

Tekninen vikasietoisuussuunnitelma palkanlaskentajärjestelmästä

Jani Kujo

Tekijä(t) Jani Kujo	
Koulutusohjelma Tietojenkäsittelyn koulutusohjelma	
Opinnäytetyön otsikko Tekninen vikasietoisuussuunnitelma palkanlaskentajärjestelmästä	Sivu- ja liitesivumäärä 30
Opinnäytetyön otsikko englanniksi Technical disaster recovery plan for payroll system	
<p>Opinnäytetyön päätavoite on toteuttaa replikoiva SQL alwaysOn järjestelmä. Opinnäytetyö on saatu toimeksiantona ja sen tarkoitus on selvittää yritykselle, miten tämä ratkaisu tehdään ja miten he voivat jatkossa ottaa kyseisen ratkaisun käyttöön omassa tuotantoympäristössään.</p> <p>Opinnäytetyössä toteutetaan SQL AlwaysOn toiminnon käyttöönotto Hyper-V virtuaaliympäristössä, jonka jälkeen asennukset suoritetaan kyseisen yrityksen Citrix pohjaisessa testiympäristössä, jonka tarkoitus on simuloida yrityksen tuotantoympäristöä.</p> <p>Opinnäytetyö jakautuu teoriaosaan ja empiiriseen osaan. Teoriaosassa käyn läpi opinnäytetyössä luotavan ratkaisuun käytettäviä tekniikoita. Empiirisessä osassa käydään läpi SQL AlwaysOn toiminnon käyttöönotto vaiheittain sekä kerrotaan, miten asennus erosi yrityksen testiverkossa Hyper-V verkosta.</p> <p>Tämän opinnäytetyön tuloksena valmistui raportti jossa on kattavat ohjeet SQL AlwaysOn toiminnon käyttöönottoon, raportissa myös kerrotaan miten tämä ratkaisu parantaa yrityksen vikasietoisuutta.</p>	
Asiasanat SQL, SQL AlwaysOn, Citrix, Replikaatio	

Author(s) Jani Kujo	
Degree programme Business Information Technology	
Report/thesis title Technical disaster recovery plan for payroll system	Number of pages and appendix pages 30
<p>The main goal of this thesis is to make functional replicating SQL AlwaysOn system. The purpose of this study is to investigate the solution for the commissioner. This thesis shows how this solution can be made and how the commissioner will be able to integrate this kind of solution into their production environment.</p> <p>The thesis project carried out the deployment of the SQL AlwaysOn feature in the Hyper-V virtual environment, after which the installation is carried out of the company's Citrix-based test environment designed to simulate the company's production environment.</p> <p>The thesis is divided into the theoretical part and the empirical part. The theoretical part reviews the techniques used to create the solution. The empirical part reviews the deployment of SQL AlwaysOn feature step by step and demonstrates how the setup shifted from the company's test environment to Hyper-V environment.</p> <p>As a result of this thesis a report was completed with comprehensive instructions for creating SQL AlwaysOn solution. The report also explains how this solution improves the fault tolerance of the company.</p>	
Keywords SQL, SQL AlwaysOn, Citrix, Replication	

Sisällys

1	Johdanto	1
1.1	Sanasto.....	2
2	Tietoperusta	3
2.1	Vikasietoisuus	3
2.1.1	Vikasietoisuuden edistäminen	3
2.1.2	Vikasietoisuus asiakkaan näkökulmasta.	3
2.2	Windows Server.....	3
2.2.1	Versiot.....	4
2.2.2	Windows Server 2008	5
2.2.3	Windows server 2012.....	5
2.2.4	Palvelimen Ylläpito.....	6
2.2.5	Windows Server vs. Linux Server.....	6
2.3	Hyper-V.....	6
2.4	Failover Cluster.....	6
2.5	SQL	7
2.5.1	SQL Versiot.....	8
2.5.2	SQL Always-on	8
2.5.3	AlwaysOn VS. Mirroring	9
2.5.4	Availability Group Listener	9
2.6	Testaaminen Personec W:llä	10
2.7	Citrix	10
2.8	Työmenetelmät	10
3	Hyper-V testiympäristön rakentaminen.....	11
3.1	Testiympäristön suunnittelu.....	11
3.2	Hyper-V ympäristön asennus	12
3.3	SQL Serverin asennus	13
3.4	Windows Failover Clusterin asennus	15
3.5	SQL-AlwaysOn toiminnon käyttöönotto vaiheittain	17
3.5.1	Kryptatun kannan liittäminen Availability Grouppiin	19
3.5.2	Availability Group Listener.....	21
3.6	Testaaminen Personec W ohjelmalla	22
4	Citrix Testiympäristö.....	23
4.1	Yleiskuva verkosta	23
4.2	Palvelin instanssit	24
4.3	SQL alwaysOn käyttöönotto.....	25
4.3.1	Ohjelmistoasennukset.....	25
4.3.2	Ongelmat ja rajoitukset	27
4.4	Työn Tavoitteet	27

5 Pohdinta.....	28
5.1 Johtopäätökset.....	28
5.2 Kehittämisen- ja jatkotutkimusehdotukset	28
5.3 Opinnäytetyöprosessi sekä oma oppiminen	28
Lähteet	30

1 Johdanto

Tämän opinnäytetyön aiheena on, rakentaa toimiva replikoiva SQL alwaysOn järjestelmä sekä kirjoittaa tekninen raportti millä tasolla vikasietoisuus on nyt ja miten tämä ratkaisu vaikuttaa vikasietoisuuden parantamiseen.

SQL AlwaysOn ratkaisu toteutetaan ensin Hyper-v virtuaaliympäristöön, jossa on oma Active Directory, kaksi SQL palvelinta sekä yksi PC testaamista varten. Kun ratkaisu on saatu toteutettua Hyper-v ympäristössä, rakennetaan samanlainen ratkaisu Citrixia hyödyntävään yrityksen omaan testiympäristöön, joka on samankaltainen kuin yrityksen tuotanto ympäristö.

Opinnäytetyössä käydään läpi vaihe vaiheelta, miten SQL AlwaysOn ratkaisu asennetaan ja miten kryptatut tietokannat saadaan replikoimaan Hyper-V ympäristössä ja miten asennus erosi, kun asennukset tehtiin yrityksen testiverkkoon.

Opinnäytetyössä on ensiksi tietoperusta, jossa käydään läpi tekniikoita ja komponentteja läpi, joita opinnäytetyössä käytetään. Toinen osuus sisältää empiirisen osuuden, jossa käydään läpi työn asennusvaiheet, käyttöönotto, ongelmat sekä lopputulos

Tämän opinnäytetyön toimeksiantaja on salainen, mainitakseni sen kyseinen yritys on kansainvälinen taloushallintoon ja it-ratkaisuihin keskittyvä yritys.

1.1 Sanasto

High availability = Tarkoittaa järjestelmien luotettavuustasoa. Koskee järjestelmiä jotka täytyvät olla tavoitettavissa kellon ympäri. High Availabilityn paremmuutta voidaan mitata "downtimella" eli ajalla jonka järjestelmä on saavuttamattomissa.

Cluster = Suom. Klusteri jolla tarkoitetaan joukkoa palvelimia jotka toimivat yhdessä.

Node = Suom. noodi on Klusterissa sijaitsevan yhden palvelimen käsite, eli yksi klusteri voi sisältää monta noodia mutta yksi noodi voi olla vain yhdessä klusterissa.

Primary = eli ensisijainen tarkoittaa SQL alwaysOn hierarkiassa palvelinta johon kaikki tietokantaa koskevat muutokset tehdään ensimäiseksi. Muutokset siirtyvät lokien perusteella toissijaisiin palvelimiin.

Secondary = eli toissijainen tarkoittaa SQL alwaysOn hierarkiassa toissijaista palvelinta. Toimii ensisijaisen palvelimen taustalla.

Replica = replikaatio eli täydellinen kopio, tässä tapauksessa toissijaiset palvelimet ovat replikoita ensisijaisesta palvelimesta.

Wizard= Ohjattu asennustoiminto, avustaa käyttäjää asentamaan komponentit/ohjelmat oikein.

Database Master Key = yksinkertaisuudessaan Master key on symmetrisesti salattu avain jonka tarkoitus on suojata yksityiset avaimet jolla sertifikaatit on salattu.

Certificate = suom. Sertifikaatti, yksinkertaisuudessaan sitä käytetään salaamaan esimerkiksi tietokantoja käyttäen epäsymmetristä salausta.

TDE = Transparent Data Encryption, on teknologia jolla salataan tiedostoja. Käytännössä salaa datan joka sijaitsee kovalevyllä sekä sen varmuuskopiot.

2 Tietoperusta

Tämän opinnäytetyön tietoperustana ovat pääosin internet-lähteet, sillä kyseisestä aiheesta ei kovin reaaliaikaista kirjallisuutta löydy. Tietoperusta sisältää keskeisiä aiheita, teknologioita ja komponentteja joita opinnäytetyössäni olen joutunut käyttämään.

2.1 Vikasietoisuus

Vikasietoisuus on hyvin tärkeä aihe mikä tuntuu usein jäävän unholaan, vikasietoisuudella tarkoitetaan sitä, että miten eri järjestelmät, alustat yms. käyttäytyy ja suojaa itsensä vika-tilanteissa. Esimerkiksi jos konesalissa jossa on kymmenen tietokantapalvelinta, tapahtuu sähkökatko, niin miten näiden tietokantapalvelinten data suojataan ja saadaan mahdollisimman nopeasti uudelleen käyttöön.

2.1.1 Vikasietoisuuden edistäminen

Yleisesti vikasietoisuutta voidaan parantaa miettimällä eri skenaarioita mitä esimerkiksi palvelimille voi sattua konesaleissa. Konesalien palvelinten dataa voidaan turvata useilla eri keinoilla, tärkeintä kuitenkin on, että data sijaitsee kahdessa eri fyysisessä maantieteellisessä sijainnissa, koska on epätodennäköistä, että kaksi kone salia eri maantieteellisessä sijainneissa kaatuu samanaikaisesti.

2.1.2 Vikasietoisuus asiakkaan näkökulmasta.

Palvelutaso sopimus eli Service level agreement (SLA) on asiakkaan ja palveluntarjoajan välinen sopimus, jossa määritellään palvelun taso, tarkoittaen että asiakkaalle pitää määrittää tietyille palveluille tietty vaatimustaso. IT:n näkökulmasta Palvelutaso sopimuksessa määritellään esimerkiksi, miten asiakkaan datat varastoidaan, kuinka kauan säilytetään varmuuskopioita, miten vikatilanteissa toimitaan ja palveluaika eli minä työtunteina asiakasta palvellaan.

2.2 Windows Server

Windows Server on Microsoftin brändäämä palvelinkäyttöjärjestelmä tuoteperhe. Ensimmäinen Windows Server tuote oli Windows Server 2003, kuitenkin tämä ei ole ensimmäinen Windows-pohjainen palvelinkäyttöjärjestelmä, vaan virallinen ensimmäinen versio

oli Microsoft Windows NT 3.1, joihin kaikki nykyiset Windowsin palvelinkäyttöjärjestelmät perustuu.

Yleisimmät toiminnot joita tarvitaan infrastruktuurin ylläpitoon ovat Active Directory, DNS Server, DHCP Server, Group Policy. Näitä teknologioita kyseisessäkin opinnäytetyössä joudun käyttämään.

Active Directory = on käytännössä tietokanta, joka sijaitsee domain controllerilla eli toimialueen pääpalvelimella. Active Directory sisältää yrityksen tiedot konetileistä, palvelintileistä sekä käyttäjätileistä. Active Directoryssä hallitaan käyttäjien sekä koneiden oikeuksia. Active Directoryn rakenne on hierarkkinen, joten jokaiselle konetilille ja käyttäjätileille voidaan asettaa erilaisia rajoituksia ja oikeuksia.

DNS Server = (Domain Name Services) on nimipalvelu joka välittää käyttäjien PC:iden sekä palvelinten välillä liikenteen oikeisiin osoitteisiin.

DHCP Server = (Dynamic host Configuration Protocol) on verkko osoitteiden jakoon käytettävä teknologia, DHCP palvelu huolehtii, että jokainen verkkoon liitettävä laite saa IP-osoitteen, oletusyhdykskäytävän sekä DNS-palvelimen osoitteen, ellei laitteelle oli staattisesti määritetty kyseisiä osoitteita.

Group Policy = on Ryhmäkäytäntöjen hallinnointi palvelu, ryhmäkäytännöillä voi määritellä esimerkiksi kieltoja, rajoitteita, käyttöoikeuksia sekä ohjelmisto asennuksia suurelle joukolle koneita tai käyttäjiä samanaikaisesti.

Windows Server ohjelmistot sisältävät myös liudan muitakin käytännöllisiä ominaisuuksia ja on täten kaiken kattava palvelinkäyttöjärjestelmä.

2.2.1 Versiot

Windows Server ohjelmistoista julkaistaan uusi versio noin joka kolmas vuosi, ja jokainen versio sisältää muutaman erilaisen "Editionin", jotka on räätälöity eri käyttötarkoituksiin. Viimeisin julkaistu version on Windows Server 2012 R2, joka julkaistiin lokakuussa 2013. Loppuvuodesta 2016 Microsoft on julkaisemassa uuden Server ohjelmiston joka kantaa nimeä Windows Server 2016.

Windows Server 2008 ohjelmistosta julkaistiin seitsemän eri editiota, jotka ovat Web, Standard, Enterprise, Datacenter, Itanium, Foundation ja HPC. Jokainen versio on eri käyttötarkoitukseen luotu.

Windows server 2012 ohjelmistosta julkaistiin neljä eri editiota, jotka ovat Standard, Datacenter, Essentials ja Foundation. Kyseiset editiot ovat hieman eri käyttötarkoituksiin luotu, esimerkiksi datacenter on täysin virtualisointiin tarkoitettu suurelle käyttäjämäärälle, kun taas Foundation on ekonomisen vaihtoehto pienelle käyttäjämäärälle.

Editionit erottuvat toisistaan kapasiteetilla, kapasiteetilla tarkoittaen esimerkiksi sitä, että kuinka paljon prosessoreita, fyysistä muistia, sekä virtuaaliyhteyksiä, palvelin osaa käyttää.

Windows Server 2008 ja 2012 versioista on alkuperäisestä julkaisusta noin vuotta myöhemmin julkaistu R2 versiot jotka ovat tuoneet palvelinkäyttöjärjestelmiin merkittäviä uudistuksia ja päivityksiä, kuten 2008 R2 toi tuen 64 bittisille prosessoreille.

Jokaisesta edellä mainitusta Editionista on myös olemassa Core edition joka sisältää melkein samat toiminnot, mutta ilman graafista käyttöliittymää eli kaikki asiat konfiguroidaan komentotulkin kautta. Kyseinen Edition on hyödyllinen silloin kun laiteresurssit ovat kehnot, eli tämä versio ei vaadi läheskään yhtä tehokasta rautaa kuin käyttöliittymän sisältävät versiot.

2.2.2 Windows Server 2008

Windows server 2008 on Windows Vista käyttöjärjestelmän koodiin perustuva palvelinkäyttöjärjestelmä. Windows server 2008 toi lukuisan määrän uusia ominaisuuksia, sillä edellisestä käyttöjärjestelmän julkaisusta oli ehtinyt kulua 5 vuotta. Merkittävimmät uudistukset olivat Bitlocker, .NET Framework ominaisuudet Hyper-V sekä Server Core.

Opinnäytetyössä käytettävä Hyper-V ympäristön Domain Controller on luotu käyttäen käyttöjärjestelmää Windows Server 2008 Standard.

2.2.3 Windows server 2012

Windows Server 2012 on kuudes julkaisu Windows Server tuoteperheestä. Tuote pohjautuu Microsoftin Windows 8 käyttöjärjestelmään. Server 2012 julkaistiin taas paljon uusia ominaisuuksia ja paranneltiin vanhoja, näkyvin ominaisuus oli kuitenkin Windows 8 käyttöjärjestelmästä tuttu "metro" käyttöliittymä.

Opinnäytetyössä käytettävät SQL1- ja SQL2 palvelimet on luotu käyttäen käyttöjärjestelmää Windows Server 2012 R2 Hyper-V ympäristössä sekä testiympäristössä.

2.2.4 Palvelimen Ylläpito

Palvelinten ylläpito on helppoa, sillä Windows Server ohjelmisto on hyvin vakaa, ja vikatilanteiden sattuessa Server Manager käyttöliittymä ilmoittaa virheistä. Server ohjelmistoissa on myös todella hyvät automaattiset korjausehdotukset sekä Wizardit.

Microsoft tarjoaa palvelimille ajantasaisia päivityksiä ohjelmistoihin ja tietoturvaan luotettavasti. Microsoftilla on myös kattava tuki kaikkiin palvelinten ominaisuuksiin, yrityksen kotisivuilta löytyy laajat ohjeet jokaiseen ominaisuuteen ja niiden konfigurointiin eri tilanteissa.

2.2.5 Windows Server vs. Linux Server

Avoimeen lähdekoodiin pohjautuvalle Linux käyttöjärjestelmälle on myös luotu Windowsin kaltaisia Server ohjelmistoja. Suurimmat erot näissä ovat hinta ja käytettävyys. Windows palvelin käyttöjärjestelmällä on suuret lisenssimaksut, kun taas Linux on ilmainen, asiantuntijoiden mukaan myös Linux on hieman nopeampi toiminnoissaan. Windows Server on taas huomattavasti helppokäyttöisempi, sillä Linux pohjaiset järjestelmät vaativat paljon enemmän asiantuntemusta ja komentotulkin käyttöä. Windows järjestelmien ylläpito on myös luotettavampaa koska sen takaa Microsoft, avoimelle Linuxille taas ei ole ketään virallista ylläpitäjää, joten siitä joutuu huolehtimaan aina itse.

2.3 Hyper-V

Hyper-V on Microsoftin kehittämä virtuaalialusta, joka on julkaistu Windows Server 2008 palvelinkäyttöjärjestelmän yhteydessä. Oletuksena nykyiset Windowsin käyttöjärjestelmät sisältävät Hyper-V:n piilotettuna ominaisuutena jonka voi ottaa halutessaan käyttöön. Hyper-v:n avulla voi käyttäjä luoda oman virtuaaliympäristön joka voi sisältää useampia verkkoja sekä 32- sekä 64-bittisiä pääasiassa Windowsin käyttöjärjestelmiä. Microsoft on myös julkaissut tuen Linuxin kernelille, mutta sen toimivuudesta en osaa sanoa mitään.

2.4 Failover Cluster

Microsoft Cluster Server on ohjelma joka mahdollistaa usean palvelimen työskentelemään yhdessä parantaen vikasietokykyä ja valmiutta vikatilanteisiin. Kyseinen ohjelma on ollut

mukana Server ohjelmistossa jo vuodesta 1996, versiossa Windows NT Server 4.0 Enterprise. Kyseistä ohjelmaa on joka vuosi paranneltu ja kehitetty, ohjelma kantaa nykyään nimeä Windows Server Failover Clustering

Failover Cluster eli vikasieto klusteri on käytännössä ryhmä palvelimia, jotka toimivat samassa verkossa vikasietoisuuden edistämiseksi. Klusterin tarkoitus on lisätä valmiustasoa ”high availability” jolloin kaikki tarvittavat palvelut ja resurssit ovat käytössä koko ajan. Jokaista klusteriin kuuluvaa palvelinta kutsutaan nodeksi eli noodiksi, jokaisessa noodissa on käynnissä haluttu palvelu tai sovellus. Klusteri valvoo jokaisen noodin elintoimintoja, joten jos yksi noodin kaatuu niin klusteri huomaa sen ja keskeyttää palvelun käytön kyseiseltä noodilta. Klustereilla voi myös tehdä kuormantasausta jolloin klusteri jakaa suoritettavat tehtävät tasaisesti keskenään jolloin yksi palvelin ei kuormitu vaan koko jokainen saa osansa.

Vikasieto klusterin hallinnointityökalu asennetaan jokaiseen noodiin, jotka halutaan tämän vikasieto klusterin vaikutuksen alle. Jokainen noodin sisältää Failover Cluster Managerin eli klusterin hallinnointityökalun, kyseisellä työkalulla voidaan tarkastella ja hallita noodeja ja palveluita jotka ovat tämän klusterin vaikutuksen alla.

Käyttäjät eivät näitä noodeja näe, käyttäjät yhdistävät vain palveluun tai ohjelmaan, jos käyttäjä on jossain tietyssä palvelussa kiinni ja noodin vioittuu, niin klusteri pyrkii siirtämään käyttäjän toisen noodin vaikutuksen alle.

2.5 SQL

SQL on 1970 luvulla kehitetty standardoitu koodikieli joka on suunniteltu käytettäväksi relaatiotietokannoissa, SQL-lauseilla voidaan hallita relaatiotietokantoja sekä tehdä kaiken kattavia kyselyitä tietokannoista.

Yksinkertainen SQL lause jolla tietokantaan lisätään tauluja sisältää Create table lauseen.

<pre>CREATE TABLE MAKSUNSAAJA (ETUNIMET varchar(255), POSTINUMERO int, POSTITOIMIPAIKKA varchar(255))</pre>	<p>Luotavan taulun nimi</p> <p>Arvoja jotka tauluun lisätään.</p> <p>Int = numeroita</p> <p>Varchar = Tekstiä 255 merkkiä</p>
---	---

Kuva 1, Yksinkertainen Create Table lause.

Yksinkertainen SQL lause jolla tauluihin lisätään dataa sisältää Insert into lauseeseen.

```
INSERT INTO MAKSUNSAAJA(ETUNIMET, POSTINUMERO, POSTITOIMIPAIKKA)
VALUES ('Elina', '00390', 'Helsinki')
```

Arvot jotka tauluun lisätään
Määritellään arvot

Kuva 2, Yksinkertainen Insert Into lause.

Yksinkertainen SQL lause jolla tietokannasta haetaan dataa sisältää SELECT ja FROM parametrit, joka indikoi sitä eli jotain valitaan jostakin.

```
use TESTipalkkakantaPW640
SELECT ETUNIMET, POSTINUMERO, POSTITOIMIPAIKKA
FROM MAKSUNSAAJA
WHERE POSTITOIMIPAIKKA='Helsinki'
ORDER BY POSTINUMERO
```

Kertoo mitä tietokantaa käytetään
Kertoo mitä arvoja tietokannasta haetaan
Kertoo mistä tietokannan taulusta kyseiset arvot haetaan
Rajaa haun koskemaan henkilöitä jonka postitoimipaikka on helsinki
Kertoo millaisenn järjestykseen lajitellaan, postinumero pienimmästä -> isompaan

Kuva 3 yksinkertainen Select lause

Maria	00350	HELSINKI
Kati	00350	Helsinki
Aino	00370	HELSINKI
Elina	00390	Helsinki
Paula Anneli	00400	HELSINKI

Kuva 4, Edellisen Select SQL-lauseen tuloste.

2.5.1 SQL Versiot

Microsoft julkaisee SQL Server ohjelmistoista noin joka toinen vuosi uuden version ja jokaiseen versioon julkaistaan muutama laaja päivitys jotka kantavat nimeä Service pack. Microsoft palveluihin kuuluu vanhojen versioiden perustuki, joka kattaa pieniä ohjelmistopäivityksiä, bugikorjauksia sekä tietoturvapäivityksiä ja tuki kestää noin 5 vuotta riippuen ohjelmistosta. Ohjelmistoihin on myös mahdollista saada extended tuki jonka avulla nämä päivitykset jatkuvat vielä seuraavat 5 vuotta perustuen jälkeen.

2.5.2 SQL Always-on

SQL AlwaysOn on SQL-server 2012 versiossa julkaistu uusi ominaisuus jolla voidaan parantaa tietokantojen "High availability" valmiutta, tarkoittaen sitä, että pyritään siihen, että tietokanta olisi aina käyttäjän käytettävissä. SQL-always on ei tuo mitään uutta teknologiaa vaan parantaa nykyisiä olemassaolevia teknologioita, sillä pyritään käytännössä luomaan tietokanta, joka on aina käytettävissä, vaikka jokin palvelin sattuisikin vioittumaan.

Perusperiaatteena on se, että sama tietokanta sijaitsee useammalla palvelimella, ns. replikoina. Kun ensisijainen palvelin vioittuu tai sammuu, niin SQL AlwaysOn uudelleenohjaa liikenteen jollekin replikoista, eli toissijaiselle palvelimelle. Jotta tämä kyseinen uudelleenohjaus toimii, täytyy jokaisen palvelimen kuulua Failover Clusterin vaikutuksen alle sekä jokaisen palvelimen pitää kuulua samaan Availability Groupiin.

2.5.3 AlwaysOn VS. Mirroring

SQL AlwaysOn perusperiaate on sama kuin Mirroring ominaisuudessa mutta tarjoaa paljon paremman ja luotettavamman vikasieto valmiuden. Ensimmäinen suuri etu AlwaysOn ominaisuudella on se, että data tietokannoissa liikkuu reaaliajassa, eli data synkronoidaan ensisijaiselta palvelimelta toissijaisille palvelimille saman tien, kun se on kirjoitettu ensisijaiselle palvelimelle ja data on luettavissa toissijaisilta palvelimilta, kun taas mirroring ominaisuudessa data kirjoitetaan ensisijaiselta palvelimelta toissijaiselle palvelimelle mutta ei ole sieltä luettavissa. Molemmat ominaisuudet kirjoittavat datansa transaktiolokien perusteella jotka ensisijainen palvelin lähettää toissijaisille palvelimille jatkuvasti.

Erona on myös se, että SQL AlwaysOn ominaisuuteen voidaan liittää jopa neljä toissijaista palvelinta joka takaa sen, että data on aina luettavissa ja on hyvin epätodennäköistä että kaikki palvelimet olisivat samaan aikaan epäkunnossa. Mirroring ominaisuudessa on mahdollista luoda vain yksi toissijainen palvelin joka ei takaa samankaltaista luotettavuutta.

Jos ensisijainen palvelin sattuu vioittumaan AlwaysOn ominaisuus vaihtaa käyttäjän huomaamatta jonkin toissijaisen palvelimen ensisijaiseksi palvelimeksi ja käyttäjä voi jatkaa toimiaan ilman mitään käyttökatkoksia, Mirroring ominaisuudessa saattaa käyttäjä huomata pienen viiveen kun ensisijainen palvelin vaihtuu toissijaiseksi sillä toissijainen palvelin on standby tilassa jolloin se vain vastaanottaa dataa ensisijaiselta palvelimelta.

2.5.4 Availability Group Listener

Availability group listener on työkalu, jonka avulla noodien liikennettä hallinnoidaan, tämä listener ohjaa liikenteen ensisijaisesta tietokannasta toissijaiseen tietokantaan, kun ensisijainen tietokanta vioittuu. Esimerkkinä kun personec W:llä ottaa yhteyden listenerin kautta ensisijaiseen tietokantaan ja se kaatuu, niin listener ohjaa personec W:n liikenteen käyttämään toissijaista tietokantaa.

Listener on tarkoitettu käytettäväksi silloin kun kolmannen osapuolen ohjelmalla yhdistetään tietokantoihin. Yhteys tietokantaan toimii Listenerin kautta. Listener keskustelee tietokantojen kanssa ja määrittää kumpaan kantaan kirjoitetaan.

2.6 Testaaminen Personec W:llä

Tietotekniikassa testaus on yksi ohjelmisto- sekä infrastruktuurikehityksen pääelementteistä, testauksessa pyritään tekemään tilanteita mahdollisimman realistisissa tilanteissa sekä ympäristöissä. Testaamisen tarkoitus on kitkeä pois kaikki virhetilanteet, jotta sovelluksesta tulisi mahdollisimman toimiva ja luotettava. Isommissa ohjelmistotuotanto ympäristöissä voi testaaajia olla kymmeniä tai jopa satoja.

Tässä opinnäytetyössä testaamiseen käytetään Personec W nimistä palkanlaskenta ohjelmaa, jolla pyritään simuloimaan palkanlaskijoiden päivittäisiä toimia. Personec W on aditron kehittämä palkanlaskentaohjelma.

2.7 Citrix

Citrix Systems Oy on vuonna 1989 vuonna perustettu yhtiö joka on iso vaikuttaja tietotekniikan saralla. Citrixin pääpaino nykypäivänä on Virtualisoinnissa sekä pilvipalveluissa.

Tässä opinnäytetyössä käytetään Personec W ohjelmistoa, joka toimii Citrixin ”yli” tarkoittaen että Citrix Receiver ohjelmalla luodaan yhteys käyttäjän ja palvelimen välillä.

Opinnäytetyössä käytettävä testiverkko on Citrix pohjainen virtuaaliverkko.

2.8 Työmenetelmät

Mallinnetaan yrityksen tuotantoympäristö yrityksen virtuaalisessa testiympäristössä.

Testiympäristöön suunniteltiin ja rakennettiin toimiva SQL AlwaysOn järjestelmä.

Työn suunnittelun ja kirjanpidon apuna on käytetty Trello nimistä selainpohjaista ohjelmaa, jonka avulla työvaiheita voidaan kuvailla ja hallinnoida.

3 Hyper-V testiympäristön rakentaminen

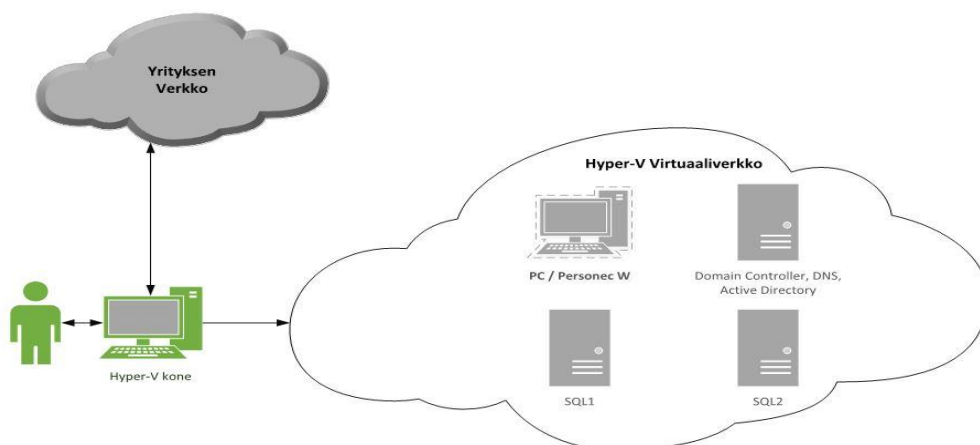
Tässä luvussa käydään läpi, miten Hyper-V Ympäristöön rakennetaan replikoiva SQL AlwaysOn järjestelmä.

3.1 Testiympäristön suunnittelu

Tässä opinnäytetyössä oli tarkoitus simuloida mahdollisimman realistinen tilanne, joten testiympäristö piti olla samankaltainen kuin yrityksen sisäinen verkko. Hyper-V:n kanssa realistista verkkoa emme voineet rakentaa ajan puutteen takia, joten päädyimme Hyper-V:n avulla testaamaan SQL Serverin asennuksen, SQL-always On asennuksen, SQL-always On toiminnon toimintaperiaatteen ja sen miten kryptatut tietokannat replikoidaan. Tietokantojen toimivuuden testaaminen suunniteltiin suoritettavaksi Personec W palkanlaskentaohjelmalla, eli käytännössä Personec W:llä otetaan yhteys tietokanta palvelimeen, sammutetaan palvelin ja katsotaan katkeeko yhteys vai siirtyykö yhteys replikoidalle palvelimelle.

Hyper-V ympäristö rakennettiin yhdelle Yrityksen tietokoneista, joka oli käytännössä minun käytössäni. Tietokone sisältää 20GB RAM-muistia sekä 120GB ja 240GB SSD-kovalevyt, jotta virtuaaliympäristö toimisi mahdollisimman tehokkaasti.

Hyper-V virtuaalialustaan tarvittiin kolme palvelinta sekä yksi PC. Ensimmäinen palvelin suunniteltiin käyttämään Windows Server 2008 palvelinkäyttöjärjestelmää. Kyseinen palvelin toimii Domain controllerina, jonka avulla luodaan toimialue, kyseiseen palvelimeen asennettiin myös Active Directory sekä DNS palvelut. Toinen ja kolmas palvelin suunniteltiin käyttämään Windows Server 2012 palvelinkäyttöjärjestelmää, kyseisiin palvelimiin asennettiin Microsoft SQL Server 2014 ohjelmistot.

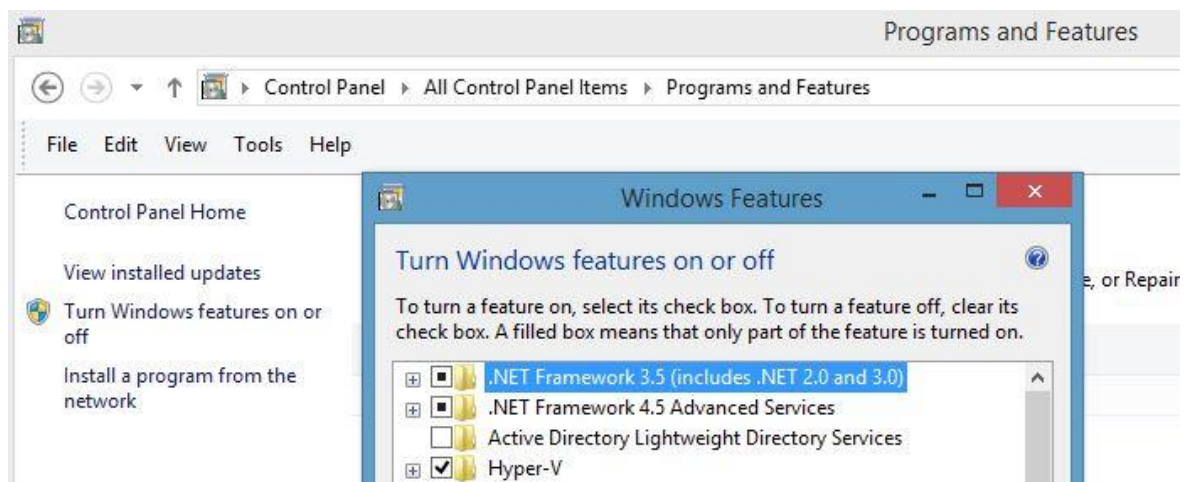


Kuva 5. Hyper-V virtuaaliympäristön verkkokuva

Kun testit oli saatu toimimaan Hyper-V virtuaaliympäristössä, niin seuraava vaihe oli tehdä samankaltainen SQL ratkaisu yrityksen testiverkkoon. Testiverkko on hyvin samankaltainen kuin yrityksen käyttämä sisäinen verkko, joten sinne ei tarvinnut luoda Domain controlleria DNS palveluita yms. toimialueen ylläpitoon vaadittavia komponentteja. Testiverkkoon luotiin vain 2 virtuaalista palvelinta, joihin asennetaan SQL Server 2014.

3.2 Hyper-V ympäristön asennus

Hyper-V:n käyttöönotto on windows 8.1: llä hyvin helppo toimenpide, sillä ohjelma on esi-asennettu käyttöjärjestelmään. Hyper-V:n saa aktivoitua käyttöön ohjauspaneelin kautta.



Kuva 6. Hyper-V:n käyttöönotto

Kun Hyper-V on asennettu, luodaan Hyper-V käyttöliittymässä uusia virtuaalikoneita. Virtuaalikoneisiin asennetaan käyttöjärjestelmät ISO-tiedostojen perusteella, jotka ovat ladattavissa Microsoftin sivuilta. Käyttöjärjestelmiä ei tässä tapauksessa pysty aktivoimaan, sillä Hyper-V virtuaaliverkosta on estetty pääsy ulkoverkkoon.

Virtuaaliympäristön Domain Controller asennetaan Windows Server 2008 ISO-tiedostolla ja domain controlleriin asennetaan featureina Active Directory Domain Services sekä DNS Server. Luodaan toimialue oppari.com. käyttäen ohjattua asennusohjelmaa. Active directory:stä luodaan käyttäjä, SQLAdmin joka on domain tason admin.

Virtuaaliympäristön SQL palvelimet asennetaan Windows Server 2012 R2 ISO-tiedostolla. Palvelimille tehdään seuraavat toimenpiteet, nimetään palvelimet SQL1 ja SQL2, liitetään toimialueelle Oppari.com.

PC:hen on asennettu Windows 8.1 Enterprise ISO-tiedostosta joka on käytössä myös Yrityksen muissa PC:issä jolloin image on mahdollisimman samankaltainen kuin muutkin tuotantoympäristön koneet. PC liitetään myös Oppari.com toimialueelle

Kaikkille ympäristön koneille on asetettu Staattiset IP-osoitteen kahdesta syystä, ensimmäinen syy on se, että tämänkaltainen testiverkko toimii luotettavammin ja toinen syy on se, että verkossa ei ole DHCP palvelua. IPV6-osoitteet on jokaisesta koneesta kytketty pois sillä huomasi, että SQL-Server saattaa yrittää käyttää näitä osoitteita, jolloin liikenne ei välttämättä kulkisi oikein.

dc	Host (A)	10.10.10.1
Listener	Host (A)	10.10.10.54
PC1	Host (A)	10.10.10.5
SQL1	Host (A)	10.10.10.2
SQL2	Host (A)	10.10.10.3
Testi	Host (A)	10.10.10.4

Kuva 7, Hyper-V verkon Staattiset osoitteet.

3.3 SQL Serverin asennus

Ennen SQL-server ohjelmiston asennusta esiasennusvaatimuksena NET.Framwork 3.5 ohjelmistokomponenttikirjasto. Kuvassa 8 SQL:n luoma virheilmoitus jos kyseistä ohjelmistokomponenttikirjasto ei ole asennettu. Asennus onnistuu Windows Serverin Add Roles and Features kohdasta. Asennettaessa on vaihtoehtoinen polku asennustiedostoille määriteltävä, jotta asennus suorituu, sillä Hyper-V virtuaaliverkosta ei ole pääsyä Microsoftin palvelimille. Jotta asennus onnistuu pitää Hyper-v koneeseen pudottaa Windows server 2012 R2 ISO-tiedosto josta asennusohjelma osaa hakea tarvittavat tiedostot. Polku kyseisiin tiedostoihin on D:\Sources\SXS

	Rule	Status
✓	Prior Visual Studio 2010 instances requiring update.	Passed
✗	Microsoft .NET Framework 3.5 Service Pack 1 is required	Failed

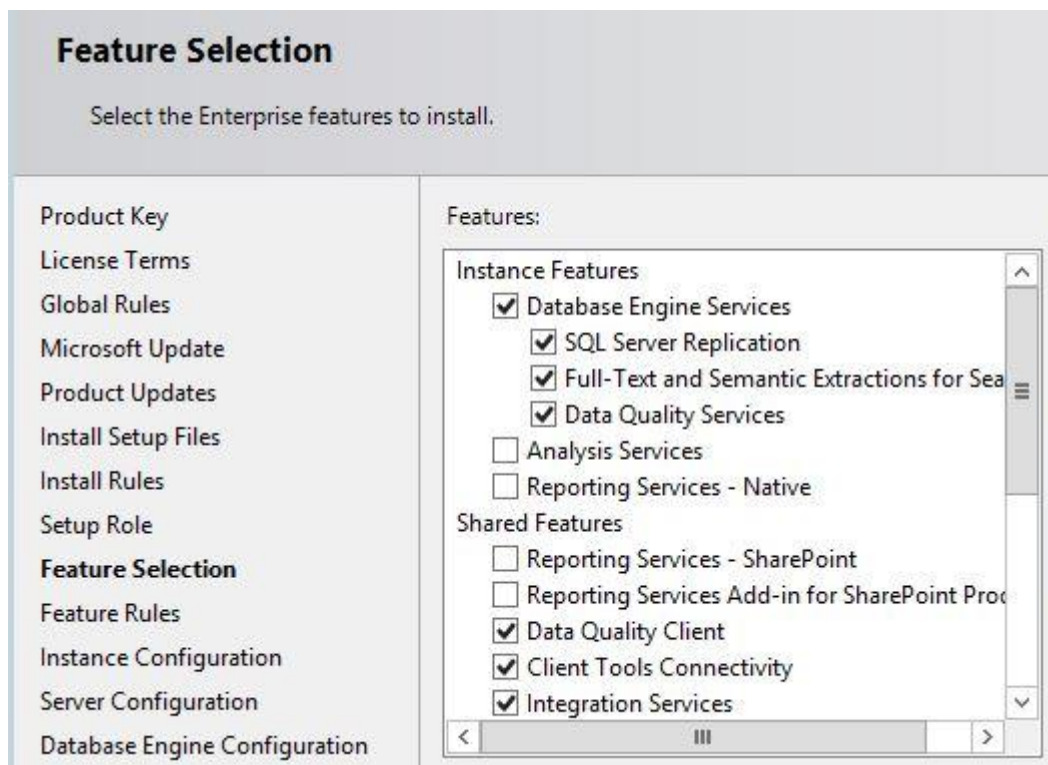
Kuva 8, SQL virheilmoitus kun .NET Frameworkia ei ole asennettu

Koneeseen asetetaan SQL2014 ISO-tiedosto ja ajetaan ISO-tiedostosta Setup.exe. SQL:n asennusikkunasta valitaan new stand-alone installation or add features to an existing installation kuten kuvassa 9 näkyy.



Kuva 9, Uuden SQL serverin asennuksen valinta.

Asetukset voidaan pitää oletusasetuksin, kunnes ollaan kohdassa kohdassa Setup role jossa, valitaan SQL server feature installation, Feature Selection kohdassa karsitaan pois Reporting services, Analysis Services sekä SharePoint liitännäiset.



Kuva 10, SQL:n lisäosien valinta.

Instance Configuration kohdasta valitaan Named Instance, tässä tapauksessa SQL1 palvelin on nimellä MASTER ja SQL 2 on nimellä SLAVE, jolloin palvelinten koko nimet ovat SQL1\MASTER ja SQL2\SLAVE. Server Configuration kohdasta Server Configuraatiot jätetään oletuksin eikä toistaiseksi muuteta Service Accountteja. Database Engine Configuration kohdasta valitaan Authentication modeksi Mixed mode, jolloin tietokanta hyväksyy sisäänkirjautumiset Windows sekä tietokannan käyttäjätileiltä. Asetetaan SQL server Sys-

tem Administratorille (sa) salasana. Datadirectories välilehdestä muutetaan kaikki hakemistot C:\SQL kansion alle, jolloin tietokannat ja backupit ovat helposti löydettävissä ja käsiteltävissä. Loput asetuksista ja tarkistuksista voidaan suorittaa oletusasetuksin, kunnes asennusohjelma käynnistyy. Asennusohjelma ilmoittaa, kun asennus on valmistunut tai jos jonkun komponentin asennuksessa oli ongelma

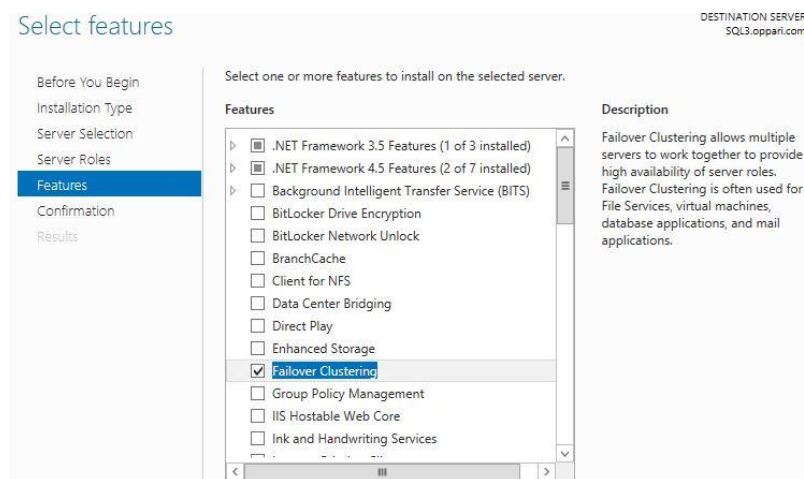
Kun asennus on suoritettu, voidaan SQL Server Configuration Managerista asettaa toimialueen järjestelmänvalvoja tunnuksella SQL Server (Serverin nimi) niminen palvelu käynnistettäväksi. Tämän lisäksi SQL Server Network Configuration kohdasta Protocols for (Palvelimen nimi) ja kohtaan TCP/IP, otetaan kaikki Dynaamiset portit poiskäytöstä ja lisätään kohtiin TCP port arvo 1433.

3.4 Windows Failover Clusterin asennus

Kun SQL Server ohjelmisto on asennettu, vaatii SQL AlwaysOn toiminto vielä toimiakseen Failover clusterin, jonka avulla noodit keskustelevat keskenään ja minkä avulla määritetään, mikä tietokanta toimii primary kantana.

Klusteria luodessa kannattaa huomioida seuraavat seikat, Windows Failover Cluster vaatii oman IP-osoitteen, Active directoryyn virtuaalisen konetilin sekä DNS recordin jotka luodaan klusterin luonnin yhteydessä. Tässäkin opinnäytetyössä huomasin että, joissakin verkoissa oikeudet eivät riitä virtuaalisen konetilin sekä DNS nimen luontiin vaan jouduimme korkeamman tason järjestelmänvalvojaa luomaan ne. Hyper-V verkossa tätä ongelmaa tietenkään ei ollut.

Windows failover Clusterin asennus alkaa Server Managerin Add roles and Features painikkeesta. Asennus ohjelma suorittaa Windows Server Failover cluster featuren asennuksen automaattisesti, kun se on valmis, voidaan käynnistää Failover Cluster Manager.



Kuva 11, Failover Clustering Featuren lisäys.

Failover Cluster Managerissa luodaan uusi klusteri, johon lisätään SQL1 sekä SQL2 palvelimet. Ohjattu klusterin luonti tarkistaa ovatko palvelimet kelvollisia klusterin käyttöön. Läpäistyään kelvollisuustestit asennusohjelmaa voidaan jatkaa. Seuraavassa vaiheessa klusteri pitää nimetä sekä antaa klusterille IP-osoite, oletuksena klusteri luodaan Active directory:ssä organisaatio yksikköön Domain -> Computers, hyper-V ympäristössä se onnistuu ongelmitta. Asennusohjelma luo myös Active Directoryyn Virtuaalisen konetilin sekä DNS nimen kyseiselle klusterille.



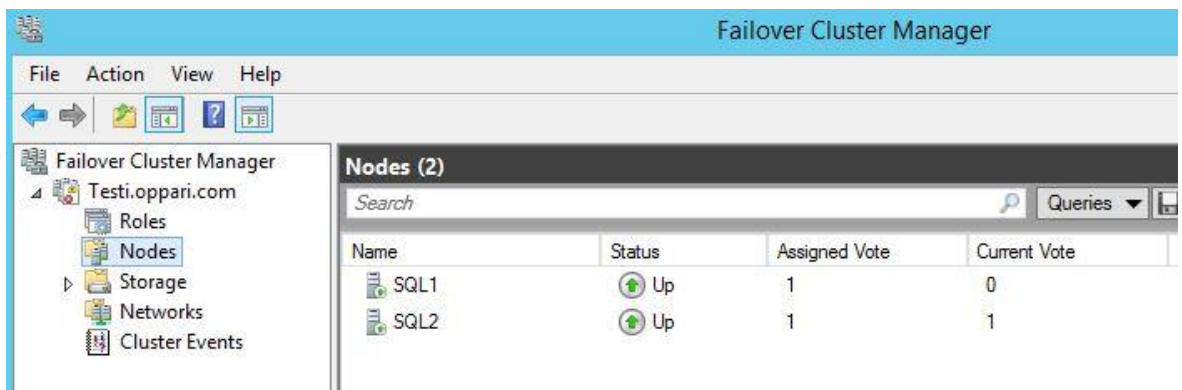
Select the servers to add to the cluster:

Enter server name:

Selected servers:

Kuva 12, Noodien valitseminen klusterille

Jos klusterin asennusohjelma on oikein suorittunut niin Failover Cluster Managerissa pitäisi päästä hallitsemaan noodeja. Toistaiseksi Roles näkymä on vielä tyhjä, koska SQL-Always On toimintoa ei ole määritetty.

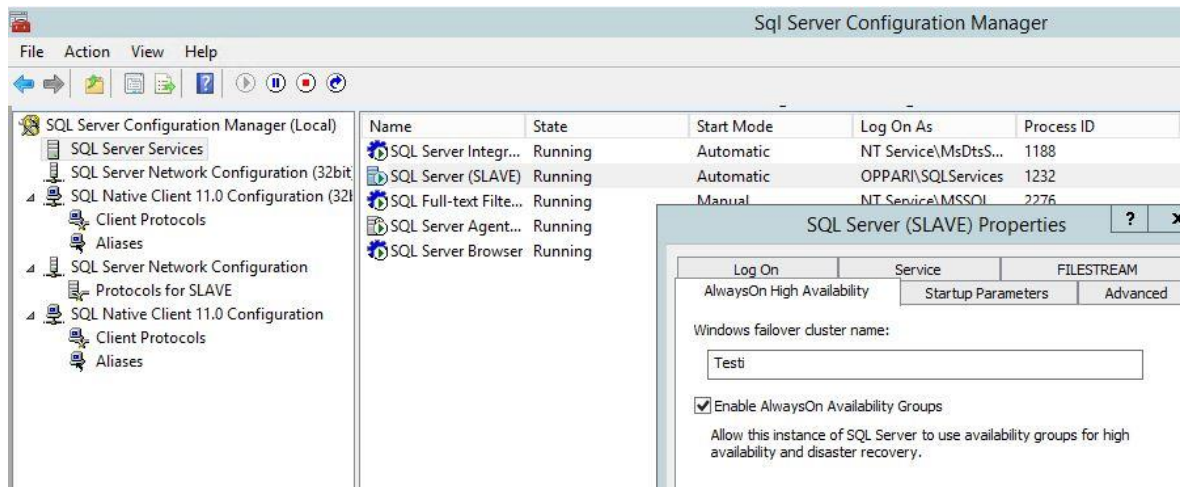


Kuva 13. Failover Clusterin oletusnäkymä.

Testaamisen aikana huomasin, että Roles kohdassa oleva AlwaysOnTesti availability group ei ole toiminnassa. Tämä johtuu siitä, että oletuksena Failover Cluster sallii vain yhden Failoverin eli palvelimen alasajon kuuden tunnin aikana. Tämän ongelman sai korjattua siten, että Roles kohdan AlwaysOnTesti asetuksista muuttaa maksimi määrän sallittuja alasajoja tämän kyseisen kuuden tunnin aikana.

3.5 SQL-AlwaysOn toiminnon käyttöönotto vaiheittain

Ensimmäinen Vaihe SQL-AlwaysOn toiminnon käyttöönottoon on sallia se SQL Console Managerista. Vaatimuksena on että, palvelimen täytyy olla liitettyä Windows Failover klusteriin ennen kuin asetuksen voi laittaa päälle.



Kuva 14, AlwaysOn asetuksen käyttöönotto.

Toinen vaihe on luoda Database Master Key, joka luodaan yksinkertaisesti Transact-SQL komennolla CREATE MASTER KEY. Database Master Key luodaan jokaiselle palvelimelle erikseen.

Database Master Key luontilause:

```
CREATE MASTER KEY  
ENCRYPTION BY PASSWORD = 'MGw66kdkdkddak'
```

Kolmas vaihe on luoda Sertifikaatti, luominen tapahtuu Transact-SQL lauseella CREATE CERTIFICATE. Kun sertifikaatti on luotu, suositeltavaa on, että se varmuuskopioidaan ja kryptataan yksityisellä avaimella.

Sertifikaatin luontilause ensisijaisella palvelimella:

```
CREATE CERTIFICATE TDECert  
WITH SUBJECT = 'TDE Certificate'
```

Sertifikaatin varmuuskopiointi lause:

```
BACKUP CERTIFICATE TDECert  
TO FILE='C:\SQL\Backup\TDECert.certbak'  
WITH PRIVATE KEY (  
FILE='C:\SQL\Backup\TDECert.pkbak',  
ENCRYPTION BY PASSWORD='salasana')
```

Sertifikaatin luontilause toissijaisella palvelimella:

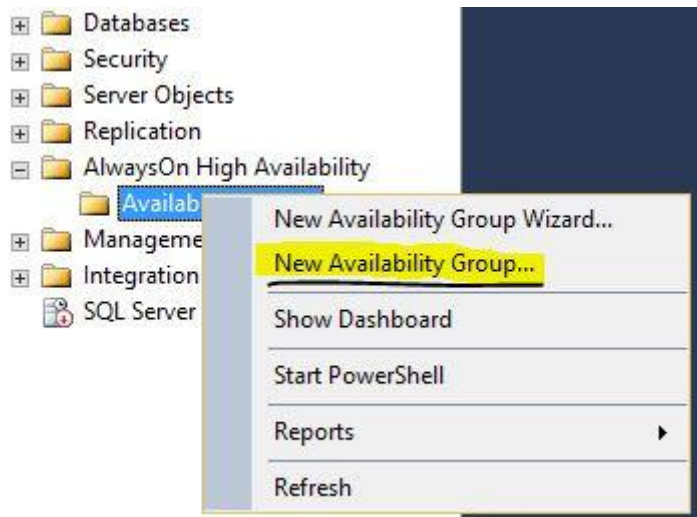
```
CREATE CERTIFICATE [TDECert]  
FROM FILE = '\\SQL1\c$\SQL\Backup\TDECert.certbak'
```



```
WITH PRIVATE KEY ( FILE = '\\SQL1\c$\SQL\Backup\TDECert.pkbak',  
DECRYPTION BY PASSWORD = 'salasana')
```

Sertifikaatti luodaan vain ensisijaiselle palvelimelle, josta se kopioidaan toissijaisille palvelimille. Tässä opinnäytetyössä sertifikaatti on luotu SQL1 palvelimelle, josta se kopioidaan SQL2 palvelimelle. Opinnäytetyön teossa huomasin että, jos sertifikaatti ei ole kopioitu oikein, tietokantojen liittäminen Availability Grouppiin epäonnistuu.

Neljäs vaihe on luoda Availability Group, luonti onnistuu SQL Management studiossa, AlwaysOn High Availability kohdasta, valitsemalla New Availability Group.

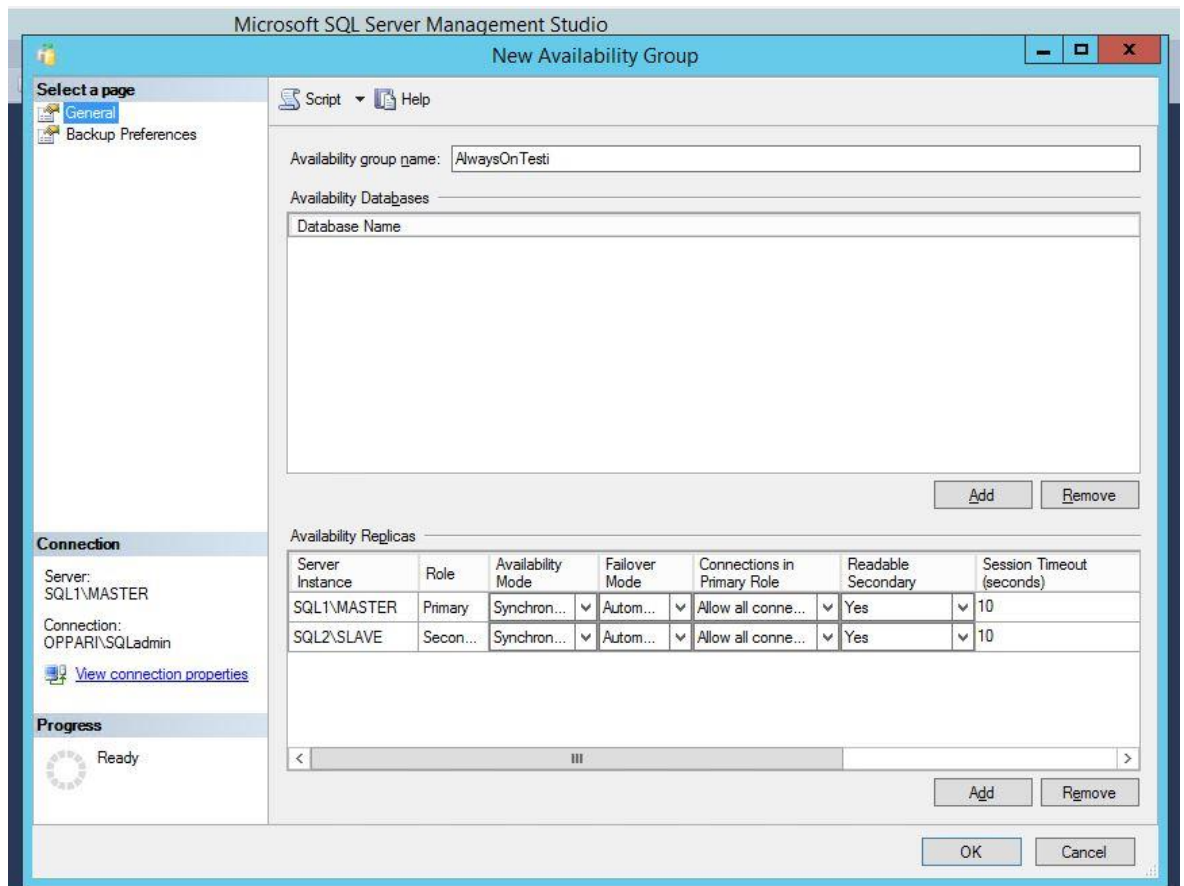


Kuva 15, Uuden Availability Groupin luonti.

Edellisessä kuvassa näkyvää Wizard vaihtoehtoa ei voi käyttää, ellei tietokantapalvelimella ole vähintään yhtä kryptaamatonta tietokantaa. Opinnäytetyötä ja testausta tehdessä huomasin, että Wizard vaihtoehto on luotettavampi sillä toiminto validoi yhteydet ja palvelimet.

Availability groupin luonti ikkunassa voidaan määrittää mitkä tietokantapalvelimet liittyvät availability grouppiin sekä availability grouppiin voidaan tässä vaiheessa liittää kryptaamattomia tietokantoja. Kryptattuja tietokantoja ei voida tässä vaiheessa liittää Availability grouppiin, kryptatut tietokannat pitää liittää Transact-SQL lausein.

Kuten Kuvassa 13 näkyy Palvelimet SQL1\MASTER sekä SQL2\SLAVE ollaan liittämässä kyseiseen "AlwaysOnTesti" nimiseen Availability grouppiin. Yhtään tietokantaa ei tässä tapauksessa olla liittämässä, sillä niitä ei vielä edes ole luotu.



Kuva 16, Uuden Availability groupin luonti ikkuna

3.5.1 Kryptatun kannan liittäminen Availability Grouppiin

Esivaatimukset ennenkuin kryptattu kanta voidaan liittää Availability grouppiin ovat seuraavat; Olemassa oleva Availability Group, jossa vähintään yksi ensisijainen ja yksi toissijainen palvelin, Toimiva TDE kryptattu tietokanta ensisijasella palvelimella sekä Database Master Key jokaisella palvelimella jotka ovat Availability Groupin vaikutuksen alla.

Kryptatun kannan liittäminen Availability grouppiin ei ole yksinkertainen asia, opinnäytetyössäni kyseistä asiaa jouduin tarkoin tutkimaan jotta, liittäminen onnistuu ongelmitta. Prosessi ei paljoa eroa siitä, että luodaanko uusi tietokanta vai palautetaanko olemassa oleva tietokantakanta.

Lyhyesti selitettynä prosessi menee näin: ensisijaisella palvelimella luodaan tai palautetaan kanta, se kryptataan, otetaan varmuuskopio kannasta ja lokeista, liitetään availability grouppiin, tehdään toissijaisella palvelimella tietokannan sekä lokien palautus RESTORE komennolla norecovery tilaan ja liitetään availability grouppiin.

SQL-komennot vaiheittain. Vaiheet 1-4 suoritetaan ensisijaisella palvelimella ja vaiheet 5-6 toissijaisella palvelimella.

Vaihe 1, tde nimisen tietokannan luonti.

```
create database tde
go
```

Vaihe 2, tde tietokannan salaus.

```
use tde
CREATE DATABASE ENCRYPTION KEY
WITH ALGORITHM = AES_256
ENCRYPTION BY SERVER CERTIFICATE TDECert
go
USE master
GO
--Asetetaan tde kantaan salaus päälle
ALTER DATABASE tde SET ENCRYPTION ON
go
```

Vaihe 3, luodaan varmuuskopio kannasta ja lokeista SQL1 palvelimen C:\SQL\backup kansioon.

```
USE master
go
BACKUP DATABASE tde TO DISK = 'C:\SQL\backup\tde.bak';

USE master
go
BACKUP LOG tde TO DISK = 'C:\SQL\backup\tde_log.trn';
```

Vaihe 4, Liitetään tde tietokanta Availability grouppiin AlwaysOnTesti

```
USE master
go
ALTER AVAILABILITY GROUP [AlwaysOnTesti] ADD DATABASE [tde]
```

Vaihe 5, Kannan palautus toissijaisella palvelimella, tämä SQL-lause ei aina toimi, joten restore Wizardin käyttäminen on suotavaa, tärkein asia on palauttaa kanta NORECOVERY tilaan.

```
USE master
go
RESTORE DATABASE tde FROM DISK = '\\SQL1\sql\backup\tde.bak' WITH NORECOVERY;
```

```
USE master
go
RESTORE LOG tde FROM DISK = '\\SQL1\sql\backup\tde_log.trn' WITH NORECOVERY;
```

Vaihe 6, Liitetään toissijasella palvelimella juuri palautettu tde kanta availability grouppiin

```
USE master
go
ALTER DATABASE tde SET HADR AVAILABILITY GROUP = [AlwaysOnTesti];
```

Jos on jo valmis tietokanta, joka halutaan liittää availability group:iin, niin Prosessi on hyvin samanlainen, ainoa ero on että vaiheessa 1 ei luoda uutta kantaa vaan tehdään tietokannan palautus käyttäen ohjattua SQL:n palautustoimintoa ja sitten jatketaan vaiheesta 2.

3.5.2 Availability Group Listener

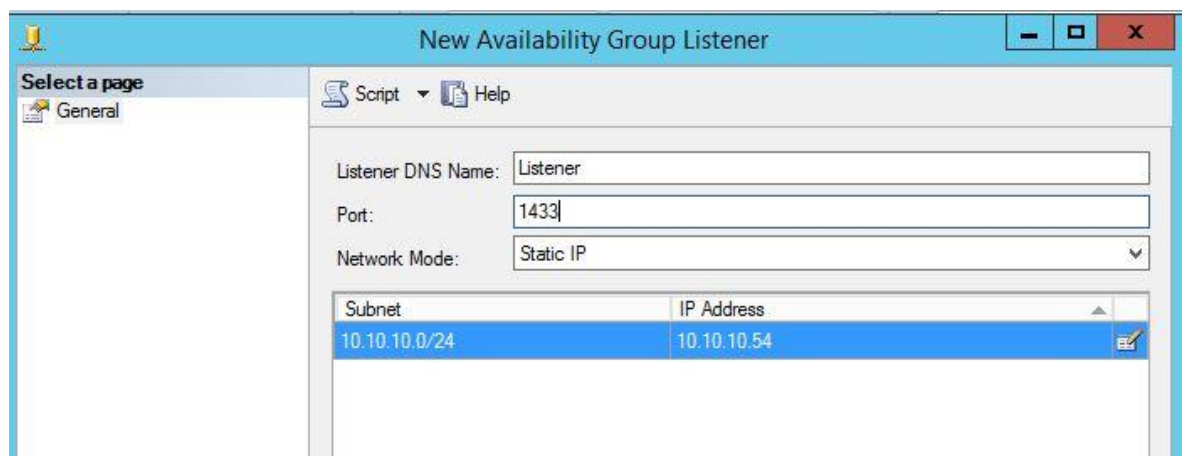
Ratkaisuun täytyi myös luoda Availability group listener, jonka tarkoitus on toimia ”porttina” tietokantoihin. Esivaatimukset Listenerin luomiseen ovat seuraavat; Availability group listener vaatii oman IP-osoitteen, Active directoryyn virtuaalisen konetilin sekä DNS recordin. Näiden toimintojen tekemiseen vaatii toimialueen järjestelmänvalvojan tunnukset.

Listenerin luominen tapahtuu SQL Management studiossa, AlwaysOn High Availability kohdasta, valitsemalla Add Listener.



Kuva 17, Availability group Listenerin luonti

Availability group Listener luotu nimellä Listener, asetettu vapaa staattinen iposoite ja asetettu kuuntelemaan TCP porttia 1433, portti 1433 sen takia koska se on oletuksena portti jota SQL käyttää TCP yhteyksillä.



Kuva 18, Availability Group Listenerin asetusten määrittely.

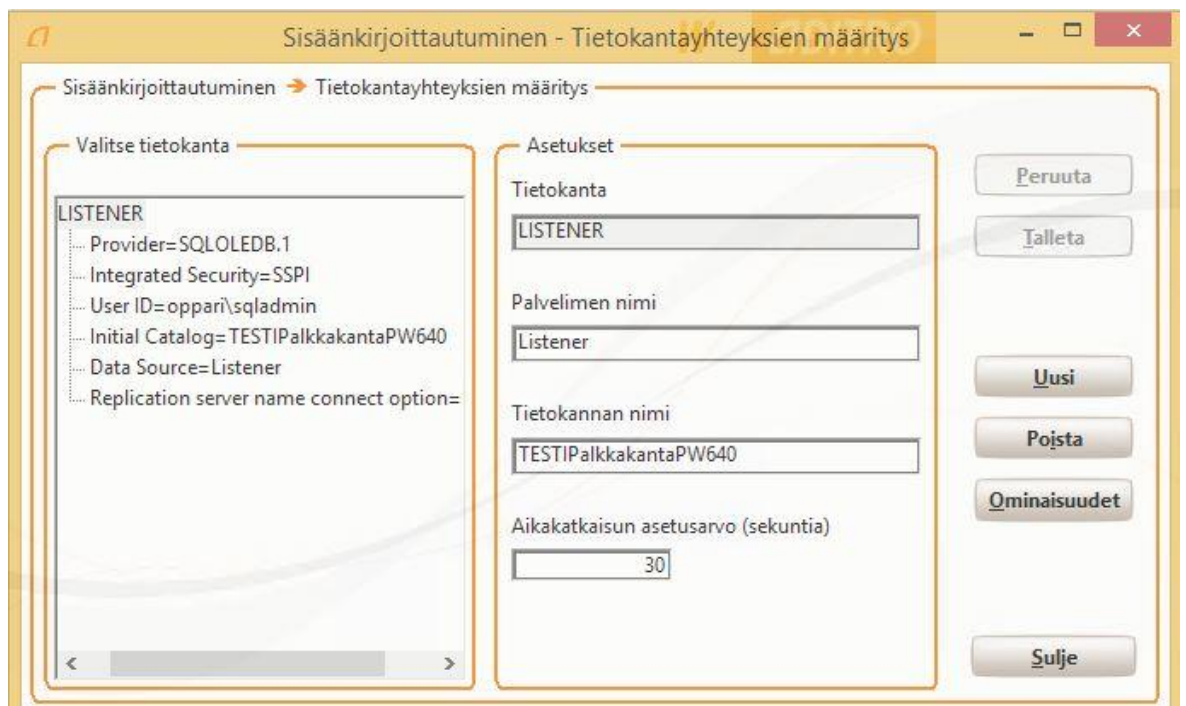
3.6 Testaaminen Personec W ohjelmalla

Testaamisen tarkoituksena oli testata kolmea asiaa. Ensimmäinen testi on ottaa personec W ohjelmalla yhteys Availability group listeneriin. Toinen testi on muokata personec W:llä sattumanvaraisesti valittua tietuetta ja katsoa näkyykö vaikutukset ensisijaisessa sekä toissijaisessa tietokannassa. Kolmas testi on testata miten Personec W käyttäytyy, kun ensisijainen tietokanta ajetaan alas.

Yhteyden testaamiseen käytetty kolmea eri käyttäjää jotka ovat Oppari\Arnold, Oppari\SQLAdmin sekä tietokannan system administrator eli SA tunnus. Jokaisella käyttäjällä on eri oikeudet toimialueella sekä tietokannoissa. Vähiten oikeuksia on Oppari\Arnold tunnuksella joka on domain user sekä tietokantaan määritetty käyttäjäryhmään palkkar ryhmä. Oppari\SQLAdmin tunnus on Domain Admin, sekä tietokannoissa kuuluu sysadmin ryhmään. SA tunnuksella ei ole toimialueella mitään oikeuksia mutta tietokannoissa täydet oikeudet.

Taulukko 1. Tunnusten järjestelmänvalvojan oikeudet.

Tunnus	Järjestelmänvalvojan oikeus Tietokantoihin	Järjestelmänvalvojan oikeus toimialueeseen
Oppari\SQLAdmin	x	x
Oppari\Arnold		
SA	x	



Kuva 19, Personec W:n yhteyden konfigurointi.

Kuten kuvassa 19 näkyy Personec W:n yhteys pitää käsin määritellä.

Tietokanta = Nimi joka näkyy Personecissa käyttäjälle.

Palvelimen nimi = Palvelimen nimi johon yhdistetään, tässä tapauksessa se on Availability group listener. Voisi olla esim SQL1 tai SQL2.

Tietokannan nimi = Tietokannan nimi SQL palvelimella

Testaamisen tulokset: Ensimmäisen testin tulokseksi voidaan todeta, että yhteys Availability Group listenerin kautta tietokantoihin toimii kaikilla käyttäjillä. Toisen testin tuloksena on, että Personec W:n kautta tehdyt muutokset näkyvät ensisijaisessa ja toissijaisessa kannassa välittömästi, joten voidaan todeta, että SQL-AlwaysOn toiminto toimii. Kolmannessa testissä ei onnistunut täysin odotetulla tavalla, ongelmana oli se, kun ajetaan alas ensisijainen tietokanta, jolloin toissijaisesta kannasta tulee uusi ensisijainen kanta niin, personec W ei kykene muodostamaan yhteyttä uuteen ensisijaiseen kantaan automaattisesti, vaan ohjelma on käynnistettävä uudelleen. Tämä ongelma todennäköisesti johtuu siitä, että personec W on hyvin alkukantainen ohjelma. Ihanne tilanteessa palvelimen vahingoittuessa tai sammussa yhteys muodostuisi uudelleen käyttäjän huomaamatta vaihdosta.

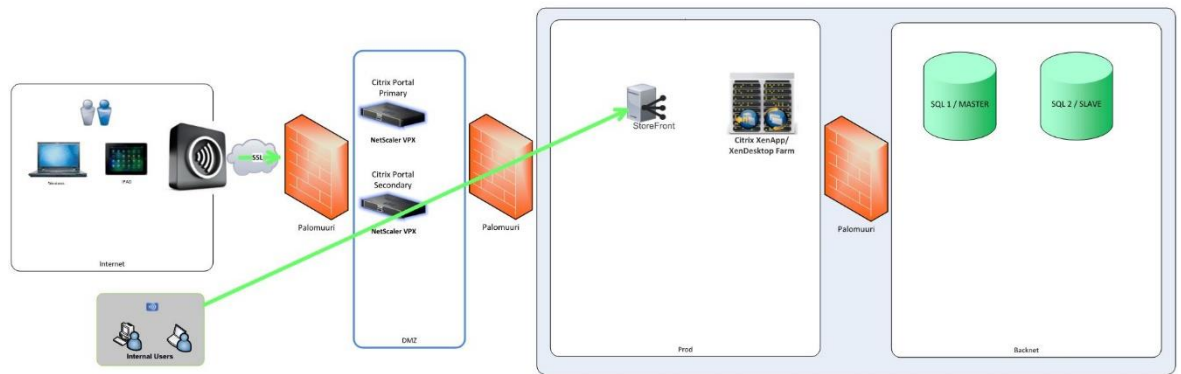
4 Citrix Testiympäristö

Tässä luvussa käyn läpi Citrix-Testiverkon toiminnan ja miten SQL AlwaysOn toiminnon käyttöönotto eroaa Hyper-V verkon käyttöönotosta.

4.1 Yleiskuva verkosta

Yrityksellä on testiverkko, joka on identtinen tuotantoverkon kanssa. Testiverkossa on tarkoitus testata palvelinten ja sovellusten toimivuutta ennen käyttöönottoa. Suurin osa yrityksen käyttämistä sovelluksista hyödyntää Citrixin virtualisointi teknologiaa.

Kuvassa 20 on prod verkko, verkosta löytyy StoreFront sekä Citrix XenApp Farm. StoreFront:illa kuvataan sovellusta joka näyttää käyttäjälle kaikki hänelle julkaistut sovellukset. XenApp Farmissa sijaitsevat kaikki sovellukset, niiden istunnot ja niihin määritetyt käyttäjäryhmät ja oikeudet.



Kuva 20, Pelkistetty kuva citrix-verkon toiminnasta

Internet on Julkinen internetverkko.

DMZ on Tulee sanasta Demilitarized Zone. Tällä tarkoitetaan verkkotasoa, joka keskustelee yrityksen sisäisenverkon ja internetin välillä. DMZ verkossa on virtuaaliset kuormantaus palvelimet.

Prod on sisäinen tuotanto verkko joka sisältää sovelluspalvelimet, tiedostopalvelimet yms. Backnet on sisäinen taustaverkko joka on suojattu ylimääräisellä palomuurilla, taustaverkossa sijaitsee pääasiassa tietokantapalvelimet.

Verkko toimii ulkopuolisesta verkosta siten että Citrix Receiverillä muodostetaan SSL yhteys Citrix Portaaliin, Portaalissa kirjaudutaan omilla käyttäjätunnuksilla StoreFronttiin, StoreFrontinssa näkyvät käyttäjälle ne sovellukset jotka on hänelle julkaistu, käyttäjän valitessa sovelluksen StoreFront ottaa tällöin yhteyden Citrix XenApp Farmiin ja käynnistää käyttäjälle istunnon, jolloin käyttäjälle aukeaa kyseinen ohjelma. Tämä StoreFrontin ja XenApp farmin käymä prosessi on käyttäjälle huomaamaton.

Sisäisen verkon käyttäjät yhdistävät SSL yhteydellä suoraan storefronttiin kun käyttäjä käynnistää Citrix Receiverin. Käyttäjältä ei tunnistautumista vaadita sillä, Citrix Receiver osaa hakea käyttäjän Active Directory tunnuksen automaattisesti. Kun kirjautuminen StoreFronttiin onnistuu ja käyttäjä valitsee haluamansa ohjelman, taustaprosessi on samanlainen kuin ulkoisen verkon käyttäjillä.

Tämän kaiken taustalla käyttäjille huomaamatta tapahtuu useita prosesseja, kuten käyttöoikeuksien tarkistuksia sekä tietokantayhteyksien luomisia.

4.2 Palvelin instanssit

Kaikki palvelin instanssi on kahdennettu takaamaan sen, että vikatilanteissa resurssit ja palvelimet eivät ole saavuttamattomissa. Kahdennukset sijaitsevat eri konesaleissa, joten

palvelimet sijaitsevat eri fyysisissä sijainneissa joka takaa sen, että resurssit pysyvät saatavutettavissa, vaikka toiseen konesaliin kohdistuisi esimerkiksi ympäristöuhka, hakkeri-isku tai sähkökatko

MPLS yhteys yrityksen sekä konesalin välillä on kahdennettu, eli käytännössä yhteydellä on kaksi eri kaapelia sekä reititintä.

Konesalien palveluntarjoajalla on tämän lisäksi vielä omat käytännöt vikatilanteissa, mainittakoon esimerkiksi omavaraisuus sähkönkäyttöön 5 päivän ajaksi, vaikka sähkön jakeluverkossa olisi katko.

4.3 SQL alwaysOn käyttöönotto

SQL-alwaysOn toiminnon rakentaminen tässä testiympäristössä olin elintärkeä saada toimintaan. Sillä on laaja vaikutus tietokantapalvelinten datan turvaamiseksi.

Asennuksia varten minulle oli luotu kaksi AD tunnusta, joilla molemmilla oli paikalliset järjestelmänvalvojan oikeudet SQL1 sekä SQL2 koneisiin. Tunnus 1 luotiin asennuksia sekä yleistä testaamista varten ja tunnus 2 luotiin pelkästään SQL palveluiden ylläpito tunnuksiksi.

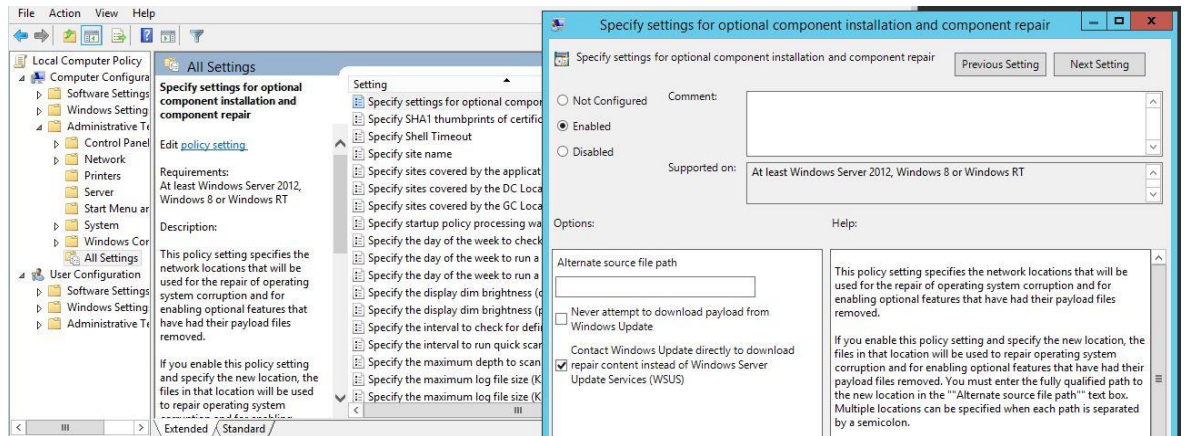
(todellisuudessa palvelinten nimet eivät ole SQL1 ja SQL2 eikä tunnukset ole tunnus1 ja tunnus2, mutta tietoturva syistä niitä en voi kertoa.)

4.3.1 Ohjelmistoasennukset

Tässä kappaleessa kerron miten komponenttien asennukset eroavat Citrix ympäristössä hyper-v ympäristöön.

.NET Framework ohjelmistokomponenttikirjaston asennus palvelimille onnistuu helpommin, sillä kyseisillä palvelimilla on pääsy verkkoon. Asennuksen yhteydessä voidaan määrittää asennusohjelma hakemaan tarvittavat tiedostot suoraan Windows update:sta, jotta

hakeminen suoraan onnistuu, vaaditaan paikallinen ryhmäkäytäntö muutos.



Kuva 21. Group policy muutos .NET Frameworkin asennusta varten

SQL Server ohjelmiston asennuksessa ei ollut eroja, ohjelma asennettiin samasta ISO-tiedostosta kuin Hyper-V ympäristössä. Kyseinen ISO-tiedosto kopioitiin tiedostopalvelimelta SQL palvelimille.

Windows Failover Clusteria asentaessa kohtasin ongelman, että minulla ei riittä oikeudet luoda kyseistä AD objektia, joten jouduin pyytämään järjestelmänvalvojaa luomaan Active Directoryyn kyseisen objektin, luonti onnistui delegoiduilla ou-admin oikeuksilla. Kun AD objekti oli luotu, asennus onnistui ongelmitta.

Availability group listeneriä luodessa kohtasin saman ongelman kuin klusteria luodessa, oikeuteni ei riittä, joten jouduin pyytämään järjestelmänvalvojaa luomaan Active Directoryyn kyseisen objektin sekä pyysin toimialueen järjestelmänvalvojaa luomaan DNS recordin listenerille. Objektin luonti onnistui, mutta listenerin luonti ei, jouduimme vielä lisäämään Active Directoryssä palvelin objektille oikeuden muokata Listenerin objektia. Näiden operaatioiden jälkeen Listenerin luonti onnistui.

Availability groupin luonti sekä tietokannan kryptaus & lisäys Availability grouppiin ei eroa mitenkään hyper-V ympäristöön.

Personec W:n käyttö testattiin tässä tapauksessa DMZ verkon portaalin kautta, kirjautumalla testi tunnuksella StoreFrontiin ja sen kautta personec W:hen. Testitunnukselle piti antaa pääsy tietokantaan ja konfiguroimaan yhteys listeneriin käsin. Normaalkäyttäjille konfigurointia ei tässä verkossa tarvitse käsin määrittää, sillä siihen löytyy scripti, jonka perusteella käyttäjälle määritetty tietokannat.

Yhteys listeneriin toimi testi tunnuksella sekä SA tunnuksella. Personecin kautta tehdyt muutokset näkyivät tietokannoissa, joten SQL AlwaysOn saatiin myös toimimaan tässä verkossa.

4.3.2 Ongelmat ja rajoitukset

Suurimmat ongelmat tässä testiverkon asennuksessa oli käyttöoikeuksien puutteellisuus. Tietoturvasyistä minulle ei voitu antaa Domain tason järjestelmänvalvojan tunnuksia joiden avulla olisi asennukset todennäköisesti onnistunut ongelmitta, lukuun ottamatta listenerin DNS-recordin luomista.

Personec W:llä ongelmia oli siinä, ettei testitunnuksella päässyt kantaan käsiksi, ongelma johtui siitä, ettei testitunnuksella ollut pääsy oikeutta tietokantoihin.

4.4 Työn Tavoitteet

Työn tavoitteena oli selvittää, miten SQL AlwaysOn ratkaisu saadaan luotua yrityksen tuotantoympäristöön ja miten tämä ratkaisu parantaa yrityksen tämänhetkistä tilaa.

Työn tavoitteet saavutettu erinomaisesti, selvitys siitä miten SQL AlwaysOn ratkaisu luodaan kyseisen yrityksen ympäristöön, on onnistuneesti selvitetty, testattu sekä raportoitu.

Tämä opinnäytetyö on osa laajempaa ”disaster recovery” suunnitelmaa. Toimeksiantaja on hyväksynyt tämän opinnäytetyön osana yrityksen suunnitelmaa, joten voidaan todeta, että tavoite on saavutettu.

5 Pohdinta

Tässä opinnäytetyön kappaleessa pohditaan työn johtopäätökset, kehittämis- ja jatkotutkimusehdotukset sekä mietitään opinnäytetyön prosessia sekä omaa oppimista.

5.1 Johtopäätökset

”Hyvin suunniteltu työ on puoliksi valmis”, henkilökohtaisesti voin sanoa, että tämä fraasi ei päde ainakaan tietotekniikassa. Sillä vaikka kuinka hyvin työn suunnittelee, niin jokin komponentti tai ohjelma aina oikuttelee. Tämänkin opinnäytetyön aikana tuli vastaan hyvin erilaisia ongelmia, liittyen käyttöoikeuksiin, lisäliitännäisten puuttumiseen sekä inhimillisen tekijän kautta saapuviin virheisiin.

Kyseisen yrityksen kannalta tämä opinnäytetyössä tutkittu ratkaisu tuo heille merkittävän edun, sillä tämä ratkaisu parantaa huomattavasti yrityksen tietokantapalvelinten tämän hetkistä tilaa, vaikka yrityksen palvelimet ovat kahdennettu, niin AlwaysOn lisää vakautta järjestelmiin.

Ongelmat yrityksen kannalta ovat siinä, että tämän ratkaisun rakentaminen heille maksaa, rakentamiseen kuluu paljon työtunteja sekä riippuen sopimuksista palveluntarjoajien kanssa saattaa siihen sisältyä lisenssimaksuja.

Mielestäni työn tavoitteet saavutettiin, sillä tämän opinnäytetyön perusteella yritys voi rakentaa kyseisen ratkaisun omaan tuotantoympäristöön.

5.2 Kehittämis- ja jatkotutkimusehdotukset

SQL AlwaysOn ratkaisua voisi tutkia hyvin laajemmin, tässä opinnäytetyössä käytiin läpi vain käyttöönotto ja hyvin pieni raapaisu järjestelmän testausta. Jatkossa kyseisestä aiheesta voisi esimerkiksi tutkia sen todellista vaikutusta tietokantapalvelimien datan eheyteen, siihen miten käyttäjälle näkyvää viivettä voisi minimalisoida, sekä sitä että miten SQL AlwaysOn otetaan käyttöön laajaan tuotantoympäristöön.

5.3 Opinnäytetyöprosessi sekä oma oppiminen

Opinnäytetyön tekeminen omasta mielestäni oli uuvuttavaa sekä palkitsevaa, omaa tekemistäni auttoi hyvin paljon se, että sain tämän työn toimeksiantona sekä se että minulla oli mahdollisuus, tehdä työtä toimeksiantajan tiloissa täyspäiväisesti. Toimeksiantajan puolesta myös projekti oli hyvin rajattu, joka helpotti tutkimusta.

Henkilökohtaisesti opin paljon uutta projektimaisista työtavoista sekä SQL järjestelmistä. Aiempaa ammatillista osaamista olen hyödyntänyt opinnäytetyöni tekemisessä, kokemusta opinnäytetyöstä on tullut myös paljon lisää.

Lähteet

Microsoft Technet, Windows Server 2008 R2 and Windows Server 2008 Luettavissa: [https://technet.microsoft.com/en-us/library/dd349801\(v=ws.10\).aspx](https://technet.microsoft.com/en-us/library/dd349801(v=ws.10).aspx). Luettu: 3.5.2016

Microsoft Technet, What's New in Windows Server Luettavissa: <https://technet.microsoft.com/library/dn250019>. Luettu: 3.5.2016

SQL Server 2012 AlwaysOn

Luettavissa: <https://www.simple-talk.com/sql/database-administration/sql-server-2012-alwayson/>. Luettu: 3.5.2016

UNDERSTANDING ALWAYSON & MIRRORING WITH SQL SERVER 2012

Luettavissa: <http://blog.fpweb.net/understanding-alwayson-mirroring-with-sql-server-2012/#.Vt6KXUKLSUk>. Luettu: 3.5.2016

MSDN Library, Transparent Data Encryption (TDE)

Luettavissa: <https://msdn.microsoft.com/en-us/library/bb934049.aspx>. Luettu: 3.5.2016

How to add a TDE encrypted database to an Availability Group (2015)

Luettavissa: <https://blogs.msdn.microsoft.com/alwaysonpro/2015/01/07/how-to-add-a-tde-encrypted-database-to-an-availability-group/>. Luettu: 3.5.2016

MSDN Library, SQL Server 2014

Luettavissa: [https://msdn.microsoft.com/en-us/library/ff929050\(v=sql.10\).aspx](https://msdn.microsoft.com/en-us/library/ff929050(v=sql.10).aspx). Luettu: 3.5.2016