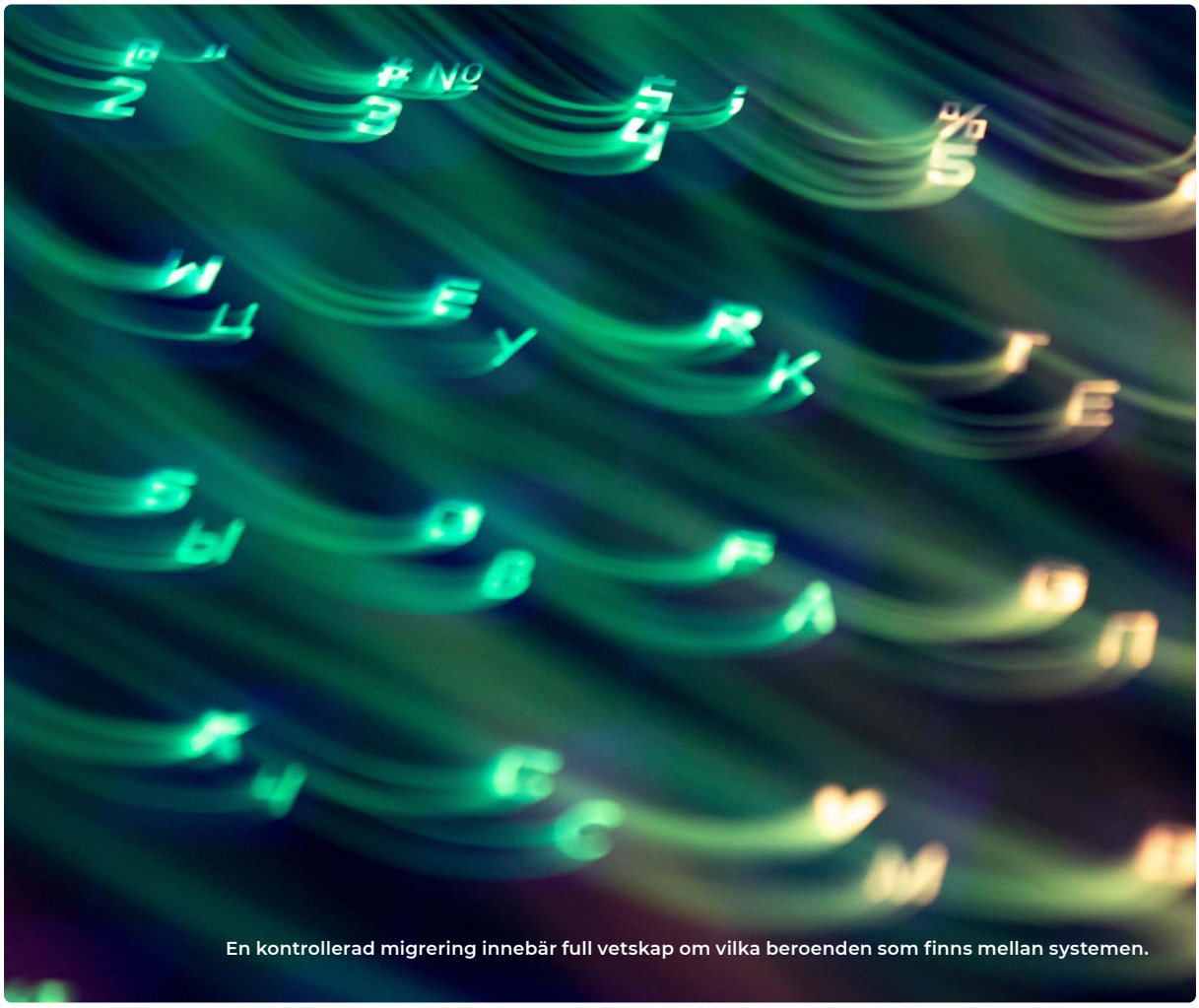
safespring

WHITE PAPER

Migrera från Azure Kubernetes Service till CK8s på Safespring

Safespring och Compliant Kubernetes



En kontrollerad migrering innebär full vetskap om vilka beroenden som finns mellan systemen.

Migrering till Compliant Kubernetes

Detta dokument sammanställer de steg som bör tas för att migrera från Azure Kubernetes Service.

Motiven till en sådan migrering är många. Att svenskt och europeiskt lagrum gäller för GDPR-efterlevnad, tillgången på expertsupport på svenska och att datat vilar tryggt i Sverige är några av dessa. Det starka säkerhetsfokuset inom Compliant Kubernetes är ytterligare en.

En migrationsplan innehåller med nödvändighet en inventeringsfas, upptäckandet av beroenden och hur dessa kan bytas ut, planering av arbetet och tester som säkerställer funktionalitet. Därefter kan migreringen inledas och verifieras med hjälp av de kravställande testerna. Kontinuerlig dokumentation och uppföljning gör att viktiga lärdomar inte går förlorade.

När migreringen väl är genomförd väntar systemadministration och övervakning i en ny miljö. Verktygen för detta har också presenterats i dokumentet och även hur de tillsammans verkar för att ge en helhetslösning med fokus på säkerhet och smidiga agila utvecklingsprocesser.

Innehåll

Bakgrund

Molntjänster har revolutionerat hur många företag arbetar idag.

Flexibiliteten i att som tjänst kunna köpa funktioner som förut inte fanns eller som var svåra att bygga själv har givit många företag ny innovationskraft och förenkling av processer. Samarbetsfunktioner, centraliserad hantering av data och dokument har löst problematiken kring vilken som är den senaste versionen av ett dokument. IT och utvecklingsavdelningar kan med några få klick slå på nya funktioner som stödjer komplicerade eller helt nya processer.

Majoriteten av de molnplattformar som företag använder idag är amerikanska. Dessa aktörer har växt till stora jättar med enorm innovationskraft och är en stor anledning till att vi idag arbetar på ett helt nytt sätt i organisationerna. Problemet är att lagrummet inom EU och USA inte är kompatibla när det kommer till hur persondata ska hanteras. Inom EU bygger GDPR (Dataskyddsförordningen) och andra lagar inom informationssäkerhet på EU:s grundlag som ger individen stor kontroll över sin data. I USA är utgångspunkten istället lagar som ger amerikanska myndigheter stora möjligheter att infiltrera det data som användarna lämnar ifrån sig för att upprätthålla nationens säkerhet.

De skilda utgångspunkterna skapar en krock som juridiskt inte är helt lätt att reda ut. För mer information om det här ämnet rekommenderars Safesprings white paper om Schrems II (**https://www.safespring.com/schrems**) som beskriver den senaste utvecklingen i samband med EU domstolens ogiltigförklarande av Privacy Shield, som de senaste åren har varit den överenskommelse som användandet av amerikanska molntjänster inom EU har vilat på.

Kvar står nu ett antal företag och organisationer som med ett fundament av molntjänster har anammat ett nytt sätt att arbeta utan laglig grund att använda dem. Det är en svår sits eftersom det inte är enkelt att gå tillbaka samtidigt som organisationer måste följa lagen.

Lagrummet inom EU och USA skiljer sig åt när det kommer till hur persondata ska hanteras.

Möjlighet att bli oberoende

Ramverk har utvecklats som tar bort beroendena till den underliggande molntjänstleverantören. Ett sådant ramverk är Kubernetes som är en orkestreringsplattform för containerteknologi med standardiserade gränssnitt för hur applikationer kan driftsättas och underhållas. Kubernetes skapar en grundplatta uppe på vilken applikationer kan hanteras genom standardiserade definitioner. Om det låter tekniskt så kan det sammanfattas med att Kubernetes hjälper organisationer att på ett standardiserat sätt hantera applikationer och tjänster med hög driftsäkerhet. Genom att systemen och dess beroenden är definierade med kod går det att ta hjälp av den kunskap som finns på internet och enkelt ta i drift komplicerade system som kan ersätta de tjänster som finns hos de etablerade molntjänstleverantörerna. Det är alltså enklare att köra de tjänster själv som organisationen har blivit beroende av.

Det kommer också fler och fler applikationer som ersätter de mer användarnära tjänsterna såsom Office 365, OneDrive eller Dropbox. Om organisationen använder Kubernetes för att köra sina applikationer och tjänster så blir driftsättande och underhåll av dessa applikationer hanterbart.

Safespring är en molntjänstleverantör med datacenter i Sverige vilket gör juridiska krockar med amerikanska lagar en ickefråga. Tillsammans med vår partner Elastisys har vi tagit fram ett gemensamt erbjudande, Compliant Kubernetes eller Ck8s. Det är en managerad tjänst som ger organisationer den grundplatta som möjliggör frigörelse till den underliggande molntjänstleverantören. Om ett företag i sin nuvarande molntjänstleverantör redan använder Kubernetes så blir migreringen enklare eftersom det då går att återanvända all kod som beskriver systemen och tjänsterna som körs.

Detta white paper beskriver hur en migrering från Microsoft Azure Kubernetes Service (AKS) ser ut. Utgångspunkt är att organisationen redan kör Kubernetes i Azure. Flera av stegen är applicerbara även för organisationer som inte använder Azure Kubernetes Service i dagsläget. Med utgångspunkt att Kubernetes fortsätter att vara det lingua franca för drift av containeriserade applikationer, är fördelen att köra det i organisationen uppenbara. Allt arbete som läggs ned på att migrera till en standardiserad plattform kan återanvändas om organisationen skulle vilja flytta sin infrastruktur någon annanstans eftersom samma definitioner för infrastrukturen kan användas så länge som mottagarplattformen också är Kubernetes. Det skapar en flexibilitet och oberoende som annars är svår att uppnå.

Fördelarna med Open Source

En stor anledning att många använder sig av molntjänster är att det finns användbara extratjänster som minskar time-to-market. Lika mycket som dessa tjänster minskar produktionstiden ökas dock beroendet till molnleverantörernas ekosystem. Ett alternativt sätt att minska produktionstiden för sina tjänster, samtidigt som man minskar leverantörsberoende, är att implementera system som ligger utanför ens kärnleverans med open source. Båda tillvägagångssätten låter dig fokusera på din applikation och lämna stödsystem åt sidan medan tillvägagångssättet med open source minskar beroendet istället för att öka det. Open source bygger på kollaboration och genom att engagera sig i de projekt man använder (främst genom att posta tillbaka de buggfixar och förbättringar man gör) så granskas det man bidrar med för större trygghet och säkerhet. Att andra som använder projekten gör samma sak skapar en kontinuerligt uppdaterad kodbas granskad av många utan licenskostnader. Genom att många använder projekten finns det också mycket färdig kod och lösningar för att driftsätta och underhålla systemen bara några sökningar bort.

Compliant Kubernetes

Compliant Kubernetes är en CNCF (Cloud Native Computing Foundation) certifierad Kubernetes distribution som är fritt tillgängligt både som open source och som en fullt managerad tjänst på Safespring. Open source-lösningen passar organisationer som gärna driftar Kubernetes och kringliggande teknikstack själva men vill dra nytta av en säkerhetshärdad Kubernetes-distribution speciellt anpassad för reglerade branscher, samtidigt som de slipper underhåll och kan förlita sig på kvartalsvis uppdateringar av paketeringen av Kubernetes och kringliggande projekt. Open source-varianten är också ett bra komplement till en managerad tjänst för dem som behöver leverera sin mjukvara i kombination av i egna serverhallar, ute hos kunder och i publika moln och vill göra det på ett sömlöst sätt med full regelefterlevnad. För kunder som önskar det tillhandahåller vår partner Elastisys både 8/17 och 24/7 support.

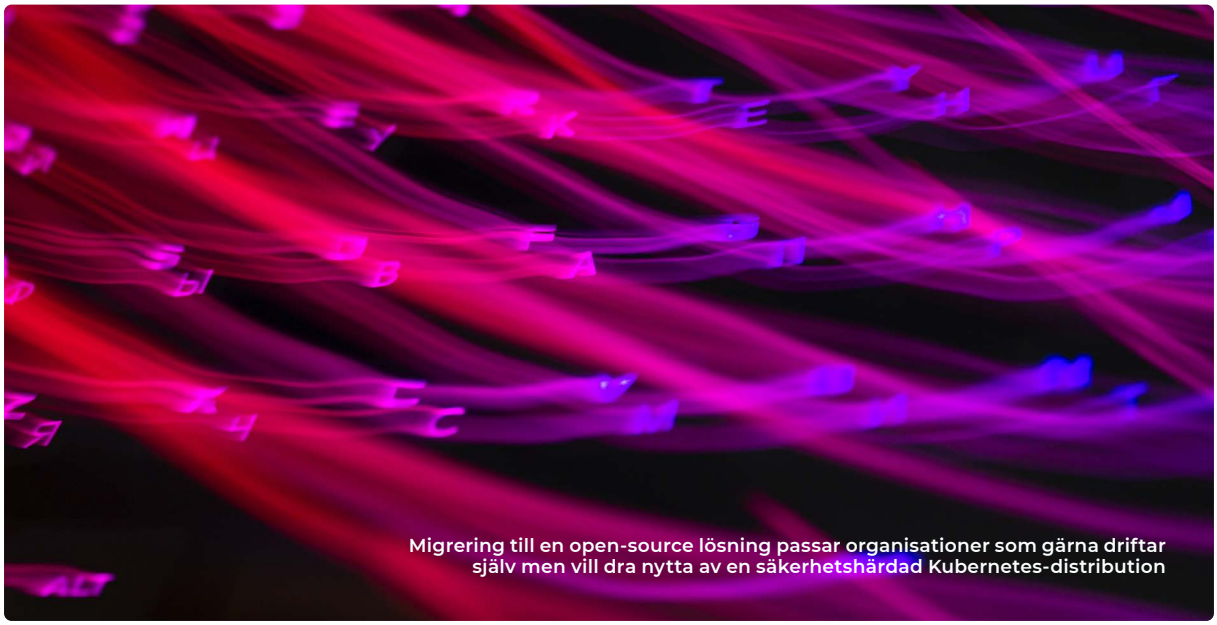
Compliant Kubernetes som öppen källkod

- **KÄLLKOD** https://github.com/elastisys/compliantkubernetes
- **DOKUMENTATION** https://compliantkubernetes.io/

Förutsättningar

För att kunna köra applikationer i Compliant Kubernetes gäller följande förutsättningar:

1. Konto till Safespring Compute och eventuellt Safespring Storage om objektlagring ska användas.
2. En eller flera domäner registrerade hos en registrar som kan peka ut tjänsterna. Compliant Kubernetes utnyttjar external-dns och cert-manager för att dynamiskt hantera såväl applikationers domännamn som automatisk certifikathantering, **så en registrar som stödjs av external-dns är att föredra**.
   - *Då hantering av domännamn inte innebär att kunders personinformation exponeras så går det att ur GDPR-synpunkt att stanna hos sin registrar, så länge den har ett kompatibelt API.*
3. Undersök vilken version av Kubernetes som körs i Azure Kubernetes Service (AKS) idag. För att undvika överraskningar är det viktigt att köra samma version i Compliant Kubernetes.



Migrering till en open-source lösning passar organisationer som gärna driftar själv men vill dra nytta av en säkerhetshärdad Kubernetes-distribution

Migrationsplan

Det här avsnittet tar upp de steg som bör tas innan själva migreringen sker.

Inventering av system som körs i organisationen

Varje migreringsprojekt startar med en inventering av de tjänster och system som körs inom organisationen. Även om det som körs i Azure Kubernetes Service (AKS) idag bara är en delmängd så kan det finnas beroenden till andra system. Exempel på system som kan skapa beroenden är:

1. **AFFÄRSLOGIKSYSTEM** Den här typen av system kan ibland hänga kvar länge och därför kan det finnas beroende till dessa på alla möjliga ställen. Körs dessa system i Azure idag eller körs det rent av in-house eller hos en annan hostingpartner?
2. **INTEGRATIONSFUNKTIONER** Den här typen av system finns ibland för att lösa små, specifika uppgifter. De har ofta tillkommit för att integrera ett system med ett annat. Det kan vara värt att kolla upp hur den här typen av system anropas och från var.
3. **DATABASER** Dessa används ofta av många system och beroende på hur stringent uppdelningen mellan olika domäner har varit så kan databaser anropas från system som egentligen inte tillhör den systemdomän där databasen ligger. Genom att gå igenom databaskopplingar och loggar går det att få en uppfattning hur databaserna används i organisationen. Om det inte redan är gjort kan konsolidering av databaser är ett projekt som körs innan själva migreringen görs för att förenkla processen.
4. **MAILSYSTEM** Det är väldigt många system som använder sig av mail för att kommunicera status eller om något går fel. Vissa av dessa mail kan till och med läsas maskinellt av andra system vilket gör dem till en länk i ett processflöde. Det kan vara så att dessa konton ligger registrerade i andra domäner än de för publika mailkonton. Genom att gå igenom vilka domäner och konton som används för den här typen av kommunikation kan obehagliga överraskningar undvikas.
5. **STÖDFUNKTIONER** Till system i den här kategorin tillhör DNS (namnuppslag), NTP (tidsynkronisering) och olika typer av service discovery system. Många av dessa körs säkert i Azure idag men det är viktigt att identifiera om de också körs internt någonstans.
6. **INTERNA APPLIKATIONER** Alla system kanske inte har migrerats till Azure (kanske tidrapportering eller internwebb). Det kan finnas olika beroende gömda i dessa system som är viktiga att identifiera.

Inventera hur säker kommunikation mellan systemen hanteras. Där finns det två typiska val:

- Virtual Private Networking (VPN), som gör att all kommunikation till och från Azure och den interna miljön går genom en VPN-tunnel, eller att
- applikationerna själva ansvarar för säker kommunikation, genom att använda TLS eller liknande protokoll.

Om VPN används så kommer en ny VPN-tunneln behöva sättas upp mellan den interna miljön och Safesprings miljö. Det kan göras i förväg så att kommunikationen är uppe när systemen flyttas över. I migreringsfasen kan också ytterligare en VPN-tunnel behöva sättas upp mellan Azure

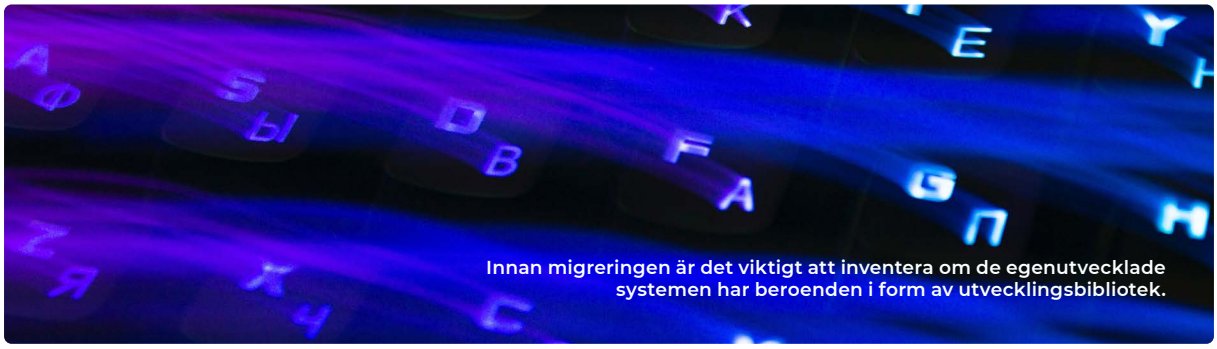
och Safesprings miljö i det fallet systemen ska kunna flyttas över ett i taget.

Om det andra alternativet används så blir det enklare eftersom det då bara är att peka om kommunikationen till Safesprings miljö med en förändring av en DNS-post. Det kan vara värt att titta på det här alternativet även om VPN-tunnel används idag eftersom alla typer av migreringar blir enklare om applikationerna hanterar den säkra kommunikationen själva.

Inventering av tjänster som körs i Azure

Inventera beroenden för de tjänster som kör i Azure.

1. **IDENTITETSHANTERING** Hur hanteras identitetshantering och rättigheter? Används Azure AD och om det används anropas det från tjänsterna som kör i Azure Kubernetes Service (AKS)? Ett steg som kan förenkla senare är att aktivera Secure LDAP (som är ett standardiserat protokoll) på Azure AD och anpassa tjänsterna så att de använder det istället. Då kommer det gå mycket lättare när migreringen ut från Azure AD sker.
2. **OBJEKTLAGRING** är ett praktiskt sätt att billigt lagra filer som system använder. Om objektlagring redan används i form av Azure Blob Storage kan datat migreras till Safespring Storage som är S3-kompatibelt. Det kommer att behöva göras anpassningar för att systemen ska använda Safesprings tjänst istället. Det kan vara värt att kolla upp om systemen är designade så att det är enkelt att ändra URI till objektlagringstjänsten på ett ställe med en variabel. Om så inte är fallet kan det vara värt att lägga ned lite arbete på att göra så att systemen är anpassade på det sättet då det blir mycket enklare att peka om längre fram. Om objektlagring inte används i organisationen idag så kan det vara värt att titta på att börja göra det även om det projektet med fördel läggs efter migreringen för att minimera frihetsgraderna.
3. **VIRTUELLA MASKINER** Körs alla system i Azure som containrar eller finns det vissa system som körs som separata virtuella maskiner? Om det är så är det bra att undersöka hur dessa maskiner är uppsatta och om det finns något enkelt sätt att replikera konfigurationen på dem. Det finns olika sätt att migrera virtuella maskiner “as is” med snapshots men det är att rekommendera att sätta upp maskinerna från början hos Safespring för en bättre integration med plattformen.
4. **DATABASTJÄNSTER** hos Azure. Om dessa används så är det bra att undersöka vilken variant som körs (MySQL, MariaDB eller PostgreSQL eller Microsoft SQL). Alla dessa varianter kan man köra själv på Safesprings infrastruktur. MariaDB och PostgreSQL går att erhålla som databas som tjänst genom Ck8s-erbjudandet. För hög tillgänglighet till dessa så är det att rekommendera att någon form av kluster används. För MySQL och MariaDB är det Galera som används. PostgreSQL och Microsoft SQL har egna inbyggda lösningar.
5. **HEMLIGHETSHANTERING** Ett bra sätt att ta bort lösenord och nycklar från själva systemen är att använda ett centralt system för hemlighetshantering. Azure Kubernetes Service (AKS) erbjuder i egenskap av att vara Kubernetes-baserat hantering av Secrets. Dessa kan användas på samma sätt i Compliant Kubernetes. I Azure finns även den specifika tjänsten Key Vault. En motsvarighet till den tjänsten är programvaran Vault av företaget Hashicorp. Det behöver göras anpassningar i tjänsterna för att byta till Hashicorp Vault och det är viktigt att identifiera andra system som också använder den här funktionaliteten.
6. **MEDDELANDEBUSS** eller meddelandeköer. Asynkron kommunikation mellan tjänster sköts ofta med hjälp av ett meddelandebussystem eller ett system för meddelandeköer. I Azure finns tjänsten Service Bus. Safespring erbjuder inte en



Innan migreringen är det viktigt att inventera om de egenutvecklade systemen har beroenden i form av utvecklingsbibliotek.

motsvarande tjänst, men rekommenderar att kunder installerar ett RabbitMQ-kluster. Detta kan köras inom Compliant Kubernetes och RabbitMQ är kompatibelt med Azure Service Bus i och med att båda stödjer samma API (AMQP 1.0). Därmed bör en migrering vara relativt okomplicerad och främst kräva att den nya tjänsten pekas ut i applikationernas konfiguration. Ett modernt alternativ med överlägsen prestanda och avancerad funktionalitet är NATS, men dock är det inte API-kompatibelt med Azure Service Bus.

Upprätta beroendematris

En kontrollerad migrering innebär full vetskap om vilka beroende som finns mellan systemen. Det visar i vilken ordning systemen migreras och vilka system som är mer centrala än andra. Beroenden kan ibland smyga sig in på oväntade ställen så en noggrann genomgång av hur tjänsterna hos Azure är konfigurerade och vilka tjänster som används i egenutvecklade system kommer att betala sig när det är dags att migrera.

Dolda beroenden finns vanligtvis kring centrala system, såsom identitetshantering (Azure AD), meddelandebussar och/eller databaser.

Det är också viktigt att inventera om de egenutvecklade systemen har beroenden i form av utvecklingsbibliotek. Om ett bibliotek anpassat för Azure har använts så behövs det bytas ut till något som är agnostiskt mot den underliggande plattformen. Detta kan skapa behov av anpassningar i själva applikationen.

Avgöra vilka tjänster som ska ersättas

Det finns många inbyggda system som har en motsvarighet byggd med öppen källkod. På sida 10 finns en samling av circa 20 stycken. I det här steget sätts också en lista på vilka tester som ska genomföras för att definiera vad som är en lyckad migrering.

Planering och rangordning

Efter en genomförd beroendeanalys kan planering göras av hur systemen ska migreras. Ofta kommer migreringen innefatta någon form av servicefönster då tjänsterna är nere så det är viktigt att planera allt som ska göras och i vilken ordning. Ingångsvärden till det här steget kommer också från test- och säkerställandefasen.

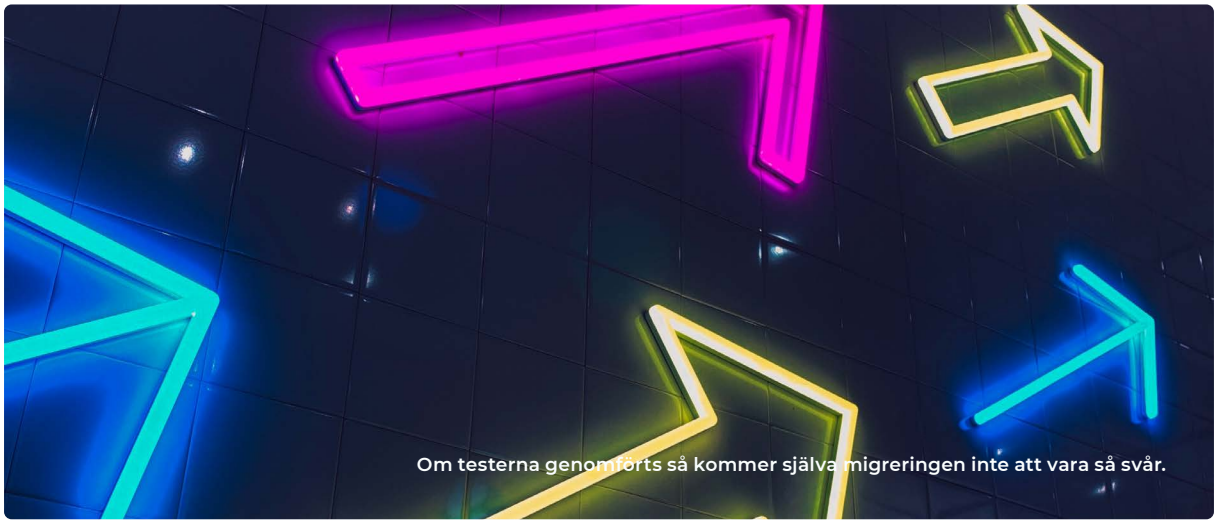
Test och säkerställande

Det första som ska testas är själva tjänsterna som kör i den nya plattformen. När det fungerar är målbilden klar och då testas migrering till testmiljön för att få en uppfattning om vilka steg som behövs för en lyckad migrering.

Efter detta ska också lasttester som speglar produktionslasten i möjligaste mån göras. Självklart gäller att ju närmare produktionslast som uppnås i testerna desto mindre risk för överraskningar när väl migreringen genomförs.

Tjänster i Azure och deras open source-motsvarighet

Tjänst i Azure	Funktion	Open source alternativ	Managerat alternativ: Safespring
Azure Kubernetes Service (AKS)	Managerad Kubernetes	Compliant Kubernetes	✔
Azure Virtual Machine	Virtuella maskiner där Kubernetes kör (master och worker noder)		✔
Azure Blob Storage	Objektlagring		✔
Azure Mysql, **Azure MariaDB,** **Azure PostgreSQL**	Databaser	Galera-kluster (för MySQL eller MariaDB) med ProxySQL som kör i Kubernetes eller i separata virtuella maskiner	✔
Azure Service Bus	Meddelandefunktion för kommunikation mellan tjänster	RabbitMQ eller NATS som kör i Kubernetes eller i separata virtuella maskiner	✔
Azure Monitor	Monitorering	Prometheus + Grafana	✔
Azure Monitor	Loggning	Elasticsearch	✔
Azure Container Registry	Container register	Harbor	✔
N/A	Intrångsdetektering	Falco	✔
Azure AD Domain Services	Hantering av organisationens användare, resurser och deras rättigheter	OpenLDAP	
Azure Active Directory	Identity Provider	Dex	✔
Azure Key Vault	Hanterar hemligheter på ett centralt och säkert sätt	Hashicorp Vault	
Azure Cosmos DB (Table API)	Key-value store	**TiKV**	
Azure Functions	Serverless runtime	**OpenFaaS** / **OpenWhisk**	
Azure Service Fabric Mesh	Service mesh	**Linkerd** / **Istio**	
Azure Virtual Network	Private networking	**Calico**	
Azure DevOps Pipelines	CI/CD	Jenkins, ArgoCD, med flera.	



Migrering

Om testerna genomförts så kommer själva migreringen inte att vara så svår.

Vid en migrering så kan det dyka upp oväntade händelser som inte kunnat förutses. Typiska saker som kan dyka upp är att testdatabasen inte är identisk med produktionsdatabasen vilken kan ge oväntade effekter. Andra vanliga problem är att en annan uppsättning nycklar och hemligheter använts i produktion än i test som kanske måste uppdateras om tjänsterna inte använder en central hemlighetshanterade (t ex Hasicorp Vault) fullt ut.

Implementation lastbalanserare

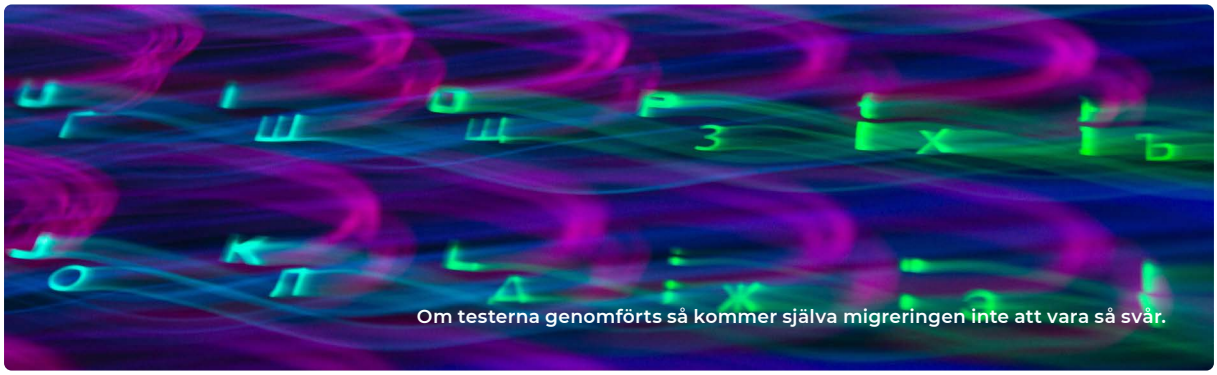
För att säkerställa hög tillgänglighet för produktionslast så kommer en lösning för lastbalanserare att behöva sättas upp. Safespring kan tillhandahålla en lösning där ni får tillgång till två eller fler virtuella maskiner som kan balansera lasten över specifika instanser som kör i plattformen. Tjänsten som sådan inbegriper några manuella steg vid uppsättning men är lätt att hantera när den väl är i drift. Vilken programvara som skall användas för lastbalanserare är valfritt men de mest populära valen är HAProxy eller Traefik. Det går även att installera MetalLB, för att få ett system som erbjuder en Kubernetes-levererad och -kompatibel tjänst som ger dynamisk lastbalanseringsfunktionalitet.

Uppföljning

Efter att migrering är gjord genomförs testerna från listan som definierar en lyckad migrering. Enhetstester som har skapats för att testa systemen före och efter migrering skall köras för att säkerställa att all funktionalitet fungerar som den skall. I de fall det uppstår avvikelser gås de igenom för att utröna om några ytterligare anpassningar behöver göras innan driftsättning.

Dokumentation

Dokumentation ska föras under hela processen men det behövs också ett separat steg för att sammanställa det som har producerats. Förutom dokumentation om hur saker och ting är uppsatta och hur systemen interagerar så är det också viktigt att få med lärda erfarenheter.



Efter genomförd migrering

Drift och bevakning av dina applikationer efter migreringen ser till att du har kontroll efter migreringen

Drift och bevakning

Applikationer i Compliant Kubernetes bevakas på två sätt:

1. Mätvärden och monitorerings-data sparas i Prometheus och visualiseras i Grafana.
2. Applikationers loggar sparas i ett Elasticsearch-kluster och visualiseras och behandlas i Kibana.

Dessa programvaror åtnjuter stort stöd från det globala DevOps-communityt och ses allmänt som best practice att använda för dessa uppgifter i Kubernetes-sammanhang.

Många programvaror exponerar mätvärden i Prometheus-specifikt format just för att systemet är så förankrat i communityt. Adaptrar finns för olika sammanhang, vilket gör datainsamlingen smidig. Exempelvis för Java-applikationer som exponerar data via Java Management Extensions (JMX), där data automatiskt kan importeras till Prometheus. Grafana tillåter systemadministratörer att skapa dashboards via Prometheus' frågespråk PromQL och därmed få grafisk översikt över dels infrastrukturens tillstånd (exempelvis hårddiskutrymme, nätverkstrafik och processoranvändning), dels nyckelvärden för applikationers prestanda (som antalet inloggade användare eller aktiva databastransaktioner).

På så sätt kan ingenjörer hålla reda på de "fyra gyllene signalerna" inom övervakning:

1. Latens
2. Trafik
3. Fel
4. Systemens mättnadsgrad

Applikationsloggar hämtas ur containrarna automatiskt och deras innehåll görs sökvänligt i Kibana via taggat metadata. Därmed kan administratörer snabbt avgöra vilken nod i Compliant Kubernetes-klustret en viss loggutskrift kom ifrån och göra root cause analysis för att felsöka effektivt. Om loggdatat konsekvent alltid följer en viss struktur, eller rent av är i ett hierarkiskt format såsom JSON, kan denna struktur göras till regelrätta fält i Elasticsearch och därmed ytterligare förenkla behandling av datat.

Continuous Integration and Deployment (CI/CD)

För att möjliggöra ett agilt arbetssätt förlitar sig många organisationer på system som låter dem automatiskt bygga, testa och driftsätta programvara i en CI/CD-process, gärna direkt vid incheckning av kod till ett versionshanteringssystem. Azure erbjuder där Azure DevOps Pipelines som helhetslösning. Andra populära alternativ är Gitlab, CircleCI, ArgoCD, Octopus Deploy, TeamCity och Jenkins, där organisationer administrerar åtminstone några av dessa själva.

Då systemen för att bygga och driftsätta programvara i en CI/CD-process i sig inte typiskt är beroende av användarens personuppgifter är det sannolikt möjligt att, även under GDPR, fortsätta använda de systemen för detta organisationen redan har. Organisationer som därför har processer och mycket kunskap inom en viss serie produkter eller tjänster kan därför tänkas vilja stanna med dessa.

Varken Safespring som sådant eller Compliant Kubernetes dikterar en viss CI/CD-lösning, utan kan göras kompatibelt med alla. Compliant Kubernetes rekommenderar av säkerhetsskäl att byggartefakterna, container images, sparas i det container image register som ingår i Compliant Kubernetes.

I egenskap av att vara en av CNCF officiellt certifierad Kubernetes-distribution är Compliant Kubernetes helt kompatibel med alla CI/CD-system som har stöd för Kubernetes.

Kontinuerlig säkerhet och regelefterlevnad via Policy as Code

Compliant Kubernetes är en Kubernetes-distribution med stort fokus på säkerhet. Att säkerställa säkerheten i system är inte en engångsföreteelse utan en kontinuerligt pågående process. Compliant Kubernetes stödjer denna process på följande sätt:

- **SÄKERHETSSCANNING** av container images efter kända fel genomförs kontinuerligt av programvaran Trivy som integrerats i container image registret Harbor.
- **INTRUSION DETECTION** via Falco, som varnar när programvaran i en container börjar bete sig på otillåtna sätt, exempelvis genom att börja försöka göra nätverkskopplingar mot system den annars inte gör eller genom att börja skriva eller läsa filer som programmerarna inte avsett.
- **BEGRÄNSNING AV NÄTVERKSTRAFIK** via brandväggsregler, uttryckta i form av Kubernetes Network Policies. Dessa implementeras och efterföljs av nätverksprogramvaran Calico.
- **AUTOMATISK CERTIFIKATHANTERING** via cert-manager, vilket gör att nätverkskrypteringscertifikat kan ges kort livslängd och roteras ofta per automatik.
- **SKYDD MOT INKORREKT KONFIGURATION** i och med Open Policy Agent, som fångar upp, inspekterar och endast släpper igenom sådana API-anrop gentemot Kubernetes API-server som uppfyller definierade policy-krav. Ett exempel här är att en policy kan förbjuda konfiguration innehållande kända standardlösenord eller att utvecklingssystem ansluter till produktionsdatabaser.

Dessa aspekter av säkerhetsprocessen är en konkretisering av organisationens policys. I och med att dessa policies konfigureras via kod som kan versionshanteras och utsättas för organisationens krav på kodgranskning uppnår organisationen enklare krav som ställs för regelefterlevnad enligt exempelvis ISO-27001.

Kontinuerlig skanning efter både kända fel och varning för beteenden som indikerar okända fel gör också att risken för dataintrång minskar. Och begränsningar i nätverkstrafik som inte applikationerna själva kan modifiera minskar risken för att eventuella intrång får stor effekt.

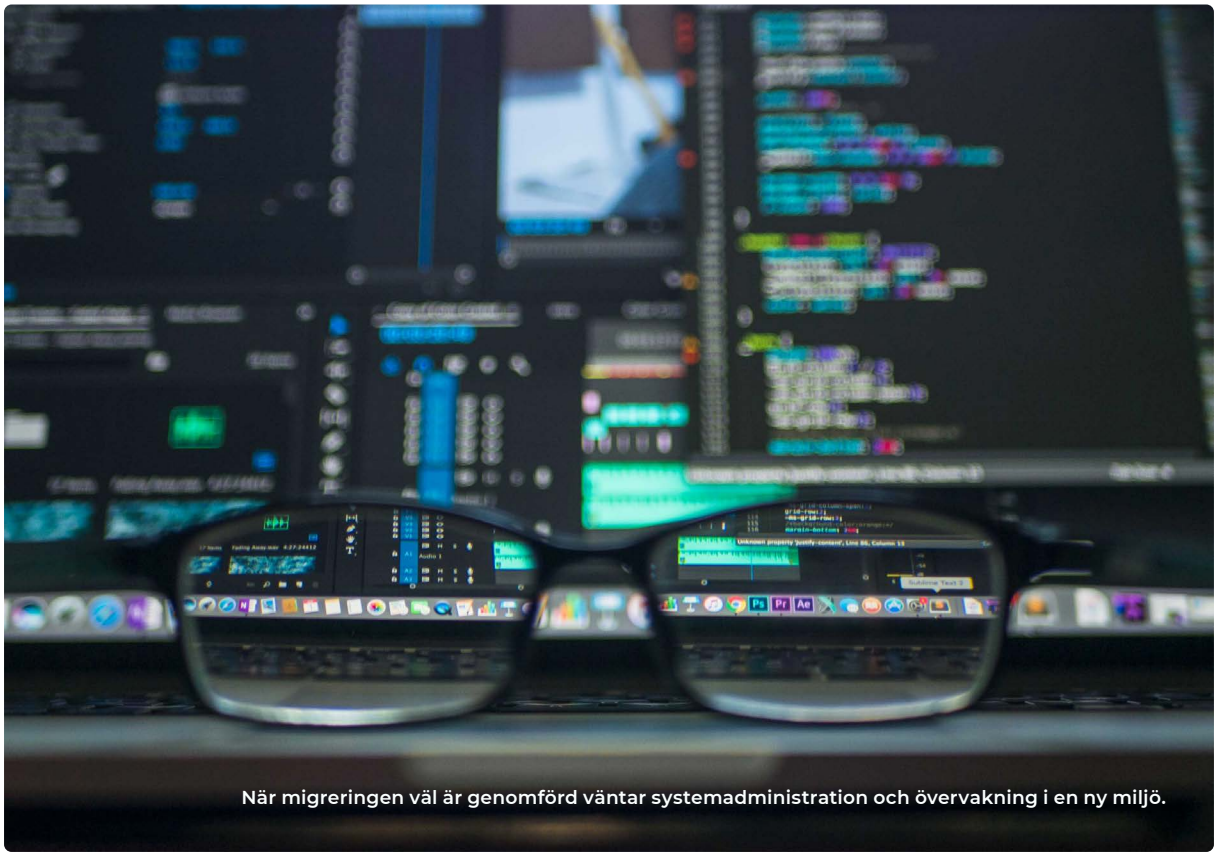
Sammanfattning

En migrationsplan innehåller en inventeringsfas, upptäckandet av beroenden, planering av arbetet och tester som säkerställer funktionalitet.

I detta dokument har de steg en organisation behöver ta sammanställts för att framgångsrikt kunna migrera från Microsoft Azure och Azure Kubernetes Service till Safespring och Compliant Kubernetes. Motiveringarna till en sådan migrering är många. Att svenskt och europeiskt lagrum gäller för GDPR-efterlevnad, tillgången på expertsupport på svenska och att datat vilar tryggt i Sverige är några av dessa. Det starka säkerhetsfokuset inom Compliant Kubernetes är ytterligare en.

En migrationsplan innehåller med nödvändighet en inventeringsfas, upptäckandet av beroenden och hur dessa kan bytas ut, planering av arbetet och tester som säkerställer funktionalitet. Därefter kan migreringen inledas och verifieras med hjälp av de kravställande testerna. Kontinuerlig dokumentation och uppföljning gör att viktiga lärdomar inte går förlorade.

När migreringen väl är genomförd väntar systemadministration och övervakning i en ny miljö. Verktygen för detta har också presenterats i dokumentet och även hur de tillsammans verkar för att ge en helhetslösning med fokus på säkerhet och smidiga agila utvecklingsprocesser.



När migreringen väl är genomförd väntar systemadministration och övervakning i en ny miljö.

Safespring är den självklara plattformen för säkra molntjänster

Besök vår webbplats för att lära dig med om molntjänster och hur Safespring kan lösa dina behov av Compute och Storage.

www.safespring.com

safespring

+46 (0)8-55 10 73 70 | info@safespring.com
Smidesvägen 12, 171 41 Solna, Sweden

www.safespring.com