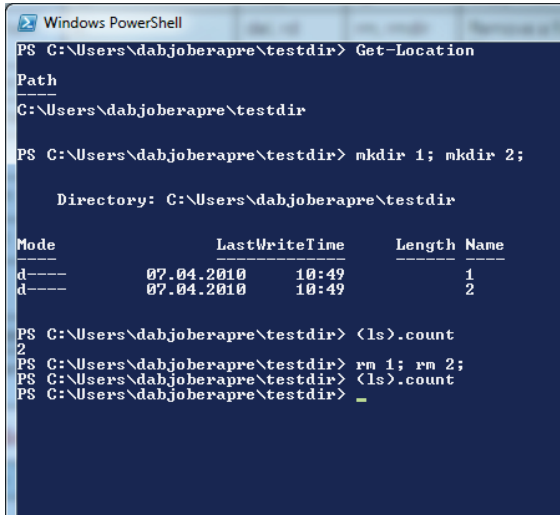
Tabell 11 - Mye brukte kommandoer og alias

¹⁹ Den originale lista kan finnes på

http://en.wikipedia.org/wiki/Powershell#Comparison_of_cmdlets_with_similar_commands

Merk altså at en kan bruke aliasene til en kommando om hverandre; de har nøyaktig samme betydning for Powershell.

En kommando skrives oftest på en egen linje, men det er også mulig å skrive flere kommandoer etter hverandre separert med semikolon.



```
PS C:\Users\dabjoherapre\testdir> Get-Location
Path
C:\Users\dabjoherapre\testdir

PS C:\Users\dabjoherapre\testdir> mkdir 1; mkdir 2;

    Directory: C:\Users\dabjoherapre\testdir

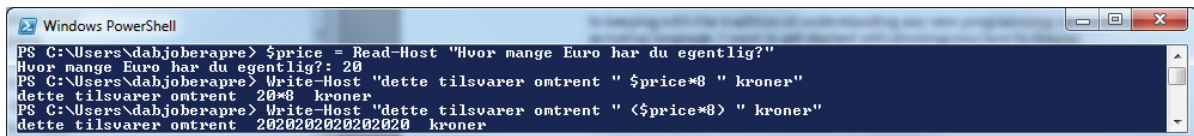
Mode                LastWriteTime         Length Name
----                -
d-----          07.04.2010    10:49             1
d-----          07.04.2010    10:49             2

PS C:\Users\dabjoherapre\testdir> (ls).count
2
PS C:\Users\dabjoherapre\testdir> rm 1; rm 2;
PS C:\Users\dabjoherapre\testdir> (ls).count
0
PS C:\Users\dabjoherapre\testdir>
```

Figur 275 - Opprettelse og utlistering av mapper

Her ser vi et Powershellskall som viser flere av kommandoene overfor. Vi ser at vi henter ut nåværende mappe, legger til to undermapper og bekrefter antallet filer og undermapper med "(ls).count", før vi sletter undermappene igjen. En kan for øvrig se at ls har en .count-egenskap ved å kjøre "get-member -in (ls)".

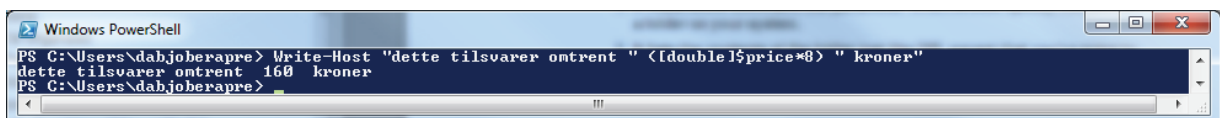
Ofte ønsker vi ikke bare å skrive ut verdier til skjerm, men også spørre brukeren om å taste inn verdier. Kommandoen for å gjøre dette, heter "Read-Host". Nedenfor har vi forsøkt å spørre brukeren om en mengde Euro, som vi så har forsøkt å konvertere til norske kroner.



```
PS C:\Users\dabjoherapre> $price = Read-Host "Hvor mange Euro har du egentlig?"
Hvor mange Euro har du egentlig?: 20
PS C:\Users\dabjoherapre> Write-Host "dette tilsvarer omtrent " $price*8 " kroner"
dette tilsvarer omtrent 20*8 kroner
PS C:\Users\dabjoherapre> Write-Host "dette tilsvarer omtrent " <$price*8> " kroner"
dette tilsvarer omtrent 2020202020202020 kroner
```

Figur 276 - Gangeoperasjon på tall kan gi tilsynelatende feil resultat

Kommandoen ser ut til å fungere, men hvorfor får vi dette store tallet? $20 \cdot 8$ er jo 160, ikke "2020202020202020". Det er nemlig slik at Powershell tolker all input som tekst, og vil derfor gjenta den innskrevne teksten (som er "20") 8 ganger, i stedet for å gange tallverdien 20 med 8. Måten å løse dette på er å tvinge Powershell til å tolke prisen som enten heltall eller desimaltall ved å bruke ordene *double* eller *int* i klammeparenteser. Her skifter vi altså datatypen fra string til double:



```
PS C:\Users\dabjoherapre> Write-Host "dette tilsvarer omtrent " <[double]>$price*8 " kroner"
dette tilsvarer omtrent 160 kroner
PS C:\Users\dabjoherapre>
```

Figur 277 - Tvinge fram datatype

4.3.2 Funksjoner, kontrollstrukturer og operatører

Ofte bruker vi Powershell for å gjøre enkle ting, slik som overfor hvor vi opprettet noen mapper. Likevel, det kan være ønskelig å gjøre noe mer avansert som krever at vi sjekker verdier opp mot fastsatte kriterier. Powershell støtter de fleste kontrollstrukturer kjent fra andre språk, slik som java. Funksjoner kan også lages for å samle kodeblokker som hører sammen. Selv om mye altså vil være kjent, er syntaksen noe forskjellig fra andre programmeringsspråk, hvor bindestrek oftest ikke er tillat i funksjonsnavn.

```
1 function get-available-updates(){
2     $updater = new-object -com "Microsoft.Update.Session"
3     $searcher = $updater.CreateUpdateSearcher()
4     return $searcher.Search("ISInstalled=0 and Type='software'").updates
5
6 }
7
8
9 $numberOfupdates = (get-available-updates).count
10
11 if ($numberOfupdates -ge 3) {
12     write-host "Du har mange oppdateringer som venter på å bli installert!"
13 } elseif ($numberOfupdates -eq 0) {
14     write-host "Du har et fullt oppdatert system!"
15 } else {
16     write-host "Du mangler å installere 1 eller 2 oppdateringer"
17 }
```

Figur 278 - Bruk av if-elseif-else

Overfor ser vi et utdrag av en kode for å finne tilgjengelige oppdateringer på Windows Update. Vi ser også flere konspteer som gjøres annerledes i Powershell i forhold til andre scriptingspråk:

Linje 1 til 7 definerer en funksjon som har til formål å returnere ei liste over tilgjengelige oppdateringer. Funksjoner i Powershell begynner ofte med "get" eller "set" fulgt av en bindestrek og deretter hva som skal skje. Dette er likevel bare en konvensjon (eller sedvane) – men den anbefales å følge. Funksjonen brukere videre "new-object" kommandoen som lager et COM-object²⁰ for å få tilgang til Windows Update Services. Vi skal gå inn i detaljer på hvordan COM-objekter kan brukes til å administrere Windows Servere seinere.

På linje 9 kalles så funksjonen vår og egenskapen "count" hentes ut fra samlinga av tilgjengelige oppdateringer. Dette tilegnes så en variabel. Funksjoner i Powershell kalles som regel direkte uten at en bruker parenteser

På linjene 11 til 17 sjekker vi så antallet oppdateringer. Har du 3 eller flere oppdateringer vil du få beskjed om at det er på tide å oppdatere systemet. Dersom det er 0 tilgjengelige oppdateringer, er systemet full oppdatert, og ellers vil det være 1 eller 2 tilgjengelig. Sammenlikningsoperatørene i Powershell er noe som de fleste med bakgrunn fra for eksempel Java vil ta feil av første gang. For mens du bruker ">=" for å signalisere en sjekk etter større-enn i Java, bruker Powershell en bindestrek fulgt av en forkortelse. På neste side viser vi en tabell som viser hvordan du vil utføre sammenlikninger av to verdier i powershell. Vi viser også hvordan dette gjøres i to andre språk, nemlig Visual Basic og Java.

²⁰ COM-objekter er Microsofts modell for å la softwarekomponenter snakke sammen, også med system

Powershell	eksempel	Java-syntaks	Vb-syntaks	Beskrivelse
-eq	5 -eq 5	==	=	Likhet
-ne	4 -ne 4	!=	<>	ulikhet
-gt	4 -gt 5	>	>	Større enn
-gte	5 -gte 5	>=	>=	Større enn eller lik
-lte	4 -lte 3	<=	<=	Mindre enn eller lik
-contains	2,3,4 -contains 2	<code>.indexOf()</code>	<code>Array.indexOf()</code>	Inneholder
-notcontains	2,4,5 -notcontains 3	<code>.indexOf()</code>	<code>Array.indexOf()</code>	Inneholder ikke
-not	-not \$true	!	<i>Not</i>	Snur en verdi
-and	2 -eq 2 -and 3 -eq 3	&&	<i>And</i>	Kombinerer verdier
-or	2 -eq 2 -or 3 -eq 4		<i>Or</i>	Kombinerer verdier

Tabell 12 - Sammenlikning av operatører

Det er likevel sjelden vi ønsker å sammenlikne slike verdier direkte med hverandre. Oftest vil minst én av verdiene vi sammenlikner med, komme fra en variabel. Dette så vi et eksempel på i Figur 278, linjene 11 og 13 hvor vi brukte antallet tilgjengelige oppdateringer på venstre side og sammenliknet disse med fastsatte verdier på høyre side av uttrykket: `$numberOfUpdates -ge 3`

Det er også mulig å kombinere flere uttrykk for å gjøre en mer kompleks logisk sammenlikning, slik som sett i de tre nederste eksemplene i tabellen overfor. Gitt at kommandoen `get-date -format %M` returnerer dagens månedsnummer kan vi sjekke om en dato faller mellom to måneder ved å kjøre følgende kommando

```
$month = get-date -format %M
```

```
$month -ge 3 -and $month -le 10
```

Gitt at denne kommandoen kjøres i April (\$month blir da "4"), blir resultatet sant. "-or" kan tilsvarende brukes for å sjekke om et uttrykk matcher én eller begge av sammenlikningene, men ikke nødvendigvis begge:

```
$month -ge 3 -or $month -ge 50
```

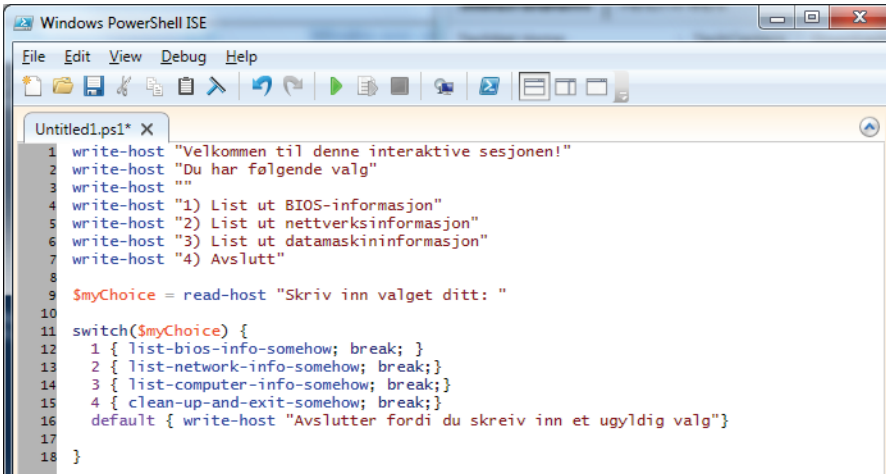
Dette uttrykket blir da sant, da måneden er større enn 3. Måneden kan samtidig ikke være større enn 50 (det er bare 12 måneder i året), men i motsetning til ”-and” krever operatøren ”-or” at bare ett av uttrykkene er sanne.

4.3.2.1 Switch som alternativ kontrollstruktur

I et program kan det tenkes vi ønsker å gjøre ting noe forskjellig basert på en inputvariabel. Anta for eksempel at vi skal finne informasjon om en gitt server, basert på hvilket nummer en bruker skriver inni kommandolinja. I slike tilfeller ville vi ofte gjort dette kjapt ved å bruke en serie med if-elseif-else-strukturer slik som i eksempelet med tilgjengelige serveroppdateringer overfor. Når valgene blir mange, blir det derimot litt tungvint å benytte if-elseif-else bl.a. for det det blir uoversiktlig, og en må gjenta hva som skal sjekkes opp i hver if-elseif (se eksempelet tidligere hvor vi måtte gjenta

\$numberOfUpdates i både if-testen og elseif-testen).

Her ber vi bruker taste inn et valg, og utfører noen fiktive kommandoer basert på hva som skrives inn. På linje 11 legger vi inn \$myChoice, og trenger ikke gjenta denne variabelen noe mer etter dette; de sjekkene som følger, sammenlikner seg opp mot \$myChoice



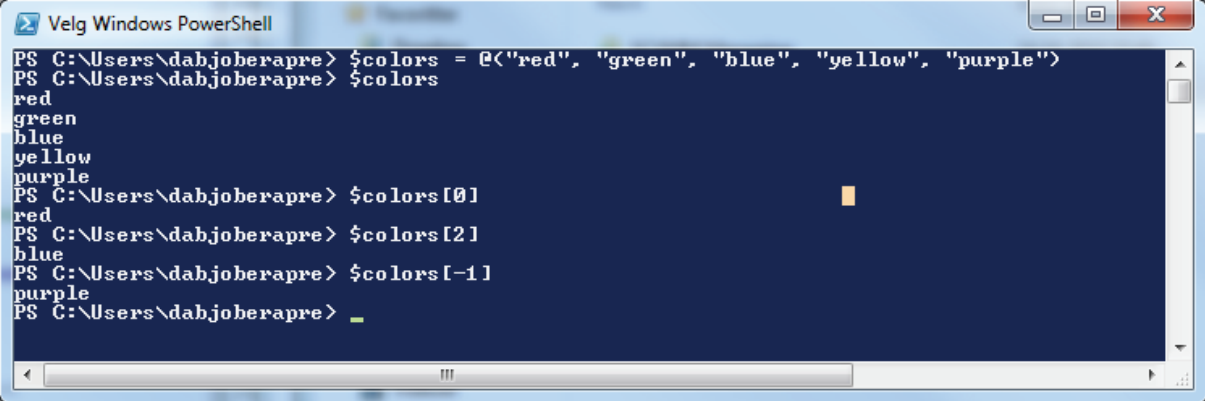
```
1 write-host "Velkommen til denne interaktive sesjonen!"
2 write-host "Du har følgende valg"
3 write-host ""
4 write-host "1) List ut BIOS-informasjon"
5 write-host "2) List ut nettverksinformasjon"
6 write-host "3) List ut datamaskininformasjon"
7 write-host "4) Avslutt"
8
9 $myChoice = read-host "Skriv inn valget ditt: "
10
11 switch($myChoice) {
12     1 { list-bios-info-somewhat; break; }
13     2 { list-network-info-somewhat; break; }
14     3 { list-computer-info-somewhat; break; }
15     4 { clean-up-and-exit-somewhat; break; }
16     default { write-host "Avslutter fordi du skreiv inn et ugyldig valg" }
17 }
18 }
```

Figur 279 - Switch når en skal gjøre potensielt mange sjekker mot samme variabel

automatisk. ”Default” er en spesiell konstruksjon, som utføres dersom ingen av de foregående sjekkene er sanne. Dette vil skje dersom det skrives inn bokstaver i stedet for tall, eller et for høyt tall skrives inn. Programmet avsluttes i dette tilfellet.

4.3.2.2 Arrays og hashtables

Arrays i Powershell fungerer som en samling av enkeltverdier, som vi så kan indeksere, og manipulere. En bruker kan for eksempel ønske å holde ett sett med farger i, for deretter å hente ut eller søke etter enkelte av disse. Dette brukes ofte i forbindelse med løkker, slik som vi skal se lengre nede. Arrays lages ved å skrive en alfakrøll etterfulgt av parenteser. Hvert element i et array separeres med et komma.

A screenshot of a Windows PowerShell window titled "Velg Windows PowerShell". The window has a dark blue background. The command prompt shows the following sequence of commands and outputs:

```
PS C:\Users\dabjoberapre> $colors = @("red", "green", "blue", "yellow", "purple")
PS C:\Users\dabjoberapre> $colors
red
green
blue
yellow
purple
PS C:\Users\dabjoberapre> $colors[0]
red
PS C:\Users\dabjoberapre> $colors[2]
blue
PS C:\Users\dabjoberapre> $colors[-1]
purple
PS C:\Users\dabjoberapre> _
```

Figur 280 - Enkel bruk av arrays

Her ser vi at vi lagrer en del farger til variabelen `$colors`. Ved å skrive variabelnavnet på en egen linje, listes alle variabler ut. Første farge hentes ut ved å referere til `$colors[0]` og videre oppover. Motsatt kan vi hente siste element, dvs fargen "purple" ved å referere til `$colors[-1]`.

Hashtabeller brukes nærmest likt som arrays. Forskjellen er at hvert element er assosiert med en verdi. Har du for eksempel en liste over byer, vil du ofte også ha deres postkoder også. Et eksempel på dette følger under.

```
$codes = @{"Trondheim" = "7000"; "Vikhammer" = "7560"}
```

Slike tabeller startes med en alfakrøll og hakeparenteser. Hvert nøkkel/verdi-par separeres så med et semikolon. Vi kan videre sortere eller hente ut spesifikke elementer fra både hashtables og arrays, se seksjonen 4.3.2.4 Pipeline,

4.3.2.5 Where-Object Filtre, 4.3.2.6 Sort-object og 4.3.2.6 Begrense resultatmengde med Select-object

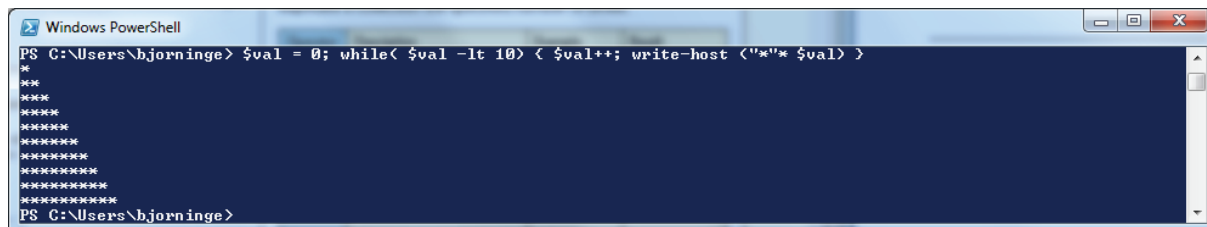
\$codes-variabelen vår, kan vi deretter legge til å slette elementer ved å bruke metoden ".remove". Flere mulige handlinger som er mulig, kan du finne ved å kjøre

```
Get-Member -in ($codes)
```

4.3.2.3 Løkker

Løkker i Powershell er metoder for å gjenta utførelse av en gitt programkode. Det er flere typer løkker som hovedsakelig gjør det samme, men på litt forskjellige måter

While kjører ei kodeblokk så lenge kondisjonen dens er sann. Argumentet i koden i Figur 281 er `"$val -lt 10"`.

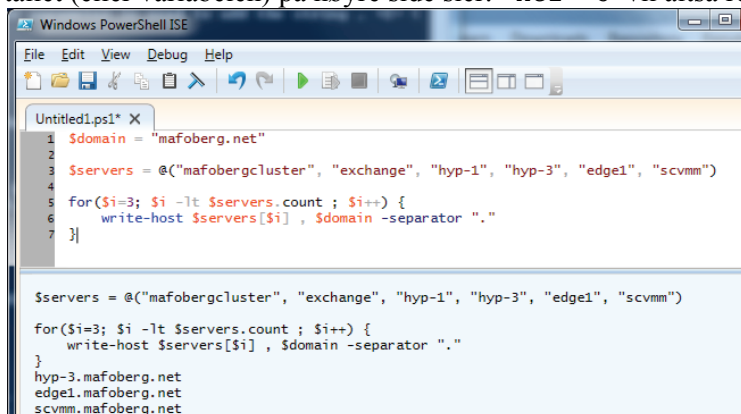


```

PS C:\Users\bjorninge> $val = 0; while( $val -lt 10) { $val++; write-host ("*" $val) }
*
*
*
*
*
*
*
*
*
*
PS C:\Users\bjorninge>
  
```

Figur 281 - Bruke løkker til å gjøre gjentakende handlinger

Koden over kjøres så lenge verdien til `$val` er under 10. For hver gang løkka kjøres, øker verdien til `$val` med én og ei stjerne ekstra skrives ut for hver gang løkka kjøres. Som vi så tidligere, vil det å gange en tekst med et tall i Powershell, sørge for at den teksten blir gjentatt like mange ganger som tallet (eller variabelen) på høyre side sier. `"hei"*5` vil altså returnere teksten "hei" fem ganger.



```

1 $domain = "mafoberg.net"
2
3 $servers = @("mafobergcluster", "exchange", "hyp-1", "hyp-3", "edge1", "scvmm")
4
5 for($i=3; $i -lt $servers.count; $i++) {
6     write-host $servers[$i], $domain -separator "."
7 }
  
```

```

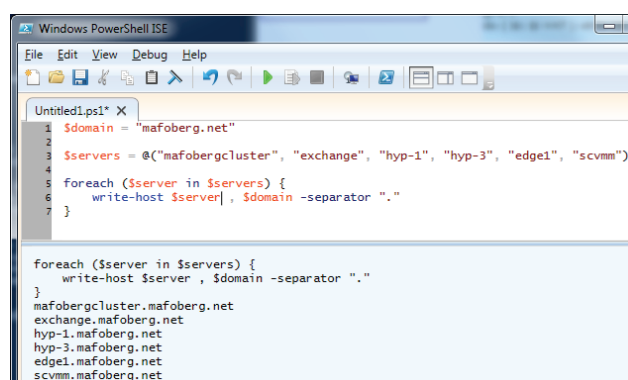
$servers = @("mafobergcluster", "exchange", "hyp-1", "hyp-3", "edge1", "scvmm")
for($i=3; $i -lt $servers.count; $i++) {
    write-host $servers[$i], $domain -separator "."
}
hyp-3.mafoberg.net
edge1.mafoberg.net
scvmm.mafoberg.net
  
```

Figur 282 - for-løkke for å liste ut enkelte element

For-løkke, som vist i er et alternativ til *while*. Den lar deg kjøre igjennom et valgfritt intervall av tall eller elementer. Under har vi et array med oversikt over alle våre servere. Vi ønsker å konsentrere oss om bare de tre siste serverne og ignorere de tre første. Linjene 5-7 forteller derfor at vi starter på element tre (fjerde element, første element er element 0) og skjører de sammen med domenetnavnet vårt i linje 6.

Hadde vi her ville skrevet ut alle elementer fra arrayet `$servers`, ville vi kunne forenkle utlistinga noe med å bruke en *foreach* i stedet.

På linje 5 i Figur 278 skriver vi arrayet på høyre side i kondisjonen til *foreach*, mens har valgt et navn `$server` som navn på det elementet som itereres igjennom. En kan lese linjene 5-7 som "for hver server i serverlista, print ut servernavnet fulgt av det felles domenenavnet".



```

1 $domain = "mafoberg.net"
2
3 $servers = @("mafobergcluster", "exchange", "hyp-1", "hyp-3", "edge1", "scvmm")
4
5 foreach ($server in $servers) {
6     write-host $server, $domain -separator "."
7 }
  
```

```

foreach ($server in $servers) {
    write-host $server, $domain -separator "."
}
mafobergcluster.mafoberg.net
exchange.mafoberg.net
hyp-1.mafoberg.net
hyp-3.mafoberg.net
edge1.mafoberg.net
scvmm.mafoberg.net
  
```

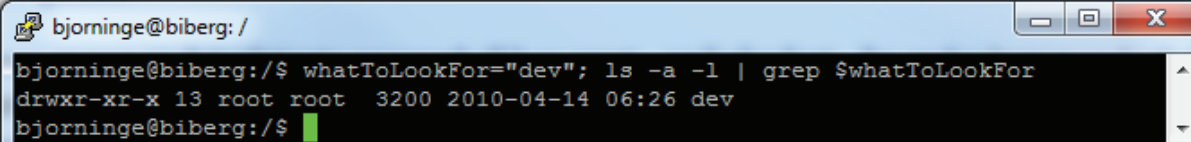
Figur 283 - foreach-løkke går igjennom alle elementer

vi

4.3.2.4 Pipeline

Pipelines er noe av det viktigste Powershell har arvet – og forbedret – fra andre kommandoskall. Det gjør effektivt at to separate Powershellkommandoer kan utføre en enveiskommunikasjon hvor resultatet av den ene kommandoen gis som input til den andre. Under *sh*-skallet i linux og i DOS var

det kun mulig å sende tekst mellom to prosesser. Dette fører ofte til at feil lett kan oppstå, da å tolke tekst korrekt er vanskelig.



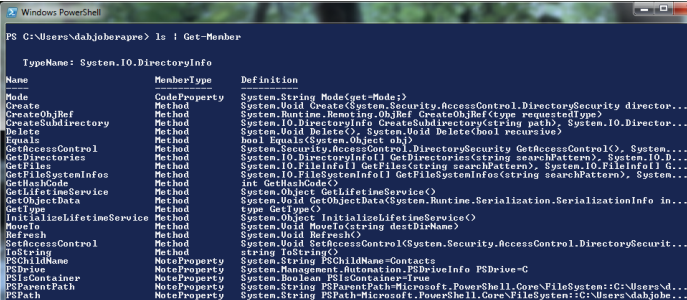
```
bjorninge@biberg: /
bjorninge@biberg:/$ whatToLookFor="dev"; ls -la -l | grep $whatToLookFor
drwxr-xr-x 13 root root 3200 2010-04-14 06:26 dev
bjorninge@biberg:/$
```

Figur 284 - Linux-skall: Utlisting av mapper som inneholder ordet "dev"

Overfor vises et kommandoskall i linux. Her ser vi kommandoen `ls` og parameterne `-a` og `-l` som sørger for at det skjer en utlisting av alle filene i stående mappe, hver fil separert med ei ny linje. Videre pipes dette resultatet til kommandoen `grep`, som søker etter en gitt tekst. Vi søkte etter ei mappe kalt `dev`, og kommandoen returnerte en oversikt over filrettigheter, eier og dato for når mappa sist var endret som tekst. Hva nå om vi ønsker og kun liste ut filas dato for siste endring? Alt som returneres er en tekst som vi må tolke direkte, da gjerne ved å benytte seg av kompliserte regulære uttrykk²¹ for å fjerne den informasjonen som ikke er relevant fra utlistinga overfor.

Powershell løser dette problemet med at å søke i tekst er vanskelig ved å velge en alternativ måte for kommunikasjon. All kommunikasjon mellom prosessene i Powershell er nemlig pakket inn i .NET-baserte objekter, og en kan derfor spørre objektene om de tilfredstiller gitte kriterier basert på såkalte filtre. Det er likevel ikke slik at vi nødvendig må bruke filtre når vi bruker en pipe.

Vi kan også la to kommandoer kommunisere uten at vi filtrere resultatet videre. Tidligere så vi at vi kunne finne ut hvor mange filer ei mappe inneholder, ved å bruke kommandoen `"get-member -in (ls)"` for deretter å liste ut `.count`-egenskapen. Da så vi på resultatet `(ls)` som en helhet returnerte. Ønsker vi å se hvert objekt fra resultatet til `ls` gir, kan en benytte en pipe mellom `ls` og `get-member`, slik:



```
PS C:\Users\dahjoberge> ls | Get-Member

TypeName: System.IO.DirectoryInfo

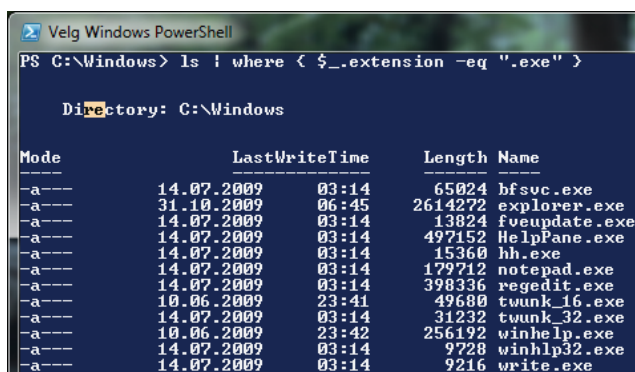
Name      MemberType Definition
-----
Mode      CodeProperty System.String Mode{get-Mode}
Create    Method      System.Void Create(System.Security.AccessControl.DirectorySecurity director...
CreateSubdirectory Method      System.Void CreateSubdirectory(System.String path); System.IO.Directory...
Delete    Method      System.Void Delete(); System.Void Delete(System.String path); System.IO.Direc...
Exists    Method      bool Exists(System.Object obj);
GetAccessControl Method      System.Security.AccessControl.DirectorySecurity GetAccessControl(); System...
GetDirectories Method      System.IO.DirectoryInfo[] GetDirectories(System.String searchPattern); System.IO.D...
GetFiles  Method      System.IO.FileInfo[] GetFiles(System.String searchPattern); System.IO.FileInfo[] G...
GetFileSystemInfos Method      System.IO.FileSystemInfo[] GetFileSystemInfos(System.String searchPattern); System...
GetHashCode Method      int GetHashCode();
GetLifetimeService Method      System.Object GetLifetimeService();
GetObjectData Method      System.Void GetObjectData(System.Runtime.Serialization.SerializationInfo in...
GetType   Method      type GetType();
InitLifetimeService Method      System.Object InitLifetimeService();
Move      Method      System.Void Move(System.String srcDirName, System.String destDirName);
Refresh  Method      System.Void Refresh();
SetAccessControl Method      System.Void SetAccessControl(System.Security.AccessControl.DirectorySecur...
ToString Method      string ToString();
PSChildName NoteProperty System.String PSChildName=Contacts
PSDrive   NoteProperty System.Management.Automation.PSDriveInfo PSDrive=C
PSHome    NoteProperty System.String PSHome=Microsoft.PowerShell.Core\FileSystem:C:\Users\d...
PSPath    NoteProperty System.String PSPath=Microsoft.PowerShell.Core\FileSystem:C:\Users\dahj...
PSProvider NoteProperty System.Management.Automation.ProviderInfo PSProvider=Microsoft.PowerShell.C...
Attributes NoteProperty System.IO.FileAttributes Attributes=ReadOnly
```

Figur 285 - Powershell lister ut alle filer som objekter istedet for tekst

Resultatet en ser her, vil variere, fordi når du bruker pipe-metoden i stedet for å spesifisere `(ls)` som argumentet `"-in"` direkte til `"get-member"` (slik som vi så tidligere), vil få få egenskapsinformasjon kun til de objektene som eksisterer. Dersom stående mappe også inneholder filer i tillegg til mapper, vil en utlisting for `FileInfo`-objekter også ramses opp her.

²¹ regulære uttrykk eller regex som det ofte forkortes, er en matematisk operasjon for å søke i tekst etter kriterier satt av programmereren. Også Powershell støtter regulære uttrykk gjennom `"-match"`-operatøren på strenger.

4.3.2.5 Where-Object Filtre



Figur 286 - Filtrering med where

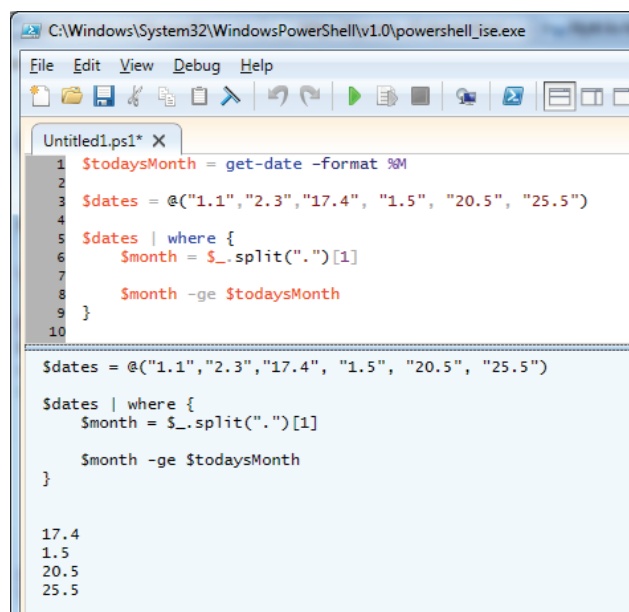
Filtre er altså som tidligere omtalt en måte å avgrense antall objekter som skal listes ut, basert på å se på hvert objekts egenskaper. Hvert objekt fra kommandoen "dir" eller "ls" vil returnere enten et filinfo-objekt eller et mappe-objekt alt ettersom om det listes ut ei mappe eller ei normal fil. Ved å bruke kommandoen "where-object" (alias: "where" eller "?") kan vi hente ut de filene som har et filnavn som slutter på ".exe."

med krøllparenteser – vil kjøres mot alle resultatene fra "ls", og fjerner de objektene som ikke treffer reglene satt i kodeblokk. Merk her at vi har tilgang til en spesiell variabel kalt "\$_" som inneholder det objektet som det itereres over for øyeblikket. Videre brukes sammenlikningsoperatørene som vist tidligere.

Det er også mulig å bruke "where" til å søke i arrays og hashtabeller.

Gitt at vi har et array med viktige datoer for milepæler dette prosjektet:

```
$dates = @("1.1","2.3","17.4", "1.5", "20.5", "25.5")
```



Figur 287 - Utlisting av frister i denne måneden og seinere

"Where" tar imot ei kodeblokk som eneste argument. Denne kodeblokk – som markeres

Vi kan hente ut de datoene som faller i denne inneværende måned og seinere ved først å hente dagens månedsnummer.

Deretter for hver gjennomkjøring henter vi månedsdelen som står etter første punktum i hver av datoene over (linje 6), og sjekker om disse er det samme eller høyere enn dagens dato.

Resultatet gir alle datoer etter fra og med april, ettersom dette scriptet ble kjørt i april måned.

4.3.2.6 Sort-object

”Sort-object” (alias: ”sort”) er en kommando som ofte brukes i samarbeid med ”where”, fordi vi ofte ønsker å redusere antallet utlistede elementer samtidig som de sorteres. Nedenfor er en utlisting som viser hvordan vi kan få Powershell til å vise fram de prosessene som bruker mest av minnet, samtidig som de sorteres fra mest til minst minneforbruk (”-desc”).

Handles	NPM(K)	PM(K)	WS(K)	UM(M)	CPU(s)	Id	ProcessName
594	43	152500	171784	383	361.80	1436	firefox
1466	67	123524	157988	470	105.36	6408	iTunes
148	11	105388	61292	242	348.90	2744	dm
765	15	84308	98208	172		924	svchost
655	49	81776	84604	199	181.18	4608	Dropbox
1345	42	70988	88248	393	129.37	2796	explorer
354	19	61000	30340	144		2296	svchost
399	10	48720	51356	185	3.18	6436	powershell
913	35	46964	31164	138		4028	SearchIndexer
235	9	46120	59176	209	29.97	5792	quasselclient-0.5.2
809	28	40724	77336	300	330.27	7292	WINWORD
914	36	40236	63404	240	152.07	3696	msnmsgr
1676	39	39788	52700	214		968	svchost
616	11	37540	50596	179	3.26	6536	powershell
618	74	34968	54668	199	31.87	4004	Skype
204	9	33784	36820	123	1.78	4732	AppleMobileDeviceHelper
460	18	32496	40968	181	6.27	3324	Dell.ControlPoint
215	9	31940	42576	125	38.97	6932	Foxit Reader
399	37	26668	28556	131	18.17	6496	spotify
411	17	24616	1060	205	7.82	3548	sptcmd
357	25	23420	23768	84		1716	svchost
459	17	23344	25240	174	0.61	3416	BcmDeviceAndTaskStatusService
551	16	23172	29212	136	13.95	5440	wlcomm
755	23	20892	22456	88		1448	svchost

Figur 288 - Sortering basert på høyest minnebruk

Her er det viktig å forstå at ”sort” vil sortere et object basert på hva objektet selv sier det ønsker. For tekst og tall sorteres alt på normal måte, basert på om vi spesifiserte et argument ”-desc”(største til minst) eller ”-asc”(minst til størst)

4.3.2.6 Begrense resultatmengde med Select-object

Så langt har vi sett flere tabellutlistinger med kolonneoverskrifter og rader. Disse radene har vi så sortert, men hvordan velge ut hvilke rader eller kolonner som skal vises?

”Select-object” (alias: ”select”) gjør dette for oss. Med ”Select -first 1” henter vi ut kun det første elementet. Vi kunne også valgt å bruke et annet tall for å hente ut flere element. Alternativt kan vi også hente ut de 5 siste elementene med ”-last 5”.

Handles	NPM(K)	PM(K)	WS(K)	UM(M)	CPU(s)	Id	ProcessName
622	47	207344	236664	416	1	344.32	4312 firefox

Figur 289 - Select kan velge ut enkeltobjekter

Videre kan vi hente ut kun enkelte kolonner av samme informasjon ved å bruke ”select” etterfulgt av kolonnenavnet.

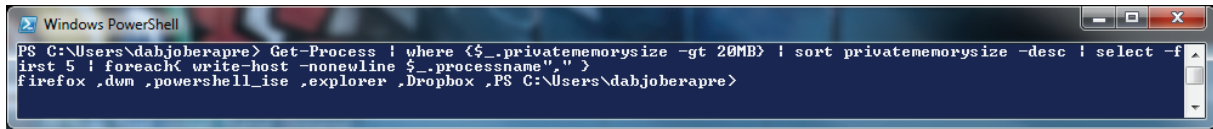
ProcessName
firefox

Figur 290 - Utvelging av prosessnavn

4.3.2.7 Foreach I en pipe

”Foreach”-løkka som tidligere omtalt, kan utføre kommandoer på et sett med resultater. Da må vi først lagre resultatet i en variabel, før vi kan nå resultatet med ”foreach”-løkka. Med ”foreach”

som siste ledd i en pipe, slipper vi derimot dette, og kan benytte oss av den spesielle variabelen kalt "\$_" for å referere til elementet som itereres.



Figur 291 - Alternativ bruk av foreach-løkken

Her ser vi at de fem mest resurskrevende applikasjonene på systemet er lista ut, separert med komma. Vi har benyttet oss av "foreach" for å liste ut applikasjonsnavnene.

4.4 Administrering og fjernstyring med Powershell og remoting

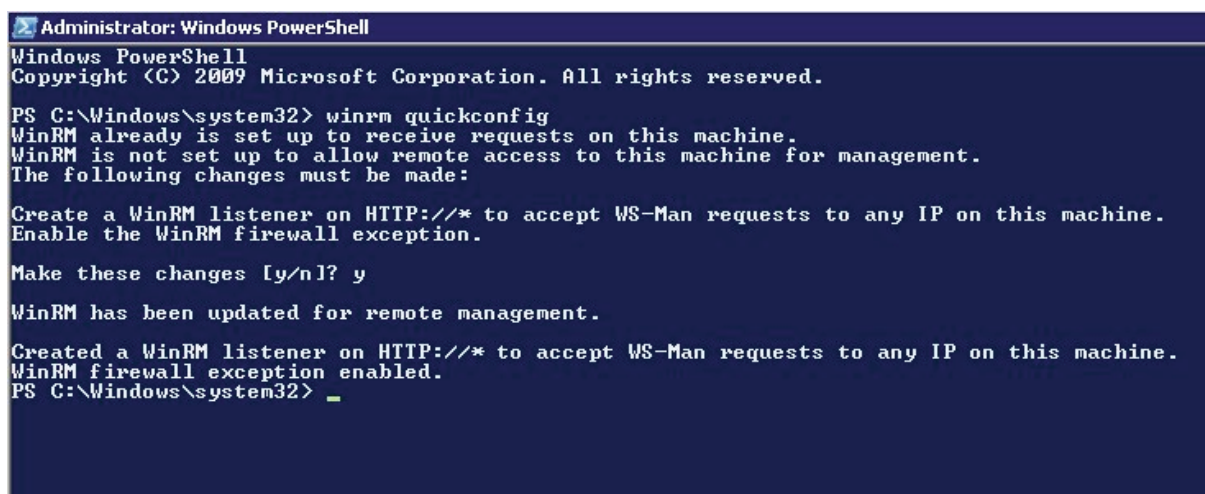
Powershell er et nyttig verktøy i seg selv, men blir straks mer nyttig når du kan kontrollere maskinene som er medlem av Active Directory-området fra én sentral plass. Typisk når en ellers administrerer andre datamaskiner, vil en bruke en kombinasjon av Remote Desktop for deretter å åpne de administrasjonsverktøyene en trenger (inkludert ev.t Powershell) for å gjøre utføre arbeidet. Til en viss grad har Powershell alltid hatt støtte for å hente informasjon fra andre datamaskiner (gjennom kommandoen `gwmi` – som bruker Windows Management Instrumentation), men det var først i Powershell v2 at støtte for remoting var inkludert.

Remoting lar deg koble til én eller flere kjørende datamaskiner som har Powershell v2 installert, og deretter utføre kommandoer på én eller flere av disse datamaskinene. Dette kan være manuelle sesjoner, eller også automatiske rutiner for å samle inn informasjon eller utføre vedlikehold.

4.4.1 Aktivere Powershell remoting

Å aktivere ekstern adgang i Powershell, består av to steg som må gjennomføres på hver server og arbeidsstasjon som det er ønskelig å kontrollere; en må aktivere WinRM, en tjeneste som lytter på nettverket, og en må aktivere Powershell remoting.

Separasjonen mellom winrm og Powershell remoting er gjort slik for å tillate at en maskin kan nås også gjennom andre kommandolinje-verktøy. For eksempel er det fullt mulig og ikke aktivere Powershell remoting, men likevel bruke winrm til å få tilgang til en ekstern maskins DOS-ledetekst (et batch-skall).



```
Administrator: Windows PowerShell
Windows PowerShell
Copyright (C) 2009 Microsoft Corporation. All rights reserved.

PS C:\Windows\system32> winrm quickconfig
WinRM already is set up to receive requests on this machine.
WinRM is not set up to allow remote access to this machine for management.
The following changes must be made:

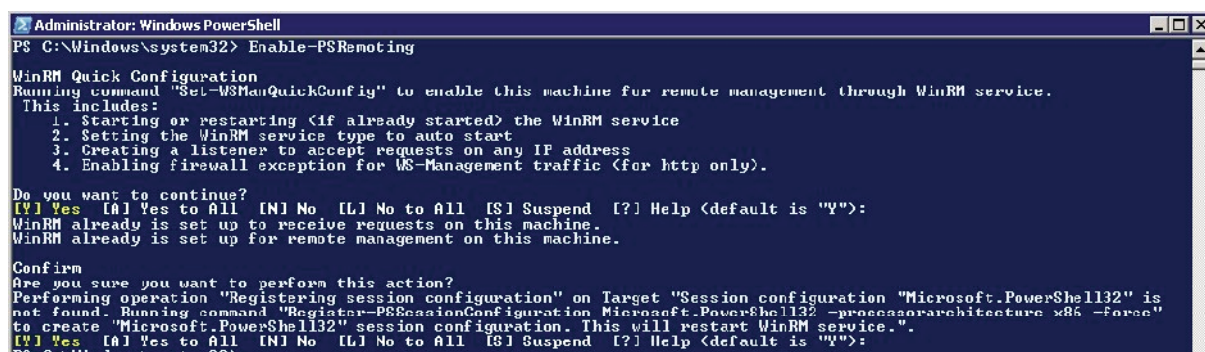
Create a WinRM listener on HTTP://* to accept WS-Man requests to any IP on this machine.
Enable the WinRM firewall exception.

Make these changes [y/n]? y
WinRM has been updated for remote management.

Created a WinRM listener on HTTP://* to accept WS-Man requests to any IP on this machine.
WinRM firewall exception enabled.
PS C:\Windows\system32> _
```

Figur 292 - Aktivering av en remote-listener

Her er vi logget inn på en server vi ønsker å aktivere Powershell remoting på. Først kjører vi kommandoen `wimrm quickconfig` og deretter skriver vi inn `y` for å akseptere.

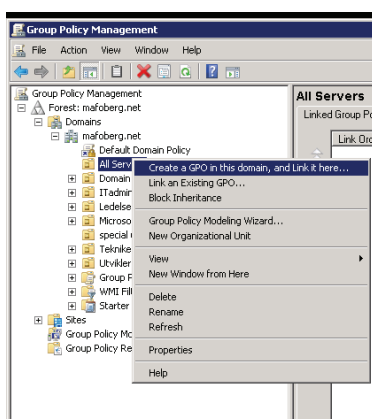


Figur 293 - Aktivisering av Powershell Remoting

Deretter har vi kjørt kommandoen "Enable-PSRemoting" for å aktivere remoting. Her er det mulig at flere valg kan dukke opp, bl.a. ser vi at det mangler en sesjonskonfigurasjonsfil, og at Powershell vil opprette denne for oss.

4.4.1.1 Aktivere Powershell remoting med GPO

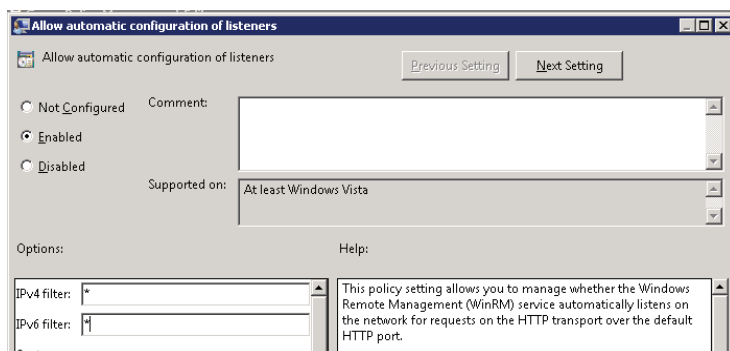
Når antall servere er begrenset, er den manuelle metoden ved å gå inn på hver server for å aktivere remoting kanskje enklest. Når vi derimot ønsker å finkontrollere hvilke maskiner som skal kunne nås ved hjelp av powershell, må vi inn på en Active Directory server.



Figur 294 - Ny GPO for servere

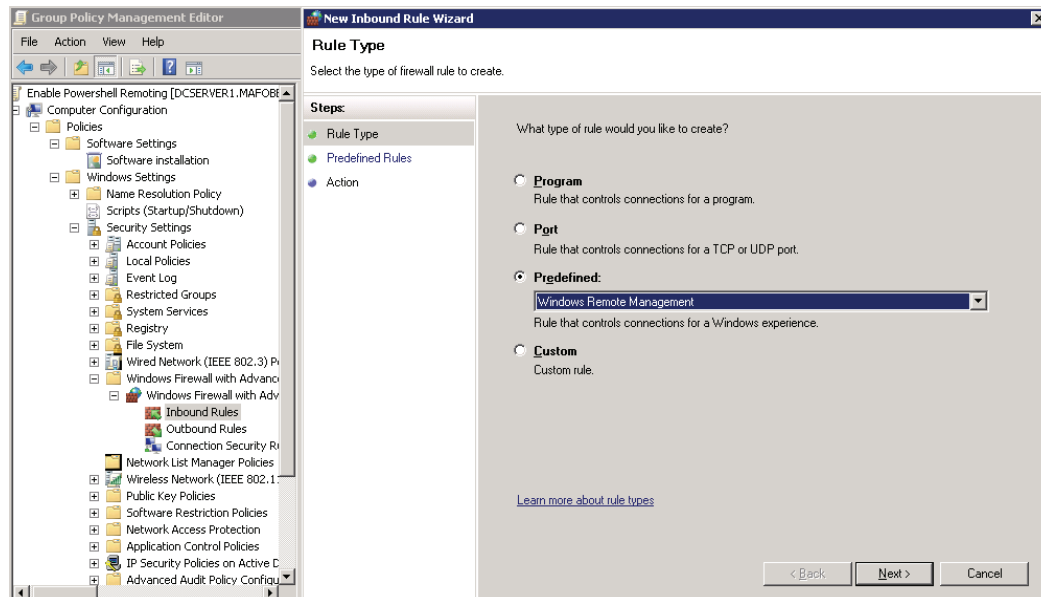
I Figur 289 ser vi at vi har laget en ny GPO for alle servere i vår OU "All Servers". Kall den for "Enable Powershell Remoting" og rediger den ved å høyreklikke og velge "Edit...". Et nytt "Policy Editor"-vindu vil dukke opp.

Vi skal nå sette en ny GPO for "Computer Configuration". Vi velger Policies -> Administrative templates-> Windows Components og aktiverer "Allow Automatic configuration of listeners".



Figur 295 - Aktivere Listeners

Her har vi aktivert WinRM Listeners. Merk at vi må skrive inn ei stjerne i filtrene for IPv4 og IPv6.



Figur 296 - Tillate innkommende forbindelser

Over har vi også lagt til en egen brannmurregel som aktiverer portene for tilgang til Remote Management. Dette gjøres under Computer Configuration -> Security settings og finner Windows Firewall With Advanced Security. Vi kunne valgt å legge til regelen manuelt, men Windows har allerede laget en predefinert regel for dette, som vi velger å ta i bruk. Gå igjennom så de to neste stegene, velg ”allow the connection” på siste steg og trykk ”Finish”

Merk at forandringer som gjøres i Active Directory ikke aktiveres umiddelbart. Det kan ta opp til én time før forandringer er synlig på alle maskiner. Dersom vi ønsker å gjøre endringer synlig med en gang, kan vi gå inn på hver server og skrive følgende kommando i enten DOSLedetekst eller i en Powershell-sesjon:

```
gpupdate /force
```

Dette tvinger maskinen til å lese inn nye konfigurasjoner fra domenekontrolleren.

4.4.2 Forskjellige typer remoting

4.4.2.1 Klassisk remoting og Windows Management Infrastructure (WMI)

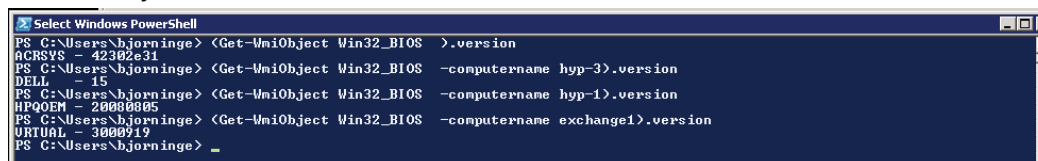
Ved hjelp av klassisk remoting, vil ikke Powershell være involvert direkte i å koble til andre maskiner. I stedet må en bruke kommandoer som støtter remoting uavhengig av om Powershell remoting er aktivert eller ikke. Flere kommandoer har denne muligheten, og de kan lett identifiseres med følgende kode:

```
man * -param computername
```

"man" er forøvrig en mye brukt alias for kommandoen "get-help". Kommandoen henter ut alle kommandoer som har et parameter "computername". Utlistinga er for stor til å gjengis her, men inneholder omtrent 40 kommandoer.

Ulempen med å bruke klassisk remoting er at ikke alle kommandoer støtter det, samtidig som en vil logges inn med sitt nåværende brukernavn og passord, med mindre en for hver gang bruker et ekstra argument "-credentials". I tillegg støtter kun omtrent halvparten av de utlistede kommandoene spesifisering av kredensialer, og det utføres en oppkobling og nedkobling mot den eksterne maskinen for hver gang en kommando kjører. Derfor egner dette seg kun for enkle oppgaver.

WMI og kommandoen for å styre dette i Powershell, nemlig "Get-WmiObject" (alias: "gwmi") er en av kommandoen som støtter klassisk remoting. WMI tilbyr et lass med kommandosett for å hente ut alt fra bios- og operativsysteminfo til å finne monitoroppløsning og mye annet. Se "gwmi -list" for en detaljert liste.



```
Select Windows PowerShell
PS C:\Users\bjorninge> Get-WmiObject Win32_BIOS .version
ACRSYS - 42302e31
PS C:\Users\bjorninge> Get-WmiObject Win32_BIOS -computername hyp-3.version
DELL - 15
PS C:\Users\bjorninge> Get-WmiObject Win32_BIOS -computername hyp-1.version
HPQOEM - 20080805
PS C:\Users\bjorninge> Get-WmiObject Win32_BIOS -computername exchange1.version
VIRTUAL - 3000919
PS C:\Users\bjorninge>
```

Figur 297 - Koble til eksterne maskiner med "-computername"

Her ser vi et eksempel hvor vi har hentet ut BIOS-versjonen til flere av maskinene våre i nettverket. Virtuelle maskiner i Hyper-V vil ha et BIOS-navn som starter på VIRTUAL som vi ser eksempel på nederst.

4.4.2.2 Powershell remoting

Powershell remoting løser problemene som klassisk remoting hadde, ved at du lar Powershell videresender alle kommandoer til den andre maskinen over en oppkobling. Kode legges i en kodeblokk som kjøres i en ny sesjon på den eksterne maskinen.

Når en manuelt skal inn på en server for å administrere den, brukes kommandoen "Enter-PSSession" etterfulgt av datamaskinnavnet som skal administreres. Nedenfor ser vi et eksempel på at vi benytter oss av dette og deretter skriver ut litt informasjon om miljøet vi befinner oss i. Vi ser brukernavnet og domenonavnet forblir det samme, men at datamaskinnavnet skifter fra "hyp-2" til "hyp-1"


```
Administrator: Windows PowerShell
PS C:\Windows\system32> write-host $env:username"@"$env:computername"."$env:userdnsdomain
bjorninge@HYP-2.MAFOBERG.NET
PS C:\Windows\system32> Enter-PSSession hyp-1
[hyp-1]: PS C:\Users\bjorninge\Documents> write-host $env:username"@"$env:computername"."$env:userdnsdomain
bjorninge@HYP-1.MAFOBERG.NET
[hyp-1]: PS C:\Users\bjorninge\Documents> _
```

Figur 298 - Remoting vha. Enter-PSSession

"Enter-PSSession" fungerer altså bra til enkel innlogging, men blir problematisk om vi skal utføre kommandoer på flere maskiner samtidig, eller om vi skal lage automatiserte oppgaver. Derfor benytter vi oss ofte i stedet av kommandoen `Invoke-Command` (alias: "icm") i stedet.

```
Select Administrator: Windows PowerShell
PS C:\Users\Administrator.MAFOBERG> write-host $env:computername
HYP-2
PS C:\Users\Administrator.MAFOBERG> icm( write-host $env:computername) -computer hyp-1, exchange1
EXCHANGE1
HYP-1
PS C:\Users\Administrator.MAFOBERG> _
```

Figur 299 - Invoke-command (icm) kan koble til flere maskiner

Her legger vi merke til at "icm" kan kjøre samme sett med kommandoer på flere maskiner samtidig. Skal vi kjøre flere kommandoer, må vi likevel lage en ny forespørsel med "icm", så problemet med oppkobling og nedkobling av forbindelsen vedvarer.

Derimot ved å bruke "New-PSSession" kan vi lagre en sesjon mot en eller flere maskiner.

```
1 $s1 = new-pssession hyp-1
2 $s2 = new-pssession hyp-2
3
4 Invoke-Command {
5     #henter ip-adresser
6     $ips = gwmi win32_NetworkAdapterConfiguration | where {$_.IPEnabled -eq $true } | foreach { $_.IPAddress }
7     #lister ut ipadresser til hver maskin
8     write-host $env:computername : ($ips)
9 } -session ($s1,$s2)
```

Figur 300 Invoke-command (icm) vha. lagrede sesjoner

Her brukte vi "-session" i stedet for "-computer", fordi vi brukte en eksisterende forbindelse. Forbindelsene "\$s1" og "\$s2" vil fortsette å være åpne, og påfølgende kall mot "invoke-command" som også spesifiserer argumentet "-session" vil bruke samme forbindelse for å utføre kommandoer. På linje 6 har vi brukt WMI lokalt på hver maskin for å hente nettverksadaptore som også har en IP-adresse.

Datamaskinas datamaskinnavn, samt IP-adresser vil da listes ut, slik:

```
PS C:\Users\Administrator.MAFOBERG> Invoke-Command {
>> #henter ip-adresser
>> $ips = gwmi win32_NetworkAdapterConfiguration | where {$_.IPEnabled -eq $true } | foreach { $_.IPAddress }
>> #lister ut ipadresser til hver maskin
>> write-host $env:computername : ($ips)
>> } -session ($s1,$s2)
>>
HYP-1 : 172.16.101.62 158.38.56.251 169.254.1.188 fe80::9131:30da:157b:3e35
HYP-2 : 158.38.56.245
PS C:\Users\Administrator.MAFOBERG> _
```

Figur 301 - Utlisting av IP-adresser for hver server

En kommando som opprettes med "New-PSSession", kan du også nå ved å bruke "Enter-PSSession" og "-session"-argumentet.

En ting å merke seg, er at du kan ha maks fem forbindelser mot én og samme maskin samtidig, så en må huske på å lukke disse forbindelsene manuelt. "Remove-PSSession *" gjør dette for deg.

Stjerna i forrige kommando signaliserer at du ønsker å lukke alle åpne sesjoner. Dersom en heller vil lukke spesifikke sesjoner, slik som de vi lagde i variablene \$s1 og \$s2 overfor, kan en skrive "remove-pssession (\$s1, \$s2)"

4.4.2.3 Bruk av lokale .ps1-filer i eksterne sesjoner

Til nå har vi sett hvordan vi importerer filer med dot-sourcing i Figur 264, aktivering av eksekverbare script i Figur 265, og vi har gått igjennom hvordan vi oppretter sesjoner og bruker de ved hjelp av "invoke-command" ("icm"). Derimot, et problem som vi kommer til å støte på er at det ikke alltid vil være slik at alle servere har tilgang til samme scriptfiler. Da kan det i utgangspunktet bli trasig å kjøre en egenlagd kommando mot flere datamaskiner samtidig. En løsning på dette er å bruke "icm" sitt "-filePath"-argument. Dette leser den lokale fila som spesifiseres og overfører den til de andre maskinene. Disse maskinene får nå tilgang til all koden som ligger i vår ".ps1"-fil.

Fordi "-filePath" ikke kan brukes samtidig med en blokk med kode, må vi utføre funksjonene i fila i et nytt kall til icm.

```
1 #en liste over servere som de to kallene til icm skal kjøres på
2 #dersom serverne er i et felles domene med maskinene som dette scriptet kjøres på,
3 #holder det med datamaskinnavnet. ellers må fullt dom navn eller IP-adresse spesifiseres
4 $servers = ("scvmm", "hyp-1", "hyp-3", "exchange1")
5
6 #opprett like mange sesjoner som det er servere i lista overfor
7 $sessions = $servers | foreach{ new-pssession $_ }
8
9 #dette trikset overfører fila til de andre sesjonene
10 #denne stien må eksistere på denne maskinen.
11 icm -Session $sessions -FilePath .\en-fil-du-vil-inkludere.ps1
12
13
14 icm {
15     write-host $env:computername ":"
16     #bruk kode fra den fila du inkluderte istad, her. Koden vil kjøres på de eksterne sesjonene
17 } -Session $sessions
18
19
```

Figur 302 – Annotert eksempel på bruk av lokale filer i eksterne sesjoner.

Dette og et til eksempel kan finnes i vedlagt script, navngitt "remote-transfer-functions.ps1"

4.4.2.4 Implisitt remoting

Vi har nå sett på former for såkalt eksplisitt remoting. Det vil si at du spesifikt ber Powershell utføre en kommando i en annen sesjon på en annen server. La oss si at du sitter på en arbeidsstasjon og da ønsker å utføre installasjon av noen tilleggsfeatures på en Exchange-server, slik at Outlook Web Access blir tilgjengelig. Vi brukte tidligere under installasjonen av Exchange en Powershell-kommando for dette:

```
1 Import-Module ServerManager
2 Add-WindowsFeature NET-Framework,RSAT-ADDS,Web-Server,Web-Basic-Auth,Web-Windows-Auth,Web-Metabase,Web-Net-Ext,Web-Lgcy-Mgmt-Console,WAS-Process-Model,RSAT-Web-Server
3 Set-Service NetTcpPortSharing -StartupType Automatic
```

Figur 303 - Implisitt installasjon av Serverprogramvare

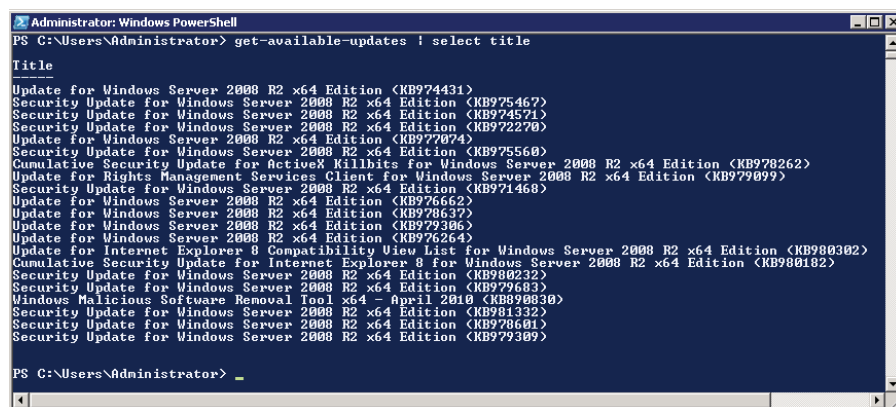
Et alternativ til å bruke ”enter-pssession” eller ”invoke-command”, er å importere servermodulen fra serveren til arbeidsstasjonen og bruke den derfra. Så selv om du står i en sesjon på din arbeidsstasjon, så vil kommandoer som importeres, kjøres på arbeidsstasjonen, men utføre sine oppgaver serveren. Vi skal ikke gå nærmere inn på dette her, men en kan se hjelpesidene til kommandoen ”import-session” for mer informasjon.

4.4.3 Generell administrasjon av servere

Vi skal nå se på hvordan vi kan hente ut og manipulere sentrale deler av operativsystemet på en server. Vi skal i denne seksjonen gå sentralt inne på generell administrering, men skal i de påfølgende seksjonene vise hvordan vi kan benytte oss av rollene som er installert på andre servere, deriblant spesielt AD, SVMM og Exchange.

4.4.3.1 Installasjon av oppdateringer

Installasjon av oppdateringer er noe av det viktigste for å holde et system sikkert mot angrep. Vi har derfor vedlagt dette dokumentet fila `updates.ps1`. Den inneholder to funksjoner, ”`get-available-updates`” og ”`install-updates`”. Disse funksjonene tar begge i bruk Windows Update Session over COM. Eksemplene nedenfor går ut ifra at fila `updates.ps1` er inkludert i kjørende sesjon. Til dette bruker du et punktum etterfulgt av mellomrom, punktum og en skråstrek, deretter filnavnet: ”`. \updates.ps1`”. Denne laster inn koden i `ps1`-fila og lar deg bruke funksjonene fra den i den lokale sesjonen. Denne prosessen kalles for øvrig dot-sourcing.

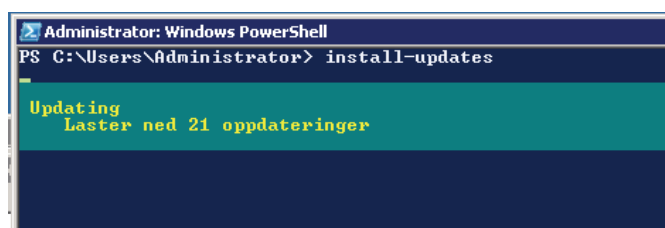


```
Administrator: Windows PowerShell
PS C:\Users\Administrator> get-available-updates | select title
Title
Update for Windows Server 2008 R2 x64 Edition (KB974431)
Security Update for Windows Server 2008 R2 x64 Edition (KB975467)
Security Update for Windows Server 2008 R2 x64 Edition (KB974571)
Security Update for Windows Server 2008 R2 x64 Edition (KB972270)
Update for Windows Server 2008 R2 x64 Edition (KB977074)
Security Update for Windows Server 2008 R2 x64 Edition (KB975560)
Cumulative Security Update for ActiveX Killbits for Windows Server 2008 R2 x64 Edition (KB978262)
Update for Rights Management Services Client for Windows Server 2008 R2 x64 Edition (KB979099)
Security Update for Windows Server 2008 R2 x64 Edition (KB971468)
Update for Windows Server 2008 R2 x64 Edition (KB976662)
Update for Windows Server 2008 R2 x64 Edition (KB978637)
Update for Windows Server 2008 R2 x64 Edition (KB979306)
Update for Windows Server 2008 R2 x64 Edition (KB976264)
Update for Internet Explorer 8 Compatibility View List for Windows Server 2008 R2 x64 Edition (KB980302)
Cumulative Security Update for Internet Explorer 8 for Windows Server 2008 R2 x64 Edition (KB980182)
Security Update for Windows Server 2008 R2 x64 Edition (KB980232)
Security Update for Windows Server 2008 R2 x64 Edition (KB979683)
Windows Malicious Software Removal Tool x64 - April 2010 (KB998830)
Security Update for Windows Server 2008 R2 x64 Edition (KB981332)
Security Update for Windows Server 2008 R2 x64 Edition (KB978601)
Security Update for Windows Server 2008 R2 x64 Edition (KB979309)
```

Figur 304 - Utlisting av oppdateringer

Overfor ser vi eksempel på at vår kommando er kjørt. Resultatet er mange objekter som representerer hver tilgjengelige oppdatering. Vi har videre valgt ut kun tittelen på hver oppdatering for visning.

Her har vi kjørt kommandoen ”`install-updates`” som kommer fra samme kildefil. Kommandoen må laste ned alle oppdateringer før de kan installeres (dersom de ikke er lasta ned fra før) og vil derfor ta noe tid å utføre.



```
Administrator: Windows PowerShell
PS C:\Users\Administrator> install-updates

Updating
Laster ned 21 oppdateringer
```

Figur 305 - Installasjon av oppdateringer begynner

```

PS C:\Users\Administrator> install-updates
Title                                                                    EulaAccepted R
-----
Update for Windows Server 2008 R2 x64 Edition (KB974431)                True S
Security Update for Windows Server 2008 R2 x64 Edition (KB975467)       True S
Security Update for Windows Server 2008 R2 x64 Edition (KB974571)       True S
Security Update for Windows Server 2008 R2 x64 Edition (KB972270)       True S
Update for Windows Server 2008 R2 x64 Edition (KB977074)                True S
Security Update for Windows Server 2008 R2 x64 Edition (KB975560)       True S
Cumulative Security Update for ActiveX Killbits for Windows Server 2008 R2 x64 Edition (KB978262) True S
Update for Rights Management Services Client for Windows Server 2008 R2 x64 Edition (KB979099) True S
Security Update for Windows Server 2008 R2 x64 Edition (KB971460)       True S
Update for Windows Server 2008 R2 x64 Edition (KB976662)                True S
Update for Windows Server 2008 R2 x64 Edition (KB978637)                True S
Update for Windows Server 2008 R2 x64 Edition (KB979306)                True S
Update for Windows Server 2008 R2 x64 Edition (KB976264)                True S
Update for Internet Explorer 8 Compatibility View List for Windows Server 2008 R2 x64 Edition (KB980302) True S
Cumulative Security Update for Internet Explorer 8 for Windows Server 2008 R2 x64 Edition (KB980182) True S
Security Update for Windows Server 2008 R2 x64 Edition (KB979683)       True S
Security Update for Windows Server 2008 R2 x64 Edition (KB979683)       True S
Windows Malicious Software Removal Tool x64 - April 2010 (KB890830)     False F
Security Update for Windows Server 2008 R2 x64 Edition (KB981332)       True S
Security Update for Windows Server 2008 R2 x64 Edition (KB978601)       True S
Security Update for Windows Server 2008 R2 x64 Edition (KB979309)       True S

```

Figur 306 - Installasjon av oppdateringer fullført

Etter kommandoen har fullført, vil det listes opp oppdateringer som er installert. Typisk vil alle oppdateringer bli installert, men som vi også ser til venstre, har én oppdatering feilet fordi vi manuelt måtte akseptere lisensavtalen til oppdateringa før vi går videre.

4.4.3.2 Belastning av servere

Det er bestandig nyttig å ha informasjon om bruk av serverne tilgjengelig. Å finne denne informasjonen er derimot ikke triviell, det må en del kode for å velge ut kun den informasjonen vi trenger. Det er derfor opprettet fila `wmi-functions.ps1` (som må inkluderes med kommandoen `". .\wmi-functions.ps1"`) som har de funksjonene vi trenger.

Disse funksjonene benytter forøvrig WMI og SQL-spørringer mot WMI-databasen for å hente informasjonen som trengs. Formålet med WMI (Windows Management Instruments, som tidligere omtalt) er å definere et sett med miljø-uavhengige spesifikasjoner som gjør det mulig å scripte tilgang til interne tjenester i et windows-miljø, slik som utlisting av all mulig metainformasjon. Hovedfordelen med WMI er at det er veldig enkelt å hente ut informasjon, og det kan hentes ut vha flere script- og programmeringsspråk.



```
Select Windows PowerShell
Windows PowerShell
Copyright (C) 2009 Microsoft Corporation. All rights reserved.

Logged in as bjorninge@HVP-2.MAF0BERG.NET
PS C:\Users\bjorninge> cd .\scripts
PS C:\Users\bjorninge\scripts> . .\wmi-functions.ps1
PS C:\Users\bjorninge\scripts> get-disk-space

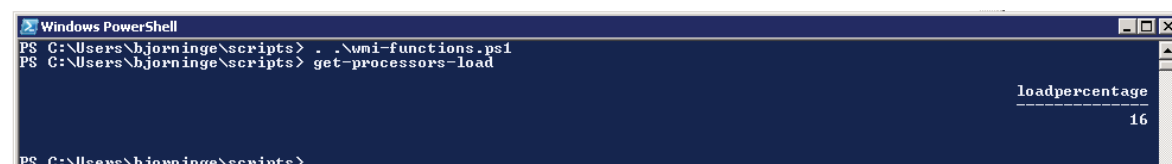
disk                               størrelse[GB]          ledig plass[GB]
-----
C:                                398.62                 379.23
D:                                205.45                 156.50

PS C:\Users\bjorninge\scripts>
```

Figur 307 - Utlisting av brukt plass

Vi ser at kommandoen `get-disk-space` ble tilgjengelig etter inkludering av `wmi-functions.ps1`. Kommandoen kjøres uten noen argument, og lister ut alle diskene og ledig kapasitet.

Vi kan også hente ut gjennomsnittlig prosessorlast fra samme inkluderte fil. Kommandoen heter `get-processor-load`, og brukes slik som gjengitt i Figur 308.



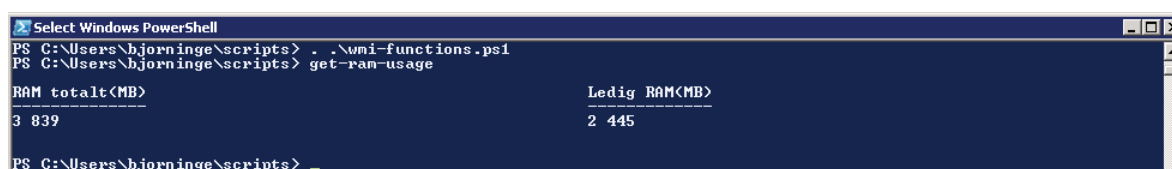
```
Windows PowerShell
PS C:\Users\bjorninge\scripts> . .\wmi-functions.ps1
PS C:\Users\bjorninge\scripts> get-processors-load

loadpercentage
-----
16

PS C:\Users\bjorninge\scripts>
```

Figur 308 - Utlisting av prosessorlast

Tilsvarende har vi også mulighet til å hente ut total minnemengde og ledig minne med kommandoen `"get-ram-usage"`.



```
Select Windows PowerShell
PS C:\Users\bjorninge\scripts> . .\wmi-functions.ps1
PS C:\Users\bjorninge\scripts> get-ram-usage

RAM totalt[MB]          ledig RAM[MB]
-----
3 839                 2 445

PS C:\Users\bjorninge\scripts>
```

Figur 309 - Utlisting av ledig og tilgjengelig ram

4.4.3.3 Sammenkjøring av all lastinformasjon.

Ved hjelp av eksempelkode fra `"remote-transfer-functions.ps1"`, kan vi nå liste ut både minnebruk, bruk av lagringsplass og bruk av prosessor for alle maskiner i domenet.

```

34 #####
35 #Avansert eksempel ved bruk av vedlagt script "wmi-functions.ps1"
36 $servers = ("scvmm", "hyp-1", "hyp-3", "exchange1")
37
38 $sessions = $servers | foreach{ new-pssession $_ }
39 #dette trikset overfører fila til de andre sesjonene
40 #denne stien må eksistere på denne maskinen.
41 icm -Session $sessions -FilePath .\scripts\wmi-functions.ps1
42
43 #vi bruker icm til å utføre kommandoene mot hver sesjon.
44 icm {
45
46     $load = get-processors-load;
47     $space = get-disk-space;
48     $ram = get-ram-usage;
49     write-host $env:computername ":" -foregroundColor DarkGreen -backgroundColor white;
50     $load;$space;$ram;
51 } -Session $sessions | format-list

```

Figur 310 - Kode for lastberegning på flere servere

```

Select Windows PowerShell
SCVMM :
loadpercentage      : 2
PSComputerName      : scvmm
RunspaceId          : fbac7b64-82fe-4efd-8e56-8f039e879cd4
PSShowComputerName  : True

disk
storrelse (GB)      : 29,90
Ledigplass(GB)      : 17,01
PSComputerName      : scvmm
RunspaceId          : fbac7b64-82fe-4efd-8e56-8f039e879cd4
PSShowComputerName  : True

RAM totalt(MB)      : 2 100
Ledig RAM(MB)       : 1 034
PSComputerName      : scvmm
RunspaceId          : fbac7b64-82fe-4efd-8e56-8f039e879cd4
PSShowComputerName  : True

hyp-3 :
loadpercentage      : 4
PSComputerName      : hyp-3
RunspaceId          : adc4d35d-e661-4970-9550-fb9311d4db56
PSShowComputerName  : True

disk
storrelse (GB)      : 148,91
Ledigplass(GB)      : 130,58
PSComputerName      : hyp-3
RunspaceId          : adc4d35d-e661-4970-9550-fb9311d4db56
PSShowComputerName  : True

disk
storrelse (GB)      : 10,00
Ledigplass(GB)      : 9,91
PSComputerName      : hyp-3
RunspaceId          : adc4d35d-e661-4970-9550-fb9311d4db56
PSShowComputerName  : True

```

Figur 311 - Utdrag fra lastberegningens utskrift

Scriptet fram til linje 44 er forklart i seksjon 4.4.2.3 Bruk av lokale .ps1-filer i eksterne sesjoner. I siste kall mot `icm` på linje 44, henter vi ut systemlastinformasjon i variabler. Vi skriver så ut maskinnavnet fulgt av alle disse beregningene. Det er her meget viktig å lagre denne informasjonen i variabler først, for å få en korrekt utlistering.

I Figur 306 ser vi et utdrag fra kjøring av koden overfor. Koden kjøres på de eksterne serverne, og kommer tilbake til sesjonen på den lokale maskina som kjører scriptet. Under denne prosessen ser vi også at Powershell har lagt til noen ekstra egenskaper slik som `PSComputerName`, `RunspaceId` og `PSShowComputerName`.

Disse ekstra egenskapene brukes av Powershell for å kommunisere mellom to sesjoner, og er helt ufarlig.

4.4.3.4 Behandling av prosesser

Under introduksjonen til Powershell har vi sett flere måter å liste ut kjørende programmer på systemet, deriblant sortering og utvelgning av de prosessene som brukt mest minne.

```
1 Get-Process | sort privatememorysize -desc
```

```
PS C:\Users\dabjoberapre\Documents\My Dropbox\mafo-bjorn\bachelor-prosjekt\scripts> Get-Process | sort privatememorysize -desc
```

Handles	NPM(K)	PM(K)	WS(K)	VM(M)	CPU(s)	Id	ProcessName
578	65	364724	370260	605	1 923,65	7356	firefox
689	28	139772	109008	398	100,92	7228	powershell_ise
1638	53	108164	107712	466	410,95	3588	explorer
239	10	99820	84124	252	62,56	1868	quasselclient-0.5.2

Figur 312 - Utlisting av prosesser sorter på minnebruk

Her ser vi koden som brukt tidligere. Det som derimot ikke vises her, er at hver prosess vet sin egen tilstand, slik at de prosessene som kjører vil ha en egenskap "Responding" satt til False. Flere slike egenskaper kan vi for øvrig finne ved å kjøre "get-process | get-member"

```
1 Get-Process | sort privatememorysize -desc | select processname, @{Name="Kræsjet?";Expression={( -not $_.responding )}}
```

```
PS C:\Users\dabjoberapre\Documents\My Dropbox\mafo-bjorn\bachelor-prosjekt\scripts> Get-Process | sort privatememorysize -desc | select processname, @{Name="Kræsjet?";Expression={( -not $_.responding )}}
```

ProcessName	Kræsjet?
powershell_ise	False
firefox	True
explorer	False

Figur 313 - utlisting av scriptet list-non-responding-apps.ps1

I Figur 314 har vi listet ut alle programmer sortert på minnebruk. Som istad ser vi at programmet Firefox har hengt seg under kjøring, og ved å finne de programmene som har sin responding-egenskap satt til en negativ verdi (her symbolisert med "-not \$_.responding"), kan vi så avslutte disse fra Powershell. Det meste under skal være kjent. Variabelen "\$_" refererer til hvert program som blir funnet til å være ikke-responderende. På linje 3 lister vi så ut programnavnet, og linje 4 bruker kommandoen "kill" på dette objektet for å avslutte det.

```
1 Get-Process | where { -not $_.responding } | foreach{
2
3     write-host "avslutter" ($_.processname)
4     $_.kill()
5 }
```

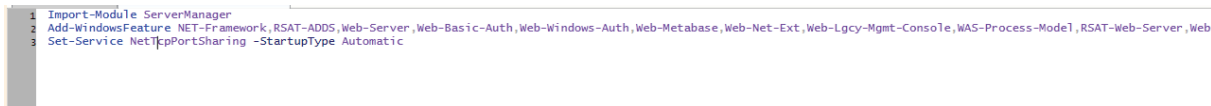
```
PS C:\Users\dabjoberapre\Documents\My Dropbox\mafo-bjorn\bachelor-prosjekt\scripts> Get-Process | where { -not $_.responding } | foreach{
    write-host "avslutter" ($_.processname)
    $_.kill()
}
avslutter firefox
```

Figur 314 - Avslutting av ikke-responderende programmer

4.4.3.5 Diagnostikk ved hjelp av Eventlog og Get-WindowsFeature

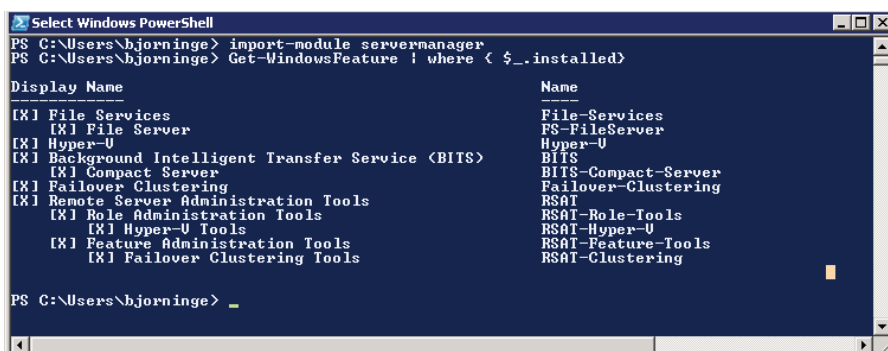
Get-WindowsFeature

Vi viste tidligere hvordan vi installerer en tjeneste på datamaskinen ved å bruke ServerManager-modulen og kommandoen "Add-Windows-Features".



Figur 315 - Installasjon av Roller eller features

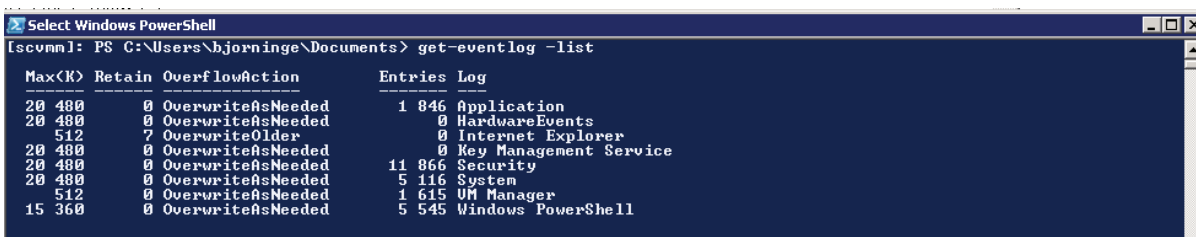
Ved feilsøking kan det være interessant og se hvilke tjenester som faktisk er installert og kjører. Dette kan vi liste ut ved å bruke Get-WindowsFeature.



Figur 316 - Liste ut installerte roller of features

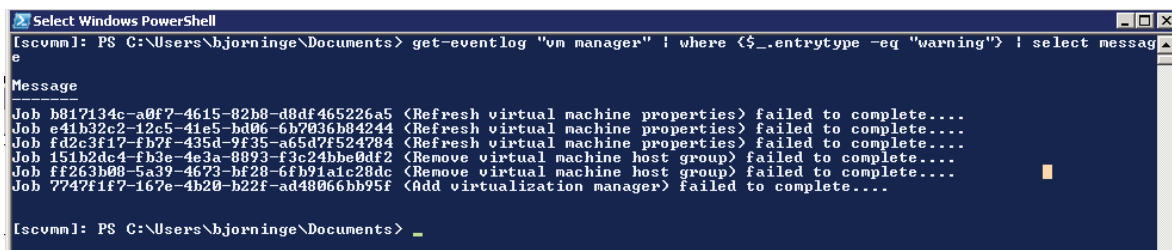
Get-Eventlog

Eventloggen er veldig greit å ha, ettersom alle programmer fra Microsoft og en del tredjepartsverktøy skriver til systemloggen dersom det oppstår situasjoner som krever vedlikehold av en systemadministrator. Nedenfor har vi kjørt "get-eventlog -list" på en System Center Virtual Machine Manager Server. Her ser vi at vi har tilgang til en egen logg for VM Manager.



Figur 317 - Finne tilgjengelige logger

"Get-eventlog "VM Manager" " henter så ut alle hendelser som SCVMM har loggført. Ved å spørre etter kun den informasjonen som har en Entrytype satt til "Warning", får vi tak i alle kritiske hendelser. .. viser dette, hvor vi i har henta ut meldingene i loggen:

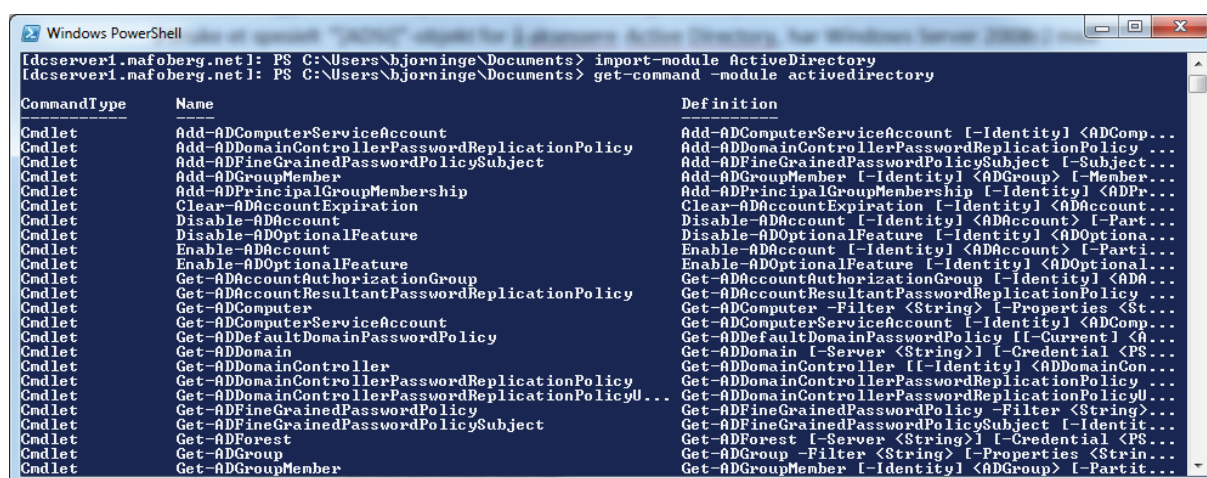


Figur 318 - Advarselslogger fra SCVMM

4.5 Powershelladministrasjon av AD .

Active Directory(AD) er en sentral base for innlegging av brukere, datamaskiner og grupper av disse. I tillegg kommer metainformasjon, slik som epostadresser, hjemmekataloger og rettigheter på datamaskinene i domenet. Ved hjelp av Powershell kan vi hente ut denne informasjonen og også endre eller legge til informasjon i AD. Mens det i tidligere versjoner av Windows Server kun var mulig å bruke et spesielt "[ADSI]"-objekt for å aksessere Active Directory, har Windows Server 2008r2 med hyper-v bygd inn en egen modul kalt "Active Directory". Vi skal nå se på hvordan vi kan benytte oss av denne modulen for å legge inn brukere både manuelt og ved utlisting fra et Microsoft Excel regneark.

Modulen må bestandig importeres med "import-module ActiveDirectory". Ved å bruke "get-command -module ActiveDirectory" kan vi så utliste en oversikt over tilgjengelige kommandoer mot AD, som vist i Figur 319.

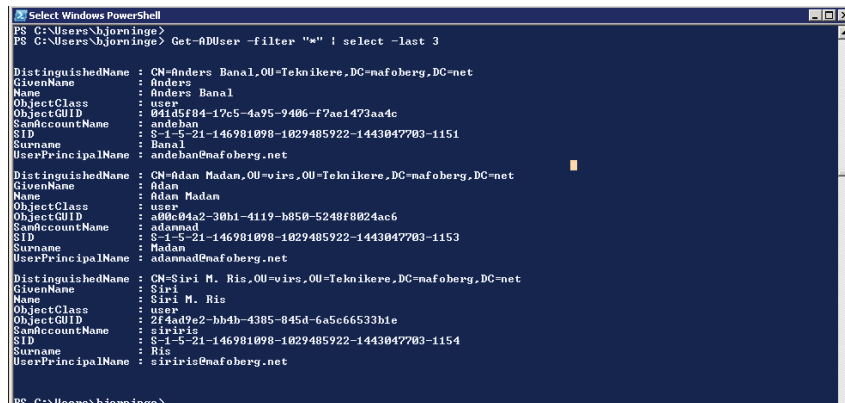


Figur 319 - Import og utforskning av kommandoene i AD-modulen

Det som en må merke seg med denne modulen, er at den kun opererer på AD-treet som du finner på ditt eget domene, slik at du må bruke kommandoen "get-addomain" om du vil endre dette. Du vil også få feilmeldinger ved import av modulen, dersom du ikke har den installert. Det er bare servere med Active Directory og Powershell 2.0 som vil ha denne modulen installert som standard. "Remote Server Administration Tools" fra Microsoft kan også installeres på en arbeidsstasjon, slik at du får tilgang til den samme modulen uten at du må være innlogga på AD-serveren.

4.5.1 Uthenting av brukere og OUer

Ved å bruke kommandoen ”get-aduser”, kan vi nå hente ut alle brukerkontoer som ligger på domenet.



```
PS C:\Users\bjorninge> Get-ADUser -filter "*" | select -last 3

DistinguishedName : CN=Anders Banal,OU=Teknikere,DC=nafoberg,DC=net
GivenName         : Anders
Name              : Anders Banal
ObjectClass       : user
ObjectGUID        : 041d5f84-17c5-4a95-9406-f7ae1473aa4c
SamAccountName    : andeban
SID               : S-1-5-21-146981098-1029485922-1443047703-1151
Surname           : Banal
UserPrincipalName : andeban@nafoberg.net

DistinguishedName : CN=Adam Madan,OU=virs,OU=Teknikere,DC=nafoberg,DC=net
GivenName         : Adam
Name              : Adam Madan
ObjectClass       : user
ObjectGUID        : a00c04a2-30b1-4119-b850-5248f8024ac6
SamAccountName    : adanmad
SID               : S-1-5-21-146981098-1029485922-1443047703-1153
Surname           : Madan
UserPrincipalName : adanmad@nafoberg.net

DistinguishedName : CN=Siri M. Ris,OU=virs,OU=Teknikere,DC=nafoberg,DC=net
GivenName         : Siri
Name              : Siri M. Ris
ObjectClass       : user
ObjectGUID        : 2f4e49e2-bb4b-4385-845d-6a5c66533b1e
SamAccountName    : siriris
SID               : S-1-5-21-146981098-1029485922-1443047703-1154
Surname           : Ris
UserPrincipalName : siriris@nafoberg.net

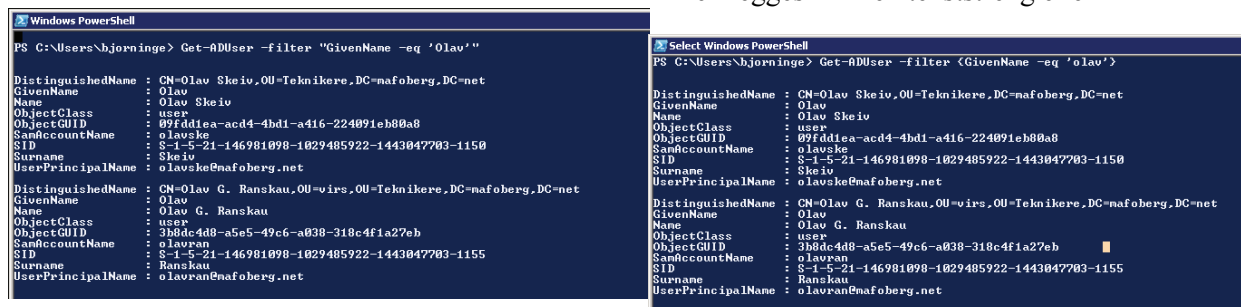
PS C:\Users\bjorninge>
```

Til dette benytter vi et såkalt filter, hvor vi spesifiser ei stjerne for å hente ut alle brukerkontoer. Dette er et såkalt wildcard²². Kommandoen i Figur 320 viser de tre siste brukerne i AD.

Figur 320 - Bruk av wildcard-filtre

Her ser vi at utlistinga er rekursiv, det vil si at alle brukere, uansett OU og underliggende OUer vil listes ut. Vi ser at brukerne Adam og Siri her ligger i OUen ”virs” som igjen ligger under OUen ”teknikere”, mens brukeren Anders ligger direkte under OUen ”teknikere”.

På utlistinga overfor, ser vi også at en del informasjon er tilgjengelig i de forskjellige egenskapene, deriblant er fornavnet utlistet i GivenName. Vi kan videre søke opp brukere ved å bruke et nytt filter. Slike filtre bruker en syntaks som likner veldig mye på vanlig Powershell filtre (”Where-Object”) men legges inn i en tekststreng eller



```
PS C:\Users\bjorninge> Get-ADUser -filter "GivenName -eq 'Olav'"

DistinguishedName : CN=Olav Skeiv,OU=Teknikere,DC=nafoberg,DC=net
GivenName         : Olav
Name              : Olav Skeiv
ObjectClass       : user
ObjectGUID        : 09fdd1ea-acd4-4bd1-a416-224091eb80a8
SamAccountName    : olavske
SID               : S-1-5-21-146981098-1029485922-1443047703-1150
Surname           : Skeiv
UserPrincipalName : olavske@nafoberg.net

DistinguishedName : CN=Olav G. Ranskau,OU=virs,OU=Teknikere,DC=nafoberg,DC=net
GivenName         : Olav
Name              : Olav G. Ranskau
ObjectClass       : user
ObjectGUID        : 3b8dc4d8-a5e5-49c6-a038-318c4f1a27eb
SamAccountName    : olavran
SID               : S-1-5-21-146981098-1029485922-1443047703-1155
Surname           : Ranskau
UserPrincipalName : olavran@nafoberg.net

PS C:\Users\bjorninge> Get-ADUser -filter (GivenName -eq 'olav')

DistinguishedName : CN=Olav Skeiv,OU=Teknikere,DC=nafoberg,DC=net
GivenName         : Olav
Name              : Olav Skeiv
ObjectClass       : user
ObjectGUID        : 09fdd1ea-acd4-4bd1-a416-224091eb80a8
SamAccountName    : olavske
SID               : S-1-5-21-146981098-1029485922-1443047703-1150
Surname           : Skeiv
UserPrincipalName : olavske@nafoberg.net

DistinguishedName : CN=Olav G. Ranskau,OU=virs,OU=Teknikere,DC=nafoberg,DC=net
GivenName         : Olav
Name              : Olav G. Ranskau
ObjectClass       : user
ObjectGUID        : 3b8dc4d8-a5e5-49c6-a038-318c4f1a27eb
SamAccountName    : olavran
SID               : S-1-5-21-146981098-1029485922-1443047703-1155
Surname           : Ranskau
UserPrincipalName : olavran@nafoberg.net
```

krøllparenteser til ”-filter”-argumentet i stedet.

Figur 321 – To AD-søk basert på fornavn

Overfor ser vi altså to måter å søke etter brukere på. Vi kunne valgt å søke basert på ett eller flere av de tilgjengelige egenskapene, slik som surname eller name, men også etter epostadresse eller gateadresse. Resultatene overfor er identiske, men når du benytter krøllparenteser i stedet for en tekststreng, får du også muligheten til å referere til variabelnavn. Vi kunne derfor valgt å søke etter en bruker gitt fra en prompt ved å kjøre:

```
$user = Read-host
```

```
Get-ADUser -filter { GivenName -eq $user }
```

I Figur 322 ser vi begrensning av søkeområdet. Vi har brukt samme søkekriterie som i sted, men vi har lagt inn en begrensning, slik at bare de brukere som befinner seg i virs OUen listes ut. Her får vi altså ett resultat i stedet for to som i forrige resultat.

²² Wildcard er en betegnelse på et tegn som fungerer som et substitutt. Her er det brukt som et kriterium for å søke etter alle brukere innenfor organisjonen (domenet) vårt.

```

Select Windows PowerShell
PS C:\Users\bjorninge> Get-ADUser -filter "GivenName -eq 'Olav'" -SearchBase "OU=virs,OU=Teknikere,DC=mafoberg,DC=net"

DistinguishedName : CN=Olav G. Ranskau,OU=virs,OU=Teknikere,DC=mafoberg,DC=net
GivenName         : Olav
Name              : Olav G. Ranskau
ObjectClass       : user
ObjectGUID        : 3b8dc4d8-a5e5-49c6-a038-318c4f1a27eb
SamAccountName    : olavran
SID               : S-1-5-21-146981098-1029485922-1443047703-1155
Surname           : Ranskau
UserPrincipalName : olavran@mafoberg.net

```

Figur 322 - Begrensing av søkebase

Det er også mulig å hente ut datamaskinobjekter på samme måte, men da ved å bruke "Get-ADComputer". Dersom du skal manipulere en OU, kan vi hente ut et OU-objekt ved å bruke "Get-ADOrganizationalUnit". Dette OU-objektet kan vi så bruke direkte som en verdi til argumentet "-Searchbase". I Figur 323 viser vi hvordan vi kan søke etter brukere med et forhåndslagt OU-objekt i variabelen \$dn.

```

Administrator: Windows PowerShell
PS C:\Windows\system32> $dn = Get-ADOrganizationalUnit -Filter "Name -like 'special users'"
PS C:\Windows\system32> $dn

City              :
Country           :
DistinguishedName : OU=special users,DC=mafoberg,DC=net
LinkedGroupPolicyObjects : {}
ManagedBy        :
Name              : special users
ObjectClass       : organizationalUnit
ObjectGUID        : e2353f1c-1d46-4439-a508-7c38600c5cef
PostalCode        :
State             :
StreetAddress      :

PS C:\Windows\system32> Get-ADUser -filter "*" -SearchBase $dn

DistinguishedName : CN=bjorninge bjorninge,OU=special users,DC=mafoberg,DC=net
Enabled           : True
GivenName         : bjorninge
Name              : bjorninge bjorninge
ObjectClass       : user
ObjectGUID        : 737897a2-93f6-4ce0-a7fa-e2113ee18fae
SamAccountName    : bjorninge
SID               : S-1-5-21-146981098-1029485922-1443047703-1110
Surname           : bjorninge
UserPrincipalName : bjorninge@mafoberg.net

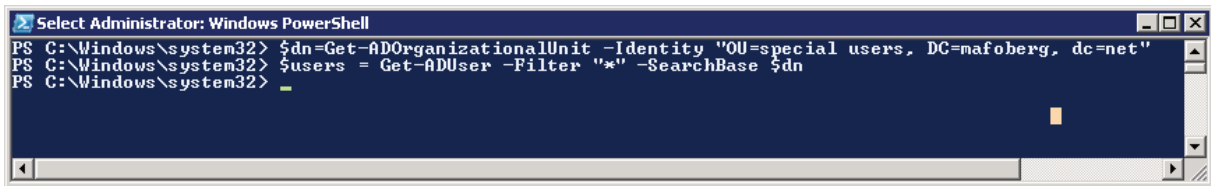
DistinguishedName : CN=mafo,OU=special users,DC=mafoberg,DC=net
Enabled           : True
GivenName         : mafo
Name              : mafo
ObjectClass       : user
ObjectGUID        : 751378a5-116b-45d3-947a-52ab6cd92e5d
SamAccountName    : mafo
SID               : S-1-5-21-146981098-1029485922-1443047703-1113
Surname           :
UserPrincipalName : mafo@mafoberg.net

```

Figur 323 - Brukere i en forhåndslagt OU

4.5.2 Utlisting og redigering av brukerinformasjon.

Når vi nå har hentet inn brukerkontoer med "get-aduser", kan vi så gjøre endringer på en eller flere av disse brukerne. Det lønner seg å lagre resultatet av "get-aduser" i en variabel, slik at vi ikke må søke opp brukerne på nytt for hver gang vi skal endre attributer på brukerne.

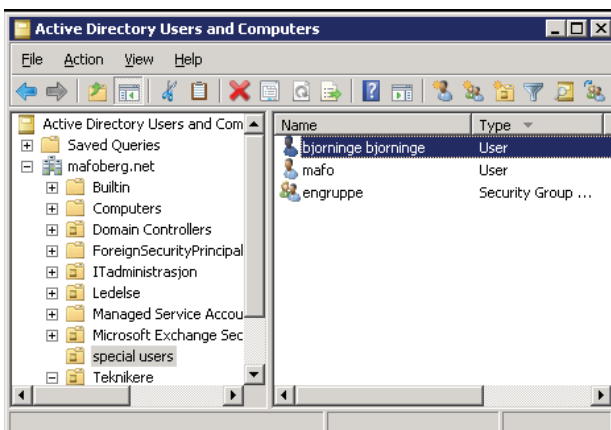


Figur 324 - Lagre brukerkontoer

OU-en "Special users" inneholder i dette tilfellet to brukere, bjørninge og mafo. Vi kan se dette ved å hente ut her brukers givenname, som vist i Figur 325:



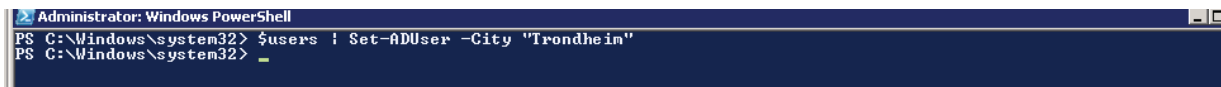
Figur 325 - Utlisting av egennavn



Figur 326 - AD-utlisting

Tilsvarende kan vi gå direkte inn i administrasjonsgrensesnittet til AD, for å bekrefte at de samme brukerne eksisterer når vi bruker det grafiske grensesnittet.

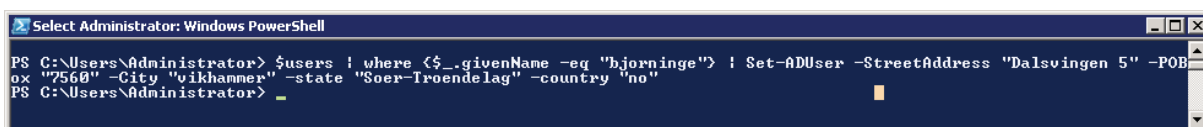
Vi kan nå endre flere egenskaper til alle eller enkelte av disse brukerne ved å benytte oss av kommandoen `set-aduser`. Denne brukes oftest i samarbeid med en pipe der vi har et resultat fra `get-aduser` på venstre side, og `set-aduser` på høyre side:



Figur 327 - Setting av brukerinformasjon

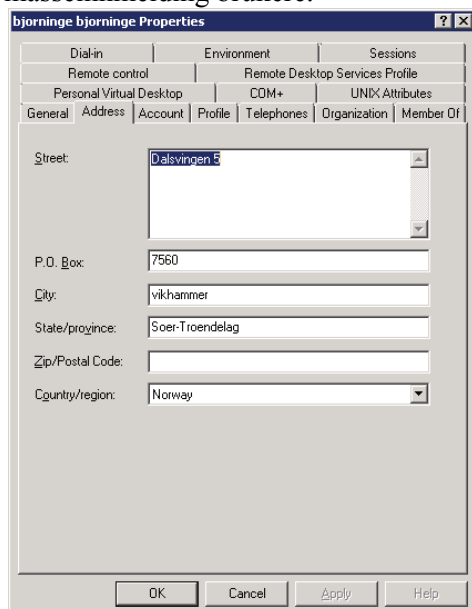
Kommandoen over setter hver brukers by. Brukerne antas allerede å eksistere i variabelen `$users`, slik som sett i Figur 324. Merk at for å sette disse verdiene, kreves det administratorrettigheter. Får du feilmeldingen "Insufficient access rights to perform the operation" betyr det at du må starte Powershell ved å høyreklikke powershellikonet og velge "Run as administrator". Dette gjelder selv om du i utgangspunktet har domeneadministratorrettigheter. Den samme feilmeldinga kan for øvrig komme selv om du har fulle rettigheter; dette vil som regel bare være et problem om Windows ikke er aktivert.

`Set-aduser` kan også brukes til å sette andre verdier slik som land, gateadresse, epostadresse, profilsti, hjemmemappe og omtrent alle verdier som en kan se i det grafiske grensesnittet til AD.



Figur 328 - Setting av mer brukerinformasjon

Overfor har vi brukt et reelt eksempel hvor vi ønsker å sette flere verdier samtidig. En kan også tenke seg at denne informasjonen mottas og settes automatisk, for eksempel i forbindelse med masseinnmelding brukere.

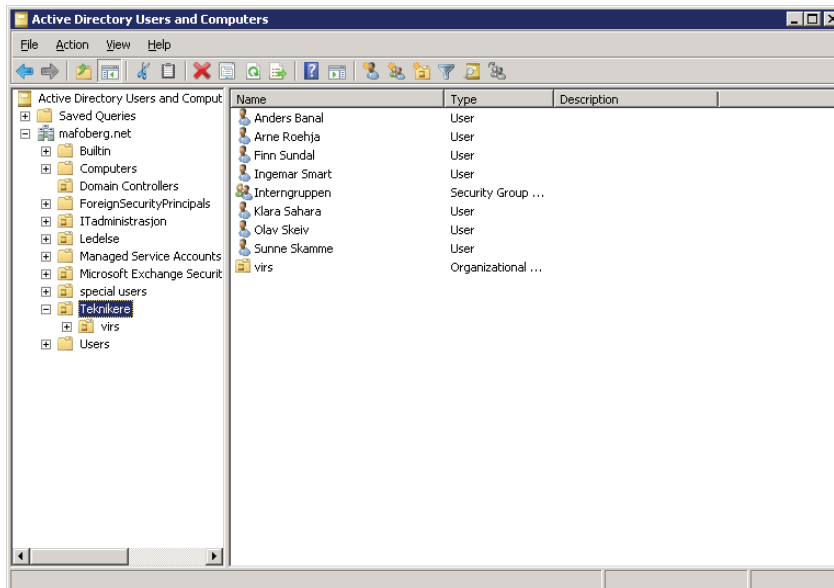


Figur 329 - Utlisting av personinformasjon i AD

Når vi går inn i det grafiske grensesnittet, får vi en bekreftelse på at informasjonen som Powershell satte, faktisk er tilgjengelig i AD.

4.5.3 Opprette OUer og grupper

Ser vi i organisasjonstreet i AD, er det allerede opprettet en struktur for å holde brukere og datamaskiner. Mens OUer oftest fungerer som en geografisk container for avdelinger innad i bedriften, brukes lokale og globale grupper for å gruppere brukere samt filrettigheter på servere. Vi skal nå se hvordan vi oppretter både OUer og grupper ved hjelp av Powershell.



Dette er utgangspunktet vårt; vi har en serie med brukere under "OU=teknikere, DC=mafoberg, DC=net" som vi ønsker å flytte til en ny OU. Vi har allerede opprettet en OU kalt "virs", men denne har allerede brukere. Vi skal i stedet opprette en ny OU kalt "swteknikere" og flytte alle brukerne dit. OUen virs skal ikke røres.

Figur 330 - OU-en teknikere

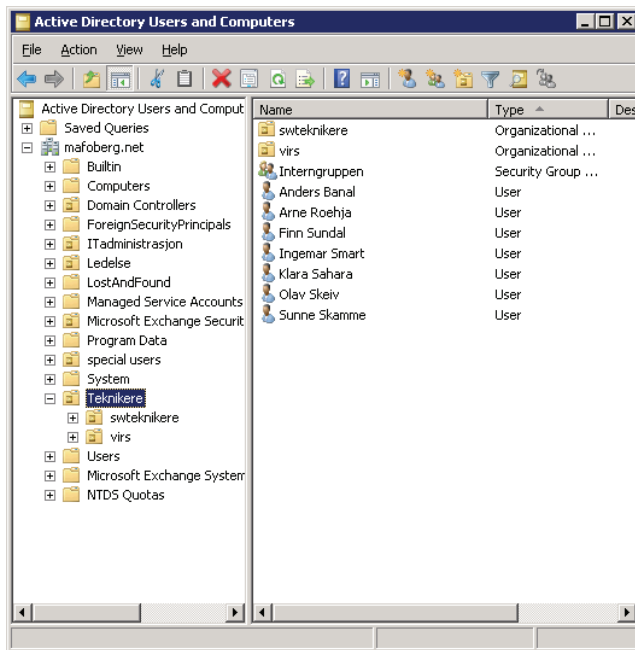
Det første vi da må gjøre, er å velge alle brukerne. Vi

bruker samme filter som før for å liste ut alle brukerne under teknikere-OUen, men gir også ved en parameter `-searchscope` som vi setter til "1":

```
$users = Get-ADUser -filter "*" -Searchbase "OU=teknikere, DC=mafoberg, DC=net" -searchscope 1
```

Dette gjør at alle brukere men kun i nåværende mappe lagres i \$users-variabelen. Standardverdien til `-searchscope` er 2, slik at den inkluderer underOUer slik som "virs". Dette er ikke ønskelig i vår situasjon fordi vi skal la brukerne under "virs" være i fred. Før vi flytter på brukerne må vi opprette en ny OU. Her bruker vi kommandoen `new-ADOrganizationalUnit`, slik:

```
New-ADOrganizationalUnit -name "swteknikere" -path "OU=teknikere, DC=mafoberg, DC=net"
```

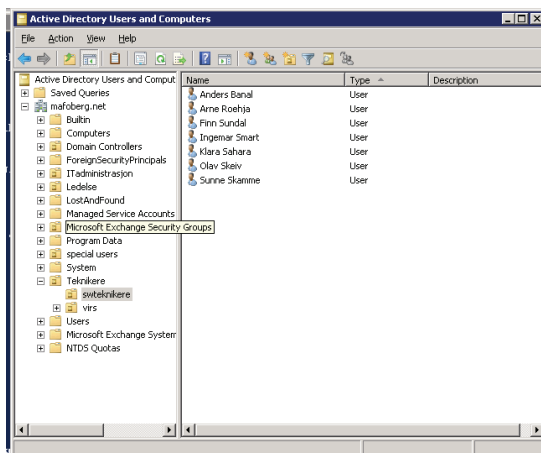


Figur 331 - Opprettet OU i AD

Om du nå går inn i det grafiske grensesnittet til AD, kan du se at det har blitt oppdatert slik at en ny OU kalt "swteknikere" eksisterer.

Ved å bruke Move-ADObject kan en nå flytte disse brukerne til den nye OU-en. Merk at denne kommandoen også kan brukes til å flytte andre ting, slik som grupper og hele OU-er

```
$users | Move-ADObject -targetpath "OU=swteknikere, OU=teknikere, DC=mafoberg, DC=net"
```



Figur 332 - Brukere flyttet

Vi ser nå at brukerne havnet under swteknikere-OUen

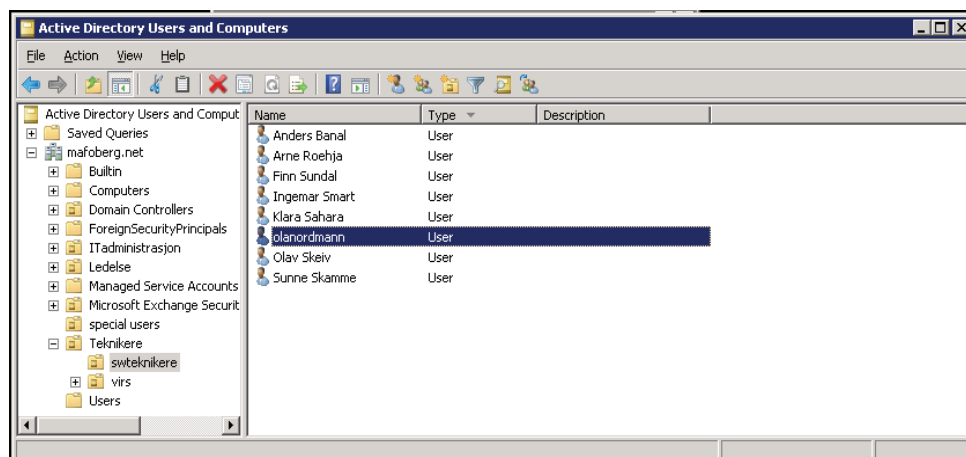
4.5.4 Innlegging av brukere

Kommandoen `new-aduser` kan brukes til å lage nye brukere en plass i AD-treet. For å aktivere én bruker, må vi i tillegg lese inn et passord fra kommandolinja og ”hashe”²³ det for så å lagre passordet med brukerkontoen. Kommandoen gitt i Figur 333 må skrives sammenhengende på én linje.

```
1 New-ADUser -Name "olanordmann" -SamAccountName "olanordmann"  
2 -GivenName "Ola" -Surname "Nordmann" -DisplayName "Ola Nordmann"  
3 -Path 'OU=swteknikere, OU=teknikere, DC=mafoberg, DC=net'  
4 -AccountPassword (Read-Host -AsSecureString "AccountPassword") -Enabled $true
```

Figur 333 - Innlegging av enkeltbruker

Active Directory skal nå oppdateres umiddelbart, og vi ser at vår bruker har blitt lagt til også i det grafiske administrasjonsgrensesnittet.



Figur 334 - AD-utlisting av innlagt bruker

4.5.4.1 ConvertTo-SecureString

Denne måten å legge til en enkeltbruker er noe tungvint, fordi det går treigere enn å bruke administrasjonsgrensesnittet i AD, og fordi vi bare legger til én enkelt bruker. I tillegg måtte vi spesifisere passordet til brukeren ved hjelp av ”read-host”. Denne kommandoen leser inn tekst fra kommandolinja og gir den tilbake til ”new-aduser”-kommandoen vår. Dette går greit for én bruker, men det kan bli særdeles plagsomt å måtte taste inn et passord for hver bruker en ønsker å legge inn. En kan tenke seg en skolesituasjon, hvor det skal legges inn brukere og passord fra en lang liste. Da er det også upraktisk å bruke denne framgangsmåten med å bruke ”read-host” fordi passordet ofte ligger sammen med brukernavnet.

Vi benytter derfor oss av kommandoen ”ConvertTo-SecureString” for å slippe dette, og lagrer passordet i rein tekst i stedet. Et eksempel på dette følger. Under har vi opprettet en ny bruker, uten at vi måtte taste inn noe mer i Powershell-konsollet. Det er i utgangspunktet ikke helt sikkert å konvertere et passord til en sikker variant fordi det først må lagres i minnet før det kan konverteres. Dette minnet er det mulig å ”avlytte” på flere måter²⁴, så for å få lov til å bruke `convertTo-SecureString` må en spesifisere både parameterne ”-asplaintext” og ”-force”.

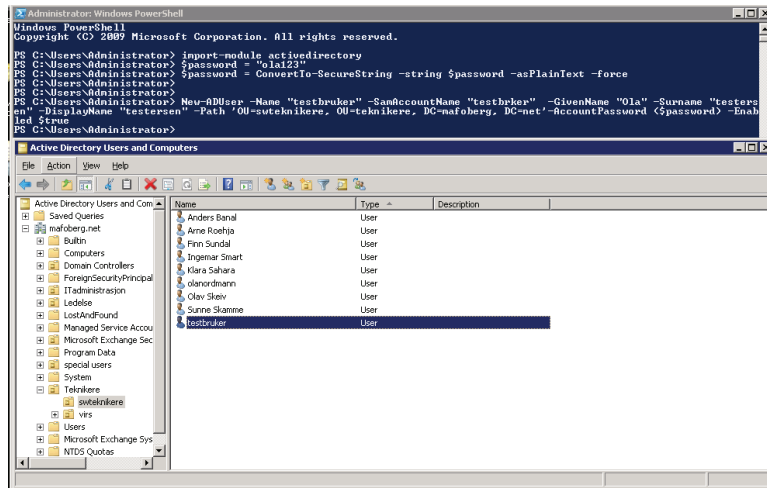
²³ Hashing er en form for enveiskryptering. Det som skjer her er at en unik tekststreng genereres basert på passordet som skrives inn. Dette hindrer at passordet kan leses direkte fra databasen i rein tekst

²⁴ Dersom du er redd for at minnet ditt kan avlyttes, har du et stort sikkerhetsproblem, og du burde ikke kjøre disse powershellkommandoene i det heletatt.


```
$password = "ola123"
```

```
$password = ConvertTo-SecureString -string $password -  
asPlainText -force
```

Når vi nå legger til en ny bruker bruker vi samme kode som før.



I Figur 330 til venstre ser vi kommandoen kjørt og resultatet i administrasjonsgrensesnittet til AD. Den eneste forskjellen i koden er at vi har valgt et annet navn ("testbruker"), samt at vi har byttet ut kallet til "read-host" med en referanse til passordet vårt, lagret i \$password .

Figur 335 - Innlegging av bruker vha. forhåndslagt passord

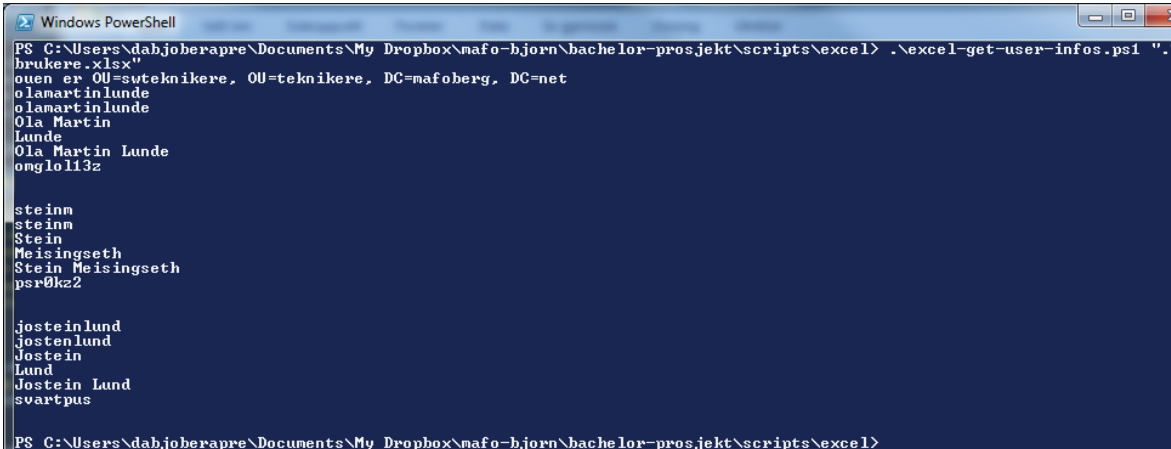
4.5.5 Automatisere utlesing og innlegging av flere brukere

Vi skal nå legge inn flere brukere i en automatisert prosess. Vi skal benytte oss av forhåndslagrede passord i et Excel regneark. Det er flere mulige måter å lese inn denne informasjonen fra, hvor den mest vanlige er å lagre Excel-dokumentet som en ”.csv”-fil²⁵ for så å lese denne fila med Powershell ved hjelp av ”import-csv”. Dette krever da at du har Excel installert og at du selv lagrer dokumentet ditt som en csv-fil. Når en likevel har Excel installert, kan en gjøre dette på en annen måte, ved å opprette et COM-objekt av et Excel-ark. På denne måten kan vi lese ut informasjon direkte fra regnearket ved hjelp av Powershell, uten å måtte gå omveien om å lagre dokumentet vårt som en csv-fil.

	A	B	C	D	E	F
1	OU:	OU=swteknikere, OU=teknikere, DC=mafoberg, DC=net				
2						
3	Name	SamAccountName	GivenName	SurName	DisplayName	Password
4	olamartinlunde	olamartinlunde	Ola Martin	Lunde	Ola Martin Lunde	omglol13z
5	steinm	steinm	Stein	Meisingseth	Stein Meisingseth	psr0kz2
6	josteinlund	jostenlund	Jostein	Lund	Jostein Lund	svartpus
7						
8						

Figur 336 - Brukerlistestruktur i Excel, fra eksempelfila ”brukere.xls”

Overfor er et eksempel på hvordan strukturen bør se ut for at vi skal kunne hente ut informasjonen vi trenger for å generere brukere. Vi har videre laget to script for å håndtere denne informasjonen, nemlig ”excel-get-user-infos.ps1” og ”excel-add-users-to-ad.ps1”. OU som brukerne skal legges inn i, må finnes i celle b1. Dersom du skriver inn et nytt navn i navnefeltet, må du i tillegg fylle ut alle de andre feltene.



```

PS C:\Users\dahjoherapre\Documents\My Dropbox\mafo-hjorn\bachelor-prosjekt\scripts\excel> .\excel-get-user-infos.ps1 ".\brukere.xlsx"
ouen er OU=swteknikere, OU=teknikere, DC=mafoberg, DC=net
olamartinlunde
olamartinlunde
Ola Martin
Lunde
Ola Martin Lunde
omglol13z

steinn
steinn
Stein
Meisingseth
Stein Meisingseth
psr0kz2

josteinlund
jostenlund
Jostein
Lund
Jostein Lund
svartpus
PS C:\Users\dahjoherapre\Documents\My Dropbox\mafo-hjorn\bachelor-prosjekt\scripts\excel>

```

Figur 337 - Bruk av excel-get-user-infos.ps1

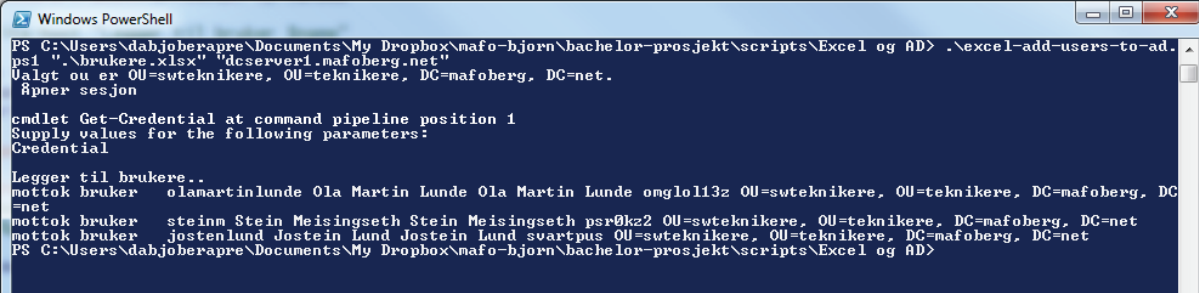
Overfor ser vi eksempel på bruk av scriptet vårt. Det skriver ut innholdet av Excel-dokumentet fortløpende nedover. Dersom informasjonen vises korrekt i dette scriptet, vil det også fungere med neste script, da de benytter seg av samme framgangsmåte for å hente ut data.

4.5.5.1 Innlegging av brukerne

Når en har fått verifisert at det forrige scriptet greier å hente ut informasjonen i Excel-arket, kan en benytte seg av ”excel-add-users-to-ad.ps1” for å legge disse brukerne inn i AD. Dette scriptet må

²⁵ CSV-filer(Comma-Separated Values) er rene tekstdokumenter, der hver linje representerer en rad i en tabell. Denne rada er så separert med kommategn for å spesifisere verdiene i kolonner.

kjøres med administrasjonsrettigheter, samt på en arbeidsstasjon eller server som har Microsoft Excel installert.



```
PS C:\Users\dabjoherapre\Documents\My Dropbox\mafo-hjorn\bachelor-prosjekt\scripts\Excel og AD> .\excel-add-users-to-ad.ps1 ".\brukere.xlsx" "dcserver1.mafoberg.net"
Valgt ou er OU=svteknikere, OU=teknikere, DC=mafoberg, DC=net.
Åpner sesjon

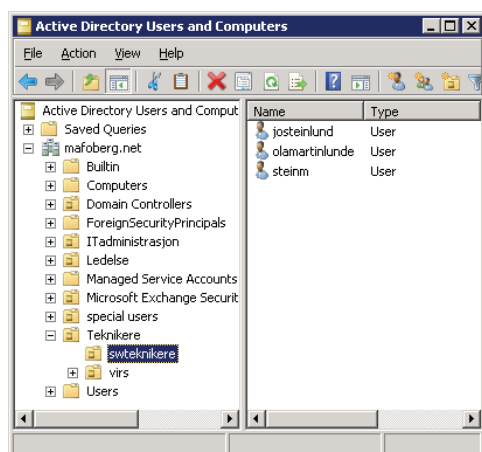
cmdlet Get-Credential at command pipeline position 1
Supply values for the following parameters:
Credential

Legger til brukere..
mottok bruker olamartinlunde Ola Martin Lunde Ola Martin Lunde onglol13z OU=svteknikere, OU=teknikere, DC=mafoberg, DC=net
mottok bruker steinn Stein Meisingseth Stein Meisingseth psr0kz2 OU=svteknikere, OU=teknikere, DC=mafoberg, DC=net
mottok bruker jostenlund Jostein Lund Jostein Lund svartpus OU=svteknikere, OU=teknikere, DC=mafoberg, DC=net
PS C:\Users\dabjoherapre\Documents\My Dropbox\mafo-hjorn\bachelor-prosjekt\scripts\Excel og AD>
```

Figur 338 - Innlegging av brukere fra Excel-arket

Videre benytter scriptet seg av Powershell Remoting for å koble til en Active Directory Server for å legge til brukerne. Merk at den lokale arbeidsstasjonen som er brukt her ikke trenger å være medlem av domenet, da scriptet vil spørre deg om å taste inn brukernavn og passord for å koble til. Vi legger merke til at scriptet bruker samme Excel-ark som tidligere som første parameter, og en Active

Directory Server som siste parameter (her: "dcserver1.mafoberg.net").



Figur 339 - Resultat av masseinnlegging

Til venstre ser vi resultatet i administrasjonsgrensesnittet til AD, etter at Powershell-kommandoen overfor har kjørt. Det skal nå også være mulig å logge seg inn på domenet med det brukernavnet og tilhørende passord som satt i regnearket.

Dette scriptet antar at brukere med ikke allerede eksisterer med samme brukernavn. Dersom dette ikke er tilfelle, vil det bli gitt en feilmelding for hver bruker som det ikke er mulig å legge til.

4.5.6 Sjeldent innloggede brukere

I Powershell er det enkelt å finne brukere som ikke har vært innlogget på serveren de siste dagene. Ved å bruke en kombinasjon av "get-date" og "new-timespan".

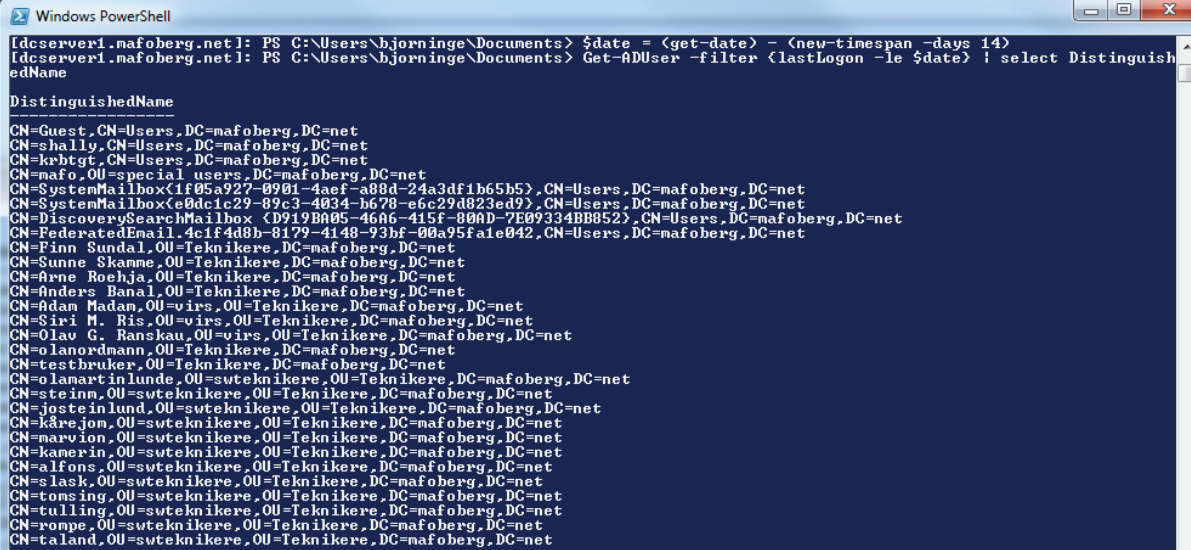
Ved å kombinere disse kommandoene, kan vi nå lagre dagens dato minus 14 dager slik:

```
$date = (get-date) - (new-timespan -days 14)
```

Vi kan videre benytte denne datoen i et søkefilter til "get-aduser", slik:

```
Get-ADUser -filter {lastLogon -le $date} | select DistinguishedName
```

Her bruker vi ei kodeblokk hvor lastLogon referere til siste påloggingsdato for hver bruker. Videre velger vi de brukerne som har logget inn sist for mer enn 14 dager siden. DistinguishedName gir den fulle unike stien til hver bruker.



```
Windows PowerShell
[dcserver1.mafoberg.net]: PS C:\Users\bjorninge\Documents> $date = (get-date) - (new-timespan -days 14)
[dcserver1.mafoberg.net]: PS C:\Users\bjorninge\Documents> Get-ADUser -filter {lastLogon -le $date} | select DistinguishedName

DistinguishedName
-----
CN=Guest,CN=Users,DC=mafoberg,DC=net
CN=shally,CN=Users,DC=mafoberg,DC=net
CN=krbtgt,CN=Users,DC=mafoberg,DC=net
CN=mafo.ou=special_users,DC=mafoberg,DC=net
CN=SystemMailbox{f1f85a922-0901-4aef-a88d-24a3df1b65b5},CN=Users,DC=mafoberg,DC=net
CN=SystemMailbox{e00c1c29-09c3-4834-b678-e6c29d023ed9},CN=Users,DC=mafoberg,DC=net
CN=DiscoverySearchMailbox {D919B005-46A6-415f-80AD-7E09334BB852},CN=Users,DC=mafoberg,DC=net
CN=FederatedEmail.4c1f4d8b-8179-4148-93bf-00a95fa1e042,CN=Users,DC=mafoberg,DC=net
CN=Finn Sundal,OU=Teknikere,DC=mafoberg,DC=net
CN=Sunne Skamme,OU=Teknikere,DC=mafoberg,DC=net
CN=Arne Roehja,OU=Teknikere,DC=mafoberg,DC=net
CN=Anders Banal,OU=Teknikere,DC=mafoberg,DC=net
CN=Adan Madan,OU=virs,OU=Teknikere,DC=mafoberg,DC=net
CN=Siri M. Ris,OU=virs,OU=Teknikere,DC=mafoberg,DC=net
CN=Olav G. Ranskau,OU=virs,OU=Teknikere,DC=mafoberg,DC=net
CN=olanordmann,OU=Teknikere,DC=mafoberg,DC=net
CN=tostbraker,OU=Teknikere,DC=mafoberg,DC=net
CN=olanartinlunde,OU=suteknikere,OU=Teknikere,DC=mafoberg,DC=net
CN=steinna,OU=suteknikere,OU=Teknikere,DC=mafoberg,DC=net
CN=josteinlund,OU=suteknikere,OU=Teknikere,DC=mafoberg,DC=net
CN=kårejon,OU=suteknikere,OU=Teknikere,DC=mafoberg,DC=net
CN=marvion,OU=suteknikere,OU=Teknikere,DC=mafoberg,DC=net
CN=kamerin,OU=suteknikere,OU=Teknikere,DC=mafoberg,DC=net
CN=alfons,OU=suteknikere,OU=Teknikere,DC=mafoberg,DC=net
CN=silask,OU=suteknikere,OU=Teknikere,DC=mafoberg,DC=net
CN=tomsing,OU=suteknikere,OU=Teknikere,DC=mafoberg,DC=net
CN=tulling,OU=suteknikere,OU=Teknikere,DC=mafoberg,DC=net
CN=ronpe,OU=suteknikere,OU=Teknikere,DC=mafoberg,DC=net
CN=taland,OU=suteknikere,OU=Teknikere,DC=mafoberg,DC=net
```

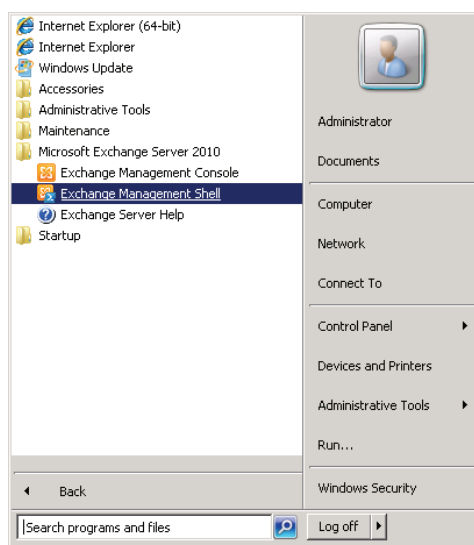
Figur 340 - Resultat av siste kommando

4.6 Powershelladministrasjon av Exchange

I Microsoft Exchange 2010 er det flere muligheter for styring og konfigurering av mailbokser, databaser og brukere via PowerShell. Siden Exchange 2007 er det utviklet enkle cmdlets som gjør effektiv styring av mailbokser og brukere enklere når man skal gjøre konfigurasjon på mange objekter samtidig.

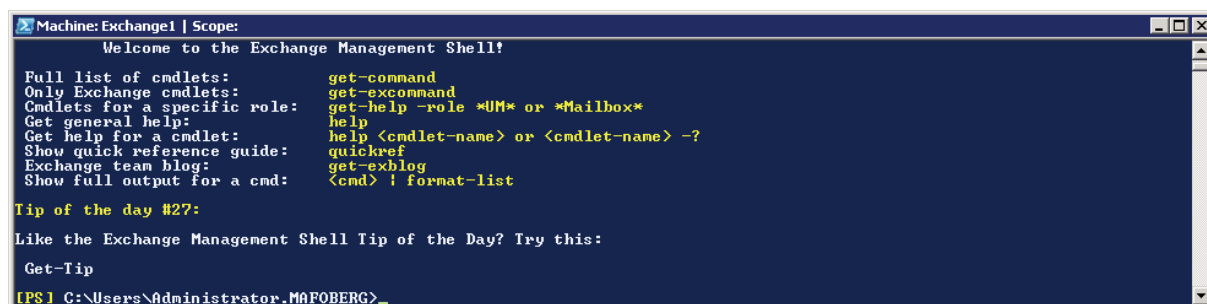
Kommandogrensesnittet heter Exchange Management Shell (EMS) og inneholder en rekke ekstra cmdlets for PowerShell. Vi skal ta for oss de to enkleste måtene å starte disse på.

Den ene måten er snarveien som følger med Exchange 2010 når den blir installert. Denne finner man igjen på startmenyen. Denne gjør det enkelt å starte PowerShell med snapins som tilhører Exchange 2010. For å gjøre det enda enklere, kan det lønne seg å høyreklikke på denne snarveien og velge *Pin to Taskbar* eller *Pin to Startmenu*.



Figur 341 - Exchange Management Shell oppstart

En annen måte å gjøre det på er å starte PowerShell selv og så kjøre Exchange oppstart med script selv. Man starter da PowerShell og kjører så kommandoen `.\Program Files\Microsoft\Exchange Server\v14\bin\Exchange.ps1`. Man får da kjørt oppstarten til Exchange kommandosett manuelt.



Figur 342 - Exchange Management Shell PSoppstart

En tredje måte er å legge til snapins til Exchange manuelt. Først kan man finne ut hvilke snapins som er registrert på serveren. Dette gjør man ved å kjøre `Get-PSSnapin -registered`. Da får man listet opp alle snapins med navn, versjonsnummer og beskrivelse.

```
Machine: Exchange1 | Scope:
[PS] C:\Users\Administrator.MAFOBERG>Get-PSSnapin -registered

Name       : Microsoft.Exchange.Management.PowerShell.E2010
PSVersion  : 1.0
Description : Admin Tasks for the Exchange Server

Name       : Microsoft.Exchange.Management.PowerShell.Setup
PSVersion  : 1.0
Description : Setup Tasks for the Exchange Server

Name       : Microsoft.Exchange.Management.PowerShell.Support
PSVersion  : 1.0
Description : Support Tasks for the Exchange Server
```

Figur 343 - Exchange 2010 PSSnapins

Vi ser at det eksisterer flere snapins for Exchange 2010. En for administrasjon, en for oppsett og en for support.

- Microsoft.Exchange.Management.PowerShell.E2010 – Et sett med administrasjons cmdlets for Exchange 2010. Disse kan man liste ut.
- Microsoft.Exchange.Management.PowerShell.Setup – Inneholder oppsettsfunksjoner og registreringer for Exchange 2010.
- Microsoft.Exchange.Management.PowerShell.Support – Inneholder supporttjenester for Exchange 2010. For eksempel valideringstjenester og oppdateringstjenester for moduler i Exchange.

Det er hovedsaklig kommandoene vi nå er interessert i. Disse ligger i E2010-modulen. Denne legger vi til ved å kjøre `Add-PSSnapin Microsoft.Exchange.Management.PowerShell.E2010`.

VI kan så liste ut kommandoene med `Get-Command -PSSnapin`

`Microsoft.Exchange.Management.PowerShell.E2010`. Denne listen blir lang så det kan lønne seg å sende denne til fil ved å legge til `> Exchangekommandoer.txt` bak.

4.6.1. Mailboks og brukerkommandoer

Det finnes mange muligheter for å liste, redigere og endre mailbokser fra powershell. Det som først kan være interessant er å se mailbokser som allerede ligger i Exchange når vi har lagt til noen.

`Get-Mailbox` er kommandoen for å liste eller hente mailbokser. Vi får da opp en liste over alle mailbokser på gjeldende server og gjeldende maildatabase.

```
Machine: Exchange1 | Scope:
[PS] C:\Users\Administrator.MAFOBERG>Get-Mailbox

Name                Alias                ServerName  ProhibitSendQuota
-----
Administrator       Administrator       exchange1  unlimited
Anders Banal         andeban             exchange1  unlimited
Bjørninge Bjørninge bjornib             exchange1  unlimited
DiscoverySearchMailbox... DiscoverySearchMa... exchange1  50 GB (53,687,091,200 bytes)
Klara Sahara         klasah              exchange1  unlimited
Olav Skeiv           olavske             exchange1  unlimited

[PS] C:\Users\Administrator.MAFOBERG>
```

Figur 344 – Get-Mailbox

`Get-Mailuser` er kommandoen for å liste brukere som ligger i kontakter. Disse kontaktene trenger ikke ha sin egen mailboks, men kan befinne seg på andre eksterne servere. Når de ligger i kontaktlisten, blir de tilgjengelige for mailboksbrukere på denne serveren.

```
Machine: Exchange1 | Scope:
[PS] C:\Users\Administrator.MAFOBERG>Get-MailUser

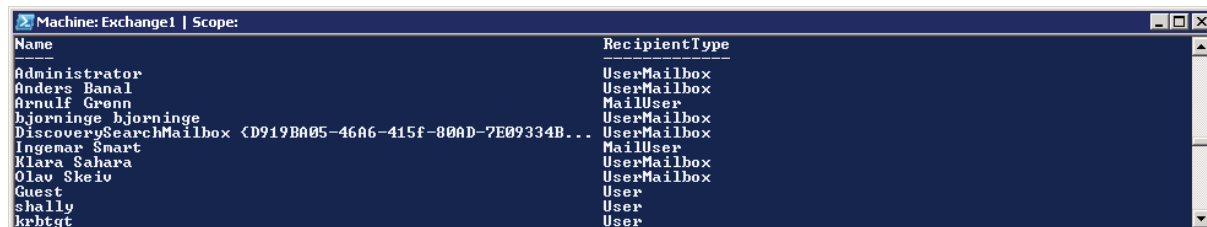
Name                RecipientType
-----
Arnulf Grønn        MailUser
Ingemar Smart       MailUser

[PS] C:\Users\Administrator.MAFOBERG>
```

Figur 345 - Get-MailUser

4.6.2.1. Mailbox brukere

I vår AD kan det være brukere uten mailbokser og brukere med mailbokser. For å få alle brukere i alle ouer, kjører vi `Get-User`.



Name	RecipientType
Administrator	UserMailbox
Anders Banal	UserMailbox
Arnulf Grønn	MailUser
bjørninge bjørninge	UserMailbox
DiscoverySearchMailbox <D919BA05-46A6-415f-80AD-7E09334B...	UserMailbox
Ingemar Smart	MailUser
Klara Sahara	UserMailbox
Olav Skeiv	UserMailbox
Guest	User
shally	User
krbtgt	User

Figur 346 - Get-User

Hvis vi skal plukke ut typen brukere som kun er merket som "User", kan vi spesifisere dette retti kommandoen. Brukere som har typen "User" er verken kontakt eller har mailboks. Vi kjører så kommandoen `Get-User -Recipient User` og får listet kun brukere uten noe tilkobling. Vi har lyst til å legge disse til som mailboksbrukere.

For å legge til en mailboks for en bruker i den ene mailboksdatabase vi har kan vi kjøre dette med en kommando.

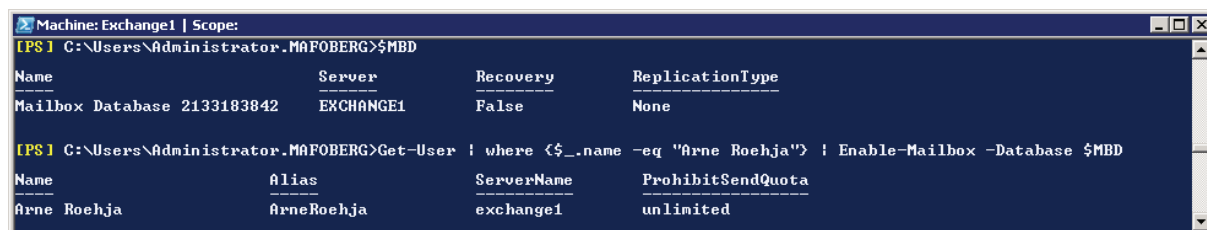
```
Get-User | where {$_.name -eq "Arne Roehja"} | Enable-Mailbox -Database (
Get-MailboxDatabase | Select -first 1 )
```

Vi kan også hente inn databaseobjektet som en variabel først isteden. Dette er en bedre metode å gå frem på. Vi legger først objektet i variabelen.

```
$MBD = Get-MailboxDatabase | Select -first 1
```

Deretter kjører vi kommandoen for å gjøre mailboksen tilgjengelig.

```
Get-User | where {$_.name -eq "Arne Roehja"} | Enable-Mailbox -Database
$MBD
```



Name	Server	Recovery	ReplicationType
Mailbox Database 2133183842	EXCHANGE1	False	None

Name	Alias	ServerName	ProhibitSendQuota
Arne Roehja	ArneRoehja	exchange1	unlimited

Figur 347 - Mailboks for en bruker

Vi har nå opprettet en mailboks for denne brukeren. Vi vil nå gjøre dette for resten av brukerne i vår AD. Vi kjører så denne for alle med "RecipientType" som er "User".

```
Machine: Exchange1 | Scope:
[PS] C:\Users\Administrator.MAFOBERG>Get-User | where {$_.recipienttype -eq "User"} | Enable-Mailbox -Database $MBD
Enable-Mailbox : The user's Active Directory account must be logon-enabled for the user's mailbox.
At line:1 char:64
+ Get-User | where {$_.recipienttype -eq "User"} | Enable-Mailbox <<<< -Database $MBD
+ ~~~~~
+ CategoryInfo          : InvalidArgument: (ADUser) [Enable-Mailbox], RecipientTaskException
+ FullyQualifiedErrorId : 25FBC5EC,Microsoft.Exchange.Management.RecipientTasks.EnableMailbox

Name                Alias                ServerName            ProhibitSendQuota
-----
shally              shally              exchange1             unlimited
Enable-Mailbox : The user's Active Directory account must be logon-enabled for the user's mailbox.
At line:1 char:64
+ Get-User | where {$_.recipienttype -eq "User"} | Enable-Mailbox <<<< -Database $MBD
+ ~~~~~
+ CategoryInfo          : InvalidArgument: (ADUser) [Enable-Mailbox], RecipientTaskException
+ FullyQualifiedErrorId : 25FBC5EC,Microsoft.Exchange.Management.RecipientTasks.EnableMailbox

mafo                mafo                exchange1             unlimited
Finn Sundal        FinnSundal          exchange1             unlimited
Sunne Skanne       SunneSkanne         exchange1             unlimited
Adam Madan         AdamMadan           exchange1             unlimited
Siri M. Ris        SiriM.Ris           exchange1             unlimited
```

Figur 348 - Mailbokser for alle brukere

Når vi kjører for alle "Users" i vår ad, vil alle brukere uten mailboks få det. Vi ser vi får feilmeldinger. Denne feilmeldingen sier at enkelte av brukerne som har recipienttype "User" ikke kan opprettes mailbokser for, fordi de ikke er påloggings-tilgjengelig på domenet. Disse blir bare utelatt.

4.6.2.2. Mailboks kvoter

For å liste kvoter på mailbokser kan vi bruke kommandoen `Get-MailboxStatistics <bruker>`. For å liste alle brukeres statistikk, kan vi kjøre dette i en pipe til `Get-User`. Altså `Get-User | Get-MailboxStatistics`. Vi vil få listet brukeres statistikk mot alle brukere.

Vi vil ha brukere som har fått opprettet mailboks. Vi vil få en gul melding dersom mailboksen er blitt opprettet, men brukeren ikke har logget på enda slik at denne er blitt aktivert.

```
Get-user | where {$_.recipienttype -eq "UserMailbox"} | Get-MailboxStatistics
```

```
Machine: Exchange1 | Scope:
[PS] C:\Users\Administrator.MAFOBERG>Get-user | where {$_.recipienttype -eq "UserMailbox"} | Get-MailboxStatistics

DisplayName          ItemCount    StorageLimitStatus    LastLogonTime
-----
Discovery Search Mailbox 1            BelowLimit            21.04.2010 14:29:53
Administrator        19           BelowLimit            19.04.2010 15:02:03
bjorninge            2            BelowLimit            19.04.2010 15:02:03
mafo                  2            BelowLimit            19.04.2010 15:02:03
Finn Sundal          2            BelowLimit            19.04.2010 15:02:03
WARNING: The user hasn't logged on to mailbox 'mafoberg.net/special users/mafo' ('381173b9-5a85-4d4d-9b93-0080edbdd80'), so there is no data to return. After the user logs on, this warning will no longer appear.
WARNING: The user hasn't logged on to mailbox 'mafoberg.net/Teknikere/suteknikere/Finn Sundal' ('97020d28-bc19-4e88-9f6b-871a666c13ef'), so there is no data to return. After the user logs on, this warning will no longer appear.
```

Figur 349 – Mailboksinformasjon

Det er greit å kunne skille av brukere som har fått mailboks men ikke har aktivert den ved å logge på kontoen enda. Når vi lister brukere med mailboks, vil vi som sagt tidligere få en gul melding for de som har men ikke logget inn enda i tillegg til at brukere med mailbokser blir listet. For å kunne fjerne disse kan vi sette en systemvariabel som heter `$WarningPreference` til å inneholde stringen "SilentlyContinue". Meldinger i gult format vil da ikke vises.

Vi kjører derfor først:

```
$WarningPreference = "SilentlyContinue"
```

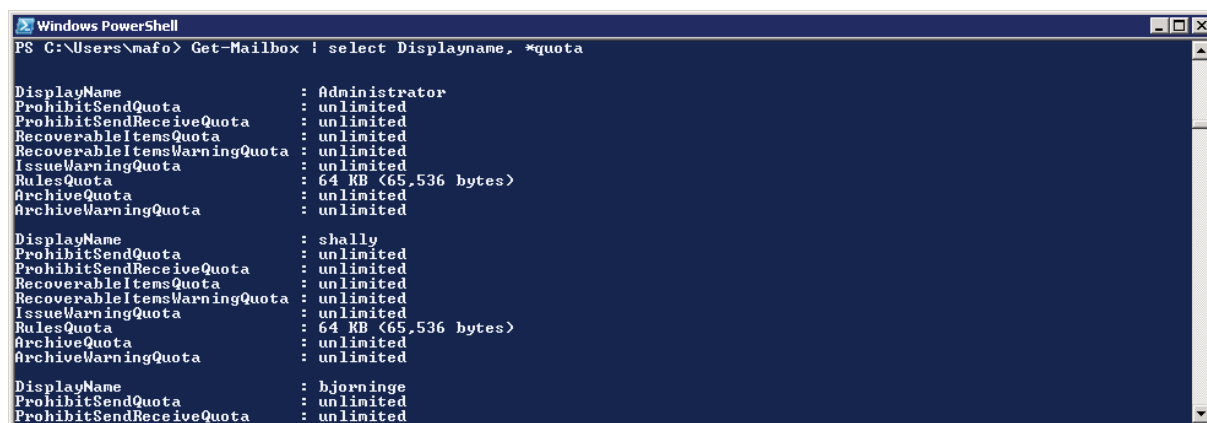
Vi henter så mailbokser til brukerne som ikke har logget på enda. I `where`-setningen plukker vi ut mailbokser som har statistikkstatus som ikke inneholder noe, altså lik verdien `$null`.

```
Get-user -RecipientTypeDetails UserMailBox | where { $(Get-MailboxStatistics $_.DistinguishedName) -eq $Null }
```

Deretter bør vi sette meldingsstatusen tilbake slik at vi kan se gule varsler igjen.


```
$WarningPreference = "Continue"
```

For å finne selve mailbokskvoten, kan vi liste ut alle med kommandoen `Get-Mailbox | select Displayname, *quota`. Med en wildcard `*` får vi med alle attributter som ender med ordet "quota". Da ser vi sendingskvoter, lagringskvoter, regler for kvoter osv.



Figur 350 - Mailboks kvotelisting

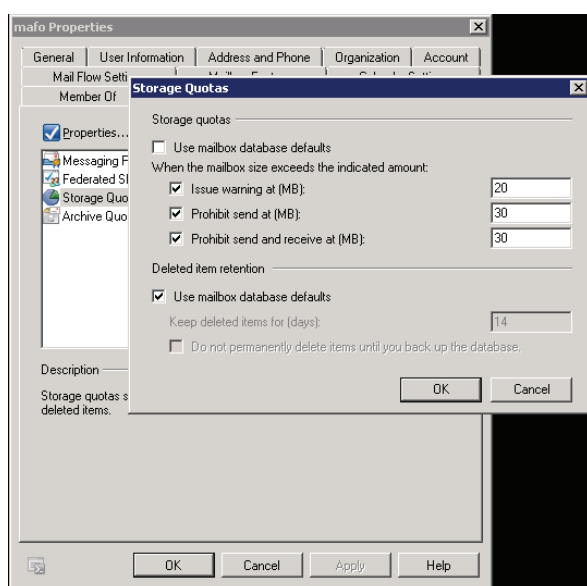
Vi kan ta i bruk disse attributtene når vi skal sette dem til en annen verdi. Vi bruker `Set-Mailbox` til å redigere disse.



Figur 351 - Set-Mailbox

```
Set-Mailbox $mbox -UseDatabaseQuotaDefaults $false -IssueWarningQuota 20MB
-ProhibitSendQuota 30MB -ProhibitSendReceiveQuota 30MB
```

Vi kan sammenlikne denne kommandoen med det som kan settes opp i konsollet til Exchange. Kommandoen vil sette begrensningene akkurat slik som man har tilgang til i Exchange.



Figur 352 - Oppsett av kvoter i konsoll

Vi ser at de to måtene å gjøre det på er forskjellige men utfører akkurat samme handling. Vi kan utnytte dette ved å sette det i et script som setter samme verdi for en hel gruppe mailbokser.

4.6.2. Distribusjonsgrupper.

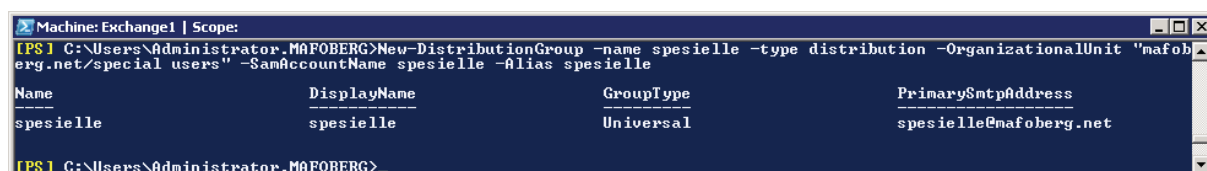
Distribusjonsgrupper er grupper man bruker fra AD hvor flere medlemmer med mailbokser kan få en felles epostadresse. Vi har allerede opprettet en tidligere distribusjonsgruppe som vi kalte *Interngruppen*.

Dynamiske grupper er grupper knyttet direkte til ouer i AD. Det gjør gruppen mer dynamisk siden administrasjon av ouer lettere kan organiseres enn gruppeobjekter.

4.6.2.1. Basert på Gruppeobjekt – Nytt opprettet

Vi lager en distribusjonsgruppe basert på et gruaåpeobjekt som vi oppretter i AD denne gangen. Objektet blir opprettet i OUen *mafoberg.net/special users*. Denne gruppen vil da få en egen type objekt som administreres av Exchange. Vi bruker kommandoen

```
New-DistributionGroup -name spesielle -type distribution -  
OrganizationalUnit "mafoberg.net/special users" -SamAccountName spesielle -  
Alias spesielle
```



Figur 353 - Ny distribusjonsgruppe

Denne gruppen genereres når det mottas en epost.

Vi må så legge til brukere i denne. Vi bruker `Add-DistributiongroupMember`. Vi kan slette dem med `Remove-DistributiongroupMember`.


```
Add-DistributionGroupMember -ID spesielle -Member "mafoberg.net/mafo"
```

Her legger vi til en enkelt medlem i denne gruppen. Ofte skal man ha med en rekke brukere. Da kan det lønne seg å finne et felles trekk for disse. Enten det er alle brukere i en OU, eller brukere fra et gruppeobjekt eller liknende.

4.6.2.2. Basert på gruppeobjekter

Når vi lager en dynamisk distribusjonsgruppe som knyttes mot en OU i AD, må vi legge til filtreringer for hvordan den skal behandle medlemmene. Vi kan for eksempel velge alle medlemmer som tilhører bedriften Mafoberg Systems AS i bedriftsnavnfilteret. Vi kan også lage andre filter med egne strenger.

```
New-DynamicDistributionGroup -name ledelse -alias ledelse -recipientfilter  
{Company -eq "mafoberg systems as"} -OrganizationalUnit  
"mafoberg.net/ledelse"
```



```
Machine: Exchange1 | Scope:
[PSP] C:\Users\Administrator.MAFOBERG>New-DynamicDistributionGroup -name ledelse -alias ledelse -recipientfilter <Company
-eq "mafodata"> -OrganizationalUnit "mafoberg.net/ledelse"

Name                                     ManagedBy
-----
ledelse
```

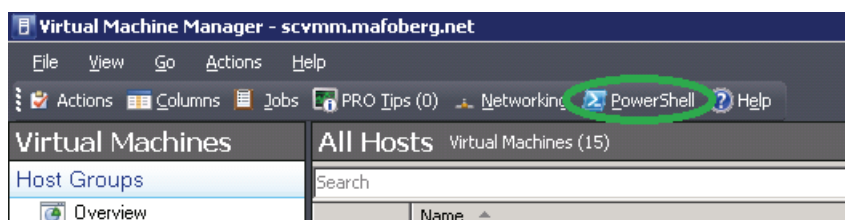
Figur 354 - Dynamisk distribusjonsgruppe

Når det sendes epost til denne gruppen, vil medlemmene av gruppen genereres på nytt ved bruk av filtrene.

4.7 PowerShelladministrasjon av SCVMM

System Center Virtual Machine Manager (SCVMM) er et stort styringssenter for virtuelle maskiner, hoster og cluster. Dette er omtalt tidligere ved installasjon og bruk. Det finnes mange muligheter for å styre funksjonene ved hjelp av PowerShell. Vi skal ta for oss noen cmdlets som kan komme til nytte istedenfor å bruke det grafiske brukergrensesnittet.

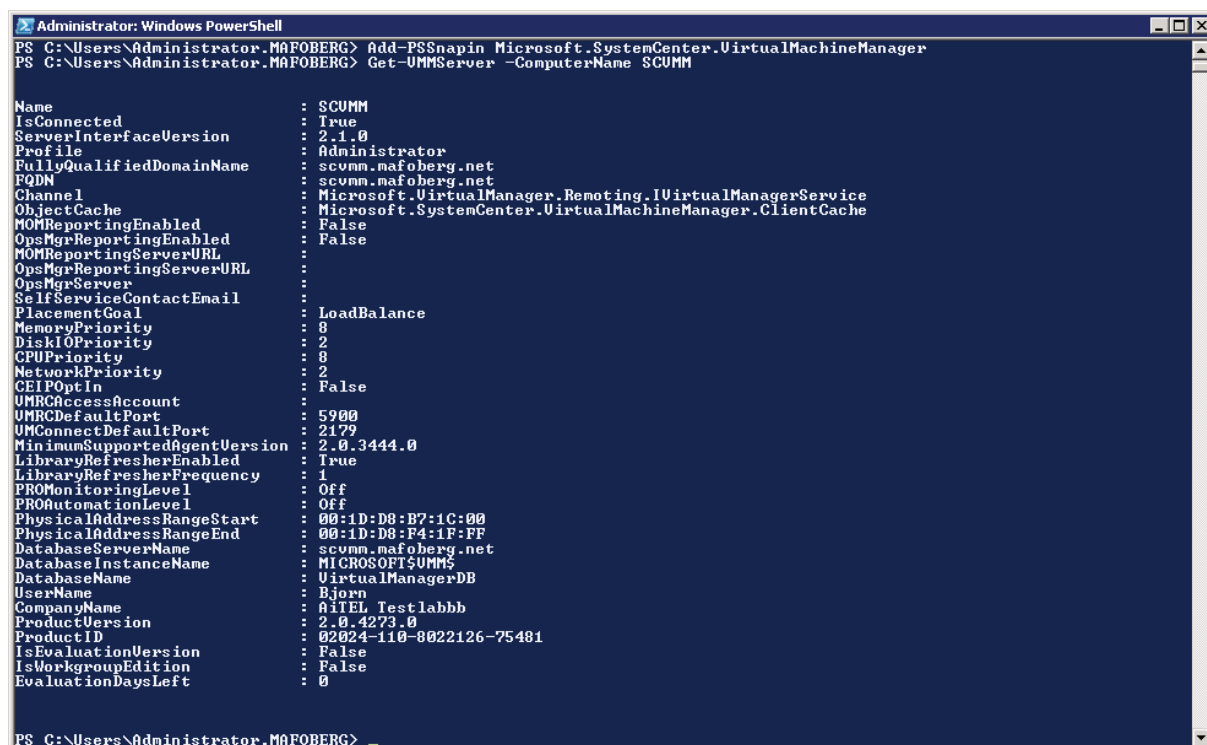
For å kunne bruke kommandoene som er utviklet for SCVMM, må vi legge til modulen for denne. Den heter `Microsoft.SystemCenter.VirtualMachineManager`. Vi kan legge til denne modulen manuelt hvis vi starter PowerShell, eller vi finner snarveien for PowerShellgrensesnittet i SCVMM.



Figur 355 - SCVMM PowerShell oppstart med snapin

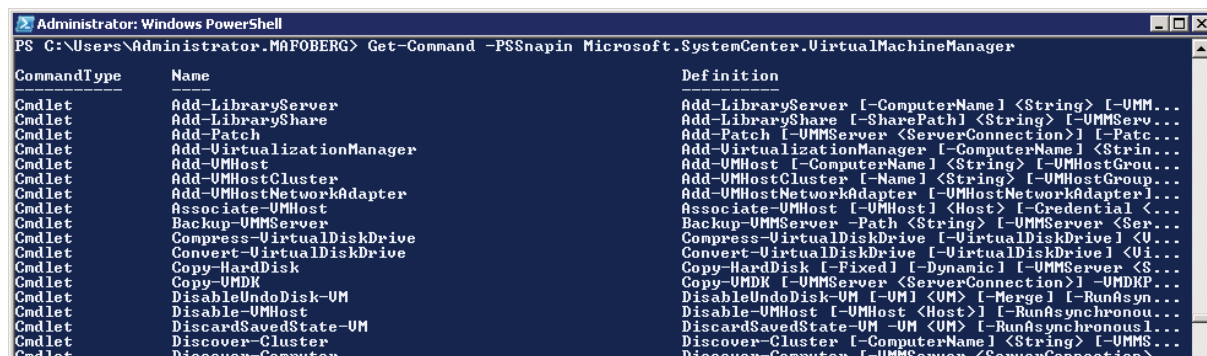
Hvis vi starter PowerShell på vanlig måte med administratorrettigheter fra Windows, må vi så legge til snapin for VMM. Dette gjøres med `"Add-PSSnapin Microsoft.SystemCenter.VirtualMachineManager"`.

Når denne er lagt til i sesjonen, må vi koble til vmm-serveren for å få brukt kommandoene til styringen. `"Get-VMMServer -ComputerName SCVMM"`. Her henter vi SCVMM-serveren og oppgir maskinnavnet til den. Navnet til vår SCVMM-server er "SCVMM". Finnes den, kobles den til.



Figur 356 - SCVMM Server tilkoblet

Vi har nå tilgang til kommandoene for SCVMM-snapin. Disse kan vi list ut.



Figur 357 - SCVMM Powershell kommandoer

Gi også gjerne output til fil ved å peke output og navn. "Get-Command -PSSnapin Microsoft.SystemCenter.VirtualMachineManager > c:\scvmm\list.txt". Eller hvis man vil ha litt mer oversikt uten type, og med formatert tabell: "Get-Command -PSSnapin Microsoft.SystemCenter.VirtualMachineManager | select name, definition | Format-Table @{exp={\$_.Name}; label="Name"; width=30}, @{exp={\$_.Definition}; label="Definition"}".

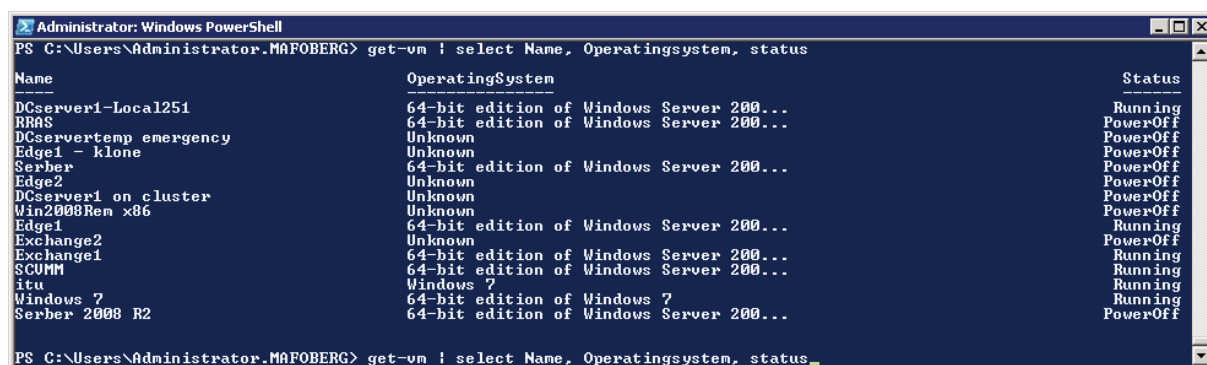
Kommandoene som lister ut informasjon om virtuelle maskiner, hoster og annen informasjon gjelder kun SCVMM-maskinen som er tilkoblet. Her er noen av informasjonskommandoene.

- Get-VM – informasjon om alle virtuelle maskiner.
- Get-VMHost – informasjon om alle hosts.
- Get-ISO – liste over alle ISO-filer som ligger i Library til SCVMM.
- Get-Template – liste over templates.

Vi kan også styre de virtuelle maskinene, slette dem, skru dem av, lagre status, hente checkpoints, starte maskiner, lage nye maskiner og mye mer. Vi skal starte med å liste ut informasjon.

4.7.1. Listing av virtuelle maskiner og informasjon

Vi skal først og fremst liste ut aktuelle maskiner i hele scvmm-miljøet. Vi vil kun ha ut navnene og informasjon om maskinene og kjører derfor "Get-VM | select name, operatingsystem, status".



Figur 358 - PS - Get-VM navn og info

Vi får med navn, hvilket operativsystem som kjører og status på maskinene (av, på og saved status). Dette er av den enkle typen informasjon vi raskt kan liste ut med PowerShell.

Vi skal gå litt videre og se på hvordan vi kan få tak i ipadresser til maskiner, fjerne unødvendig informasjon, lage egen kolonne for adressene og fjerne eventuelle feilmeldinger.

4.7.2. Listing av maskiner med ip-adresser fra DNS

Vi bruker samme metode som før. En av listene må utføre et kall mot DNS for hvert oppslag av maskinene som skal være med. Vi lager oss derfor en funksjon som tar seg av et slikt kall. Den vil da returnere en liste over nettverksadresser den finner.

Siden vi bare skal bruke denne funksjonen til en bestemt kjøring, legger vi også kommandoen for kjøring av selve listingen i samme script.

```

1 #####
2 ## Funksjon for å hente nettverksadresser fra DNS-tjenesten. ##
3
4 # Funksjonshode med maskinnavn som innargument
5 function get-ip-from-host ($hostname) {
6
7     # Try og Catch feilhåndtering
8     try{
9         # Bruker en eksterk klasse for å hente adressene
10        $ips = [System.net.DNS]::getHostEntry($hostname).AddressList -join ","
11        return $ips
12    }
13    catch {
14        # Returnering av Feilhåndteringer
15        return ""
16    }
17 }
18
19 # Kjøpe listing av maskiner med adresser
20 # $vms =
21 get-vm | select name, computernameString, @{name="uppercase"; exp={$_.name.ToUpper()}},
22        @{name="IPadr"; exp={get-ip-from-host -hostname $_.ComputerNameString}}

```

Figur 359 - SCVMM PowerShell adresselisting


I Figur 359 har vi laget et script som gjør bruk av en funksjon som vi har lagt med i scriptet. Denne funksjonen er fint anvendelig andre steder en bare for dette scriptet. Denne må da kjøres for at den skal bli aktiv, men da bør vi unngå kjøring av spesifikke kommandoer sammen med scriptet.

Da er det heller lurt kun å ha et script med funksjoner samlet som man kan aktivere. Kommandoer kan man lagre i variabler som vist i figuren linje 20 og 21. Dette er en enkel måte å lage snarveier til avanserte kommandoer på. Her har vi lagt en Get-VM kommando i variabelen \$vms.

I selectsetningen har vi valgt med fire kolonner. Name, ComputerString, Uppercase og IPadr. Hvis vi ser på kolonnen for IPadr ser vi at vi har tatt med funksjonen *get-ip-from-host* til å returnere nettverksadresse.

Funksjonen *get-ip-from-host (\$hostname) {}* mottar et argument som vi skal bruke. Funksjonen inneholder egentlig kun en kommando som lagres i variabelen \$ips. Vi bruker klassen med kommandoen `[System.net.DNS]::getHostEntry($hostname).AddressList -join ","` som henter nettverksadressen(e) fra maskinen som er navngitt i variabelen \$hostname. `-join ","` gir oss mulighet til å gjøre skille adressene med komma isteden for mellomrom.

Ved kjøring av kun denne kommandoen med en gitt hostname får vi adressene til denne hosten.




```

Administrator: Windows PowerShell
PS C:\Users\Administrator.MAFOBERG> [System.net.DNS]::getHostEntry("Exchange1").AddressList -join ","
2002:9e26:38f8::9e26:38f8,150.30.56.248
PS C:\Users\Administrator.MAFOBERG>

```

Figur 360 - SCVMM Hostadresseliste

Når denne settes i en funksjon vil den kunne kalles med funksjonen *get-ip-from-host*.



```

Administrator: Windows PowerShell
>>
PS C:\Users\Administrator.MAFOBERG> get-ip-from-host Exchange1
2002:9e26:38f8::9e26:38f8,150.30.56.248
PS C:\Users\Administrator.MAFOBERG>

```

Figur 361 - scvmm get-ip-from-host

I funksjonen har vi lagt alt inne i en unntakstilstand "Try {} Catch {}". Dette gjør at vi kan gjøre hva vi vil med eventuelle feilmeldinger som kommer i retur. Beskrive dem eller kanskje fjerne dem om vi vil. Vi har lagt selve kommandoen for å hente adressene inne i Try {}. Dette gjør at kommandoen kjøres. Men dersom den skulle feile eller sende tilbake en feilmelding, vil denne kunne behandles fra blokka Catch {}.

I linje 21 og 22 i Figur 359 kjøres kommandoen som bruker funksjonen. Vi får da listet ut fire kolonner.



```
PS C:\Users\Administrator.MAFOBERG> get-vm | select name, computernameString, @(<name='uppercase'; exp=<$_.name.ToUpper()>), @(<name='IPAdr'; exp=<get-ip-from-host -hostname $_.ComputerNameString>)
```

Name	ComputerNameString	uppercase	IPAdr
DCserver1-Local251	DCserver1.mafoberg.net	DCSERVER1-LOCAL251	158.38.56.249
RRAS	RRAS.mafoberg.net	RRAS	<2002:9e26:38fd::9e26:38fd...
DCservertemp emergency	Unknown	DCSERVERTEMP EMERGENCY	
Edge1 - kloner	Unknown	EDGE1 - KLONE	
Serber	WIN-C8GG0HCHPP8	SERBER	
Edge2	Unknown	EDGE2	
DCserver1 on cluster	Unknown	DCSERVER1 ON CLUSTER	
Win2008Rem x86	Unknown	WIN2008REM X86	158.38.56.247
Edge1	edge1.mafoberg.net	EDGE1	
Exchange2	Unknown	EXCHANGE2	
Exchange1	Exchange1.mafoberg.net	EXCHANGE1	<2002:9e26:38f8::9e26:38f8...
SCUM	scvm.mafoberg.net	SCUM	<fe80::9c60:8bf3:d2f6:ae28...
ITU	Unknown	ITU	
Windows 7	testpc.mafoberg.net	WINDOWS 7	<2002:9e26:3830::9e26:3830...
Serber 2008 R2	Unknown	SERBER 2008 R2	

```
PS C:\Users\Administrator.MAFOBERG>
```

Figur 362 – IP-adresse listing

Dette kan lett gi oss oversikt over hvilke maskiner som har hvilke adresser knyttet til vår DNS-server.

4.7.2. Skifte maskinnavn på en maskin.

Med kommandoen Set-VM kan vi forandre en hel masse for virtuelle maskiner. Det vi lettest kan forandre er slik som navn, eier av virtuell maskin og beskrivelse av maskinen. Vi lister ut maskiner med beskrivelse først. Get-VM | select name, description.

For å skifte navn på en maskin, henter vi først maskinen vi vil skifte navn på, deretter sender Set-VM inn i en pipe som skifter navn på denne. Get-VM -Name "Server 2008R2 Karsten" | Set-VM -Name "Server 2008 R2 Karsten". Navnet blir forandret bare for denne maskinen.

Det kan forekomme tilfeller av å forandre navn på mange maskiner ved hvis man har en rekke testservere som har like navn eller andre parametere som alle skal forandres i samme serie. Dette kan dreie seg om flere testservere for kursdeltakere, maskiner som er kalt "test1", "test2", "test3" osv. Dette kommer vi tilbake til.

Man kan sette flere opsjoner på en maskin ved hjelp av Set-VM enn bare navn, for eksempel hardwareoppsett som cpu-type, cpubegrensning, oppførsel (startaction og stopaction), eier og mye mer.

4.7.3. Lage en ny virtuell maskin.

En ny maskin blir gitt et navn og har en rekke hardwarekomponenter. Alt dette kan bestemmes igjennom cmdleten New-VM i PowerShell. For å lage en ny maskin på enkleste mulige måte er å angi navnet til maskinen, stedet hvor maskinen skal lagres, virtuell harddisk som skal benyttes og hosten hvor den virtuelle maskinen skal være tilgjengelig fra.

Selve hosten hvor maskinen skal ligge, legger vi gjerne i en variabel. Denne henter vi ut med Get-VMHost.

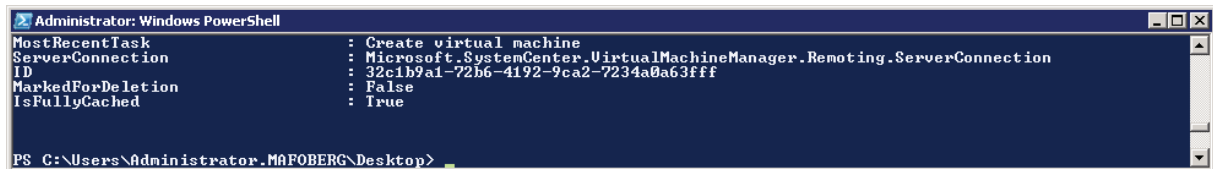
```
$VMHost = Get-VMHost -VMMServer scvmm | where {$_.Name -eq "hyp-1.mafoberg.net"}
```

Den virtuelle harddisken som skal brukes, henter vi fra library-serveren. Der finnes det to standardtyper som følger med ved installasjon av SCVMM. "Blank Disk – Large" og "Blank Disk – Small". Vi bruker her den store. Vi legger også denne i en variabel.

```
$VirtualHD = Get-VirtualHardDisk | where {$_.Name -eq "Blank Disk - Large"}
```

Deretter oppretter vi den nye maskinen. Vi angir navn, sted på hosten hvor maskinen skal ligge, den virtuelle harddisken ved hjelp av variabel, hosten ved hjelp av variabel og til slutt parameteren `-RunAsynchronously` for å kjøre prosessen i bakgrunn.

```
New-VM -Name "Virtuellmaskin" -Path "C:\ClusterStorage\Volum1" -  
VirtualHardDisk $VirtualHD -VMHost $VMHost -RunAsynchronously
```



Figur 363 - Ny virtuell maskin

Når man ikke definerer noe maskinvare for en virtuell maskin, vil det bli angitt en standard profil for maskinvare fra SCVMM. Standard CPU er da 1GHz Pentium 3 Xeon, og minne på 512MB RAM.

4.7.4. Lage maskin ut ifra template.

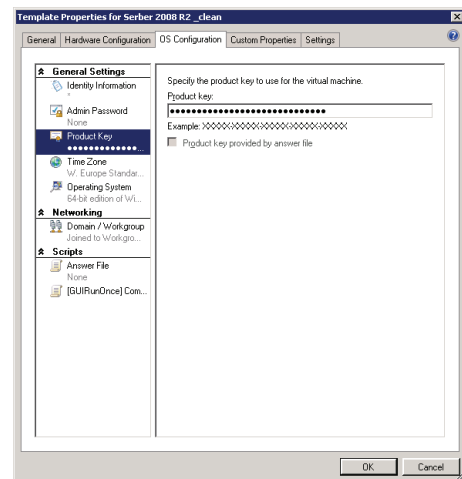
Å lage maskiner ut ifra en template kan være meget effektivt hvis vi skal opprette mange maskiner med ferdiginstallert operativsystem. Da trenger kun operativsystemet på den nye virtuelle maskinen kun og konfigureres. Skal man distribuere mange slike maskiner, bør vi sette opprettelsen av maskinene i kø. Det kan ta sin tid å distribuere mange maskiner og det bør derfor kjøres på et tidspunkt hvor man slipper å sitte å vente på jobben.

Jobben er å kopiere disk og maskinfil til det nye området. Det bør derfor være sikret at trafikken ikke går for tregt. Etter at data er kopiert, vil prosessen starte maskinen for konfigurering. Det er mulig under denne prosessen å koble til skjermen til maskinen igjennom GUI-konsollet for å se på prosessen. Eventuelt se etter hvor prosessen har stoppet om dette har skjedd.

4.7.4.1. Template forhåndsoppsett.

Hvis man skal lage en maskin ut ifra en template ved hjelp av PowerShell, bør templateen være satt opp og klar til å fullstendig gjennomføres. Det vil si oppsett av operativsystem bør ligge klart i templateen om det skal gjøres effektivt. Et eksempel på problem som kan oppstå er produktnøkkel som ikke er gyldig. Det er flere måter å gjøre dette på, men vi tar to her.

En metode er å bruke en gyldig produktnøkkel for å kunne sette opp maskinen. Valget med aktivering vil da være på. Det vil si aktiveringen vil skje neste gang maskinen slås på og er koblet mot internett. Er da produktnøgkelen allerede i bruk, får man da beskjeden om å bytte den eller kjøre i prøvemodus en stund.



Figur 364 - Template OS-konfigurasjon

En annen metode er å bruke scvmm-konsollet for å koble til skjermen på maskinen som er under opprettelse. Der kan man velge å aktivere windows senere, og deretter gå videre for fullføring av å lage maskinen.

Hvis det er flere spesielle oppsett man vil bruke, er det mulig å legge til en *Answer File* til konfigurasjonen. Denne filen kan inneholde spesifikke oppsett ut over det scvmm-library attributtene inneholder. Man kan lage slike answerfiles ut i fra Windows Automated Installation Kit (WAIK).

4.7.4.2. Enkelt script for en maskin.

Vi vil opprette en maskin fra templateen *Serber 2008 R2 _clean*. Denne templateen vil vi plassere på hosten *hyp-1.mafoberg.net*. Vi legger begge disse i sine egne variabler.

```
$Template = Get-Template -VMMServer SCVMM.mafoberg.net | where {$_.Name -eq "Serber 2008 R2 _clean"}
```

`Get-Template` henter informasjon om templatene. `-VMMServer` bestemmer hvilken library-server vi skal velge å hente informasjonen fra.

Vi vil plukke ut en bestemt template fra library-serveren. Dette gjør vi ved å sende en where-setning i pipe til `Get-Template` kommandoen. Ved å finne `$_Name` (navnet på objekt fra `Get-Template`) lik navnet på templateen vi vil ha, får vi kun denne i retur.

```
$VMHost = Get-VMHost -VMMServer localhost | where {$_.Name -eq "hyp-1.mafoberg.net"}
```

Det samme gjelder VMM-hosten vi vil plassere den nye maskinen på. Vi plasserer host-objektet i variabelen `$VMHost`.

Ny maskin med kommandoen `New-VM`.

```
New-VM -Template $Template -Name "testserver" -VMHost $VMHost -Path "C:\ClusterStorage\Volume1" -JobGroup 4e5b50e5-120c-41da-a9f2-154851f03255 -RunAsynchronously -Owner "MAFOBERG\Administrator" -AnswerFile $null -RunAsSystem -StartAction NeverAutoTurnOnVM -StopAction SaveVM
```

`New-VM` er en lang kommando. Vi skal ta for oss alle parameterne.

`-Template $template` – Angir hvilken template vi skal bruke. Vi har allerede plukket ut og lagret templateen-objektet i variabelen `$template`.

`-Name "testserver"` – Navn på maskinen i Hyper-V.

`-VMHost $VMHost` – Hosten hvor den virtuelle maskinen skal kjøres. Host-objektet angis. Denne har vi også hentet på forhånd og lagt i variabelen `$VMHost`.

`-Path "C:\ClusterStorage\Volume1"` – Hvor på hosten maskinen skal plasseres. Dette er plassen hvor vi lagrer alle maskiner i testmiljøet. Lønner seg å sortere disse.

`-JobGroup 4e5b50e5-120c-41da-a9f2-154851f03255` – Et jobb-gruppenummer for SCVMM. Dette må være unikt for hver gruppe av arbeid SCVMM skal utføre. Eventuelt kan man legge til arbeidsoppgaver i en slik gruppe. Ved opprettelse av mange maskiner skal vi se på hvordan å lage et automatisk opprettet navn.

`-RunAsynchronously` – gjør at prosessen med å kjøre maskinen om det skulle kjøres flere setter jobben i bakgrunn og sender en tilbake til kommandolinjen. Uten denne ville kommandolinjen ikke blitt tilgjengelig før jobben var utført.

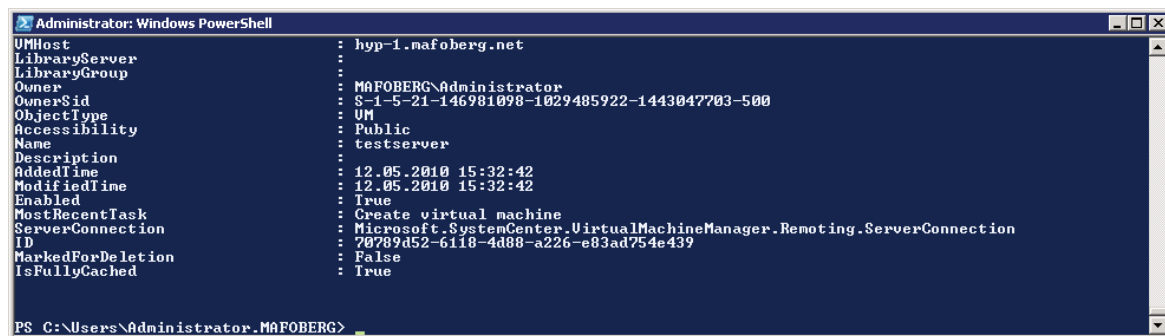
`-Owner "MAFOBERG\Administrator"` – Eier av maskinen. Dette er ikke nødvendig, men vil da bli listet som ukjent.

`-AnswerFile $null` – Automatisk svarfil for opprettningen av den nye maskinens operativsystem. Vi bruker variabelen `$null`²⁶ som en bekreftelse på at ingenting skal være der.

`-RunAsSystem` – Kjører maskinen som lokal maskin med lokale innstillinger. Hvis maskinen skulle vært direkte medlem av et domene, hadde vi brukt `-RunAsUserCredential` med domenedata.

`-StartAction NeverAutoTurnOnVM` og `-StopAction SaveVM` – bestemmer hva som skal skje med maskinen under hostens omstendigheter.

Ved kjøring av disse tre setningene vil vi få et resymé av maskinen som blir opprettet. Da får vi på forhånd se all informasjon om eventuelt vi ser noe som ikke er riktig, og har da mulighet til å avbryte jobben.



```

Administrator: Windows PowerShell
VMHost                : hyp-1.mafoberg.net
LibraryServer         :
LibraryGroup          :
Owner                 : MAFOBERG\Administrator
OwnerSid              : S-1-5-21-146981098-1029485922-1443047703-500
ObjectType            : VM
Accessibility         : Public
Name                  : testserver
Description            :
AddedTime              : 12.05.2010 15:32:42
ModifiedTime          : 12.05.2010 15:32:42
Enabled               : True
HostRecentTask        : Create virtual machine
ServerConnection      : Microsoft.SystemCenter.VirtualMachineManager.Remoting.ServerConnection
ID                    : 70789d52-6118-4d88-a226-e83ad754e439
MarkedForDeletion     : False
IsFullyCached         : True

PS C:\Users\Administrator.MAFOBERG>
  
```

Figur 365 - PowerShell ny maskin fra template

4.7.4.3. Opprette mange maskiner

Når en skal opprette mange maskiner, kan dette bli dumt å gjøre manuelt. Dette kan gjøres enkelt ved å lage en funksjon som forandrer navnet til jobbgruppen for hvert opplag av maskiner som lages. Vi kan også legge til en beskrivelse til hver maskin.

Det er flere måter å opprette flere maskiner etter hverandre på. Dette kommer an på hvordan man setter jobbene i kø på. Det er to viktige detaljer man må tenke på. Jobbkø-ID og parameteren

`-RunAsynchronously` i cmdleten `New-VM`.

Å sette maskiner i forskjellig jobbkø vil gjøre at alle jobber som blir satt i

²⁶ Variabelen `$null` inneholder ingen ting. `""` er en tom streng, altså mer enn ingenting.

En helt enkel måte å generere tall på er `$gen = new-object random`. Denne lager en integer-verdi²⁷. Vi lager så verdi ved å bruke objektet til å lage en verdi med. `$t = $gen.next()`. Da vil variabelen `$t` inneholde et generert tall som for eksempel 805863160.

En JobGroup krever et litt spesielt oppsett av verdi. Dette blir et objekt. For eksempel `b1fa6569-0775-4919-9686-a5d5c9794e91`. Da genereres objektet og legges i en variabel. `$TemplateJobGroup = [System.Guid]::NewGuid()`. Vi har da en unik JobGroup ID.



Figur 366 - \$TemplateJobGroup

Hvis man ser i GUI-konsollet under kategorien Jobs, ser man status på alle jobber. De som kjører, feilet, stoppet osv. Disse jobbene kjøres samtidig. Hvis man starter mange opprettelser i hver sin jobbgruppe, kjøres disse samtidig. Dette kan være ulønnsomt i enkelte tilfeller om man ønsker å ta i bruk en maskin så fort den er ferdig etter at neste maskin har startet opp. Kjøres de samtidig, tar de av felles ressurser og kan skape problemer. Fullt bruk av minne og tungt bruk av nettverket. Derfor kan det lønne seg å starte jobbene etter hverandre. En jobbgruppe har steg (steps) som kjører i rekkefølge. Når et step er ferdig, kjøres neste.

```

1 #####
2 ### Funksjon for å Lage ny VM ut ifra template.          ###
3
4 # Add-PSSnapin for SCVMM.
5 try {
6     # Test for snapin og tilkobling til vmm-server
7     Add-PSSnapin microsoft.systemcenter.virtualmachinemanager
8     Get-VMMServer -ComputerName scvmm
9 }
10 catch {
11 }
12
13 # Funksjonshode med maskinnavn som innargument
14 function New-vm-from-template ($vmname="", $vmcount=1) {
15
16     # Avbryt hvis navn ikke er angitt
17     if ($vmname -eq $null -or $vmname -eq "") {
18         write-host "Ingen maskinnavnstandard gitt!"
19         return
20     }
21
22     # Hente host Informasjon - statisk
23     $VMHost = Get-VMHost -VMMServer SCVMM | where {$_.Name -eq "hyp-1.mafoberg.net"}
24
25     # Hente template-informasjon - statisk
26     $Template = Get-Template -VMMServer SCVMM | where {$_.Name -eq "Serber 2008 R2 _clean"}
27
28
29     for ($i = 1; $i -le $vmcount; $i++) {
30         Write-host "Maskin $i - $vmname"
31
32         # Generering av JobGroup ID
33         $TemplateJobGroup = [System.Guid]::NewGuid()
34
35         # Ny maskinjobb
36         New-VM -Template $Template -Name ($vmname+$i) -VMHost $VMHost -Path "C:\ClusterStorage\Volume1" `
37         -JobGroup $TemplateJobGroup -RunAsynchronously -Owner "MAFOBERG\Administrator" -AnswerFile $null `
38         -RunAsSystem -StartAction NeverAutoTurnOnVM -StopAction SaveVM
39     }
40 }

```

Figur 367 - SCVMM New-vm-from-template

Dette er ett enkelt script som inneholder en funksjon. Funksjonen tar imot navn for maskin og antall maskiner og oppretter maskiner i hver sin jobbgruppe. Disse vil da kjøres samtidig. Vi skal ta for oss hver av delene i scriptet. Scriptet ligger som tillegg til dokumentasjonen med navnet *NewVM.ps1*.

²⁷ Integer er betegnelse for heltallsverdi. Signert verdi: -2,147,483,648 til +2,147,483,647. Usignert: 0 til +4,294,967,295

Linje 5 til 11 er en unntaksfunksjon som tester to kommandoer. Innhentning av snapin for scvmm-tillegg og deretter tilkobling til scvmm-serveren. Eventuelle feilmeldinger kommer som unntak som vi kan gjøre hva vi vil med. Det vil fortsatt komme vanlige meldinger fra disse to kommandoene.

Linje 14 og 40 er selve funksjonen hvor vi angir variabel for argument 1 og 2. Vi setter en minimumsverdi for antall maskiner direkte i hodet. Dermed trenger vi bare maskinnavnet om det skulle være bare en maskin.

Linje 16 til 20 er en if-setning med blokkinnhold hvor vi tester navnevariabelen gitt i funksjonshodet for om det inneholder navn. Ellers ville maskinen bare blitt kalt "1" hvis ingen parameter hadde blitt gitt. Vi skriver ut en feilmelding og returnerer ut av funksjonen med `return` uten verdi. Hadde vi hatt en avansert funksjon med en liste av feilgrunnlag, kunne vi gitt feilkode ut av funksjonen.

Linje 23 og 26 er standardvariabler vi setter. `$VMhost` for hosttilkobling og `$Template` for templateinformasjon har vi sett tidligere. Linje 29 til 39 er en for-loop som skal opprette hver av maskinene. I linje 30 skriver ut navn for hver maskin den skal til å opprette. Linje 33 bruker en metode for å generere en ny jobbgruppe id som den legger i variabelen. Linje 36 til 38 er kommandoen for å opprette en maskin.

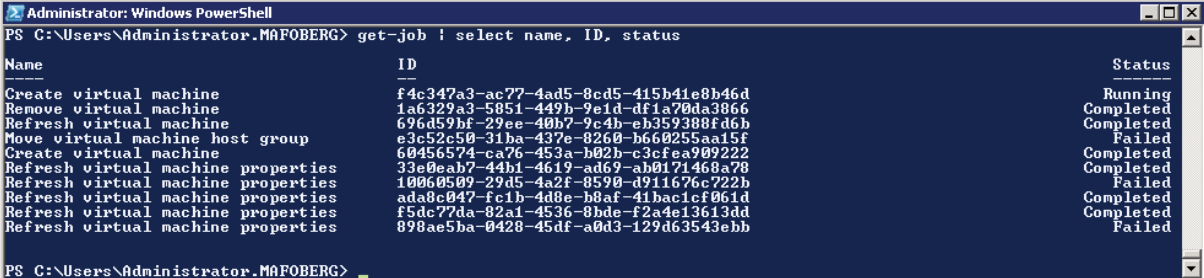
Setningene for New-VM er delt opp. Dette gjøres med escape-tegnet ``` alene.

4.7.4.4. Opprette maskiner i kø.

For å opprette flere maskiner i kø uten å kjøre samtidig, kan vi kjøre prosessen i PowerShell slik at den ikke settes i bakgrunn. Dette gjøres ved ikke å bruke parameteren `-RunAsynchronously` i cmdleten `New-VM`. Da vil jobben gjøre seg ferdig før den lar PowerShell-sessjonen fortsette.

`-JobGroup` vil da spille en viktig rolle om jobben skal avbrytes eller ikke. Hvis jobbgruppen genereres forskjellig for hver maskin som skal opprettes i en loop, vil det å avbryte en jobb kun avbryte den jobben for den aktuelle jobbgruppen. Deretter vil loopen fortsette å opprette nye jobbgrupper for hver maskin og fortsette. Hvis det ikke blir generert ny jobbgruppe, vil resten av loopen fortsette å prøve å legge til en jobb hvor jobbene allerede er avbrutt eller ferdig. Dermed vil den fortsette helt ut uten å opprette flere maskiner.

4.7.5. Avbryte en jobb og slette en maskin



```

PS C:\Users\Administrator.MAFOBERG> get-job | select name, ID, status
Name                                     ID                                     Status
----
Create virtual machine                  f4c347a3-ac77-4ad5-8cd5-415b41e8b46d Running
Remove virtual machine                  1a6329a3-5851-449b-9e1d-df1a70da3866 Completed
Refresh virtual machine                  696d59bf-29ee-40b7-9c4b-b359388fd6b Completed
Move virtual machine host group          e3c52c50-31ba-437e-8260-b660255aa15f Failed
Create virtual machine                  60456574-ca76-453a-b02b-c3cfa909222 Completed
Refresh virtual machine properties       33e0eab7-44b1-4619-ad69-ab0171468a78 Completed
Refresh virtual machine properties       10060509-29d5-4a2f-8590-d911676c722b Failed
Refresh virtual machine properties       ada8c047-fc1b-4d8e-b8af-41bac1cf061d Completed
Refresh virtual machine properties       f5dc77da-82a1-4536-8bde-f2a4e13613dd Completed
Refresh virtual machine properties       898ae5ba-0428-45df-a0d3-129d63543ebb Failed
PS C:\Users\Administrator.MAFOBERG>

```

Figur 368 - SCVMM jobblisting

`Get-Job | Select name, ID, status` – gir oss jobbene som ligger i cache og status på dem. Vi vil da se at vi kan ha flere jobber, men ikke alle er kjørende. Vi vil ha tak i maskinen som er kjørende og denne skal vi stoppe.

```
Get-Job | where {$_.Name -eq "Create virtual machine" -and $_.Status -eq "Running"} | Stop-job
```

Vi abryter med dette alle kjørende jobber. Hvis vi skulle valgt en spesifikk jobb, måtte vi filtrert ut denne i where-setningen. For eksempel `where {$_.Name -eq "Create virtual machine" -and $_.Status -eq "Running" -and $_.ID -eq "f4c347a3-ac77-4ad5-8cd5-415b41e8b46d"}.`

Jobben som startet å opprette maskinen har ikke fjernet komponentene etter avbrytelsen. Dette må vi i så fall gjøre selv. Vi gjør dette ved å kjøre `Remove-VM -VM "testserver"`. Testserveren blir da slettet helt.

Vi vil slette flere servere av samme type når vi har opprettet noen av dem. Vi har testserver1, testserver2, testserver3 osv.

4.7.5.1. Slette maskiner

Man bruker kommandoen `Remove-VM` for å fjerne virtuelle maskiner. Denne kommandoen kan komme til nytte dersom man skal fjerne flere maskiner om gangen. Men hvordan kan man finne en felles verdi på virtuelle maskiner? Det er flere typer fellesverdier. Hvis det tilfeldigvis ikke er noen, kan man lage fellesverdier.

Eksempler på felles verdier:

- Navn på maskiner i serie: maskin1, maskin2, maskin3...
- Beskrivelse (description) på maskin kan redigeres etter ønske.
- Status på maskin: "CreationFailed", "Running", "Stopped", "Saved" osv.

Når vi har avbrutt jobber, kan det være dumt å ha mange halvliggende maskiner på hosten. Når man lager maskiner, vil avbrutte jobber gi maskinene statusen "CreationFailed". Det er viktig å merke seg at status på en maskin er forskjellig fra statusen til en jobb.



Figur 369 - Status på maskiner

I tillegg ligger scriptet *RemVM.ps1* som inneholder flere av funksjonene vi trekker frem her.

Det kan være en risiko å fjerne maskiner etter status dersom en har prøvd å opprette en maskin som allerede eksisterer. Hvis maskinen eksisterer, vil den havne i status "CreationFailed" selv om den egentlig fungerer. Maskinen slettes da uansett siden status er satt til "CreationFailed".

```

53 # Funksjonshode med maskinserie som innargument
54 function RemVM-by-status ($status) {
55
56     # Avbryt hvis navn ikke er angitt
57     if ($status -eq $null -or $status -eq "") {
58         Throw "Ingen status gitt!"
59     }
60
61     # Hente host Informasjon - statisk
62     $VMHost = Get-VMHost -VMMServer SCVMM | where {$_.Name -eq $hostname}
63
64     # Henter maskiner
65     $machines = get-vm | where {$_.status.toString().toLowerCase() -eq $status}
66
67     # Tester variabelen $machines for innhold.
68     if (-not $machines) {
69         Throw "Ingen maskiner med gitt status. Status = " + $status
70     }
71
72     foreach ($item in $machines) {
73         Write-host ("Sletter maskin " + $item.name)
74
75         # Fjerner maskin fra $machines
76         Remove-VM -RunAsynchronously -VM $item
77     }
78
79     Write-Host (" " + $machines.count + " maskiner fjernet.")

```

Figur 370 - SCVMM Funksjon RemVM-by-status

I Figur 370 har vi tatt fellesbetegnelsen status for å fjerne maskiner. Den egner seg godt til maskiner som var påbegynt.

Funksjonen her vil kreve en statustekst som sjekkes i en if-setning. Hvis status er gitt, kjører funksjonen videre. Vi henter alle maskiner med angitt status. Hvis ingen maskiner eksisterer med angitt status, vil if-setningen i linje 68 til 70 gi en feilmelding om dette. \$machines inneholder derfor ingen maskiner, og løkken i linje 72 til 77 vil derfor ikke kjøres.

Foreach-løkken kjøres og sletter alle maskiner samtidig. Alle slettejobber havner i sin egen jobbgruppe og de slipper dermed å vente på hverandre.

4.7.6. Andre kommandoer

Kommandoer som er relevante for å styre maskiners tilstand som vi ikke går nærmere inn på detaljer om, er slike som

- Move-VM – Flytter maskin fra en host til en annen.
- New-VM – Lager ny maskin ut ifra standard format eller templates fra library-servere.
- Refresh-VM – Oppdaterer status på virtuelle maskiner fra hosten som kjører dem. Alt som har med maskinen å gjøre blir da hentet på nytt. Hvis noe skulle vise seg å være feil, vil status på maskinen bli satt til "missing", eller "Failed" alt etter hva som er galt eller nytt.
- Remove-VM – Fjerner maskiner fra host. Maskiner blir da slettet fullstendig.
- Repair-VM – Reparerer maskin. Hvis maskinen skulle være ufullstendig behandlet og har fått status der etter, kan denne cmdleten fortsette jobben der den slapp sist. Den kan også bare reparere konfigurasjonsfeil som gjør at maskinen ikke kan kjøre. Slik som feilmontert ISO-filer og andre koblinger som hører til maskinen.
- Resume-VM – Gjenoppretter kjøring av maskin. Enten den har havnet i lagret status, slått av eller kritisk avsluttet.
- SaveState-VM – Lagrer maskinen og avbryter kjøring. Maskinen vil da havne i *saved state* modus. Den har da ikke avsluttet men kun stoppet all kjøring .
- Shutdown-VM – Sender avsluttnings-signal til den virtuelle maskinen for å avslutte. Den blir da avsluttet på normal måte.
- Start-VM – Starter opp en virtuell maskin som er slått av.
- Stop-VM – Slår maskinen direkte av uten å avslutte den på normalt vis.

- Store-VM – Hvis man ikke har bruk for maskinen på en stund, er det mulighet for å lagre maskinen på en library-server. Maskinen blir da flyttet dit.
- Suspend-VM – Stopper maskinen midlertidig for kjøring. Dette gjør at den ikke bruker maskinkraft og andre ressurser.

Ved å bruke `Get-Command -PSSnapin`

`Microsoft.SystemCenter.VirtualMachineManager` som vi tidligere har vært inne på, vil vi få en liste over kommandoer. Da vil vi få en rekke kommandoer som kan brukes til detaljer for virtuelle maskiner, hoster og jobber.

Ved å bruke for eksempel `Get-Help ShutDown-VM -Examples` kan man få eksempler på hvordan kommandoen kan brukes. Dette kan være svært nyttig om man ikke vet hvordan man helt skal benytte en kommando.

4.8 Nyttige Scripts i forbindelse med rapportskriving

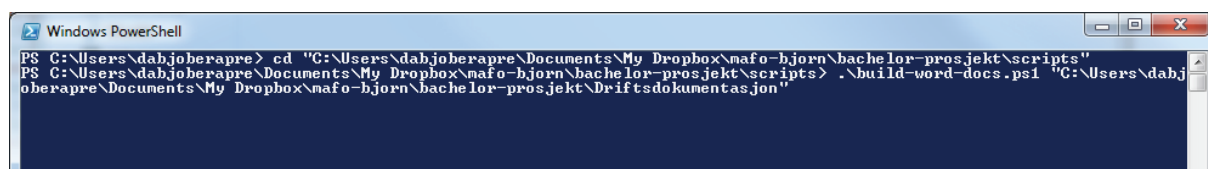
Ved skriving av denne rapporten, har vi tatt i bruk en del script for å gjøre hverdagen lettere. Dissescriptene er vedlagt og vi skal her forklare hva de gjør og hvordan de brukes.

4.8.1 Sammenfletting av Word-dokumenter

Når en skriver en rapport som inneholder flere separate deler, vil en måtte sette sammen disse delene til én sammenhengende fil ved innlevering. Dette blir noe tungvint å gjøre manuelt så vi har laget et script "build-word-docs.ps1" som gjør dette mer automatisk. Det fungerer ved først å gå igjennom ei spesifisert mappe og dens undermapper for å søke etter Word-filer. Deretter oppretter den et word-dokument, og bruker COM til å legge til hvert av dokumentene i ett felles dokument. Scriptet er delvis basert på en Word-makro skrevet i VBA.

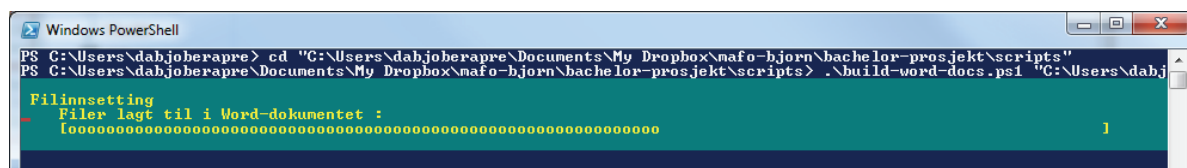
Scriptet kjøres slik, gitt at det finnes I stående mappe:

```
.\build-word-docs.ps1 "lokasjon-til-mine-word-dokumenter"
```



Figur 371 - Eksempel på bruk

Scriptet vil så gå igjennom og finne alle ".docx"-filer og åpne et nytt Word-dokument satt sammen av alle de individuelle delene.



Figur 372 - Scriptet samler inn filer

Sluttrapport

Virtualisering og
bruk av software

Gruppe 6E – Bachelor 2010

Marius Foss, Bjørn Inge V. Berg

Forord

Drift av datasystemer blir et viktigere område i tiden fremover. Datasystemer blir mer kompliserte, og sikkerheten blir mer og mer viktig. Vi har fokusert på områdene virtualisering og automatisering i dette prosjektet.

I sluttrapporten vil vi gå igjennom en oppsummering av prosjektet. Vi tar derfor med en beskrivelse av oppgaven, hvordan vi gikk frem for å løse oppgaven, hvordan samarbeidet har vært, hva som kan gjøres videre og en referanseliste over kilder vi har brukt i løpet av prosjektet.

Innhold

1. Oppgavebeskrivelse	3
2. Hvordan oppgaven ble utført	3
2.1. Standarder og metoder	4
2.2. Bruk av informasjon	4
2.3. Maskinvare som ble brukt	4
2.4. Programvare som ble brukt	4
2.5. Fordeling av arbeid	5
2.6. Dokumentasjon	5
3. Gjennomføring av prosjektet	5
4. Videre arbeid	6
5. Referanser	7

1. Opppgavebeskrivelse

Oppgaven er en bacheloroppgave fra et treårig studium på Høgskolen i Sør-Trøndelag, tildelt av faglærerne Stein Meisingseth og Jostein Lund. De har også vært veilederne for prosjektet. Selve oppgaven har hatt en varighet på et halvt år, et prosjekt som skulle foregå på ca 450 timer.

Vi er to deltakere som har fått tildelt en case som omhandler en fiktiv bedrift som vi har gitt navnet Mafoberg Systems AS. Bedriften er en stor bedrift med flere hundre ansatte. Bedriften hadde vokst, og de har lagt til tjenester og maskinvare etter hvert som tiden har gått. Den består derfor av gammel maskinvare som kjører som en fysisk tjener med tjenester direkte på.

Formålet med oppgaven har vært å simulere en oppgradering og hvordan man kan starte overgang fra et fysisk miljø til et virtuelt miljø. Vi har fokusert på kun å bruke Microsofts programvare i alle sammenhenger. Fra virtualiseringsprogramvare til styringsprogramvare og tjenester.

For å gjøre det lettere for de som skal bruke dette videre, har vi gjort omfanget på prosjektet til å omfatte et mindre nettverk. Deretter når dette går greit, vil utvidelse av virtualiseringen bli lettere.

Konvertering av gamle servere er en viktig mulighet for å effektivisere overgangen. Derfor er konvertering av fysiske maskiner over til virtuelle maskiner en viktig del av prosessen.

Siden virtualisering gjør det lettere å behandle servere, vil det også mest sannsynlig øke med antall servere og muligheten for flere brukere. Systemdrift ved hjelp av automatisering med PowerShell vil derfor være en vesentlig del av oppgaven. Vi har derfor laget en del script som kan brukes og videreutvikles etter eget ønske som passer andre systemer når bedriften oppgraderes videre.

2. Hvordan oppgaven ble utført

Oppgaven for en slik stor bedrift kan være omfattende. Det er derfor flere viktige grunner til å dele opp et slikt arbeid.

- Å ta hele systemet for en bedrift på en gang gir større risiko for bedriftens mulighet til å fungere i mellomtiden.
- Lettere å arbeide med mindre deler om gangen for de som utvikler det nye systemet.
- Man kan se og rette problemer tidlig uten at dette skal gå ut over bedriften.
- Videre arbeid blir lettere å iverksette.

For at fremgangen skal være konstruktiv har vi derfor satt retningslinjer å forholde oss til. Prosjektet inneholder derfor

- forstudie av oppdraget og bedriften
- systemkrav for prosjektet
- driftsdokumentasjonen som er hovedinnholdet
- tillegg til driftsdokumentasjonen
- sluttrapport som inneholder oppsummering og gjennomgang av prosjektet

2.1. Standarder og metoder

Vi har satt oss standarder for å lette oppgavens arbeid. Forskjellig utviklingsmiljø ville gjort det vanskeligere å samle dokumentasjon og innhold. Vi har derfor holdt oss til bruk av et sett utviklingsmiljø.

For dokumentasjon har vi brukt Microsoft Office 2007 med Word, Excel, Power Point, Microsoft Visio og Microsoft Project.

Vi har brukt maler og retningslinjer tildelt fra veileder. Vi har sett eksempler på utgangspunkt for oppsett av forstudierapport, systemkravdokumentasjon, sluttrapporter og timelister.

2.2. Bruk av informasjon

På forhånd har vi gått inn for å samle dokumentasjon for systemene vi skulle ta i bruk. Vi visste at vi ikke ville komme til å få nok litteratur på forhånd. Det måtte også hentes informasjon underveis i prosjektet. Vi har derfor brukt søkemotorer for å finne relevant stoff i tillegg til bøker og papirversjoner. Dette vil også bli fremtidens måte å jobbe på når man utvikler slike løsninger.

Vi oppgir derfor kun domener for søk av litteratur og linker til eventuelle presise informasjonskilder. Bøker vil ofte forekomme i PDF-format og vil bli henvist til tittelen på denne.

Mye av informasjonen vil også bli funnet på Microsofts nettsider og Technets avdelinger ved Microsoft.

2.3. Maskinvare som ble brukt

Vi har startet med to 64-bits servere. En med AMD-arkitektur og en med Intel-arkitektur. Dette ga oss problemer i starten. Derfor har deler av prosjektet med virtualisering blitt flyttet mot slutten. Mot slutten har vi fått en ny server med Intel-arkitektur. Da var muligheten for videre testing av live failovering og tjenester med livefunksjoner mulig.

Til sammen har vi fire servere. To Intel-maskiner koblet mot et SAN i cluster, en som står utenfor koblet mot clusterserverene og en server med 32-bits arkitektur som også står i nettverk med de andre.

Vi har i tillegg tilknytning mot eksterne navnetjenere i forbindelse med domenet mafoberg.net .

2.4. Programvare som ble brukt

Programvaren har vært basert på Microsoft. Vi har egne servere vi kjører på og derfor hatt mulighet til å styre det hele selv. Programvare på serverne har vært

- Microsoft Windows Server 2008 R2 x64
- Microsoft Windows Server 2008 x32

- Microsoft SCVMM 2008 R2
- Microsoft Exchange 2010
- Microsoft Windows 7 x64
- Microsoft Windows XP x32
- Apache Webserver

I tillegg er programvare for dokumentasjon brukt som beskrevet i 2.1. *Standarder og metoder*.

For å utvikle script har vi brukt Windows PowerShell ISE.

2.5. Fordeling av arbeid

Vi har vært to stykker som har arbeider med prosjektet. Vi har fordelt arbeidet i deler slik at vi kunne arbeide selvstendig. Vi har begge tatt tak i deler og stått som hovedansvarlig for hver del i tillegg til at vi har utiklet litt for hverandre. Fordeling har skjedd underveis.

Fordelingen har gått greit, og vi har hjulpet hverandre når den ene har kjørt seg fast. Vi har hatt tilgang til fjernstyring og har derfor også kunnet jobbe fra andre lokasjoner enn skolen.

2.6. Dokumentasjon

Vi har utviklet dokumenter etter maler etter maler som var ønskelig fra veileder. Følgende dokumenter er utviklet:

- Arbeidskontrakt.
- Forstudierapport.
- Systemkravdokument.
- Driftsdokumentasjon.
- Sluttrapport.
- Timelister, ukerapporter, møteinnkallinger, møtereferater og gantt-diagram.

Dokumentasjonsmateriale kan hentes fra biblioteket ved HiSTs avdeling i Trondheim. Den er også tilgjengelig fra <http://bachelor2010.mafo.net>

3. Gjennomføring av prosjektet

Gjennomføringen av prosjektet har gått greit, og vi har fått mye erfaring innen virtualiseringsteknologi og prosjekt for drift av virtuelle maskiner og styring av disse.

Arbeidsoppgavene har blitt fordelt greit. Grunnet forsinkelser i maskinvareleveranser har vi blitt forskjøvet litt i prosjektet, men har holdt oss innenfor tidsrammer. Det har blitt mye arbeid mot slutten, men motivasjonen har vært høy for å kunne gjennomføre dette slik at vi kom i mål.

Vi har utviklet et gantt-diagram som viser hvilke tidspunkt vi ble ferdig med delmål av prosjektet. Dette er blitt justert litt underveis etter hvert som forsinkelser har kommet i veien.

Grunnet sykdom og helseproblemer har også en av gruppe medlemmene slitt litt med å kunne utføre sine oppgaver til tider. Men ved god hjelp av samarbeidspartner er også dette blitt tatt godt hånd om, og hensyn har blitt vist.

Bedriften Mafoberg Systems AS har fått gjennomført igangsetting av overgang til virtualiseringsteknologi. Bedriften ville fått en innføring i hvordan maskiner kan styres og betjenes effektivt i et driftsmiljø, og kunne utført videre prosjekter for å komme helt over til virtualiseringsteknologi.

4. Videre arbeid

Prosjektet er beregnet på videreføring. Dette var kun en start av overgang til virtualiseringsteknologi. Arbeidet kan videreføres med nye prosjekter der fokus blir satt på eksisterende servere og tjenester som må konverteres til nye virtuelle maskiner.

Erfaringen med virtualisering og script i PowerShell kan vi som har kjørt prosjektet bruke videre i liknende prosjekter og i andre arbeid.

5. Referanser

Vi har brukt søkemotorer for å finne frem til stoff underveis i prosjektet i tillegg til Microsofts egne sider og forum hvor script har blitt diskutert.

Lenker:

- Microsoft generelt: <http://www.microsoft.com>
- Microsoft Technet: <http://technet.microsoft.com>
- Wikipedia: <http://www.wikipedia.com>
- Youtube: <http://www.youtube.com>
- Failover Clustering: http://technet.microsoft.com/en-us/library/cc732181%28WS.10%29.aspx#BKMK_Plan
- SCVMM: <http://www.microsoft.com/systemcenter/virtualmachinemanager/en/us/r2.aspx>
- Chapter 17. Processes, Services, Event Logs - Master-PowerShell With Dr. Tobias Weltner - PowerShell.com : <http://powershell.com/cs/blogs/ebook/archive/2009/04/10/chapter-17-processes-services-event-logs.aspx#starting-processes>

Bøker og PDF-filer vi har brukt:

- Windows Powershell 2 for Dummies
- Administrators Guide to Windows PowerShell Remoting (<http://powershell.com/cs/media/p/4908.aspx>)
- Sams Microsoft Exchange Server 2010 Unleashed
- DZone Refcards: Windows Powershell
- Introducing Windows Server 2008 r2

Andre kilder vi har brukt:

- Leksjoner i fag fra kurs på HiST. Fag som omhandler Microsoft Exchange og Windows Server 2008.
- Hjelp og lenker i programmene vi brukte.
- Feilsøkingmetoder og prøve-feilemetoder.

Møteinnkalling Gruppe 6E

Møtenummer: 01/2010

Innkalte: Stein Meisingseth
Bjørn Inge Vikhammermo Berg
Marius Foss

Møtested: Brygghuset 4 etg, AiTEL, Trondheim

Møtetid: 11.1.2010 Tid: 10.00-10.30

SAKSLISTE

Sak 001/2010 Innkalling og saksliste.

Sak 002/2010 Omfang av prosjektoppgaven.

Sak 003/2010 Stoff for prosjektoppgaven.

Sak 004/2010 Forstudierapportens omfang.

Ta kontakt med Bjørn (biberg@stud.hist.no), eller Marius Foss (mariusfo@stud.hist.no) dersom du ikke kan møte.

Velkommen!
-Bjørn Inge V. Berg

Møteinnkalling Gruppe 6E

Møtenummer: 02/2010

Innkalte: Stein Meisingseth
Bjørn Inge Vikhammerno Berg
Marius Foss

Møtested: Brygghuset 4 etg, AiTEL, Trondheim

Møtetid: 18.1.2010 Tid: 09.30-10.00

SAKSLISTE

Sak 005/2010 Godkjenning av innkalling, referat og saksliste.

Sak 006/2010 Godkjenning av forstudierapport med gantt-diagram

Sak 007/2010 Oppstart av systemkravdokumentasjon

Sak 008/2010 Eventuelt (maskinvare, webside...)

Ta kontakt med Bjørn (biberg@stud.hist.no), eller Marius Foss (mariusfo@stud.hist.no) dersom du ikke kan møte.

Velkommen!
Marius Foss

Møteinnkalling Gruppe 6E

Møtenummer: 03/2010

Innkalte: Stein Meisingseth
Bjørn Inge Vikhammerno Berg
Marius Foss

Møtested: Brygghuset 4 etg, AiTEL, Trondheim

Møtetid: 25.1.2010 Tid: 09.30-10.00

SAKSLISTE

Sak 009/2010 Godkjenning av innkalling, referat og saksliste.

Sak 010/2010 Godkjenning av revidert forstudierapport med Gantt-diagram

Sak 011/2010 Godkjenning av systemkravdokument.

Sak 012/2010 Eventuelt.

Ta kontakt med Bjørn (biberg@stud.hist.no), eller Marius Foss (mariusfo@stud.hist.no) dersom du ikke kan møte.

Velkommen!
-Bjørn Inge V.Berg

Møteinnkalling Gruppe 6E

Møtenummer: 04/2010

Innkalte: Stein Meisingseth
Bjørn Inge Vikhammerno Berg
Marius Foss

Møtested: Brygghuset 4 etg, AiTEL, Trondheim

Møtetid: 1.2.2010 Tid: 09.30-10.00

SAKSLISTE

Sak 013/2010 Godkjenning av innkalling, referat og saksliste.

Sak 014/2010 Godkjenning av systemkravdokument revisjon 2.

Sak 015/2010 Godkjenning av forstudierapport revisjon 3.

Sak 016/2010 Eventuelt (timeslister, driftsdokument og videre arbeid).

Ta kontakt med Bjørn (biberg@stud.hist.no), eller Marius Foss (mariusfo@stud.hist.no) dersom du ikke kan møte.

Velkommen!
-Marius Foss

Møteinnkalling Gruppe 6E

Møtenummer: 05/2010

Innkalte: Stein Meisingseth
Bjørn Inge Vikhammerno Berg
Marius Foss

Møtested: Brygghuset 4 etg, AiTEL, Trondheim

Møtetid: 8.2.2010 Tid: 09.30-10.00

SAKSLISTE

Sak 017/2010 Godkjenning av innkalling, referat og saksliste.

Sak 018/2010 Start og oppsett av driftsdokumentasjon.

Sak 019/2010 Gantt-diagrammet revidert.

Sak 020/2010 Eventuelt

- Maskinvare - får vi to maskiner eller må vi forholde oss til én?
Gjelder flytting av virtuelle maskiner mellom fysiske maskiner.
- IPadresser
- Video-dokumentasjon. Gjelder bare vanskelige deler?? Eller skal alt av installasjon recorderes?

Ta kontakt med Bjørn (biberg@stud.hist.no), eller Marius Foss (mariusfo@stud.hist.no) dersom du ikke kan møte.

Velkommen!
Marius Foss

Møteinnkalling Gruppe 6E

Møtenummer: 06/2010

Innkalte: Stein Meisingseth
Bjørn Inge Vikhammermo Berg
Marius Foss

Møtested: Brygghuset 4 etg, AiTEL, Trondheim

Møtetid: 15.2.2010 Tid: 09.30-10.00

SAKSLISTE

Sak 021/2010 Godkjenning av innkalling, referat og saksliste.

Sak 022/2010 Videre arbeid med driftsdokumentasjon.

Sak 023/2010 Eventuelt

- Får vi tilbake maskin vi mistet? Hva type?

Ta kontakt med Bjørn (biberg@stud.hist.no), eller Marius Foss (mariusfo@stud.hist.no) dersom du ikke kan møte.

Velkommen!
Marius Foss

Møteinnkalling Gruppe 6E

Møtenummer: 07/2010

Innkalte: Stein Meisingseth
Bjørn Inge Vikhammermo Berg
Marius Foss

Møtested: Brygghuset 4 etg, AiTEL, Trondheim

Møtetid: 1.3.2010 Tid: 12.30-13.00

SAKSLISTE

Sak 024/2010 Godkjenning av innkalling, referat og saksliste

Sak 025/2010 Driftsdokumentasjon første del.

- Ipadresser begynner å bli få. Skulle vi satt opp NAT-funksjon i nettverket vårt og dokumentert det?

Sak 026/2010 Eventuelt

- Problemer med Hyp-2. Ved restart henger den seg. Reinstallasjon for testing er lite aktuelt. Vi må da en del inn på serverrommet. Tynger arbeidet.

Ta kontakt med Bjørn (biberg@stud.hist.no), eller Marius Foss (mariusfo@stud.hist.no) dersom du ikke kan møte.

Velkommen!
Marius Foss

Møteinnkalling Gruppe 6E

Møtenummer: 08/2010

Innkalte: Stein Meisingseth
Marius Foss

Møtested: Brygghuset 4 etg, AiTEL, Trondheim

Møtetid: 15.3.2010 Tid: 11.30-12.00

SAKSLISTE

Sak 027/2010 Godkjenning av innkalling, referat og saksliste

Sak 028/2010 Driftsdokumentasjon installasjoner

Sak 029/2010 Eventuelt

- Ny maskin for hyp 2 ?
- Videre arbeid?

Ta kontakt med Bjørn (biberg@stud.hist.no), eller Marius Foss (mariusfo@stud.hist.no) dersom du ikke kan møte.

Velkommen!
Marius Foss

Møteinnkalling Gruppe 6E

Møtenummer: 09/2010

Innkalte: Stein Meisingseth
Bjørn Inge Vikhammermo Berg
Marius Foss

Møtested: Brygghuset 4 etg, AiTEL, Trondheim

Møtetid: 12.4.2010 Tid: 13.00-13.30

SAKSLISTE

Sak 030/2010 Godkjenning av innkalling, referat og saksliste

Sak 031/2010 Driftsdokumentasjon installasjoner

- SCVMM

Fortsettelse på dokumenter

Sak 032/2010 PowerShell og dokumentasjon

- Innholdsfortegnelse for PowerShell
- Hva må med av script?
- Hva må vi ha med av dokumentasjon?

Sak 033/2010 Eventuelt

- Nytt maskinutstyr. Failovering.
- Annet?

Ta kontakt med Bjørn (biberg@stud.hist.no), eller Marius Foss (mariusfo@stud.hist.no) dersom du ikke kan møte.

Velkommen!
Bjørn Inge Berg

Møteinnkalling Gruppe 6E

Møtenummer: 10/2010

Innkalte: Stein Meisingseth
Bjørn Inge Vikhammermo Berg
Marius Foss

Møtested: Brygghuset 4 etg, AiTEL, Trondheim

Møtetid: 26.4.2010 Tid: 13.00-13.30

SAKSLISTE

Sak 034/2010 Godkjenning av innkalling, referat og saksliste

Sak 035/2010 PowerShell og dokumentasjon

Sak 036/2010 Eventuelt

Ta kontakt med Bjørn (biberg@stud.hist.no), eller Marius Foss (mariusfo@stud.hist.no) dersom du ikke kan møte.

Velkommen!

Møtereferat

Møtenummer: 01/2010
Dato: 11.1.2010
Sted: 4. etg. AITeL
Møteleder: Bjørn Inge Vikhammermo Berg
Møtereferent: Marius Foss

Til stede: Stein Meisingseth
Marius Foss
Bjørn Inge Vikhammermo Berg
Frafall: Ingen

- Sak 001/2010 **Innkalling og saksliste:**
Innkalling og saksliste godkjent. Forandringer av hvordan sakslisten skal være ble kommentert. Følgende tre saker skal være med: Godkjenning av innkalling, godkjenning av referat og godkjenning av saksliste.
Fast innkalling til møter fremover i tid skal bli satt opp. Møter til og med 1. mars skjer hver uke kl 09:30. Deretter annenhver uke til og med påske. Deretter hver tredje uke til 10. mai. Arbeidskontrakt skal leveres innen neste møte.
- Sak 002/2010 **Omfang av prosjektoppgaven:**
Oppgavens omfang ble diskutert og revidert. Omfanget ble nå satt til...
Virtualisering med Hyper-V.
 - Windows Server 2008 R2 som styres av Powershell og SCVMM.
 - Exchange server som styres av Powershell.En fiktiv bedrift *MafoBerg System* beregnet på størrelse med 500 ansatte. Det skal forberedes for mulig HiST-organisering av systemet. Økonomisk overslag kan være greit å få med.
Plan for hele prosjektet skal utføres og leveres neste gang til veileder. Denne blir en del av forstudierapporten.
- Sak 003/2010 **Stoff for prosjektoppgaven:**
Stoff for prosjektet ble vedtatt som dokumentasjon fra Microsoft selv, og fra søking for Powershell. Vi søker under arbeidet. Planlegging også i forhold til HiST og derfor finne relevant informasjon til dette.
- Sak 004/2010 **Forstudierapportens omfang:**
Rapporten blir regnet til 10 til 25 sider.

Gruppe 6E
Marius Foss
Referatansvarlig

Bjørn Inge V. Berg
Møteansvarlig

Møtereferat

Møtenummer: 02/2010
Dato: 18.1.2010
Sted: 4. etg. AITeL
Møteleder: Marius Foss
Møtereferent: Bjørn Inge V.Berg

Til stede: Stein Meisingseth
Marius Foss
Bjørn Inge Vikhammermo Berg
Frafall: Ingen

Sak 005/2010 **Innkalling, referat og saksliste:**
Innkalling, referat og saksliste godkjent.

Sak 006/2010 **Godkjenning av forstudierapport:**
Forstudierapport var ikke lest igjennom av veileder enda. Vi ble instruert til å komme tilbake seinere på ettermiddagen, etter denne var lest igjennom. Torsdag 21.01. fikk vi videre tilbakemelding på forstudierapporten. Vi måtte gjøre det mer klarere hva prosjektet går ut på fra starten av. Beskrivelse av nytten og kostnadsanalysen må redigeres.

Sak 007/2010 **Oppstart av systemkravdokumentasjon:**
Veileder mente dette ville ta mindre tid enn de fem dagene som var satt opp i prosjektoversikten.

Sak 008/2010 **Eventuelt:**
Veileder ville se todo-lista vår for installasjon og dokumentasjon. Hardware ville ikke være tilgjengelig før om maks én uke.

Gruppe 6E
Marius Foss
Møteansvarlig

Bjørn Inge V. Berg
Referatansvarlig

Møtereferat

Møtenummer: 03/2010
Dato: 27.1.2010
Sted: 4. etg. AITeL
Møteleder: Bjørn Inge V.Berg
Møtereferent: Marius Foss

Til stede: Jostein Lund
Marius Foss
Bjørn Inge Vikhammermo Berg
Frafall: Stein Meisingseth

Sak 009/2010 **Innkalling, referat og saksliste:**
Innkalling, referat og saksliste godkjent.

Sak 010/2010 **Godkjenning av revidert forstudierapport og gantt:**
Jostein påpekte at vi kunne skrevet litt mer om applikasjonene vi skal bruke, hva de er for noe. Detaljene vil gjøre det klarere for de som ikke kjenner applikasjonene. Dette vil gi mer forståelse for de som leser dokumentasjonen. Gantt-diagrammet ble ikke kommentert.

Sak 011/2010 **Godkjenning av systemkravdokumentasjon:**
Her skulle også programvare beskrives litt mer. Detaljer burde komme frem. Ellers var denne grei.

Sak 012/2010 **Eventuelt:**
Det er satt i gang tiltak for å få tak i maskinvare. Men vi venter fortsatt.

Gruppe 6E
Bjørn Inge V. Berg
Møteansvarlig

Marius Foss
Referatansvarlig

Møtereferat

Møtenummer: 04/2010
Dato: 1.2.2010
Sted: 4. etg. AITeL
Møteleder: Marius Foss
Møtereferent: Bjørn Inge V.Berg

Til stede: Stein Meisingseth
Marius Foss
Bjørn Inge Vikhammermo Berg
Frafall: Ingen

Sak 013/2010 **Innkalling, referat og saksliste:**
Innkalling, referat og saksliste godkjent.

Sak 014/2010 **Godkjenning av systemkravdokument revisjon 2**
Systemkravet godkjent. Veileder påpekte at vi burde markere forandringene våre, slik at det blir lettere å se hva som var endret.

Sak 015/2010 **Godkjenning av forstudierapport revisjon 3**
Forstudierapporten godkjent. Veileder påpekte også her at vi burde markere forandringene våre, slik at det blir lettere å se hva som var endret.

Sak 016/2010 **Eventuelt**
Vi var noe usikker på hvordan timelistene skulle skrives. Tvilen stod til om de skulle reflektere vårt arbeid, eller et arbeid utført av en gruppe i arbeidslivet. Vi fikk greie på at listene skulle reflektere det arbeidet vi utførte. Vi spurte også om hvordan PowerShell-dokumentasjonen skulle foregå. Vi ble enige om at scriptene skulle legges ved som vedlegg, samt at bruk av de skulle forklares i driftsdokumentasjonen.

Gruppe 6E
Marius Foss
Møteansvarlig

Bjørn Inge V. Berg
Referatansvarlig

Møtereferat

Møtenummer: 05/2010
Dato: 8.2.2010
Sted: 4. etg. AITeL
Møteleder: Bjørn Inge V.Berg
Møtereferent: Marius Foss

Til stede: Stein Meisingseth
Marius Foss
Bjørn Inge Vikhammermo Berg
Frafall: Ingen

Sak 017/2010 **Innkalling, referat og saksliste.**
Innkalling, referat og saksliste godkjent.

Sak 018/2010 **Oppsett av driftsdokumentasjon og servere.**
Vår kladd av driftsdokumentasjon ble godkjent. Kladden har gitt oversikt over selve driftsdokumentasjonsinnholdet. Punkter Stein påpekte:

- Webadministrasjon for PowerShell-script ble godkjent som panel.
- Outlook-klienter skal testes med oppsett med og uten AD
- Outlook med integrert VPN i Windows 7
- Migrering fra fysisk Windows Server (2003) til virtuell ved hjelp av SCVMM funksjon (P2V)

Sak 019/2010 **Gantt-diagrammet revidert.**
Det blir ofte flere revisjoner av Gantt-diagram i løpet av en periode når tidsstrekk av uventede situasjoner oppstår. Gantt-diagrammet er godkjent.

Sak 020/2010 **Eventuelt**

- Vi fikk 1 maskin til å starte med. Maskinutstyret ligger under de kravene vi stilte, men det skulle holde til de første delene vi arbeider med. For overføringer mellom flere maskiner venter vi på mer maskinutstyr senere.
- IP-adresser ble satt til stein og HiSTs oppgave å lage en plan for. I mellomtiden bruker vi de adresser vi finner. Vi regner med at det kan bli et begrenset antall til å starte med.
- Vi må lage videodokumentasjon av de viktige delene av oppsett. Gjør det enklere for drift og installasjon for de som gjør dette første gang.

Gruppe 6E
Bjørn Inge V. Berg
Møteansvarlig

Marius Foss
Referatansvarlig

Møtereferat

Møtenummer: 06/2010
Dato: 15.2.2010
Sted: 4. etg. AITeL
Møteleder: Marius Foss
Møtereferent: Bjørn Inge Vikhammermo Berg

Til stede: Stein Meisingseth
Marius Foss
Bjørn Inge Vikhammermo Berg
Frafall: Ingen

Sak 021/2010 **Saksliste**
Innkalling, referat og saksliste godkjent.

Sak 022/2010 **Driftsdokumentasjon**
Veileder ville ha stoff når det blir ferdiggjort, det vil si ikke alt til slutt. På denne måten vil han bedre kunne gi tilbakemelding underveis.

Sak 023/2010 **Eventuelt**

- Maskinvare er kommet i hus, og Tor Halsan har begynt å konfigurere serverne. De vil derfor være klare i løpet av dagen.

Gruppe 6E
Marius Foss
Møteansvarlig

Bjørn Inge V. Berg
Referatansvarlig

Møtereferat

Møtenummer: 07/2010
Dato: 1.3.2010
Sted: 4. etg. AITeL
Møteleder: Bjørn Inge Vikhammermo Berg
Møtereferent: Marius Foss

Til stede: Stein Meisingseth
Marius Foss
Bjørn Inge Vikhammermo Berg
Frafall: Ingen

Sak 024/2010 **Saksliste**
Innkalling, referat og saksliste godkjent.

Sak 025/2010 **Dokumentasjon første del**

- Første del av dokumentasjon levert til Jostein. Han skulle ta seg av å lese igjennom og gi tilbakemelding om dette. Deretter fikk vi grei tilbakemelding om at dette var greit skrevet. En del korreksjoner må gjøres.
- To maskiner med forskjellig cpu-arkitektur kan ikke brukes til failover-clustering i Hyper-V. Den har også problem med at den låser seg. Dermed blir denne delen av prosjektet utsatt en ukes tid til inntil vi får byttet denne maskinen ut med en intel-maskin. Da kan vi kjøre denne prosjektdelen på nytt.

Sak 026/2010 **Eventuelt**

- IP-adressene kan bli for få. Hvis vi kommer ut over dette området, blir vi nødt til å sette opp en nat-løsning. Vi ble foreslått å sette opp RRAS for også å dokumentere dette. For at dette ikke skal bli kluss, tester vi dette først.

Gruppe 6E
Bjørn Inge V. Berg
Møteansvarlig

Marius Foss
Referatansvarlig

Møtereferat

Møtenummer: 08/2010
Dato: 15.3.2010
Sted: 4. etg. AITeL
Møteleder: Marius Foss
Møtereferent: Marius Foss
Møte avsluttet: 11:40

Til stede: Stein Meisingseth
Marius Foss

Frafall: Bjørn Inge Vikhammermo Berg

Sak 027/2010 **Saksliste**
Innkalling, referat og saksliste godkjent.

Sak 028/2010 **Dokumentasjon**

- Dokumentasjon av Exchange, Failoverclustering og SCVMM godkjent.
- SCVMM dokumentasjon ville Stein ha for testing selv. Ekstra kopi.
- Vi kjører videre med RRAS og det vi har fore, sier Stein.

Sak 029/2010 **Eventuelt**

- Stein skulle ta opp med Jostein angående ny maskin igjen. For dette skulle vi jo få i dag. Så vi får vente igjen!

Gruppe 6E
Marius Foss
Møteansvarlig

Marius Foss
Referatansvarlig

Møtereferat

Møtenummer: 09/2010
Dato: 12.4.2010 – 13:30
Sted: 4. etg. AITeL
Møteleder: Bjørn Inge Vikhammermo berg
Møtereferent: Marius Foss
Møte avsluttet: 13:23

Til stede: Stein Meisingseth
Marius Foss
Bjørn Inge Vikhammermo Berg

Frafall: Ingen

Sak 030/2010 **Saksliste**
Innkalling, referat og saksliste godkjent.

Sak 031/2010 **Dokumentasjon installasjoner**

- Dokumentasjon av SCVMM er tynt. Dette er på grunn av at den ikke er komplett til nå. Den var ok, men det er potensialer til forbedringer. Den burde inneholde hvordan å opprette ny maskin. Prerequisites for hva som må til osv.
- Vi har laget innholdsfortegnelse for PowerShell. Vi utvikler der etter. Stein synes dette var greit, og kom med forslag om forbedringer.

Sak 032/2010 **PowerShell og dokumentasjon**

- Hva må med av Script?
- Hva må med av PowerShell dokumentasjon?

Sak 033/2010 **Eventuelt**

- Nå har vi fått maskin som har lik cpu-arkitektur som Hyp-1. Intel Dual Core. Vi kommer til å bruke litt tid for å sette opp denne og få denne til å fungere med eksisterende server. Melde inn i cluster og domene. Deretter skal vi teste *Failovering* og *Live Migration*.
- Vi ligger etter med en del av prosjektet, og må planlegge hvordan vi skal komme etter nok.
- Vi må lage videorecording for hvordan failovering og live migration skal foregå.

Gruppe 6E
Bjørn Inge Vikhammermo Berg
Møteansvarlig

Marius Foss
Referatansvarlig

Møtereferat

Møtenummer: 10/2010
Dato: 26.4.2010 – 11:00
Sted: 4. etg. AITeL
Møteleder: Bjørn Inge Vikhammermo berg
Møtereferent: Marius Foss
Møte avsluttet: 11:22

Til stede: Stein Meisingseth
Marius Foss
Bjørn Inge Vikhammermo Berg

Frafall: Ingen

Sak 034/2010 **Saksliste**
Innkalling, referat og saksliste godkjent.

Sak 035/2010 **PowerShell og dokumentasjon**

- Første del av dokumentasjonen bør bearbeides. Forklaringene er ikke pedagogisk bra. Men ellers har vi fått en god start. Forbedringene kan gjøres frem til punkt 4.3.1.3.
- Begrep kan forklares bedre og settes opp mer oversiktlig.
- Bruke for eksempel Get-ChildItem som overskrift. Ofte brukte kommandoer kan settes opp slik.
- Arrays, forklare hva dette er
- Videre kan vi fortsette med script og kommandoer mot AD og DNS. Dette er et stort område.

Sak 036/2010 **Eventuelt**

- Når det gjelder andre deler, skal vi også scripte mot disse. Det er forholdsvis Marius som bør ta seg av en del av dette.
- Ellers nærmer vi oss slutten.

Gruppe 6E
Bjørn Inge Vikhammermo Berg
Møteansvarlig

Marius Foss
Referatansvarlig

Arbeidsarter

- 1. Arbeidskontrakt*
- 2. Forstudierapport*
- 3. Systemkravdokument*
- 4. Installasjon*
- 5. Konfigurasjon*
- 6. Driftsdokumentasjon*
- 7. Oppsett av Exchange 2010*
- 8. Arbeid med oppsett av Exchange 2010*
- 9. Powershellscript*
- 10. Windows 2008r2, hyper-v oppsett*
- 11. Oppsett av Hyper-V*
- 12. Forbereding, presentasjon*
- 13. Presentasjon*
- 14. Sluttrapport*
- 15. Innlevering av prosjekt*

Timelister for hovedprosjekt våren 2010

Gruppe 6E

UKE: 1, 2

Fra

07.01.2010

Til

17.01.2010

Arbeidsart refereres til ID-nummer i gantt-diagram.

Marius Foss

Arbeidsart	Mandag	Tirsdag	Onsdag	Torsdag	Fredag	Lørdag	Søndag
Fra uke 1				1	1		
1	2						
2	4			7	6	4	
	6	0	0	8	7	4	0

Sum uke: 25 Fra før: 0 Totalt hittil: 25

Bjørn Inge Vikhammermo Berg

Arbeidsart	Mandag	Tirsdag	Onsdag	Torsdag	Fredag	Lørdag	Søndag
Fra uke 1				1	2		
1	2						
2	4			6	6	4	2
	6	0	0	7	8	4	2

Sum uke: 27 Fra før: 0 Totalt hittil: 27

Timelister for hovedprosjekt våren 2010

Gruppe 6E

UKE: 3

Fra

18.01.2010

Til

24.01.2010

Arbeidsart refereres til ID-nummer i gantt-diagram.

Marius Foss

Arbeidsart	Mandag	Tirsdag	Onsdag	Torsdag	Fredag	Lørdag	Søndag
2	1			1	1		
3	6			6	4		
Annet rel.	1				2		
	8	0	0	7	7	0	0

Sum uke: 22 Fra før: 0 Totalt hittil: 22

Bjørn Inge Vikhammermo Berg

Arbeidsart	Mandag	Tirsdag	Onsdag	Torsdag	Fredag	Lørdag	Søndag
2	1				1		
3	6			6	6		
Annet rel.	1			1	1		
	8	0	0	7	8	0	0

Sum uke: 23 Fra før: 0 Totalt hittil: 23

Timelister for hovedprosjekt våren 2010

Gruppe 6E

UKE: 4

Fra

25.01.2010

Til

31.01.2010

Arbeidsart refereres til ID-nummer i gantt-diagram.

Marius Foss

Arbeidsart	Mandag	Tirsdag	Onsdag	Torsdag	Fredag	Lørdag	Søndag
3				6	6		
Annet rel.	2						
	2	0	0	6	6	0	0

Sum uke: 14 Fra før: 0 Totalt hittil: 14

Bjørn Inge Vikhammermo Berg

Arbeidsart	Mandag	Tirsdag	Onsdag	Torsdag	Fredag	Lørdag	Søndag
3				6	6	1	1
Annet rel.	1		1				
	1	0	1	6	6	1	1

Sum uke: 16 Fra før: 0 Totalt hittil: 16

Timelister for hovedprosjekt våren 2010

Gruppe 6E

UKE: 5

Fra

01.02.2010

Til

07.02.2010

Arbeidsart refereres til ID-nummer i gantt-diagram.

Marius Foss

Arbeidsart	Mandag	Tirsdag	Onsdag	Torsdag	Fredag	Lørdag	Søndag
3	2						
6	6			6	1		
10					3		
Annet rel.					2		
	8	0	0	6	6	0	0

Sum uke: 20 Fra før: 0 Totalt hittil: 20

Bjørn Inge Vikhammermo Berg

Arbeidsart	Mandag	Tirsdag	Onsdag	Torsdag	Fredag	Lørdag	Søndag
9			1	8	8		
10	6						
Annet rel.	2					1	
	8	0	1	8	8	1	0

Sum uke: 26 Fra før: 0 Totalt hittil: 26

Timelister for hovedprosjekt våren 2010

Gruppe 6E

UKE: 6

Fra

08.02.2010

Til

014.02.2010

Arbeidsart refereres til ID-nummer i gantt-diagram.

Marius Foss

Arbeidsart	Mandag	Tirsdag	Onsdag	Torsdag	Fredag	Lørdag	Søndag
4	2			2	2		
6	1			1	1		
10	2			1	2		
Annet rel.	2						
	7	0	0	4	5	0	0

Sum uke: 16 Fra før: 0 Totalt hittil: 16

Bjørn Inge Vikhammermo Berg

Arbeidsart	Mandag	Tirsdag	Onsdag	Torsdag	Fredag	Lørdag	Søndag
4	2						
10	5						
Annet rel.	2	2					
	9	2	0	0	0	0	0

Sum uke: 11 Fra før: 0 Totalt hittil: 11

Timelister for hovedprosjekt våren 2010

Gruppe 6E

UKE: 7

Fra

15.02.2010

Til

021.02.2010

Arbeidsart refereres til ID-nummer i gantt-diagram.

Marius Foss

Arbeidsart	Mandag	Tirsdag	Onsdag	Torsdag	Fredag	Lørdag	Søndag
4	2			1			
5	3			1	2	2	3
6	2				2	3	3
10							
Annet rel.							
	7	0	0	2	4	5	6

Sum uke: 24 Fra før: 0 Totalt hittil: 24

Bjørn Inge Vikhammermo Berg

Arbeidsart	Mandag	Tirsdag	Onsdag	Torsdag	Fredag	Lørdag	Søndag
4	2						
5	1			3	1		
6	3			2	2		
10	2			3	4		
Annet rel.							
	8	0	0	8	7	0	0

Sum uke: 23 Fra før: 0 Totalt hittil: 23

Timelister for hovedprosjekt våren 2010

Gruppe 6E

UKE: 8

Fra

22.02.2010

Til

28.02.2010

Arbeidsart refereres til ID-nummer i gantt-diagram.

Marius Foss

Arbeidsart	Mandag	Tirsdag	Onsdag	Torsdag	Fredag	Lørdag	Søndag
4	2						
5							
6	3			2			4
8	2			1	2		
10				4	3		
Annet rel.							1
	7	0	0	7	5	0	5

Sum uke: 24 Fra før: 0 Totalt hittil: 24

Bjørn Inge Vikhammermo Berg

Arbeidsart	Mandag	Tirsdag	Onsdag	Torsdag	Fredag	Lørdag	Søndag
5							
6	3			5	5		
9			3				
10	4			2	1		
Annet rel.					2	2	1
	7	0	3	7	8	2	1

Sum uke: 28 Fra før: 0 Totalt hittil: 28

Timelister for hovedprosjekt våren 2010

Gruppe 6E

UKE: 9

Fra

01.03.2010

Til

07.03.2010

Arbeidsart refereres til ID-nummer i gantt-diagram.

Marius Foss

Arbeidsart	Mandag	Tirsdag	Onsdag	Torsdag	Fredag	Lørdag	Søndag
4	1						
6	4			4	4		3
8	2			2	2		1
Annet rel.	1						
	8	0	0	6	6	0	4

Sum uke: 24 Fra før: 0 hittil: 24

Bjørn Inge Vikhammermo Berg

Arbeidsart	Mandag	Tirsdag	Onsdag	Torsdag	Fredag	Lørdag	Søndag
9	1				4		
10	7			8	4		
Annet rel.					2		
	8	0	0	8	10	0	0

Sum uke: 26 Fra før: 0 hittil: 26

Timelister for hovedprosjekt våren 2010

Gruppe 6E

UKE: 10

Fra

08.03.2010

Til

14.03.2010

Arbeidsart refereres til ID-nummer i gantt-diagram.

Marius Foss

Arbeidsart	Mandag	Tirsdag	Onsdag	Torsdag	Fredag	Lørdag	Søndag
4	1			1	3		
6	4			4	2		2
8	3			3			
10					4		
Annet rel.							
	8	0	0	8	9	0	2

Sum uke: 27 Fra før: 0 Totalt hittil: 27

Bjørn Inge Vikhammermo Berg

Arbeidsart	Mandag	Tirsdag	Onsdag	Torsdag	Fredag	Lørdag	Søndag
4	4						
6	1			6	6		
10	2			2	2		
Annet rel.							
	7	0	0	8	8	0	0

Sum uke: 23 Fra før: 0 Totalt hittil: 23

Timelister for hovedprosjekt våren 2010

Gruppe 6E

UKE: 11

Fra

15.03.2010

Til

21.03.2010

Arbeidsart refereres til ID-nummer i gantt-diagram.

Marius Foss

Arbeidsart	Mandag	Tirsdag	Onsdag	Torsdag	Fredag	Lørdag	Søndag
4				1			
6				2	3		
10				3	4		
Annet rel.					2		
	0	0	0	6	9	0	0

Sum uke: 15 Fra før: 0 Totalt hittil: 15

Bjørn Inge Vikhammermo Berg

Arbeidsart	Mandag	Tirsdag	Onsdag	Torsdag	Fredag	Lørdag	Søndag
4							
6					8		
10							
Annet rel.							
	0	0	0	0	8	0	0

Sum uke: 8 Fra før: 0 Totalt hittil: 8

Timelister for hovedprosjekt våren 2010

Gruppe 6E

UKE: 12

Fra

22.03.2010

Til

28.03.2010

Arbeidsart refereres til ID-nummer i gantt-diagram.

Marius Foss

Arbeidsart	Mandag	Tirsdag	Onsdag	Torsdag	Fredag	Lørdag	Søndag
4							
5							
6	2				2		
10	3				3		
Annet rel.							
	5	0	0	0	0	5	0

Sum uke: 10 Fra før: 0 Totalt hittil: 10

Bjørn Inge Vikhammermo Berg

Arbeidsart	Mandag	Tirsdag	Onsdag	Torsdag	Fredag	Lørdag	Søndag
4		2					
6							
9							
10							
Annet rel.							
	0	2	0	0	0	0	0

Sum uke: 2 Fra før: 0 Totalt hittil: 2

Timelister for hovedprosjekt våren 2010

Gruppe 6E

UKE: 13

Fra

29.03.2010

Til

04.04.2010

Arbeidsart refereres til ID-nummer i gantt-diagram.

Marius Foss

Arbeidsart	Mandag	Tirsdag	Onsdag	Torsdag	Fredag	Lørdag	Søndag
6				1	1		
8							
10				2	1		
Annet rel.				1	1		
	0	0	0	4	3	0	0

Sum uke: 7 Fra før: 0 Totalt hittil: 7

Bjørn Inge Vikhammermo Berg

Arbeidsart	Mandag	Tirsdag	Onsdag	Torsdag	Fredag	Lørdag	Søndag
6				2	2		
10							
Annet rel.							
	0	0	0	2	2	0	0

Sum uke: 4 Fra før: 0 Totalt hittil: 4

Timelister for hovedprosjekt våren 2010

Gruppe 6E

UKE: 14

Fra

05.04.2010

Til

11.04.2010

Arbeidsart refereres til ID-nummer i gantt-diagram.

Marius Foss

Arbeidsart	Mandag	Tirsdag	Onsdag	Torsdag	Fredag	Lørdag	Søndag
4			1				
6		2	2	4	3	2	
8			7	4	4		
10							
Annet rel.					1		
	0	2	10	8	8	2	0

Sum uke: 30 Fra før: 0 hittil: 30

Bjørn Inge Vikhammermo Berg

Arbeidsart	Mandag	Tirsdag	Onsdag	Torsdag	Fredag	Lørdag	Søndag
4		2					
6		6	8	7	6		
Annet rel.							
	0	8	8	7	6	0	0

Sum uke: 29 Fra før: 0 hittil: 29

Timelister for hovedprosjekt våren 2010

Gruppe 6E

UKE: 15

Fra

12.04.2010

Til

18.04.2010

Arbeidsart refereres til ID-nummer i gantt-diagram.

Marius Foss

Arbeidsart	Mandag	Tirsdag	Onsdag	Torsdag	Fredag	Lørdag	Søndag
4	2						
5		1	3	3	4		
6		2	3	3	2		
8	2	1	3	3			
Annet rel.	1	1					
	5	5	9	9	6	0	0

Sum uke: 34 Fra før: 0 Totalt hittil: 34

Bjørn Inge Vikhammermo Berg

Arbeidsart	Mandag	Tirsdag	Onsdag	Torsdag	Fredag	Lørdag	Søndag
6	4	4	4	4	2		
9	5	5	5	5	4		
Annet rel.							
	9	9	9	9	6	0	0

Sum uke: 42 Fra før: 0 Totalt hittil: 42

Timelister for hovedprosjekt våren 2010

Gruppe 6E

UKE: 16

Fra

19.04.2010

Til

25.04.2010

Arbeidsart refereres til ID-nummer i gantt-diagram.

Marius Foss

Arbeidsart	Mandag	Tirsdag	Onsdag	Torsdag	Fredag	Lørdag	Søndag
4	2	2	1				
5	3	3		6	6		
6	1	1	4	3	3		
8	1	3	3				
Annet rel.	2						
	9	9	8	9	9	0	0

Sum uke: 44 Fra før: 0 Totalt hittil: 44

Bjørn Inge Vikhammermo Berg

Arbeidsart	Mandag	Tirsdag	Onsdag	Torsdag	Fredag	Lørdag	Søndag
6	5	5	5	5	5		
9	3	3	3	3	3		
Annet rel.							
	8	8	8	8	8	0	0

Sum uke: 40 Fra før: 0 Totalt hittil: 40

Timelister for hovedprosjekt våren 2010

Gruppe 6E

UKE: 17

Fra

26.04.2010

Til

02.05.2010

Arbeidsart refereres til ID-nummer i gantt-diagram.

Marius Foss

Arbeidsart	Mandag	Tirsdag	Onsdag	Torsdag	Fredag	Lørdag	Søndag
5	4	1			3		
6	1	1	3	3	4		
9		3	3	3			
10	1	3			1		
Annet rel.	2						
	8	8	6	6	8	0	0

Sum uke: 36 Fra før: 0 hittil: 36

Bjørn Inge Vikhammermo Berg

Arbeidsart	Mandag	Tirsdag	Onsdag	Torsdag	Fredag	Lørdag	Søndag
6	5	5	5	5	5		
9	3	3	3	3	3		
Annet rel.							
	8	8	8	8	8	0	0

Sum uke: 40 Fra før: 0 hittil: 40

Timelister for hovedprosjekt våren 2010

Gruppe 6E

UKE: 18

Fra

03.05.2010

Til

09.05.2010

Arbeidsart refereres til ID-nummer i gantt-diagram.

Marius Foss

Arbeidsart	Mandag	Tirsdag	Onsdag	Torsdag	Fredag	Lørdag	Søndag
5							
6		3			1		
9				2	2		
10							
13	2					2	3
Annet rel.	4						
	6	3	0	2	3	2	3

Sum uke: 19 Fra før: 0 Totalt hittil: 19

Bjørn Inge Vikhammermo Berg

Arbeidsart	Mandag	Tirsdag	Onsdag	Torsdag	Fredag	Lørdag	Søndag
5						2	3
13							
9					2	2	1
Annet rel.	4						
	4	0	0	0	2	4	4

Sum uke: 14 Fra før: 0 Totalt hittil: 14

Timelister for hovedprosjekt våren 2010

Gruppe 6E

UKE: 19

Fra

10.05.2010

Til

16.05.2010

Arbeidsart refereres til ID-nummer i gantt-diagram.

Marius Foss

Arbeidsart	Mandag	Tirsdag	Onsdag	Torsdag	Fredag	Lørdag	Søndag
6	4		1	2			
8							
9	2		3	3			
10				2			
Annet rel.							
	6	0	4	7		0	0

Sum uke: 17 Fra før: 0 Totalt hittil: 17

Bjørn Inge Vikhammermo Berg

Arbeidsart	Mandag	Tirsdag	Onsdag	Torsdag	Fredag	Lørdag	Søndag
4	8	12	6	4			
Annet rel.							
	8	12	6	4		0	0

Sum uke: 30 Fra før: 0 Totalt hittil: 30

Timelister for hovedprosjekt våren 2010

Gruppe 6E

UKE: 20

Fra

17.04.2010

Til

23.05.2010

Arbeidsart refereres til ID-nummer i gantt-diagram.

Marius Foss

Arbeidsart	Mandag	Tirsdag	Onsdag	Torsdag	Fredag	Lørdag	Søndag
6	2				1		
8							
9	3	3					
13			2				
14							4
Annet rel.							
	5	3		2	0	1	0
							4

Sum uke: 15 Fra før: 0 Totalt hittil: 15

Bjørn Inge Vikhammermo Berg

Arbeidsart	Mandag	Tirsdag	Onsdag	Torsdag	Fredag	Lørdag	Søndag
6							5
12		5	2				
13			1				
Annet rel.							
	0	5		3	0	0	0
							5

Sum uke: 13 Fra før: 0 Totalt hittil: 13

Timelister for hovedprosjekt våren 2010

Gruppe 6E

UKE: 21

Fra

24.04.2010

Til

25.05.2010

Arbeidsart refereres til ID-nummer i gantt-diagram.

Marius Foss

Arbeidsart	Mandag	Tirsdag	Onsdag	Torsdag	Fredag	Lørdag	Søndag
6	3						
9	1						
15	4	6					
Annet rel.							
	8	6	0	0	0	0	0

Sum uke: 14 Fra før: 0 Totalt hittil: 14

Bjørn Inge Vikhammermo Berg

Arbeidsart	Mandag	Tirsdag	Onsdag	Torsdag	Fredag	Lørdag	Søndag
9	4						
15	4	6					
Annet rel.							
	8	6	0	0	0	0	0

Sum uke: 14 Fra før: 0 Totalt hittil: 14

Rapport Uke 1 og 2 Prosjektgruppe 6E

Denne uka har vi tatt for oss oppstart. Opprettelse av noen dokumenter, kontrakt, lister, maler og dokumenter. Også begynte vi med å få tak i stoff og linker til informasjon vi trengte.

Rapport Uke 3 Prosjektgruppe 6E

Jobbet med systemkravdokument for det aller meste. I tillegg til revisjon av forstudierapporten.

I tillegg har vi satt oss mer fore for hva vi skal gjøre med to-do lista. Klargjort system for orden i prosjektet. Oppdatert hjemmeside.

Rapport Uke 4 Prosjektgruppe 6E

Denne uka nærmet seg brudd av kontrakten hvor vi ikke har vært alt for produktive.

Vi har blant annet jobbet med revisjoner av både systemkravdokument og forstudierapport. På forstudierapporten ble kostnytte revidert, og i systemkravet er det dokumentert mer om programvarene.

Jobbet også litt med forberedelse til driftsdokumentasjon.

Rapport Uke 5 Prosjektgruppe 6E

Utvikling av Powershell miljø. Revisjonsarbeid av systemkravdokument igjen som ikke er helt ferdig. Flyttet informasjon om software til systemkravdokumentet. Revisjoner Gantt.

Brukt tid på utvikle deler av driftsdokumentasjonen. Maskinvareoppsett kladd. Testing av muligheter i Windows Server 2008.

Rapport Uke 6 Prosjektgruppe 6E

Arbeidet har gått ut på å jobbe videre med oversikt over software vi skulle installere. Se på forskjeller fra andre systemer installere dem og få iverblikk hvilke retninger vi må gå i når vi skal sette opp softwaren.

Rapport Uke 7 Prosjektgruppe 6E

I denne uka merka vi en del problemer. Starten begynte med sykdom og skader for en av oss. Dette gikk ut over arbeidsmengden som måtte taes igjen seinere. Dette ble blant annet helga brukt til. Men videre arbeid med installasjon og konfigurering, screenshots av deler for dokumentasjon var det vi fortsatte med.

Rapport Uke 8 Prosjektgruppe 6E

Denne Uka har gått med på testing av Hyper-V og testing av NAS. Masse problemer og grunnlag for at dette ikke kunne gå. I tillegg har vi hatt problemer med AMD-maskinen som har tatt tid fra oss. Dette håper vi å kunne løse med et bytte.

Rapport Uke 9 Prosjektgruppe 6E

Startet med møte hvor vi fikk beskjed om at vi nok en gang skal få en ny maskin. Dette vil ta en uke eller 2. I mellomtiden må vi omstrukturere arbeidet litt for hva vi har mulighet til å gjøre og få ferdig. Vi har en fortsatt domkumentasjon å skrive, og installasjon av noen tjenester på virtuelle maskiner å teste ut.

Rapport Uke 10 Prosjektgruppe 6E

Vi har stadig tjenere å sette opp. Flere virtuelle maskiner med tjenester hver for seg. Omgjøring av nettet blir tatt når vi kjører opp RRAS-rollen. Vi har laget templatemaskin for enkelt nytt oppsett. En virtuell for scvmm som egentlig burde vært fysisk. Men i dette prosjektet kjører vi det slik.

Rapport Uke 11 Prosjektgruppe 6E

Denne uke har vi holdt på med SCVMM. Litt feilplasseringer av tjenere har laget litt kaos. I tillegg til ustabil server som har tatt beslag på tida fordi vi ikke har fått fysisk tilgang en del av gangene. De trengte fysisk restart med mer. Dermed har vi stått der med kun en server mot clusteret. Dette har ikke vært bra nok, og SCVMM har blitt hengende etter. Bjørn var på foretningsreise første del av uka.

Rapport Uke 12 Prosjektgruppe 6E

SCVMM krever mye tid for samkjøring med servere. Serverne er igjen et problem siden domenekontrolleren befinner seg på clusteret hvor også dns-serveren er. Vi har litt tilgangsproblemer noe vi må fortsette med neste uke.

I tillegg kjører vi PowerShell og setter opp enkelte script som plukker ut informasjon. Bjørn var bortreist på klassetur denne uka

Rapport Uke 13 Prosjektgruppe 6E

Uken går med til påske og ferie. Vi gjør forefallende arbeid. Dette gjør vi remote hjemmefra. Klargjøring av fysisk maskin til test.

Rapport Uke 14 Prosjektgruppe 6E

Denne uken har vi gjort flere ting samtidig. Mens vi tester ut SCVMM, oppretter maskiner, flytter og dokumenterer. Det som tar mest tid setter vi til å gå over natten. Konvertering fra fysisk maskin til virtuell. Samtidig kjører vi Edge-serveren opp parallelt på en annen maskin. Samtidig tester vi script i PowerShell mot servere som er oppe. Trekker ut informasjon.

Rapport Uke 15 Prosjektgruppe 6E

Det gikk mye på PowerShell med innledning. Dokumentasjon av syntaks og generell bruk av scriptsystemet. Exchange testing og oppsett. Noe som har tatt tid med treg maskinvare. Overgang til SCVMM og bruk av virtuelle maskiner.

Rapport Uke 16 Prosjektgruppe 6E

Vi har arbeidet med Exchange og edge. Testet ut funksjonalitet så langt vi hadde mulighet til forhold til begrensninger. I tillegg har vi installert og testet litt av Forefront Security. I tillegg har vi testet og laget en god del Powershellscript. Prøvd ut en hel del funksjonalitet med PowerShell generelt og med remoting.

Rapport Uke 17 Prosjektgruppe 6E

Vi har jobbet med PowerShell generelt, og mot AD og SCVMM. I tillegg er det brukt mer tid på SCVMM, og ferdiggjøring av bruk med SCVMM gui. Litt bruk av PowerShell for SCVMM står fortsatt igjen.

Rapport Uke 18 Prosjektgruppe 6E

Mye tid gikk med til forberedelse til eksamen. Ellers var det jobbing med presentasjon og vårt nettbaserte kontrollpanel og powershellscripts for SCVMM og generelt.

Rapport Uke 19 Prosjektgruppe 6E

Arbeid med det nettbaserte kontrollpanelet fortsetter. Powershelldokumentasjon er ca ferdig.

Rapport Uke 20 Prosjektgruppe 6E

Mye av denne uka har gått bort til 17.mai og eksamen, samt presentasjon av bachelorprosjektet.
I tillegg har vi skrevet ferdig sluttrapport for prosjektet, og starter med klargjøring av all dokumentasjon.

Rapport Uke 21 Prosjektgruppe 6E

Uka går med til forberedelse til innlevering av bachelorprosjekt.

Utdrag fra gannt-diagrammet

Arbeidsarter	Starttidspunkt	Slutttidspunkt
Arbeidskontrakt	08.01.2010 08:00	08.01.2010 17:00
Forstudierapport	11.01.2010 08:00	15.01.2010 17:00
Systemkravdokument	15.01.2010 08:00	22.01.2010 17:00
Installasjon	11.02.2010 08:00	26.02.2010 17:00
Konfigurering	11.02.2010 08:00	30.04.2010 17:00
Driftsdokumentasjon	27.01.2010 08:00	18.05.2010 17:00
Exchangeoppsett Windows Server 2008r2, Exchange Server 2010	01.03.2010 08:00	05.05.2010 17:00
Powershellscript for windows 2008r2 og exchange	01.02.2010 08:00	07.05.2010 17:00
Windows 2008r2 med hyper-v-oppsett og Scvmm	11.02.2010 08:00	23.04.2010 17:00
Forbereding av presentasjon	03.05.2010 08:00	05.05.2010 17:00
Presentasjon	06.05.2010 08:00	06.05.2010 17:00
Sluttrapport	10.05.2010 08:00	21.05.2010 17:00
Innlevering av prosjekt	24.05.2010 08:00	24.05.2010 17:00