# Oplevelser med opsætning

Vi har prøvet at deploye TM12 til Amazon på en Ubuntu server. Det viste sig at være yderst enkelt at oprette serveren efter beskrivelsen. Der var dog en enkelt ting. Man skal eksekvere en ”sudu apt-get update” før man installere java og ant. Selv Putty virkede uden problemer (efter vi lige læste brugervejledningen ordentlig igennem).

Det er overraskende hvor enkelt det er at få opsat en virtuel server og få den til at virke.

# Overvejelser omkring anvendelse af ”skyen”

At lægge noget op i skyen har mange fordele og ulemper. Ligesom med alle andre services hvor man overvejer at anvende en tredjeparts leverandør.

1. Kan vi stole på leverandøren?
2. Er leverandøren stabil nok?
3. Kan leverandøren levere det produkt vi har behov for? Med hvilken QoS?
4. Kan vi stole på at leverandøren vil hjælpe os hvis der er noget der ikke virker?
5. Hvis vi beslutter at skifte leverandør kan vi så det? Hvordan gemmes data? Standard, API, …

Der er andre overvejelser der er specielle for skyen.

1. Hvilken del skal vi lægge op i skyen?
2. Er der nogle juridiske spekter omkring ejerskab af data samt fysisk placering af data?
3. Hvilke udsving i QoS kan vi risikere (determinisme)?

Som med alle leverandører er det vigtigt at man stoler på firmaet. Hvis der f.eks. er tale om data omkring et fremtidigt produkt, skal man jo være sikker på at udbyderen ikke vil stjæle ens data og sælge dem til en konkurrent. Der er selvfølgelig mulighed for at anvende en passende kryptering, men hvis man ikke stoler på sin leverandør, skal man nok overveje om der findes en alternativ leverandør.

Der er naturligvis forskel på firmaer. Hvis man ønsker at sikre en global og konstant adgang til ens data er det nok ikke en god ide at vælge et mindre eller nystartet firma eller et firma med usikre regnskaber eller egenkapital. Selv om det ren juridiske med rettigheder til data er i orden, vil det utvivlsomt resultere i en længere nede periode hvis firmaet pludselig lukker dørene og man skal til at finde en alternativ leverandør.

Her er det vigtigt at kikke på ens behov. Udbyder firmaet det produkt man har behov for, med den QoS og stabilitet der er behov for? Kan det tilbyde den skalering i ressourcer og båndbredde som der er behov for? Et firma der har mange kunder vil ofte have meget bedre mulighed for ressource sharing, og dermed for at supportere korte peek perioder eller op-skalering efter behov.

Når man overdrager ansvaret for dele af ens system til en anden har det naturligvis den fordel at det er dennes ansvar at sikre at det kører efter aftale. Det er som regel derfor man vælger en anden udbyder; fordi man enten ikke selv har ressourcer til det eller det er ufordelagtigt. Disse ressourcer kan være i form af penge, erfaring (der er jo nogle der skal drifte det) eller tid (det tager tid at starte op). Om det er fordelagtigt er mere et spørgsmål om hvem der kan gøre det billigst og bedst. Selv om vi har ressourcer til at oprette og drifte vores eget system, så er det meget tænkeligt at et firma der ikke laver andet end at udbyde skyer vil kunne gøre det billigere og bedre end vi selv kan – også selv om de skal tjene penge på det. Det er et simpelt spørgsmål om stordrift. Men der hvor det som regel giver knuder er når der er noget der ikke virker. Hvis man selv drifter det kan man smide alt hvad man har i hænderne, hyre de bedste konsulenter, og få det til at virke. Men, hvis man anvender Amazon’s sky og man udgør 0,0001% af deres total omsætning, hvor meget vil de så lægge i at få løst ens problem, koste hvad det vil?

Data kan lagres på mange måder, og ofte er en sky mere end bare data, det er et specifikt API eller nogle virtuelle ressourcer der fungerer på en bestem måde. Set fra leverandørens synspunkt er det en fordel at gøre det så svært så muligt at skifte leverandør, så det er vigtigt som minimum danne sig et overblik over hvad det vil kræve at skifte leverandør, og hvilket risici det ville indebærer efter systemet er operationelt i skyen.

Når der er tale om at have data og/eller ressourcer i skyen er det vigtigt at også tænke interface. Meget ofte har firmaer nogle legacy systemer der er nødt til at holdes kørende. Kan disse overhovedet fungere i skyen? Hvis ikke, hvad vil det så betyde for QoS hvis disse systemer skal forblive internt, men data ligger i skyen (længere netværkskommunikation)? Skal vi købe et service API i skyen (SaaS), hvis et sådanne findes. Hvis et firma udbyder en end user application som passer til vores behov, så er det sandsynligvis både den hurtigste, nemmeste, mest stabile og billigste løsning. Dette skyldes at mange andre bruger samme løsning og det derfor er disse firmaers samlede ”vægt” der afgør hvor hurtigt firmaet reagere på problemer. Samtidig vi API’et og performance være skræddersyet til denne løsning og er ikke generelle interfaces som så skal bruges til at implementere end user applikationen. Dette er sammenligneligt med at en general purpose architecture (GPA) på en CPU altid er mindre effektiv end en custom application direkte i silicium. Findes der ikke en SaaS der er et perfekt match så er der måske nogle der udbyder et framework i en sky hvorpå vi kan bygge vores applikation (PaaS). Dette kræver mere arbejde, men frameworket gør at vi kan anvende nogle kendte og afprøvede moduler til at implementere vores applikation, og disse moduler vil være optimeret i forhold til den service de udbyder. Ikke så performance optimalt som SaaS, men mere fleksibelt og hurtigere end at skrive det selv. Endelig kan det være at vores applikationer så speciel at den skal skrives fra bunden, eller at vi vil forsøge at lægge en legacy applikation op i skyen. I det tilfælde kan vi købe nogle virtuelle ressourcer (Iaas), som vi så kan implementere vores applikation på. Dette tager naturligvis længere tid (hvis ikke vi har legacy koden i forvejen), og ansvaret for at applikationen fungerer og er god nok performancemæssigt påhviler fuldt og fast os selv, ligesom ressourcerne man køber er general purpose, hvilket betyder at de skal kunne bruges til ”alt”, og derfor ikke kan være optimerede mod en speciel brug.

Dette kan sammenlignes med at arbejde på et OS på en GPA (IaaS). Alternativt kan man skrive applikationen direkte på stålet ud fra et library fra producenten (PaaS). Og endelig kan man købe en chip der er skræddersyet til ens behov (SaaS).

Meget ofte er data ikke bare data. Der kan være tale om nationale hemmeligheder eller personfølsomme data, som er beskyttet af specielle juridiske regler for hvor og hvordan disse oplysninger fysisk skal lagres og slettes. Når man arbejder i skyen er man som udgangspunkt ligeglad med hvor ens data lagres, men disse juridiske aspekter kan gøre det nødvendigt at sikre at data fysisk lagres et bestemt sted, og at data når den slettes gøres efter de rigtige retningslinjer (f.eks. DoDs regler for data sletning). Hvis en server i en virtuel farm går ned flytter data og server øjeblikkeligt over på en anden server, da data gemmes redundant (raid), men hvad med den server der er gået ned? Hvis den kommer op og fungere igen, vil den så blive brugt til noget andet hvor data evt. kan genskabes af den nye bruger? Eller risikere vi at den fysiske server/disk smides på lossepladsen hvor dens data evt. kan genskabes? Alt efter de juridiske krav er det vigtigt at tage højde for disse ting i kontrakten.

Med virtuel placering af data i skyen kan det være meget svært at garantere svartider af to på hinanden følgende kald, specielt med load distribution og multiple servere. Udbyderen vil måske garantere en maximum svartid, men da det går imod ideen om skyen at holde tæt styr på den fysiske placering af de forskellige virtuelle servere. Ofte er det rigeligt at kende den maksimale svartid, men i specielle tilfælde kan det være vigtigt at svartiden er deterministisk, eller ikke varierer ret meget mellem forskellige servere. I dette tilfælde kan det være problematisk at anvende skyen.