

Emil Antoni Brasø
Endre Aalrust

En automatisert reise fra datarom til sky

Mai 2020

NTNU

Norges teknisk-naturvitenskapelige universitet.
Fakultet for informasjonsteknologi og elektroteknikk
Institutt for datateknologi og informatikk

Bacheloroppgave

2020



Emil Antoni Brasø
Endre Aalrust

En automatisert reise fra datarom til sky

Bacheloroppgave
Mai 2020

NTNU

Norges teknisk-naturvitenskapelige universitet.
Fakultet for informasjonsteknologi og elektroteknikk
Institutt for datateknologi og informatikk



Kunnskap for en bedre verden

Innholdsfortegnelse

Sluttrapport _____	3
Forstudierappport _____	21
Designrapport _____	47
Driftsrapport _____	81



SLUTTRAPPORT

En automatisert reise fra datarom til sky

Emil Antoni Brasø, Endre Aalrust

Revisjonslogg

DATO	VERSJON	BESKRIVELSE	FORFATTER
15.05.20	1.0	Opprettelse av dokument	Endre, Emil

Kontaktinfo

Emil Antoni Brasø - Bachelorstudent, Informatikk, drift av datasystemer

emilab@stud.ntnu.no | emilantoni@hotmail.no

Telefon: +47 46805784

Endre Aalrust - Bachelorstudent, Informatikk, drift av datasystemer

endreaal@stud.ntnu.no | aalrust.endre@gmail.com

Telefon: +47 48097056

Forord

OM DE INVOLVERTE OG STUDIERETNINGEN

Denne rapporten er skrevet i sammenheng med Bachelorprosjektet *En automatisert reise fra datarom til sky*. Bachelorprosjektet, eller oppgaven, er utarbeidet av Emil Antoni Brasø og Endre Aalrust, i samarbeid med IT-selskapet Sopra Steria.

«Hos oss lærer du det markedet etterspør av drifts-teknisk IKT-kunnskap ... Videre er tema som Linux, Windows server, «Cloud Computing» og overvåkning og sikkerhet i digital infrastruktur representert ...»¹

Prosjektgruppens medlemmer var Emil Antoni Brasø og Endre Aalrust. Begge to var studenter på linja *Informatikk, Drift av Datasystemer*. Dette studieprogrammet tilhørte *Institutt for datateknologi og informatikk, ved Fakultet for informasjonsteknologi og elektroteknikk på NTNU*.

Sitatet i starten av dette forordet forteller mye av grunnlaget som lå bak valget av oppgave. Sammen med oppdragsgiver ble denne oppgaven utarbeidet med den hensikt å lære mer om hvordan en bedrift migrerer sine systemer fra et on-premises miljø til et skybasert miljø. Dette er et veldig aktuelt tema da bruken av skybaserte systemer har økt betraktelig de siste årene, og forventes å øke enda mer². Selve prosessen med å migrere et system er kjernen i denne oppgaven, men oppgaven ser også på hvordan miljøet i skyen etter endt migrering burde settes opp.

OM PROSJEKTET OG HVA SOM ER LÆRT

Oppgaven er utformet som en brukerguide på hvordan prosessen med å migrere kan gjennomføres. Derfor diskuteres valg av verktøy og tjenester, hovedsakelig under migrering og etter migrering. For å gjøre oppgaven så reell som mulig var det viktig å få på plass et velfungerende system snarest mulig etter migrering, dette var blant annet for å unngå tap av effektivitet som følge av migreringsprosessen.

Automasjon og IaC (Infrastructure as Code) var tema som ble vektlagt i oppgaven, og ble ansett som sentrale virkemidler for å oppnå de ovennevnte målene. Mange av funksjonene og tjenestene som ble iverksatt gjennom prosjektets løp, hadde mulighet for å bli helt eller delvis automatisert. Automasjon og IaC er svært nyttig og relevant fordi det gjør det lettere og raskere å gjenta disse oppgavene senere. For sluttbrukers del anses dette som en viktig fordel.

Ved prosjektets oppstart hadde prosjektgruppen grunnleggende kunnskap om skytjenesten Azure, og nettopp derfor ble denne plattformen valgt som miljø i denne oppgaven. Målet var til dels at prosjektgruppen skulle utvikle en god praktisk forståelse av hva Azure, og teknologien rundt det, faktisk innebærer. Underveis i prosjektet ble det åpenbart hvor mange funksjoner Azure faktisk tilbyr, og det ble til tider vanskelig å separere hvilke funksjoner som var hensiktsmessig å iverksette i forhold til oppgaven. Dette er særlig med tanke på at nødvendige tjenester varierer sterkt mellom ulike bedrifter. De avgrensningene som ble gjort, samt utvalget av tjenester, er forsøkt argumentert der det er relevant.

¹ <https://www.ntnu.no/studier/itbainfodr>

² (Liu, 2019)

Studiet har lært prosjektgruppen hva som teknisk sett er nødvendig for et «*bare-minimum*» system. Dette prosjektet tillot derimot rom for egen tolkning, og derfor valgte prosjektgruppen å utforske de ulike mulighetene som Azure tilbyr. Dette ble holdt innenfor rimelige grenser, og etter rammene som ble definert tidlig i prosjektet. Særlig derfor var det viktig å holde en god og åpen dialog med veiledere både i bedrift, og på NTNU.

Ved prosjektets slutt sitter prosjektgruppen igjen med kunnskap om hvordan man setter opp ett fungerende system i Azure, samt hvordan man kan migrere en bedrifts eksisterende systemer. IaC og automasjon har blitt nevnt som sentrale virkemidler. Dette er kanskje noe av det prosjektgruppen har lært aller mest om i løpet av prosjektets gang. Migreringsprosessen er noe prosjektgruppen er blitt veldig trygge på, og gjennom prøving og feiling har de lært hva som kan gå galt og hvorfor det går galt, slik at feil kan rettes opp og prosessen gjennomføres med sikkerhet. Overordnet er det blitt klart at Azure tilbyr utrolig mange muligheter, og prosjektgruppen gleder seg veldig til å lære mer i årene som kommer.

TAKK TIL

Prosjektgruppen ønsker å rette en stor takk til alle som har vært til hjelp under utførelsen av dette prosjektet. Prosjektet ville ikke vært det samme uten deres hjelp!

- Rune Haugen og Kristian Aamodt, våre veiledere i Sopra Steria. Takk for all veiledning, tilbakemelding, IT-teknisk innsikt og disponering av miljø i Azure.
- Jostein Lund og Stein Meisingseth, våre veiledere og professorer fra NTNU. Takk for eksepsjonell veiledning gjennom tre år med skolegang, upåklagelig entusiasme i alle emner, og for generelt godt lynne!
- Kristian Grannes, IT-konsulent for Moveo AS. Takk for din tekniske innsikt og anbefalinger når det kom til valg av verktøy i oppgaven.
- Marthe Torp og resten av dere på Brattørkaia. Takk for at dere tok oss imot med åpne armer!
- Camilla Vågan, tante. Takk for korrektur og motiverende ord!
- Alle medstudenter. Takk for et supert studiemiljø, vi har utviklet oss godt sammen, vi håper på reunion om noen år!

Innholdsfortegnelse - Sluttrapport

Revisjonslogg.....	1
Kontaktinfo	2
Forord.....	3
Innholdsfortegnelse - Sluttrapport	5
1 Oppgavebeskrivelse	6
2 Hvordan oppgaven ble løst	7
2.1 Maskinvare.....	7
2.2 Verktøy og programvare	8
2.3 Arbeidsfordeling under prosjektet.....	10
3 Gjennomføring av prosjektet	11
3.1 Effektmål	12
3.2 Resultatmål	13
3.3 Prosessmål	15
4 Videre arbeid.....	16
5 Endringer i løpet av prosjektet	16

1 Oppgavebeskrivelse

Prosjektet tok utgangspunkt i å utarbeide en optimalisert og effektiv fremgangsmåte for hele migreringsprosessen. Videre skulle gevinsten, eller ROI (Return Of Investment), ses på i forbindelse med dette. Det er nødvendig å ta for seg slike utfordringer for at en skal kunne ta informerte beslutninger i et marked som går over til sky i svært høyt tempo. Det er følgelig viktig å ikke kun se på IT-tekniske gevinster, et fokus må rettes mot sluttbruker også.

Oppgaven utføres i samarbeid med bedriften Sopra Steria som er basert i Oslo, representert ved Rune Haugen og Kristian Ellingsen Aamodt.

Når det kommer til prosjektets format skulle dette fungere som en driftsoppgave hvor det ble satt opp et helhetlig system. Følgende mål ble definert gjennom møter med oppdragsgiver, og for en mer detaljert beskrivelse henvises til forstudierapporten³ og designrapporten⁴.

- En fremgangsmåte som beskriver selve prosessen med å migrere dagens løsning fra on-premises til en hel eller delvis skybasert løsning.
- Ett system som utnytter skybasert teknologi
- En analyse av typen løsning; hel eller delvis og hva som vil gagne bedriften i det lange løp.
- En forståelse av kunders behov og OKV (Opplevd Kunde Verdi)⁵
- Migreringsprosessen skal påvirke den daglige driften minst mulig.

Derfor ble følgende egenskaper ved oppgaven konkretisert:

- Mest mulig dyp integrasjon
- Mest mulig grad av automasjon og infrastruktur som kode innenfor prosjektets rammer
- Sluttkundens behov og ønsker i fokus.

I tillegg ble det definert i designrapporten at det endelige systemet skulle inneholde hovedsakelig to komponenter:

- Et fungerende test-system
- En effektivisert fremgangsmåte ved hjelp av automatisering og IaC.

³ (Brasø & Aalrust, Forstudierapport, 2020)

⁴ (Brasø & Aalrust, Designrapport, 2020)

⁵ (Kotler & Keller, 2016)

2 Hvordan oppgaven ble løst

Fordi prosjektet baserte seg stort på Microsoft sin teknologi, både i form av miljøet on-premises (Windows server og Hyper-V) og når det gjaldt det skybaserte miljøet (Azure), benyttet prosjektgruppen seg av Microsoft sin egen dokumentasjon i de aller fleste tilfeller. Noen få deler av prosjektet omhandlet teknologi som ikke er utviklet av Microsoft, og dokumentasjon for dette er referert til, eller nevnt der det er relevant. Fordi dokumentasjonen er hentet direkte fra Microsoft kan prosjektgruppen med stor grad av sikkerhet konkludere med at informasjonen som ble brukt er pålitelig.

Overordnet fulgte prosjektet en tredelt struktur, i dette tilfellet definert slik:

Fase 1 – Oppstart

Fase 2 – Utførelse

Fase 3 – Ferdigstilling

Dette er i tråd med hvordan prosjektgruppen faktisk gjennomgikk prosjektet. Denne strukturen ble i seg selv etablert i prosjektplanen utarbeidet under oppstarten. Prosjektplanen, og alle revisjoner av den, er vedlagt. Denne inndelingen blir forklart nærmere i Driftsrapporten, kapittel 2.1. Ved at prosjektet hadde en klar struktur, og rom for testing i forhold til det tekniske, fikk prosjektgruppen mulighet til å opprettholde de fleste mål som ble satt.

2.1 Maskinvare

Maskinvaren som ble brukt i sammenheng med prosjektet kan sies å ha vært todelt, den fysiske infrastrukturen on-premises kontra infrastrukturen som ligger bak skytjenesten Azure. For den fysiske delen hadde prosjektgruppen fullstendig kontroll og oversikt, men for Azure så er/var den fysiske infrastrukturen kontrollert av Microsoft. Dette er en av hovedfordelene med Azure, men i praksis hadde prosjektgruppen ingen kontroll over denne maskinvaren. Fordelen kommer av at sluttbruker selv ikke trenger å investere i maskinvare eller drift av denne. Prosjektgruppen hadde derimot kontroll over størrelsen på tjenestene som ble tatt i bruk gjennom Azure, eksempelvis når det kom til størrelsen til en VM (Virtuell Maskin).

Under vises en enkel oversikt over maskinvaren som ble brukt som on-premises testmiljø, en mer detaljert oversikt for den fysiske serveren finnes i dokumentet *Win16serv.pdf*. Figur 1 viser et utklipp fra dette dokumentet.

Speccy report for WIN-PR3H1PGK4A8, [18.02.2020 08.22.28]
Summary
Operating System Windows Server 2016 Datacenter 64-bit
CPU Intel Core i7 4790 @ 3.60GHz 28 °C Haswell 22nm Technology
RAM 32,0GB Dual-Channel DDR3 @ 794MHz (10-10-10-30)
Motherboard ASUSTeK COMPUTER INC. B85M-G (SOCKET 1150) 28 °C

Figur 1 - Speccy report

2.2 Verktøy og programvare

I prosjektet ble hovedsakelig programvare utviklet av Microsoft benyttet. Fordi løsningen i stor grad benytter skript til å utføre oppgaver har det også blitt brukt programvare relatert til utviklingen av disse.

Tabellen under viser en oversikt over programvaren som er benyttet i prosjektet.

Det er viktig å påpeke at tabellene tar utgangspunkt i programvare som var nødvendig for selve gjennomføringen av prosjektet, altså verktøy og programmer som ble brukt for å oppnå målene i oppgaven. Det som senere blir henvist til som tjenester i det ferdige miljøet blir derfor ikke nevnt her, slik som Windows Virtual Desktop.

Microsoft	Brukt til
Server Manager	Administrere servere, i dette tilfellet Windows Server 2016
Hyper-V	Tjeneste for å opprette og administrere virtuelle maskiner på en Windows Server.
Windows Assessment and Deployment Kit	En samling verktøy for å distribuere egne Windows-OS til datamaskiner eller virtuelle harddisker. Ble brukt i Hyper-V miljøet.
SQL	Structured Query Language. Brukes til å kjøre operasjoner mot relasjonsdatabaser.

Verktøy for Programmering og skript	Brukt til
PowerShell	Rammeverk for å automatisere oppgaver for Microsoft Windows. For eksempel brukt til å lage brukere mot Azure-AD eller rulle ut ARM (Azure Resource Manager) templates.
CMD	Kommandolinjeverktøy i Windows. Brukt til blant annet å kjøre enkle .exe filer, eller andre verktøy slik som azcopy.
Bash	Unix-skall og kommandospråk. Ble brukt til å skripe oppgaver mot Linux-maskiner.
Visual Studio Code	Microsoft sitt integrerte utviklingsmiljø. Prosjektgruppens valg av kodeverktøy.
Chocolatey	«Pakkeverktøy» som tilbyr automatisk installasjon av programvare for bruk mot Windows.

Planlegging/Samhandlingsverktøy	Brukt til
GitHub	Plattform for programvare-utvikling. Tilbyr svært god kildekontroll og ble brukt til å samordne filer. Ble i tillegg brukt til å gi lesere tilgang til koden som ble utviklet i prosjektet.
Microsoft Teams	Samhandlingsverktøy levert av Microsoft. Ble brukt til å samordne dokumentskriving, avholde møter og planlegging.
LucidChart	Flowchart verktøy som ble mye brukt til å lage diagrammer og figurer for prosjektet.
MS Project	Brukt til å utforme prosjektplanen som ble brukt i prosjektet.

Verktøy i Azure	Brukt til
Azure Migrate	Sentralisert «Hub» for å vurdere og migrere on-premises til Azure.
Azure Migrate Appliance	En applikasjon for å oppdage virtuelle maskiner i miljøet det skal migreres fra. Ble brukt under migreringen.
ARM	Azure Resource Manager er tjenesten som håndterer administrasjon og utrulling av ressurser i Azure.
ARM templates	.JSON filer som definerer infrastrukturen og konfigurasjonen for ressurser i Azure. En «mal» som kan rulle ut forespørsler opp mot ARM. Representerer IaC i dette prosjektet.
Azure Active Directory	Skybasert tjeneste for å administrere identitet og aksess.
Azure Automation	Tjeneste for skybasert automasjon og konfigurasjon.
Azure Update Management	Brukes for å administrere operativsystem og oppdateringer på Windows og Linux.
Azure Storage Account	Lagrer data og gir tilgang gjennom Azure AD.
Azure Active Directory Domain Services	Tilbyr domenetjenester slik som LDAP, Access Control og Group Policy Objects. Fullstendig kompatibelt med Windows Server Active Directory.
Azure SQL Databases	PaaS løsning for relasjonelle databaser. Avtale-definert oppetid på 99.99%. Destinasjon for migrering når det kommer til SQL i det fysiske miljøet.

2.3 Arbeidsfordeling under prosjektet

Mange av oppgavene var felles, slik som skrivingen av diverse dokument og planleggingen innledningsvis i prosjektet. Tabellen under beskriver diverse arbeidsoppgaver som hver enkeltperson har hatt ansvaret for, men det er verdt å nevne at det også har vært mye samarbeid på tvers av disse.

Emil Antoni Brasø	Endre Aalrust
ARM-templates	Azure Automation
Azure Migrate	Azure Update Managemnt
Azure Backup	Azure Log Analytics
Windows Virtual Desktop	Azure Active Directory Access Panel
Oppsett Hyper-V Host	Skripting i PowerShell
Webtjener i Azure og on-premises	Oppsett av filtjener on-premises
Azure AD og ADDS	Oppsett av ADDS on-premises
Azure Storage Accounts	Automatisering av Windows 10 installasjon
OpenVPN server on-premises	Oppsett Hyper-V Host
Bash-scripting	
Oppsett av SQL on-premises og Azure	
Azure Key Vaults	
Utforming av figurer	

3 Gjennomføring av prosjektet

Starten av prosjektet gikk for det meste ut på å definere oppgaven og samle dokumentasjon for hvordan løsningen kunne utføres. Et problem som oppsto tidlig i prosjektet var spørsmål om hvorvidt det var mulig å få gjennomført testmigreringene, med utgangspunkt i et fysisk miljø. Selv om det var teoretisk mulig å ha simulert hele prosessen i et virtuelt miljø, ville dette vært vanskelig og muligens utenfor prosjektets rammer.

Dette ble heldigvis løst ved at Jostein Lund ved NTNU lot prosjektgruppen ta i bruk en fysisk server som han disponerer, som nevnt i kapittel 4.1⁶.

Et annet problem kom i form av at prosjektgruppen ikke hadde et passende Azure-miljø fra starten av prosjektet. Prosjektgruppen hadde fått tilgang til et miljø gjennom NTNU og Azure for students⁷, men begge alternativene hadde visse restriksjoner som ikke tillot ressurser som prosjektet trengte. Dette problemet ble derimot løst kort tid etter at prosjektgruppen oppdaget det, gjennom rask og effektiv dialog med Sopra Steria. Sopra Steria endte opp med å disponere et Azure-miljø med kostnadstak på 4000 NOK i måneden for dette prosjektet.

Når alt av nødvendige verktøy, miljø, maskiner og rettigheter var på plass gikk prosjektgruppen i gang med å utvikle fremgangsmåtene og skriptene som var målet med prosjektet. Dette gikk for det meste uten store problemer, men det ble en del testing og feiling, noe som anses naturlig i et slikt prosjekt. Microsoft Docs ble flittig brukt til å implementere løsningene, og mye tid ble brukt til å dokumentere og avgjøre en god fremgangsmåte. Det ble dokumentert en del underveis, men mye av dokumentasjonen ble gjennomført først etter at alle systemer var testet og implementert som en del av løsningen.

Til slutt kom finpussen, hvor en del tid ble dedikert til å korrekturlese dokumentene og sette dette sammen til ett helhetlig sluttprodukt. Dette gjelder også utarbeidelsen av dette dokumentet.

I de videre kapitlene drøftes måloppnåelse i forhold til målene som ble definert i forstudiet.

⁶ Maskinvare

⁷ <https://azure.microsoft.com/nb-no/free/students/>

3.1 Effektmål

Effektmålene skal vise til hvilke effekter som skal komme ut av prosjektet, i form av konkrete resultater. Alle effektmålene er grundigere definert i kapittel 4.1 av forstudiet, men denne tabellen gir en oversikt over disse, samt graden av måloppnåelse ved prosjektets slutt.

Mål	Oppnådd?	Hvordan
1. Effektivisert framgangsmåte i forhold til denne typen migrasjon	Ja	Ved å ta i bruk automasjon, IaC og ARM-templates ble det utviklet effektive metoder for å migrere systemene. Ved bruk av disse metodene vil tiden som brukes på migrering avhenge av systemets størrelse og tilkobling, og ikke tiden brukt på å iverksette prosessen.
2. Minimere kostnader knyttet til on-premises	Delvis	Det blir anbefalt og gjennomført en del tjenester som er kostnadsbesparende i forhold til en alternativ løsning. Kostnadsbildet ble i stor grad sett vekk fra senere i prosjektet.
3. Flytte bedriftens lagring og prosessering over til skytjenester	Ja	Microsoft sin programvare slik som Azure Migrate og azcopy ble brukt til å gjennomføre dette.
4. Ta i bruk muligheter tilbudt av skytjenester, med følgende delmål: <ul style="list-style-type: none"> a. Økt SLA b. Skalerbarheten som Azure tilbyr c. Sikkerhetskopiering 	Ja til samtlige	Spesifiserte VM-er og SQL-servere har SLA (Service Level Agreement) - definert opptid opptil 99%. Skalerbarhet er en innebygd fordel med Azure. Et eksempel i driftsrapporten er hvordan WVD kan utvides enkelt. Azure Backup ble tatt i bruk og forklart.
5. Skreddersy og finne fram til hvilken komposisjon av skytjenester som passer for kunden.	Ja	Public Cloud ble valgt, og ble begrunnet der det var relevant.
6. Gjøre arbeidsdagen lettere for de ansatte/brukerne	Ja	Brakerundersøkelse ble gjennomført og tjenestene ble implementert for å imøtekomme sluttbruker.

I forhold til prosjektet er alle effektmålene oppnådd, med unntak av kostnadsbildet. Dette var vanskelig å ta et standpunkt til, fordi et kostnadsbilde i veldig stor grad avhenger av hver enkelt bedrift. Uten tilgang til reell data ville det vært svært komplisert, og sannsynligvis utenfor prosjektets rammer å ta stilling til dette. I dialog med veiledere i Sopra Steria ble det fattet at prosjektgruppen skulle se bort fra kostnadsbildet for denne prosjektoppgaven.

3.2 Resultatmål

Resultatmålene skulle konkret beskrive hva som skulle foreligge som et resultat ved prosjektets slutt.

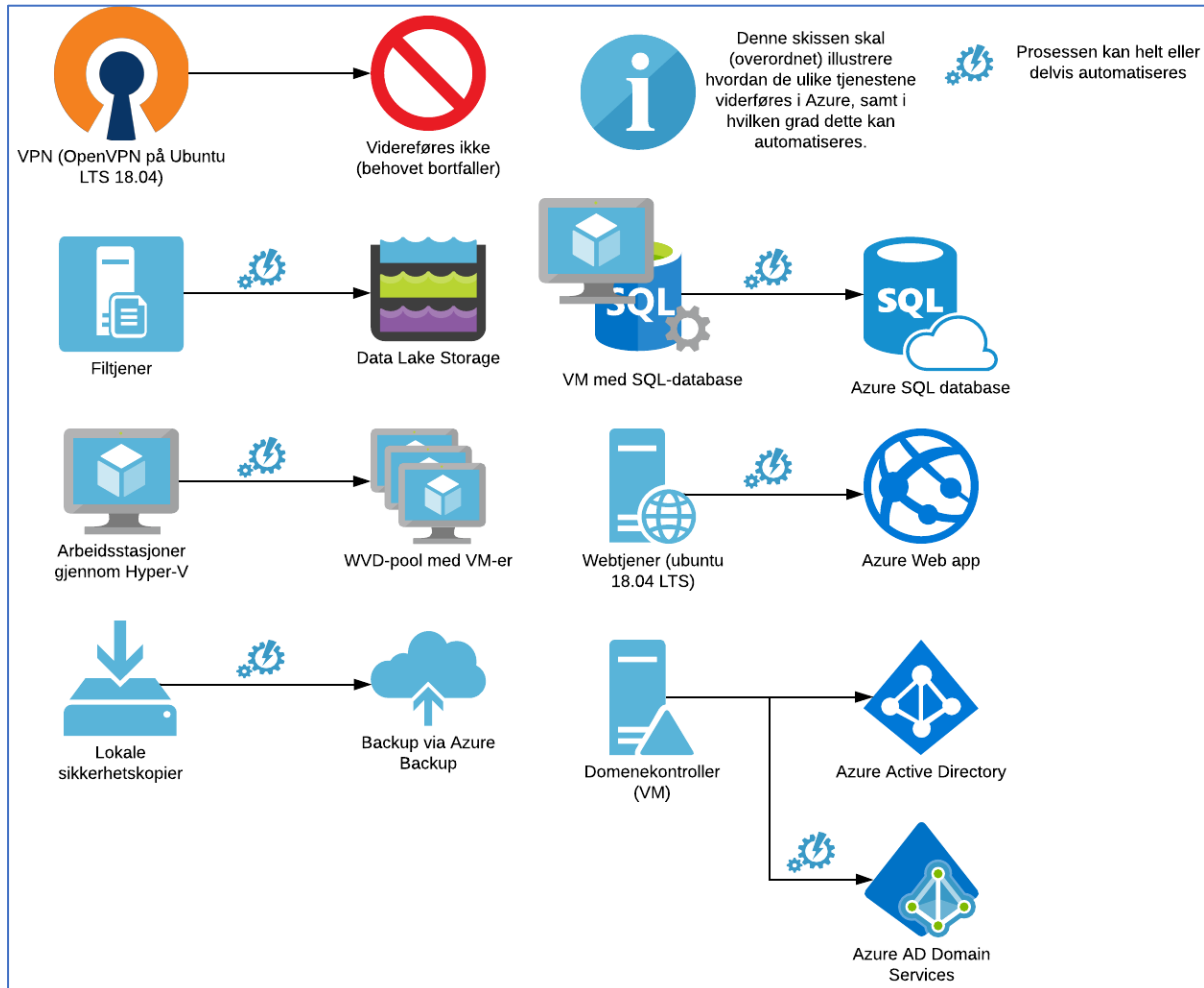
Mål	Oppnådd?	Hvordan
1. En fremgangsmåte som beskriver selve prosessen med å migrere dagens løsning fra on-premises til en hel eller delvis sky-basert løsning.	Ja	Dette var kjernen for oppgaven. Den praktiske løsningen som ble utarbeidet beskriver hele prosessen for migrering, og beskriver valg og eventuelle problemer underveis.
2. En overordnet analyse av det økonomiske aspektet ved å flytte bedriftens IT-systemer til en sky-basert løsning.	Nei	Av samme grunn som i kapittel 5.1 ⁸ , ble kostnadsbildet i stor grad sett vekk fra i prosjektet.
3. Ett system som utnytter skybasert teknologi.	Ja	Ettersom oppgaven tar for seg skybasert teknologi anses dette som oppnådd.
4. En analyse av typen løsning; hel eller delvis og hva som vil gagne bedriften i det lange løp.	Ja	Valget av typen løsning er begrunnet, og andre løsninger drøftes også.
5. En forståelse av kunders behov og OKV (Opplevd kunde Verdi)	Ja	En brukerundersøkelse ble gjennomført og prosjektgruppen har imøtekommer sluttbruker og deres ønsker.
6. Migreringsprosessen skal påvirke den daglige driften minst mulig.	Ja, delvis	Ved at migreringen ikke krever at on-premises systemet stanser (for eksempel så trengs ikke noen harddisker å flyttes), vil migreringsprosessen ikke påvirke den daglige driften, men eventuell opplæring før fullstendig migrering kan bli nødvendig.

Det første resultatmålet definert i forstudierapporten er kjernen i oppgaven. Den praktiske løsningen som ble utarbeidet i driftsrapporten beskriver hele migreringsprosessen. Denne løsningen tar for seg hvordan prosjektgruppen har gjennomført migreringen, i tillegg til en beskrivelse av valg som ble tatt underveis. Det er prosjektgruppens formening at dette sto til forventningene som beskrevet i starten av prosjektet.

Som nevnt i forrige kapittel har mye av det økonomiske aspektet ved oppgaven bortfalt, det ble bestemt i samråd med veiledere i Sopra Steria og NTNU. Det ble derfor ikke utført en analyse av kostnadsbildet. Kost og nytte ble beskrevet i kapittel 8 av forstudiet, og er eneste unntak for dette.

⁸ Effektmål

Det ble et omfattende arbeid å bestemme hvilke tjenester i Azure som skulle ta over for tjenestene on-premises. I stor grad er dette på grunn av at Azure tilbyr svært mange tjenester. Løsningen som ble fattet til slutt var et produkt av dette arbeidet, men også en vurdering av hvor mye tid prosjektgruppen hadde til rådighet når det kom til selve utførelsesfasen. Følgende figur er hentet fra kapittel 2.2.1 av driftsrapporten, og viser hvordan tjenestene ble videreført i den endelige løsningen.



Figur 2 - Hvordan tjenester videreføres

Det er verdt å nevne at løsningen som til slutt ble levert er basert fullstendig i Azure. En tjeneste ble ikke videreført i det hele tatt, dette var VPN (Virtual Private Network)-tjenesten fra on-premises. VPN ville ha blitt brukt til å gi tilgang til on-premises nettverk fra internett. Azure AD gjorde derimot at VPN ble sett på som overflødig for dette prosjektet, ettersom AD brukes til å autentisere tilgangen til tjenestene i skyen.

3.3 Prosessmål

Prosessmål er ikke reelle for prosjektets resultat, men sier noe om hva slags effekt prosjektet skulle ha på prosjektdeltakerne.

Hva	Oppnådd?	Hvordan
1. Øke kompetansen innenfor de fagområdene som prosjektet utforsker.	Ja	Gjennom prosjektets gang var prosjektgruppen nødt til å lære mye forskjellig innenfor de berørte fagområdene, for å levere et godt resultat.
2. Øke kompetansen når det gjelder samarbeid og kommunikasjon mellom prosjektgruppen og bedriften.	Ja	Det var kontinuerlig god dialog mellom prosjektgruppen og veilederne. Prosjektgruppen er fornøyd med hvordan samarbeidet har vært. I tillegg har prosjektgruppen gjennomført diverse organisatoriske oppgaver.

Som studenter har prosjektgruppen lært utrolig mye i løpet av dette prosjektet. Ved starten av prosjektet forelå kun grunnleggende kunnskaper om Azure, men en del kunnskap slik som virtualisering og Windows Server var anvendelige i dette prosjektet.

Migreringsprosessen var helt ukjent for prosjektgruppen, svært mye ble lært av å prøve og feile. Ettersom IaC og automasjon var sentralt for prosjektet så ble det lært mye om dette. Totalt sett har prosjektgruppen, subjektivt sett, nettopp gått gjennom en ekstremt lærerik periode. En sentral gevinst, er at prosjektgruppen per i dag er mye mer klar over hvordan IaC kan brukes sammen med Azure for å gjøre oppgaver mer effektivt, slik som ved automasjon.

Gjennom studiet har prosjektgruppen erfart at kommunikasjon er viktig for praktisk talt alle prosjekt. Derfor ble det tidlig bestemt at det skulle benyttes verktøy for samhandling og versjonskontroll slik at alle involverte i prosjektet hadde tilgang til alt av dokumenter til enhver tid. Tankegangen var at veiledere kunne gå igjennom hva prosjektgruppen jobbet med, på eget initiativ, uten å ha fått kopier tilsendt av prosjektgruppen i hvert tilfelle.

Her ble Microsoft Teams brukt for kommunikasjon og dokumentdeling, og GitHub til versjonskontroll og deling av skript. Gjennom disse tjenestene ble kommunikasjon og samordning gjort betraktelig lettere. En annen fordel er at det er mye lettere å holde orden dersom mest mulig er tilgjengelig gjennom samme sted.

4 Videre arbeid

Systemet som er blitt satt opp for bedriften i Azure er fungerende, men kan bli utbedret. Fordi oppgaven i all hovedsak omhandlet migreringsprosessen, ble det valgt å legge mest tid inn i akkurat dette.

Azure Runbooks

Et eksempel på en tjeneste som ble implementert, men i en veldig grunnleggende form, er Azure Runbooks. Denne tjenesten lar brukeren legge inn skript som blir kjørt etter brukerens ønske. Her er det mulighet for å utvikle flere skript som gjør arbeidet med å drifte systemet lettere og mer effektivt.

Azure update Manager

Det som ble nevnt om Azure Runbook, stemmer også for Azure Update Manager. Denne tjenesten styrer systemoppdateringer for alle maskiner i domenet. I løsningen er det gitt et eksempel på dette, men det kan konfigureres bedre med tanke på ulike behov og sikkerhet.

Azure Blueprints

Azure Blueprints er et overordnet dokument som setter opp et helt miljø med en enkelt utrulling. Dette innebærer utrulling av templates og ressursgrupper, men også tildeling av policyer og roller i miljøet. Ved videre utvikling av systemet er dette en veldig attraktiv tjeneste å ta i bruk ettersom den gjør utrulling av miljøer, og eventuelle endringer av eksisterende miljøer enklere.

Azure Load Balancer

Azure Load Balancer er et verktøy for å automatisk balansere last mellom to eller flere servere, et eksempel er i forhold til web-servere. Dersom lasten på et gitt tidspunkt på året er langt høyere enn normalt, vil tjenesten automatisk fordele lasten mellom tilgjengelige servere. Dette forhindrer at sluttbruker opplever forsinkelser eller nedetid.

5 Endringer i løpet av prosjektet

I løpet av hele prosjektet ble en del endringer foretatt. Ingen av disse var radikale i den grad at de krevde en ny definering av hvordan prosjektet skulle løses. Unntaket er kanskje Azure Blueprints. Blueprints var planlagt brukt i løpet av designrapporten, men ble i starten av utførelsesfasen og utarbeidelsen av driftsrapporten vurdert som unødvendig for dette prosjektet. Dette skyldtes at Blueprints var mer passende for å sette opp fullstendige miljøer med fokus på governance, kontroll og policy. På grunn av prosjektets tidsbegrensning ble dette rett og slett vedtatt å ikke tas i bruk, men ville vært svært interessant å ha implementert som en del av videre drift.

En annen endring som er verdt å nevne er hvordan kostnadsbildet ble sett vekk fra i mesteparten av prosjektet. Dette ble nevnt i forhold til målene i kapittel 5.1⁹ og 5.2¹⁰, men for å oppsummere var dette på grunn av prosjektets natur, og var noe som prosjektgruppen kom til enighet med veilederne om.

⁹ Effektmål

¹⁰ Resultatmål

FORSTUDIERAPPORT

En automatisert reise fra datarom til sky

Emil Antoni Brasø, Endre Aalrust

DATO	VERSJON	BESKRIVELSE	FORFATTER
15.01.20	1.0	Opprettelse av dokument	Endre, Emil
28.01.20	1.1	Revisjon av dokument	Endre, Emil
29.01.20	1.2	Ferdigstilling av dokumentets innhold	Endre, Emil
12.02.20	1.3	Generell rettskriving, forbedret formulering	Emil
18.02.20	1.4	Stiliserte, inkluderte tittel på oppgaven	Emil
15.05.20	1.5	Stilsetting, en referanse	Emil, Endre
18.05.20	1.6	Formuleringer, språk	Emil

1 Introduksjon til forstudierapporten

Denne forstudierapporten er skrevet i sammenheng med en bacheloroppgave som utført i samarbeid med IT-selskapet Sopra Steria.

1.1 Hensikten med forstudierapporten

Hensikten med denne rapporten er å få en oversikt over alle faktorer og detaljer som er avgjørende for hvorvidt prosjektet lykkes. Rapporten skal gi en grundig oversikt over hva prosjektet skal omhandle, og fungerer derfor også som en planleggingsrapport med alle detaljer som er nødvendige både for oppdragsgiveren Sopra Steria og prosjektgruppen, Emil Antoni Brasø og Endre Aalrust.

Utarbeidelsen av ett forstudie før prosjektets start bidrar til at alle parter har oversikt over hva som skal skje i tiden fremover og hvordan arbeidsoppgaver skal løses. Blant annet vil for eksempel kostnadsanalysen være svært interessant for kunden slik at de kan vurdere om dette er en investering som lønner seg. For prosjektgruppen vil punkter som risikoanalyse og mål være interessante gjennom hele prosjektet, i tillegg til at prosjektgruppen utarbeider nye versjoner av disse etter hvert som prosjektet går fremover.

1.2 Innholdet i forstudierapporten

I dette dokumentet beskrives en rekke punkter som er relevant for en grundig planlegging av prosjektet og for at prosjektet skal lykkes. Først blir bakgrunnen for hvorfor prosjektet oppsto forklart. For dette prosjektet foreligger det også en kort beskrivelse av dagens system og hvilke problemer og mangler det har, for den tenkte bedriften Salmo AS. Siden dette er en tenkt bedrift vil ikke mye vekt ilegges her, men generelle antagelser som prosjektgruppen anser som rimelige for enhver bedrift blir nevnt.

Videre kommer en beskrivelse av alle målene, både hovedmål og delmål, som skal nås i løpet av prosjektet. Noen av disse kan bli endret i løpet av de forskjellige delene av prosjektet, men de som blir satt opp i denne rapporten er en tydelig pekepinn på hva prosjektets retning er, og hva som skal være oppnådd ved slutten av prosjektet.

Ved at retningen til prosjektet blir definert, fungerer dette dokumentet som et hjelpemiddel for at agent og prinsipal skal komme til enighet om hvor prosjektet er, i tillegg til hvordan prosjektet skal gå videre, ved det tidspunktet forstudiet ble utarbeidet.

Under kapittel 5¹ er det en beskrivelse av hvilke interesser de forskjellige gruppene som er knyttet til prosjektet har. Her gis en oversikt over:

- Eksterne interessenter, eksempelvis ledelsen i Sopra Steria og sluttbrukerne av systemet.
- Interne interessenter, i all hovedsak prosjektgruppen, men gjelder også for veilederne involvert i prosjektet.

En viktig del av forstudierapporten var å fastslå kritiske suksessfaktorer, i tillegg til å utføre en risikoanalyse. I disse delene av forstudiet, henholdsvis kapittel 6² og 7³, kommer det frem hva som skal til for å lykkes med dette prosjektet og hvilke risikofaktorer man må ta hensyn til underveis.

¹ Kritiske suksessfaktorer

² Risikoanalyse

³ Analyse av kost og nytte

En slik risikoanalyse er viktig for å kartlegge risikofaktorer på forhånd slik at man kan vurdere hvilke risikoer som er verdt å ta og hva som blir konsekvensen hvis man skulle støte på problemer. Dette hjelper også til å planlegge hva som skal gjøres hvis noe først skulle gå galt.

Analysen av Kost og nytte definerer et estimat av utgiften dette prosjektet vil medføre for bedriften, stilt opp mot nytten dette prosjektet vil gi. Siden det er snakk om en tenkt bedrift, vil det være ett særskilt søkelys på kostnader knyttet til selve implementasjonen av systemet. Forstudierapporten vil ikke ha et null-alternativ å sammenligne med.

Til slutt i dette dokumentet blir de forskjellige retningslinjene for prosjektet beskrevet. Her defineres ulike krav til dokumentasjon, standarder og kvalitetstester, samt endringshåndtering og prosjektorganisering.

Innholdsfortegnelse - Forstudierapport

1	Introduksjon til forstudierapporten	2
1.1	Hensikten med forstudierapporten	2
1.2	Innholdet i forstudierapporten	2
	Innholdsfortegnelse - Forstudierapport	4
2	Bakgrunn for prosjektet	6
2.1	Beskrivelse av kundens problemer og behov	6
2.2	Kort om dagens systemer og rutiner	7
3	Prosjekt mål	8
3.1	Effekt mål – Hva oppnås med prosjektet?	8
3.2	Resultat mål	8
3.3	Prosess mål	9
3.4	Prosjektets omfang	9
3.5	Prosjektets milepæler og hovedaktiviteter	9
4	Interessenter og rammebetingelser	10
4.1	Interessentanalyse	10
4.1.1	Eksterne interessenter:	10
4.1.2	Interne interessenter:	10
4.1.3	Rammebetingelser	11
5	Kritiske suksessfaktorer	12
5.1	Suksessfaktorer	12
5.2	Informasjonsbehov	12
5.3	Prinsipal-agent-teori	12
6	Risikoanalyse	13
6.1	Drøfting av risikofaktorer og konsekvensanalyse	14
6.1.1	Diagram for konsekvens/sannsynlighet	16
6.2	Tiltak for å kontrollere risiko	17
7	Analyse av kost og nytte	19
7.1	Ikke-kvantifiserbar nytte	19
7.2	Estimerte kostnader ved innføring av systemet	20
8	Retningslinjer og standarder	21
8.1	Krav til dokumentasjon	21
8.1.1	Forstudierapport	21
8.1.2	Designrapport	21
8.1.3	Driftsrapport	21
8.1.4	Sluttrapport	21

8.2	Krav til kvalitetsgjennomganger	22
8.2.1	Dokumenter og rapporter.....	22
8.2.2	Microsoft Teams	22
8.2.3	Møter	22
8.2.4	Revisjon av dokumenter	22
8.3	Krav til standarder og metoder.....	22
8.3.1	Dokumentmaler	22
8.3.2	Verktøy for implementasjon	22
8.3.3	Microsoft.....	22
8.4	Endringshåndtering.....	23
8.4.1	Dokumentasjon av endringens innhold	23
8.4.2	Konsekvensanalyse for prosjektet	23
8.4.3	Beregne eventuell kost/nytte etter endring.....	23
8.4.4	Godkjenning og aksept.....	23
8.4.5	Loggføring av endring	23
8.4.6	Planjustering	23
8.4.7	Informering av interessenter	23
8.4.8	Gjennomføring av endring	23
9	Prosjektorganisering	24
10	Referanser	24

2 Bakgrunn for prosjektet

I samarbeid med Sopra Steria ble det utarbeidet en caseoppgave hvor prosjektgruppen på vegne av en tenkt bedrift skulle undersøke om det var hensiktsmessig å overføre driften av deres it-systemer til hel eller delvis sky. Denne casen er vedlagt som dokumentet *case.pdf*.

Prosjektet tar utgangspunkt i å utarbeide en optimalisert og effektiv fremgangsmåte for hele migreringsprosessen. Det ble sett på gevinsten, eller ROI (Return Of Investment), forbundet med prosessen. For at en skal kunne ta informerte beslutninger på vegne av en kunde i et marked som går over til sky i svært høyt tempo, er det nødvendig å undersøke disse problemstillingene. Det er også viktig å ikke kun se på it-tekniske gevinster, det må rettes fokus mot sluttbruker også.

2.1 Beskrivelse av kundens problemer og behov

Den tenkte bedriften Salmo har per dags dato tilnærmet alle sine IT-systemer on-premises. Bedriften har såpass store mengder data at de ønsker å undersøke mulighetene ved skybasert drift. Bedriften har i tillegg til dette behov for prosessering som øker kostnadene ved en on-premises løsning. Dette prosjektet skal løse problemer knyttet til den dagligdagse driften som følge av behovet knyttet til store mengder data og prosessering. Dette blir løst ved at driften overføres til en skybasert løsning.

Definerte problemstillinger med dagens system:

- Store mengder data som fører med seg store kostnader i form av drift og fysiske komponenter.
- Behov for dataprosessering som krever mange ressurser i form av maskinvare og kostnader knyttet til dette.

Dette er problemstillinger prosjektgruppen gransker når det kommer til valg av en skybasert eller delvis skybasert løsning. Videre ser prosjektgruppen på om dette vil gagne bedriften økonomisk og over lengre tid, blant annet ved at den daglige driften blir effektivisert.

Ved å innføre ett skybasert system vil man få mer driftssikkerhet, samtidig som det vil gjøre den daglige driften enklere. Prosjektgruppen tar samtidig søkelys på selve sluttbrukeren, som vektlegger andre egenskaper og aspekter enn de med IT-teknisk bakgrunn.

Et problem som kunne ha oppstått i en bedrift med on-premises system er at det bare er noen få som vet hvordan systemet fungerer eller har tilgang til en gitt tid.

Hvis feil oppstår når disse personene ikke er til stede vill bedriften nesten definitivt oppleve nedetid som ellers kunne blitt unngått. I tillegg til at infrastrukturen kan endres enklere gjennom et samordnet grensesnitt, kan man også automatisere mange oppgaver som ellers ville blitt gjort manuelt. En annen fordel knyttet til dette er tilgjengelighet, det er lettere å gjøre endringer på et system som er tilgjengelig over internett, enn på et system som er bak lukkede dører on-premises.

2.2 Kort om dagens systemer og rutiner

Bedriften Salmo driver med lakseoppdrett og har ingen skytjenester eller IT-tekniske satsningsområder i dag, utenom lagring og prosesseringsbehovet knyttet til oppdrett. Med dette menes at de ikke har noen skytjenester i form av IaaS (Infrastructure as a Service) eller PaaS (Platform as a Service). Skytjenester som tilbyr programvare slik som Office365 er SaaS (Software as a Service), og blir ikke tatt særlig hensyn til her, selv om man med sikkerhet kan anta at bedriften benytter seg av dette.

De øvrige systemene finner sted i form av overvåkning, samt vanlige bedriftsapplikasjoner slik som Office365 eller lignende. Når det gjelder dataprosesseringen trekkes fram Machine Learning som ett eksempel på prosjekter en bedrift i industrien kan ha gående⁴.

For øyeblikket bruker de on-premises lagring for dataene, men dette begynner å gå tomt og er en av årsakene til at de vurderer å flytte dette over til skytjenester.

De drifter og eier systemene internt.

Sluttbrukeren hos Salmo vil i hovedsak være de ansatte i bedriften, som varierer fra produksjonsarbeider til ansatte i administrasjonen.

⁴ (Sealab, 2020)

3 Prosjektmål

For å effektivt kunne utarbeide ett godt prosjekt var det nødvendig at prosjektgruppen satte seg klare mål. Disse målene ble:

- Brukt for å komme til enighet med oppdragsgiver i forhold til omfang og forventet resultat.
- Brukt til å vurdere resultatet av prosjektet etter fullført prosjekt.
- Brukt underveis til å måle fremgang, men også for å opprettholde en klar visjon for prosjektet.

Følgelig ble det viktig at disse målene er målbare og styrbare, slik at prosjektgruppen kunne rette seg etter disse. Klar formulering var et viktig virkemiddel for å oppnå dette, slik at det ikke er forela noe tvil om hva målene besto av. Alle lesere skal ha samme oppfatning av hva målet innebærer.

Videre ble målene oppdelt etter effektmål, resultatmål og prosessmål.

Effektmålene omhandlet verdiskapning og langsiktig avkastning, resultatmålene tok for seg hva som kom ut av prosjektet, mens prosessmålene omhandlet innvirkningen som prosjektet skulle ha på prosjektdeltakerne.

3.1 Effektmål – Hva oppnås med prosjektet?

Dette tar for seg hvilke effekter som skal komme ut av prosjektet. Effektene skal virke mot konkrete resultater, gitt i delkapittel 3.2.

- Effektivisert framgangsmåte i forhold til denne typen migrasjon.
- Minimere kostnader knyttet til on-premises.
- Flytte bedriftens lagring og prosessering over på skytjenester.
- Ta i bruk muligheter tilbudt av skytjenester, med følgende delmål:
 - o Økt SLA (Service Level Agreement). Azure tilbyr 99.99% SLA i flere tilfeller⁵
 - o Skalerbarhet
 - o Sikkerhetskopiering
- Skreddersy og finne fram til hvilken komposisjon av skytjenester som passer for kunden. Det kan være seg hel, delvis, private eller public sky.
- Gjøre arbeidsdagen lettere for de ansatte/brukerne

3.2 Resultatmål

Resultatmålene beskriver hva som konkret skal foreligge som resultat når prosjektet er ferdig.

Følgende resultat skal foreligge:

- En framgangsmåte som beskriver selve prosessen med å migrere dagens løsning fra on-premises til en hel eller delvis skybasert løsning.
- En overordnet analyse av det økonomiske aspektet ved å flytte bedriftens IT-systemer til en skybasert løsning.
- Ett system som utnytter skybasert teknologi.
- En analyse av typen løsning; hel eller delvis og hva som vil gagne bedriften i det lange løp.
- En forståelse av kunders behov og OKV (Opplevd Kunde Verdi)⁶
- Migreringsprosessen skal påvirke den daglige driften minst mulig.

Alle disse punktene skal være gjennomført innen prosjektets sluttdato den 20.05.2020.

⁵ (Microsoft, Sammendrag av SLA for Azure-tjenestene, 2020)

⁶ (Kotler & Keller, 2016)

3.3 Prosessmål

Prosessmålene forteller hva slags effekt prosjektet vil ha på prosjektdeltakerne. Disse er ikke direkte forbundet med det som skal levers, men heller hva prosjektgruppen skal gjennom i løpet av prosjektets gang.

- Øke kompetansen innenfor de fagområdene prosjektet utforsker.
- Øke kompetansen når det gjelder samarbeid og kommunikasjon mellom prosjektgruppen og bedriften.

3.4 Prosjektets omfang

Prosjektets omfang fungerer som et supplement til prosjektmålene. Her beskrives ytterlige grenser for hvor langt prosjektet skal gå.

- Prosjektet skal utarbeide en løsning som fører til økt verdi for bedriften, og erstatter de gamle systemene i bruk.
- Prosjektet skal konkludere med en løsning og ikke flere løsninger. Alternative løsninger skal likevel dokumenteres.
- Prosjektgruppen skal ikke drive brukeropplæring, men vil ta for seg kostnader knyttet til dette.

3.5 Prosjektets milepæler og hovedaktiviteter

Ved hjelp av MS Project har prosjektgruppen satt opp en oversikt over hvordan gruppen ønsker å fordele tiden som er tilgjengelig i løpet av prosjektet. Følgende frister fungerer som milepæler:

- Avtale (møte) 24.01.2020
- Levering 20.05.2020
- Presentasjon 26.05.2020

Prosjektplanen og alle revisjoner er vedlagt.

4 Interessenter og rammebetingelser

4.1 Interessentanalyse

Interessenter deles inn i eksterne og interne interessenter. Interne omhandler alle interne ressurser og vil i denne sammenhengen være prosjektgruppen, veileder fra IDI AIT og veileder fra Sopra Steria. De eksterne interessentene vil være den tenkte bedriften Salmo, men også Sopra Steria som en organisasjon. Prosjektgruppens blikk på denne relasjonen er at det utvikles ett prosjekt på vegne av Sopra Steria, og verdien skapt skal komme til nytte for dem og deres kunde, Salmo AS.

4.1.1 Eksterne interessenter:

- Sopra Steria
 - o Sopra Steria fungerer som oppgavestiller, og vil stille med en ekstern sensor. Det er viktig å opprettholde god og jevnlig kommunikasjon mellom prosjektgruppen og Sopra Steria.
- Ledelsen i Salmo AS
 - o Ledelsen i Salmo vil være opptatt av å forhindre de langsiktige problemene med datalagring og dataprosessering de blir utsatt for. Derfor ønsker de at prosjektgruppen skal vurdere hvorvidt hel eller delvis skyløsning lønner seg for dem.
 - o Denne interessenten fungerer som oppdragsgiver og er med på å godkjenne resultatet. Ledelsen vil være involvert i alle aspekter ved prosjektet og være med på store avgjørelser. Videre vil de bidra med innsyn i bedriftens behov og ønske slik at løsningen blir skreddersydd til bedriften på best mulig måte.
 - o Suksesskriterier til ledelsen er som følger: Utarbeid en løsning for migrering fra on-premises til sky og dokumenter denne prosessen. Et overordnet mål er at denne løsningen blir lønnsom.
- Sluttbruker
 - o Sluttbrukerne vil være med på å gi informasjon om hvordan systemet fungerer i bedriften når implementasjonen trer i kraft.
 - o Det er ønskelig å opprettholde effektiviteten i bedriften så mye som mulig under selve migreringsprosessen slik at bedriften ikke lider tap.
 - o Sluttbrukerne skal oppleve økt effektivitet og brukervennlighet når det prosjektet er gjennomført.

4.1.2 Interne interessenter:

- Prosjektgruppen
 - o Består av to medlemmer, Emil Antoni Brasø og Endre Aalrust. Prosjektgruppen skal planlegge, utvikle og levere oppgaven. For å oppnå dette er det viktig at prosjektgruppen sitter med så mye relevant informasjon som mulig.
 - o Suksesskriteriene vil gå ut på få til en vellykket og god prosess, men også ett godt resultat for prosjektet. Dette med den hensikt å opparbeide verdifull erfaring og kompetanse. Prosjektgruppen ønsker å gjøre sitt beste og vise hva de er gode for, noe som fungerer som motivasjon for å få levert et godt resultat og ett godt prosjekt.

Interessent	Kriterier for suksess	Bidrag i forhold til prosjekt
Eksterne Interessenter:		
Sopra Steria	At oppgaven gir verdifull informasjon om prosessen med å migrere et IT-system fra on-premises til sky.	Arbeidslokaler, teknisk hjelp, veiledning.
Ledelsen i Salmo	Effektiv og lønnsom overgang fra gammelt til nytt system. Gjøre det nye systemet skreddersydd til bedriftens behov og ønsker.	Bidra med innsikt og informasjon til prosjektgruppen som gjør det lettere å utarbeide en løsning som passer bedriften best mulig.
Sluttbruker	Løsningen skal gjøre det enklere og mer effektivt for hverdagsbrukeren i bedriften å utføre sine arbeidsoppgaver.	Sluttbruker er med på å gi tilbakemelding før, under og etter prosjektløpet.
Interne Interessenter:		
Prosjektgruppen	Utarbeide en løsning som ganger bedriften innenfor gitt tidsfrist. Holde høy kvalitet over arbeidet som gjøres underveis i prosjektet.	Utviklingen av selve løsningen, god dialog med de eksterne interessentene for å sikre at det ikke blir fravikende mål mellom det eksterne og interne, i henhold til agentteori ⁷ . Ansvarsfull utvikling av løsningen.

4.1.3 Rammebetingelser

Rammebetingelser er styrende og absolutte krav til prosjekt-gjennomføringen og resultatet. Disse betingelsene må oppfylles for at prosjektet skal regnes som en suksess.

For å forstå rapporten må det presiseres at disse i stor grad er konkretiserte krav som avgrenser prosjektet. Rammebetingelsene har stor innvirkning på hensyn prosjektgruppen må ta under både planlegging og utvikling. Det er viktig å være grundig når det gjelder dette, siden prosjektgruppen vil være ute av stand til å endre rammebetingelsen på kortsiktig eller langvarig basis.

1. Ferdigdato

- a. For dette prosjektet er 20. Mai endelig innleveringsdato.

2. Kostnadsrammer

- a. For testmiljøet i Azure er en øvre kostnadsramme på 4000 NOK satt.

3. Drifts- og utviklingsmiljø

- a. Løsningen skal ta i bruk, og være på, Microsoft Azure.

⁷ (Kotler & Keller, 2016)

5 Kritiske suksessfaktorer

5.1 Suksessfaktorer

- Løsningen som presenteres etter prosjektets slutt må gjenspeile behovet presisert av agent og være fleksibelt i forhold til videre utvidelse ved behov.
- Krav spesifisert i prosjektet må være klare og definerte slik at problemer eller misforståelser ikke oppstår.
- Gode metoder for håndtering av problemer som kan oppstå underveis i prosjektet for å sikre at minst mulig tid blir brukt på problemløsning.

5.2 Informasjonsbehov

God informasjonsformidling er avgjørende for at prosjektet skal lykkes. Dette er prosessen med å gjøre resultater og leveranser i et prosjekt tilgjengelig for alle involverte parter i prosjektet. I kapittel 6.3⁸ diskuteres de viktigste elementene i informasjonsformidling og interessentenes behov for informasjon.

5.3 Prinsipal-agent-teori

Prinsipal-agent-teori dreier seg om hvordan man skal sikre samsvar mellom målene til oppdragsgiver (prinsipal) og utfører (agent).

I denne analysen betraktes ledelsen i Salmo AS som prinsipal, og prosjektgruppen som agent. Kommunikasjonen mellom disse partene er åpenbart viktig for at prosjektet skal lykkes og informasjonen som blir delt mellom disse to partene vil ha direkte innvirkning på prosjektets rammer og krav.

Agent behøver informasjon om krav, spesifikasjoner og eventuelle endringer som skal foretas, fra prinsipal, slik at prosjektet kan gjennomføres. På den andre siden krever prinsipal dokumentasjon og statusrapporter på at prosjektet faktisk blir gjennomført i tillegg til eventuelle endringer som blir gjort under arbeidet på prosjektet. Denne informasjonen kan betraktes som administrativ og formålet er å oppdatere begge parter på hva som skjer i prosjektet slik at ingen problemer oppstår.

Oppgavene som blir utført vil ha en tidsramme for når de skal være ferdige, og det er viktig at prinsipal får oppdateringer på hvor prosjektet er i forhold til den endelige tidsfristen. Hvis krav til endringer skulle forekomme fra en av partene er det viktig at dette blir informert om så fort som mulig slik at en avgjørelse kan tas for å minimere eventuelle forsinkelser.

⁸ Prinsipal-agent-teori

6 Risikoanalyse

I Risikoanalysen drøftes ulike faktorer som kan ha innvirkning på prosjektets suksess. Siden oppgaven omhandler skytjenesten Azure, vil de fleste risikoene gå på problemstillinger forbundet med dette. På den annen side så fjerner skytjenester noen av de tradisjonelle risikofaktorene man ville vært nødt til å ta hensyn til ved on-premises, dette er en fordel og et eksempel på risiko som bortfaller er risikoen for maskinvaresvikt. Det totale risikobildet blir likevel ikke særlig mindre, det må fremdeles legges nok vekt på risikoplanlegging i prosjektet.

Risikofaktor 1 – Utilstrekkelig kartlegging eller forståelse av bedriftens behov

Risikofaktor 2 – Teknologien er ikke moden for bruk

Risikofaktor 3 – Teknologien er ikke tilpasset bedriftens særskilte behov

Risikofaktor 4 – Urealistiske tidsplaner eller budsjetter

Risikofaktor 5 – Dårlig utførelse eller implementasjon

Risikofaktor 6 – Dårlig planlegging

Risikofaktor 7 – Kravene holdes ikke konsistente eller endres underveis

Risikofaktor 8 – Løsningens fremtidige brukere eller driftere har ikke nødvendig kompetanse

Risikofaktor 9 – «Gold-plating»

Risikofaktor 10 – Det politiske miljøet innvirker på systemet

Risikofaktor 11 – Feilaktig eller utdatert informasjon brukt som kunnskapsgrunnlag

For å få en forståelse av disse risikoene, må det ses på konsekvensen av risikoen, sannsynligheten for at risikoen inntreffer, og til slutt hvilke tiltak prosjektgruppen velger å ta i bruk mot et utvalg av risikoene. Det er verken nødvendig eller hensiktsmessig å iverksette konkrete tiltak mot alle risikoene, men denne analysen skal begrunne hvilke man må gjøre noe med.

Etter drøfting av de nevnte risikofaktorene så blir sannsynlighet og konsekvens sammenstilt i en tabell, og de rangeres i fem trinn:

Svært Lav (0-2) – Lav (3-4) – Medium (5-6) – Høy (7-8) (alvorlig)– Svært Høy (9-10) (svært alvorlig)

6.1 Drøfting av risikofaktorer og konsekvensanalyse

Risikofaktor 1:

- Dersom forståelsen av bedriftens behov er utilstrekkelig eller ikke kartlagt nok kan man komme til feil konklusjon på forskjellige beslutningspunkt i prosessen. Dette er viktig for å kunne skreddersy en løsning som vil passe til flest mulig behov hos bedriften.
- Konsekvensen dersom dette inntreffer vil variere litt fra beslutning til beslutning, men vil i verste fall kunne føre til at behov hos bedriften ikke blir innfridd, derfor kan dette plasseres på **Høy (8)**.
- Sannsynligheten er **Middels (5)** for at dette inntreffer.

Risikofaktor 2:

- Dersom teknologien som skal brukes i prosjektet ikke er moden eller velutviklet nok til å oppfylle målene så kan prosjektet ende opp med å ikke tilfredsstillere kravene i god nok grad. I verste konsekvens kan behov være fullstendig uinnfridd.
- Konsekvensen dersom dette inntreffer er **Svært Høy (9)**. Dette er en av pilarene for at prosjektet skal bli vellykket.
- Sannsynligheten vil være **Svært Lav (1)**. Siden Azure tilbyr et vidt spekter med dyp eller grunn integrasjon så vil det være veldig mange muligheter for å løse problemer. Et eksempel er dersom et system er avhengig av Legacy programvare. I stedet for å løse dette med en tjeneste i Azure så kan man rett og slett sette opp en VM (Virtuell Maskin) med denne programvaren.

Risikofaktor 3:

- Dersom teknologien ikke kan tilpasses bedriftens særskilte behov er valget av teknologi fullstendig feil. Det kan være sammenlignbart med å spise suppe med gaffel. Det kan fungere til en viss grad, men det er virkelig ikke verdt tiden eller innsatsen.
- Ergo kan konsekvensen sies å være **Høy (7)**, sannsynligheten er derimot **Lav (4)**, i stor grad av samme årsak som risikofaktor 2. Videre kan det nevnes at tjenestene i bruk av Salmo ikke stiller spesielle krav til infrastrukturen. Om det hadde gjort det så ville sannsynligvis en VM med Legacy programvare vært et reelt alternativ.

Risikofaktor 4:

- Urealistiske tidsplaner og budsjett kommer som følge av dårlig planlegging eller manglende støtte fra ledelsen eller omgivelsene.
- Konsekvensen av dette kan være at alle mål ikke blir oppnådd i tide, eller at enkelte deler blir mangelfulle. Dette vil bety en **Høy (7)** konsekvens. Prosjektgruppen velger ikke å fremstille dette som svært høy, siden systemet fremdeles vil kunne brukes.
- Sannsynligheten vil tilsvarende være **Medium (5)**.

Risikofaktor 5:

- For lite kompetanse vil føre til dårlig utførelse eller implementasjon. Vil ha innvirkning på hele eller deler av løsningen. Derfor er det viktig å ha, eller bygge, nødvendig kompetanse gjennom prosjektet.
- Konsekvensen vil være **Høy (8)**. Sannsynligheten vil være **Svært Lav (2)**. Det er mye dokumentasjon om temaene som prosjektet tar for seg, både gjennom utdanningen og litteratur på internett. I tillegg har prosjektgruppen allerede en del kompetanse gjennom utdanningen og erfaring.

Risikofaktor 6:

- Dårlig planlegging vil ofte skje gjennom forhastet planlegging eller manglende forståelse for arbeidet som inngår som del av utførelsen.
- Konsekvensen vil være **Høy (7)**. Sannsynligheten vil være **Lav (3)**. Prosjektgruppen har utarbeidet prosjekt om beslektede tema tidligere, men på mindre skala.

Risikofaktor 7:

- Konsistente eller endrede krav kan komme som følge av dårlig kommunikasjon mellom prinsipal og agent, eller som følge av at partene har ulike meninger om hva som går inn i kravene eller målene. Særlig derfor er det viktig å ha formulert disse på så klart vis som mulig. Kravene og målene må utarbeides i samarbeid mellom partene.
- Konsekvensen er **Medium (5)**, sannsynligheten er **Svært Lav (1)**.

Risikofaktor 8:

- Systemet vil ikke oppfylle sin funksjon dersom brukerne er ute av stand til å bruke det. Kan komme som følge av at systemet er veldig komplekst, eller at prosjektgruppen mister fokus rettet mot sluttbruker.
- Konsekvensen vil være **Høy (7)**. Sannsynligheten vil være **Lav (4)**. Prosjektgruppen er kjent med å skrive detaljert dokumentasjon, og er i stand til å tenke på utforming fra brukernes ståsted.

Risikofaktor 9:

- Gold-plating er et begrep som brukes om fenomenet hvor utvikler legger på flere og flere funksjoner som bruker ikke har behov for. Dette kan ta opp unødig med ressurser og tid. Utvikler burde ha fokus på å ferdigstille og polere de konkrete kravene brukerne har.
- Konsekvensen er **Medium (5)**, siden det kan gå utover systemets overordnede kvalitet. Sannsynlighet er **Svært Lav (0)**. Prosjektgruppen holder seg innenfor prosjektets rammer, og kontrollerer dette med veiledere.

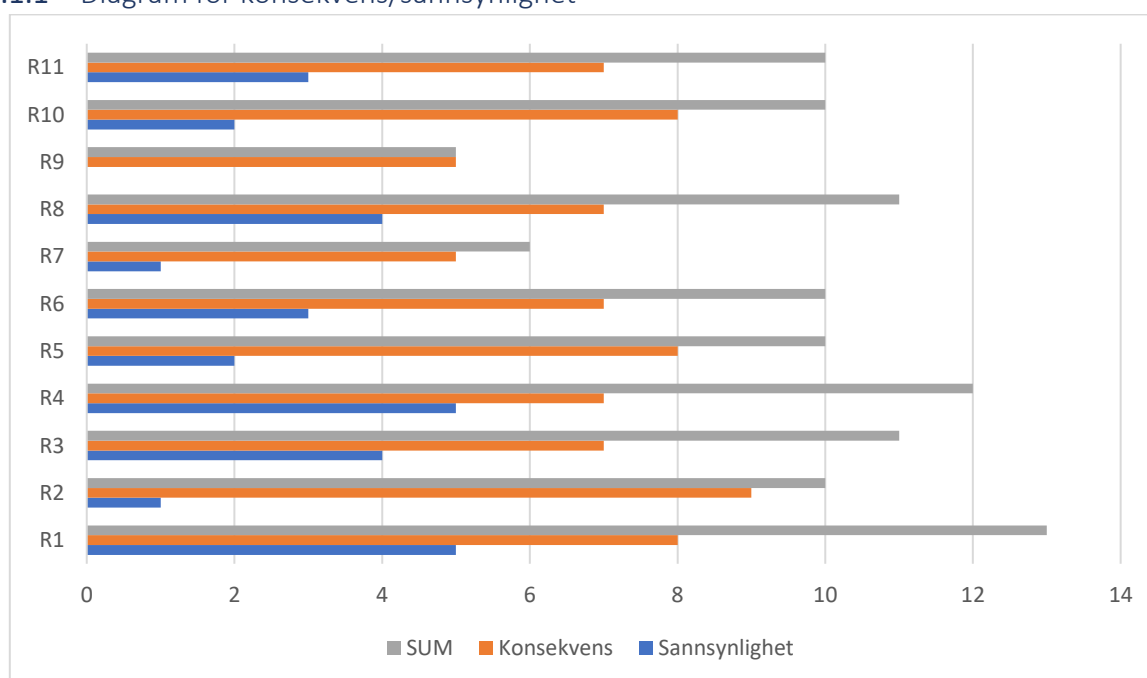
Risikofaktor 10:

- Især i Norge har det tidligere vært en del diskusjon om hvorvidt skytjenester er sikre nok, eller om det er trygt å la utenlandske selskap slik Microsoft håndtere bedriftskritiske tjenester i denne kapasiteten.⁹ Et spørsmål som ofte er hørt er: Hvor er våre data? Hvordan vet vi at ingen stjeler disse dataene fra datasentrene?
- Konsekvensen av denne risikoen vil i stor grad ha innvirkning på støtten fra bedriftens ledelse. Dersom de blir usikre på sikkerheten til systemet kan dette ha en **Høy (8)** konsekvens.
- Sannsynligheten settes til **Svært Lav (2)**. Azure har nettopp åpnet datasentre i Norge, som derfor blir nødt til å imøtekomme norske krav, i tillegg til at Azure uansett krypterer all data som lagres på sine datasentre¹⁰.

Risikofaktor 11:

- Dersom kunnskapsgrunnlaget er mangelfullt vil dette føre til ett feilaktig eller utdatert system, dersom litteraturen eller kunnskapene prosjektgruppen tar som utgangspunkt ikke stemmer vil beslutninger som tas undervis kunne tas på feil grunnlag. Et annet eksempel på en situasjon som kan oppstå er at systemet til dels blir utdatert allerede før det er lansert.
- Konsekvensen er **Høy (7)**, men sannsynligheten er **Lav (3)**. Dette er fordi veldig mye av støttelitteraturen som prosjektgruppen bruker kommer fra Microsoft, som har svært høye standarder for sin dokumentasjon i tillegg til at de oppdateres ofte.

6.1.1 Diagram for konsekvens/sannsynlighet

⁹ (Zachariassen, 2014)¹⁰ (Microsoft, Azure Data Encryption-at-Rest, 2020)

Tiltak for å kontrollere risiko

Her er ett utvalg av tiltak for å kontrollere de risikofaktorene nevnt i kapittel 7.1¹¹. Prosjektgruppen iverksetter konkrete tiltak for alle risikoene som oppnådde en sum over 10 i diagrammet for konsekvens/sannsynlighet i kapittel 7.1.1¹², siden de har forholdsvis høy konsekvens og sannsynlighet. Disse er faktorer 1, 3, 4 og 8.

Risikofaktor 1 – Utilstrekkelig kartlegging eller forståelse av bedriftens behov

For å forebygge mot planleggingsfeil slik som dette er det viktig at prosjektgruppen bruker nok tid til planlegging, også i oppstarten. Selv om det er her usikkerheten er størst, så er det også her det er størst mulighet til å gjøre endringer eller påvirke resultatet av prosjektet. Prosjektgruppen må bruke en del tid på planlegging i tillegg til at det er viktig å kommunisere godt og kontinuerlig med oppdragsgiver.

Gruppen skal:

- Holde jevnlige møter og statusoppdateringer
- Være ærlig med hverandre og oppdragsstiller.
- Utføre intervjuer og undersøkelser der det er hensiktsmessig med bedriften
- Bruke nok tid og arbeid på oppstarten

Risikofaktor 3 – Teknologien er ikke tilpasset bedriftens særskilte behov

Det må tas kontinuerlige vurderinger om hvorvidt teknologien eller tjenestene som skal tas i bruk passer for formålet. Prosjektgruppen skal:

- Kjøre jevnlige tester på løsningen, både som en helhet, men også på enkeltstående systemer involvert i prosessen.
- Påse at testene kommer nærmest mulig et reelt miljø
- Se på hva kunder som allerede bruker teknologien savner.

Risikofaktor 4 – Urealistiske tidsplaner eller budsjetter

Unødvendige skippertak eller hungersnød i prosjektet må unngås. Prosjektgruppen skal:

- Revidere prosjektplanen for kommende uke, fredagen forut.
 - Være realistiske, holde tidsbruken transparent og nøyaktig.
 - Holde tett dialog med oppgavestiller, når det kommer til både tidsplaner og ressurser
-

¹¹ Drøfting av risikofaktorer og konsekvensanalyse

¹² Diagram for konsekvens/sannsynlighet

Risikofaktor 8 – Løsningens fremtidige brukere eller driftere har ikke nødvendig kompetanse

Dette løses ved å dokumentere, brukertesting og legge opp til at opplæring kan utføres senere. Prosjektgruppen skal:

- Dokumentere kontinuerlig.
 - Kommunisere og drive brukertester med sluttbrukerne før, under og etter prosjektet.
 - Legge opp til at brukeropplæring kan utføres senere.
-

7 Analyse av kost og nytte

Her vises hvilke kostnader som forventes å oppstå som følge av implementasjon av skytjenester. Kostnadene definert i 7.2¹³ kommer direkte på grunn av tjenester som blir kjøpt og tatt i bruk på Azure. I 7.1¹⁴ defineres nytter som det ikke kan sette like konkrete tall på. Ved å fremstille kostnadene forbundet med implementasjon av skytjenester, så synliggjøres hvorvidt denne investeringen lønner seg for bedriften.

Ved at prosjektgruppen har en god kostnadmessig analyse allerede nå, så er det mindre sannsynlighet for å møte på «smellen» med uforutsette kostnader senere i prosjektet. I tillegg er det et greit utgangspunkt for å finne ut om hvorvidt oppdragsgiver er enig med, og aksepterer, disse kostnadene.

Innholdet i denne analysen tar utgangspunkt i dagens dato, 29.01.2020, og er selvsagt subjekt for endring over tid. Om markante endringer skjer mens prosjektet utføres, vil disse tallene oppdateres.

Det er verdt å nevne at prosjektgruppen ikke består av økonomer, og dette er en del av rapporten som fint kan overlates til eksperter, men er et godt punkt for diskusjon mellom oppdragsgiver og prosjektgruppe.

Prosjektgruppen drøfter ikke hvilke systemer og kostnader bedriften sitter på fra før av. Dette er fordi hovedfokus skal være på fremgangsmåten, verdien og selve implementeringen av systemet.

7.1 Ikke-kvantifiserbar nytte.

Her er det snakk om nytteverdier som man ikke kan sette en konkret monetær verdi på.

- Bedriften får økt omdømme ved å være tidlig ute med skytjenester.
- Bedriften trenger ikke å ta hensyn til egen infrastruktur og kapasitet i like stor grad som før.
- Mulighetene som blir skapt ved å ta i bruk en sky-basert infrastruktur.
- Skreddersydd komposisjon som passer til bedriften. Dette skaper effektivitet og dermed mindre kostnader for bedriften.

Disse er følgelig mer langsiktige enn den umiddelbare virkningen som bedriften får av innføringen. Dette er likevel viktige fordeler siden det vil gi strategiske fordeler.

¹³ Estimerte kostnader ved innføring av systemet

¹⁴ Ikke-kvantifiserbar nytte.

7.2 Estimerte kostnader ved innføring av systemet

Estimeringen tar for seg kostnader per måned, og starter med å se på lagringskontoen. Alle kostnader her er hentet fra Azure sin egen priskalkulator¹⁵. Dette er tjenester som trengs for å oppfylle målene definert tidligere i kapittel 3¹⁶. Dette er subjekt for endring – en av fordelene med Azure er at disse tjenestene enkelt kan skaleres opp eller ned ved behov.

Tjeneste	Alle priser er per måned										
Storage Konto: Data Lake Storage Gen2	<p>Lagringskonto tilpasset analyse av data. Siden Salmo har behov for dataprosessering og analyse, så vil denne typen være tilpasset deres bruk</p> <table border="1"> <tr> <td>Lagringsmengde</td> <td>100 TB</td> </tr> <tr> <td>Lese/skrive operasjoner</td> <td>4 MB, 10000 Operasjoner</td> </tr> <tr> <td>SUM</td> <td>23 447,70 NOK</td> </tr> </table>	Lagringsmengde	100 TB	Lese/skrive operasjoner	4 MB, 10000 Operasjoner	SUM	23 447,70 NOK				
Lagringsmengde	100 TB										
Lese/skrive operasjoner	4 MB, 10000 Operasjoner										
SUM	23 447,70 NOK										
VM-er for Legacy-system	<p>Kanskje har Salmo noen Legacy-systemer som må overføres og driftes videre på en VM. La oss si at de har behov for ett regnskapsprogram via en server med RDS – Remote Desktop Services. Her er et innledende ønske å ha en ekstra VM, for redundans og lastbalansering.</p> <table border="1"> <tr> <td>Størrelse</td> <td>B4MS</td> </tr> <tr> <td>vCPUer</td> <td>4</td> </tr> <tr> <td>RAM</td> <td>16 GB</td> </tr> <tr> <td>Delsum</td> <td>1 724,68 NOK</td> </tr> <tr> <td>SUM x 2</td> <td>3 449,36 NOK</td> </tr> </table>	Størrelse	B4MS	vCPUer	4	RAM	16 GB	Delsum	1 724,68 NOK	SUM x 2	3 449,36 NOK
Størrelse	B4MS										
vCPUer	4										
RAM	16 GB										
Delsum	1 724,68 NOK										
SUM x 2	3 449,36 NOK										
Azure Machine Learning (Modelloplæring)	<p>Den dataprosesseringen bedriften har behov for blir representert som eksempelvis Azure Machine Learning. Dette skal de bruke for å lære opp en modell for å overvåke og lese laksehelsen. Det er viktig å påpeke at dette er kun et eksempel på prosessering.</p> <table border="1"> <tr> <td>Størrelse</td> <td>D5 v2</td> </tr> <tr> <td>vCPUer</td> <td>16</td> </tr> <tr> <td>RAM</td> <td>56 GB</td> </tr> <tr> <td>SUM</td> <td>3 575,60 NOK</td> </tr> </table>	Størrelse	D5 v2	vCPUer	16	RAM	56 GB	SUM	3 575,60 NOK		
Størrelse	D5 v2										
vCPUer	16										
RAM	56 GB										
SUM	3 575,60 NOK										
Azure SQL Database	<p>De fleste Bedrifter har behov for SQL-servere. Ved at prosjektet skal bruke Azure, så er det ordensmessig å bruke Azure SQL Database i stedet for å videreføre en eksisterende SQL-server som VM. I tillegg er Azure SQL Databases mer kostnadmessig enn mange alternativer.</p> <table border="1"> <tr> <td>Forekomster</td> <td>1 x 730 timer</td> </tr> <tr> <td>Lagring</td> <td>5 x 32 GB</td> </tr> <tr> <td>SUM</td> <td>11 549 NOK</td> </tr> </table>	Forekomster	1 x 730 timer	Lagring	5 x 32 GB	SUM	11 549 NOK				
Forekomster	1 x 730 timer										
Lagring	5 x 32 GB										
SUM	11 549 NOK										
MÅNEDLIG SUM TJENESTER	42 021,07 NOK										

¹⁵ (Microsoft, Priskalkulator, 2020)

¹⁶ Prosjekt mål

8 Retningslinjer og standarder

8.1 Krav til dokumentasjon

Det stilles krav til god dokumentasjon underveis i prosjektet, og alt som er nødvendig å loggerføre eller rapportere skal dokumenteres grundig. I tillegg til dokumentene beskrevet i dette kapitlet kan det bli nødvendig å legge til flere rapporter underveis i prosjektet.

8.1.1 Forstudierapport

Et av de viktigste dokumentene i prosjektet er forstudierapporten. Denne må være klar så tidlig som mulig fordi den inneholder alle retningslinjer og krav til gjennomføring av prosjektet, samt at den kan fungere som en kontrakt mellom prosjektgruppe og kunde.

For dette prosjektet settes fristen for ferdigstilling av forstudierapporten til **5. Februar 2020**. Den skal først og fremst være godkjent av alle i prosjektgruppen før den legges frem for de andre partene. Begge parter skal etter dette ha satt seg inn i og være enige om betingelsene og retningslinjene som kommer frem i dokumentet, og godkjenne dette før videre arbeid treer i kraft.

8.1.2 Designrapport

Frist for innlevering den **18. Februar 2020**

8.1.3 Driftsrapport

Frist for innlevering den **5. Mai 2020**

8.1.4 Sluttrapport

Denne rapporten er den avsluttende dokumentasjonen for hele prosjektet. Denne vil inneholde en oppsummering av prosjektperioden, vurderinger av prosesser og måloppnåelse. Dokumentet fungerer som en skriftlig presentasjon av prosjektet og spiller en sentral rolle i videre arbeid.

Vilkårene for godkjenning av dette dokumentet er likt som for de andre rapportene. I hovedsak er det prosjektgruppen som har en felles godkjenning av rapporten før den leveres til godkjenning/revisjon. Det trengs ikke godkjenning fra kunde-siden, men det er likevel en fordel at kunden tar del i utviklingen av denne rapporten for å få med kundens erfaringer og synspunkter i løpet av prosjektets livstid.

8.2 Krav til kvalitetsgjennomganger

8.2.1 Dokumenter og rapporter

Disse må sees nøye gjennom av prosjektgruppens medlemmer før de leveres inn og kontrolleres av veiledere.

8.2.2 Microsoft Teams

Denne plattformen benyttes for å dele dokumenter og planer alle involverte i prosjektet. På denne måten sikres det at alle har tilgang til viktige dokumenter og godkjenning av disse.

8.2.3 Møter

For å sikre at alle involverte i prosjektet er enige om planene holdes det møter regelmessig for å gjennomgå eventuelle endringer i prosjektplanen.

8.2.4 Revisjon av dokumenter

For å sikre at alle involverte har god oversikt over hva som har blitt gjort så er det viktig med en god revisjonslogg. Denne viser hva som har blitt gjort når og hvilke dokumenter som er blitt endret på.

8.3 Krav til standarder og metoder

8.3.1 Dokumentmaler

I dette prosjektet benyttes det dokumentmaler som er gitt i faget *IDRI2007 Prosjekt- og dokumentasjonsarbeid* for å sikre kvalitet over dokumentasjonen.

8.3.2 Verktøy for implementasjon

8.3.2.1 GitHub

Benyttes for samkjøring og kontrollsjekking av skripts og andre dokumenter som inneholder kode.

8.3.2.2 Microsoft Azure

IaaS plattform som tilbyr de fleste tjenestene prosjektgruppen skal benytte seg av. Er snakk om VM-er, automatisering, osv. Bli drøftet i større detalj på designrapporten.

8.3.2.3 Visual Studio Code

Utviklingsprogrammet brukt av prosjektgruppen. Alt av skript og programmering blir gjort med denne programvaren.

8.3.3 Microsoft

8.3.3.1 Teams

Benyttes som samhandlingsverktøy under prosjektperioden. Brukes for å administrere dokumentasjon og kommunikasjons.

8.3.3.2 Word

Benyttes for all dokumentasjon og rapportskrivning.

8.3.3.3 MS Project

Benyttes for planlegging av prosjektet og inneholder en tidsplan for de forskjellige delene i prosjektet.

8.4 Endringshåndtering

Endringer i prosjekter både kan og vil oppstå og kan komme fra alle parter involvert i prosjektet. Det er viktig å ha gode rutiner for hvordan endringer håndteres. Prosjektgruppen har derfor utarbeidet noen punkter for hvordan endringer i prosjektet skal håndteres.

8.4.1 Dokumentasjon av endringens innhold

Dette innebærer å fastsette hva endringen dreier seg om og hvilke deler av prosjektet endringen omhandler. Der som en endring er ønskelig er det viktig å få dokumentert dette slik at alle involverte parter har en forståelse av hva endringen innebærer.

8.4.2 Konsekvensanalyse for prosjektet

Hvis det skal foretas en endring vil dette også føre til nye konsekvenser for prosjektet. Det er derfor viktig å analysere hva denne endringen medfører og opplyse alle parter om dette. Å gjennomføre en slik analyse er viktig for å skaffe oversikt over omfanget av endringen og dermed komme til en konklusjon om endringen er nødvendig og økonomisk lønnsom.

8.4.3 Beregne eventuell kost/nytte etter endring

Dette punktet inngår noe i konsekvensanalysen. Ved å se på konsekvenser av utgifter vil man også få en oversikt over kostnadsrammen endringen medfører i forhold til hvor stor nytte den har. Dette er, på samme måte som konsekvensanalysen, viktig for å vurdere lønnsomheten av endringen.

8.4.4 Godkjenning og aksept

Før en endring trer i kraft er det viktig at alle parter involvert i prosjektet godkjenner endringen og de konsekvenser denne medfører.

8.4.5 Loggføring av endring

Etter endringens godkjenning skal dette loggføres slik at man kan begynne å legge den inn i planene for prosjektet.

8.4.6 Planjustering

Når en endring er vedtatt og loggført er neste skritt å implementere den i de eksisterende planene for prosjektet. Dette vil innebære å justere planene og omorganisere prosjektets løp slik at endringen får plass uten at dette går over eksisterende frister for prosjektet.

8.4.7 Informering av interessenter

Når prosjektplanene er justert og endringen har fått plass må alle interessenter bli informert om endringene i prosjektet. Her er det viktig at alle parter får en innføring i at en endring forekommer (eller ikke, avhengig av de øvrige punkter), og hvordan dette påvirker planene for resten av prosjektet.

8.4.8 Gjennomføring av endring

Hvis det blir vedtatt at en endring skal gjennomføres og alle punktene over er gjennomført kan utførelsen begynne.

9 Prosjektorganisering

Her drøftes hvordan prosjektet blir organisert.

- 1. Oppdragsgiver – Sopra Steria ved Kristian Ellingsen Aamodt og Rune Haugen**
Oppdragsgiver og styringskomite for prosjektet er Sopra Steria representert av Kristian Ellingsen Aamodt og Rune Haugen.
- 2. Kvalitetskontroll – Jostein Lund**
Jostein Lund er veileder og fyller rollen som kvalitetskontroll.
- 3. Referansegruppe – Representanter fra Salmo.**
I ett reelt prosjekt vil bedriften som skal ta i bruk systemet stille med referansegrupper, som skal gi oss tilbakemelding på selve produktet underveis og etter prosjektet.
- 4. Prosjektgruppe – Emil Brasø og Endre Aalrust.**
Prosjektdeltakerne har samarbeidet gjennom flere gruppearbeider i skolegangen. Medlemmene har god kommunikasjon og har fått jevnt gode resultater sammen.

Ettersom prosjektgruppen ikke er kompleks så brukes prinsippet om Gjensidig tilpasning. Prosjektgruppen tilpasser seg hverandres behov, og er svært avhengig av god kommunikasjon og samarbeidsvilje. Det blir da heller ikke definert en lederrolle, men noen skiller har gruppen, når det kommer til kommunikasjon med eksterne. Denne rollen faller for det meste på Emil Brasø.

Arbeidsfordelingen må vurderes etter hvert som oppgavene klargjøres. Mange oppgaver må tas i plenum, et eksempel er denne rapporten, mens noen oppgaver er mindre og kan tas individuelt. Prosjektplanen kommer til å oppdateres med disse oppgavene etter hvert.

10 Referanser

Kotler, P., & Keller, K. L. (2016). *Markedsføringsledelse 4. utg.* Gyldendal.

Microsoft. (2020, Juli 04). *Azure Data Encryption-at-Rest*. Hentet fra Microsoft Docs: <https://docs.microsoft.com/en-us/azure/security/fundamentals/encryption-atrest>

Microsoft. (2020). *Prikalkulator*. Hentet fra Microsoft Azure: <https://azure.microsoft.com/nb-no/pricing/calculator/>

Microsoft. (2020, April). *Sammendrag av SLA for Azure-tjenestene*. Hentet fra Microsoft: <https://azure.microsoft.com/nb-no/support/legal/sla/summary/>

Sealab. (2020). *More than meets the eye*. Hentet fra Sealab: <https://www.sealab.no/>

Zachariassen, E. (2014, Oktober 2). *Norske bedrifter holder seg unna nettskyen*. Hentet fra tu.no: <https://www.tu.no/artikler/norske-bedrifter-holder-seg-unna-nettskyen/232375>



DESIGNRAPPORT

En automatisert reise fra datarom til sky

Emil Antoni Brasø, Endre Aalrust

Revisjonslogg

DATO	VERSJON	BESKRIVELSE	FORFATTER
18.02.2020	1.0	Opprettelse av førsteutkast	Endre, Emil
25.02.20 20	1.1	Revisjon av dokument	Endre, Emil
04.03.2020	1.2	Reviderte, revurderte om AADDs er nødvendig. Skrev flere definisjoner. La til SMART.xlsx	Emil
08.05.2020	1.3	La til en beskrivelse om VMware i kap. 2.1.1.2	Emil
08.09.2020	1.4	Rettskriving og revidering av dokumentstruktur	Endre
12.05.20	1.5	Lå inn SMART-diagram	Endre, Emil
14.05.20	1.6	Mindre rettskrivinger og presiseringer	Emil
15.05.20	1.7	Topptekst, referanser og stilsetting, formuleringer	Emil

Innholdsfortegnelse - Designrapport

Forkortelser og definisjoner.....	4
1 Om denne rapporten	6
1.1 Hensikten med dokumentet	6
1.2 Brukerundersøkelse gjennomført i forkant av designrapporten.....	6
1.3 Avgrensning	6
1.4 Kort om kunden og kundens behov	7
2 Tekniske løsninger.....	7
2.1 Verktøy og ressurser	7
2.1.1 Testmiljø.....	8
2.1.2 Forarbeid.....	11
2.1.3 Innledende vurdering og migrering	11
2.1.4 Systemdrift etter migrering og generelle verktøy	12
2.1.5 Oversikt over tjenestene i Azure etter migrering	13
2.1.6 Ideelt miljø	15
3 Prosessdesign.....	16
3.1 Testmiljøet	16
3.2 Fremgangsmåte	17
3.2.1 Overordnet prosessløp	17
3.2.2 Klargjøring av testmiljø	18
3.2.3 Oppdage og vurdere miljøet	19
3.2.4 Test-migrering og migrering	19
4 Om alternative løsninger	20
5 Løsningsdesign for sluttbruker.....	21
5.1 Tjenester	21
5.1.1 Azure	21
5.1.2 Office 365.....	21
5.1.3 Skype for Business.....	21
5.1.4 OneDrive	21
5.2 Struktur	22
5.3 Ressurser i Azure.....	23
5.3.1 Virtuelle maskiner	24
5.3.2 Azure Automation.....	25
5.3.3 Bruk av virtuelle maskiner	26
6 SMART-diagrammet.....	27
Figurer	29

Tabeller	29
Referanser	30
Vedlegg	32

Forkortelser og definisjoner

Begreper/ Forkortelser	Definisjon
AD	Active Directory – Microsofts katalogtjeneste for å håndtere brukere, ressurser og brukerrettigheter m.m.
Server	En datamaskin med programvare som tilbyr en eller flere tjenester til andre datamaskin over et nettverk
OKV	Opplevd Kunde Verdi – Begrep om verdien en sluttkunde opplever av et produkt, kontra «kostnaden» forbundet med å levere produktet.
VPN	Virtual Private Network – Også kalt tunnel-teknologi. Brukes til å sette opp private nettverksforbindelser mellom nett, blant annet for å få tilgang til interne ressurser utenfra.
Ubuntu	Linux-operativsystem. Mye brukt for servere. For alle tilfeller i dette prosjektet brukes Ubuntu 18.04, som er LTS-versjon. LTS betyr Long-Term-Support, som vil si at denne versjonen er støttet i lang tid framover.
VM	Virtuell Maskin
DC	Domenekontroller
AADDS	Azure Active Directory Domain Services – Lar oss videreføre domenetjenester fra tradisjonell AD i Azure.
AAD	Azure Active Directory – Active Directory i Azure. Tar hånd om brukerautentisering og rettigheter.
POC	Proof of Concept – Vil si en slags prototype, hvor hensikten er å bevise at konseptet fungerer.
LDAP	Protokoll som brukes til oppslag i en katalogtjeneste på en server/server
On-premises	Infrastruktur som er «plassert» hos bedriften
IaC	Infrastructure as Code - Prinsipp som handler om å kunne deklare og rulle ut infrastruktur ved hjelp av kode og automasjon.
SQL	Structured Query Language – Brukes til å kjøre spørringer opp mot databaser.
Nested Virtualization	«Virtualisering ved hjelp av Virtualisering» Det opprettes et virtualisert miljø på en allerede virtualisert server.
ARM	Azure Resource Manager
RDP	Remote Desktop – Brukes til å koble seg til en ekstern maskin.
SSH	Secure Shell – Brukes til å koble seg til kommandolinjen på en ekstern maskin
CMD/Ledetekst	Kommandotolker i Windows

Azure	En gruppe skytjenester som benyttes til å organisere, utvikle, administrere og distribuere programmer.
Maskinlæring/machine learning/ML	Maskinlæring er en spesialisering innenfor kunstig intelligens hvor man bruker statistiske metoder for å la datamaskiner finne mønstre i store datamengder.
Hyper-V	Programvare som benyttes til å lage VMer.
Automasjon	Prosesen hvor en eller flere arbeidsoppgaver gjøres gjennom for eksempel et skript og ikke manuelt.
SLA (Service Level Agreement)	Avtale som beskriver kvalitet, tilgjengelighet og ansvar mellom kunde og tjenesteleverandør.
Legacy-System	En gammel metode, teknologi, datasystem eller applikasjon.

Tabell 1 - Begreper/forkortelser

1 Om dette dokumentet

1.1 Hensikten med dokumentet

Denne designrapporten skrives i forbindelse med en bacheloroppgave som utarbeides av Emil Antoni Brasø og Endre Aalrust i samarbeid med bedriften Sopra Steria.

Hensikten med denne designrapporten er å dokumentere hvordan prosjektgruppen vil imøtekomme behovene stilt av kunden i caseoppgaven. Mange av emnene her er tatt opp i forstudierapporten¹, men drøftes i større detalj på bakgrunn av at denne rapporten skal beskrive den ferdige løsningen som skal være gjennomført ved prosjektslutt.

I denne rapporten beskrives det hvilke teknologier som brukes, hvordan infrastrukturen i den ferdige løsningen blir tilrettelagt og litt om hvordan de forskjellige delene i prosessen blir utført. Utførelse er tema for driftsrapporten².

Innledende forklares det kort om kundens behov, samt at det begrunnes hvorfor prosjektgruppen velger å bruke de gitte verktøyene. Deretter beskrives sammensetningen av tekniske løsninger som skal komme sammen til det helhetlige systemet.

Prosjektgruppen vurderer flere alternativ enn det som er beskrevet i forstudiet, men kun en endelig løsning presenteres, i tråd med målene definert i forstudierapporten.

Målet med denne rapporten er å utarbeide et design som er mest mulig hensiktsmessig for casen³ som prosjektgruppen definerte sammen med veilederne.

1.2 Brukerundersøkelse gjennomført i forkant av designrapporten.

I forkant av denne rapporten utførte prosjektgruppen en brukerundersøkelse⁴ om skytjenester, hvor hensikten var å se på hvordan sluttbrukere hos forskjellige bedrifter opplever skytjenester, både som begrep, men også i forhold til påvirkning i arbeidsdagen. Gjennom dette ble det avduket noen hensyn hos sluttbrukeren, og prosjektgruppen ble klar over noen ekstra synsvinkler. Det er viktig å ha et kundefokus, og denne undersøkelsen er ett av tiltakene som har blitt gjennomført for å imøtekomme dette fokuset.

Selve analysen av brukerundersøkelsen er vedlagt. Den tar for seg et sammendrag av alle resultatene, prosjektgruppens konklusjon som følge av undersøkelsen, i tillegg til en gjennomgang av noen interessante svar. Denne analysen er blitt brukt i sammenheng med utformingen av dette dokumentet.

Fotnoter og referanser inkluderes der det er relevant, men det er greit å huske at det meste av dokumentasjonen som prosjektgruppen benytter kommer fra Microsoft.

1.3 Avgrensning

Resultatmålene med dette prosjektet, som definert i forstudiet, er å utarbeide:

- En fremgangsmåte som beskriver selve prosessen med å migrere dagens løsning fra on-premises til en hel eller delvis skybasert løsning.
- Ett system som utnytter skybasert teknologi
- En analyse av typen løsning; hel eller delvis og hva som vil gagne bedriften i det lange løp.
- En forståelse av kunders behov og OKV (Opplevd Kunde Verdi)⁵
- Migreringsprosessen skal påvirke den daglige driften minst mulig.

¹ (Brasø & Aalrust, Forstudierapport, 2020)

² (Brasø & Aalrust, Driftsrapport, 2020)

³ (Brasø & Aalrust, Caseoppgave, 2020)

⁴ (Brasø & Aalrust, Analyse av brukerundersøkelse, 2020)

⁵ (Kotler & Keller, 2016)

I tråd med denne avgrensingen skal derfor det endelige systemet inneholde hovedsakelig to komponenter:

- Ett fungerende testsystem. Skal fungere som POC (Proof of Concept)
- En effektivisert framgangsmåte ved hjelp av automatisering og IaC (Infrastructure as Code).

Det planlagte systemet og framgangsmåten begrunnes i dette dokumentet. Selve systemet er derimot ikke gjennomført, dokumentert og testet fullt ut før arbeidet begynner på driftsrapporten. Dersom deler av løsningen må endres som følge av hva prosjektgruppen avdekker under testing av miljøet, vil dette nevnes og forklares i sluttrapporten⁶. Videre vil ikke steg som går med på oppsettet dokumenteres her, men i driftsrapporten.

1.4 Kort om kunden og kundens behov

Kunden som har bestilt oppdraget er den tenkte oppdrettsbedriften Salmo AS. Oppdraget går i all hovedsak ut på å undersøke om migrering av Salmo AS sine systemer til hel eller delvis sky gagnar bedriften.

Totalt er de femti ansatte, og driver på med betydelige mengder med datalagring, samt noe prosesseringsbehov. For å gi et praktisk eksempel på prosesseringen, bruker prosjektgruppen Machine Learning. Dette er en arbeidstype som bedrifter i oppdrettsnæringen tar i bruk, men hovedsaken er kun at det skal gi et håndfast eksempel på prosessering. Bedriftens on-premises infrastruktur består av en Windows server med Hyper-V installert, som driver noen VMer (Virtuelle maskiner) og en SQL-server (Structured Query Language). I tillegg brukes serveren til å lagre store mengder med data. Oppsummert er det stort behov for lagring og prosessering av data.

2 Tekniske løsninger

Prosjektet tar utgangspunkt i Azure, og derfor gir det mening å ta i bruk verktøy som er innebygd i Azure, eller som er en del av Azure miljøet allerede. Løsningen skal hovedsakelig inneholde fire deler – eller arbeidsoppgaver, derfor deles dette kapitlet opp etter det. Noen av verktøyene kommer til å brukes over flere av disse.

Ved at Salmo AS tar i bruk Azure får de en løsning som gir flere fordeler. De drar nytte av forenklet administrasjon, høy grad av skalering og sikkerhet⁷, samt at de kan skilte med at de er tidlig ute med ny teknologi.

2.1 Verktøy og ressurser

Selve implementasjonen deles hovedsakelig inn i tre kategorier:

- Testmiljø
- Forarbeid som er uavhengig av testmiljø
- Innledende vurdering og Migrering

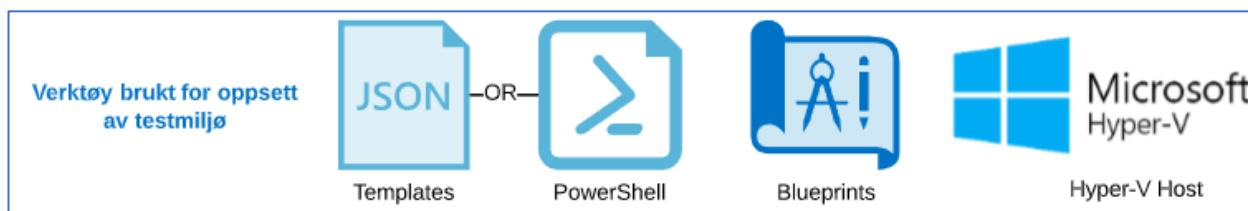
Det blir ikke tatt spesielt hensyn til drift og oppsett som kommer etter implementasjonen, men de blir nevnt i kapittel 3⁸.

⁶ (Brasø & Aalrust, Sluttrapport, 2020)

⁷ (Microsoft, 2020)

⁸ Prosessdesign

2.1.1 Testmiljø



Figur 1 - Verktøy brukt for oppsett av testmiljø

For testmiljøet disponerer prosjektgruppen to miljøer:

- Ett miljø i Azure disponert av Sopra Steria.
- En fysisk server disponert av Jostein Lund ved NTNU.

2.1.1.1 Azure miljø disponert av Sopra Steria

For bruk under prosjektet har prosjektgruppen fått tilgang til ett Azure abonnement med et tak på 4000 NOK per måned. Det er blitt iverksatt flere tiltak for å minimere kostnaden. Imidlertid vil faktisk prosessering være ganske begrenset i miljøet, siden prosjektet i stor grad går ut på migrering og utrulling ved hjelp av automasjon og IaC. Det blir gjentatte testmigreringer, men det blir få tjenester som kjører i skyen over lengre tid.

Under forhandlinger ble følgende kostnadsestimat fremstilt for testmiljøet:

Tjeneste	Beskrivelse	Pris i måneden
WVD- Windows Virtual desktop	Pooled, 100 Users, 0.85 Peak concurrency, 0.05 Off peak concurrency, 220 Usage hours/month, multi-session, light workload, D4s v3 (4 vCPU(s), 16 GB RAM), S10 (128 GiB, 500 IOPS) Disks	kr2604
Azure AD	Free tier, Standard directory objects, 730 User forest hours, 730 Resource forest hours.	kr888,49
Azure SQL Database	Single Database, vCore Purchase Model, General Purpose Tier, Provisioned, Gen 5, 1 2 vCore instance(s) x 20 Hours, 32 GB Storage, 0 GB Backup Storage	kr115,68
	Sum i måneden	kr3 608,17

Tabell 2 - Kostnadsestimat testmiljø

Merk: disse kostnadene er ett godt stykke over hva som forventes brukt. Det er lagt inn en del overhead, dette er særlig for å hindre problemer knyttet med utilstrekkelig kostnadsplanlegging.

Verktøyet Azure Cost Calculator⁹ ble brukt for kostnadsestimatet, og eksportert estimat finnes som vedlegget *exportedEstimate*.¹⁰

⁹ (Microsoft Azure, 2020)

¹⁰

Azure Automation¹¹ benyttes for å sette opp automatisk opprensing av ressursgruppene. Slik kan man være sikker på at ingen ressurser kjører lengre enn hva det er behov for. I tillegg gjøres mindre tester i så stor grad som mulig utenfor miljøet disponert av Sopra Steria.

2.1.1.2 Beskrivelse av den fysiske serveren disponert av Jostein Lund

Prosjektgruppen får låne en relativt liten server for prosjektet. Den fungerer likevel greit for testmiljøet sin del. Windows Server skal installeres på den, samt Hyper-V med noen tilhørende VMer. Prosessen med å sette opp dette blir drøftet mer i kapittel 3.1¹². Det er ingen kostnad knyttet til dette for Sopra Steria eller NTNU sin del.

Det ville ikke ha utgjort særlig forskjell dersom on-premises miljøet hadde tatt i bruk eksempelvis VMware i stedet for Hyper-V, da fokus uansett ligger på å ta i bruk tjenester i Azure som tillater migreringen, slik som Azure Migrate. Det er noe egen funksjonalitet for VMware, som vises innledende i kapittel 4.7 av driftsrapporten. Forskjellen ligger i hvordan Azure Migrate oppdager og kopierer maskiner fra serveren, men ellers vil prosessen for å migrere fungere tilnærmet likt for VMware som for Hyper-V.

Selve serveren har 32 GB RAM, samt en CPU med klokkefrekvens på 3.60GHz. Utfyllende informasjon om enheten er vedlagt i form av dokumentet *Win16serv.pdf*. Per dette dokumentets revisjon 1.2 er det planlagt å sette opp følgende ressurser:

- Fem arbeidsstasjoner
- En SQL-server
- En mengde test-data.
- En Domenekontroller
- En Ubuntu-server med OpenVPN, for å tillate tilkobling med testmiljøet eksternt.

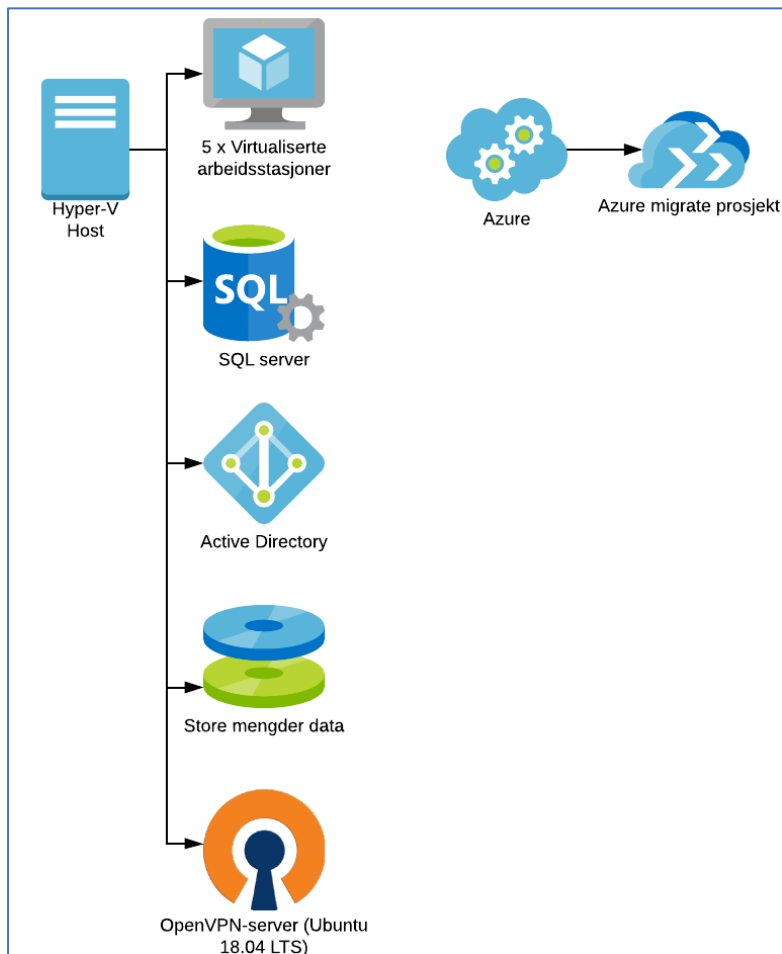
For dette formålet blir det satt opp en rekke VMer på Hyper-V serveren.

¹¹ (Microsoft, 2018)

¹² Testmiljøet

2.1.1.3 Ressurser i testmiljøet

Denne figuren forsøker å beskrive ressursene som blir satt opp i testmiljøet. Det er greit å presisere at det kun ser slik ut i de innledende delene av prosjektet.



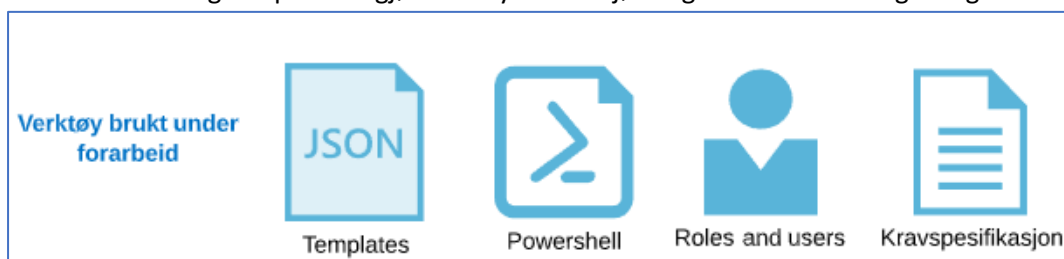
Figur 2 - Ressurser i testmiljøet

Oppsummert så vil testmiljøet bestå av følgende ressurser før selve migreringsprosessen:

- Hyper-V host med:
 - Fem Arbeidsstasjoner
 - En SQL-server
 - En Active Directory Domenekontroller
 - En OpenVPN server
- Azure med et Azure Migrate prosjekt.

2.1.2 Verktøy for bruk under forarbeid

Forarbeid her vil gå ut på å klargjøre det fysiske miljøet og Azure for testmigreringen.



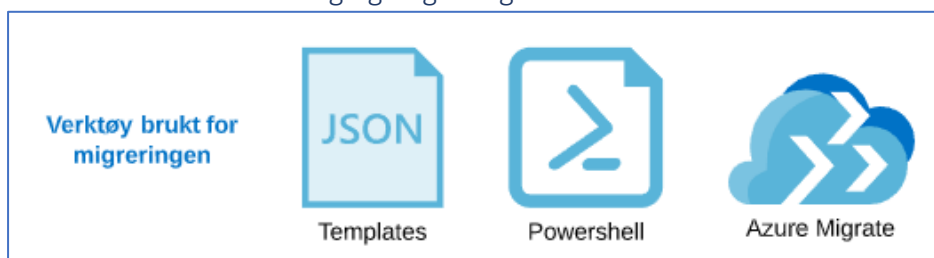
Figur 3 - Verktøy brukt under forarbeid

Verktøy	Brukes til
Active Directory Roles and Users	Å gi Migrate Appliance tilgang til å administrere Azure Migrate og diverse andre applikasjoner
Kravspesifikasjon	Verifisere at ressursene møter kravene til migrering.
ARM Templates	Rulle ut ressurser som trengs for å migrere, deriblant: <ul style="list-style-type: none"> - Azure Migrate Project
PowerShell	Brukes til å klargjøre den fysiske Hyper-V hosten til migrering. Brukes også til å laste ned en del av programvaren som trengs for å vurdere og migrere miljøet.

Tabell 3 - Verktøy brukt under forarbeid

- For Azure må det settes tillatelser slik at migreringen blir mulig.
- Det rulles ut ett Azure Migrate prosjekt ved hjelp av en ARM-template ARM står for Azure Resource Manager.
- Det blir klargjort en Hyper-V server med skriptet *MicrosoftAzureMigrate-Hyper-V.ps1*¹³.
- Det må verifiseres at Hyper-V serveren imøtekommer kravene ved hjelp av kravspesifikasjonen fra Microsoft¹⁴

2.1.3 Innledende vurdering og migrering



Figur 4 - Verktøy brukt for migreringen

For å utføre selve migreringen benyttes tre verktøy innebygd i Azure:

- ARM Templates¹⁵
- Azure Migrate

¹³ (Microsoft, Prepare for assessment and migration of Hyper-V VMs to Azure, 2020)

¹⁴ (Microsoft, Azure Migrate support matrix, 2020)

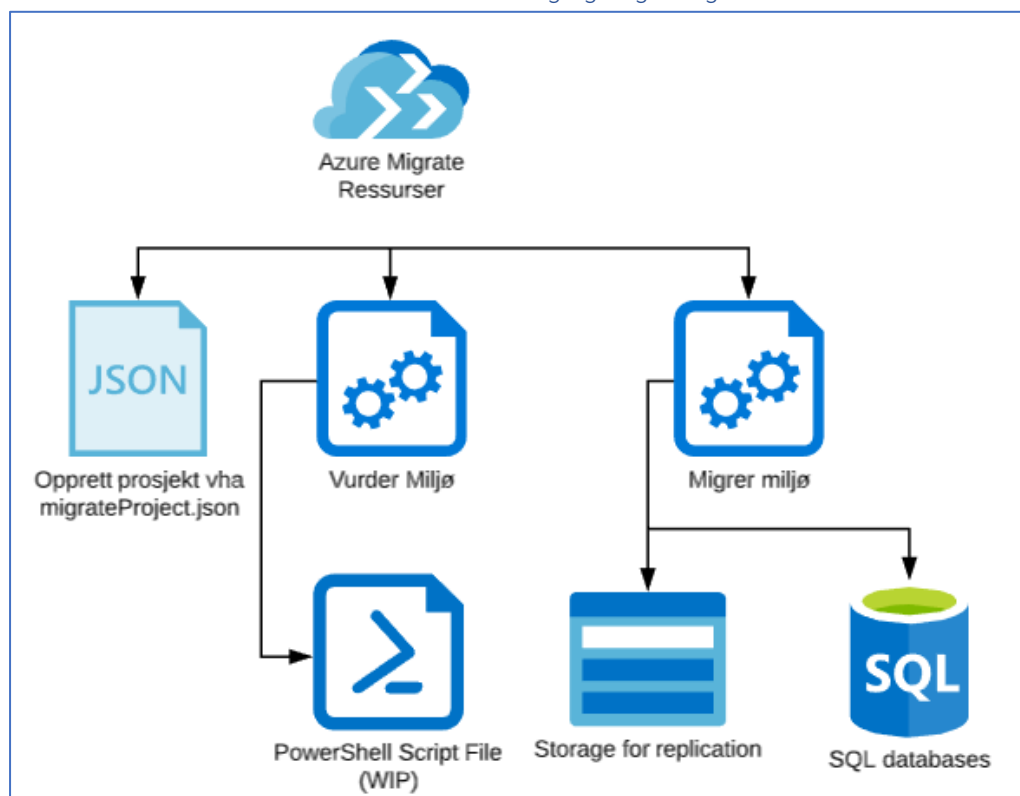
¹⁵ (Microsoft, What are ARM templates?, 2020)

- PowerShell

Azure Migrate¹⁶ utfører flere oppgaver, det brukes til å lage estimat for miljøet on-premises, samt at verktøyet inneholder funksjonalitet som skal utføre selve migreringsarbeidet. Dette kan gjøres for miljøer med Hyper-V, VMware eller fysiske servere.

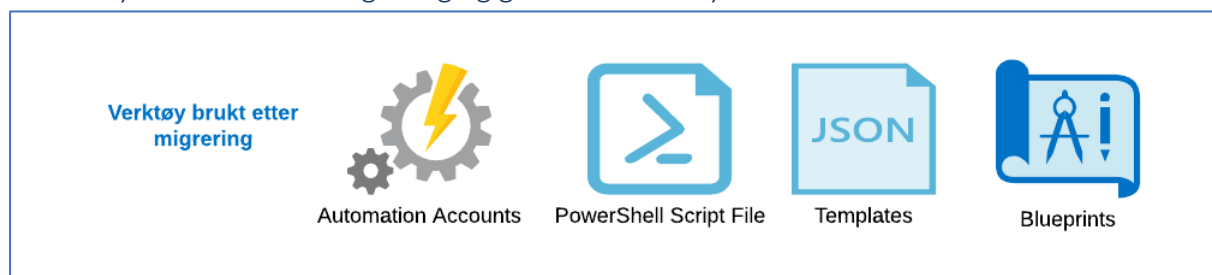
Verktøy	Brukes til
Azure Migrate	Brukes til å utføre vurderingen, og deretter migreringen.
PowerShell	Automatisere installasjon av Migrate Appliance og utrulling av Templates.

2.1.3.1 Ressurser under innledende vurdering og migrering



Figur 5 - Ressurser under innledende vurdering og migrering

2.1.4 Systemdrift etter migrering og generelle verktøy



Figur 6 - Verktøy brukt etter migrering

¹⁶ (Microsoft, About Azure Migrate, 2019)

Her beskrives det hvordan driften av Azure miljøet bli etter endt migrering og hvilke verktøy som tas i bruk for å gjøre den hverdagslige driften enklest mulig for de ansvarlige og for brukerne av systemet.

2.1.4.1 Azure Automation

Etter migreringsprosessen er gjennomført vil prosjektgruppen benytte automasjon til å holde miljøet ved like og begrense kostnadsbildet. En del av dette innebærer å fjerne ubrukte ressursgrupper og ressurser som ikke er i bruk. Dette blir gjort gjennom et powershellskript som lastes opp til Azure Automation¹⁷ og kjøres regelmessig ved hjelp av funksjonen Runbook. Azure Automation brukes også til noen andre oppgaver i Azure, blant annet for oppgaver i daglig drift.

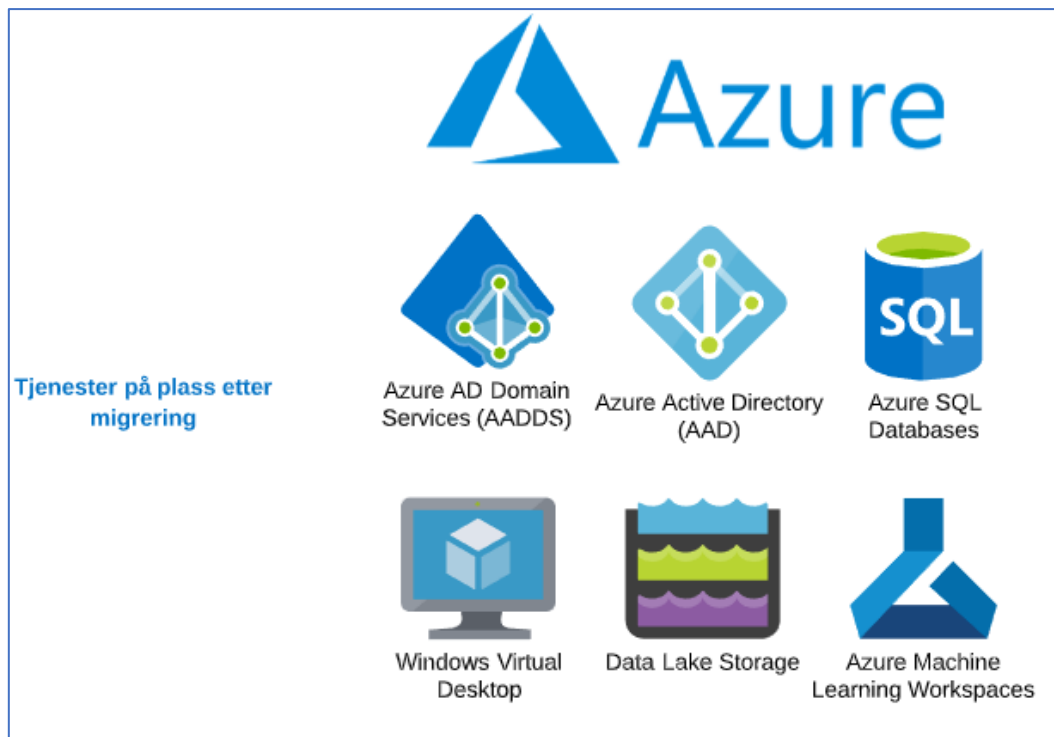
2.1.4.2 Azure Blueprints

Azure Blueprints¹⁸ vil tas i bruk som en overordnet planskisse for å definere strukturen i Azure. Her kan det defineres hvordan miljøet fungerer overordnet, men også små detaljer som restriksjoner og rettigheter for forskjellige brukere. Komponenter som ARM templates, Azure policy og andre Azure funksjoner vil også være inkludert i slike Blueprints. Disse rulles ut til miljøet og kan forandres etter behov. Det vil hovedsakelig brukes til å administrere miljøet.

2.1.4.3 ARM Templates

ARM templates er en løsning for IaC. For det meste vil ARM templates brukes for å rulle ut nye ressurser etter behov, for eksempel ved økt ressursbehov eller lignende. Vil også brukes sammen med Blueprints.

2.1.5 Oversikt over tjenestene i Azure etter migrering



Figur 7 - Tjenester etter migrering

¹⁷ (Microsoft, An Introduction to Azure Automation, 2018)

¹⁸ (Microsoft, What is Azure Blueprints?, 2019)

Azure AD Domain Services (AADDs)	Tilbyr domene-tjenester slik som LDAP, Access Control og Group Policy Objects. Fullstendig kompatibelt med Windows Server Active Directory
Azure Active Directory (AAD)	Identitetskontroll i Azure. Brukes mest med interne ressurser eller SaaS tjenester (Office365)
Azure SQL Databases	PaaS løsning for relasjonelle databaser. SLA-oppetid på 99.99%. Migreres ved hjelp av Database Migrate Assistant, som er innebygd i Azure Migrate
Windows Virtual Desktop	Skrivebord og app virtualisering. Erstatte mange av tjenestene Remote Desktop Services gjorde tidligere.
Data Lake Storage	Tilpasset analyse-tjenester og kombinerer et hierarkisk navneområde med skalerbarheten og prisen som Azure Blob Storage tilbyr.
Azure Machine Learning Workspaces	Ressurser brukt i Azure Machine Learning. Etter at utvikler har utviklet en modell kan den ruller ut her, hvor den kan testes ved hjelp av kteinere, Kubernetes m.m. Holder en oversikt over alle testene og historien til modellen.

Tabell 4 - Tjenester i Azure etter migrering

2.1.5.1 Azure SQL Databases

Azure SQL Databases¹⁹ er Azure sin egen PaaS (Platform as a Service) løsning for relasjonelle databaser. Ifølge Azure er det det mest kostnadseffektive alternativet for SQL databaser i skyen²⁰. I tillegg garanteres 99.99% oppetid gjennom SLA-en til Azure²¹. De fleste bedrifter bruker databaser til å administrere kunder eller lignende, ved at Salmo benytter seg av Azure SQL Databases sikres også bedre redundans, skalering og oppetid.

2.1.5.2 Azure Storage konto: Data lake Storage gen2

Data lake Storage²² er lagringskontoer tilpasset analyse av data. I tillegg til at det er tilpasset Big Data så passer det godt for oppbevaring av store mengder data slik som Salmo har behov for. Det er mye video, som bedriften trenger å ta ned og bruke fra tid til tid, samt at prosjektene som involverer prosessering må kunne ta i bruk denne dataen så enkelt og effektivt som mulig. Dette tilrettelegger denne typen lagringskonto for.

I tillegg vil dette brukes for å videreføre fildelingen fra miljøet on-premises. For dette formålet opprettes såkalte *fileshares*²³, som VMer og brukere kan benytte.

¹⁹ (Microsoft, Azure SQL Database Documentation, 2020)

²⁰ (Chirapurath, 2019)

²¹ (Microsoft, Sammendrag av SLA for Azure-tjenestene, 2020)

²² (Microsoft, Introduction to Azure Data Lake Storage Gen2, 2020)

²³ (Microsoft, What is Azure Files?, 2018)

2.1.5.3 AAD (Azure Active Directory) og AADDS (Azure Active Directory Domain Services)

I stedet for en tradisjonell domenekontroller vill det brukes AAD (Azure Active Directory)²⁴ og AADDS (Azure Active Directory Domain Services)²⁵ til å utføre oppgavene som domenekontrolleren gjorde i det fysiske miljøet.

I det første utkastet av dette dokumentet ble det vurdert å ta i bruk AADDS for å videreføre tradisjonelle domenetjenester i Azure. I stedet ble det ved revisjon besluttet at dette ikke er nødvendig. For en bedrift på denne størrelsen blir det sett på som mer hensiktsmessig å opprette alt på nytt, det vil si at strukturen på domenekontrolleren og domenet blir liggende igjen på miljøet som det migreres bort fra.

AADDS ville tilbudt domene-tjenester slik som LDAP (Lightweight Directory Access Protocol), domeneinnmelding og kerberos autentisering som er fullstendig kompatibelt med Windows Server Active Directory. I stedet brukes AADDS for å tillate bruk av WVD (Windows Virtual Desktop), samt oppsett av GPO-er (Group Policy Objects) på arbeidsstasjonene.

AAD er skybasert identitetskontroll og brukes for det meste sammen med SaaS (Software as a Service) applikasjoner eller interne ressurser i skyen. For eksempel brukes dette til å gi brukerkontoer tilgang til å utføre operasjoner på Azure, eller gir tilgang til Office365.

2.1.5.4 WVD (Windows Virtual Desktop)

Kravene til å bruke WVD²⁶ er å sette opp AAD og AADDS. WVD erstatter i stor grad RDS (Remote Desktop Services) som tradisjonelt ble brukt til å la flere brukere dele et arbeidsmiljø. Dermed er det ikke nødvendig å sette opp virtuelle RDS servere, samtidig som at det ikke er behov for å sette opp enkeltstående VMer for bruk som arbeidsstasjoner. WVD brukes uten en tradisjonell RDS server som vi drifter, og er derfor uavhengig av dette. I stedet for å administrere infrastrukturen er det bare nødvendig å tenke på reservoaret av VMer som skal brukes for WVD.

2.1.6 Ideelt miljø

Ideelt er det ønskelig at minst mulig av tjenestene blir drevet gjennom grunn integrasjon etter at migreringsprosessen er ferdig. Med grunn integrasjon menes det all integrasjon hvor det blir emulert et eldre system eller tjeneste ved hjelp av Virtualisering. Et eksempel på dette kan være et foreldet regnskapssystem som kun kjører på Windows XP. I et slikt tilfelle ville man ha tatt i bruk en egendefinert VM for å kjøre Legacy programvare.

Dersom dette unngås, og alle tjenestene drives som tjenester innebygd i Azure, medfører dette flere fordeler:

Høyere interoperabilitet mellom tjenestene. De er tilpasset samme miljø.

Høyere skalering. Ett virtualisert Legacy system vil normalt sett ikke kunne skaleres like enkelt.

Samordnet administrasjon. Alle tjenestene kan vises og endres gjennom Azure portalen eller kommandolinje.

Enkel utrulling og automasjon. Ved hjelp av powershell og ARM vil prosessen med å rulle ut de fleste av tjenestene som Azure tilbyr gjøres vesentlig enklere.

Enkel kjøring av skript. Ved å aktivere Azure Remote Management vil det bli enkelt kjøre powershell-skript på de virtuelle maskinene.

Bedre SLA. Azure kan tilby svært høy oppetids SLA for mange av de innebygde tjenestene.

²⁴ (Microsoft, What is Azure Active Directory?, 2019)

²⁵ (Microsoft, What is Azure Active Directory Domain Services?, 2020)

²⁶ (Microsoft, What is Windows Virtual Desktop?, 2020)

Minst mulig teknisk gjeld. Teknisk gjeld kan gjerne defineres som «... alt det uferdige, unødvendig kompliserte, og utdaterte i løsningene våre som hindrer oss i å drifte, forvalte og videreutvikle så effektiv som vi burde være i stand til»²⁷. Det kan eksempelvis være problemer som oppstår som følge av krav ved Legacy programvare, eller så kan det være forbundet med snarveier eller uferdige løsninger som er gjort på et tidligere tidspunkt.

Tabell 5 - Fordeler med Azure - Ideelt miljø

Prosjektgruppen tar likevel forbehold om at det ikke bestandig er mulig å foreta en dyp integrasjon av absolutt alt. Dette ansees som fullstendig mulig for bedriften Salmo, siden de har få, eller ingen, Legacy systemer som må tas i betraktning. Delkapittel 2.1.5²⁸, og resten av dokumentet ellers, tar utgangspunkt i dette.

3 Prosessdesign

Dette kapitlet vil gå detaljert gjennom systemet som skal settes opp, i tillegg til at selve migreringsprosessen blir presentert og begrunnet. Dette skal levere best mulig resultat for sluttbrukeren. Merk at det er i driftsrapporten at alle deler av prosessen og implementasjonen dokumenteres, dette er en beskrivelse av hvordan det ønskes gjort.

I sin helhet vil hele prosessen som skal gjennomgås bestå av følgende deler:

- Oppsett av testmiljø
- Infrastrukturen på Azure
- Automasjon ved hjelp av Automation Account, ARM, PowerShell og Blueprints.
- Migrerings-verktøyet Azure Migrate som er innebygd i Azure.

Hovedfokuset er å beskrive selve prosessen. Verktøyene er beskrevet tidligere i dokumentet, i kapittel 2.1²⁹.

3.1 Testmiljøet

For å repetere så består testmiljøet av følgende elementer:

- En fysisk server disponert av Jostein Lund. Denne skal symbolisere det on-premises miljøet som Salmo AS befinner seg på per i dag.
- Et miljø i Azure disponert av Sopra Steria. I utgangspunktet skal dette miljøet være tomt når migreringen utføres, med unntak av ressurser som er nødvendig for å utføre dette. Vil være elementer slik som et Azure Migrate Project, og lagrede templates.

²⁷ (Reinertsen, Hagberg, & Lees, 2013)

²⁸ Oversikt over tjenestene i Azure etter migrering

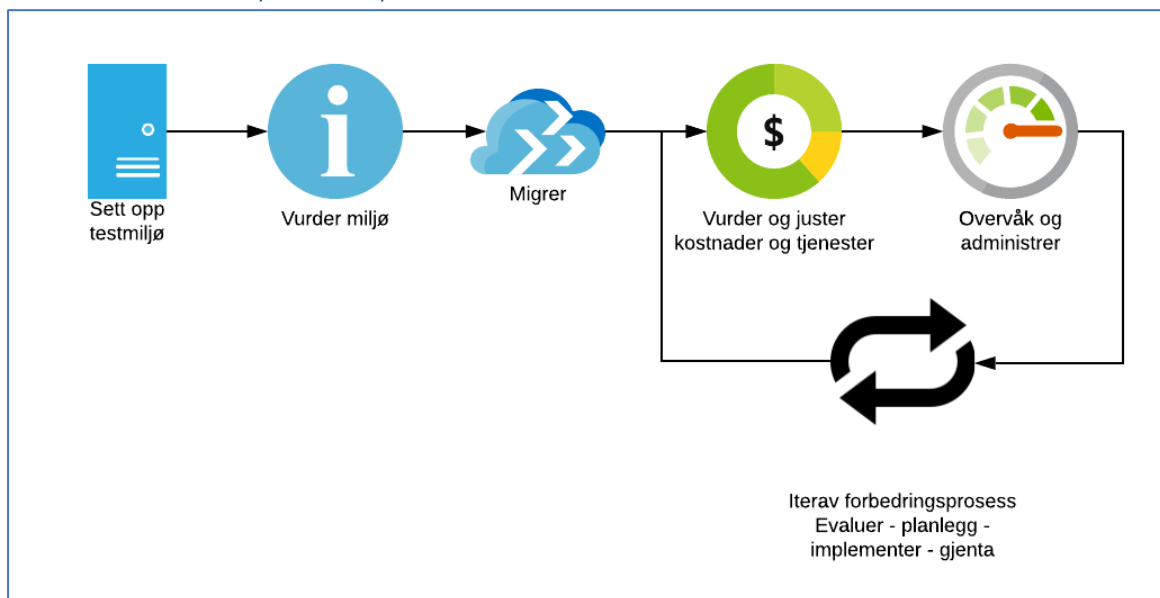
²⁹ Verktøy og ressurser

3.2 Fremgangsmåte

Selve prosessen med å gjennomføre migreringen deles opp i tre faser: forarbeidet, vurdering og migrering. Figur 7 viser hele den overordnede prosessen, men det er spesielt de tre første (i blått) prosjektgruppen tar for seg i dette dokumentet. De resterende delene på figuren er for kontinuerlig drift og forbedring etter implementasjon, derfor blir ikke de tatt opp i dette dokumentet. For selve migreringen tok prosjektgruppen inspirasjon i en guide³⁰ fra Microsoft.

Per dokumentets revisjon 1.7 er det verdt å presisere skillet mellom dette prosessdesignet og prosjektets overordnede struktur. Prosessdesignet tar for seg forarbeidet, vurderingen og migreringen knyttet til selve migreringsprosessen, mens den overordnede strukturen til prosjektet består av oppstart, utførelse og ferdigstilling. Denne strukturen beskrives i detalj i kapittel 2.1 av driftsrapporten.

3.2.1 Overordnet prosessløp



Figur 8 - Overordnet prosess

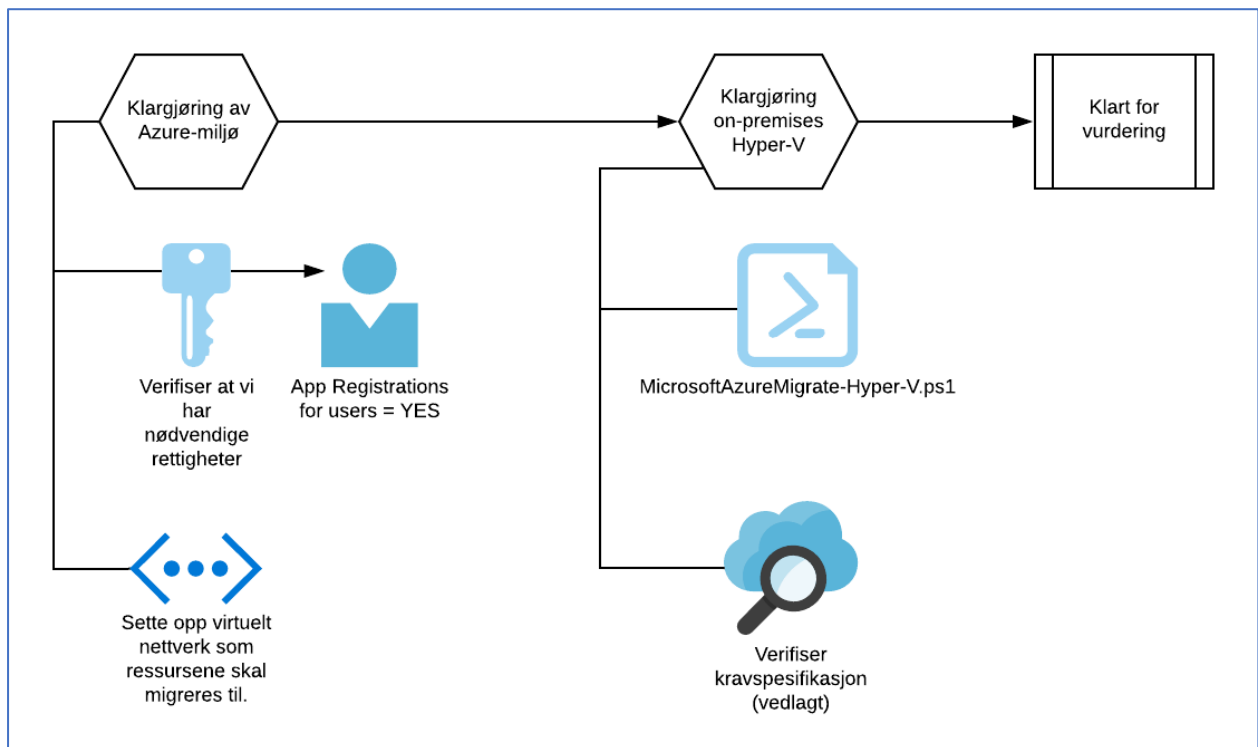
Overordnet vil prosessen skje over følgende seks punkter:

1. Oppsett av testmiljø som representerer bedriftens on-premises infrastruktur
2. Vurdering av miljøet før migrering
3. Selve migreringen
4. Vurder og juster kostnader og tjenester
5. Overvåk og administrer
6. Iterativ forbedringsprosess

Vedlagt er en skisse som inneholder alle fasene som blir beskrevet i de neste kapitlene.

³⁰ (Microsoft, Migrate Hyper-V VMs to Azure, 2020)

3.2.2 Klargjøring av testmiljø



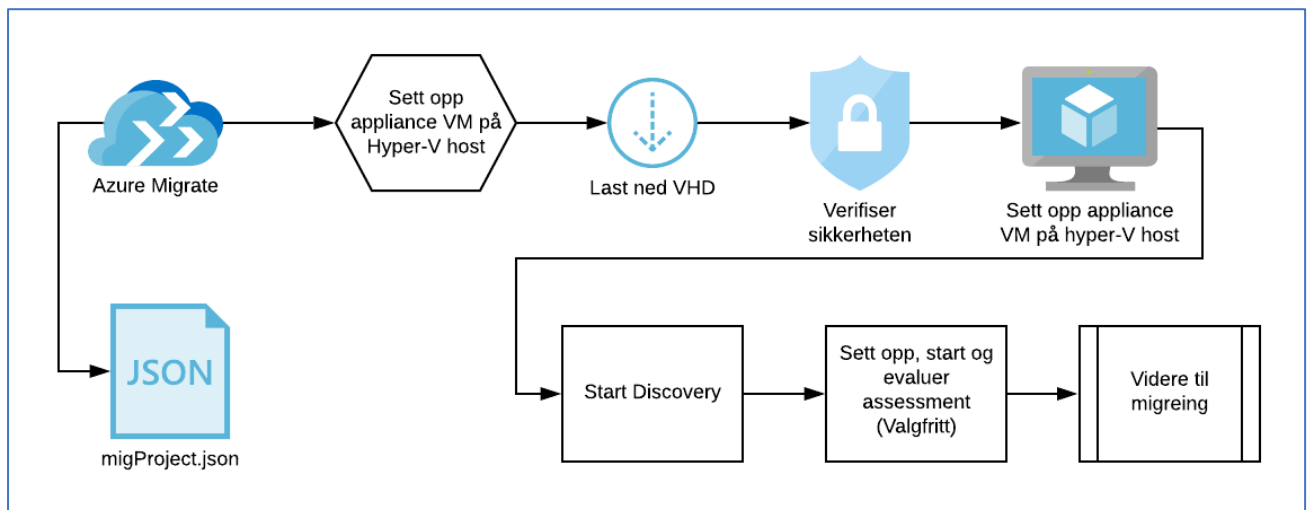
Figur 9 - Prosess – klargjøring av testmiljø

For å klargjøre testmiljøet er det nødvendig å klargjøre både den skybaserte infrastrukturen og den fysiske infrastrukturen. Under Azure må rettigheter tildeles, samt at det må settes opp et virtuelt nettverk.

På den fysiske siden må Hyper-V serveren være klargjort gjennom skriptet *MicrosoftAzureMigrate-Hyper-V.ps1*. Deretter må det verifiseres at serveren oppfyller kravene spesifisert av Microsoft, ved hjelp av kravspesifikasjonen³¹.

³¹ Vedlegget *Support Matrix for Hyper-V assessment.pdf*

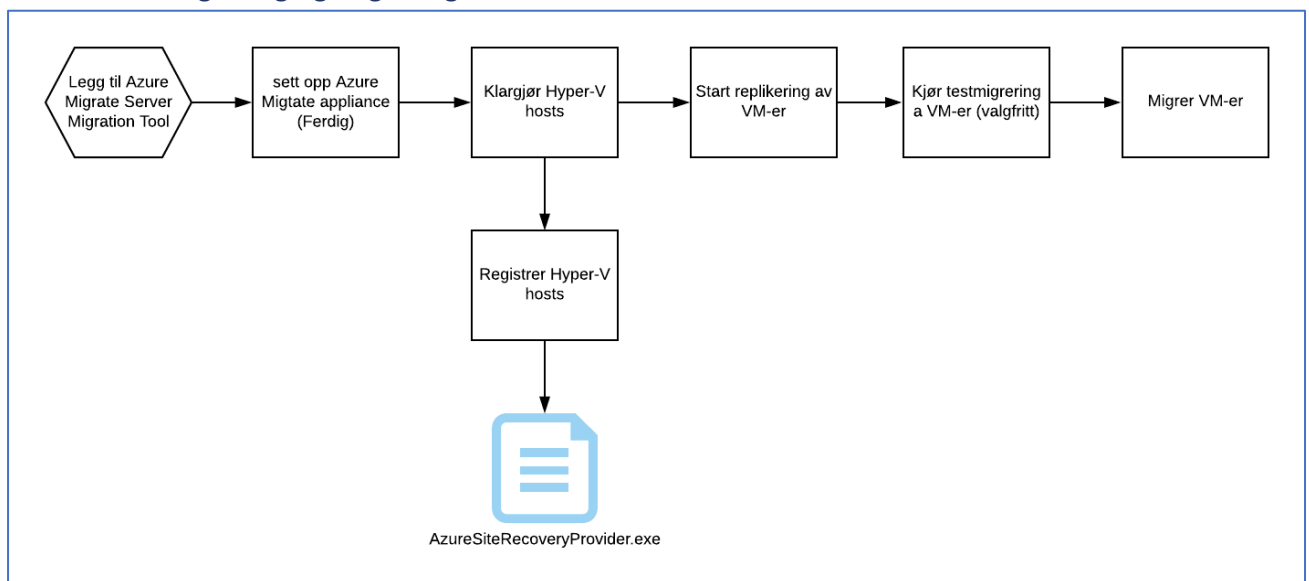
3.2.3 Oppdage og vurdere miljøet



Figur 10 - Prosess – oppdag og vurder miljøet

Før prosessen med å migrere kan starte må discovery gjennomføres, ved hjelp av en appliance-VM på Hyper-V serveren. Dette er for at Azure Migrate prosjektet skal oppdage, og se VMene aktuelle for migrering. Deretter kan en assessment kjøres. Dette er valgfritt, men svært hensiktsmessig da det gir en ide på hvordan kostnadsbildet og oppsettet kommer til å se ut etter migrering.

3.2.4 Test-migrering og migrering



Figur 11 - Prosess – test-migrering og migrering

Før en endelig migrering gjennomføres kan det kjøres en testmigrering på et utvalg av VMer. Dette er ofte lurt, for da ser man om det er noen problemer med helsen til VMen som må rettes opp i før migrering. Når faktisk migrering senere blir gjennomført kan man med en gang etter migrering velge å skru disse av, for å forhindre overlapp. Et annet alternativ er å la begge systemene kjøre samtidig, slik at man har tid til å avdekke potensielle problemer med de migrerte maskinene.

4 Om alternative løsninger

Det er valgt å ta utgangspunkt i fullstendig Public Cloud og derfor forklares det her hvilke alternative løsninger som eksisterer. Det er i all hovedsak to løsninger, uavhengig av sky-leverandør:

- Hybrid Cloud
- Private Cloud

Ved Hybrid Cloud ville Salmo AS sine tjenester bli separert etter behov. Noen tjenester, slik som arbeidsstasjoner og utvikling av ML-modeller ville ha blitt utført på on-premises infrastruktur. Dette ville ha forebygget mot latens eller nedetid i forhold til avvik i forhold til nett-tilgang osv. På lignende vis ville andre tjenester blitt holdt der det er hensiktsmessig.

Prosjektgruppen mener uansett at fullstendig Public Cloud er den beste løsningen, grunnet størrelsen til bedriften. Med denne løsningen kontra hybrid, slipper bedriften blant annet å investere særlig i infrastruktur, noe som i seg selv er en betraktelig kilde for besparelse. Dette er begrunnet ytterligere i forstudiet.

En Private Cloud vil for det meste være utelukket for denne bedriften. Særlig grunnet store krav til infrastruktur, i tillegg til implementasjon og kontinuerlig drift av systemene. Drift er til en viss grad nødvendig på Public Cloud også, men her trenger man ikke å ta hensyn til infrastrukturen.

5 Løsningsdesign for sluttbruker

Dette kapitlet omhandler hvordan sluttbrukeren i bedriften vil ta i bruk systemene gjennom daglig drift. Sentrale spørsmål for dette er:

- Hvordan vil sluttbruker oppleve systemet?
- Hva er det første de møter ved innlogging?
- Hvor vil de finne de forskjellige funksjonene de har bruk for?

Tjenestene og strukturen beskrevet i dette kapitlet vil benyttes av forskjellige typer ansatte i bedriften.

5.1 Tjenester for sluttbruker

Dette er en samling av tjenester som vil benyttes i løsningen for systemet. Her inngår både tjenester som skal kjøres lokalt og som leies fra andre tilbydere.

5.1.1 Azure

Brukes for å organisere tjenester og det virtuelle miljøet som inneholder bedriftens IT-system. I Azure kan man administrere og overvåke virtuelle miljøer. Azure brukes primært for å bygge, teste og rulle ut virtuelle infrastrukturer. Gjennom bruk av Azure får man tilgang til en mengde verktøy som kan brukes til å gjøre den daglige driften av en bedrift mer smidig og man kan også automatisere oppgaver. Ved å ta i bruk Azure vil Microsoft ivareta sikkerheten på all dataen som går igjennom systemet.

5.1.2 Office 365

Dette er en softwarepakke fra Microsoft som tilbyr en rekke programmer og tjenester som det er behov for i en bedrift. Denne pakken inneholder programvare som Word, Excel og PowerPoint. Dette er standardprogram som dekker de fleste behov til dokumentasjon, budsjettering og økonomistyring, og presentasjoner til møter eller lignende. Disse blir regelmessig oppdatert med nye funksjoner og sikkerhetsoppdateringer. Denne pakken inkluderer også skybaserte løsninger for e-post, kalender. Deriblant Microsoft Teams. Avhengig av lisensen som er kjøpt, følger det også med en rekke andre programmer for kundebehandling, fakturering og lignende som kan være nyttig for styring av bedriften.

5.1.3 Skype for Business

Dette er en applikasjon som er kjent for de fleste og brukes i sammenheng med kommunikasjon mellom ansatte og eventuelt eksterne. Med Skype For business kan man sende direktemeldinger og foreta lyd og videosamtaler der man har mulighet til å dele bilde av skrivebordet på PC-en. Man kan også opprette møter slik at flere deltakere kan delta i en lyd- eller videosamtale.

5.1.4 OneDrive

Med Office365 får hver bruker en terrabyte skylagring hos Microsoft. OneDrive er tilgjengelig fra alle enheter og fra hvor som helst. Også her vil Microsoft ha ansvaret for sikkerheten rundt dataene som lagres.

5.2 Struktur

Systemet som blir innført vil være basert på Microsofts egne løsninger og eventuelle superbrukere, i form av en IT-administrator eller lignende, vil for det meste ha interaksjoner med disse. For å benytte seg av det nye systemet vil vedkommende nå gjøre alt via en portal i nettleseren. Denne portalen vil gi tilgang til bedriftens Azure konto og alle medfølgende ressurser. Her vil man kunne opprette ressursgrupper, VMer, nettverkskomponenter og mye mer.

The screenshot displays the Azure portal interface, organized into several sections:

- Azure services:** A row of ten service tiles: Create a resource, Resource groups, Automation Accounts, Subscriptions, Blueprints, Virtual machines, App Services, Storage accounts, SQL databases, and More services.
- Recent resources:** A table listing recently viewed resources.

Name	Type	Last Viewed
AutomationCleanup (TestBruker/AutomationCleanup)	Runbook	12 min ago
automationCleanupTest	Virtual machine	12 min ago
testMiljø	Resource group	12 min ago
TestBruker	Automation Account	6 d ago
AzureAutomationTutorialScript (TestBruker/AzureAutomationTutorialScript)	Runbook	6 d ago
AzureAutomationTutorialPython2 (TestBruker/AzureAutomationTutorialPytho...	Runbook	6 d ago
Azure subscription 1	Subscription	6 d ago

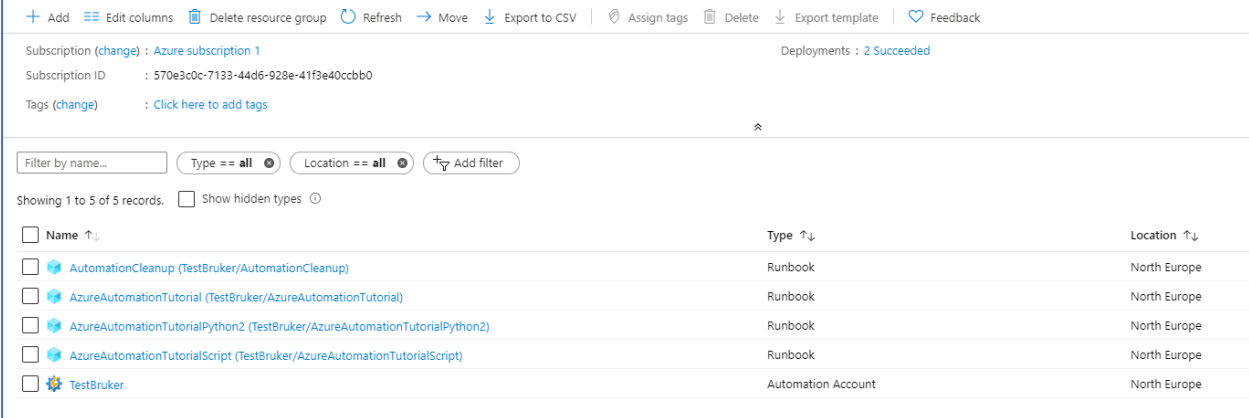
- Navigate:** Tiles for Subscriptions, Resource groups, All resources, and Dashboard.
- Tools:** Links to Microsoft Learn, Azure Monitor, Security Center, and Cost Management.

Figur 12 - Azure brukergrensesnitt

5.3 Ressurser i Azure

Gjennom Azure kan superbruker enkelt opprette og koble seg til VMer av ulike størrelser og typer. Her vil typen maskin med tanke på størrelse eller operativsystem variere basert på behov.

For eksempel vil en SQL-server ofte være store og ressurskrevende maskiner, mens vanlige arbeidsstasjoner vil være i mindre størrelse. All tilkobling til maskiner skjer enten via SSH (Secure Shell) eller RDP (Remote Desktop Protocol). Disse maskinene finnes i forskjellige ressursgrupper som brukeren selv definerer:



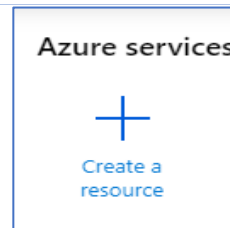
<input type="checkbox"/> Name ↑↓	Type ↑↓	Location ↑↓
<input type="checkbox"/> AutomationCleanup (TestBruker/AutomationCleanup)	Runbook	North Europe
<input type="checkbox"/> AzureAutomationTutorial (TestBruker/AzureAutomationTutorial)	Runbook	North Europe
<input type="checkbox"/> AzureAutomationTutorialPython2 (TestBruker/AzureAutomationTutorialPython2)	Runbook	North Europe
<input type="checkbox"/> AzureAutomationTutorialScript (TestBruker/AzureAutomationTutorialScript)	Runbook	North Europe
<input type="checkbox"/> TestBruker	Automation Account	North Europe

Figur 13 - Ressurser og ressursgrupper

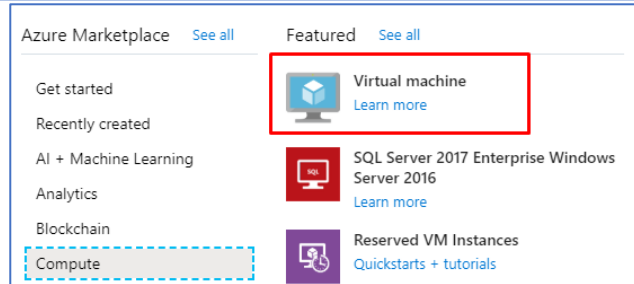
5.3.1 VMer (Virtuelle Maskiner)

Her beskrives hvordan VMer kan opprettes gjennom Azure portalen. Alternativ kan dette gjøres gjennom PowerShell eller ARM

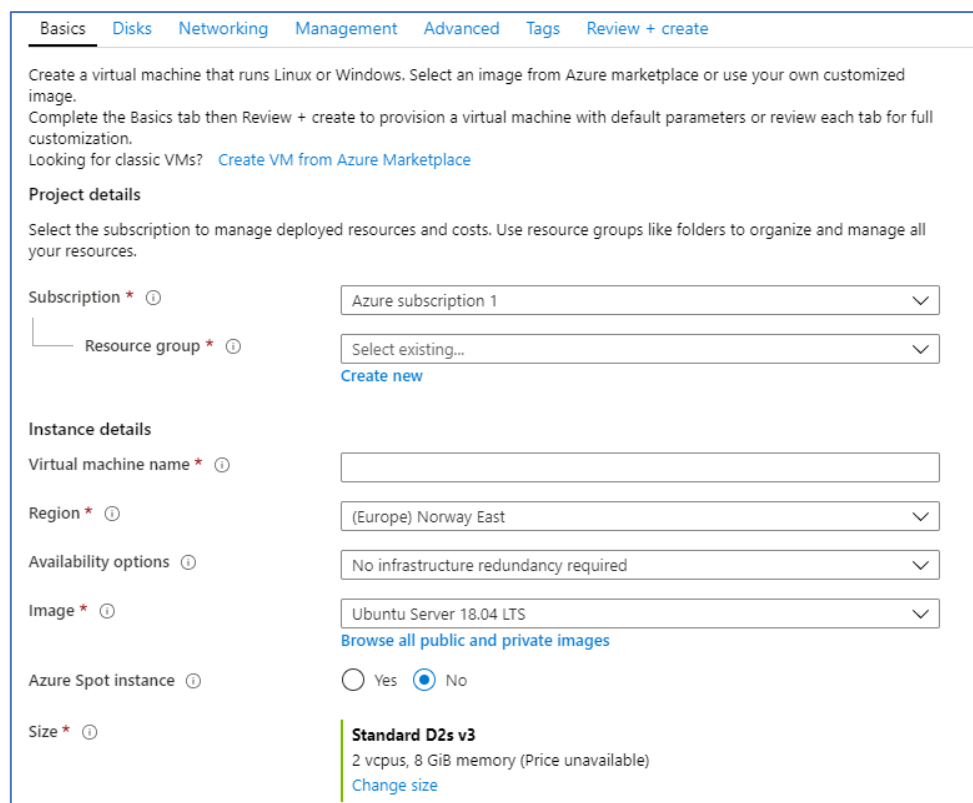
1. Klikker på knappen *Create a resource*



2. Velger deretter hvilken ressurs som skal opprettes. I dette tilfellet *Virtual Machine*



3. Så kommer man til en oversikt over hvordan den nye VMen skal se ut.



Basics Disks Networking Management Advanced Tags Review + create

Create a virtual machine that runs Linux or Windows. Select an image from Azure marketplace or use your own customized image. Complete the Basics tab then Review + create to provision a virtual machine with default parameters or review each tab for full customization. Looking for classic VMs? [Create VM from Azure Marketplace](#)

Project details

Select the subscription to manage deployed resources and costs. Use resource groups like folders to organize and manage all your resources.

Subscription * ⓘ Azure subscription 1

Resource group * ⓘ Select existing... [Create new](#)

Instance details

Virtual machine name * ⓘ

Region * ⓘ (Europe) Norway East

Availability options ⓘ No infrastructure redundancy required

Image * ⓘ Ubuntu Server 18.04 LTS [Browse all public and private images](#)

Azure Spot instance ⓘ Yes No

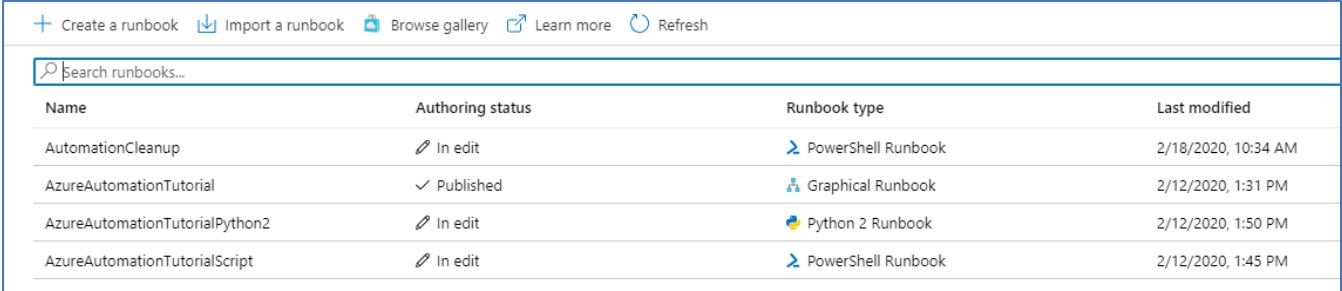
Size * ⓘ **Standard D2s v3**
2 vcpus, 8 GiB memory (Price unavailable) [Change size](#)

Figur 14 - Opprettelse av VM

5.3.2 Azure Automation

I Azure er mulighetene store for å automatisere mange funksjoner i miljøet, men dette krever at sluttbruker kan skrive kode i JSON og PowerShell. Som et eksempel på dette vil bare en liten del av denne funksjonen beskrives her ved et enkelt skript som renser opp miljøet.

Under Azure Automation Accounts kan Runbooks opprettes. Disse brukes i sammenheng med en Schedule som sier hvor ofte og når en Runbook skal kjøres. En Runbook kan ansees som et skript som lagres og kjøres i Azure sammen med en Schedule.

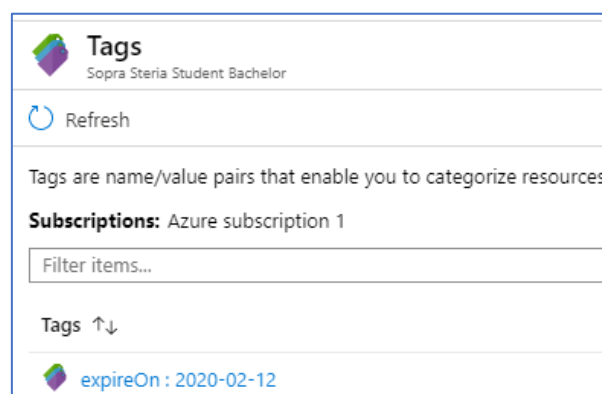


Name	Authoring status	Runbook type	Last modified
AutomationCleanup	In edit	PowerShell Runbook	2/18/2020, 10:34 AM
AzureAutomationTutorial	Published	Graphical Runbook	2/12/2020, 1:31 PM
AzureAutomationTutorialPython2	In edit	Python 2 Runbook	2/12/2020, 1:50 PM
AzureAutomationTutorialScript	In edit	PowerShell Runbook	2/12/2020, 1:45 PM

Figur 15 - Azure Runbooks

Skriptet *AutomationCleanUp.ps1*³² går gjennom Azure miljøet klokken 05.00 hver morgen og sletter ressurser som har gått ut på dato og ressursgrupper som er tomme. I store Azure miljøer kan man fort miste oversikten over hvilke ressurser som ikke er i bruk lenger og dette kan koste bedriften penger uten at de vet om et. Derfor er dette et smart verktøy å bruke for å opprettholde orden i Azure miljøet samtidig som det kan spare bedriften penger.

Skriptet baserer seg på at hver enkelt ressurs har en TAG som beskriver når en ressurs har gått ut på dato. En TAG er en komponent som brukeren kan sette på en hvilken som helst ressurs og er fullstendig egendefinert. TAG-en som benyttes i dette skriptet kalles «expireOn» og inneholder en dato. Hvis denne datoen er eldre enn dagens dato vil ressursen bli slettet.

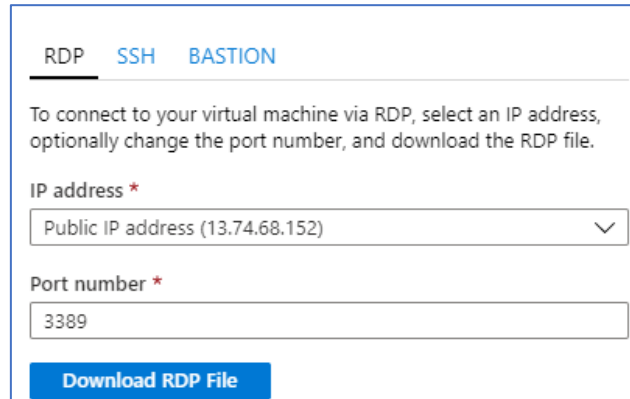


Figur 16 – Tags

³² (Brasø & Aalrust, CleanupAutomation.ps1, 2020)

5.3.3 Bruk av virtuelle maskiner

Som nevnt over vil all bruk av virtuelle maskiner skje over enten RDP eller SSH. Ved RDP benyttes IP-adressen til den virtuelle maskinen. Gjennom Azure kan man enkelt laste ned en RDP-fil og benytte denne som et aksesspunkt. Det samme kan gjøres via SSH, men her må man bruke CMD på den lokale maskinen.



RDP **SSH** BASTION

To connect to your virtual machine via RDP, select an IP address, optionally change the port number, and download the RDP file.

IP address *

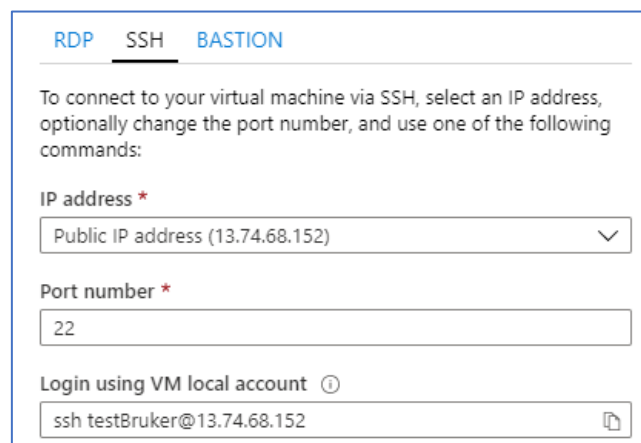
Public IP address (13.74.68.152) ▾

Port number *

3389

Download RDP File

Figur 17 - RDP



RDP **SSH** BASTION

To connect to your virtual machine via SSH, select an IP address, optionally change the port number, and use one of the following commands:

IP address *

Public IP address (13.74.68.152) ▾

Port number *

22

Login using VM local account ⓘ

ssh testBruker@13.74.68.152

Figur 18 - SSH

6 SMART-diagrammet

Under skrivingen av designrapporten tok prosjektgruppen i bruk et såkalt SMART-diagram³³. Selve dokumentet er vedlagt, men det er et verktøy brukt for å planlegge migreringsprosjekt. SMART er her en forkortelse for Strategic Migration Assessment & Readiness Tool. Det bruker følgende nøkkelområder til å vurdere hvor klar en gitt bedrift er for å utføre en migrering av sitt on-premises miljø:

- Business Strategy
- Migration Project Plan
- Partner Support
- Application discovery and assessment
- Migration Process
- Technical skilling
- Landing Zone
- Cloud Operations
- Security Compliance

Skjemaet stiller en del spørsmål som gjelder disse nøkkelområdene, hvor to eksempler er:

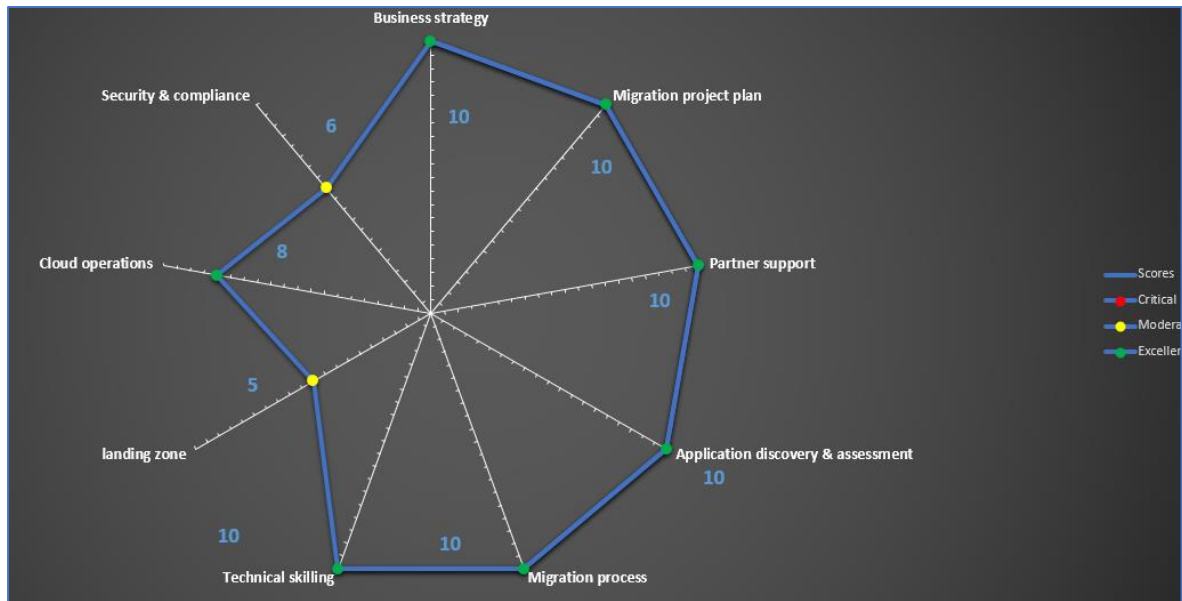
- «Do you have a strategy for backup, disaster recovery and monitoring in Azure? »
- «Do you have a strategy for OS patching and configuration/change management? »

Disse to spørsmålene bidro til å avdekke viktige punkt som burde inkluderes i prosjektet. Etter bruk av diagrammet ble det vedtatt at det skulle settes opp Update Management i Azure, delvis for å vise mulighetene i Azure, men også for å tilfredsstille en strategi som det andre spørsmålet etterspør.

På den annen side var en strategi for backup og disaster recovery allerede en implisitt del av prosjektet, ettersom dette er en av fordelene Azure tilbyr. Det ble etter gjennomgang av diagrammet derimot bestemt at backup og retention burde dokumenteres når det kommer til SQL-databaser, siden det er her kundekritiske data kommer til å ligge.

Ved ferdig utført prosjekt ser resultatet av SMART-diagrammet slik ut:

³³ (Korolis, 2020)



Figur 19 - SMART

Merk at ikke alle områder er like reelle for dette prosjektet. Det er særlig med tanke på drift etter prosjektets gjennomføring.

Figurer

Figur 1 - Verktøy brukt for oppsett av testmiljø	8
Figur 2 - Ressurser i testmiljøet	10
Figur 3 - Verktøy brukt under forarbeid.....	11
Figur 4 - Verktøy brukt for migreringen	11
Figur 5 - Ressurser under innledende vurdering og migrering	12
Figur 6 - Verktøy brukt etter migrering.....	12
Figur 7 - Tjenester etter migrering.....	13
Figur 8 - Overordnet prosess	17
Figur 9 - Prosess – klargjøring av testmiljø	18
Figur 10 - Prosess – oppdag og vurder miljøet	19
Figur 11 - Prosess – test-migrering og migrering.....	19
Figur 12 - Azure brukergrensesnitt	22
Figur 13 - Ressurser og ressursgrupper	23
Figur 16 - Opprettelse av VM.....	24
Figur 17 - Azure Runbooks	25
Figur 18 – Tags	25
Figur 19 - RDP.....	26
Figur 20 - SSH	26
Figur 21 - SMART.....	28

Tabeller

Tabell 1 - Begreper/forkortelser	5
Tabell 2 - Kostnadsestimat testmiljø.....	8
Tabell 3 - Verktøy brukt under forarbeid	11
Tabell 4 - Tjenester i Azure etter migrering	14
Tabell 5 - Fordeler med Azure - Ideelt miljø	16

Referanser

- Brasø, E. A., & Aalrust, E. (2020, Februar 18.). Analyse av brukerundersøkelse.
- Brasø, E. A., & Aalrust, E. (2020, Januar 15). Caseoppgave.
- Brasø, E. A., & Aalrust, E. (2020). *CleanupAutomation.ps1*. Hentet fra Bachelor-Sopra-Steria-2020: <https://github.com/Zypheros/Bachelor-Sopra-Steria-2020/blob/master/PowerShell/Azure/CleanupAutomation.ps1>
- Brasø, E. A., & Aalrust, E. (2020). *Driftsrapport*.
- Brasø, E. A., & Aalrust, E. (2020). *Forstudierapport*.
- Brasø, E. A., & Aalrust, E. (2020). *Sluttrapport*.
- Chirapurath, J. (2019, Desember 02). *Faster and cheaper: SQL on Azure continues to outshine AWS*. Hentet fra Microsoft Azure: <https://azure.microsoft.com/en-us/blog/faster-and-cheaper-sql-on-azure-continues-to-outshine-aws/>
- Korolis, T. (2020, Februar 2). *Azure Migration Live Event*. Hentet fra Best Practices for Windows Server and SQL server: https://vshow.on24.com/vshow/AMVE/exhibits/AMVE_OD
- Kotler, P., & Keller, K. L. (2016). *Markedsføringsledelse 4. utg.* Gyldendal.
- Microsoft. (2018, Oktober 18). *An Introduction to Azure Automation*. Hentet fra Microsoft Docs: <https://docs.microsoft.com/en-us/azure/automation/automation-intro>
- Microsoft. (2018, Oktober 18). *An Introduction to Azure Automation*. Hentet fra Microsoft Docs: <https://docs.microsoft.com/en-us/azure/automation/automation-intro>
- Microsoft. (2018, Oktober 3). *What is Azure Files?* Hentet fra Microsoft Docs: <https://docs.microsoft.com/en-us/azure/storage/files/storage-files-introduction>
- Microsoft. (2019, 12 29). *About Azure Migrate*. Hentet fra Microsoft Azure Documentation: <https://docs.microsoft.com/nb-no/azure/migrate/migrate-services-overview>
- Microsoft. (2019, November 19). *Azure Migrate Appliance*. Hentet fra Microsoft Azure Documentation: <https://docs.microsoft.com/en-us/azure/migrate/migrate-appliance>
- Microsoft. (2019, August 29). *Azure Resource Manager Overview*. Hentet fra Microsoft Azure Documentation: <https://docs.microsoft.com/en-us/azure/azure-resource-manager/management/overview>
- Microsoft. (2019, Juni 31). *What is Azure Active Directory?* Hentet fra Microsoft Docs: <https://docs.microsoft.com/en-us/azure/active-directory/fundamentals/active-directory-what-is>
- Microsoft. (2019, 11 21). *What is Azure Blueprints?* Hentet fra Microsoft Azure Documentation: <https://docs.microsoft.com/en-us/azure/governance/blueprints/overview>
- Microsoft. (2020, April 15). *About Azure Migrate*. Hentet fra Microsoft Docs: <https://docs.microsoft.com/en-us/azure/migrate/migrate-services-overview>
- Microsoft. (2020, April 19). *Azure Migrate support matrix*. Hentet fra Microsoft Docs: <https://docs.microsoft.com/en-us/azure/migrate/migrate-support-matrix>

- Microsoft. (2020). *Azure SQL Database Documentation*. Hentet fra Microsoft Docs: <https://docs.microsoft.com/en-us/azure/sql-database/>
- Microsoft. (2020, Mai 15). *Bli kjent med Azure*. Hentet fra Microsoft Azure: <https://azure.microsoft.com/nb-no/overview/#security>
- Microsoft. (2020, Februar 25.). *Introduction to Azure Data Lake Storage Gen2*. Hentet fra Microsoft Docs: <https://docs.microsoft.com/nb-no/azure/storage/blobs/data-lake-storage-introduction>
- Microsoft. (2020, April 15). *Migrate Hyper-V VMs to Azure*. Hentet fra Microsoft Docs: <https://docs.microsoft.com/en-us/azure/migrate/tutorial-migrate-hyper-v>
- Microsoft. (2020). *PowerShell Documentation*. Hentet fra Microsoft Docs: <https://docs.microsoft.com/en-us/powershell/>
- Microsoft. (2020, April 15). *Prepare for assessment and migration of Hyper-V VMs to Azure*. Hentet fra Microsoft Docs: <https://docs.microsoft.com/en-us/azure/migrate/tutorial-prepare-hyper-v>
- Microsoft. (2020, April). *Sammendrag av SLA for Azure-tjenestene*. Hentet fra Microsoft: <https://azure.microsoft.com/nb-no/support/legal/sla/summary/>
- Microsoft. (2020, April 06). *What are ARM templates?* Hentet fra Microsoft Docs: <https://docs.microsoft.com/en-us/azure/azure-resource-manager/templates/overview>
- Microsoft. (2020, Mars 30). *What is Azure Active Directory Domain Services?* Hentet fra Microsoft Docs: <https://docs.microsoft.com/nb-no/azure/active-directory-domain-services/overview>
- Microsoft. (2020, Mai 06). *What is Azure Blueprints?* Hentet fra Microsoft Docs: <https://docs.microsoft.com/en-us/azure/governance/blueprints/overview>
- Microsoft. (2020, Januar 27). *What is Windows Virtual Desktop?* Hentet fra Microsoft Docs: <https://docs.microsoft.com/nb-no/azure/virtual-desktop/virtual-desktop-fall-2019/tenant-setup-azure-active-directory>
- Microsoft Azure. (2019, 05 31). *What is Azure Database Migration Service?* Hentet fra Microsoft Azure Documentation: <https://docs.microsoft.com/en-us/azure/dms/dms-overview>
- Microsoft Azure. (2020). *Prikalkulator*. Hentet fra Microsoft Azure: <https://azure.microsoft.com/nb-no/pricing/calculator/>
- Microsoft Azure. (2020, 08 01). *Support matrix for Hyper-V assessment*. Hentet fra Microsoft Azure Documentation: <https://docs.microsoft.com/en-us/azure/migrate/migrate-support-matrix-hyper-v>
- Piriform. (2020, 02 18). *Speccy*. Hentet fra ccleaner.com: <https://www.ccleaner.com/speccy>
- Reinertsen, Ø., Hagberg, S., & Lees, R. (2013, Desember 3). *Teknisk gjeld - Tør vi å snakke om det?*

Vedlegg

Følgende frittstående filer er vedlagt:

- *Skript.pdf* – Inneholder alle skript som brukes i sammenheng med prosjektet.
- *Hele Prosesløpet.png* – Figur beskriver hele prosesløpet.
- *Support Matrix for Hyper-V assessment.pdf* – Inneholder kravene som stilles til en Hyper-V server før migrering. Hentet fra Microsoft. Også kalt kravspesifikasjon.
- *Win16serv.pdf* – Viser detaljert oversikt over enheten som blir brukt til det fysiske miljøet. Importert med verktøyet Speccy³⁴
- *Exportedestimate.xlsx*

I tillegg er hele brukerundersøkelsen inkludert. Brukerundersøkelsen inneholder:

- *Analyse av brukerundersøkelse.pdf*
- *Brukerundersøkelse Skytjenester (Svar).xlsx*
- *Svar Nr 8, 14, 22, 31 og 45* som .pdf dokument

³⁴ (Piriform, 2020)



DRIFTSRAPPORT

En automatisert reise fra datarom til sky

Emil Antoni Brasø, Endre Aalrust

Revisjonslogg

DATO	VERSJON	BESKRIVELSE	FORFATTER
25.02.20	1.0	Opprettelse av dokument	Endre, Emil

Forord til driftsrapporten

Dette dokumentet er utarbeidet på vegne av den fiktive bedriften Salmo AS. Casen, eller arbeidsoppgaven, går ut på å migrere bedriftens systemer fra et on-premises miljø til et skybasert miljø. Dette prosjektet innebærer å flytte bedriftens eksisterende systemer til en skybasert plattform. Tjenestene skal derved videreføres i Azure.

Hensikten med dette dokumentet er å ta detaljert for seg hvordan prosjektet er gjennomført og hvordan løsningen ser ut helhetlig etter prosjektet er avsluttet. Dokumentet tar for seg oppstart i on-premises miljøet, installasjon av pilotprosjektet og i tillegg kommer beskrivelser av ulike installasjoner, skript og andre verktøy som blir brukt i sammenheng med migreringsprosessen. Dokumentet vil også ta for seg verktøy i Azure som blir brukt i etterkant av migreringen for å sette opp miljøet og gjøre det brukervennlig. Diverse definisjoner og begreper vil forklares, deriblant ved hjelp av definisjonslisten i starten av dokumentet.

Selve prosjektgruppen består av Emil Antoni Brasø og Endre Aalrust. Emil og Endre har lenge jobbet sammen i gruppe gjennom studiet, og til gode resultat. Veiledere i bedriften Sopra Steria er Kristian Aamodt og Rune Haugen. Fra NTNU så er Jostein Lund veileder. For prosjektgruppen står et læringsønske sentralt, og er på dette punktet i prosjektet allerede et mål prosjektgruppen ser som oppnådd.

Innholdsfortegnelse - Driftsrapport

Revisjonslogg.....	1
Forord.....	1
Innholdsfortegnelse - Driftsrapport.....	2
Forkortelser og definisjoner.....	5
Kode	6
1 Om dette dokumentet	7
1.1 Hensikten med dokumentet	7
1.2 Avgrensning	7
1.3 Dokumentets oppbygging.....	8
2 Prosjektets faser og overordnet struktur	9
2.1 Faser.....	9
2.2 Overordnet struktur på prosjektet	10
2.2.1 Hvordan tjenestene blir videreført i Azure	10
3 Oppstart	11
3.1 On-premises miljø.....	11
3.2 Miljø i Azure.....	13
3.3 Fremgangsmåte – On-premises miljø	14
3.3.1 OpenVPNserver.....	14
3.3.2 Oppsett ADDS – Domenekontroller	15
3.3.3 Filserver.....	17
3.3.4 SQL-Server.....	20
3.3.5 Enkel webserver.....	23
4 Utførelse	24
4.1 Designrapporten	24
4.2 Forhåndsarbeid før migrering	25
4.2.1 Azure SQL Database	26
4.2.2 Figur on-premises miljø før utførelsen	26
4.2.3 Figur Azure-miljø etter utførelse	26
4.2.4 Laste ned og klargjøre Migrate Appliance	27
4.2.5 Klargjøre Hyper-V serveren med et skript	28
4.2.6 Key Vault	29
4.2.7 Konfigurasjon VM-er (Hyper-V)	31
4.2.8 Opprette ressursgruppe i Azure.....	33
4.3 Main.json	34
4.3.1 Oppbygging av Main.json.....	34

4.3.2	Utrulling og sporing av main.json	36
4.4	App service i Azure.....	37
4.5	Storage Account i Azure.....	39
4.6	Azure AD Domain Services.....	40
4.6.1	Manuelt oppsett etter opprettelse av ressurser	40
4.6.2	Endre til kerberos autentisering	42
4.7	Discovery og Assessment av Virtuelle Maskiner	43
4.8	Registrer Hyper-V host (Azure Migrate)	47
4.9	Replication	50
4.10	Test migration	54
4.11	Migrering fra on-prem til azure	57
4.11.1	Virtuelle Maskiner.....	57
4.11.2	SQL-Databaser.....	59
4.11.3	Migrere filserver over til Azure	67
4.11.4	Potensielle feil og løsninger	72
4.12	Gjør klart et VM-image	74
4.13	WVD	76
4.13.1	Avhengigheter	78
4.13.2	Sett opp rettigheter og prinsipaler	78
4.13.3	Lag sesjonsverter med ARM og egendefinerte VM-images.....	82
4.13.4	Sette app groups	86
5	Brukerguide for verktøy og funksjoner i Azure.....	87
5.1	Azure Backup	87
5.1.1	Sette opp Recovery Services Vault.....	87
5.1.2	Sette opp sikkerhetskopiering av VM	88
5.1.3	Konfigurerere backup-retention på SQL-servere.....	89
5.2	Azure Runbook.....	90
5.2.1	Skript	93
5.2.2	Automatisk opprensing i Azure-miljø	94
5.3	Log Analytics Workspace	95
5.3.1	Log Analytics datakonfigurasjon	97
5.4	Azure Update Management.....	99
5.4.1	Konfigurer Update Manager	100
5.4.2	Legge til maskiner i Update Manager	102
5.4.3	Planlegging og utrulling av oppdatering	103
5.5	Azure Active Directory Access Panel.....	105

5.6	Windows Virtual Desktop Brukerguide.....	108
5.6.1	WVD web-client	108
5.6.2	WVD lokal app.....	110
6	Problemer/bugs	111
6.1	Problem ved opprettelse av AADDS	111
6.1.1	Recreate a missing service principal.....	112
6.1.2	Re-register the Microsoft AAD namespace.....	112
6.2	Tapt tilkobling til Hyper-V serveren	113
6.3	Feilmelding som følge av tidligere migrering.....	114
6.4	Registrert med annet recovery services vault	115
7	Utfaset funksjonalitet.	116
7.1	Automatisering av Windows 10 installasjon.....	116
7.1.1	Klargjøring av ISO-fil.....	117
8	Referanser.....	124
9	Figurer	121
10	Tabelloversikt.....	121

Forkortelser og definisjoner

Begreper / Forkortelser	Definisjon
AD	Active Directory - Microsofts katalogtjeneste for å håndtere brukere, ressurser og brukerrettigheter m.m.
Server	En dedikert maskin som tilbyr en eller flere tjenester til andre datamaskin over et nettverk
VM	Virtuell Maskin
VHD	Virtuell Harddisk som kan knyttes til en VM (Hyper-V)
DC	Domenekontroller
AADDs	Azure Active Directory Domain Services. Lar oss videreføre domenetjenester fra tradisjonell AD i Azure.
AAD	Azure Active Directory - Active Directory i Azure. Tar hånd om brukerautentisering og rettigheter.
LDAP	Protokoll som brukes til oppslag i en katalogtjeneste på en server/server
On-premises	Infrastruktur som er «plassert» hos bedriften
IaC – Infrastructure as code	Prinsipp som handler om å kunne deklare og rulle ut infrastruktur ved hjelp av kode og automasjon.
SQL	Structured Query Language – Brukes til å kjøre spørringer opp mot databaser.
MySQL	En av de første open-source databasene med SQL som kom på markedet. I prosjektet er det opprettet et eget skript for å migrere dette.
MSSQL	Microsoft Server SQL – Full støtte for Azure Migrate, og mer kostnadseffektivt enn MySQL i Azure.
Nested Virtualization	«Virtualisering ved hjelp av Virtualisering» Det opprettes et virtualisert miljø på en allerede virtualisert server.
ARM	Azure Resource Manager
Template	En “mal” for bruk opp mot ARM. Ruller ut ressurser ved hjelp av IaC.
RDP	Remote Desktop – Brukes til å koble seg til en ekstern maskin.
SSH	Secure Shell – Brukes til å koble seg til kommandolinjen på en ekstern maskin
CMD/Ledetekst	Kommandotolker i Windows
Azure	En gruppe skytjenester som benyttes til å organisere, utvikle, administrere og distribuere programmer.
Maskinlæring/machine learning/ML	Maskinlæring er en spesialisering innenfor kunstig intelligens hvor man bruker statistiske metoder for å la datamaskiner finne mønstre i store datamengder.

Hyper-V	En tjeneste som benyttes for å lage virtuelle maskiner
Automasjon	Prosesen hvor en eller flere arbeidsoppgaver utføres gjennom for eksempel et skript og ikke manuelt.
SLA (Service Level Agreement)	Avtale som beskriver kvalitet, tilgjengelighet og ansvar mellom kunde og tjenesteleverandør.
Legacy-System	En gammel metode, teknologi, datasystem eller applikasjon.
Port-forward	Port forwarding er en teknikk som brukes til å videresende forespørsler på en eller flere IP-porter til en IP-adresse på innsiden av et nettverk.

Tabell 1 - Begreper/forkortelser

Kode

I løpet av dette dokumentet vil det bli gjennomgått mye kode. For å gjøre lesbarheten bedre er det tatt i bruk fargekoding for å beskrive hva slags kode som benyttes. Fargekodingen beskrives under.

Ledetekst/CMD er benyttet hvis fargekodingen er svart/hvitt:

```
C:\Users\Administrator>fsutil file createnew c:\Testdata\100gb.test 10737418200
File c:\Testdata\100gb.test is created
```

Figur 1 – Ledetekst/CMD

PowerShell er benyttet hvis fargekodingen er blå/hvit/gul:

```
CertUtil -HashFile `
C:\Users\Administrators\Desktop\MicrosoftAzureMigrate-Hyper-V.ps1
SHA256
```

Figur 2 – PowerShell

JSON er benyttet hvis fargekodingen er svart/oransje/blå:

```
{
  "administratorLoginPassword": {
    "type": "securestring"
  },
  "location": "f
```

Figur 3 – JSON

Det kan forekomme avvik fra denne fargekodingen, men de fleste plasser er det beskrevet hvilket kodespråk som er benyttet. I tillegg vil de fleste filer ha en filending som beskriver kodespråket.

- Filer som slutter med `.ps1` vil være PowerShell-skript
- Filer som slutter med `.json` vil være JSON filer
- Filer som slutter med `.exe` vil være executables, og brukes ofte gjennom CMD.

1 Om dette dokumentet

1.1 Hensikten med dokumentet

Denne driftsrapporten er skrevet i forbindelse med en bacheloroppgave utarbeidet av Emil Antoni Brasø og Endre Aalrust, sammen med bedriften Sopra Steria.

Hensikten med driftsrapporten er å detaljert dokumentere hvordan løsningen ble iverksatt, og hvilke problemer som oppsto underveis. Denne fasen av prosjektet, utførelse, og derav driftsrapporten, tar utgangspunkt i den tidligere utarbeidede designrapporten¹. Designrapporten bygger i seg selv på forstudiet², som står for mesteparten av litteraturen og begrunnelsene lagt inn i planleggingen av prosjektet.

I dette dokumentet blir noen nye momenter tatt opp, da er det med tanke på problemer eller løsninger som ble oppdaget først på dette punktet i prosjektet. Det er greit å påpeke at det derimot ikke var nødvendig med noen radikale endringer på løsningen presentert i designrapporten. Endringer er oppsummert i sluttrapporten.

1.2 Avgrensning

Innledningsvis blir den overordnede strukturen i prosjektet forklart. Deretter blir pilotprosjektets overordnede deler – Oppstart, Utførelse og ferdigstilling. Hvordan bedriftstjenestene blir videreført i Azure blir forklart, og er et av hovedpunktene med dokumentet. Først blir dette tatt opp fra et overordnet synspunkt for å få oversikt over hele prosessen, men hver enkelt delprosess blir også forklart med en detaljert fremgangsmåte. Hva de ulike tjenestene går ut på eller hvilken rolle de utfyller, er for det meste forklart tidligere som en del av designrapporten.

Denne rapporten tar for seg den ene løsningen som ble fattet gjennom designrapporten. Denne løsningen kan konkretiseres med følgende egenskaper:

- Mest mulig dyp integrasjon
- Mest mulig grad av automasjon og IaC (Infrastructure-as-code) innenfor prosjektets rammer
- Sluttkundes behov og ønsker i fokus.

En del resultatmål ble definert i forstudierapporten, og innarbeidet i designrapporten. Følgende resultat skal foreligge:

- En fremgangsmåte som beskriver selve prosessen med å migrere dagens løsning fra on-premises til en hel eller delvis skybasert løsning.
- En overordnet analyse av det økonomiske aspektet ved å flytte bedriftens IT-systemer til en skybasert løsning.
- Ett system som utnytter skybasert teknologi.
- En analyse av typen løsning; hel eller delvis og hva som vil gagne bedriften i det lange løp.
- En forståelse av kunders behov og OKV (Opplevd Kunde Verdi)
- Migreringsprosessen skal påvirke den daglige driften minst mulig.

Konkret skal dette dokumentet beskrive den endelige framgangsmåten og implementasjonen for:

- Ett fungerende test-system
- En effektivisert framgangsmåte gjennom automatisering og IaC

¹ (Brasø & Aalrust, Designrapport, 2020)

² (Brasø & Aalrust, Forstudierapport, 2020)

1.3 Dokumentets oppbygging

Grunnet prosjektets omfang og alle de mindre delsystemene som inkluderes i rapporten er dette et ganske stort og komplekst dokument. For å sikre en god forståelse av dokumentet blir oppbyggingen forklart her, i tillegg til den overordnede innholdsfortegnelsen tidligere i dokumentet.

Denne driftsrapporten er skrevet med tanke på at det skal være mulig å sette opp alle delsystemer som et helhetlig system, men samtidig også at de fleste delsystemene skal kunne trekkes ut og brukes hver for seg.

Ett eksempel er kapittel 4.3.1. Dette delkapitlet handler om hvordan en hoved-template er bygd opp etter prinsippet for linked-templates, samt at den forklarer fordeler med dette. Bruken opp mot prosjektet er hovedpoenget, men det er ingenting i veien med at dette kan trekkes ut og brukes i forbindelse med et annet prosjekt, bare med mindre endringer.

For en mer konkret fremgangsmåte, er overføringen av filer fra on-premises filserver som vist i 4.11.3 et mer reelt eksempel. Beskrivelsen her er tilpasset bruk opp mot dette prosjektet, men kan enkelt trekkes ut for bruk andre steder.

Når det kommer til skript og annen kode, så blir det ikke forklart utdypende her. For dette formålet foreligger vedlegget *Skript*⁴, som er tiltenkt en mer IT-teknisk leser. Det dokumentet, og de forskjellige skriptene i det, blir referert til eller nevnt der det er hensiktsmessig i denne rapporten.

Rapporten følger en tilnærmet kronologisk rekkefølge, hvor det som ble utført først i prosjektet er skrevet først. Mange av delene er uavhengige av hverandre, og det er for det meste ikke nødvendig å gjennomføre de sekvensielt. Dersom slike avhengigheter forekommer, blir det beskrevet der det er relevant.

Videre skal denne rapporten ta for seg detaljert hva som ble utført på teknisk side. De forskjellige verktøyene og teknologiene blir forklart der det er hensiktsmessig, men det blir for eksempel ikke tatt ett dypdykk i hvert enkelt skript eller template.

⁴ (Brasø & Aalrust, Skript, 2020)

2 Prosjektets faser og overordnet struktur

2.1 Faser

Overordnet ble bachelorprosjektet delt inn i tre deler. Oppstart, Utførelse og Ferdigstilling. Det meste av arbeidet ligger i utførelsen, og da især i form av dokumentasjon for driftsrapporten. Under oppstartfasen ble det utarbeidet en prosjektplan som etterfølgende skulle kontrolleres hver uke, og revideres ved behov.

Fase 1 – Oppstart

I oppstarten inngikk følgende deloppgaver:

1. Avtale med bedrift
2. Generell planlegging og administrative oppgaver
3. Utarbeidelse av litteraturliste
4. Forstudierapport og revidering av den
5. Klargjøring av testmiljøet.

Fase 2 – Utførelse

For utførelse inngikk følgende oppgaver:

1. Utarbeide Designrapport
2. Utarbeide driftsrapport
3. Sette opp testmiljøet.
 - a. Teste alle komponenter
 - b. Verifiser krav
 - c. Testmigrering
4. Fullstendig migrering av pilotprosjekt.

Fase 3 – Ferdigstilling

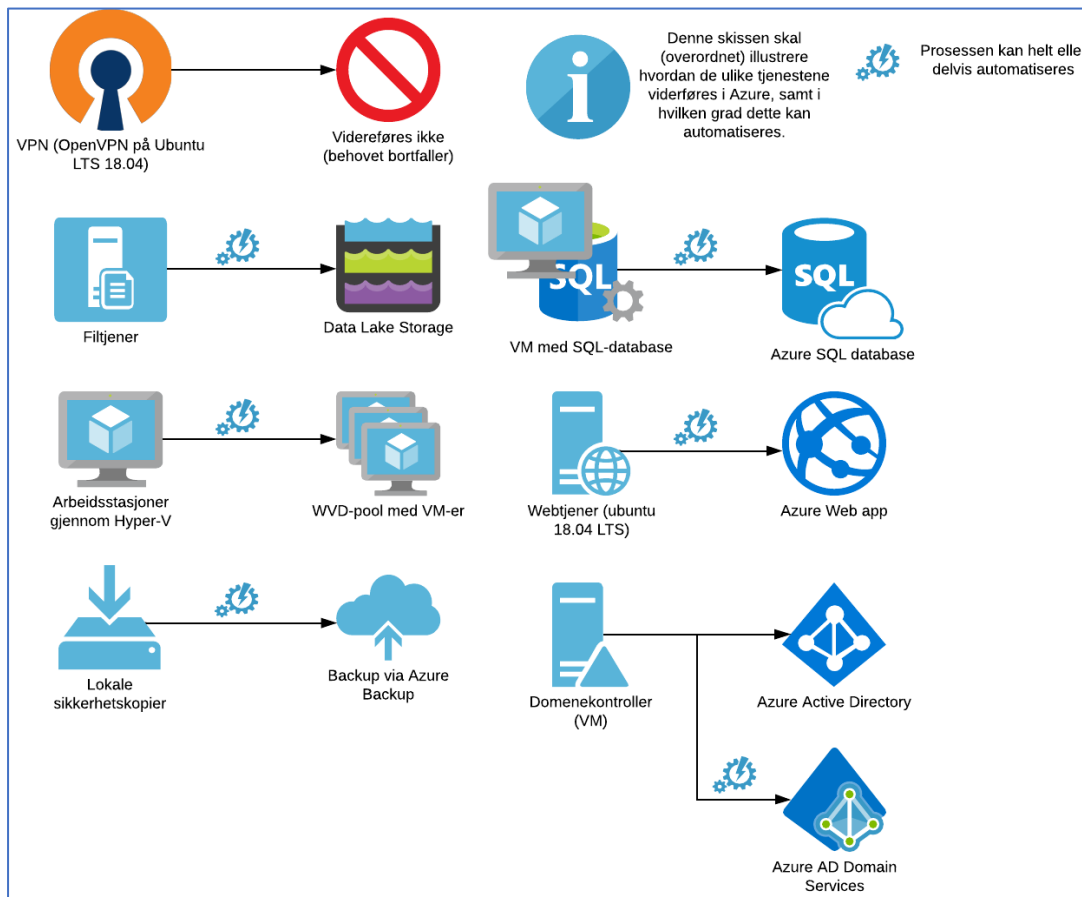
Prosjektets siste del innebærer mange administrative og formelle oppgaver. Inkluderer å:

1. Utarbeide Sluttrapport
2. Levere oppgaven
3. Presentasjon for oppgaven

2.2 Overordnet struktur på prosjektet

2.2.1 Hvordan tjenestene blir videreført i Azure

Følgende figur illustrerer hvordan tjenestene fra det opprinnelig on-premises miljøet videreføres til Azure, i tillegg til hvilken grad av automasjon som kan oppnås på selve prosessen med å føre det over.



Figur 4 – Hvordan tjenester videreføres

Tjeneste	Kommentar
OpenVPN server	Videreføres ikke, siden behovet bortfaller. Autentisering gjennom Azure AD vil i dette prosjektet ta over for VPN når det gjelder sikkerhet og autentisering.
Filserver	Videreføres til Storage Account med fileshares. Prosessen kan automatiseres
Arbeidsstasjoner	Videreføres til Windows Virtual Desktop. Kan automatiseres
Lokale sikkerhetskopier	Videreføres som en tjeneste innebygd i Azure, Azure Backup
SQL-server på en VM	Videreføres som Azure SQL database (både MSSQL og MySQL vises), kan automatiseres.
Webserver med Ubuntu 18.04	Videreføres som Azure Web app, kan automatiseres
Domenekontroller	Bortfaller, men Azure AD og Azure AD Domain Services tar over oppgavene til denne.

3 Oppstart

I oppstartfasen var en rekke administrative oppgaver nødvendige, hovedsakelig gikk dette ut på å avholde et oppstartsmøte, samt å avklare prosjektets omfang. Utarbeidelse av forstudierapporten og designrapporten ble gjort under, og som del av, denne fasen. Testmiljøet ble satt opp her, så derfor nevnes det i forbindelse med driftsrapporten.

Testmiljøet som ble satt opp består av to deler:

- En on-premises server som fungerer som Hyper-V server.
- Ett miljø i Azure

3.1 On-premises miljø

Jostein Lund disponerer fysisk server. Den fysiske serveren står hjemme hos Emil og Endre, i Trondheim. Den har 32 GB RAM og en Intel Core i7 prosessor på 3.6Ghz. en utfyllende oversikt er vedlagt⁵. Følgende er et utsnitt fra det vedlegget:

```
Speccy report for WIN-PR3H1PGK4A8, [18.02.2020 08.22.28]
Summary
Operating System
Windows Server 2016 Datacenter 64-bit
CPU
Intel Core i7 4790 @ 3.60GHz 28 °C
Haswell 22nm Technology
RAM
32,0GB Dual-Channel DDR3 @ 794MHz (10-10-10-30)
Motherboard
ASUSTeK COMPUTER INC. B85M-G (SOCKET 1150) 28 °C
```

Figur 5 - Server-spesifikasjoner skapt med programvaren Speccy

Vedlegget beskriver utfyllende om selve serveren, og kan være spesielt nyttig når det skal utføres feilsøking. Det beskrives kort her hvordan serveren settes opp.

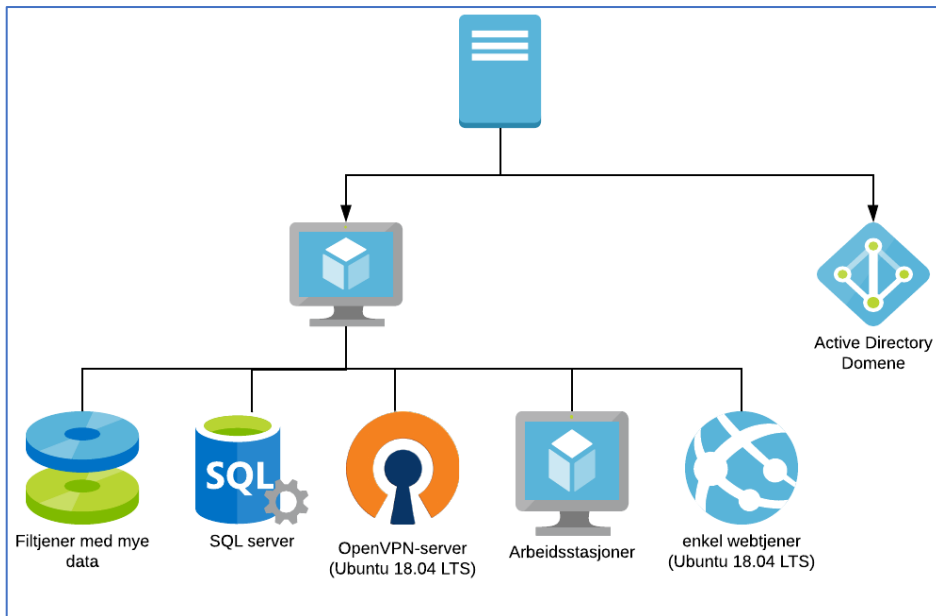
Operativsystemet som ble installert var Windows Server 2016, deretter ble tjenesten Hyper-V lagt til gjennom Server Manager. Hyper-V er tradisjonelt brukt for å drifte VMer (Virtuelle maskiner) på on-premises miljø, og fungerer da som bedriften sitt startpunkt før arbeidet med å migrere over til sky starter. I utgangspunktet er dette Salmo sitt reelle miljø per i dag.

I tillegg til Hyper-V ble det satt opp OpenVPN og RDP (Remote Desktop Services). Dette er for å tillate tilkobling fra eksternt nettverk. RDP for å få tilgang til Hyper-V serveren fra utsiden, og OpenVPN for å få tilgang til det interne nettverket som den ligger på.

Dette er funksjonalitet man ville hatt i en reell setting, men er også nyttige for prosjektgruppens arbeid. Oppsettet av VPN-en beskrives i kapittel 3.3.1⁶

⁵ (Brasø & Aalrust, win16serv.pdf, 2020)

⁶ Oppsett av OpenVPNserver



Figur 6 - Oppsett av Hyper-V server

For at testmiljøet skulle bli mest mulig likt en generell bedrifts miljø ble Hyper-V serveren satt opp med:

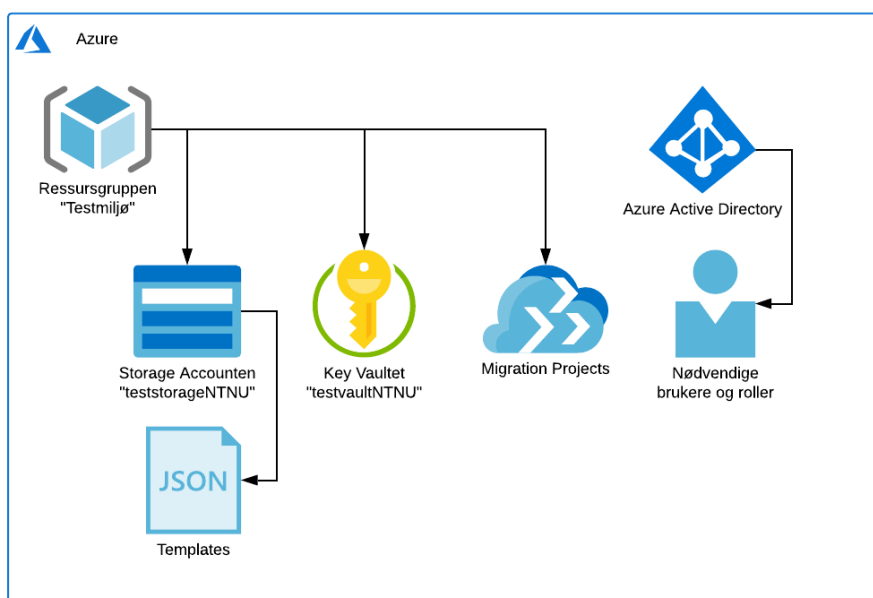
- Ett Active Directory domene.
- En VM med SQL server (både MSSQL og MySQL, for å vise begge alternativene)
- En OpenVPN server
- Noen arbeidsstasjoner

3.2 Miljø i Azure

Selve miljøet i Azure disponeres av Sopra Steria. Dette miljøet er satt opp med et kostnadstak på 4000 NOK, ettersom det var dette som var den øvre grensen prosjektgruppen anslo under arbeidet med designrapporten.

Ved starten av utførelsen av prosjektet var miljøet nesten tomt. Følgende ressurser som var nødvendige for migreringen ble derimot lagt til i forkant:

- En Storage Account for å oppbevare ARM (Azure Resource Manager) templates og andre ressurser som må være tilgjengelig på nett.
- Et Key Vault for å bruke og oppbevare hemmelige verdier på et sikkert vis.
- Et domene Azure Active Directory. Ble brukt til å lage brukerne og rollene nødvendig for migrering. Emil og Endre tok i bruk egne brukere for å kunne skille mellom utrullinger i Azure OSV.
- Et Azure Migrate Project, for å kjøre oppgavene tilknyttet selve migreringen.



Figur 7 - Ressursgruppen "testmiljø"

Det er verdt å presisere at ressursgruppen «TestMiljø» ble brukt til administrative oppgaver knyttet til prosjektet. For testmigreringene ble det lagd egne dedikerte ressursgrupper som deretter befolkes med ressurser, som beskrevet i denne rapporten. Ressursgruppen holdte ressursene nødvendig for å kunne utføre disse oppgavene.

3.3 Fremgangsmåte – On-premises miljø

On-premises miljøet blir her satt opp på en enkel måte, men inneholder alle komponenter som er nødvendige for å få et fungerende system. Hensikten med dette systemet er ikke å opprette et utfyllende system som en bedrift kunne ha hatt i produksjon, men illustrerer heller hvordan de ulike delene kan migreres over til Azure.

3.3.1 Oppsett av OpenVPNserver

Noe av det første som ble gjort var å sette opp VPN, slik at prosjektgruppen kunne kommunisere med miljøet on-premises utenfra. Den tenkte bedriften ville svært sannsynlig ha benyttet seg av VPN for hjemmekontor og lignende.

Dette ble løst med programvaren OpenVPN, satt opp på en VM. For å gjøre det mest mulig praktisk utarbeidet prosjektgruppen et skript i bash, *OpenVPNsetup.sh*⁷. Dette skriptet automatiserer store deler av oppsettet, slik at en tilsvarende løsning blir tilgjengelig på kortest mulig tid.

Operativsystemet i dette tilfellet var Ubuntu 18.04.

VPN blir ikke videreført i Azure, i stedet blir Azure Active Directory benyttet for å autentisere tilgang til tjenester. Siden tjenestene blir flyttet opp på internett uansett, trenger denne bedriften ikke lenger en vei inn på et on-premises nettverk.

Hele *OpenVPNsetup.sh* er tilgjengelig på GitHub repositoret, men et eksempel på kommandoer som blir brukt er kommandoen sed:

```
sed -i "s/cert openvpnserver.crt/cert ${serverName}.crt/" /etc/openvpn/server.conf
```

Dette brukes til å endre en linje i filen */etc/openvpn/server.conf*. I all hovedsak vil skriptet laste ned alle nødvendige filer, og vil deretter endre noe innhold i disse filene, avhengig av bruker input.

For å bruke tjenestene tilbudt av OpenVPN benyttes ytterligere to skript, *make_config.sh*⁸ og *client_make.sh*⁹

- **client_make.sh** tar i bruk navnet som skal settes på bruker-konfigurasjonen, signerer sertifikatene og kaller skriptet *make_config.sh*
- **make_config.sh** brukes deretter for å sette inn korrekte filbaner og oppretter til slutt konfigurasjonen som brukes for sluttbruker.

Innehaver av skriptet *make_config.sh* er DigitalOcean, men de øvrige skriptene har prosjektgruppen utarbeidet.

⁷ (Brasø & Aalrust, *OpenVPNsetup.sh*, 2020)

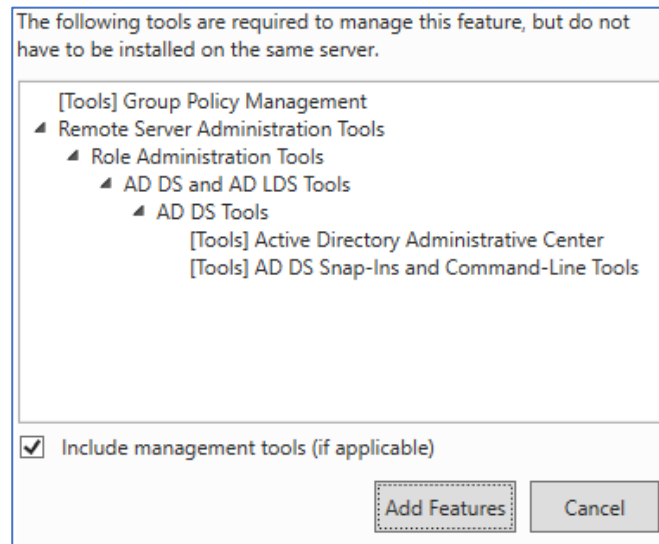
⁸ (DigitalOcean, 2018)

⁹ (Brasø & Aalrust, *client_make.sh*, 2020)

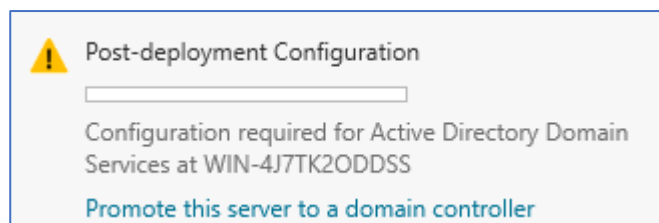
3.3.2 Oppsett av ADDS – Domenekontroller

Domenekontrolleren ble konfigurert på vanlig måte med ADDS og ett nytt lokalt domene: «Salmo.local». Alle maskiner i domenet er satt med statisk IP-adresse for å kunne nå hver enkelt maskin via RPD gjennom VPN-løsningen som blir satt opp. Noen enkle brukerkontoer og grupper opprettes også.

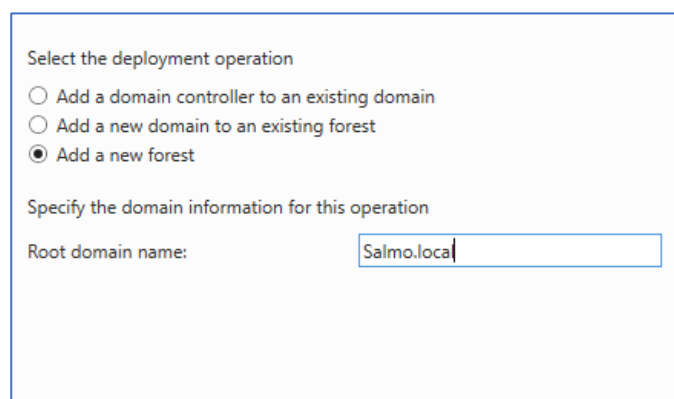
Installerer ADDS gjennom *roles and features i server manager*



Starter konfigurasjon



Velger *Add a new forest* for å opprette et nytt domene til testmiljøet.



Velger at domenekontrolleren skal fungere som en DNS-server, dette gjøres slik at klientene i domenet skal finne domenekontrolleren.

Domenekontrolleren settes også som *global katalog*, dette gjør at et objekt kan få informasjon fra domenekontrolleren uavhengig om det er medlem i domenet eller ikke.

Til slutt settes det et gjenopprettingspassord

Select functional level of the new forest and root domain

Forest functional level:

Domain functional level:

Specify domain controller capabilities

Domain Name System (DNS) server

Global Catalog (GC)

Read only domain controller (RODC)

Type the Directory Services Restore Mode (DSRM) password

Password:

Confirm password:

Her kan lagringsplassen til ADDS spesifiseres. Vanligvis ville disse mappene bli satt på en annen server/disk, men i dette tilfellet vil det ikke ha noe å si for testmiljøet og derfor gjøres det ikke noe med det.

Specify the location of the AD DS database, log files, and SYSVOL

Database folder:

Log files folder:

SYSVOL folder:

Sjekken gikk bra og ADDS kan installeres

✔ All prerequisite checks passed successfully. Click 'Install' to begin installation. [Show more](#) ✕

Deployment Configuration

Domain Controller Options

DNS Options

Additional Options

Paths

Review Options

Prerequisites Check

Prerequisites need to be validated before Active Directory Domain Services is installed on this computer

[Rerun prerequisites check](#)

View results

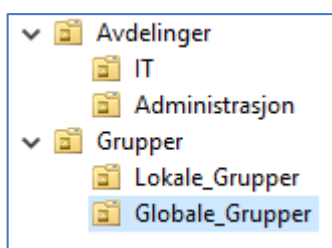
⚠ Windows Server 2016 domain controllers have a default for the security setting named "Allow cryptography algorithms compatible with Windows NT 4.0" that prevents weaker cryptography algorithms when establishing security channel sessions.

Tabell 2 - Oppsett av ADDS

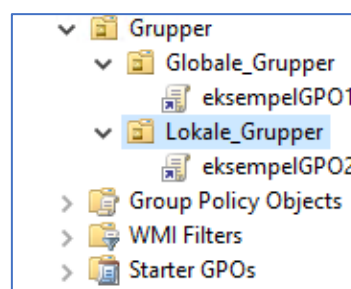
3.3.2.1 Oppsett i Active Directory

For testmiljøets skyld opprettes det noen brukere, organizational units og grupper. Det nye systemet vil være en mulighet til å omstrukturere og rydde opp i miljøet, og på grunn av dette vil ikke disse komponentene bli migrert, men heller bli gjenskapt i Azure.

For å administrere brukere finnes det fire organizational units IT, Administrasjon, Teknikere og arbeidere. For å administrere rettigheter til disse brukerne er det blitt opprettet globale og lokale grupper. Det har blitt satt opp noen vilkårlige Group Policies på disse gruppene for testmiljøets skyld, men disse har ingen funksjon. GPO'ene er satt på gruppene og ikke direkte på brukerne for ordens skyld.



Figur 8 - Active Directory struktur



Figur 9 - GPO

Tabell 3 - Active Directory

3.3.3 Filserver

Dette er maskinen som tar vare på alle filer. Maskinen fylles med data (ca. 130GB) slik at migrering av data blir simulert. For å simulere datamengden opprettes noen test-filer ved hjelp av en kommandolinjeverktøyet *fsutil*.

```
C:\Users\Administrator>fsutil file createnew c:\Testdata\100gb.test 10737418200
File c:\Testdata\100gb.test is created
```

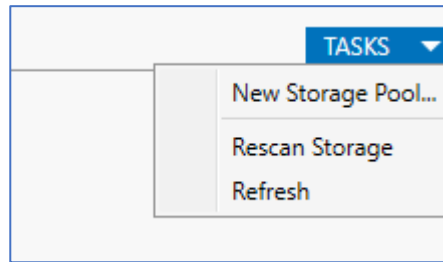
Figur 10 - Testfil-kommando

10gb.test	3/5/2020 3:07 AM	TEST File
10gb2.test	3/5/2020 3:07 AM	TEST File
10gb3.test	3/5/2020 3:07 AM	TEST File
100gb2.test	3/5/2020 3:07 AM	TEST File

Figur 11 - Testfiler

For at dette skal bli en fungerende filserver som ansatte kan benytte seg av må det legges til noen roller i server manager.

I server manager, naviger til *File and Storage Services – Storage Pools*. Under *Tasks* velg *New Storage Pool*



Skriver inn informasjon om *Storage Pool*

Name:

Description:

Select the group of available disks (also known as a prime...

Managed by	Available to
WIN-N5R9EOJ01FF	WIN-N5R9EOJ01FF

Legger til en ekstra disk som en *failover*

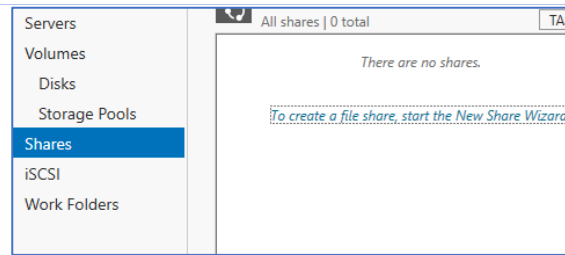
Physical disks:

<input checked="" type="checkbox"/>	Slot	Name	Capacity	Bus	RPM
<input checked="" type="checkbox"/>		Virtual HD (WI...	60.0 GB	ATA	

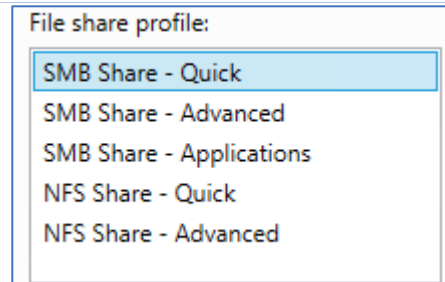
Oppretter *storage pool*

STORAGE POOL LOCATION	
Server:	WIN-N5R9EOJ01FF
Cluster role:	Not Clustered
Storage subsystem:	Windows Storage
STORAGE POOL PROPERTIES	
Name:	filtjener
Description:	Samlet storage pool for Salmo AS
Capacity:	60.0 GB
PHYSICAL DISKS	
Virtual HD (WIN-N5R9EOJ01FF)	Automatic

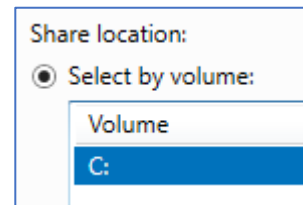
Velg shares og klikk på *To create a file share, start the New Share Wizard*



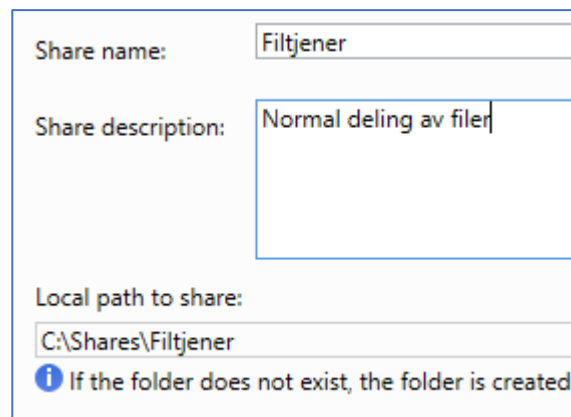
Velger her *SMB Share – Quick*



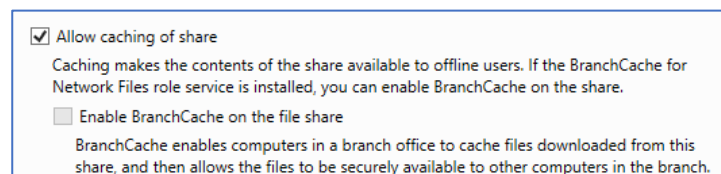
Share Location settes på c-disken



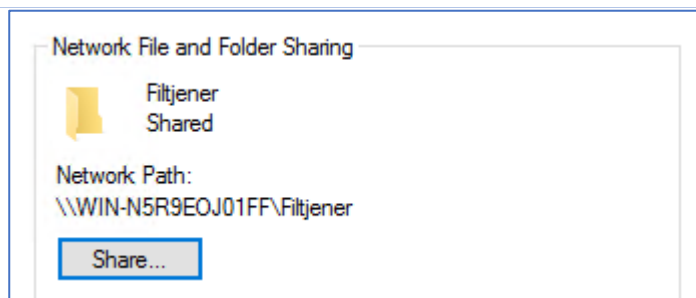
Fyller inn nødvendig informasjon



Under *Other Settings* og *Permissions* kan alt stå som det er og Wizarden kan fullføres



Sjekk om mappen er delt ved å høyre-klikke på mappen og velge "Properties"



Tabell 4 - SMB-share

3.3.4 SQL-Server

En SQL-server er en server som tilbyr databasetjenester til andre dataprogrammer eller til datamaskiner som definert av klient-servermodellen.

3.3.4.1 Installasjon av SQL-server

SQL-serveren ble satt opp som en VM (Virtuell Maskin) på operativsystemet Windows server 2016. Installasjonen av MySQL ble utført ved hjelp av verktøyet chocolatey¹⁰, som fungerer fint til å foreta installasjoner som ikke behøver direkte bruker-input, på lignende vis ble powershellskriptet *installMSSQL.ps1*¹¹ brukt for MSSQL sin del.

Det ble besluttet å fremvise prosessene rundt både MySQL og MSSQL, dette er fordi det er to vanlige former for databaser, men også fordi det bidrar til å fremvise mer av funksjonaliteten i Azure. Uansett er det viktig å fremheve noe av forskjellen mellom MySQL og MSSQL. MySQL er en av de første databasene med SQL som kom på markedet, og er utviklet av Oracle. MSSQL er Microsofts egen løsning på database-server med SQL, og har bedre støtte for Azure Migrate. Derfor blir det en annerledes prosess for disse to alternativene.

Først drøftes MySQL, deretter MSSQL.

MSSQL er etter prosjektgruppens mening klart å foretrekke, siden den kan overføres enklere til Azure. Videre er det mulig å føre MySQL videre med verktøyet «Azure databases for MySQL», men det er å foretrekke å videreføre MySQL som en Azure SQL Database i stedet, da det er bedre støtte for det i Azure, og det er beviselig billigere i drift.¹²

¹⁰ (Chocolatey Software, Inc., 2020)

¹¹ (Brady, 2018)

¹² (Microsoft, Choose the right deployment option in Azure SQL, 2019)

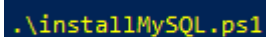
3.3.4.1.1 MySQL

Kjører skriptet *installMySQL.ps1*¹³

Skriptet gjør hovedsakelig to ting:

1. Det laster ned pakke-verktøyet chocolatey
2. Installerer mysql med chocolatey

Etter installasjon vil verktøyet ligge på
C:\tools\mysql



```
.\installMySQL.ps1
```

Starter MySQL med kommandoen

```
cd C:\tools\mysql-8.0.19-win64  
mysqld --initialize --console
```

Oppstart går raskt og det blir generert et passord som deretter skal brukes til å logge seg på, med følgende kommando

```
mysql.exe -u root -p
```

Dermed er MySQL klart til bruk og kommandolinjen er endret til *mysql*

```
mysql>
```

Til slutt installeres MySQL workbench for å forenkle administrasjon

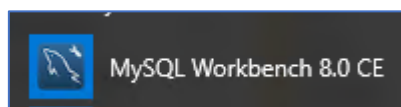
```
choco install mysql.workbench -y
```

Tabell 5 - Installasjon MySQL

3.3.4.1.1.1 Test-database for MySQL

Når det er ferdig installert, skal selve databasen settes opp.

Går inn på MySQL Workbench og setter opp en test-database, her brukes test-databasen «classicmodels»¹⁴
Logger inn med root, og et passord som er satt og bruker verktøyet *Data Import/restore* til å importere databasen.



For å teste at importen fungerer kjøres spørringen

```
USE classicmodels;  
SELECT * FROM customers;
```

Tabell 6 - MySQL test-database

¹³ (Brasø & Aalrust, installMySQL.ps1, 2020)

¹⁴ (MySQLTutorial, 2020)

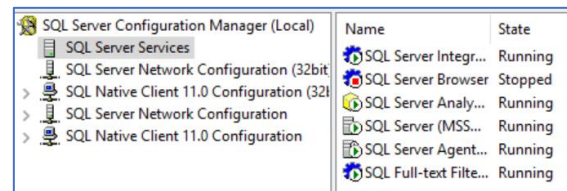
3.3.4.1.2 MSSQL

For MSSQL er prosessen litt annerledes.

3.3.4.1.2.1 Installasjon

Installerer MSSql ved hjelp av skriptet *installMSSQL.ps1*¹⁵

Går inn på SQL Server Configuration Manager og påser at alle SQL Server Services kjører



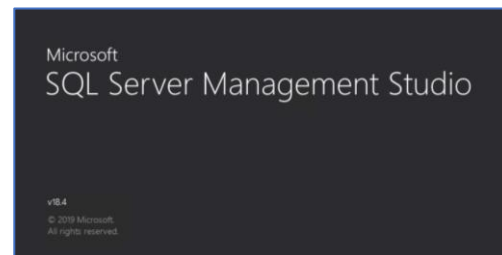
Name	State
SQL Server Integr...	Running
SQL Server Browser	Stopped
SQL Server Analy...	Running
SQL Server (MSS...	Running
SQL Server Agent...	Running
SQL Full-text Filte...	Running

Tabell 7 - MSSQL installasjon

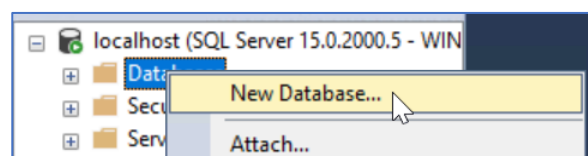
3.3.4.1.2.2 Test-database for MSSQL

Her legges det til en database til SQL-serveren slik at det finnes data som skal migreres til Azure.

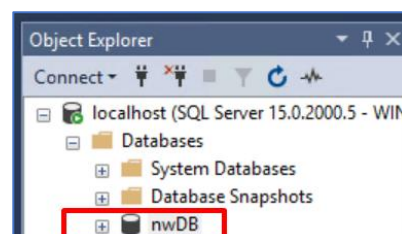
Åpner opp SSMS



Kobler til *localhost*



Benytter SQL-spørringen *insntwnd.sql*¹⁶
Denne spørringen kjøres på den nye databasen. Databasen er deretter klar til bruk.



¹⁵ (Brady, 2018)

¹⁶ (Microsoft, instnwnd.sql, 2018)

Tabell 8 - Northwind database

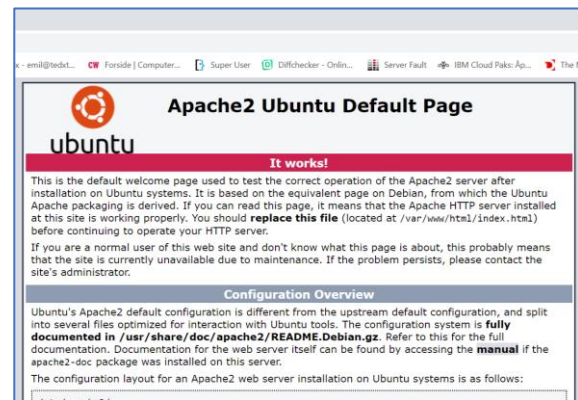
3.3.5 Enkel webserver

Det ble satt opp en webserver på en Linux-VM, med operativsystemet Ubuntu 18.04 LTS. For å drive webserveren ble pakken Apache installert, samt at port 80 ble port-forwarded for å gi ekstern tilgang til nettsiden. Det ble ikke foretatt noen endring på innholdet på web-serveren.

Inne på Ubuntu-serveren ble følgende kommando kjørt for å installere apache2¹⁷

```
sudo apt-get install apache2
```

Etter at dette er kjørt kan opprettelsen kontrolleres ved å gå inn på serverens offentlige IP-adresse. Dette viser kun Apache2 Ubuntu Default Page, siden innholdet ikke er endret.



Tabell 9 - Enkel Webserver

¹⁷ (The Apache Software Foundation, 2020)

4 Utførelse

4.1 Designrapporten

Første utkast av designrapporten forela 18.02.2020. Hensikten med designrapporten er å dokumentere hvordan prosjektgruppen møter kravene som kunden i caseoppgaven stiller. Selv om denne rapporten delte mange av emnene som forstudierapporten tok opp, skulle alt som er relevant for løsningen på prosjektet drøftes i større detalj.

For dette prosjektet betyr dette at alle valg av verktøy og teknologier skulle presenteres og begrunnes, i tillegg til at prosjektet ble avgrenset noe. Her er målene som ble trukket fram:

- En fremgangsmåte som beskriver selve prosessen med å migrere dagens løsning fra on-premises til en hel eller delvis skybasert løsning.
- Ett system som utnytter skybasert teknologi
- En analyse av typen løsning; hel eller delvis og hva som vil gagne bedriften i det lange løp.
- En forståelse av kunders behov og OKV (Opplevd Kunde Verdi)
- Migreringsprosessen skal påvirke den daglige driften minst mulig.

Prosjektgruppen fremstilte at følgende tjenester skulle være på plass i Azure etter migrering:

- AADDS
- AAD
- Azure SQL Databases
- Windows Virtual Desktop
- Data Lake Storage

Overordnet ble det stilt et krav til all teknologien og verktøyene:

- Mest mulig skulle være native til Microsoft Azure.

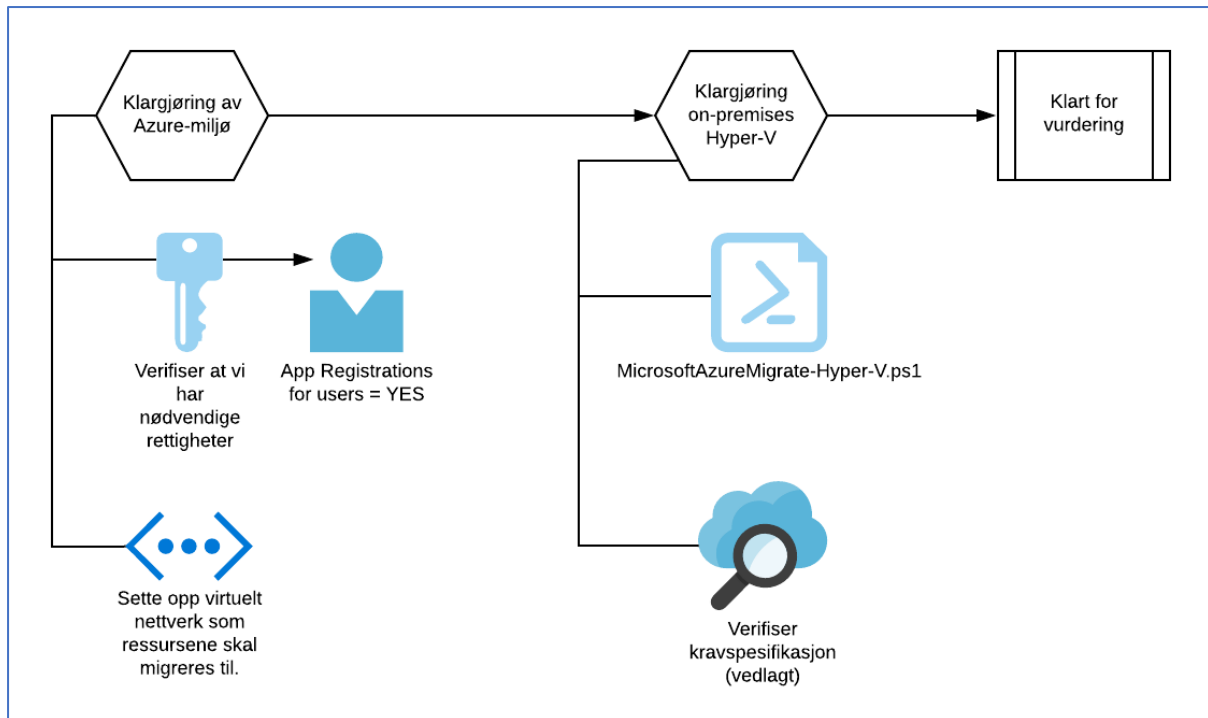
Videre ble følgende verktøy og tjenester brukt på overordnet nivå, med tanke på Azure og on-premises miljø:

Azure	On-Prem	Uavhengig
ARM-templates	Hyper-V	Powershell
Blueprints		Bash
Azure AD Roles and users		GitHub
Azure Migrate		Chocolatey
Azure Automation Accounts		
SQL databases		
Storage Accounts		
WVD		
Azure AD		
Azure AADDS		

Tabell 10 - Verktøy brukt

4.2 Forhåndsarbeid før migrering

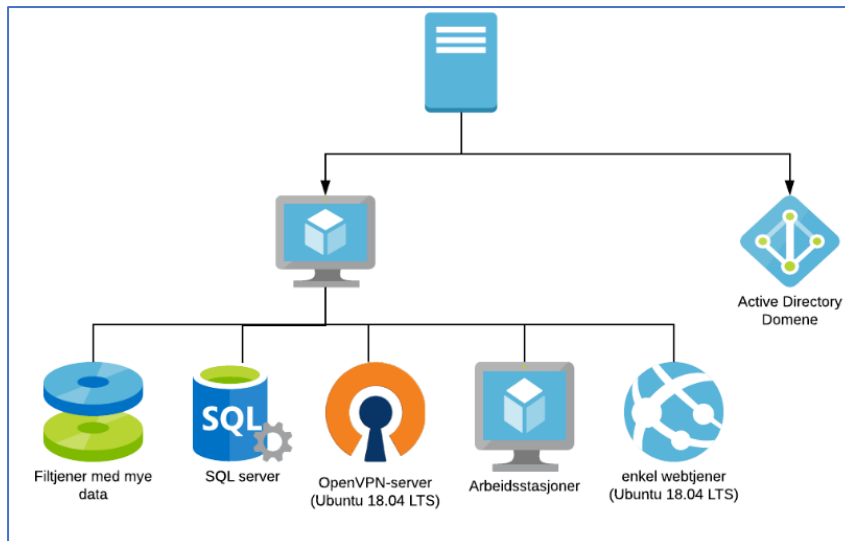
Før selve migreringen kan starte må et par verktøy og systemer være på plass i miljøet. Følgende figur viser en oversikt over dette. I resten av kapittel 4 beskrives det hvordan disse delene er satt opp.



4.2.1 Azure SQL Database

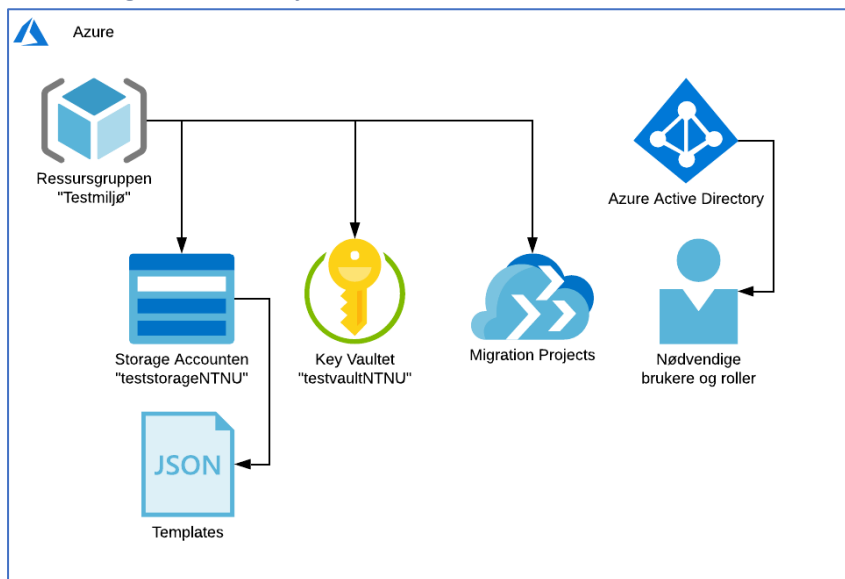
Azure SQL Databaser skal brukes som destinasjon når begge alternativer (MSSQL og MySQL) skal migreres. Dette settes opp ganske enkelt med templatene *sqlDB.json*¹⁹ og *mysqlDb.json*²⁰

4.2.2 Figur on-premises miljø før utførelsen



Figur 12 - On-premises før migrering

4.2.3 Figur Azure-miljø etter utførelse



Figur 13 - Azure miljø etter utførelse

¹⁹ (Brasø & Aalrust, sqlDB.json, 2020)

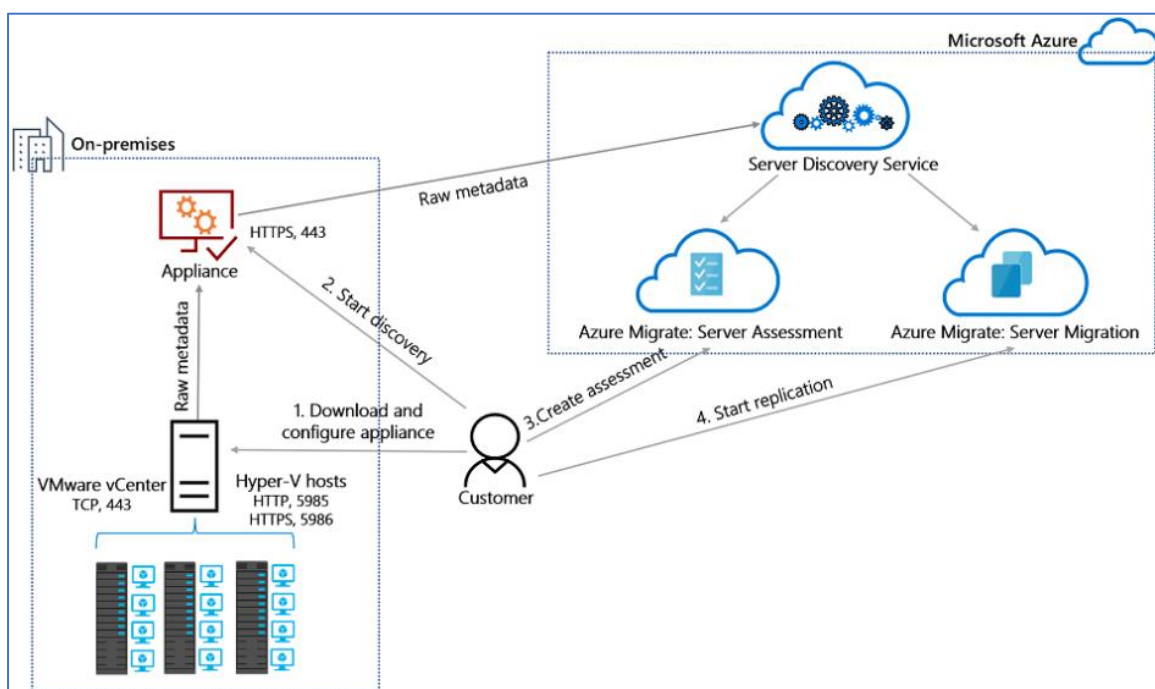
²⁰ (Brasø & Aalrust, mysqlDb.json, 2020)

4.2.4 Laste ned og klargjøre Migrate Appliance

Ved hjelp av skriptet *DownloadAppliance.ps1*²¹ lastes Migrate Appliance ned. Migrate Appliance er et verktøy utviklet av Microsoft som brukes til å migrere on-premises systemet til Azure. Verktøyet kommer i form av en virtuell harddisk (vhd) som inneholder følgende komponenter:

- Management app: En web-app som brukes til assessment for migrering til Azure.
- Discovery agent: Denne agenten samler konfigurasjonsfiler fra maskinene i miljøet og brukes til assessment for migrering til Azure
- Assessment agent: Denne agenten samler ytelsesdata. Brukes til assessment for migrering til Azure
- Auto update service: Oppdaterer appliance komponenter

Bildet ²² under viser en oversikt over de forskjellige komponentene i Migrate Appliance og hvordan de knyttes opp mot Azure.



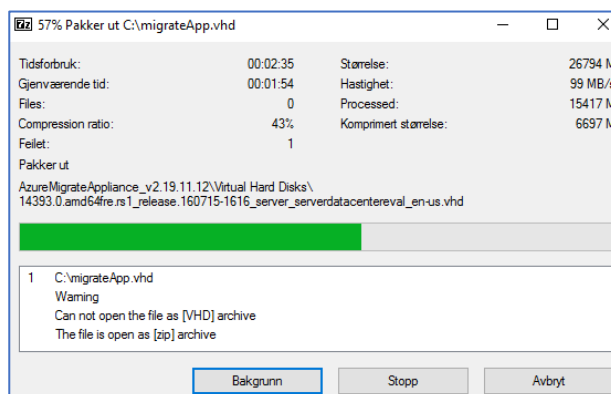
Figur 14 - Azure Migrate Appliance

²¹ (Brasø & Aalrust, DownloadAppliance.ps1, 2020)

²² (Microsoft, Azure Migrate Appliance Architecture, 2020)

Når migrate appliance disken er lastet ned gjøres følgende steg på Hyper-V serveren:

Pakk ut arkivet med et valgfritt verktøy. Når dette er ferdig, inneholder mappen en .vhd-fil. Dette er en virtuell harddisk som kan knyttes opp til en VM.



Før man går videre må sikkerheten på filen verifiseres, som gjøres med et powershell-kommandoen

```
PS C:\Users\endre> Get-FileHash -Path  
>> ./AzureMigrateAppliance_v1.19.06.27.zip-Algorithm SHA256
```

Tabell 11 - Laste ned MigrateAppliance

4.2.5 Klargjøre Hyper-V serveren med et skript

Ved å kjøre *MicrosoftAzureMigrate-Hyper-V.ps1*²³ vil Hyper-V serveren kontrolleres og klargjøres for migrering. Innehaver av skriptet er Microsoft, og de anbefaler at man validerer skriptets integritet før kjøring. Dette gjøres slikt:

Etter at skriptet er lastet ned kjøres kommandoen

```
CertUtil -HashFile ` C:\Users\Administrators\Desktop\MicrosoftAzureMigrate-Hyper-V.ps1  
SHA256
```

Etter at dette er godkjent, så skal skriptet kjøres på hver enkelt Hyper-V server. I dette tilfellet foreligger det kun en server.

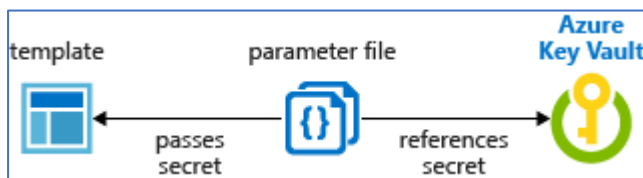
Tabell 12 - Klargjøring av Hyper-V server

²³ (Microsoft, Prepare for assessment and migration of Hyper-V VMs to Azure, 2020)

4.2.6 Key Vault

For at linked-templates skal fungere ordentlig og sikkert, er det nødvendig å sette opp et Key Vault på Azure. Azure Key Vault er en tjeneste som brukes for å administrere nøkler, hemmelige filer og sertifikater, og skal blant annet eliminere behovet for å oppbevare passord i koden.

Følgende figur beskriver forholdet mellom template, template-parameter-filer og et Key Vault.



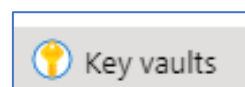
Figur 15 - Forhold Template – Parametere – Key Vault

En template får sine parametere fra en parameter-fil. For hemmelige verdier vil da parameter-filen gi verdien til template, men henter den selv gjennom en referanse til en secret i Azure Key-vaultet. Det er særlig her det ekstra laget med sikkerhet kommer inn. Dersom man ikke er autentisert med en konto med riktig tilgang (gjennom kommandoen Connect-AzAccount e.l.), vil man ikke kunne hente ned denne verdien. Slik sikres det at kun autoriserte brukere får tilgang til verdiene.

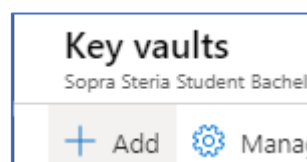
En Secret vil her være hemmelige og krypterte verdier. Dette kan være i form av strenger, objekter eller lignende. Som sagt brukes dette for å gi passord til template-filene som blir brukt.

4.2.6.1 Sette opp key-vault

Går inn på tjenesten *Key Vaults*



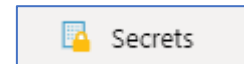
Velger *Add* for å legge til nytt Key Vault
Det er ikke nødvendig å endre noen parametere, og etter en stund er Key vaultet klart.



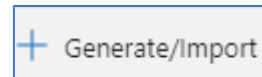
Tabell 13 - Key Vault

4.2.6.2 Legge til en Secret

Går inn på *Secrets* fra oversikten til nøkkelhvelvet.



Generate/Import



Velg et dekkende navn (med mange Secrets kan det bli rot), og skriv inn den hemmelige verdien under «Value»

Bruker kommandoen `Set-AzKeyVaultAccessPolicy` for å tillate Templates å ta i bruk key-vaultet.

```
Set-AzKeyVaultAccessPolicy -VaultName
"ExampleVault"
-EnabledForTemplateDeployment
```

Tabell 14 - Legg til Secret

4.2.6.3 Referere til en Secret fra en Template

For å referere til en Secret i et Key Vault i Azure må følgende del legges til i parameter-filen til en template.

```
"reference": {
  "keyVault": {
    "id": "/subscriptions/<subscription-id>/resourceGroups/<rg-
name>/providers/Microsoft.KeyVault/vaults/<vault-name>"
  },
  "secretName": "ExamplePassword"
}
```

Følgende Template viser et eksempel på hvordan dette blir brukt i dette oppsettet: (hentet fra *mysqlDb.json*)

```
},
"administratorLoginPassword": {
  "type": "securestring"
},
"password": {
```

Og tilhørende parameter i parameter-filen

```
"administratorLoginPassword": {  
  "reference": {  
    "keyVault": {  
      "id": "/subscriptions/[redacted]/resourceGroups/testMiljoe/providers/Microsoft.KeyVault/vaults/testVault/  
    },  
    "secretName": "vmAdminPwd"  
  }  
}
```

Tabell 15 - Referere til Secret

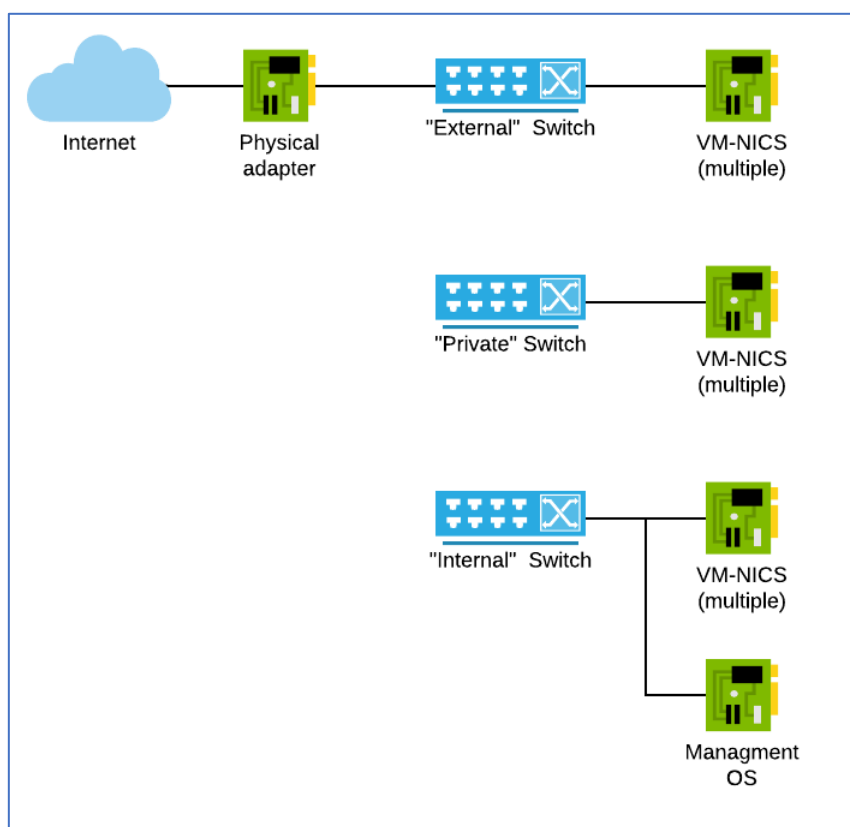
Dersom alt er fulgt, skal kan nå Key Vaultet brukes fra Templates.

4.2.7 Konfigurasjon VM-er (Hyper-V)

Utenom oppsettet av OpenVPN på serveren er det nødvendig å endre noen IP-regler, samt legge til et ekstra nettverkskort. Dette er for å kunne påse at kommunikasjon mellom VMene (og nettet som VMene tilhører) kan kommunisere med selve Hyper-V hosten.

Dette var et problem som oppstod under installasjonen av testmiljøet. Til å starte med var OpenVPN-serveren og de andre VM-ene kun koblet til en «External» Vswitch. Denne kan kun brukes til å tilby tilkobling ut mot internett.

Etter litt feilsøking ble det funnet ut at VM-ene må være tilkoblet til en «Internal» Vswitch også, for å kunne kommunisere med den fysiske hosten direkte. Figuren under forsøker å vise skillet mellom disse:



Figur 16 - Nettverkskort på en vpnserver

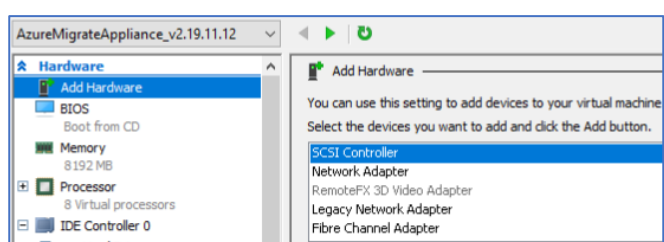
- En *External Switch* tilbyr tilkobling mellom Nettverkskortene til VM-ene og det fysiske nettverkskortet på serveren, og derved ut mot internett, i tillegg til tilkobling mellom VM-ene.
- En *Private Switch* tilbyr tilkobling kun mellom nettverkskortene til VM-ene.
- En *Internal Switch* tilbyr tilkobling mellom en nettverkskortet til en VM, og operativsystemet som administrerer Hyper-V på serveren. (Serverens OS, eller Management OS på engelsk)

Utenom nettverks-konfigurasjonen er det ikke nødvendig å endre særlig mye på oppsettet til VM-ene.

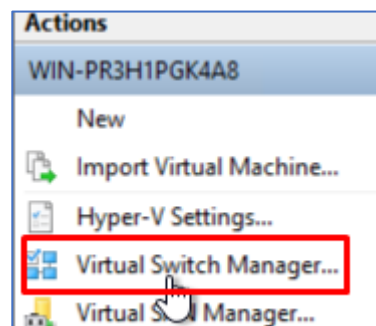
4.2.7.1 Sette opp Internal Switching på VM-ene som trenger det.

Slik ble det gjort:

Fra Settings og Hardware kan et nytt nettverkskort legges til, ved hjelp av opsjonen *Network Adapter*

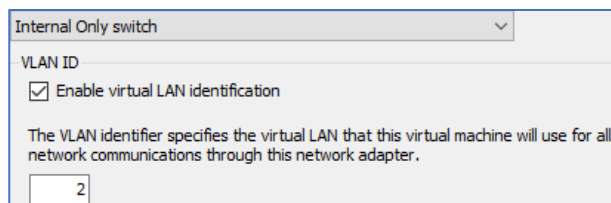


Samtidig må det være opprettet en Virtuell Switch av korrekt type. Dette gjøres ved å gå inn på *Virtual Switch Manager*



Deretter *New virtual network switch*, hvor typen skal være *Internal* i dette tilfellet.

Siden dette miljøet ikke skal være spesielt stort, trengs det ikke å settes opp ytterlige switcher.



Tabell 16 - Internal switching på vm

4.2.8 Opprette ressursgruppe i Azure

For å kunne migrere maskinene i on-premises miljøet er det nødvendig å opprette en ressursgruppe i Azure som kan ta imot maskinene. Dette kan gjøres gjennom Azure portalen, eller gjennom en egentemplate som vist under. Templateen som tas i bruk er *resourceGroup.json*²⁴

Andre steder i denne rapporten vil kommandoen for *New-AzResourceGroupDeployment* brukes.

Her brukes derimot *New-AzSubscriptionDeployment*. Dette har med at ressurs-gruppen må rulles ut på abonnements-nivå, den kan ikke rulles ut i en allerede eksisterende ressursgruppe

```
C:\Users\endre> New-AzSubscriptionDeployment  
-templateURI "C:\Scripts\resourceGroup.json"
```

Definerer plassering *northeurope* og navnet på ressursgruppen, i dette eksemplet *testMig31*

```
cmdlet New-AzDeployment at command pipeline position 1  
Supply values for the following parameters:  
(Type |?) for Help.)  
Location: northeurope  
resourceGroupName: testMig31
```

Når utrulling er fullført fra PowerShell, vil resultatet vises. Ettersom det her står at *provisioningState* er *Accepted*, betyr det at utrulling ble godkjent av ARM.

```
}  
  "mode": "Incremental",  
  "provisioningState": "Accepted"  
}
```

Tabell 17 - Opprette ressursgruppe med PowerShell

²⁴ (Brasø & Aalrust, resourceGroup.json, 2020)

4.3 Main.json

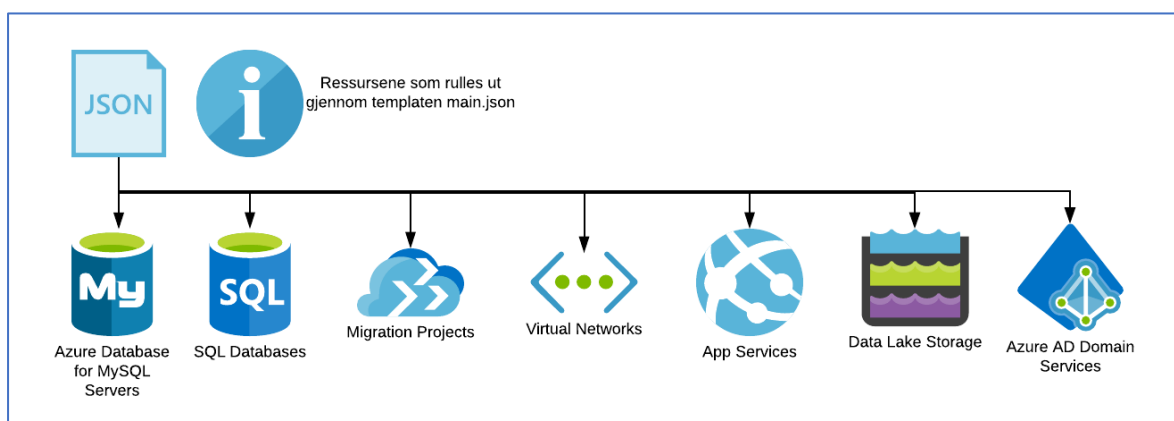
For å rulle ut templates til miljøet i Azure blir en ARM-template kalt *main.json*²⁵ brukt.

Bruk av denne template er avhengig av at det foreligger en ressursgruppe som ressursene kan ruller ut til. Dette er gjort i 4.2.8²⁶. I tillegg er det nødvendig med Storage Accounts for å oppbevare template-filene i Azure. Oppsett av dette er beskrevet i 4.5²⁷

For å rulle ut disse ressursene til Azure, benyttes Azure Resource Manager Templates. Det går ikke inn på nøyaktig syntaks her, men generell fremgangsmåte for oppsett og bruk av ARM-Templates blir beskrevet. Følgende ressurser ruller ut gjennom template main.json.

MySQL DB
MSSQL DB
Et Azure Migrate Project
Storage Account
Et Virtuelt nettverk
Web app
Azure Active Directory Domain Services

Disse komponentene er beskrevet nærmere i Designrapporten²⁸.



Figur 17 - Ressurser som ruller ut gjennom main.json

4.3.1 Oppbygging av Main.json.

Azure Resource Manager er en styringsfunksjon i Azure som lar brukerne opprette, oppdatere og slette ressurser på området og dette er denne som tas i bruk med templates. For at man skal kjøre ut mange ressurser gjennom en template er det god praksis å enten bruke metodene for **nested deployment**, eller **linked-templates**. Førstnevnte framgangsmåte vil medføre at alle ressursene blir opprettet fra en og samme Template.

Siden mange ressurser skal ruller ut i dette tilfellet, blir heller linked-templates brukt. Dette for å gjøre hoved-template så liten som mulig, samtidig som lesbarheten opprettholdes.

²⁵ (Brasø & Aalrust, main.json, 2020)

²⁶ Opprette ressursgruppe i Azure

²⁷ Storage Account i Azure

²⁸ (Brasø & Aalrust, Designrapport, 2020)

Denne framgangsmåten går ut på at main.json ruller ut flere små Templates ved hjelp av referanser til disse, i dette tilfellet en Template for hver ressurs. I tillegg er det referanser til parameter-filer, som gir parameterne som skal brukes. Figurene under viser noen av disse.

Template-filer:

```
"VMsURI": "https://teststoragentnu.file.core.windows.net/t",  
"mySQLdbURI": "https://teststoragentnu.file.core.windows.r",  
"MSSQLdbURI": "https://teststoragentnu.file.core.windows.r"
```

Figur 18 - lenker til template-filer

Parameter-filer:

```
"mySQLdbParams": "https://teststoragentnu.file.core.windows",  
"MSSQLdbParams": "https://teststoragentnu.file.core.windows"
```

Figur 19 - Lenker til parameter-filer

Disse lenkene/referansene **må** ligge på en plassering som er tilgjengelig for Azure Resource Manager, og her ble det valgt å ta i bruk en Storage Account med File share på Azure-miljøet. Når disse referansene er lagt inn, kan man rulle ut disse ressursene ved hjelp av følgende syntaks:

```
{  
  "type": "Microsoft.Resources/deployments",  
  "apiVersion": "2017-05-10",  
  "name": "linkedSQLdbTemplate",  
  "properties": {  
    "mode": "Incremental",  
    "templateLink": {  
      "uri": "[variables('MSSQLdbURI')]",  
      "contentVersion": "1.0.0.0"  
    },  
    "parametersLink": {  
      "uri": "[variables('MSSQLdbParams')]",  
      "contentVersion": "1.0.0.0"  
    }  
  }  
},
```

Figur 20 - Deployment av MSSQL fra main.json

For at man skal få tilgang til filene i en Storage Account, må URL-ene skapes slik som vist i 4.6.1.2²⁹

²⁹ Shared Access Signatures for bruk med Templates

4.3.2 Utrulling og sporing av main.json

For å rulle ut en Template og tilhørende parameter-fil ble følgende PowerShell-kommandoer brukt:

```
#koble til Azure
Connect-AzAccount

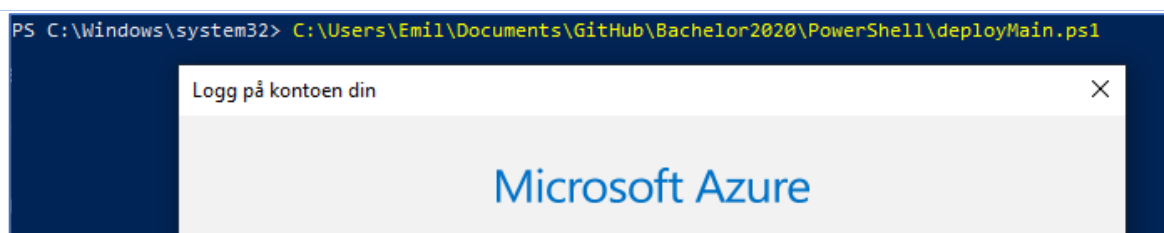
#Deklarere filstier
$templateFile = "filsti"
$templateParameterFile = "filsti"

#Deklarere ressursgruppen
$resourceGroup = "RessursGruppe"

#Rulle ut på ressursgruppen
New-AzResourceGroupDeployment -ResourceGroupName $resourceGroup `
-TemplateFile $templateFile -TemplateParameterFile $templateParameterFile
```

Figur 21 - Rulle ut templer-fil

Figuren under viser et eksempel på en slik utrulling fra PowerShell-filen som er opprettet.



Her må man logge inn fordi kommandoen *Connect-Az* brukes.

På Azure-portalen vil utrullingens spores gjennom *Deployments* på tilsvarende ressursgruppe. I dette tilfellet «Test1604» På figuren ser man hoved-templatene *main.json* på bunnen, samt linked-templatene over. *linkedStorageAccountTemplate* er følgelig en lenket template som skal opprette en Storage Account

Deployment name	Status
<input type="checkbox"/> linkedStorageAccountTemplate	✓ Succeeded
<input type="checkbox"/> linkedSQLdbTemplate	✓ Succeeded
<input type="checkbox"/> linkedVMsTemplate	✓ Succeeded
<input type="checkbox"/> linkedmySQLdbTemplate	✓ Succeeded
<input type="checkbox"/> linkedmigrateProjectTemplate	✓ Succeeded
<input type="checkbox"/> main	✓ Succeeded

Tabell 18 - Utrulling main.json

Dette er til felles for alle utrullingene som skjer gjennom *main.json*. for å repetere, så er *main.json* til stede for å klargjøre for selve migreringen som skal skje i etterkant. Ved at det er flere templates som er lenket til *main.json* blir formatet mer modulært og oversiktlig under utrullingene. Ett eksempel-skript for å rulle ut til en ressursgruppe er lagret som *deployToRG.ps1*³⁰

³⁰ (Brasø & Aalrust, *deployToRG.ps1*, 2020)

4.4 App service i Azure

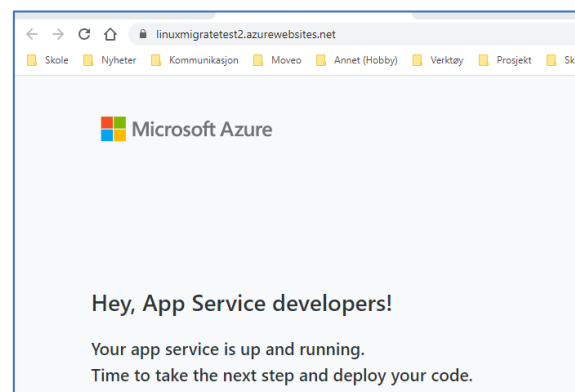
En Web app, eller App Service i Azure er en innebygd tjeneste for å drive web-applikasjoner og nettsteder. Dette er en enkel og fleksibel løsning på å videreføre nettstedet i Azure. Etter at tjenesten er satt opp er det mange ulike måter man kan publisere nettstedet sitt på. Et eksempel er GitHub eller Azure repos, her blir det vist med et enkelt eksempel, FTP.

Selve opprettelsen av Web appen som ressurs i Azure blir håndtert av den linked-templaten *newWebApp.json*³¹ og parameterfila *webAppParameters.json*³², som rulles ut gjennom *main.json*

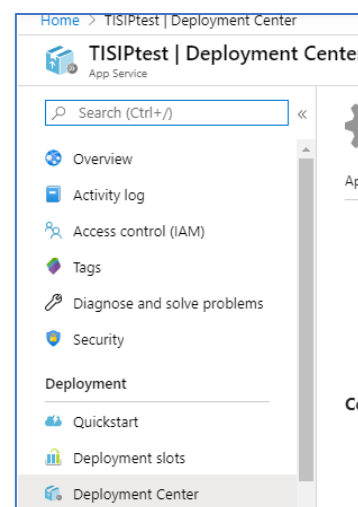
4.4.1.1 Fremgangsmåte

Det er noen steg som er nødvendig for å få web-appen oppe og kjøre, gjennom Azure portalen også.

Uten at noe av innholdet er endret, kan man se standard-siden til web-appen gjennom URL-en <https://linuxmigratetest2.azurewebsites.net/> URL-en er avhengig av det man definerte under opprettelse, men kan hentes gjennom oversikten til ressursen.



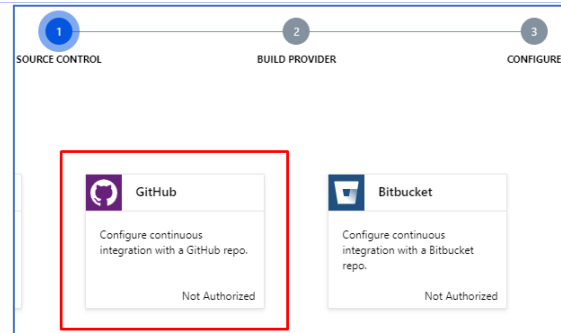
For å rulle ut kode til denne tjenesten, vil man gå gjennom *Overview > Deployment Center*.



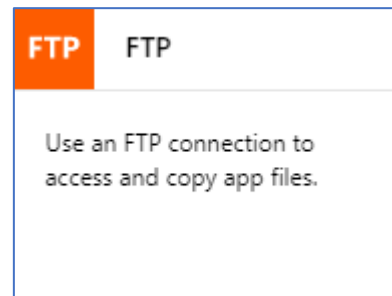
³¹ (Brasø & Aalrust, newWebApp, 2020)

³² (Brasø & Aalrust, webAppParameters.json, 2020)

Her kan man bruke sitt eget valg av kildekontroll, deriblant GitHub eller Azure Repos.



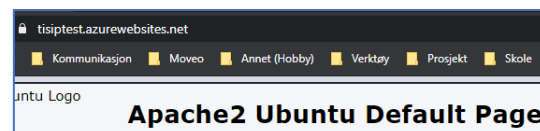
På denne testen brukes simpeltent FTP, siden det uansett illustrerer hvordan web app-en fungerer.



Et eksempel på verktøy som kan brukes for å overføre filer over til webserveren er WinSCP.



Dersom innholdet fra den opprinnelige apache-serveren føres over til web appen på Azure, kan man se at standard-nettstedet som kom med Apache, kommer opp. Dette viser at web-appen fungerer som den skal.



Tabell 19 - App Service

4.5 Storage Account i Azure

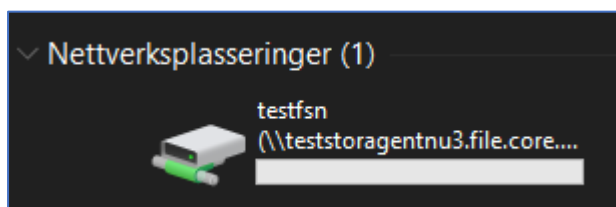
I stedet for en dedikert filserver i form av en VM, blir det tatt i bruk Azure Storage Accounts. Dette tillater blant annet å opprette server-løse SMB file shares, som kan monteres som tradisjonelle drev på en hvilken som helst arbeidsstasjon, med PowerShell. For å opprette denne ressursen brukes da linked-templaten *storageAccountFileShare.json*³³, som blir rullet ut via *main.json*

4.5.1.1 Koble til oppsatt file share.

Ved opprettelse av file-shares på Azure blir det automatisk lagd et PowerShell-skript for å koble til den angitte sharen. Hver share vil ha et individuelt skript, men et eksempel på hvordan dette ser ut er vedlagt i form av filen *mapShare.ps1*³⁴

Dette tillater mye forskjellig funksjonalitet, og vil ta over som hjemme-mapper i dette miljøet. Brukerne vil være i stand til å få disse på sine lokale datamaskiner i tillegg til arbeidsstasjonene på Azure, svært enkelt.

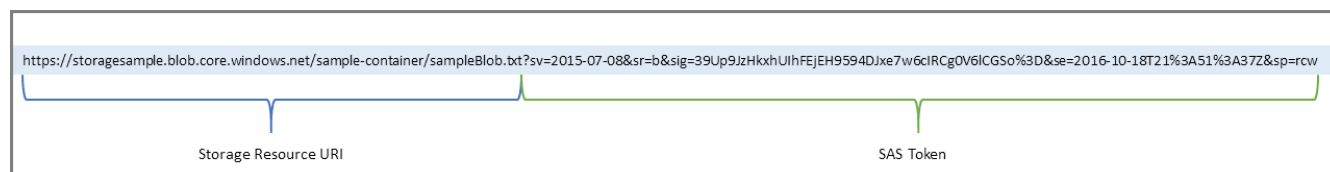
Når skriptet for å legge til sharen er kjørt, dukker det opp som en nettverks plassering på datamaskinen.



Figur 22 - Nettverks plasseringen vises i filutforsker

4.5.1.2 Shared Access Signatures for bruk med Templates

For at skriptene skal ha tilgang til filer på en Storage Account i Azure, uten intervensjon fra brukere, må det settes opp Shared Access Signatures. Dette er genererte strenger som gir tilgang til ressurser. Det er to måter å gjøre det på, enten gjennom portalen eller igjennom PowerShell. Her blir det tatt utgangspunkt i PowerShell. Skriptet *addSASToken.ps1*³⁵ skaper en slik streng. Denne strengen må deretter kombineres med URI-en til filen, slik som beskrevet på denne figuren³⁶:



Figur 23 - Shared access signature

Dette også blir håndtert av skriptet. Skriptet vil til slutt produsere en variabel «*\$connectionString*» som kan legges inn i *main.json*

³³ (Brasø & Aalrust, storageAccount'WihtFileShare.json, 2020)

³⁴ (Brasø & Aalrust, mapShare.ps1, 2020)

³⁵ (Brasø & Aalrust, addSASToken.ps1, 2020)

³⁶ (Microsoft, Grant limited access to Azure Storage resources using shared access signatures (SAS), 2019)

4.6 Azure AD Domain Services

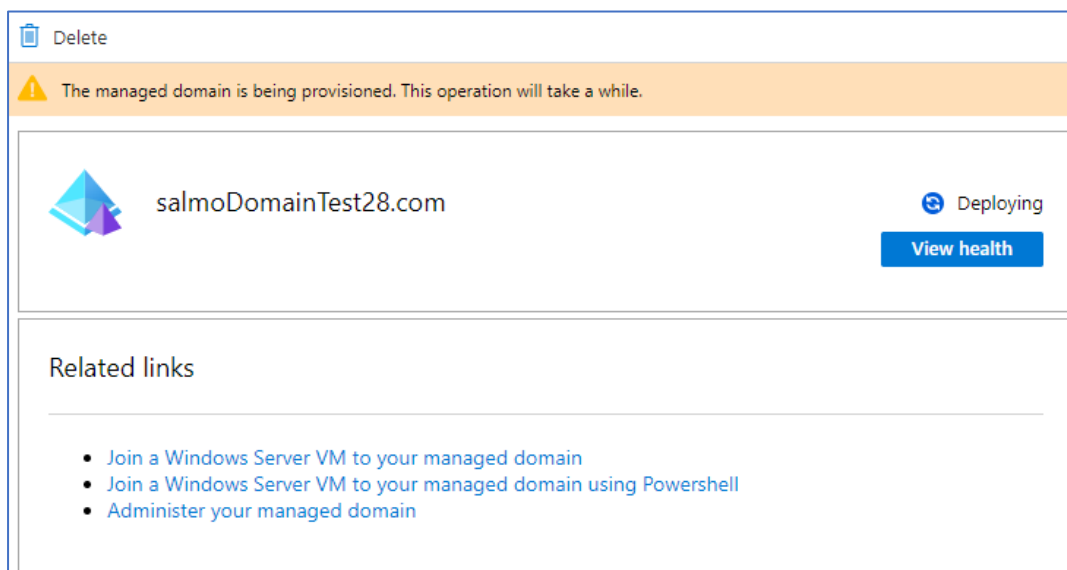
Mesteparten av denne løsningen kan rulles ut med IaC, men noen steg må gjøres manuelt via portalen. For IaC-delen blir det håndtert gjennom hoved-templaten *main.json*, og linked-templaten *aadds.json*³⁷. Her dekkes de stegene som må gjøres manuelt.

I denne templatene blir det faktisk tilrettelagt for WVD, som videre beskrives i 4.13³⁸. WVD er avhengig av en fungerende instans av AADDs for at den skal kunne fungere. Det eneste som gjøres annerledes i når det gjelder oppsettet av AADDs, er det blir lagd et ekstra Subnet, for bruk av WVD.

Grunnen til at dette gjøres er for at WVD, og ressursene som opprettes sammen med WVD, skal kunne ha tilgang til Azure AADDs. Derfor er de plassert i samme vNet. Et annet alternativ kunne ha vært å ha satt opp Peering mellom to adskilte vNet, men for denne løsningen var ikke dette nødvendig.

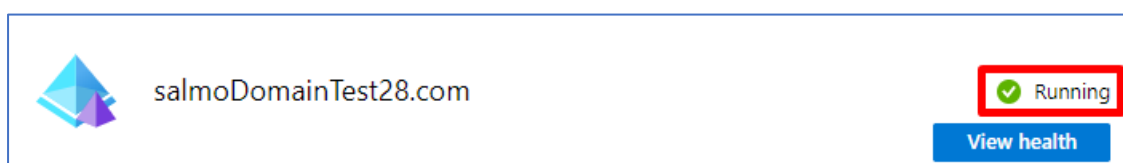
4.6.1 Manuelt oppsett etter opprettelse av ressurser

Etter at domenet er rullet ut fra templatene, vil videre klargjøring av domenet ta en stund.



Figur 24 - Azure managed domain

Men når den er ferdig klargjort, kan man se at statusen er endret fra *Deploying* til *Running*




Figur 25 - Domain running

³⁷ (Brasø & Aalrsut, aadds.json, 2020)

³⁸ WVD

For at domenet skal bli fullstendig klart for bruk, må handlingene som er beskrevet under *Required configuration steps* fullføres.


Required configuration steps



Update DNS server settings for your virtual network

Update the DNS server settings for your virtual network to point to the IP addresses (10.1.0.5 and 10.1.0.4) where Azure AD Domain Services is available.
[More information](#)

Configure



Enable Azure AD Domain Services password hash synchronization

Users cannot bind using secure LDAP or sign in to the managed domain, until you enable password hash synchronization to Azure AD Domain Services. Follow the instructions below, depending on the type of users in your Azure AD directory. Complete both sets of instructions if you have a mix of cloud-only and synced user accounts in your Azure AD directory.

- [Instructions for cloud-only user accounts](#)
- [Instructions for synced user accounts](#)

Figur 26 - Configuration steps

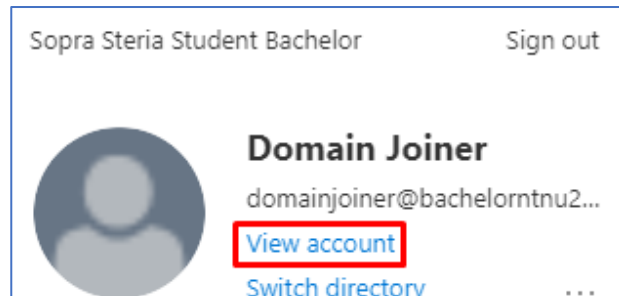
Av dette er det kun «*Update DNS server settings for your virtual network*» som må gjøres som en del av denne prosessen. Dette gjøres enkelt ved å bruke knappen *Configure*, deretter startes en automatisk prosess som setter opp det virtuelle nettverket.

Dersom brukerne i domenet ble opprettet før Azure AD Domain Services ble lagt til, må man endre til Kerberos autentisering. Dette beskrives i neste delkapittel.

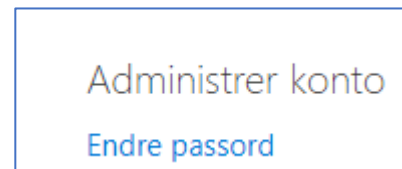
4.6.2 Endre til kerberos autentisering

Etter at AADDS er lagt til i miljøet, må brukerne som skal ta i bruk dette endre passordet sitt. Dette er for å endre autentiseringstypen på passord til kerberos, om ikke dette gjøres vil ikke kontoen kunne brukes til en rekke tjenester på Azure.

Logg inn på brukeren og velg *View Account*



Endre passord og vent 20 minutter, så er kontoen klar for bruk igjen.



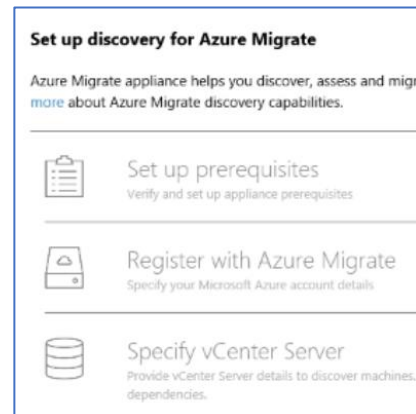
Tabell 20 - Kerberos autentisering

I første omgang er det kun nødvendig å gjøre dette med *domainjoiner*-kontoen, ettersom den skal brukes til å melde arbeidsstasjonene inn i domenet (dersom man følger denne fremgangsmåten kronologisk).

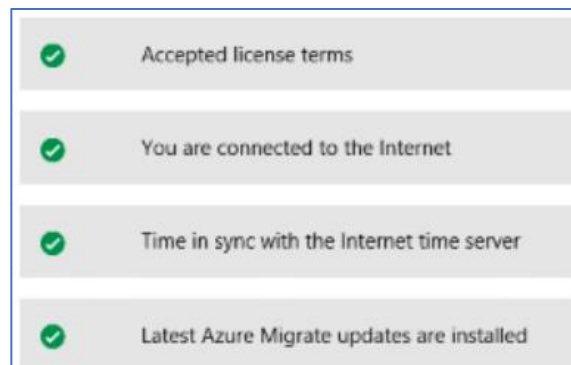
4.7 Discovery og Assesment av Virtuelle Maskiner

Når et Hyper-V miljø skal vurderes, må det lastes ned en virtuell harddisk (.vhd) som er klaggjort og forhåndsinstallert med migrerings-applikasjonen til Azure Migrate. Etter at .vhd-en er nedlastet må den monteres på en VM, som deretter kan brukes til å oppdage og vurdere miljøet på den fysiske hosten. Dette beskrives nærmere i 4.2.4³⁹. Dersom serveren hadde vært VMware i stedet, ville nedlastning og installasjon av Migrate Appliance blitt gjort mer eller mindre manuelt på en passende VM.

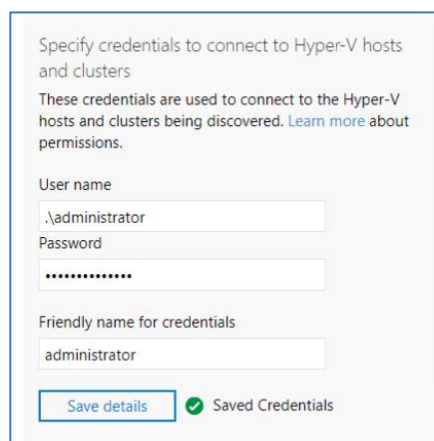
Når VM-en er ferdig satt opp og tilkoblet, blir nettleser automatisk åpnet.



Azure Migrate Appliance vil først gå gjennom en sjekkliste for å verifisere at miljøet er klart til å migreres.

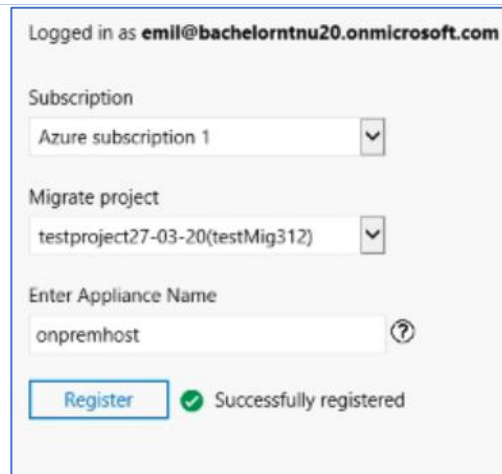


Deretter må en konto på Azure-miljøet logges på, samt administratorbrukeren til Hyper-V hosten.



³⁹ Laste ned og klaggjøre Migrate Appliance

Når forhåndskravene er ferdige, registreres denne *Appliance* opp mot korrekt Migrate Project i Azure

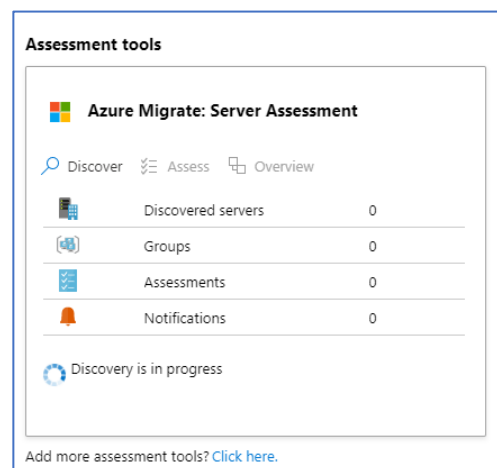


Når dette er gjort, legges IP-adressen til hosten til. Her er det viktig å ha satt opp swithcingen riktig, slik at VM-en for Migrering faktisk oppdager OS-et på Hyper-V hosten. Dette er beskrevet i 4.2.8⁴⁰

Host / Cluster	VMs	Status	Action
WIN-PR3H1PGK4A8	10	✓	Delete

Initiating discovery and configuring appliance.

På Azure kan man se at Discovery kjører på miljøet



Assessment tools	
Azure Migrate: Server Assessment	
Discover Assess Overview	
Discovered servers	0
Groups	0
Assessments	0
Notifications	0
Discovery is in progress	

Add more assessment tools? [Click here.](#)

⁴⁰ Sette opp Internal Switching på VM-ene som trenger det.

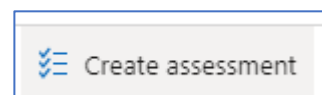
Etter en stund er discovery ferdig og maskinene som ble oppdaget gjennom migrate appliance vises

En detaljert oversikt over serverne vises, og en assessment kan startes, under vises resultatet av en slik assessment

En assessment er ikke pålitelig etter en kort stund, men vil bli det etter hvert. Det er derfor greit å vente noen timer.

Name	IP address	Applications Discovered
OpenVPNserver	-	Not supported
migrateAppliance	192.168.1.13,192.168.38.141,fe8...	Not supported
SQLserver	192.168.38.123,fe80::2589:6f93:2...	Not supported
Filtjener	192.168.38.125,fe80::aca6:760b:7...	Not supported
Workstation1	-	Not supported
Workstation0	192.168.38.120,fe80::85ca:ec57:a...	Not supported
webtjener	-	Not supported
MLTjener	192.168.38.124,fe80::45f6:ac7d:5...	Not supported
SalmoDC	192.168.38.132,fe80::dd7d:efefe...	Not supported
apache	-	Not supported

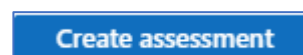
Før maskiner som skal være med migreres, velges *create assessment* slik at Azure får informasjon om hver enkelt maskin.



Her kan maskiner som skal være med i migreringen velges. Det er også mulig å dele opp samlingen med maskiner i flere deler, slik at for eksempel de mest kritiske maskinene migreres først.

<input checked="" type="checkbox"/>	Filtjener
<input type="checkbox"/>	Workstation1
<input checked="" type="checkbox"/>	Workstation0
<input type="checkbox"/>	webtjener
<input checked="" type="checkbox"/>	MLTjener

Bekrefter at *assessmentet* opprettes



Det er nå blitt opprettet en gruppe og et assessment

Assessment tools

Azure Migrate: Server Assessment

Discover Assess Overview

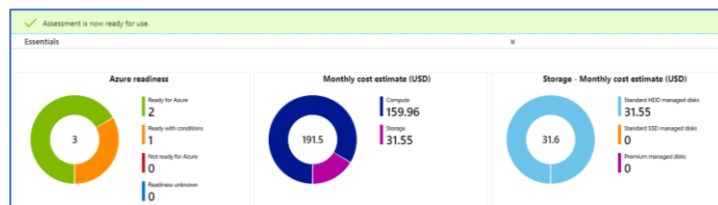
	Discovered servers	10
	Groups	1
	Assessments	1
	Notifications	0

Next step: Start migrating your servers or optionally you can refine your application grouping with dependency analysis

Her kan man se innholdet i hver enkelt assessment

Name	Group	Status	Machines
assessment1	tjenere	Ready	3

Inne på et assessment kan man også få informasjon om kostnader og mer. Confidence-ratingen på dette vil være lav. Helst burde det ventes minst en dag slik at det kan samles inn mer data på dette.

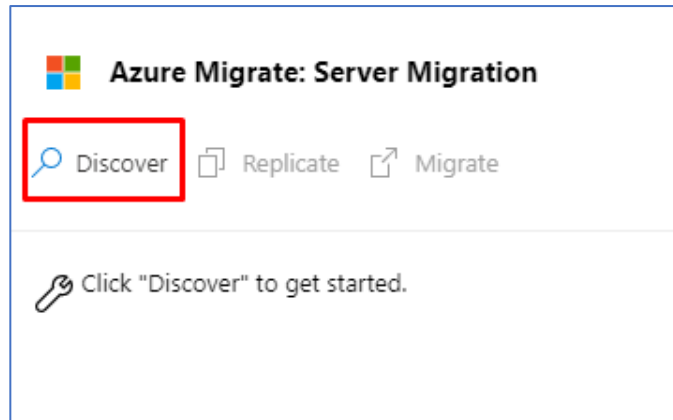


Tabell 21 - Discovery og Assessment

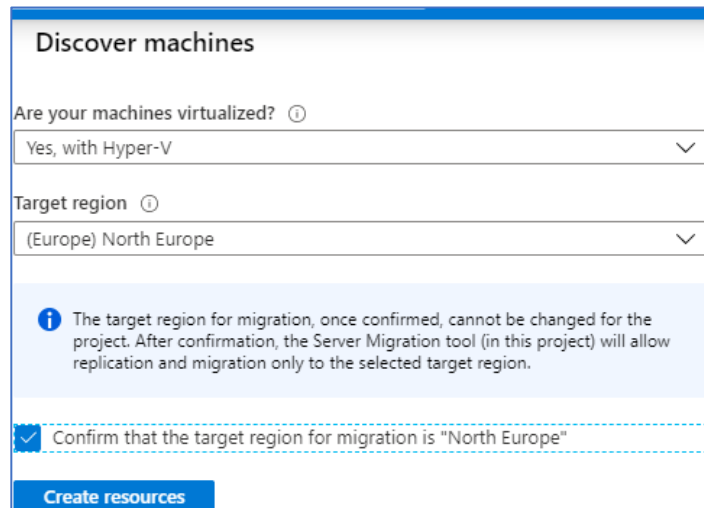
4.8 Registrer Hyper-V host (Azure Migrate)

Før migrering kan starte må Hyper-V hosten registreres opp mot Azure Migrate.

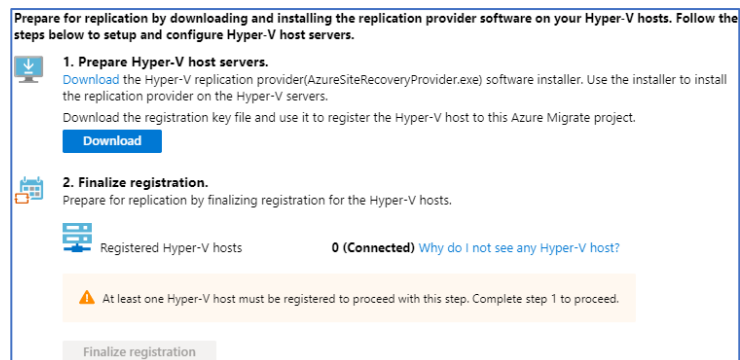
Etter at assessment er gjennomført kan replikasjon og migrering starte. Først må Discover velges fra Azure Migrate: Server Migration.



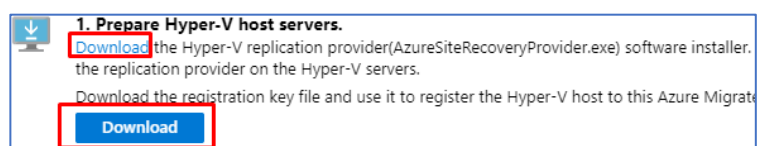
Her velgers det hva slags maskiner som skal oppdages og hvilken region de befinner seg i



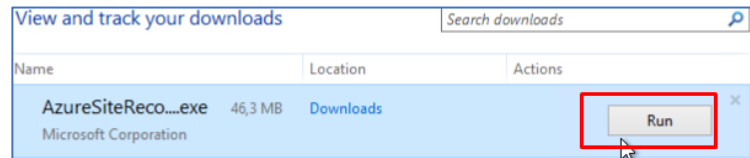
Et program for Azure Site Recovery må lastes ned på Hyper-V hosten. Azure Migrate benytter Site Recovery til å utføre migreringsprosessen.



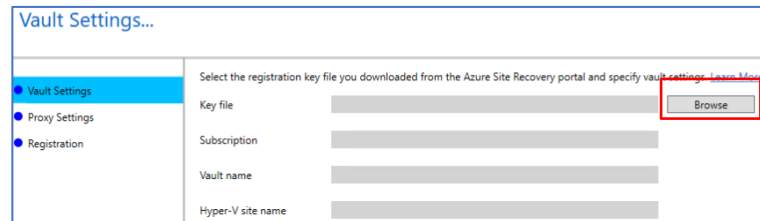
På Hyper-V hosten lastes først ned programmet



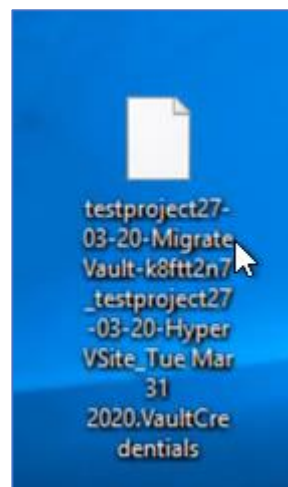
Kjører programmet



Programmet åpner seg, og nøkkelfilen må lastes ned gjennom den andre lenken på Prepare Hyper-V host servers



Slik ser credentials-filen ut. Den må brukes under Vault Settings.



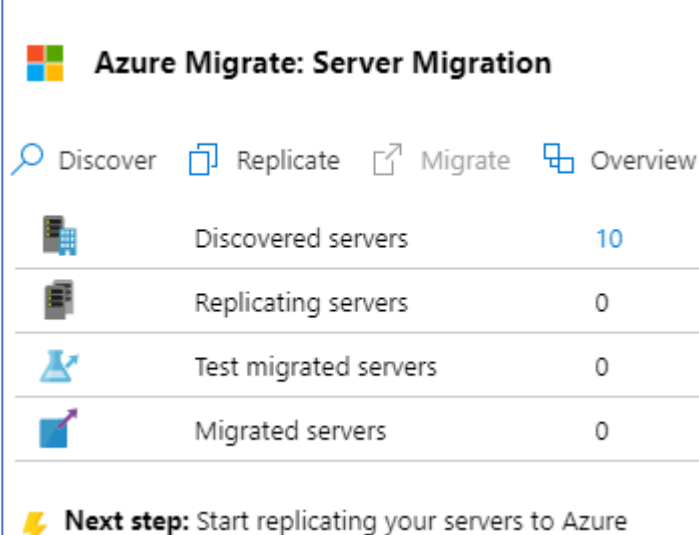
Etter at credentials-filen er lagt inn, trengs det ikke å endre noen andre instillinger her. Går videre.







Serveren ble lagt til i Site Recovery



Videre i Azure er 10 servere klare til migrering



Azure Migrate: Server Migration			
Discover	Replicate	Migrate	Overview
	Discovered servers	10	
	Replicating servers	0	
	Test migrated servers	0	
	Migrated servers	0	

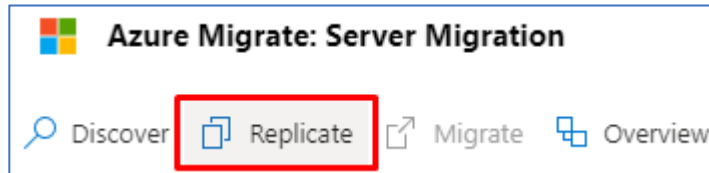
Next step: Start replicating your servers to Azure

Tabell 22 - Discovery for Migrate

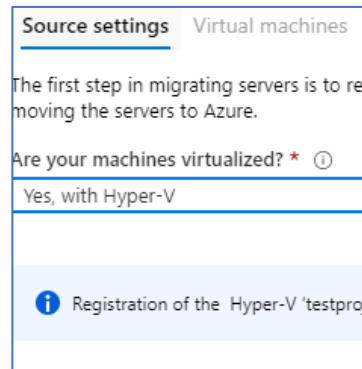
4.9 Replication

Replication vil bety prosessen med å kopiere filene VM-ene består av, over til Recovery Services Vaultet som Azure Migrate benytter. Dette er ikke å forveksle med faktisk migrering. Replication kan forklares som kopiering av data fra en maskin til en annen, blant annet med den hensikt å bruke begge maskiner. Migrering er en prosess som kun kan skje en gang, hvor målet er at destinasjonsmaskinen tar over videre drift. Dataen i dette tilfellet blir svært ofte utsatt for transformasjon, og derfor er prosessen ikke reversibel.

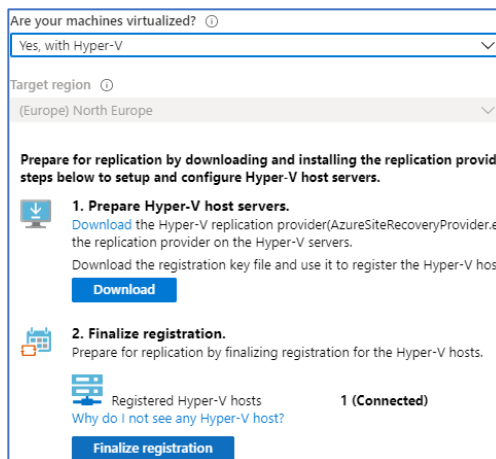
Velger *replicate*



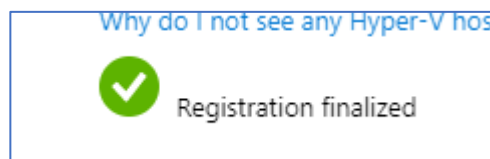
Spesifiserer at maskinene er virtualiserte på Hyper-V



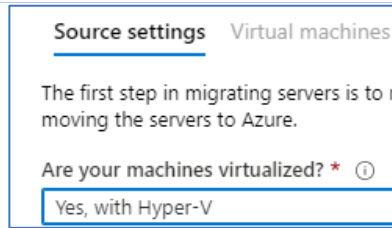
Går tilbake, velger *Finalize registration*



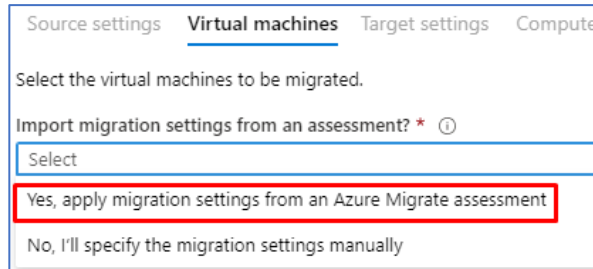
Ser at registreringen er ferdig og Azure fant Hyper-V hosten



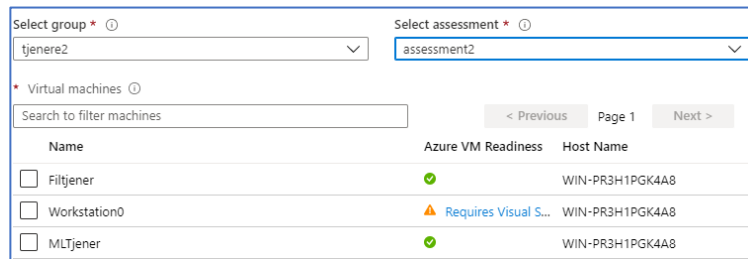
Tilbake i replicate spesifiseres det at maskinene er virtualisert på en Hyper-V host



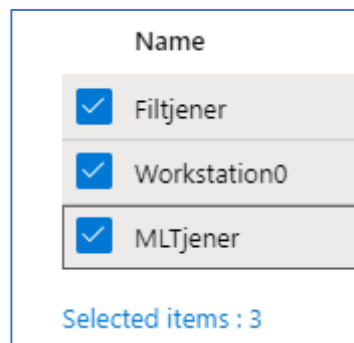
Siden det ble kjørt en valgfri assessment kan den nå brukes til å migrere. Da vil de foreslåtte størrelsene på VM-ene tas i bruk



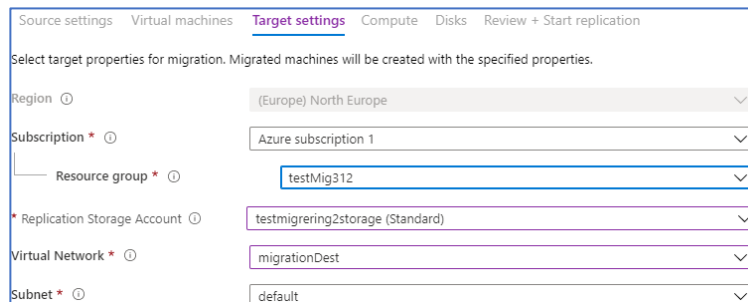
Påser deretter at riktig gruppe og assessment brukes



Her velges alle tre servere, selv om workstation0 har et problem knyttet til Azure VM Readiness



Velger ressursgruppen og det nettverk som serverene skal migreres til.



Endrer ingenting på compute, men slik foreslår assessmentet at det skal bli

Azure VM Size

Standard_A2_v2 (2 Cores, 4 GB RAM)	▼
Standard_A1_v2 (1 Cores, 2 GB RAM)	▼
Standard_A2_v2 (2 Cores, 4 GB RAM)	▼

Derimot må os og disk velges

OS Type	OS Disk
Windows ▼	Filtjener ▼
Windows ▼	Workstation1 ▼
Windows ▼	ML-Tjener ▼

Det er nå klart for replikasjon

Virtual machines selected : 3

Target settings

Subscription	Azure subscription 1
Resource group	testMig312
Region	North Europe
Virtual Network	migrationDest
Azure Hybrid Benefit	Do not apply

Når initiell replikasjon er ferdig, kan en test-migrering og eventuell fullstendig migrering utføres

>>> **Next steps:** After initial replication completes

Replicate « Previous: Disks

Ved å gå inn på *Site Recovery Jobs* ser man at replication er i gang med å starte

Home > Site Recovery jobs

Site Recovery jobs

testproject27-03-20-MigrateVault-k8ftt2n7

Filter Export jobs

Filter items...

Name	↑↓	Status
Enable replication		🔄 In progress
Enable replication		🔄 In progress
Enable replication		🔄 In progress

Utfallet i dette tilfellet

✔ Successful	Protected item	MLTjener
✔ Successful	Protected item	Workstation0
✘ Failed	Protected item	Filtjener

Synkroniseringen av maskiner har nå begynt, men denne prosessen tar lang tid

Name	Status
MLTjener	1% synchronized
Filtjener	Protection couldn't be enabled
Workstation0	0% synchronized

Etter en god stund kan man se at statusen endres til *protected*. Nå er det mulig å foreta en testmigrering.

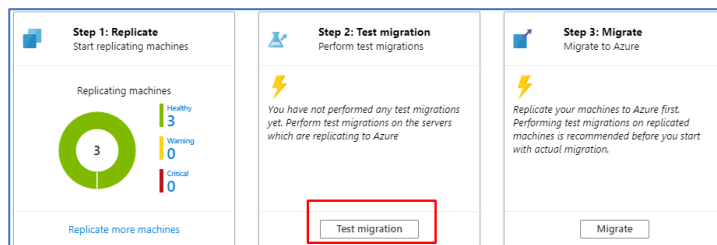
MLTjener	Protected	✓ Healthy
Filtjener	Protection couldn't b...	❗ Critical
Workstation0	Protected	✓ Healthy

Tabell 23 - Replication

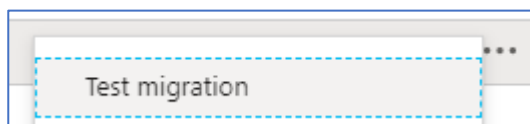
4.10 Test migration

Før en full migrering er det hensiktsmessig å gjennomføre en test migration for å forhindre at uventede problemer oppstår. Når man kjører en test migration sjekker dette at migreringen går som forventet uten å påvirke on-premises miljøet som vil fungere som normalt og fortsette replikasjonen.

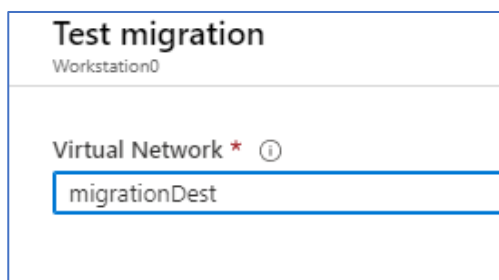
Neste steg i prosessen er å teste migreringen.



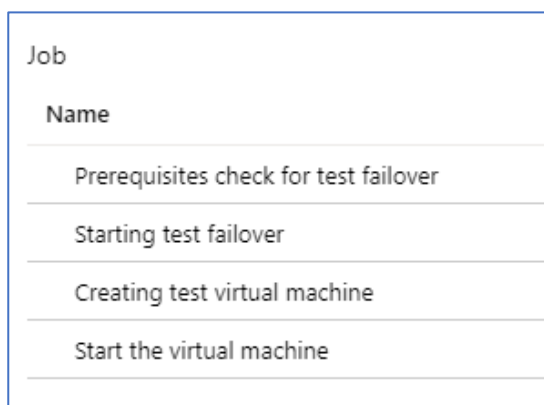
Velger *Test Migration* på en enkelt vm, i dette tilfellet «Workstation0»



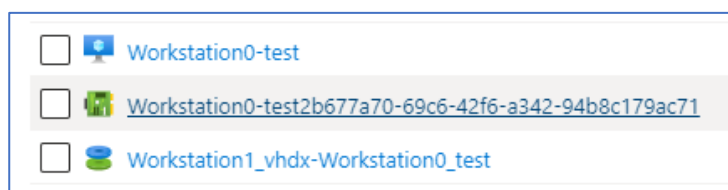
Velger det virtuelle nettverket som skal benyttes for testmigreringen



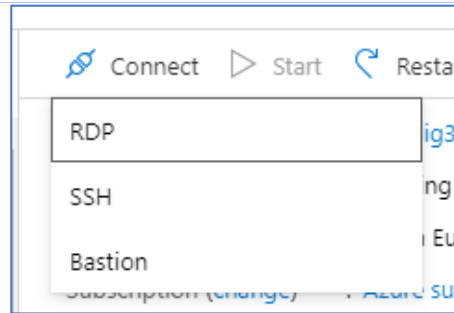
Azure vil da gå gjennom en sjekkliste



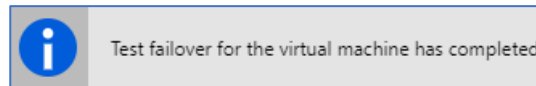
Når sjekken er ferdig kan man finne VMen som ble migrert over til Azure i ressursgruppen



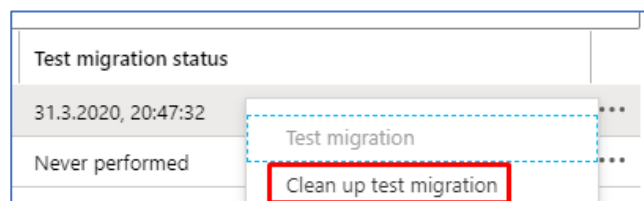
Denne VMen kan brukes som en vanlig vm, men har bare en privat ip.



Ser nå at prosessen er ferdig



Man blir så bedt om å *clean up test migration* etter migrering



Cleanup bli iverksatt, status på replikasjon endres. Det er fremdeles 2 jobs in progress

Step 1: Replicate
Start replicating machines

Replicating machines

3

Healthy 3

Warning 0

Critical 0

[Replicate more machines](#)

Step 2: Test migration
Perform test migrations

Test migration pending	0	⚠
Test migration in progress	0	🔄
Cleanup pending	0	ℹ
Test migration done	0	✅

[Perform more test migrations](#)

Jobs [View all](#)

Failed	1	❌
In progress	2	🔄

Attention required

✅ No issues found.

Etter en stund kan man se at testmigreringen er gjennomført for de to vm'ene

Step 2: Test migration

Perform test migrations

Test migration pending	0	
Test migration in progress	0	
Cleanup pending	0	
Test migration done	2	

[Perform more test migrations](#)

Migreringen kan nå utføres

Step 1: Replicate

Start replicating machines

Replicating machines

3

Healthy
3

Warning
0

Critical
0

[Replicate more machines](#)

Step 2: Test migration

Perform test migrations

Test migration pending	0	
Test migration in progress	0	
Cleanup pending	0	
Test migration done	2	

[Perform more test migrations](#)

Step 3: Migrate

Migrate to Azure

Replicate your machines to Azure first. Performing test migrations on replicated machines is recommended before you start with actual migration.

Migrate

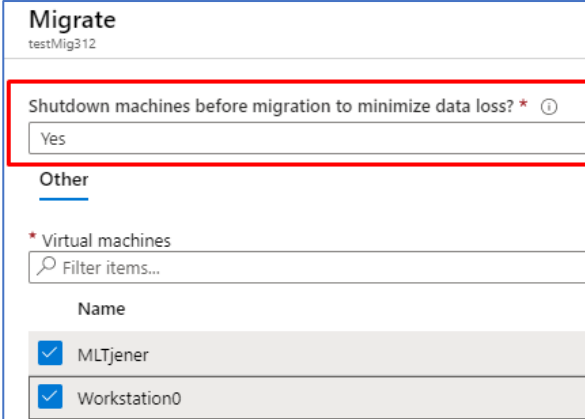
Tabell 24 - Test Migration

4.11 Migrering fra on-prem til azure

Det er som beskrevet tidligere er en rekke ressurser som må eller bør migreres over til Azure. Samtidig er det noen ressurser som ikke migreres, men som heller skapes på nytt.

4.11.1 Virtuelle Maskiner

Velger maskinene som skal migreres rett over, samt at de skrus av. Dette er for at det ikke skal være tap av data mellom on-prem og azure.



Migrate
testMig312

Shutdown machines before migration to minimize data loss? * ⓘ
Yes

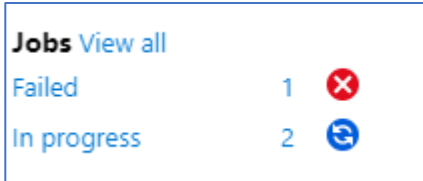
Other

* Virtual machines
Filter items...

Name
<input checked="" type="checkbox"/> MLTjener
<input checked="" type="checkbox"/> Workstation0

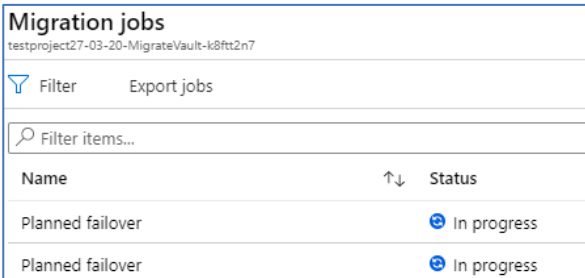
Klikker på *migrate* og to prosesser vil bli satt i gang. På figuren kan man se at en jobb har feilet, og at to er i progress.

Den feilede jobben på dette tidspunktet skyldtes at det en VM som ikke var skikket til migrering var valgt for migrasjon. Dette var en VM som hadde blitt kjørt sysprepp på, og derfor ikke skal startes igjen, men prosjekgruppen hadde feilaktig valgt den under migreringen. Den kan ses bort fra når det kommer til resten av migreringen.



Jobs	View all
Failed	1 ❌
In progress	2 🔄

Azure Migrate benytter seg av tjenesten Site Recovery. Dette er en tjeneste for å rette opp feil og sikre oppetid i on-prem og Azure miljøer. Azure migrate tar i bruk planned failover, gjennom Site Recovery, for å utføre migreringsjobben.



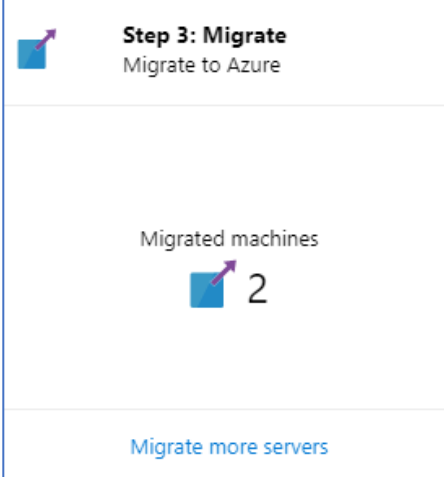
Migration jobs
testproject27-03-20-MigrateVault-k8ft2n7

Filter Export jobs

Filter items...

Name	↑↓	Status
Planned failover		🔄 In progress
Planned failover		🔄 In progress

Det er nå blitt migrert over 2 maskiner til Azure














Step 3: Migrate
Migrate to Azure

Migrated machines
2

Migrate more servers

I ressursgruppen igjen kan man se at maskinene er migrert over med samme navn som i Hyper-V

<input type="checkbox"/>	 ML_Tjener_vhdx-MLTjener
<input type="checkbox"/>	 MLTjener
<input type="checkbox"/>	 MLTjenera66c6d01-b40b-43b4-b37c-cd1723b8082a
<input type="checkbox"/>	 onprehostb276kv
<input type="checkbox"/>	 sqltestmigration2
<input type="checkbox"/>	 testDB (sqltestmigration2/testDB)
<input type="checkbox"/>	 testmigring2storage
<input type="checkbox"/>	 testproject27-03-20-MigrateVault-k8ftt2n7
<input type="checkbox"/>	 Workstation0
<input type="checkbox"/>	 Workstation09c4dce6b-ee50-4799-a932-96d067fa9467
<input type="checkbox"/>	 Workstation1_vhdx-Workstation0

Tabell 25 - Migrering Virtuelle Maskiner

4.11.2 SQL-Databaser

4.11.2.1 Valg av verktøy

I dette prosjektet ble programvaren *Microsoft SQL Server Migration Assistant for MySQL* benyttet til selve migreringen av MySQL, og for MSSQL ble *Data Migration Assistant* brukt. Alternativt kunne den helintegreerte tjenesten *Azure Database Migration Service* blitt benyttet. Grunnen til at dette ikke ble brukt her er fordi det er en tjeneste som egner seg mer for flytting av databaser i stor skala. For en bedrift på denne størrelsen og denne casen, var de valgte verktøyene hensiktsmessige, særlig siden utrulling av *main.json* og linked-templates gjør oppsettet i skyen delvis automatisk.

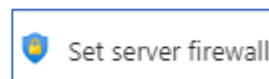
4.11.2.2 Tillate tilkobling til databasen

Dette gjelder for alle typer databaser som tas opp i dette prosjektet. For at serveren det skal migreres ifra skal kunne koble til databasen på Azure, må brannmuren åpnes. Slik gjøres dette:

Gå inn på databasen (her SQL database)



Fra *Overview*, velg *Set server firewall*



Legger til en regel, her er det simpeltent å tillate Ip-adressen som migreringen skal skje fra



Tabell 26 - Tilkobling til database

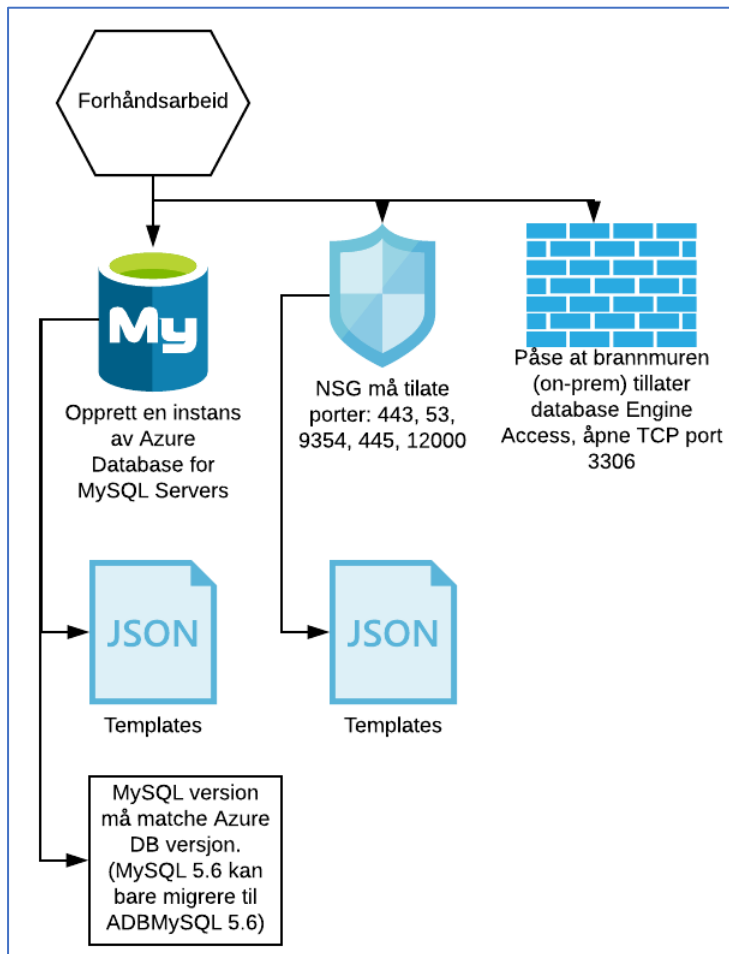
Merk: Dette tar forbehold om at tilkobling er tillatt fra andre nivå også. Dersom trafikken blir stoppet andre steder, er det ikke nok å «kun» legge til klient-IP-en her.

4.11.2.3 Assessment

Selv om de ligger på VM-er, må det kjøres egen assessment på databasene på de gitte VM-ene. Dette blir gjort gjennom de respektive migreringsprogrammene. Det er anbefalt å kjøre vurderinger (assessments) før migreringen, slik at man får så komplett bilde over sluttresultatet av migreringen som overhodet mulig.

4.11.2.4 Migrere MySQL

Figuren viser på et generelt grunnlag forhåndsarbeidet som må gjøres på Azure:



Figur 27 - Oversikt migrering MySQL

Man må:

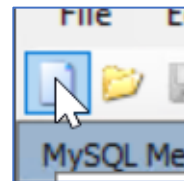
- Opprette en Azure Database for MySQL Servers. Versjonen av MySQL on-premises må matche Azure DB versjonen. Denne databasen blir utrullet som del av main.json.
- Påse at NSG-en (Network security Group) som Databasene blir opprettet i tillater trafikken. Samtidig må det påses at brannmuren on-premises gjør det samme. På Azure-siden blir dette også fikset med main.json

4.11.2.4.1 Fremgangsmåte

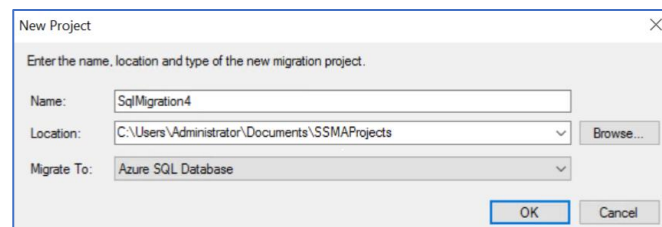
Åpne *Microsoft SQL Server Migration Assistant for MySQL*



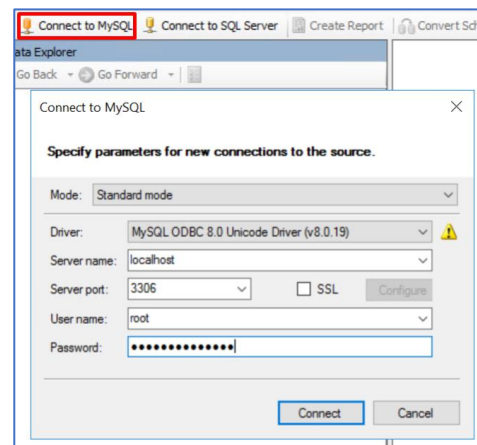
Velger *new project*



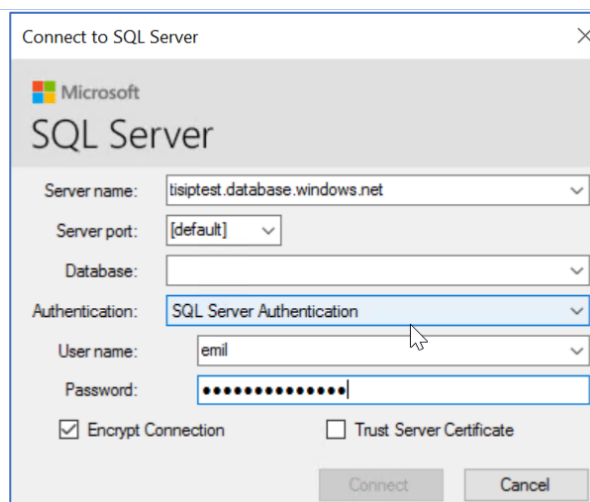
Påse at prosjektet stemmer overens med destinasjonen som det skal migreres til. Her *Azure SQL Database*



Videre må MySQL databasen som er på den lokale serveren kobles til, ved hjelp av *Connect to MySQL*



Kobler opp SQL-Serveren til Azure og dersom brannmurs-oppsettet som beskrevet på 4.11.2.4.1⁴¹ er gjort, blir tilkoblingen vellykket. Her blir *SQL Server authentication* valgt. Dette er fordi SQL serveren ble satt opp med dette under installasjonen, men det hadde ikke vært noe i veien med å ha satt det opp med Windows Autentisering i stedet.



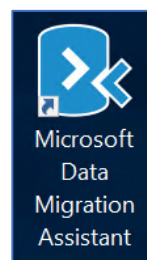
Tabell 27 - Migrere SQLdb

4.11.2.5 Migrere MSSQL DB

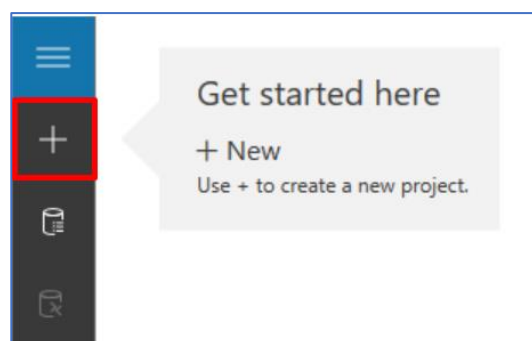
Framgangsmåte for å migrere MSSQL DB er tilnærmet likt framgangsmåten for MySQL. Det samme gjelder forhåndsarbeidet som må gjøres på Azure.

4.11.2.5.1 Fremgangsmåte

Installer og åpne programmet *Microsoft Data Migration Assistant*

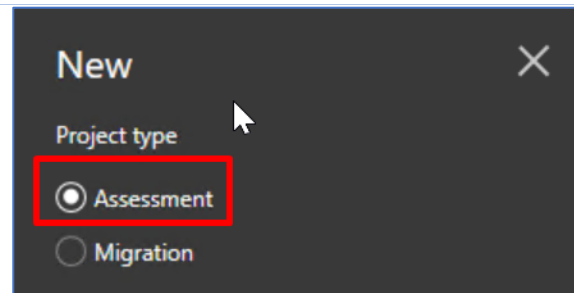


Lag et nytt prosjekt



⁴¹ Tillate tilkobling til databasen

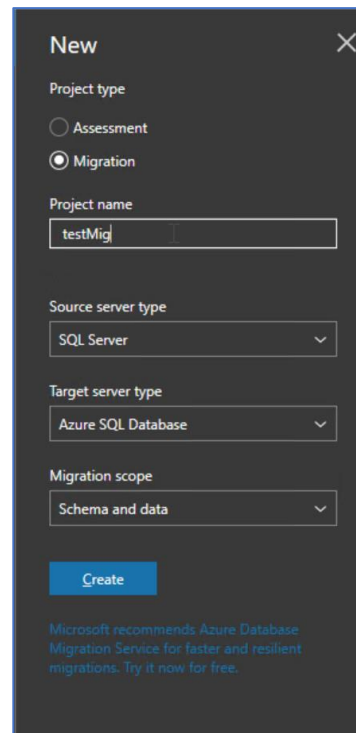
Dersom ønskelig, start en assessment før migreringen ved å velge *assessment* i stedet for *migration*.



The screenshot shows a dark-themed 'New' dialog box with a close button (X) in the top right corner. Under the heading 'Project type', there are two radio button options: 'Assessment' and 'Migration'. The 'Assessment' option is selected, and its label is highlighted with a red rectangular box. A mouse cursor is positioned over the 'Assessment' label.

Velger *Migration*, og påser at:

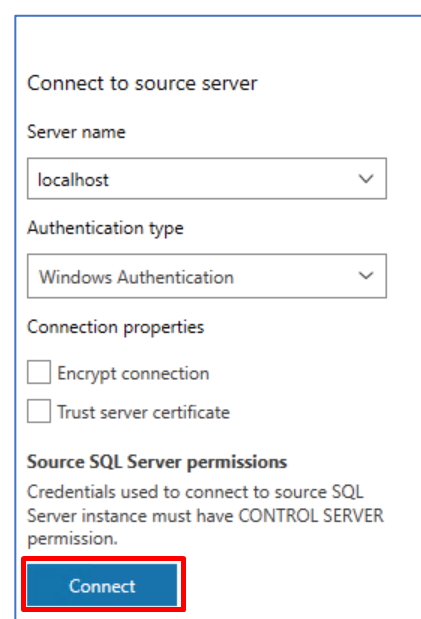
- *Target Server type* = Azure SQL Database
- *Migraton scope* = Schema and data



The screenshot shows a dark-themed 'New' dialog box with a close button (X) in the top right corner. Under the heading 'Project type', there are two radio button options: 'Assessment' and 'Migration'. The 'Migration' option is selected. Below this, there are several input fields: 'Project name' with the text 'testMig', 'Source server type' with a dropdown menu showing 'SQL Server', 'Target server type' with a dropdown menu showing 'Azure SQL Database', and 'Migration scope' with a dropdown menu showing 'Schema and data'. At the bottom, there is a blue 'Create' button. Below the button, there is a small text note: 'Microsoft recommends Azure Database Migration Service for faster and resilient migrations. Try it now for free.'

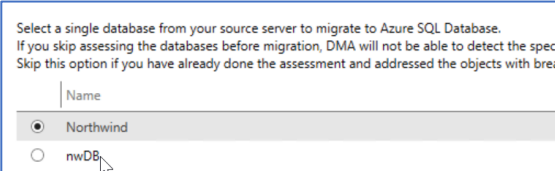
Kobler til kilde-server. I dette tilfellet simpeltent localhost (siden den er lokal på maskinen som migreringen foregår fra). Eventuelt så kunne en ekstern kilde blitt brukt.

Velger også riktig form for autentisering, her *Windows Authentication*.



The screenshot shows a light-themed 'Connect to source server' dialog box. It contains several sections: 'Server name' with a dropdown menu showing 'localhost', 'Authentication type' with a dropdown menu showing 'Windows Authentication', and 'Connection properties' with two unchecked checkboxes: 'Encrypt connection' and 'Trust server certificate'. Below these is a section titled 'Source SQL Server permissions' with the text: 'Credentials used to connect to source SQL Server instance must have CONTROL SERVER permission.' At the bottom, there is a blue 'Connect' button highlighted with a red rectangular box.

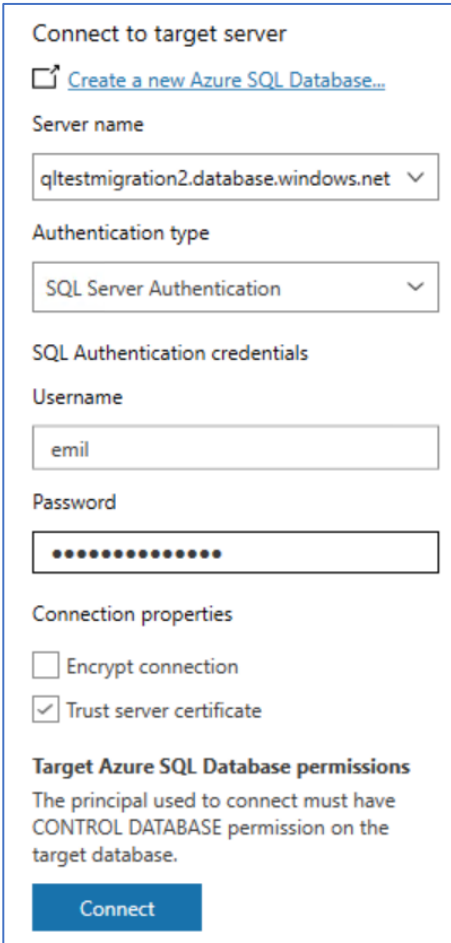
Velger databasen å migrere fra. Dette blir test-databasen som ble opprettet tidligere, som vist i 3.3.4.1.2.2⁴²



Select a single database from your source server to migrate to Azure SQL Database. If you skip assessing the databases before migration, DMA will not be able to detect the spec. Skip this option if you have already done the assessment and addressed the objects with bre

Name
<input checked="" type="radio"/> Northwind
<input type="radio"/> nwDB

Kobler til SQL Databasen på Azure, ved hjelp av servernavnet. Eks: sqltestmigration2.database.windows.net Her blir det brukt *SQL Server Authentication*, siden Azure SQL Database ikke støtter *Windows Authentication*.



Connect to target server

[Create a new Azure SQL Database...](#)

Server name
sqltestmigration2.database.windows.net

Authentication type
SQL Server Authentication

SQL Authentication credentials

Username
emil

Password
.....

Connection properties

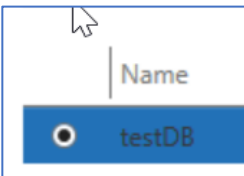
Encrypt connection

Trust server certificate

Target Azure SQL Database permissions
The principal used to connect must have CONTROL DATABASE permission on the target database.

Connect

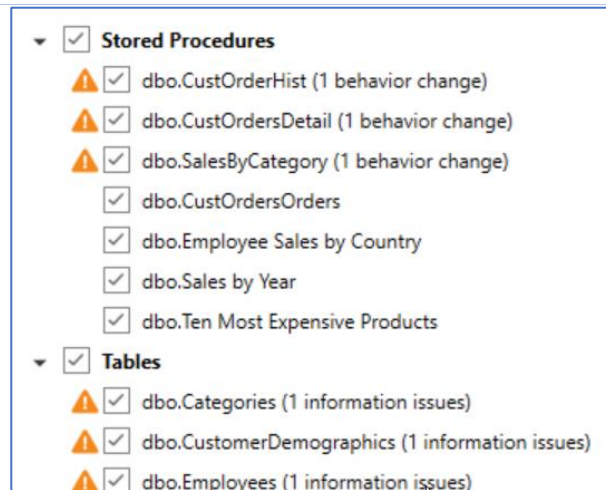
Velger «testDB» databasen på SQL-databasen



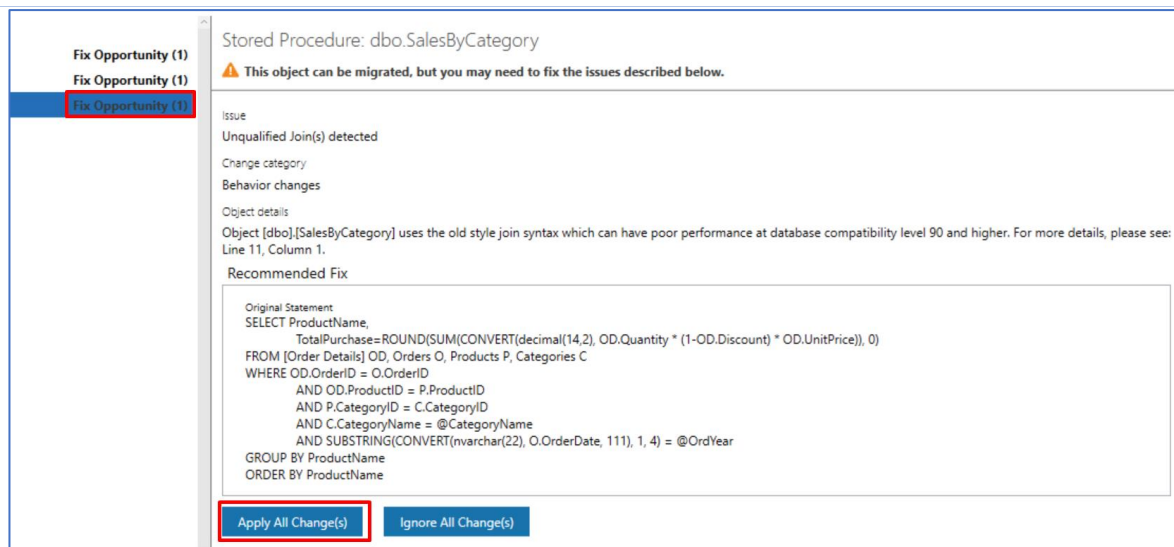
Name
<input checked="" type="radio"/> testDB

⁴² Test-database for MSSQL

Velger hvilke objekter som skal migreres (i dette tilfellet tilnærmet alt)

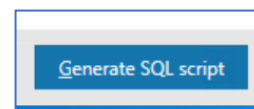


Her ser man derimot at det er noen advarsler (informasjon issues) som vises. Programmet foreslår ofte løsninger på dette. Hvis objektet har *Fix Opportunity* ved siden av seg, kan man trykke på den for å iverksette foreslått fiks.



Programmet foreslår at join-spørringen i tabellen burde oppdateres på grunn av ytelse, dette er noe som er veldig greit å endre, så i dette eksempel blir det gjort.

Generate SQL skript er da det neste steget. Dette skaper et skript for å sette opp database-skjemaet for den kommende databasen



Slik ser det ut. I teorien kunne dette faktisk kunne blitt kjørt på hvilken som helst SQL-database manuelt.


```

Recommendation: Deprecated data types are marked to be discontinued on next versions of SQL Server, should use new data types such
More information: ntext, text, and image (Transact-SQL) (https://go.microsoft.com/fwlink/?LinkId=798558)
**/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
IF NOT EXISTS (SELECT * FROM sys.objects WHERE object_id = OBJECT_ID(N'[dbo].[Categories]') AND type in (N'U'))
BEGIN
CREATE TABLE [dbo].[Categories](
[CategoryID] [int] IDENTITY(1,1) NOT NULL,
[CategoryName] [nvarchar](15) COLLATE Danish_Norwegian_CI_AS NOT NULL,
[Description] [ntext] COLLATE Danish_Norwegian_CI_AS NULL,
[Picture] [image] NULL,
CONSTRAINT [PK_Categories] PRIMARY KEY CLUSTERED

```

Men dette inneholder derimot kun database-skjemaet, den faktiske dataen blir overført etterpå.

Deploy Schema kjører skriptet på Azure-databasen

Deploy schema

I dette tilfellet rulles skjemaet ut helt uten problem

Executed 205 of 205: IF EXISTS (SELECT Command executed successfully.

Dersom man da velger *Migrate Data* begynner overførselen av data.

Migrate data

Velger tabellene som skal overføres. I dette prosjektet vil det være alle. *CustomerCustomerDemo* og *CustomerDemographics* er eksempler på tabeller.

[dbo].[CustomerCustomerDemo]
 [dbo].[CustomerDemographics]

Når alt er sjekket og klart, så startes dataoverførselen ved å klikke på *Start data migration*

Start data migration

På den neste siden vises oversikt over migreringen, og hvorvidt det lyktes eller ikke. I dette eksemplet ble to tabeller overført, som begge lyktes.

1 Select source		2 Select target							
2	Server objects	0	In-progress	2	Successful	0	Warnings	0	Failed
▼ Tables (2)									
Status	Table name	Migration details							
✓	[dbo].[CustomerCustomerDemo]	Migration successful. Duration: 0 hrs 0							
✓	[dbo].[CustomerDemographics]	Migration successful. Duration: 0 hrs 0							

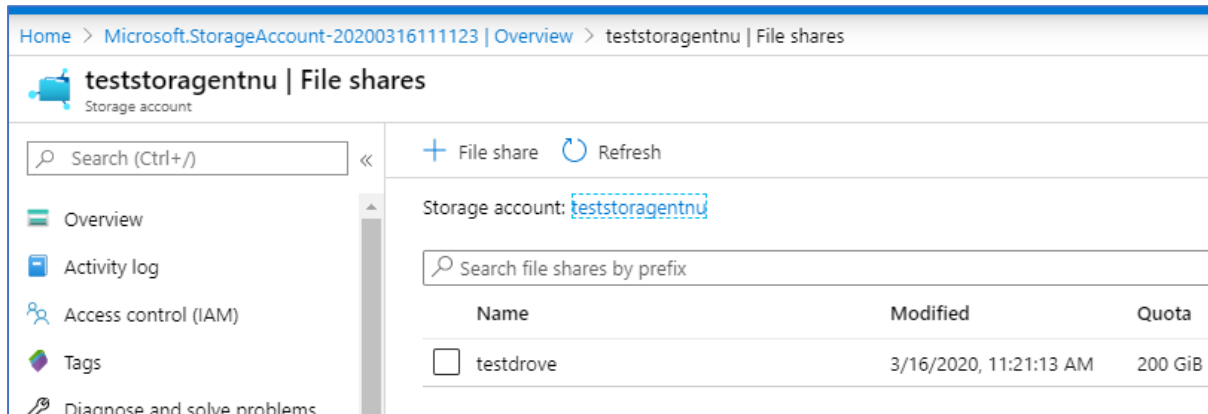
Tabell 28 - Migrere MSSQL database

4.11.3 Migrere filserver over til Azure

4.11.3.1 Opprett Storage Account

Først må det opprettes en Storage Account, som skal ha dataen.

Dette gjøres enkelt ved hjelp av templatene `storageAccountFileShare.json`, som beksrevet i 4.5⁴³. Det er ingen særskilte behov for hvordan denne lages, men det blir tatt i bruk Data Lake Storage, i tillegg til at det blir opprettet en Konteiner med SMB file share samtidig.



4.11.3.2 AzCopy

AzCopy må lastes ned og kjøres på filserveren som filene skal tas fra. I dette tilfellet er VM-en «filserver» på Hyper-V serveren.

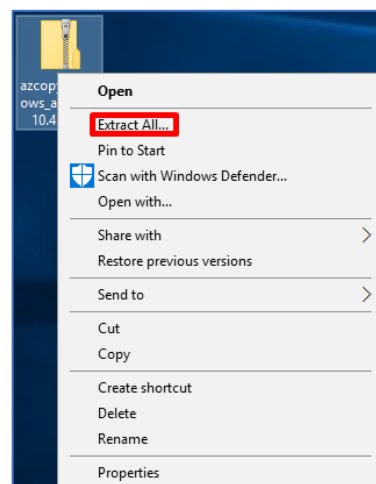
Last ned AzCopy v10 fra <https://docs.microsoft.com/en-us/azure/storage/common/storage-use-azcopy-v10>

Download AzCopy

First, download the AzCopy V10 executable file nothing to install.

- Windows 64-bit (zip)
- Windows 32-bit (zip)
- Linux (tar)
- MacOS (zip)

Denne blir lastet ned som ett .zip arkiv som må pakkes ut



⁴³ Storage Account i Azure

Åpne cmd, og endre filbanen til inne i mappen

```
C:\Users\Administrator\Desktop\azcopy_windows_amd64_10.4.0 (1)\azcopy_windows_amd64_10.4.0>
```

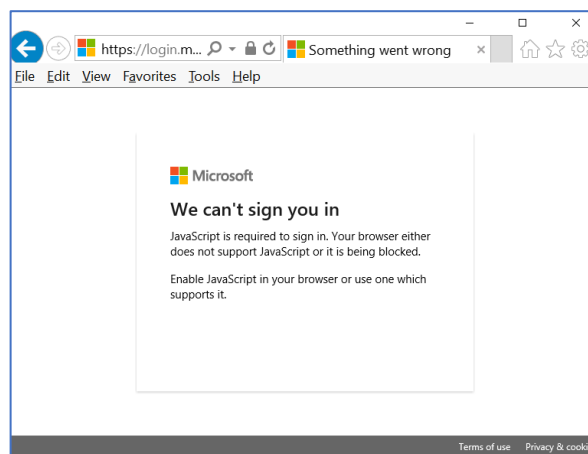
For å gå videre må enn bruker med rollen *storage Blob Data Contributor* vær tilgjengelig for bruk. I dette tilfellet foreligger det en bruker *storageuser* som brukes for dette formål. Fremdeles i CMD, brukes *azcopy login*

azcopy login

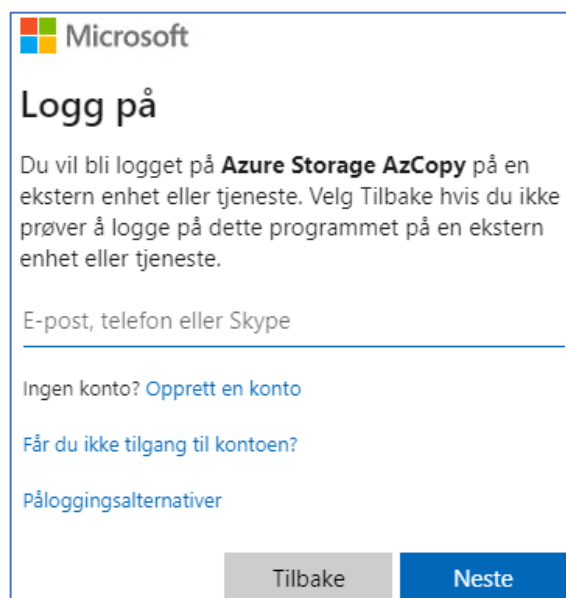
Etter å ha kjørt kommandoen blir det generert en tok som skal brukes for å koble azcopy opp mot Azure-kontoen, gjennom nettstedet

<https://microsoft.com/devicelogin>

(Skjermbildet viser at det kommer feilmelding dersom man prøver å åpne i Internet explorer. Ingen problemer med Firefox eller Chrome.)



Etter å ha skrevet inn verifiseringskoden, må det logges inn med brukeren.



Etter pålogging kan man gå tilbake til CMD igjen.



Azure Storage AzCopy

Du har logget på programmet Azure Storage AzCopy på enheten. Du kan nå lukke dette vinduet.

Kjører følgende kode for å kopiere angitt mappe over til storage-accounten i Azure

```
azcopy copy '<local-file-path>' 'https://<storage-account-name>.file.core.windows.net/<file-share-name>/<file-name><SAS-token>'
```

For å autentisere en overførsel er det nødvendig å generere en sas-token på storage accounten. Kjør følgende kode i PowerShell

```
Install-Module PowerShellGet -Repository PSGallery -Force  
Connect-AzAccount
```

Bruker fremgangsmåten på 4.5.1.2⁴⁴ for å kjøre klart en SAS-token, men tilpasser det for denne delen av løsningen.

Når dette er gjort er en tilfeldig streng lagd, som gir tilgang til lagringsstedet

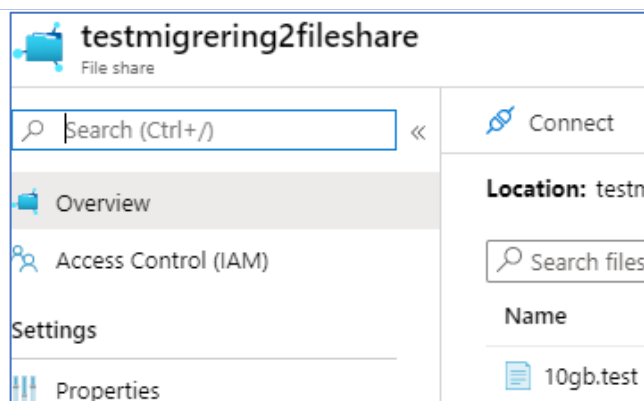
```
PS C:\Users\endre> azcopy C:\Testdata\10Gb.test  
>> "https://testmigrering2storage.file.core.windows.  
>> /testmigrering2fileshare ---DIN STRING HER ---"
```

Når denne blir ferdig, vil dette vises.
I dette tilfellet ble en test-fil på 10 gb lastet opp

```
INFO: Scanning...  
INFO: Any empty folders will not be processed, because  
  
Job 13349974-a666-8847-7fc1-56fcb6ce43bf has started  
Log file is located at: C:\Users\Administrator\.azcopy\  
  
97.8 %, 0 Done, 0 Failed, 1 Pending, 0 Skipped, 1 Total  
  
Job 13349974-a666-8847-7fc1-56fcb6ce43bf summary  
Elapsed Time (Minutes): 0.8391  
Number of File Transfers: 1  
Number of Folder Property Transfers: 0  
Total Number of Transfers: 1  
Number of Transfers Completed: 1  
Number of Transfers Failed: 0  
Number of Transfers Skipped: 0  
TotalBytesTransferred: 10737418240  
Final Job Status: Completed
```

⁴⁴ Shared Access Signatures for bruk med Templates

Går deretter inn i Azure for å se at den er på plass.

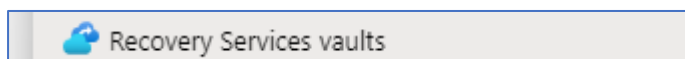


Tabell 29 - AzCopy

AzCopy kan også brukes til å laste opp filer basert på når de er modifisert sist, og ved hjelp av å automatisere kjøring av oppgaver. Dette er en mulig nyttig måte å synkronisere on-premises lagring med lagring i Azure, som i et hybrid miljø.

4.11.3.3 Backup i Azure.

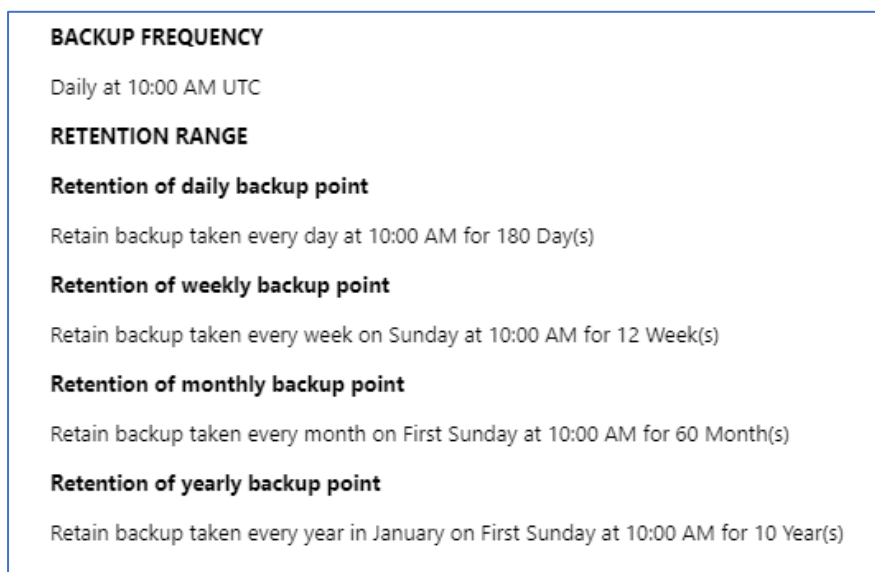
Ethvert system må et system for sikkerhetskopiering. I Azure er dette i stor grad innebygd og aktivert allerede, men her aktiveres backup ved hjelp av *recovery services Vault*.



Figur 28 - Recovery vaults

Dette opprettes med templatene *recoveryVault.json*⁴⁵

Deretter må backup-polisier defineres. Her er backup-polisier som defineres for VM-er brukt i WVD. Dette blir opprettet med templatene *customBackupVM.json*⁴⁶. Utsnittet under viser parameterne på en slik policy.



Figur 29 - Backup policy

⁴⁵ (Brasø & Aalrust, recoveryVault.json, 2020)

⁴⁶ (Brasø & Aalrust, customBackupPolicyVM.json, 2020)

I tillegg er det nødvendig med en policy for LTR - Long Term Retention når det gjelder SQL-databasene. Her er det ikke en enkel ARM-template som rulles ut, men et PowerShell-skript gjør nytten. Skriptet *retentionPolicy.ps1*⁴⁷ inneholder eksempler på dette, men for dette tilfeller er følgende linjer hensiktsmessige:

```
Set-AzSqlDatabaseBackupLongTermRetentionPolicy -ServerName $serverName -DatabaseName $dbName `
`
-ResourceGroupName $resourceGroup -WeeklyRetention P12W -YearlyRetention P5Y -
WeekOfYear 16
```

Dette skriptet vil da brukes til å sette opp *weekly retention* til 12 uker, årlig til fem år. Dette er en type lagring som kommer på siden av 35-dagers backup policy som ligger som standard for alle SQL-databaser. Dette er særlig nyttig for databaser med særskilte krav til langtidslagring.

Dersom det senere er nødvendig å rulle tilbake kan følgende skript brukes:

1. Sjekk hvilke backups som fins for databasen ved hjelp av *viewLTRBackups.ps1*⁴⁸

```
# backups are grouped by the logical database id, within each group they are ordered by the
timestamp, the earliest backup first
```

```
$ltrBackups = Get-AzSqlDatabaseLongTermRetentionBackup -Location $server.Location
```

2. gjenopprett valgt backup med *rollbackToBackup.ps1*⁴⁹
restore a specific LTR backup as an P1 database on the server \$serverName of the resource group \$resourceGroup

```
Restore-AzSqlDatabase -FromLongTermRetentionBackup -ResourceId $ltrBackup.ResourceId -
ServerName $serverName -ResourceGroupName $resourceGroup `
```

```
-TargetDatabaseName $dbName -ServiceObjectiveName P1
```

⁴⁷ (Brasø & Aalrust, retentionPolicy.ps1, 2020)

⁴⁸ (Brasø & Aalrust, viewLTRBackups.ps1, 2020)

⁴⁹ (Brasø & Aalrust, rollbackToBackup.ps1, 2020)

4.11.4 Potensielle feil og løsninger

4.11.4.1 VM-en har ikke en ekstern IP-adresse

Ofta har VM-er fra et on-premises miljø ingen *ekstern IP-adresse* knyttet til seg. I dette testmiljøet er det behov for å kunne nå VM-en fra utsiden, så derfor skal det legges til offentlig IP-adresse.

Azure Spot	: N/A
Public IP address	: -
Private IP address	: 10.1.0.4
Public IP address (IPv6)	: -
Private IP address (IPv6)	: -
Virtual network/subnet	: migrationDest/default
DNS name	: -

Figur 30 - Ingen offentlig ip-adresse

Dette fikses ved hjelp av en template kalt *addPublicIPtoNIC.json*⁵⁰ som oppretter den nødvendige ip-adressen og kobler den til nettverket til vmen.

Detaljert fremgangsmåte:

Finn fram nødvendige parametre med powershell. Dette er ressurs-iden til det virtuelle nettverket og navnet til nettverkskortet.

Først finnes fram ressurs-id til korrekt virtuelt nettverk

```
PS C:\Users\Emil.EMIL-HP> Get-AzVirtualNetwork -Name MigrationDest_

Name                : migrationDest
ResourceGroupName   : testMig312
Location             : northeurope
Id                  : /subscriptions/[REDACTED]/resourceGroups/testMig312/virtualNetworks/migrationDest
Etag                 : W/"a315e952-bca8-4eb9-8c8f-a447234495ff"
ResourceGuid        : 9aa3a57c-fbe7-42a2-ab8f-5b10e32e2fe7
ProvisioningState    : Succeeded
Tags                 :
AddressSpace        : {
                        "AddressPrefixes": [
                          "10.1.0.0/16"
                        ]
                      }
```

Ressursiden til VM-en (*ResourceID*)

```
PS C:\Users\Emil.EMIL-HP> Get-AzResource -Name MLTjener

Name                : MLTjener
ResourceGroupName   : testMig312
ResourceType        : Microsoft.Compute/virtualMachines
Location            : northeurope
ResourceId          : /subscriptions/[REDACTED]/resourceGroups/testMig312/providers/Microsoft.Compute/virtualMachines/MLTjener
```

⁵⁰ (Brasø & Aalrust, addPublicIPtoNIC.json, 2020)

Nettverkskortet til de virtuelle maskinene

```
PS C:\Users\Emil.EMIL-HP> Get-AzVM

ResourceGroupName Name Location VmSize OsType NIC ProvisioningState Zone
-----
TESTMIG312 MLTjener northeurope Standard_A2_v2 Windows MLTjenera66c6d01-b40b-43b4-b37c-cd1723b8082a Updating
TESTMIG312 Workstation0 northeurope Standard_A1_v2 Windows Workstation09c4dce6b-ee50-4799-a932-96d067fa9467 Succeeded
```

Før endringen gjøres må VM-en stoppes

```
PS C:\Users\Emil.EMIL-HP> Stop-AzVM
```

Deretter kjøres ut template `addPublicIPtoNic.json` med tilhørende parameter-fil `addPublicIPtoNicparams.json`

```
New-AzResourceGroupDeployment `
-TemplateFile "C:\Users\Emil.EMIL-HP\Documents\GitHub\Bachelor2020\JSON\legg til NIC\addPublicIPNic.json" `
-TemplateParameterFile "C:\Users\Emil.EMIL-HP\Documents\GitHub\Bachelor2020\JSON\legg til NIC\addPublicIPNicParams.json"
```

```
cmdlet New-AzResourceGroupDeployment at command pipeline position 1
Supply values for the following parameters:
(Type !? for Help.)
ResourceGroupName: testMig312
```

Sporer det i Azure-portalen

<input type="checkbox"/> addPublicIPtoNic	✔ Succeeded	4/2/2020, 11:55:47 AM
---	--	-----------------------

Og på VM-en

```
Public IP address : 137.135.187.188
```

Starter VM-en igjen

```
PS C:\Users\Emil.EMIL-HP> Start-AzVM

cmdlet Start-AzVM at command pipeline position 1
Supply values for the following parameters:
(Type !? for Help.)
Name: MLTjener
ResourceGroupName: testMig312
```

Tabell 30 - Ekstern ip-adresse for VM

For at tilkoblingen skal fungerer skikkelig må det i tillegg settes riktige brannmurs-regler på VM-ene. Dette fikses ved å sette en NSG på subnett. Dette blir den samme NSG-en som settes opp tidligere i dokumentet.


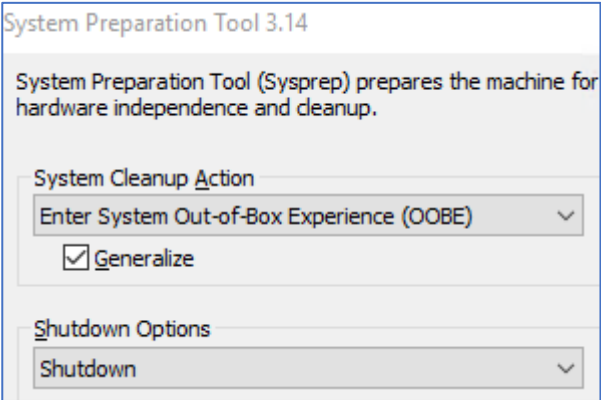
Alternativ kunne det ha bli tatt i bruk en VPN gateway for å kunne ta i bruk det interne 10.1.0.4 nettverket som VM-en er på. Siden denne løsningen skal ta i bruk WVD og autentisering gjennom azure AD, så er det ikke nødvendig.

4.12 Gjør klart et VM-image

For at man skal kunne opprette VM-er basert på en allerede eksisterende VM er det veldig greit å opprette et VM-image. Dette kan enkelt benyttes til å opprette nye VM-er med særskilte behov, for eksempel så kan man slippe å gjøre mye tilpasning av den nye VM-en, dersom det foreligger et allerede oppsatt image.

4.12.1.1 Generaliser VM-en

Dette gjøres for å rense personlig og sikkerhetsmessig informasjon fra VM-en. Etter at en VM er generalisert så kan ikke VM-en startes igjen. Så dette er en endelig prosess som gjøres på VM-er hvor originalen ikke er tiltenkt brukt senere. Dersom originalen må brukes, er løsningen å kloner hele VM-en for deretter å generalisere kopien.

Gå inn på VM-en (via RDP e.l.)	
Åpne cmd og kjør kommandoen	<pre>C:\Users\Administrator>cd %windir%\system32\sysprep</pre>
Etterfulgt av <i>sysprep.exe</i>	<pre>C:\Windows\System32\Sysprep>sysprep.exe</pre>
Velger <i>OOBE – generalize – Shutdown</i> . Etter en stund vil VM-en skrus av. Den skal ikke skrus på eller brukes videre.	

Tabell 31 - Klargjøring VM-image

4.12.1.2 Lag imaget ved hjelp Powershell

Kjør skriptet *createImageFromVM.ps1*⁵¹

Følgende figur er et utsnitt av det skriptet. Variablene *vmName*, *rgName*, *location* og *imageName* må endres etter behov.

```
# Skript som benyttes til å lage et image av en VM
# Her benyttes AZ-modulen

$vmName = "MLTjener"
$rgName = "testmigration"
$location = "northeurope"
$imageName = "testImage"

Stop-AzVM -ResourceGroupName $rgName -Name $vmName -Force
Set-AzVM -ResourceGroupName $rgName -Name $vmName -Generalized

$vm = Get-AzVM -Name $vmName -ResourceGroupName $rgName
$image = New-AzImageConfig -Location $location -SourceVirtualMachineId $vm.Id

New-AzImage -Image $image -ImageName $imageName -ResourceGroupName $rgName
```

Figur 31 - Utsnitt fra skript

Etter en stund så er dette ferdig. Da kan imaget ses i ressursgruppen:

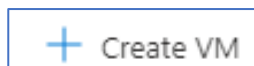


Figur 32 - Image i ressursgruppe

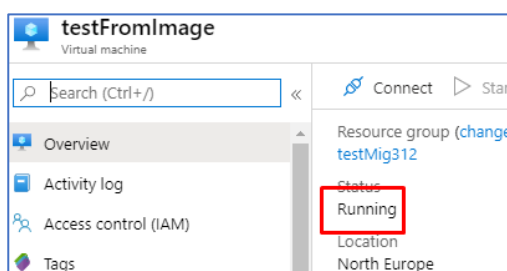
Før neste steg er det nødvendig å teste at det går an å lage en fungerende VM ut av imaget.

4.12.1.2.1 Tester at Imaget fungerer

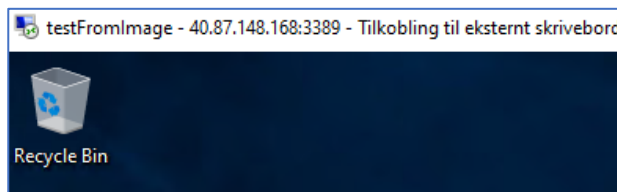
Gå inn på selve imaget, velg *Create VM*



Fyller inn alle nødvendige parametere og ser etter en stund at VM-en med navnet *testFromImage* er opprettet



Sjekker om tilkobling er mulig. Nå er det klart for å opprette WVD-host-poolen ved hjelp av en template



⁵¹ (Brasø & Aalrust, createImageFromVM.ps1, 2020)

Tabell 32 - Test av image

4.13 WVD

WVD skal brukes til å videreføre terminaltjenester fra det tradisjonelle on-premises systemet. Det er flere fordeler med å ta i bruk WVD over RDS eller VDI, og for denne case-bedriften vil WVD utfase VPN-løsningen helt. Ettersom VPN tidligere ble brukt til å la brukere koble til utenfra, så skal nå WVD brukes til å gi brukeren skrivebord og applikasjoner, ved hjelp av autentisering via Azure AD-omenet.

Før arbeidet med dette kan startes må følgende gjøres klart:

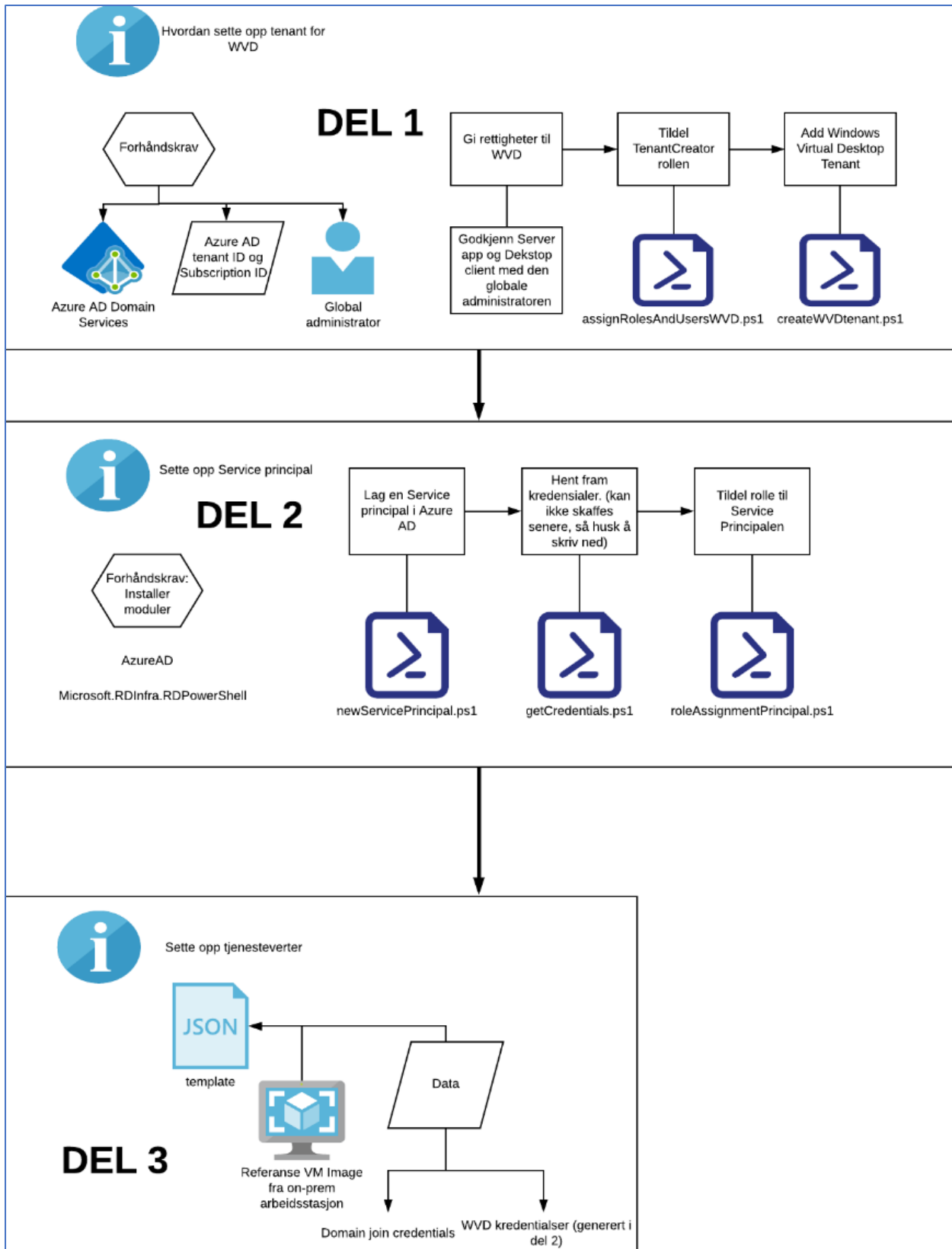
Tenant ID på Azure AD – Finnes på oversikten til Azure AD. I dette tilfellet «Sopra Steria Student Bachelor»



En global administrator-konto. I dette tilfellet kommer det til å være «emil@bachelorntnu20.onmicrosoft.com»



Denne prosessen vil gjennom portalen kreve en god del steg, men gjennom ARM og PowerShell spares mye arbeid. Store deler av denne prosessen kan automatiseres, men ikke alt. Figuren på neste side viser overordnet hvilke steg man må gjennom.



Figur 33 - Overordnet prosess WVD

4.13.1 Avhengigheter

Opprinnelig var det planlagt at WVD skulle rulles ut som en del av *main.json*. Dette viste seg derimot uhensiktsmessig, siden mye av ressursene som brukes i WVD kommer som følge av både *main.json*, men også selve migreringen. Derfor anbefales det at WVD iverksettes sist i utførelsen. Da kan man:

- Bruke *AADDS* og *Storage Account* som er rullet ut av *main.json*
- Bruke VM-images fra on-premises arbeidsstasjoner som er migrert over til Azure.

4.13.2 Sett opp rettigheter og prinsipaler

Logg inn som domene-administrator og aksepterer Windows Virtual Desktop server app og Windows Virtual Desktop Client App via følgende lenker:

- [Windows Virtual Server App](#)
- [Windows Virtual Desktop Client](#)

Følgende vinduer vil dukke opp:

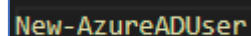
The image shows two side-by-side screenshots of Microsoft consent prompts. Both prompts are for the user 'endre@bachelorntnu20.onmicrosoft.com' and are titled 'Tillatelser forespurt Godta for organisasjonen'. The left prompt is for 'Windows Virtual Desktop' and lists permissions such as 'Read all users' basic profiles', 'Read all users' full profiles', and 'Read directory data'. The right prompt is for 'Windows Virtual Desktop Client' and lists permissions such as 'Access Windows Virtual Desktop (Windows Virtual Desktop)' and 'Aktiver pålogging og les brukernes profiler'. Both prompts include a warning: 'Dette programmet er ikke publisert av Microsoft eller organisasjonen din.' and a note: 'Denne appen ønsker å:'. At the bottom of each prompt are two buttons: 'Avbryt' (grey) and 'Godta' (blue).

Velg «Godta» på begge

The image shows a screenshot of a 'Thank You!' message. The text reads: 'Thank You!' followed by 'AAD Application has been successfully registered'. The message is enclosed in a blue border.

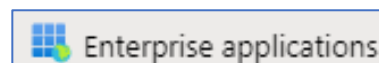
4.13.2.1 Legger til en ny bruker som tenantCreator

Bruker skriptet `newTenantUser.ps1`⁵² som oppretter en ny bruker, blant annet med CMD-letten `New-AzureADUser`.

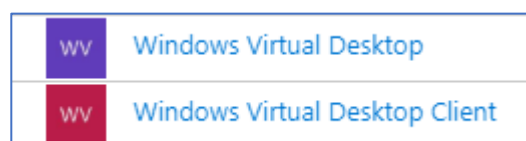


4.13.2.2 TenantCreator application role

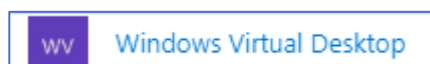
Gå inn på *Enterprise Applications*



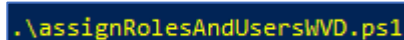
Søk *Windows Virtual Desktop*. De to applikasjonene som ble gitt tilgang tidligere vil komme opp.



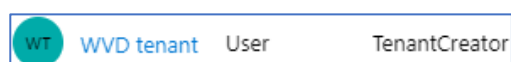
Går inn på *Windows Virtual Desktop* og velger *assign users and groups*, velger brukeren *WVD Tenant* og tildeler *TenantCreator*



Alternativt kan dette gjøres med skriptet `assignRolesAndUsersWVD.ps1`⁵³



Etter begge tilfeller kan man verifisere at brukeren har fått rollen *Tenant Creator*, gjennom *Users and Groups* som ble gjennomgått.



Tabell 33 - *TenantCreator* application role

Videre trengs følgende informasjon for å fortsette med oppsettet av WVD:

- Azure Tenant ID
 - Finnes gjennom Azure AD
- Azure Subscription ID
 - Finnes gjennom Subscriptions

⁵² (Brasø & Aalrust, `newTenantUser.ps1`, 2020)

⁵³ (Brasø & Aalrust, `assignRolesAndUsersWvd.ps1`, 2020)

4.13.2.3 Create a Windows Virtual Desktop tenant

I dette kapittelet må en del mindre og enkeltstående kommandoer benyttes og forklares. Dette er fordi man i denne delen skal benytte en del verdier man har hentet fra Azure-miljø, men gjelder særlig for det neste delkapittelet, hvor brukerlegitimasjon genereres, kopieres av brukeren og deretter brukes.

Last ned og importer WVD modulen for PowerShell

```
C:\Users\endre> Install-Module -Name  
"Microsoft.RDInfra.RDPowershell"
```

logg inn med WVD tenant

```
C:\Users\endre> Add-RdsAccount -DeploymentUrl  
"https://rdbroker.wvd.microsoft.com"
```

Følgende informasjon om *tenanten* vises og innloggingen er vellykket

```
TenantGroupName      UserName  
-----  
Default Tenant Group wvdtenant@bachelorntnu20.onmicrosoft.com
```

Lag en WVD tenant, med de verdiene som ble gjort klare.

```
C:\Users\endre> New-RdsTenant -Name  
"Sopra Steria Student Bachelor"  
-AadTenantId [REDACTED]  
-AzureSubscriptionId [REDACTED]
```

Får opp følgende informasjon om *tenanten*

```
TenantGroupName      : Default Tenant Group  
AadTenantId          : [REDACTED]  
TenantName           : Sopra Steria Student Bachelor  
Description           :  
FriendlyName         :  
SsoAdfsAuthority     :  
SsoClientId          :  
SsoClientSecret      :  
AzureSubscriptionId  : [REDACTED]  
LogAnalyticsWorkspaceId :  
LogAnalyticsPrimaryKey :  
WARNING: 1.0.1534.2001
```

Ofte er det greit å gi en ekstra bruker tilgang på tenanten, slik at systemet ikke blir «lammet» dersom du blir kjørt på av trikken.

```
C:\Users\endre> New-RdsRoleAssignment  
-TenantName <navnpåtenant>  
-SignInName <innloggingsnavn>  
-RoleDefinitionName "RDS Owner"
```

Tabell 34 - WVD tenant

4.13.2.4 Create service principals and role assignments with Powershell

Installer AzureAD modul

```
Install-Module -Name AzureAD
```

Installer WVD modul

```
C:\Users\endre> Install-Module -Name "Microsoft.RDInfra.RDPowershell"
```

Kjører skriptet *newServicePrincipal.ps1* og logg inn med WVD *tenant* brukeren. Dette lager en såkalt *multitenant service principal* i Azure.

```
.\newServicePrincipal.ps1
```

Henter ut følgende tre verdier

```
$svcPrincipalCreds.Value  
$aadContext.TenantId.Guid  
$svcPrincipal.AppId
```

Kobler til WVD og viser *tenants*

```
Add-RdsAccount -DeploymentUrl  
"https://rdbroker.wvd.microsoft.com"  
Get-RdsTenant
```

Kjører følgende skript for å tildele rolle slik at *Service Principal* kan logge inn på WVD

```
$myTenantName = "<Windows Virtual Desktop Tenant Name>"  
New-RdsRoleAssignment -RoleDefinitionName "RDS Owner"  
-ApplicationId $svcPrincipal.AppId  
-TenantName $myTenantName
```

Eksempel på utskrift etter det ble opprettet en *Service Principal*

```
RoleAssignmentId : ████████████████████  
Scope : /Default Tenant Group/Sopra Steria Student Bachelor  
TenantGroupName : Default Tenant Group  
TenantName : Sopra Steria Student Bachelor  
DisplayName :  
SignInName :  
GroupObjectId :  
AADTenantId :  
AppId : ████████████████████  
RoleDefinitionName : RDS Owner  
RoleDefinitionId : ████████████████████  
ObjectId : ████████████████████  
ObjectType : ServicePrincipal
```

Tillatt innlogging med *Service Principal* brukeren som nettopp ble lagd, og logg inn.

```
$creds = New-Object System.Management.Automation.PSCredential  
($svcPrincipal.AppId, (ConvertTo-SecureString  
$svcPrincipalCreds.Value -AsPlainText -force))  
Add-RdsAccount -DeploymentUrl  
"https://rdbroker.wvd.microsoft.com"  
-Credential $creds -ServicePrincipal  
-AADTenantId $aadContext.TenantId.Guid
```

Tabell 35 - Service Principal Role og Role Assignments

4.13.3 Lag sesjonsverter med ARM og egendefinerte VM-images

En host pool er en samling med en eller flere VM-er for bruk med WVD. Siden en arbeidsstasjon er overført fra det eksisterende on-premises miljøet vil dette brukes som utgangspunkt, i stedet for en Azure Marketplace VM. Den eksisterende VM-en er gjort til et image for å opprette flere VM-er. Det må likevel utdeles et WVD host pool, dette løses med en template.

Logg inn på WVD tenant med PowerShell

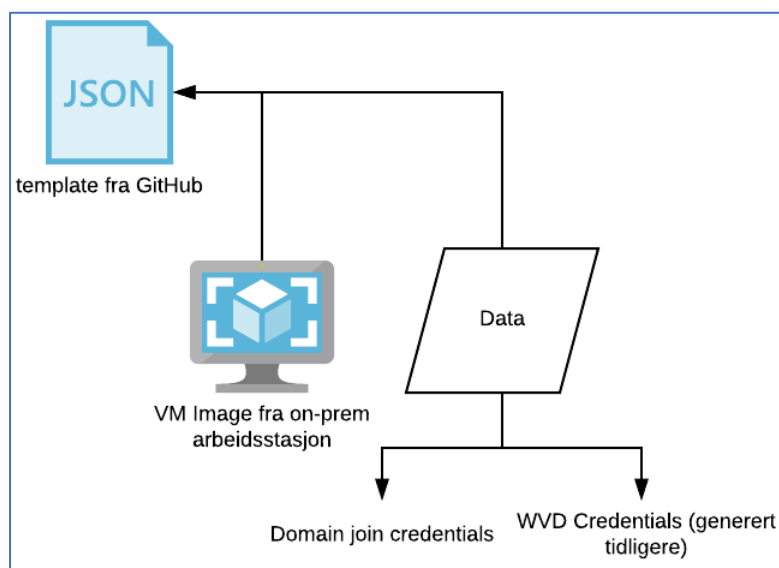
```
Add-RdsAccount -DeploymentUrl  
"https://rdbroker.wvd.microsoft.com"
```

Bruker template *provisionHostPool.json*⁵⁴ for å rulle ut tjenesteverter. Dette er en offentlig tilgjengelig template, som blir tilbydt gjennom github.

Tabell 36 – Sesjonsverter med ARM

Det er tidligere opprettet en egen bruker for å legge vm-er til i domenet. Dette er brukeren domainjoiner@bachelorntnu20.onmicrosoft.com. Brukeren ble opprettet før AADDS ble aktivert i miljøet, derfor var det nødvendig å endre passordet på brukeren en gang først. Dette er vist i 4.7.2⁵⁵. Denne brukeren skal brukes i resten av dette delkapitlet.

Følgende info trengs før utrulling kan skje.



Figur 34 - Oversikt custom image

- Kilden til imaget. (custom)
- Legitimasjon for å melde inn i domenet.
- Legitimasjon tilknyttet WVD. Det som ble opprettet tidligere.

⁵⁴ (Microsoft, mainTemplate.json, 2020)

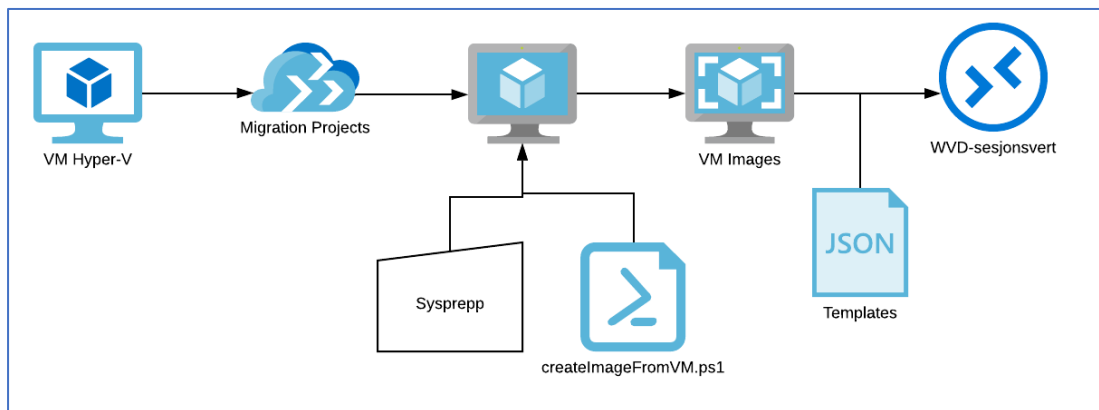
⁵⁵ Endre til kerberos autentisering

Templaten som brukes lager VM-er og registrere de som sesjons-verter på en ny eller eksisterende WVD host pool.

På denne templatene kunne man ha valgt mellom:

- Azure Gallery Image
- Custom VHD from blob storage
- Custom Azure Image resource from a resource group.

Oftest er det greit å velge den aller første, ettersom det vil gi oppdaterte Operativsystem. Men i denne casen tas det forbehold om at bedriften har en rekke spesialiserte arbeidsstasjoner med operativsystem og programvare tilpasset deres behov. Derfor brukes valg tre: *Custom Azure Image Resource From a resource Group*. Dette vil da være en image som kommer fra en arbeidsmaskin som ble migrert tidligere i prosessen. Figuren under forsøker å illustrere gangen fra en VM på hyper-V til en WVD-sesjonshost. En wvd-host er sluttresultatet av denne prosessen, og vil være VM-en som sluttbruker skal kobles til.














Figur 35 - Oversikt WVD sesjonsvert

Prosessten kan oppsummeres i følgende steg:

1. Det eksisterer fra før av en VM på Hyper-V hosten som er On-premises. VMen vil i denne casen inneholde en skreddersydd arbeidsoppgave, som er vanskelig å innføre som en ren tjeneste per dags dato i Azure. Derfor blir den migrert over til en ny VM i Azure.
2. Migreringen blir gjort med Azure Migrate (et Migration Project).
3. Når VM-en på Azure er klar, må det kjøres Sysprepp på den, før det deretter kjøres skriptet *createImageFromVM.ps1* opp mot Azure-miljøet.
4. Dette oppretter da et VM Image i ressursgruppen
5. Deretter brukes templatene til å rulle ut WVD-systemet, hvor disse VM-imagene blir brukt til å opprette WVD-sesjonsverter. (maskinene som tilbyr WVD).


Ved utrulling av templatene kan framgangen spores i ressursgruppen, og vil se noenlunde slik ut: her ble det prefikset «wvd» definert på parameterne i templatene, og derfor følger alle VM-ene denne navne-konvensjonen.

Resource	Type	Status	Operation details
 wvd-0	Microsoft.Compute/virtualMachines	Created	Operation details
 wvd-1	Microsoft.Compute/virtualMachines	Created	Operation details
 wvd-3	Microsoft.Compute/virtualMachines	Created	Operation details
 wvd-4	Microsoft.Compute/virtualMachines	Created	Operation details
 wvd-2	Microsoft.Compute/virtualMachines	Created	Operation details
 wvd-1-nic	Microsoft.Network/networkInterfaces	Created	Operation details
 wvd-0-nic	Microsoft.Network/networkInterfaces	Created	Operation details
 wvd-2-nic	Microsoft.Network/networkInterfaces	Created	Operation details
 wvd-3-nic	Microsoft.Network/networkInterfaces	Created	Operation details
 wvd-4-nic	Microsoft.Network/networkInterfaces	Created	Operation details
 wvd-availabilitySet	Microsoft.Compute/availabilitySets	OK	Operation details





Figur 36 - WVD maskiner

Etter en stund er operasjonen *joindomain* ferdig på de fleste- VM-ene, og man er da klar til å gå videre.

■ ■ ■ Your deployment is underway

 Deployment name: wvd-template Start time: 4/8/2020, 12:46:13 PM
Subscription: Azure subscription 1 Correlation ID: 8cff9b00-a551-450e-a611-f04b2
Resource group: testMig312

^ Deployment details [\(Download\)](#)

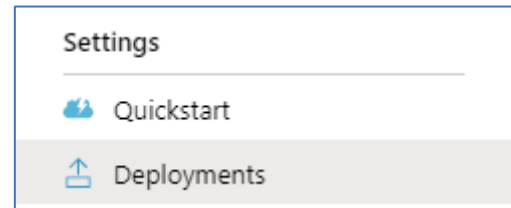
Resource	Type	Status	Operation details
 wvd-0/dscentensi...	Microsoft.Comput...	Created	Operation details
 wvd-1/joindomain	Microsoft.Comput...	OK	Operation details
 wvd-0/joindomain	Microsoft.Comput...	OK	Operation details
 wvd-2/joindomain	Microsoft.Comput...	OK	Operation details

Figur 37 - Joindomain

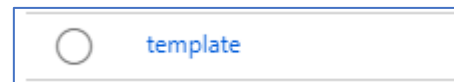
4.13.3.1 Legge til flere WVD-host

Ofte er det nødvendig å oppskalere WVD-miljøet, som for eksempel dersom det blir ansatt nye arbeidere. Her trengs ikke et nytt skript eller template, template som er brukt før kan brukes om igjen, men endrer antallet sesjonsverter til det ønskede antallet. Slik gjøres det:

Gå inn på Deployments under ressursgruppen (test1604)



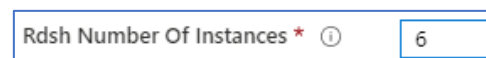
Fordi det ble brukt en ARM-template heter den forrige utrulling for *template*



Velger *Redeploy*



endre *Rdsh Number Of Instances* til det nye antallet



Deretter må passord skrives inn på nytt. Dette gjelder for *domainjoiner*-brukeren sitt passord som er satt fra før av, men også for *tenant admin*. Dette er da passordet som ble genert tilfeldig tidligere i prosessen, med PowerShell.

trykker på *purchase* for å bekrefte (den oppdaterte) utrulling

Purchase

Tabell 37 – Legge til flere sesjonsverter

Etter en stund kan man da se at antallet sesjonsverter er oppdatert.

4.13.4 Sette app groups

Dette gjøres for å gi ut ressurser og applikasjoner gjennom WVD. Det kan opprettes egendefinerte grupper med egne applikasjoner, men som standard blir det opprettet en *Desktop Application Group*, som gir tilgang til å bruke sesjonsvertene som eksterne skrivebord.

Logg inn med brukeren som er satt opp som *rds owner*, og som er brukt tidligere i oppsettet. (*wvdtenant* i dette tilfellet)

```
Add-RdsAccount -DeploymentUrl  
"https://rdbroker.wvd.microsoft.com"
```

Kjører følgende kode for å få en oversikt over brukeren

```
Get-RdsAppGroup "TestTenantNTNU"  
"testpool2"
```

Finner fram app-gruppene som er i miljøet

```
TenantGroupName : Default Tenant Group  
TenantName      : TestTenantNTNU  
HostPoolName    : testpool2  
AppGroupName    : Desktop Application Group  
Description     : The default desktop application  
FriendlyName    : Desktop Application Group  
ResourceType    : Desktop
```

Siden det for øyeblikket kun skal brukes som eksternt skrivebord må brukerne legges til i gruppen *Desktop Application Group*.

```
Add-RdsAppGroupUser "TestTenantNTNU" "  
testpool2" "Desktop Application Group"  
-UserPrincipalName «bruker»
```

Tabell 38 - App groups

Etter alt dette er gjort, er WVD tilgjengelig for brukerne, slik som beskrevet i 4.1.6 Windows Virtual Desktop Brukerguide.

5 Brukerguide for verktøy og funksjoner i Azure

I dette kapitlet beskrives det nye miljøet til bedriften etter at migreringen er gjennomført. Mange funksjoner og tjenester er nye etter migrering til Azure, de beskrives i relasjon til sluttbrukere.

5.1 Azure Backup

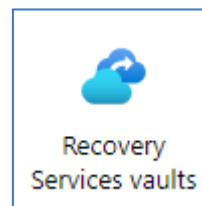
Azure Backup er Azure sin innebygde løsning sikkerhetsmessig. Det er noen fordeler og ekstra-funksjonalitet som er greit å trekke fram:

- «øyeblikksbilder» man kan ta nøyaktige snapshots av virtuelle maskiner (både Windows og Linux slik det unngås å miste data som er i minnet eller lignende.
- Mindre komplisert og mer kostnadseffektiv enn mange alternativer.
- Et-klikks-sikkerhetskopiering av SQL-databaser og Virtuelle Maskiner. (etter at backup er satt opp)
- Ekstra beskyttelse mot ransomware, i form av at uautoriserte slettinger av data beholdes i opptil 14 dager.

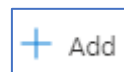
Siden dette er en innebygd tjeneste, er det ikke nødvendig med særlig mye tilpasning for å ta Azure Backup i bruk. Dersom det ikke foreligger en Recovery Services Vault fra før av, kan man opprette en ny.

5.1.1 Sette opp Recovery Services Vault

Går inn på *Recovery Services Vault*



Add

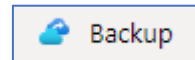


Det er få parametre, men det er greit å velge en region som stemmer overrensse med resten av Miljøet. I dette tilfellet *North-Europe*

Tabell 39 - Oppsett Recovery Vault

5.1.2 Sette opp sikkerhetskopiering av VM

På oversikten til VM-en velges *Backup*



Velger vaultet.

Recovery Services vault ⓘ

Create new Select existing

testVault ▾

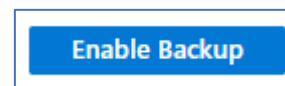
Velger en policy. Hvis det ikke eksisterer en passende policy, kan det lages gjennom denne menyen.

Choose backup policy ⓘ

DefaultPolicy ▾

Create a new policy

Enable Backup

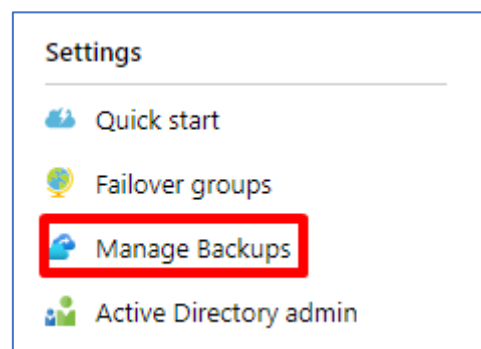


Tabell 40 - Sikkerhetskopiering av vm

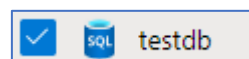
5.1.3 Konfigurerere backup-retention på SQL-servere.

Retention betyr hvor lenge sikkerhetskopier oppbevares, i dette tilfellet sikkerhetskopiene av SQL-serveren. Dette har innvirkning på hvor lang tilbake i tid man kan gjenopprette.

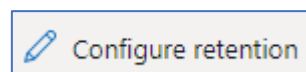
Fra SQL-Serveren det gjelder går man på *Settings > Manage Backups*



Velger Databasen(e) det gjelder

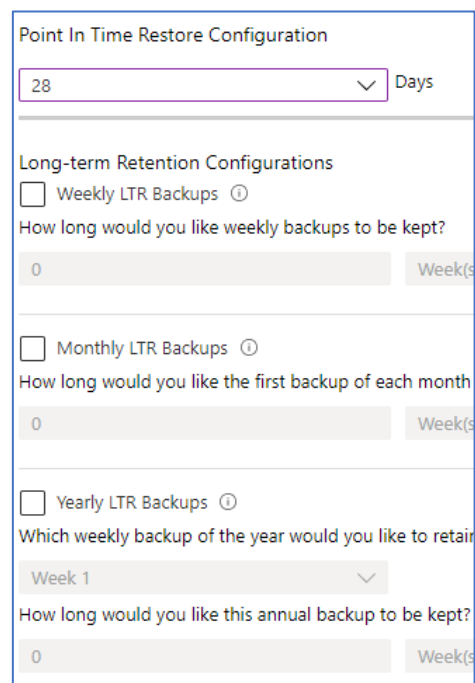


Configure retention



Velger:

- Hvorvidt det skal tas ukentlig, månedlige og/eller årlige backups
- Hvor lenge disse lagres
- Hvor langt tilbake *Point-In-Time Restoration* gjelder.



Tabell 41 - Konfigurasjon for backup retention

5.2 Azure Runbook

Azure Runbook er et verktøy i azure som i all hovedsak benyttes til å automatisere hverdagslige oppgaver i Azure miljøet og for å administrere for eksempel Windows-oppdateringer og lignende.

Runbooks kan benyttes på 3 forskjellige måter:

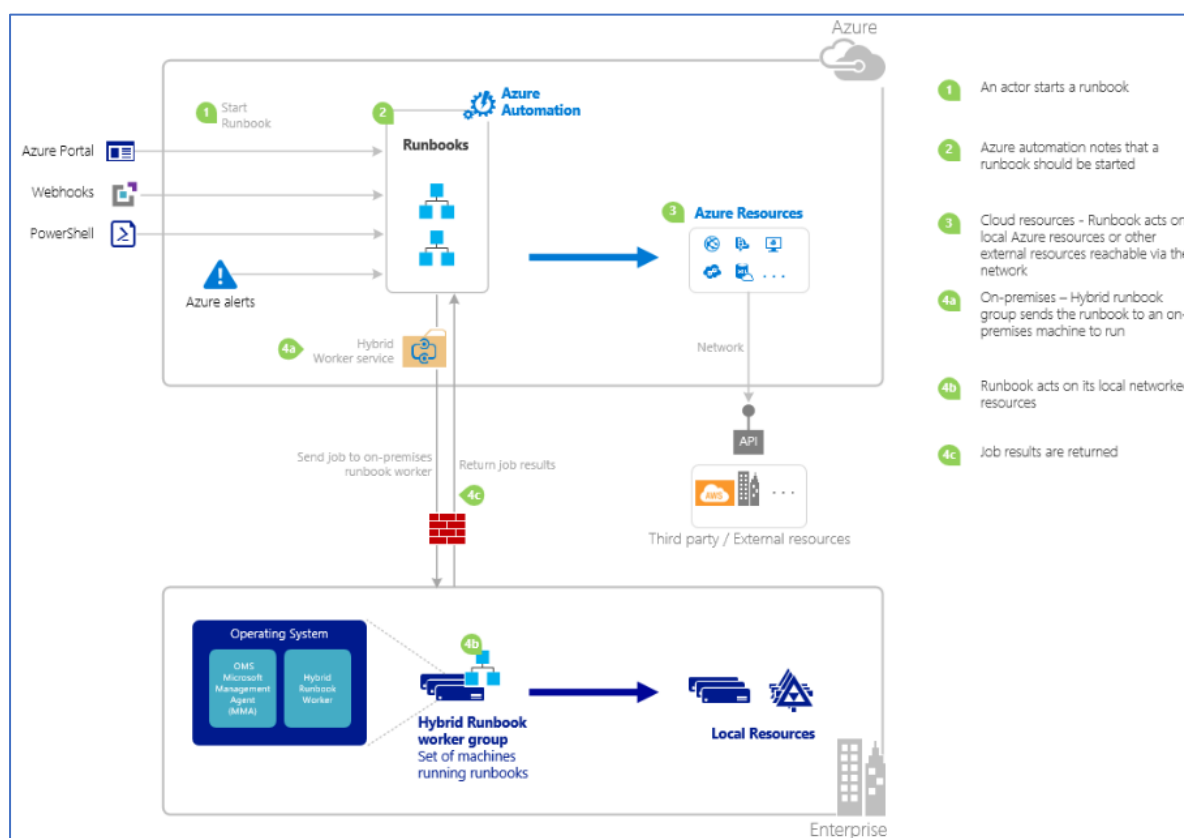
- Gjennom Azure Portalen
- Gjennom en webhook (for eksempel en ekstern tjeneste som GitHub)
- Gjennom PowerShell

I denne oppgaven benyttes Azure Portalen som et eksempel sammen med skriptet *cleanupAutomation.ps1* som et eksempelskript.

Bildet ⁵⁶ på neste side forklarer hvordan runbooks fungerer ved de forskjellige måtene og sammen med resten av Azure.

Azure Runbooks benyttes i denne oppgaven på grunn av funksjonene knyttet opp mot kjøring av skript på miljøet og Azure Update Manager. Runbooks gjør det lettere å drifte et miljø fordi vedlikehold kan bli gjort gjennom skripts som blir kjørt automatisk. Hvis et skript ikke blir kjørt eller krasjer vil en detaljert feilmelding vise hva som gikk galt og det blir dermed lett å fikse og få miljøet opp å kjøre igjen. Runbooks kan også brukes i sammenheng med Log Analytics til for eksempel overvåking og lage egendefinerte varsler for operasjoner.

Som man ser på bildet er det også mulig å benytte Runbooks i sammenheng med en hybridløsning slik at skriptene også kjører på on-premises miljøet. Denne funksjonaliteten blir ikke brukt i dette prosjektet, men kan være hensiktsmessig der deler av on-premises miljøet fortsatt eksisterer.

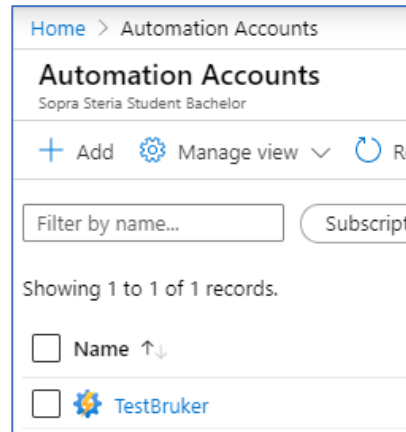


Figur 38 - Oversikt Azure Runbooks

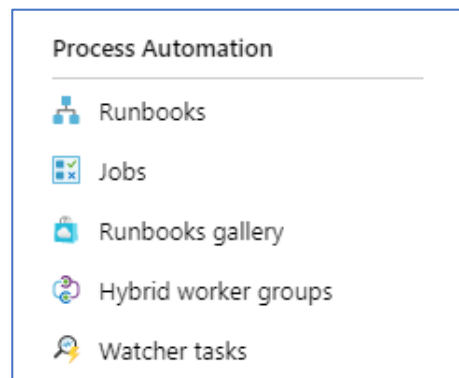
⁵⁶ (Microsoft, 2018)

Under beskrives det hvordan runbook opprettes og hvordan de brukes. Starter her med å opprette en automasjonskonto som skal benyttes i sammenheng med runbooks.

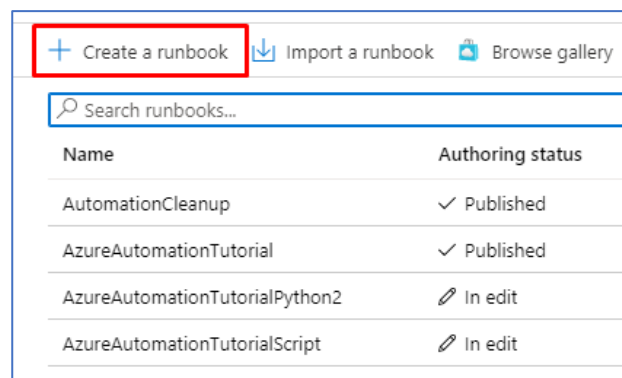
Søker på *Automation-Accounts* og oppretter en konto hvis det er nødvendig



Klikker på *Runbooks* i menyen som kommer opp



Klikker deretter på *Create a runbook* i fanen



Fyller inn boksen som dukker opp på høyre side med nødvendig informasjon

Når runbooken er opprettet vil en editor dukke opp. Her kan skriptet limes inn.

Når skriptet er ferdig, klikk på *save* etterfulgt av *publish*

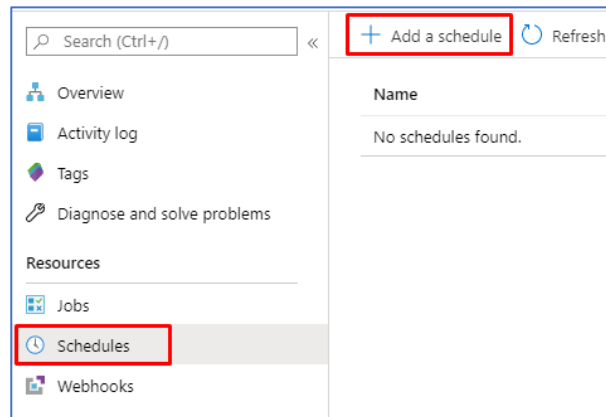
Vinduet som kommer opp, inneholder informasjon om runbooken.

Tabell 42 - Azure Runbook

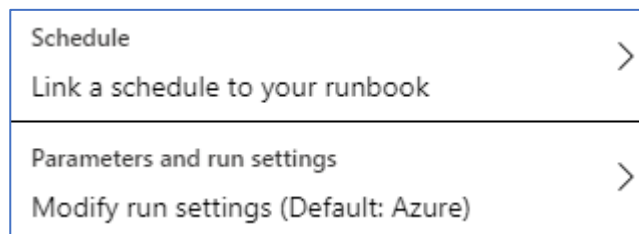
5.2.1 Skript

For å kjøre skripts i Azure opprettes en runbook til hvert enkelt skript. Her beskrives det hvordan en runbook brukes sammen med et skript.

Opprette en *Schedule* for når skriptet skal kjøre.



Her velges en schedule etter når den skal kjøres og hvilke runbooks som benytter den.



Velger Create a new schedule og fyller inn informasjon

New Schedule

Name *

Description

Starts * ⓘ

03/16/2020 3:31 PM

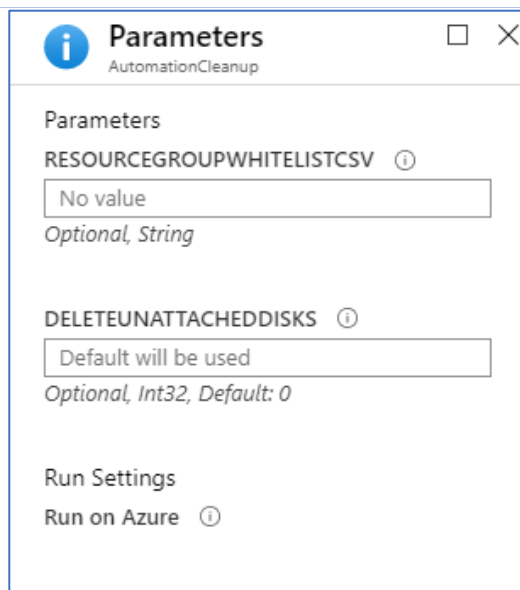
Time zone

Norway - Central European Time

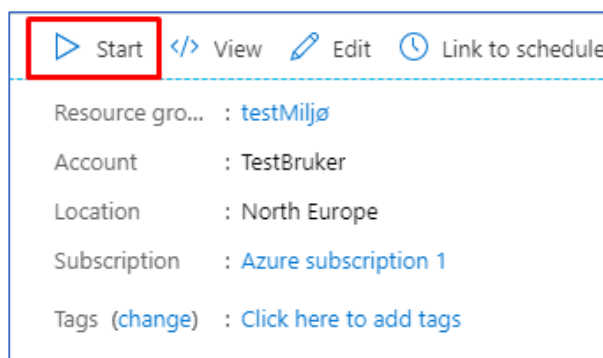
Recurrence

Once Recurring

Når dette er gjort kan *Parameters and run settings* definere hvis skriptet trenger parametere når det kjører.



Når dette er gjort kan runbooken startes og den vil kjøre etter den valgte schedulen.



Tabell 43 - Schedule og runbook

5.2.2 Automatisk opprensing i Azure-miljø

For å sørge for at det nye azure miljøet til bedriften ikke oversvømmes av tomme ressursgrupper og vm'er som ikke brukes opprettes et skript som kjøres ved hjelp av Azure Runbooks internt i Azure.

Skriptet sjekker hver enkelt ressurs og sletter den hvis den er «utgått». Måten skriptet sjekker dette på er gjennom en komponent i Azure kalt «TAG». Når en ressurs opprettes kan flere forskjellige tags settes på denne ressursen, disse kan være hvilken kostnadsgruppe ressursen skal falle inn under eller i dette tilfellet, en dato. Taggen som blir brukt i dette skriptet heter «expireOn» og har en dato hvor ressursen «går ut». Skriptet inneholder også en liste med ressursgrupper som ikke skal slettes og det er opp til bruker å legge inn disse i skriptet.

5.3 Log Analytics Workspace

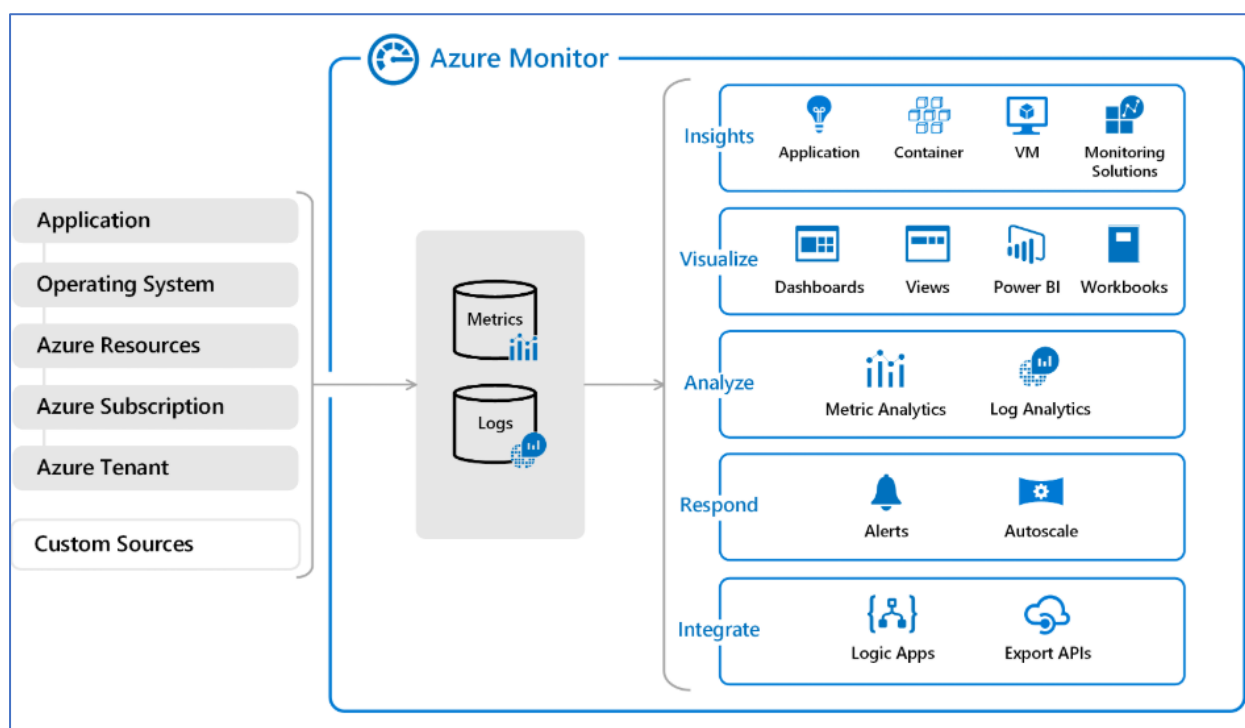
Dette er en tjeneste som tilbys gratis fra Azure sin side. Log Analytics benyttes til å samle data fra Azure miljøet og brukes av andre funksjoner i Azure som for eksempel Azure Update Management og Azure Runbooks. I log Analytics kan man utføre spørringer på dataene og se feilmeldinger og rapporter fra hver enkelt komponent.

Etter å ha opprettet et arbeidsområde for log analytics, må innsamling av data konfigureres. Her er dette et område med alle komponentene til testmiljøet etter migrering, men flere områder kan opprettes hvis nødvendig. Dette arbeidsområdet kan hente data fra følgende kilder:

- Azure Resources i abonnementet
- On-premises maskiner overvåket av System Center Operations Manager
- Enheter fra Configuration Manager
- Diagnostikk eller loggdata fra Azure Storage

I dette tilfellet er det mest aktuelt å hente data fra Azure Resources og Azure Storage fordi et on-premises miljø ikke eksisterer lenger.

Diagrammet ⁵⁷under viser hvordan Log Analytics fungerer i Azure (her i sammenheng med Azure Monitor).

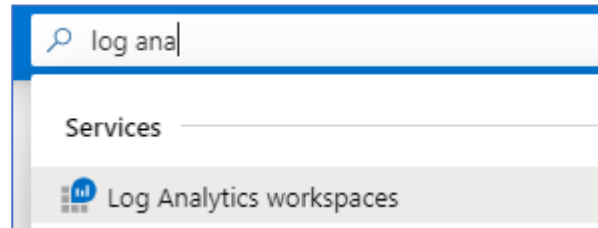


Figur 39 - Oversikt Log Analytics

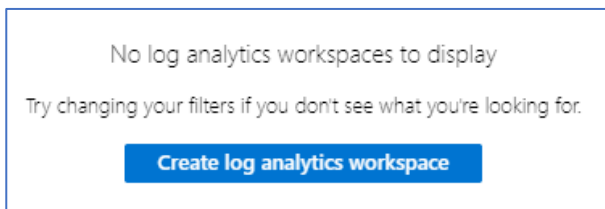
For å benytte denne tjenesten må det opprettes et arbeidsområde (workspace). I dette arbeidsområdet kan man se logger og utføre spørringer på dataene om nødvendig.

⁵⁷ (Microsoft, Azure Monitor overview, 2019)

Velger *Log Analytics workspace* i Azure



Oppretter et *workspace*. Merk at man kan ha mange *workspaces*.



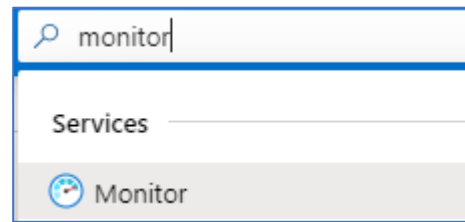
Fyller inn informasjon og klikker ok

A screenshot of the 'Log Analytics workspace' creation dialog box. The title is 'Log Analytics workspace' with a close button. Below the title is the subtitle 'Create new or link existing workspace'. There are two radio buttons: 'Create New' (selected) and 'Link Existing'. Below this is a text input field for 'Log Analytics Workspace *' containing 'SalmoAS-Workspace' with a green checkmark. Below that is a dropdown menu for 'Subscription *' with 'Azure subscription 1' selected. Below that is another dropdown menu for 'Resource group *' with 'TestUpdateManager' selected and a 'Create new' link. Below that is a dropdown menu for 'Location *' with 'West Europe' selected. At the bottom, there is a section for '*Pricing tier' with 'Pay-as-you-go' selected and a right-pointing arrow.

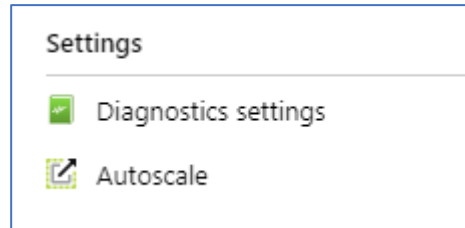
Tabell 44 - Log Analytics

5.3.1 Log Analytics datakonfigurasjon




Går inn på *Monitor* i Azure



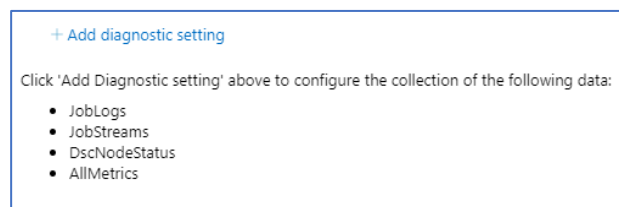
Klikker på *Diagnostics settings* i menyen på venstre side under *Settings*



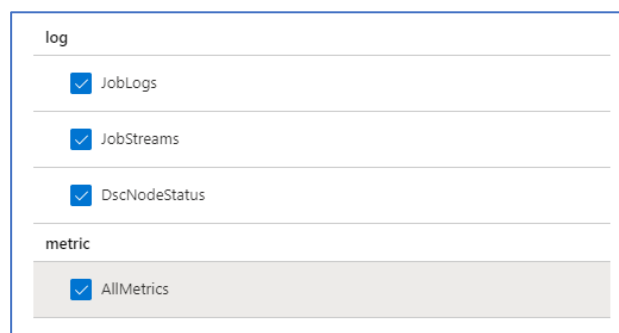
Her kan man velge hvilke ressurser som skal sende data til log analytics. Starter her med *TestBruker*

Name	Resource type
 TestBruker	Automation Account
 testAppliancee2ackv	Key vault
 recVault	Recovery Services vault

Velger *Add diagnostic setting*



Huker av hvilke typer data som skal sendes



Destinasjon er da i *Log Analytics* og arbeidsområdet som ble opprettet.

Destination details

Send to Log Analytics

Subscription

Log Analytics workspace

Archive to a storage account

Stream to an event hub

Tabell 45 - Log Analytics Konfigurasjon

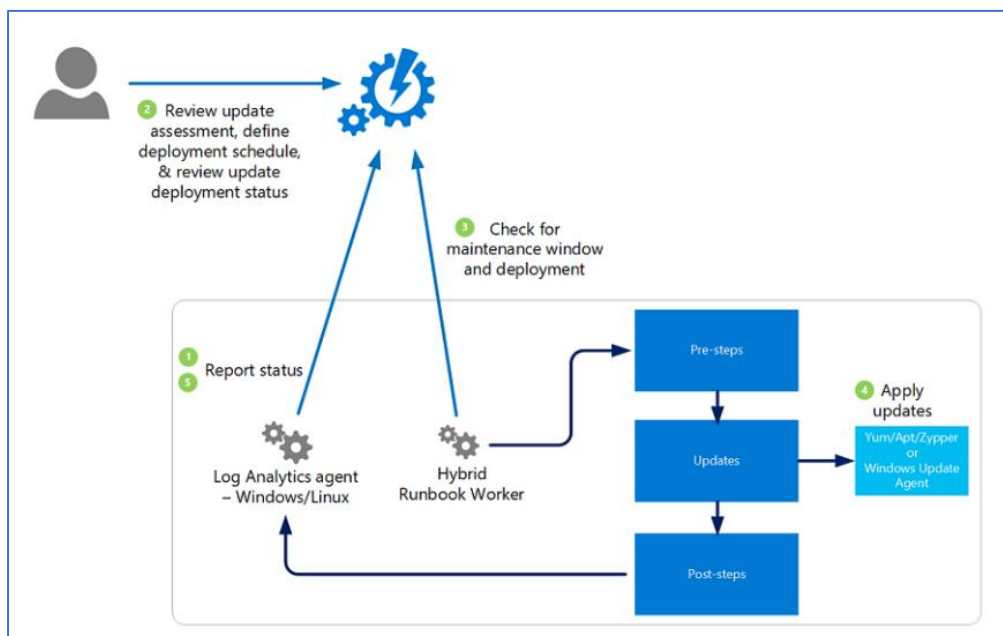
5.4 Azure Update Management

For å opprettholde sikkerheten i et system er det viktig at alle maskiner får oppdateringer og derfor benyttes Azure Update Management fordi dette verktøyet gir en oversiktlig måte å rulle ut disse oppdateringene på. Dette er en funksjon som erstatter SCCM i både on-premises og virtualiserte systemer.

Update Management er en funksjon i Azure som benyttes til å styre systemoppdateringer til vm'er Azure miljøet. Maskiner som administreres av Update Management bruker følgende konfigurasjoner til å vurdere og oppdatere utrullinger:

- Log Analytics agent for Windows eller Linux
- PowerShell Desired State Configuration (DSC) for Linux
- Automation Hybrid Runbook Worker
- Microsoft Update eller Windows Server Update Services (WSUS) for Windows maskiner

Følgende diagram hentet fra Microsofts hjemmesider⁵⁸ og viser hvordan Update Manager vurderer og distribuerer sikkerhetsoppdateringer til alle tilkoblede maskiner i arbeidsområdet.



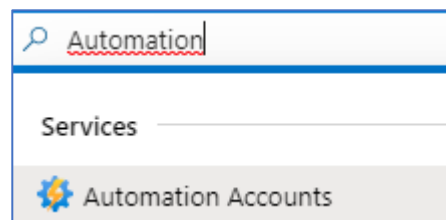
Figur 40 - Oversikt Azure Update Management

⁵⁸ (Microsoft, 2020)

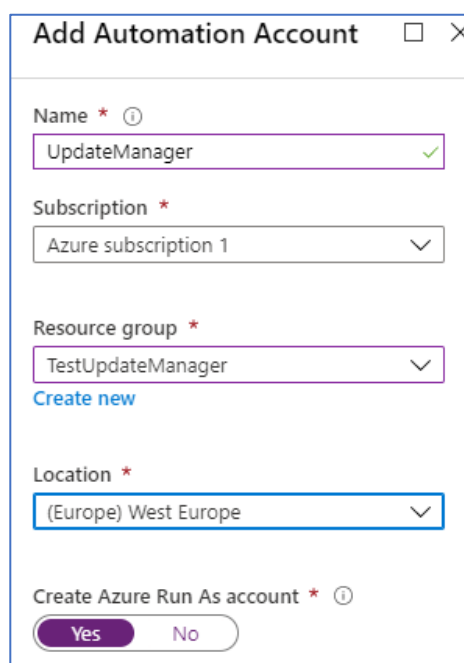
5.4.1 Konfigurere Update Manager

For å lettere kunne styre alle vm'ene i miljøet brukes Azure Update Manager. Her beskrives det hvordan Azure Update Manager konfigureres for å kunne brukes i miljøet. Etter å ha opprettet et arbeidsområde for Log Analytics og en automasjonskonto kan Update Management aktiveres i Azure. Her er det viktig at en automasjonskonto opprettes i en region⁵⁹ som samsvarer med log Analytics arbeidsområdet som ble opprettet.

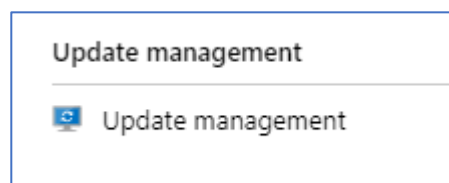
Går inn på *Automation Accounts*



Oppretter en konto

A screenshot of the 'Add Automation Account' dialog box in the Azure portal. The dialog has a title bar with a close button. It contains several fields: 'Name' with the value 'UpdateManager' and a green checkmark; 'Subscription' with a dropdown menu showing 'Azure subscription 1'; 'Resource group' with a dropdown menu showing 'TestUpdateManager' and a 'Create new' link below it; 'Location' with a dropdown menu showing '(Europe) West Europe'; and 'Create Azure Run As account' with 'Yes' and 'No' radio buttons.

Går inn på brukeren som ble opprettet og velger *Update Management* i menyen til venstre.



⁵⁹ (Microsoft, 2020)

Hvis alt er gjort riktig skal arbeidsområdet kunne velges og klikker *Enable*

Log Analytics workspace location ⓘ
West Europe

Log Analytics workspace subscription ⓘ
Azure subscription 1

*Log Analytics workspace ⓘ
SalmoAS-Workspace

Automation account ⓘ
Update-Manager

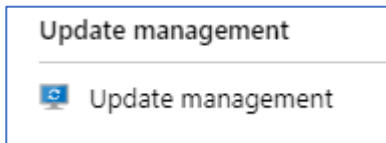
Enable

Tabell 46 - Azure Update Management

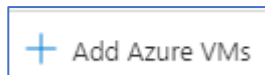
5.4.2 Legge til maskiner i Update Manager

Azure Update Management er nå konfigurert og kan benyttes på et utvalg maskiner.

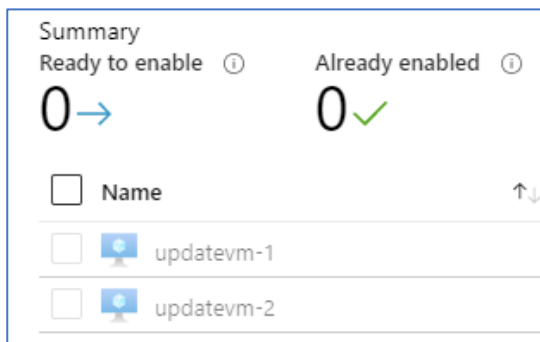
Inne på Azure Automation velges *Update Management*



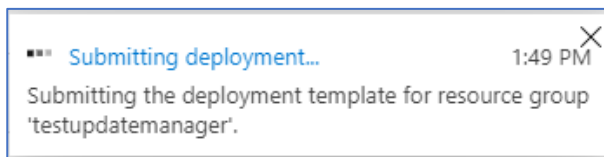
Velger *Add Azure VMs*



Får opp alle maskinene som kan legges til i Azure Update Management. Her velges alle.



Klikker *Enable* og venter på at utrulling skal bli ferdig



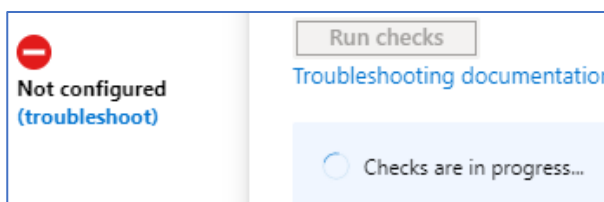
Etter en stund vil maskinene komme opp i denne oversikten.

Machine name	Compliance
UpdateVM-1 Azure: TestUpdateManager/UpdateVM-1	❗ Non-compliant as of 4/2/2020, 3:11 PM
updateVM-2 Azure: TestUpdateManager/updateVM-2	❗ Non-compliant as of 4/2/2020, 3:34 PM

Feilmeldingen *non-compliant* i dette tilfellet betyr at maskinene ikke har alle nødvendige oppdateringer.

Machines need attention (3)		Missing updates (6)	
Critical and security	2 ■	Critical	1 ■
Other	0	Security	0
Not assessed	1 ■	Others	5 ■

Det kan også være nødvendig å kjøre en sjekk på hver enkelt vm, hvis de er *non-compliant*

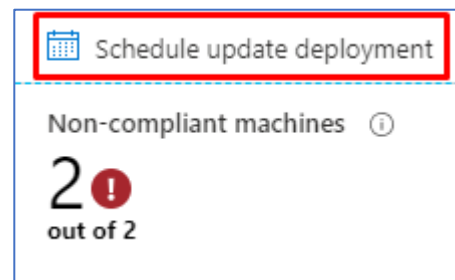


Tabell 47 - Legge til maskiner i Update Manager

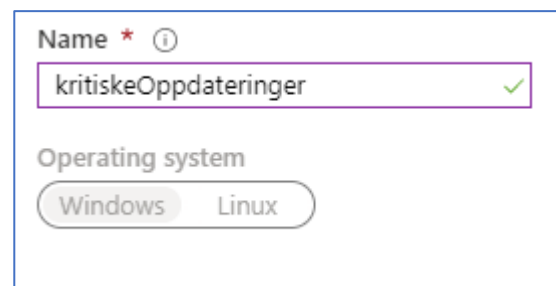
5.4.3 Planlegging og utrulling av oppdatering

Azure Update Manager skal nå tas i bruk til å lage en plan for utrulling av oppdateringer.

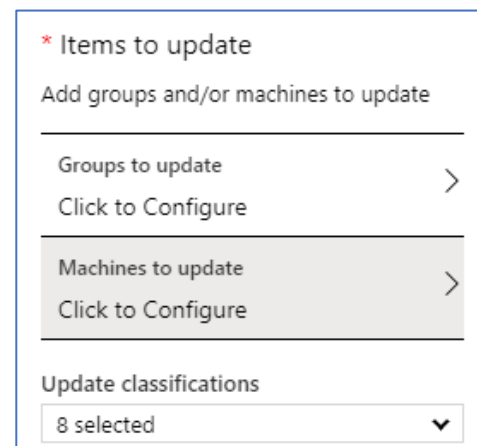
Velger *Schedule update deployment* i Update Management



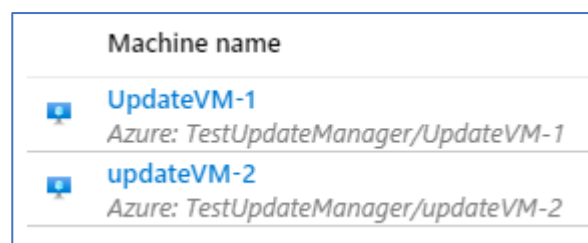
Oppretter et navn for planen og hvilken type operativsystem maskinene som skal legges til har



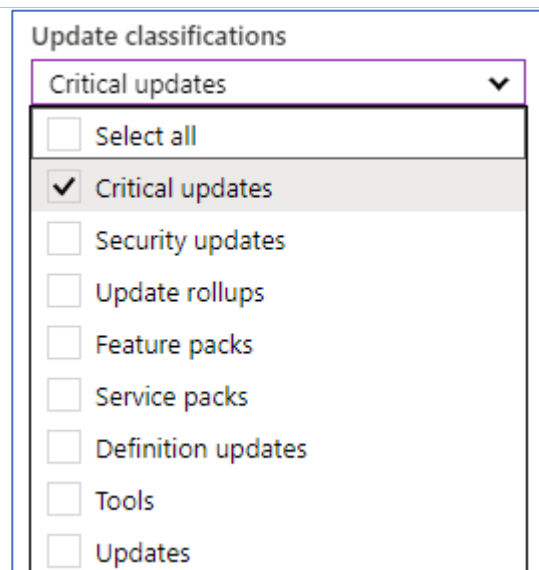
Videre velges hvilke maskiner som skal assosieres med denne utrullingsplanen. Her får man to valg; enten å legge til maskiner fra både grupper eller velge rett fra maskinene i azure miljøet.



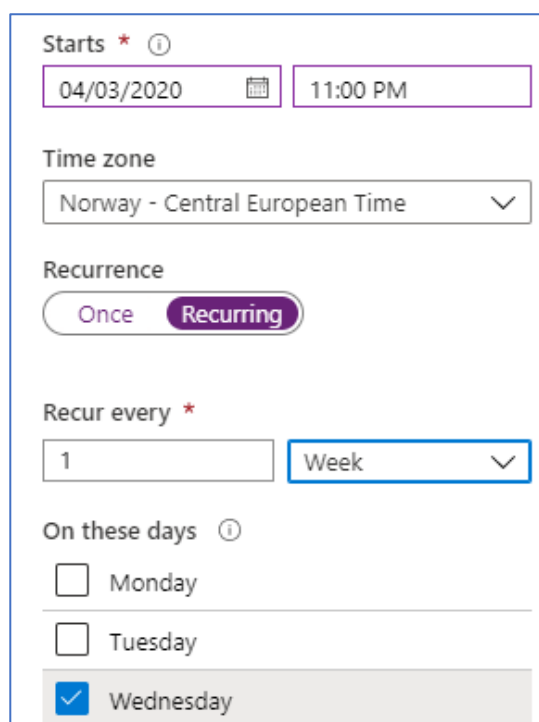
Fordi miljøet har veldig få maskiner assosiert med Update Manager er det her lettere å velge fra *Machines to update*



Under *Update classifications* kan man velge hvilke oppdateringer som skal rulles ut til maskinene. Her vil det være hensiktsmessig å ikke velge alle for en enkelt utrullingsplan siden dette tar lang tid å installere på hver enkelt maskin. Derfor velges bare de kritiske oppdateringene i denne utrullingsplanen.

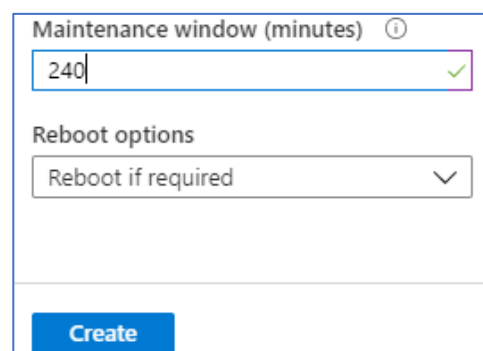


Konfigurerer når oppdateringene skal rulles ut til maskinene. For å forhindre at oppdateringene krasjer med standard arbeidstid kan man velge for eksempel hver onsdag kl. 23.00. For at utrullingen skal skje hver onsdag må utrullingen settes til *recurring*. For kritiske oppdateringer kan det være hensiktsmessig å kjøre oftere.



Hvis det er mye aktivitet på maskinene, kan man også sette et tidsvindu for hvor lang tid oppdateringene får til installeringen.

Det er også mulighet til å styre omstartskonfigurasjon. Velger her å sette omstart hvis nødvendig og klikker *Create*



Tabell 48 - Rulle ut oppdateringer i Update Manager

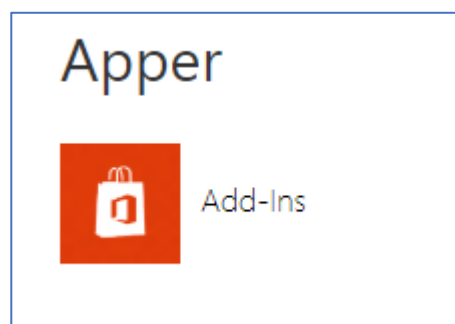
5.5 Azure Active Directory Access Panel

Azure Ad Access panel er en web-basert portal som gjør det mulig for en sluttbruker å se og kjøre applikasjoner som de har fått tilgang til i domenet. Denne portalen gjør det mulig for brukeren å benytte *single sign-on* til alle applikasjoner i portalen. Her kan man også få tilgang til å forandre medlemskap i gruppene assosiert med domenet og forandre tilgangsrettigheter på gruppenivå eller brukernivå til de forskjellige applikasjonene.

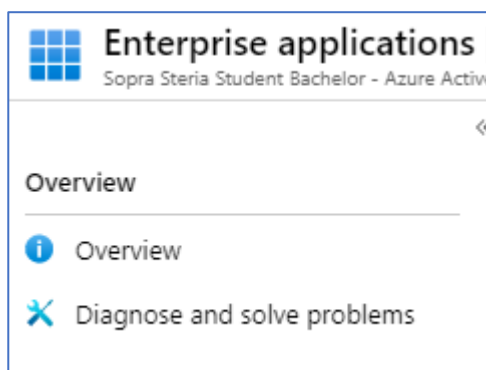
I dette miljøet er det hensiktsmessig å implementere denne funksjonen fordi det gjør det lettere for alle sluttbrukere å få tilgang til applikasjoner brukt i hverdagen. Ved å innføre denne funksjonaliteten vil også at sikkerheten i miljøet mot eksterne trusler blir forbedret fordi brukerne slipper å laste ned applikasjoner på egenhånd. Access panel kan også konfigureres slik applikasjoner bare er synlige for en gruppe personer hvis dette er nødvendig for å opprettholde sikkerheten internt i bedriften.

Web-portalen finnes på siden: <https://myapps.microsoft.com> og forutsetter at man har en azure konto eller er medlem av et domene.

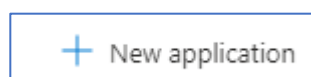
Fra denne siden får man en oversikt over alle appene som er lagt til



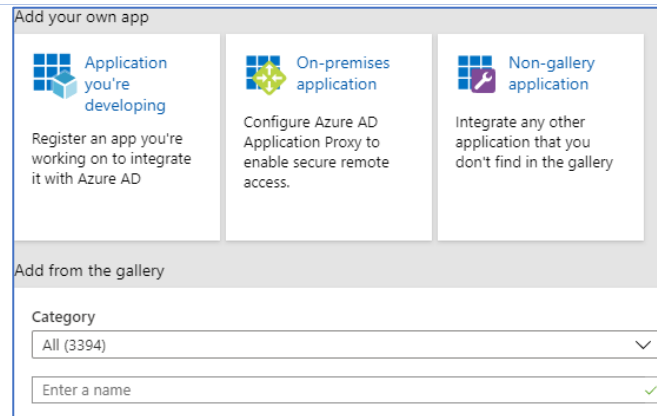
Som administrator kan man legge til flere applikasjoner, enten gjennom *myApps* eller Azure portalen. Velger her å gjøre dette i Azure med *Enterprise Apps*



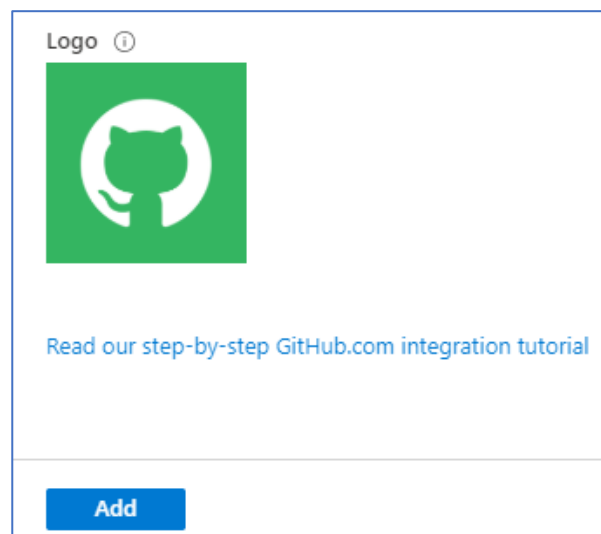
Velger *New application*



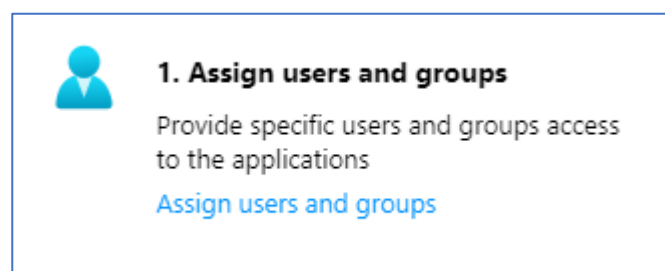
Gjennom Azure kan en egen applikasjon legges til eller finne andre applikasjoner fra galleriet.



Søker etter applikasjonene som skal legges til og velge *add*



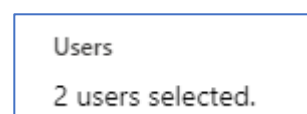
Begynner nå med konfigurering av applikasjonen. Første steg er *Assign users and groups*



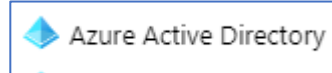
Velger *add user* for å legge til brukere



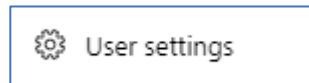
Velger aktuelle brukere og klikker *assign*



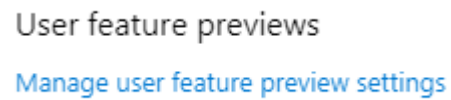
Går inn i Azure Active Directory for å vise applikasjonen hos brukeren



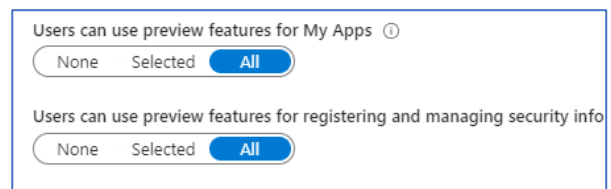
Velg *User Settings*



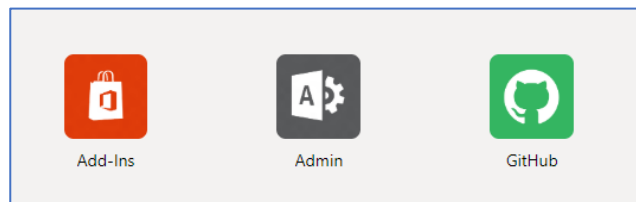
Deretter *Manage user feature preview settings*



Her kan man velge hvilke typer brukere som skal se applikasjonene.



Går inn på *myapps* og se applikasjonen ble lagt til.



Tabell 49 - AAD Access Panel

5.6 Windows Virtual Desktop Bruerguide

Først må brukeren som skal bruke WVD ha blitt gitt tilgang gjennom kommandoen:

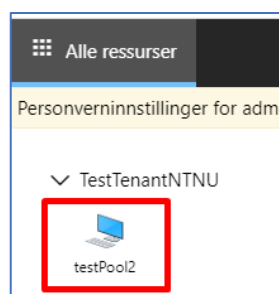
```
Add-RdsAppGroupUser <tenantname> <hostpoolname> <appgroupname> -UserPrincipalName <userupn>
```

Siden brukerne i dette miljøet er Cloud only, må det påses at de bruker såkalt Kerberos autentisering. Dersom brukerne er opprettet **før** AADDS ble implementert, må brukerne endre passordet sitt, og deretter vente 30 min. Dette er for å endre autentiseringstypen til Kerberos. Brukere som er opprettet **etter** AADDS, har allerede riktig form for autentisering. Dette er beskrevet i kapittel 4.6.2⁶⁰

5.6.1 WVD web-client

Gå inn på <https://rdweb.wvd.microsoft.com/webclient/index.html> og logg inn

Når du er pålogget vil du se navnet på WVD-tenanten, deretter ressurser under denne. For øyeblikket er det kun «testPool2», som er et eksternt skrivebord bruker kan koble til

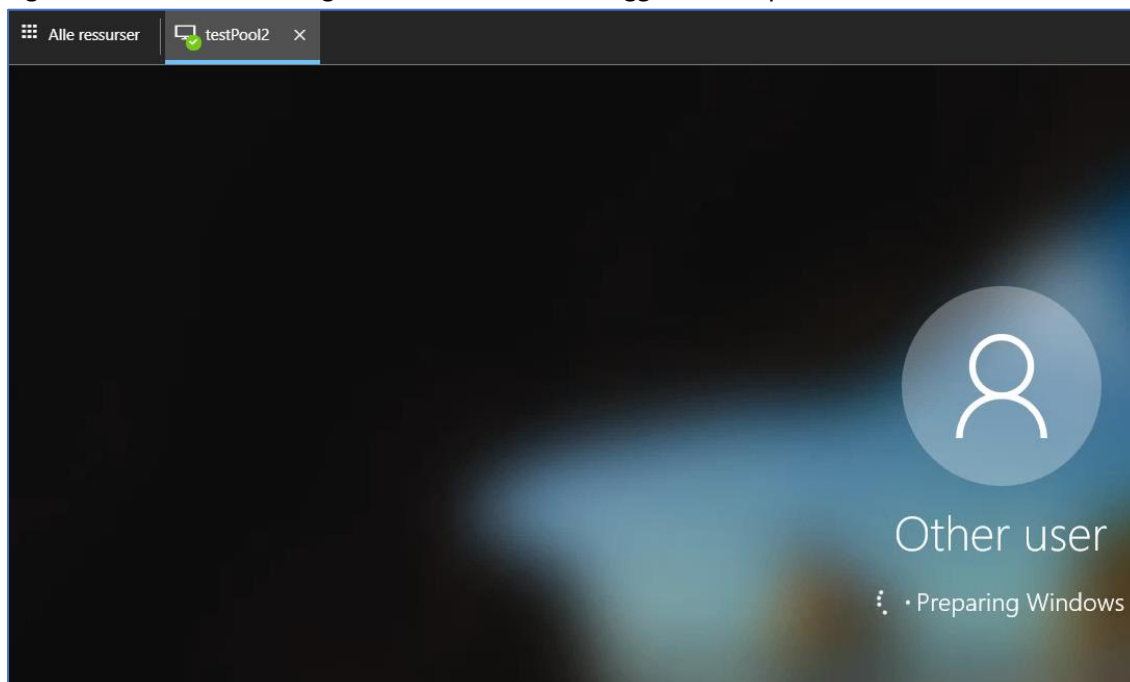


Dersom du man kobler til, vil man kunne bruke skriverbordet direkte i nettleseren.

Tabell 50 - WVD web client

⁶⁰ Endre til kerberos autentisering

Figuren under viser brukergrensesnittet for en innlogget bruker på en maskin.



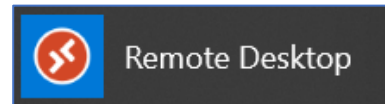
Figur 41 - Brukergrensesnitt WVD

5.6.2 WVD lokal app

I stedet for å bruke nettleser er det også mulig å ta i bruk en skrivebords-klient.

Last ned applikasjonen fra linken <https://go.microsoft.com/fwlink/?linkid=2068602>

Installer Remote Desktop



Logg inn via knappen «abonner»

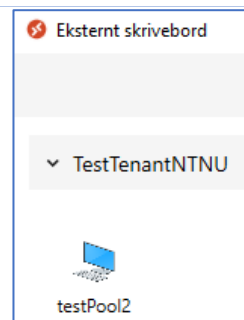
La oss komme i gang

Abonner for å få tilgang til administrerte apper og skrivebord fra organisasjonen din. Disse ressursene vil være tilgjengelige fra denne klienten og Start-menyen.

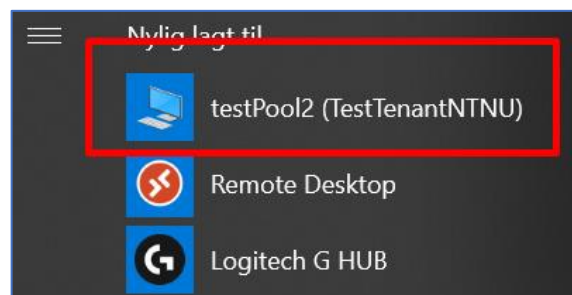
Person vern innstillingene for eksterne ressurser er forhåndsinnstilt av

Abonner

Følgende visning lar bruker koble til det eksterne skrivebordet (via RDP)



I tillegg kan man senere koble til det skrivebordet via snarveier slik som dette:

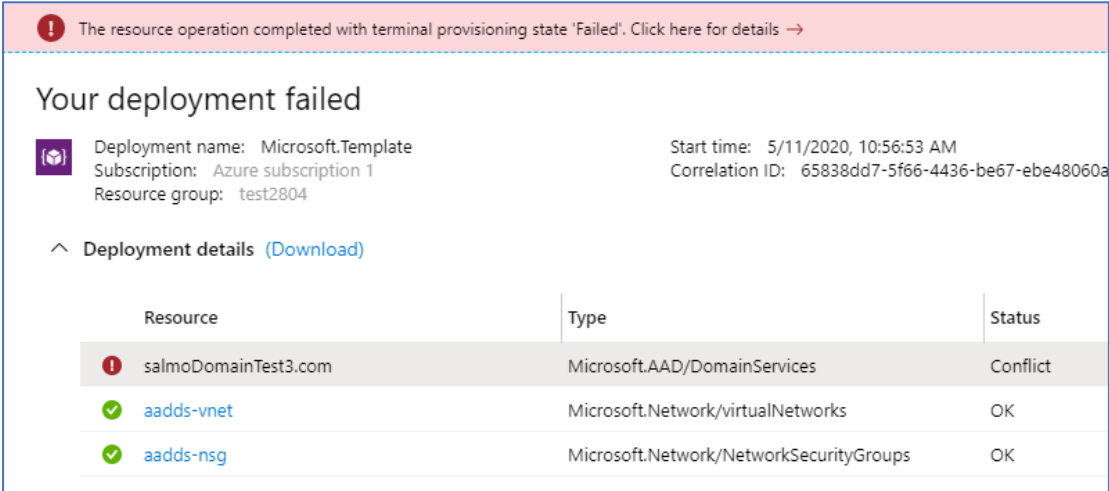


Tabell 51 - WVD lokal app

6 Problemer/bugs

6.1 Problem ved opprettelse av AADDs

Noen ganger, under opprettelsen av AADDs fra template ville utrullingene feile.



The resource operation completed with terminal provisioning state 'Failed'. Click here for details →

Your deployment failed

Deployment name: Microsoft.Template
Subscription: Azure subscription 1
Resource group: test2804

Start time: 5/11/2020, 10:56:53 AM
Correlation ID: 65838dd7-5f66-4436-be67-ebe48060a

Deployment details (Download)


Resource	Type	Status
salmoDomainTest3.com	Microsoft.AAD/DomainServices	Conflict
aadds-vnet	Microsoft.Network/virtualNetworks	OK
aadds-nsg	Microsoft.Network/NetworkSecurityGroups	OK

Figur 42 - Feilmelding: appid


«The service principal with appId '2565bd9d-da50-47d4-8b85-4c97f669dc36' could not be found in the Azure Active Directory tenant. Please retry the operation.»

Dette betyr at Tjenesten AADDs ikke klarer å finne applikasjonen med app-iden i Azure AD-tenanten. Slik løses det ifølge Microsoft Docs⁶¹:

Gå inn på Azure AD i Azure-portalen, og deretter *Enterprise Applications*

 Enterprise applications

All Applications

 All applications

Bruk tabellen til å se om det eksisterer apper med tilsvarende app-id

2565bd9d-da50-47d4-8b85-4c97f669dc36
443155a6-77f3-45e3-882b-22b3a8d431fb
abba844e-bc0e-44b0-947a-dc74e5d09022
d87dcbc6-a371-462e-88e3-28ad15ec4e64

Tabell 52 - Løsning for manglende appid

Den første App-Iden er direkte korrelert med feilmeldingen. Dersom kun den første App-iden ikke eksisterer følg delkapittel 6.1.1, dersom noen av de andre ikke eksisterer følg 6.1.2. Dersom nummer en og noen av de andre ikke eksisterer, følg både 6.1.1. og 6.1.2.

⁶¹ (Microsoft, Known issues: Service principal alerts in Azure Active Directory Domain Services, 2019)

6.1.1 Recreate a missing service principal

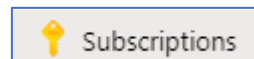
Dersom app-ID lik `2565bd9d-da50-47d4-8b85-4c97f669dc36` ikke eksisterer i Azure AD, må Azure AD PowerShell brukes for å opprette denne på nytt. Prosjektgruppen brukte skriptet `recreateSP.ps1`⁶² for dette.

Etter kjøring av skriptet, vil Azure AD DS domenet automatisk oppdatere seg innenfor to timers tid.

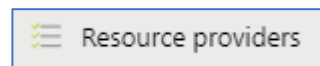
6.1.2 Re-register the Microsoft AAD namespace

Dersom app-ID lik `443155a6-77f3-45e3-882b-22b3a8d431fb`, `abba844e-bc0e-44b0-947a-dc74e5d09022`, eller `d87dcbc6-a371-462e-88e3-28ad15ec4e64` ikke eksisterer, utfør følgende steg:

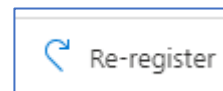
Gå inn på *Subscriptions* via Azure-portalen.



Gå inn på *Resource providers*



Finn fram *Microsoft.AAD* og velg *Re-register*

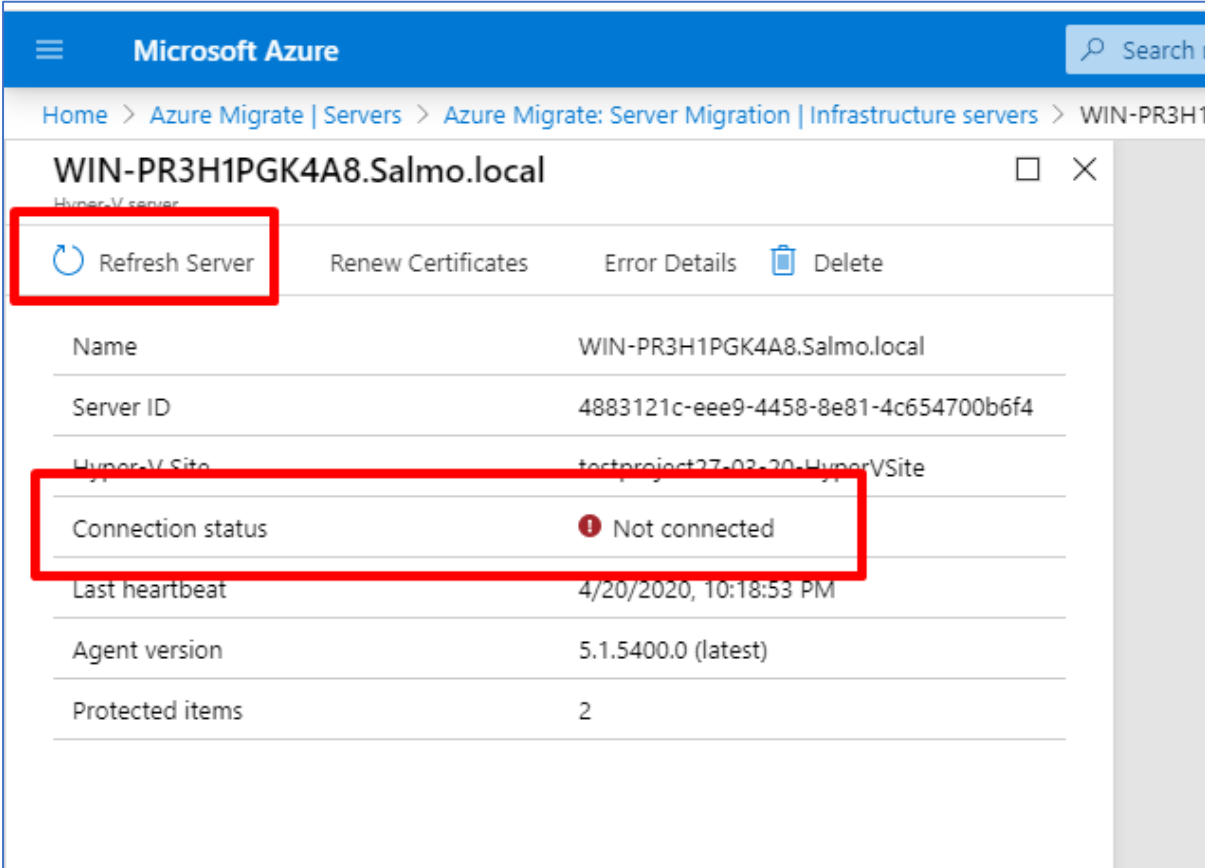


Tabell 53 - Reregistrere namespace

Etter omtrent to timer vil feilmeldingen på Azure AD Domain Services gå vekk.

⁶² (Microsoft, `recreateSP.ps1`, 2019)

6.2 Tapt tilkobling til Hyper-V serveren



The screenshot shows the Microsoft Azure portal interface. The breadcrumb navigation path is: Home > Azure Migrate | Servers > Azure Migrate: Server Migration | Infrastructure servers > WIN-PR3H1. The main heading is 'WIN-PR3H1PGK4A8.Salmo.local' with a 'Hyper-V server' sub-label. Below the heading are four action buttons: 'Refresh Server' (highlighted with a red box), 'Renew Certificates', 'Error Details', and 'Delete'. A table below displays server details:

Name	WIN-PR3H1PGK4A8.Salmo.local
Server ID	4883121c-eee9-4458-8e81-4c654700b6f4
Hyper-V Site	testproject27-02-20-HyperVSite
Connection status	Not connected
Last heartbeat	4/20/2020, 10:18:53 PM
Agent version	5.1.5400.0 (latest)
Protected items	2

Figur 43 - Feilmelding: Tapt tilkobling

Dette kan skje dersom serveren går ned, eller mister nettverkstilkobling etter at serveren er lagt til i Azure Migrate.

Løses ved å oppdatere serveren, som kan ta 2-5 minutter avhengig av lasten på serveren. Dette var ofte nødvendig for prosjektgruppen, da serveren var skrudd av om nettene. Vanligvis ville serveren komme tilbake automatisk, men det tok kortere tid å oppdatere serveren manuelt.

6.3 Feilmelding som følge av tidligere migrering

En feilmeldingen var beslektet med et tidligere forsøk på migrering:

```
70094

Protection couldn't be enabled for the virtual machine.

Provider error code: 31404

Provider error message:
  Enable Protection for virtual machine 'Filtjener' failed as this machine has some stale settings from the previous protection.

Provider error possible causes:
  The replication settings from a previous protection attempt is not cleaned up from the on-premises virtual machine.

Provider error recommended action:
  Clean up the stale protection settings by following the instructions at (https://aka.ms/clean\_hyperv\_settings).

Protection can't be enabled with the virtual machine in its current state. Check the Provider errors for more information.

Fix any issues in the Event Viewer logs (Applications and Service Logs - MicrosoftAzureRecoveryServices) on the Hyper-V host server. For details see
http://aka.ms/asr\_troubleshoot/#troubleshoot-on-premises-hyper-v-issues. If this virtual machine is enabled for replication on the Hyper-V host, disable this setting. Then try to enable
protection again.

http://aka.ms/asr\_troubleshoot/#troubleshoot-on-premises-hyper-v-issues
https://aka.ms/clean\_hyperv\_settings

3/31/2020, 2:20:21 PM
```

Figur 44 - Feilmelding: tidligere migrering

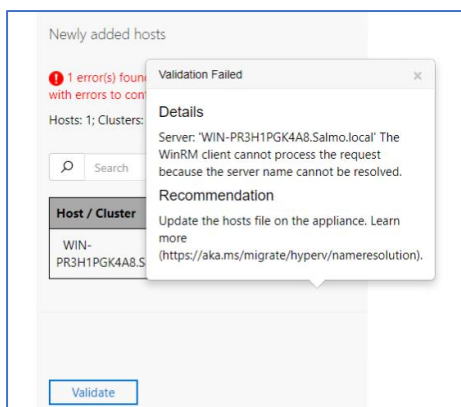
Dette kommer som en følge av at det er forsøkt å skru på protection for VM-en, men den er allerede forbundet med et annet forsøk på migrering. Slik kan det løses:

1. Skru av replikasjon av VM-ene på Hyper-V. Dette gjøres ved å:
 - a. Gå inn på Hyper-V Manager
 - b. Høyreklikk på VM-ene som replikeres (om noen) > replication > disable

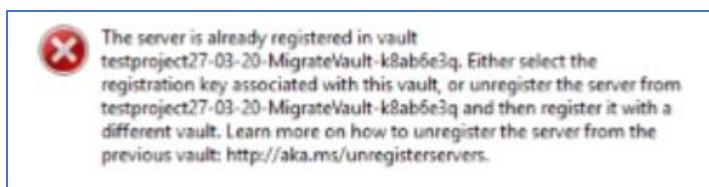
6.4 Registrert med annet recovery services vault

Underveis i prosjektutførelsen fikk prosjektgruppen feilmelding om at Hyper-V serveren allerede var knyttet til et *Recovery Services Vault* (i forbindelse med Azure Migrate). Dette var forbundet med en tidligere testmigrering. Løsningen var å sku av replikasjon på VM-ene, og deretter avregistrere Hyper-V serveren på Azure, ifølge Microsoft Docs⁶³. Til slutt kunne man serveren til det riktige Migrate-prosjektet igjen.

Følgende er en feilmelding prosjektgruppen inntraff, fordi det ble glemt å legge til riktig nettverkskonfigurasjon på VM-en.



Figur 45 - feilmelding nettverkskonfigurasjon



Figur 46 - Feilmelding for hyper-v host

Dersom feilmeldingen «*Unsupported boot type vises*» så kan dette fikses ved å endre BIOS, noe som azure migrate kan håndtere selv, ifølge docs.

Issue	Fix
Unsupported boot type	Azure doesn't support VMs with an EFI boot type. We recommend that you convert the boot type to BIOS before you run a migration. You can use Azure Migrate Server Migration to handle the migration of such VMs. It will convert the boot type of the VM to BIOS during the migration.

Figur 47 - Feilmelding: *Unsupported boot type*

Her ble det derimot feilmeldingen «*requires Visual Studio Subscription*»

Requires a Microsoft Visual Studio subscription	The machine is running a Windows client operating system, which is supported only through a Visual Studio subscription.
---	---

Figur 48 - Feilmelding: *visual studio*

Dette kan fikses ved å skaffe en Visual Studio Subscription, men dette er ikke noe som ble disponert under prosjekt-gjennomføringen.

⁶³ (Microsoft, Remove servers and disable protection, 2019)

7 Utfaset funksjonalitet.

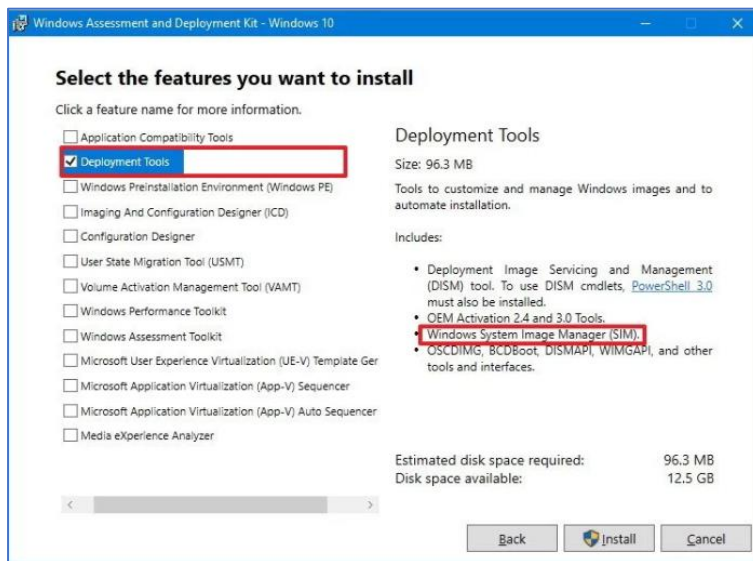
I løpet av prosjektet var det en rekke verktøy og teknologier som ble brukt, men som ette hvert ble erstattet av andre verktøy eller ble rett og slett unødvendig for prosjektets endelige løsning.

7.1 Automatisering av Windows 10 installasjon

For å kunne rulle ut flere maskiner samtidig og slippe å gå gjennom installasjonsprosessen på hver enkelt maskin har denne prosessen blitt automatisert. I dette avsnittet beskrives prosessen og nødvendige verktøy.

Nødvendige ressurser for automatisering:

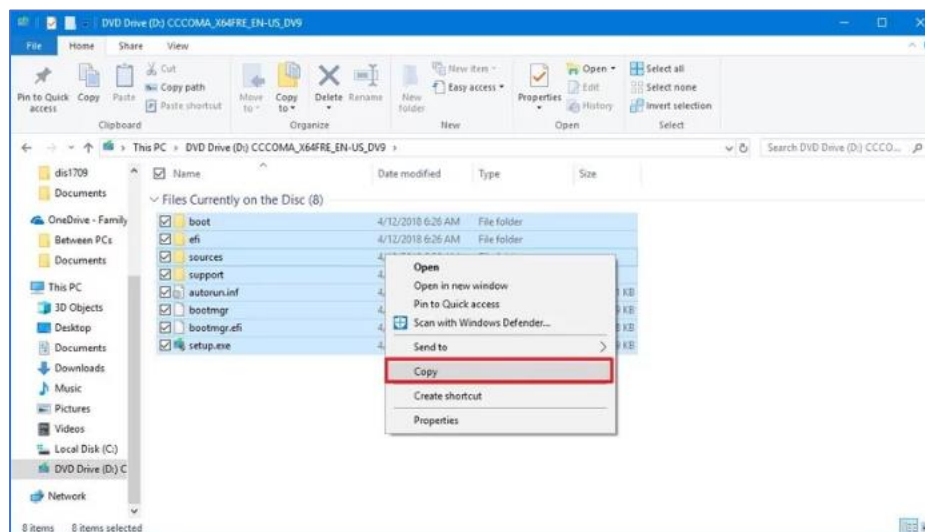
- Windows 10 iso-fil
- Windows ADK (1809)



Figur 49 - Windows ADK

7.1.1 Klargjøring av ISO-fil

Første steg er å kopiere alle filene fra Windows iso-filen inn i en ny mappe. I mappen *sources* skal det finnes en fil ved navn *install.wim*.



Figur 50 - ISO-filer

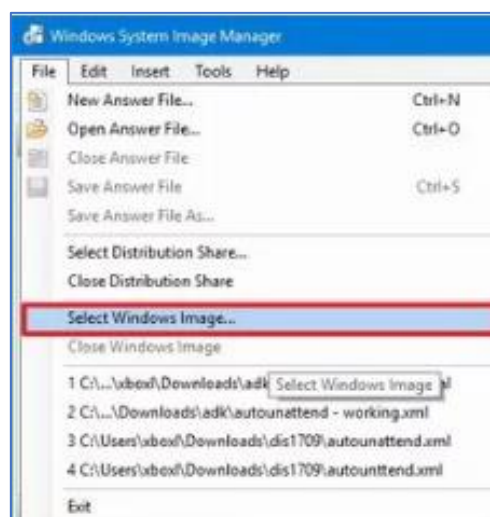
Neste steg er å opprette filen som gjør installasjonen automatisk. Det gjøres i programmet *Windows System Image Manager*. Det er her viktig å vite detaljer som språk på *install.wim*. Hvis språk her er satt til en-US må språket i svarfilen settes tilsvarende. For å finne disse detaljene kjøres følgende kommando via *cmd*:

```
C:\Windows\system32>dism /Get-WimInfo /WimFile:G:\sources\install.wim /index:8
```

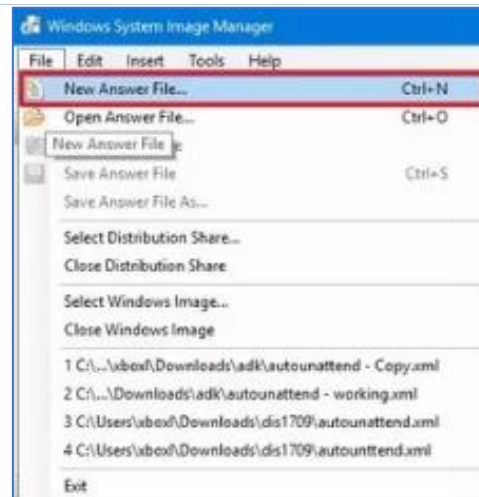
Figur 51 - wim info

Starter med konfigurasjon av svarfilen:

Her velges wim-filen som nevnt tidligere (*install.wim*) og «windows pro» i prompten som kommer opp. Hvis spørsmål om å opprette en katalog fil kommer velges *yes*.



Neste steg er å opprette en ny svarfil



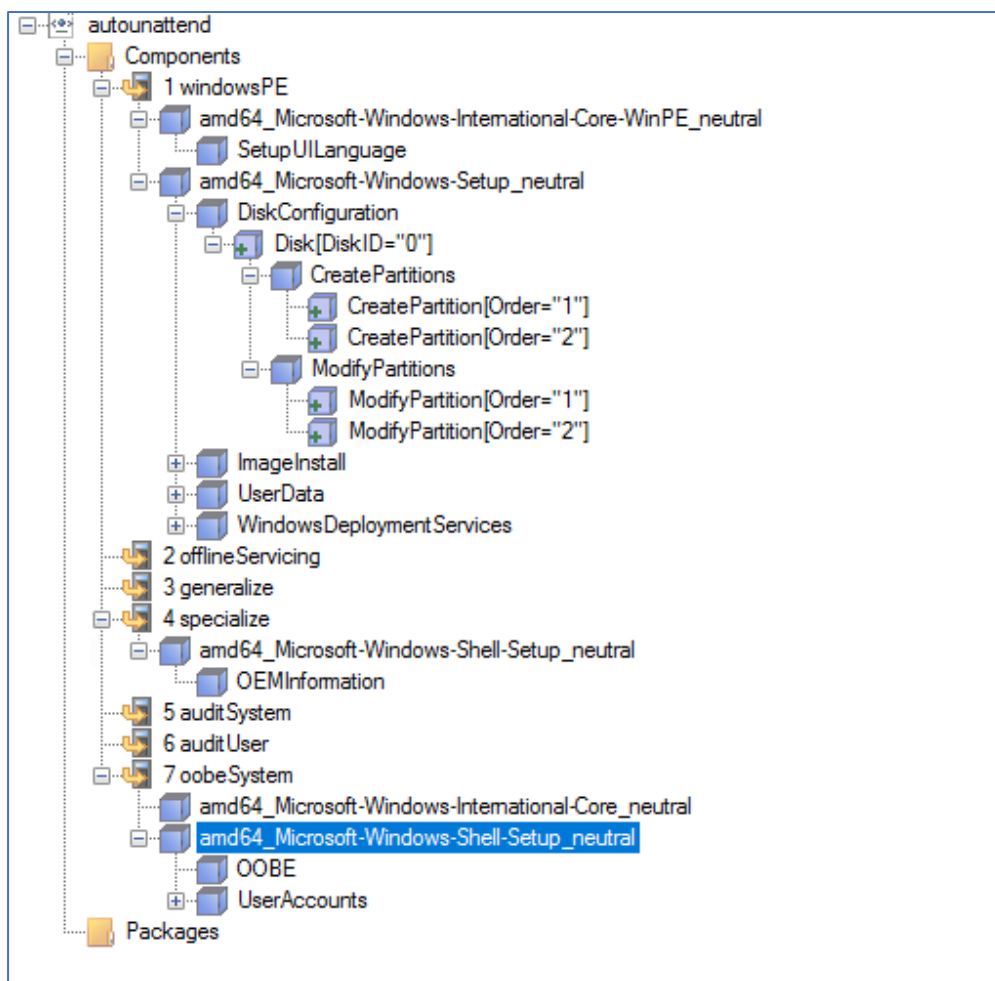
Tabell 54 - Konfigurering av svarfil

Begynner å konfigurere svarfilen med de forskjellige komponentene. Alle steg vil ikke bli dokumentert her, men de komponentene som er brukt legges ved under i tillegg til selve svarfilen som blir opprettet etter konfigurasjon i XML format.

Alle disse komponentene har felter som skal fylles ut med nødvendig informasjon. Svarfilen som konfigureres nå kan bare brukes hvis maskinen som skal ta den i bruk er satt opp til å bruke BIOS.

Komponenter:

- 1. WindowsPE
 - amd64_Microsoft-Windows-Internal-Core-WinPE_neutral
 - SetupUILanguage
 - amd64_Microsoft-Windows-Setup_neutral
 - DiskConfiguration
 - ImageInstall
 - UserData
 - WindowsDeploymentService
- 4. Specialize
 - amd64_Microsoft-Windows-Shell-Setup_neutral
 - OEMInformation
- 7. oobeSystem
 - amd64_Microsoft-Windows-Internal-Core_neutral
 - amd64_Microsoft-Windows-Shell-Setup_neutral
 - OOBE
 - UserAccounts



Figur 52 - Komponenter i svarfil

Når svarfilen er ferdig lagres den i mappen hvor resten av Windows-filene er lagret. Svarfilen heter her *autounattend.xml*

Name	Date modified	Type	Size
boot	03.03.2020 10.32	File folder	
efi	03.03.2020 10.32	File folder	
sources	03.03.2020 10.35	File folder	
support	03.03.2020 10.35	File folder	
autorun.inf	30.10.2018 00.32	Setup Information	1 KB
autounattend.xml	03.03.2020 10.27	XML File	8 KB
bootmgr	30.10.2018 00.32	File	399 KB
bootmgr.efi	30.10.2018 00.32	EFI File	1 419 KB
setup.exe	30.10.2018 00.32	Application	81 KB

Figur 53 - Isofiler med ny svarfil

Neste steg er å pakke sammen alle filene i mappen til en iso-fil. Det finnes mange måter å gjøre dette på, men her benyttes programmet *Deployment and Imaging Tools Environment* som følger med Windows ADK. I denne konsollen kjøres kommandoen:

```
oscdimg.exe -m -o -u2 -udfver102 -  
bootdata:2#p0,e,bd:\iso_files\boot\etfsboot.com#pEF,e,bd:\iso_files\efi\mic  
rosoft\boot\efisys.bin d:\iso_files d:\autoUnattended.iso
```

Figur 54 - Kode for å bygge ny isofil

Denne kommandoen oppretter en iso-fil med svarfilen pakket inn. Navnet på denne kan defineres selv, men her kalles den *autoUnattended.iso*. Hvis alt er gjort riktig er det bare å starte en tom vm, koble til iso-filen og maskinen skal installere Windows automatisk.

Figurer

Figur 1 - Ledetekst/CMD	6
Figur 2 – PowerShell	6
Figur 3 - JSON	6
Figur 4 – Hvordan tjenester videreføres.....	10
Figur 5 - Server-spesifikasjoner skapt med programvaren Speccy	11
Figur 6 - Oppsett av Hyper-V server.....	12
Figur 7 - Ressursgruppen "testmiljø"	13
Figur 8 - Active Directory struktur	17
Figur 9 - GPO	17
Figur 10 - Testfil-kommando.....	17
Figur 11 - Testfiler	17
Figur 12 - On-premises før migrering.....	26
Figur 13 - Azure miljø etter utførelse.....	26
Figur 14 - Azure Migrate Appliance	27
Figur 15 - Forhold Template – Parametere – Key Vault.....	29
Figur 16 - Nettverkskort på en vpnserver	31
Figur 17 - Ressurser som rulles ut gjennom main.json	34
Figur 18 - lenker til template-filer	35
Figur 19 - Lenker til parameter-filer.....	35
Figur 20 - Deployment av MSSQL fra main.json	35
Figur 21 - Rulle ut templer-fil.....	36
Figur 22 - Nettverks plasseringen vises i filutforsker.....	39
Figur 23 - Shared access signature.....	39
Figur 24 - Azure managed domain.....	40
Figur 25 - Domain running	40
Figur 26 - Configuration steps.....	41
Figur 27 - Oversikt migrering MySQL	60
Figur 28 - Recovery vaults	70
Figur 29 - Backup policy	70
Figur 30 - Ingen offentlig ip-adresse	72
Figur 31 - Utsnitt fra skript.....	75
Figur 32 - Image i ressursgruppe.....	75
Figur 33 - Overordnet prosess WVD	77
Figur 34 - Oversikt custom image	82
Figur 35 - Oversikt WVD sesjonsvert.....	83
Figur 36 - WVD maskiner	84
Figur 37 - Joindomain.....	84
Figur 38 - Oversikt Azure Runbooks.....	90
Figur 39 - Oversikt Log Analytics	95
Figur 40 - Oversikt Azure Update Management	99
Figur 41 - Brukergrensesnitt WVD	109
Figur 42 - Feilmelding: appid.....	111
Figur 43 - Feilmelding: Tapt tilkobling.....	113
Figur 44 - Feilmelding: tidligere migrering.....	114
Figur 45 - feilmelding nettverkskonfigurasjon	115
Figur 46 - Feilmelding for hyper-v host.....	115

Figur 47 - Feilmelding: Unsupported boot type.....	115
Figur 48 - Feilmelding: visual studio	115
Figur 49 - Windows ADK	116
Figur 50 - ISO-filer	117
Figur 51 - wim info	117
Figur 52 - Komponenter i svarfil.....	119
Figur 53 - Isofiler med ny svarfil.....	119
Figur 54 - Kode for å bygge ny isofil.....	120

Tabeller

Tabell 1 - Begreper/forkortelser	6
Tabell 2 - Oppsett av ADDS	16
Tabell 3 - Active Directory	17
Tabell 4 - SMB-share	20
Tabell 5 - installasjon MySQL	21
Tabell 6 - MySQL test-database	21
Tabell 7 - MSSQL installasjon	22
Tabell 8 - Northwind database.....	23
Tabell 9 - Enkel Webserver	23
Tabell 10 - Verktøy brukt	24
Tabell 11 - Laste ned MigrateAppliance	28
Tabell 12 - Klargjøring av Hyper-V server	28
Tabell 13 - Key Vault	29
Tabell 14 - Legg til Secret	30
Tabell 15 - Referere til Secret	31
Tabell 16 - Internal switching på vm	32
Tabell 17 - Opprette ressursgruppe med PowerShell.....	33
Tabell 18 - Utrulling main.json	36
Tabell 19 - App Service.....	38
Tabell 20 - Kereberos autentisering.....	42
Tabell 21 - Discovery og Assessment	46
Tabell 22 - Discovery for Migrate.....	49
Tabell 23 - Replication.....	53
Tabell 24 - Test Migration	56
Tabell 25 - Migrering Virtuelle Maskiner	58
Tabell 26 - Tilkobling til database	59
Tabell 27 - Migrere SQLdb	62
Tabell 28 - Migrere MSSQL database.....	66
Tabell 29 - AzCopy.....	70
Tabell 30 - Ekstern ip-adresse for VM.....	73
Tabell 31 - Klargjøring VM-image.....	74
Tabell 32 - Test av image.....	76
Tabell 33 - TenantCreator application role	79
Tabell 34 - WVD tenant.....	80
Tabell 35 - Service Principal Role og Role Assignments	81
Tabell 36 – Sesjonsverter med ARM	82
Tabell 37 – Legge til flere sesjonsverter.....	85
Tabell 38 - App groups	86
Tabell 39 - Oppsett Recovery Vault	87
Tabell 40 - Sikkerhetskopiering av vm	88
Tabell 41 - Konfigurasjon for backup retention	89
Tabell 42 - Azure Runbook.....	92
Tabell 43 - Schedule og runbook.....	94
Tabell 44 - Log Analytics.....	96

Tabell 45 - Log Analytics Konfigurasjon	98
Tabell 46 - Azure Update Management	101
Tabell 47 - Legge til maskiner i Update Manager	102
Tabell 48 - Rulle ut oppdateringer i Update Manager	104
Tabell 49 - AAD Access Panel	107
Tabell 50 - WVD web client	108
Tabell 51 - WVD lokal app	110
Tabell 52 - Løsning for manglende appid	111
Tabell 53 - Reregistrere namespace	112
Tabell 54 - Konfigurering av svarfil	118

Referanser

- Brady, C. N. (2018, April 25). *Install SQL Server 2017 PowerShell script*. Retrieved from just another windows noob ? : <https://www.niallbrady.com/2018/04/25/install-sql-server-2017-powershell-script-download-it-here/>
- Brasø, E. A., & Aalrust, E. (2020). *aadds.json*. Retrieved from Bachelor-Sopra-Steria-2020: <https://github.com/Zypheros/Bachelor-Sopra-Steria-2020/blob/master/JSON/aadds/aadds.json>
- Brasø, E. A., & Aalrust, E. (2020). *addPublicIPtoNIC.json*. Retrieved from Bachelor-Sopra-Steria-2020: <https://github.com/Zypheros/Bachelor-Sopra-Steria-2020/blob/master/JSON/addPublicIP/addPublicIPtoNic.json>
- Brasø, E. A., & Aalrust, E. (2020). *addSASToken.ps1*. Retrieved from Bachelor-Sopra-Steria-2020: <https://github.com/Zypheros/Bachelor-Sopra-Steria-2020/blob/master/PowerShell/Storage%20Account/addSASToken.ps1>
- Brasø, E. A., & Aalrust, E. (2020). *assignRolesAndUsersWvd.ps1*. Retrieved from Bachelor-Sopra-Steria-2020: <https://github.com/Zypheros/Bachelor-Sopra-Steria-2020/blob/master/PowerShell/Windows%20Virtual%20Desktop/assignRolesAndUsersWVD.ps1>
- Brasø, E. A., & Aalrust, E. (2020). *client_make.sh*. Retrieved from Bachelor-Sopra-Steria-2020: https://github.com/Zypheros/Bachelor-Sopra-Steria-2020/blob/master/Annet/Bash%20scripts/client_make.sh
- Brasø, E. A., & Aalrust, E. (2020). *createImageFromVM.ps1*. Retrieved from Bachelor-Sopra-Steria-2020: <https://github.com/Zypheros/Bachelor-Sopra-Steria-2020/blob/master/PowerShell/Azure/createImageFromVM.ps1>
- Brasø, E. A., & Aalrust, E. (2020). *customBackupPolicyVM.json*. Retrieved from Bachelor-Sopra-Steria-2020: <https://github.com/Zypheros/Bachelor-Sopra-Steria-2020/blob/master/JSON/Backup/customBackupPolicyVM.json>
- Brasø, E. A., & Aalrust, E. (2020). *deployToRG.ps1*. Retrieved from Bachelor-Sopra-Steria-2020: <https://github.com/Zypheros/Bachelor-Sopra-Steria-2020/blob/master/PowerShell/Deploy%20ARM/deployToRG.ps1>
- Brasø, E. A., & Aalrust, E. (2020). *Designrapport*.

- Brasø, E. A., & Aalrust, E. (2020). *DownloadAppliance.ps1*. Retrieved from Bachelor-Sopra-Steria-2020: <https://github.com/Zypheros/Bachelor-Sopra-Steria-2020/blob/master/PowerShell/Deploy%20ARM/DownloadAppliance.ps1>
- Brasø, E. A., & Aalrust, E. (2020). *Forstudierapport*.
- Brasø, E. A., & Aalrust, E. (2020). *installMySQL.ps1*. Retrieved from Bachelor-Sopra-Steria-2020: <https://github.com/Zypheros/Bachelor-Sopra-Steria-2020/blob/master/PowerShell/installMySQL.ps1>
- Brasø, E. A., & Aalrust, E. (2020). *main.json*. Retrieved from Bachelor-Sopra-Steria-2020: <https://github.com/Zypheros/Bachelor-Sopra-Steria-2020/blob/master/JSON/main.json>
- Brasø, E. A., & Aalrust, E. (2020). *mapShare.ps1*. Retrieved from Bachelor-Sopra-Steria-2020: <https://github.com/Zypheros/Bachelor-Sopra-Steria-2020/blob/master/PowerShell/Storage%20Account/mapShare.ps1>
- Brasø, E. A., & Aalrust, E. (2020). *mySQLdb.json*. Retrieved from Bachelor-Sopra-Steria-2020: <https://github.com/Zypheros/Bachelor-Sopra-Steria-2020/tree/master/JSON/Azure%20MySQL%20Databases>
- Brasø, E. A., & Aalrust, E. (2020). *newTenantUser.ps1*. Retrieved from Bachelor-Sopra-Steria-2020: <https://github.com/Zypheros/Bachelor-Sopra-Steria-2020/blob/master/PowerShell/Windows%20Virtual%20Desktop/newTenantUser.ps1>
- Brasø, E. A., & Aalrust, E. (2020). *newWebApp*. Retrieved from Bachelor-Sopra-Steria-2020: <https://github.com/Zypheros/Bachelor-Sopra-Steria-2020/blob/master/JSON/Webapp/newWebApp.json>
- Brasø, E. A., & Aalrust, E. (2020). *OpenVPNsetup.sh*. Retrieved from Bachelor-Sopra-Steria-2020: <https://github.com/Zypheros/Bachelor-Sopra-Steria-2020/blob/master/Annet/Bash%20scripts/OpenVPNsetup.sh>
- Brasø, E. A., & Aalrust, E. (2020). *recoveryVault.json*. Retrieved from Bachelor-Sopra-Steria-2020: <https://github.com/Zypheros/Bachelor-Sopra-Steria-2020/blob/master/JSON/Backup/recoveryVault.json>
- Brasø, E. A., & Aalrust, E. (2020). *resourceGroup.json*. Retrieved from Bachelor-Sopra-Steria-2020: <https://github.com/Zypheros/Bachelor-Sopra-Steria-2020/blob/master/JSON/resourceGroups/resourceGroup.json>
- Brasø, E. A., & Aalrust, E. (2020). *retentionPolicy.ps1*. Retrieved from Bachelor-Sopra-Steria-2020: <https://github.com/Zypheros/Bachelor-Sopra-Steria-2020/blob/master/PowerShell/Azure/retentionPolicy.ps1>
- Brasø, E. A., & Aalrust, E. (2020). *rollbackToBackup.ps1*. Retrieved from Bachelor-Sopra-Steria-2020: <https://github.com/Zypheros/Bachelor-Sopra-Steria-2020/blob/master/PowerShell/Azure/rollbackToBackup.ps1>
- Brasø, E. A., & Aalrust, E. (2020). *Skript*.
- Brasø, E. A., & Aalrust, E. (2020). *sqlDB.json*. Retrieved from Bachelor-Sopra-Steria-2020: <https://github.com/Zypheros/Bachelor-Sopra-Steria-2020/tree/master/JSON/Azure%20SQL%20Databases>

- Brasø, E. A., & Aalrust, E. (2020). *storageAccount'WihtFileShare.json*. Retrieved from Bachelor-Sopra-Steria-2020: <https://github.com/Zypheros/Bachelor-Sopra-Steria-2020/blob/master/JSON/Storage/storageAccountWithFileShare.json>
- Brasø, E. A., & Aalrust, E. (2020). *viewLTRBackups.ps1*. Retrieved from Bachelor-Sopra-Steria-2020: <https://github.com/Zypheros/Bachelor-Sopra-Steria-2020/blob/master/PowerShell/Azure/viewLTRBackups.ps1>
- Brasø, E. A., & Aalrust, E. (2020). *webAppParameters.json*. Retrieved from Bachelor-Sopra-Steria-2020: <https://github.com/Zypheros/Bachelor-Sopra-Steria-2020/blob/master/JSON/Webapp/webAppParameters.json>
- Brasø, E. A., & Aalrust, E. (2020). *win16serv.pdf*.
- Chocolatey Software, Inc. (2020). *Chocolatey Software*. Retrieved from chocolatey.org: <https://chocolatey.org/>
- DigitalOcean. (2018, Mai 24). *How To Set Up an OpenVPN server*. Retrieved from digitalocean.com: <https://www.digitalocean.com/community/tutorials/how-to-set-up-an-openvpn-server-on-ubuntu-18-04>
- Microsoft. (2018, November 2). *instnwnd.sql*. Retrieved from sql-server-samples: <https://github.com/microsoft/sql-server-samples/blob/master/samples/databases/northwind-pubs/instnwnd.sql>
- Microsoft. (2018, 03 16). Microsoft Azure docs. *Start a runbook in Azure Automation* .
- Microsoft. (2019, Juli 10). *Azure Monitor overview*. Retrieved from Microsoft Docs: <https://docs.microsoft.com/en-us/azure/azure-monitor/overview>
- Microsoft. (2019, August 22). *Choose the right deployment option in Azure SQL*. Retrieved from Microsoft Docs: <https://docs.microsoft.com/en-us/azure/sql-database/sql-database-paas-vs-sql-server-iaas>
- Microsoft. (2019, Desember 18). *Grant limited access to Azure Storage resources using shared access signatures (SAS)*. Retrieved from Microsoft Docs: <https://docs.microsoft.com/en-us/azure/storage/common/storage-sas-overview#sas-token>
- Microsoft. (2019, September 20). *Known issues: Service principal alerts in Azure Active Directory Domain Services*. Retrieved from Microsoft Docs: <https://docs.microsoft.com/en-us/azure/active-directory-domain-services/alert-service-principal>
- Microsoft. (2019, Septemer 20). *recreateSP.ps1*. Retrieved from Microsoft Docs: <https://docs.microsoft.com/en-us/azure/active-directory-domain-services/alert-service-principal#recreate-a-missing-service-principal>
- Microsoft. (2019, Juni 18). *Remove servers and disable protection*. Retrieved from Microsoft Docs: <https://docs.microsoft.com/en-us/azure/site-recovery/site-recovery-manage-registration-and-protection#unregister-a-connected-configuration-server>
- Microsoft. (2020, 03 23). *Azure Migrate Appliance Architecture*. Retrieved from Microsoft Docs: <https://docs.microsoft.com/en-us/azure/migrate/migrate-appliance-architecture>
- Microsoft. (2020, 02 27). Azure Update Manager.

- Microsoft. (2020, April 30). *mainTemplate.json*. Retrieved from RDS-Templates:
<https://github.com/Azure/RDS-Templates/blob/master/wvd-templates/Create%20and%20provision%20WVD%20host%20pool/mainTemplate.json>
- Microsoft. (2020). *Prepare for assessment and migration of Hyper-V VMs to Azure*. Retrieved from Microsoft Docs: <https://docs.microsoft.com/en-us/azure/migrate/tutorial-prepare-hyper-v>;
<https://github.com/Zypheros/Bachelor-Sopra-Steria-2020/blob/master/PowerShell/Deploy%20ARM/MicrosoftAzureMigrate-Hyper-V.ps1>
- Microsoft. (2020, 04 23). *Workspace Mappings*. Retrieved from <https://docs.microsoft.com>:
<https://docs.microsoft.com/en-us/azure/automation/how-to/region-mappings>
- MySQLTutorial. (2020). *MySQL Sample Database*. Retrieved from MySQL Tutorial:
<https://www.mysqltutorial.org/mysql-sample-database.aspx>
- Pablos, J. J. (2011, 06 06). *Unattended, A Windows Deployment system*. Retrieved from unattended.sourceforge.net: <http://unattended.sourceforge.net/installers.php>
- The Apache Software Foundation. (2020). *Apache HTTP Server Documentation*. Retrieved from Apache HTTP Server Project: <https://httpd.apache.org/docs/>

Vedlegg

Følgende filer er vedlagt:

- *Win16serv.pdf* – Beskriver maskinvaren som ble tatt i bruk på serveren. Generert ved hjelp av Speccy
- *Skript.docx* – Inneholder alle skript som brukes i sammenheng med prosjektet.

