

Containers - en snabbare väg ut i drift?

POPULÄRVETENSKAPLIG SAMMANFATTNING **Per-Gustaf Stenberg**

Dagens krav på ständig uppkoppling ställer allt högre förväntningar på våra IT-system. Nya lösningar och funktioner måste leveras i en högre takt och kvalitet. Är containerramverket Docker nyckeln till snabbare leverans?

Dagens samhälle är ett uppkopplat samhälle. Den ökade internetanvändningen ställer större krav på IT-system och tiden det tar att utveckla, underhålla och uppdatera dessa. Målet för de flesta IT-företag är att kunna uppdatera sina system utan att man påverkar kundens tillgång till tjänsten, detta genom så kallad “zero-downtime deployment”.

Dessa krav bidrar gemensamt till att infrastrukturen bakom IT-systemen blir allt större och mer komplicerade. Ny programkod måste levereras kontinuerligt. Detta bidrar till att problem upptäcks och kan åtgärdas snabbare. För att uppnå den kvalitetshöjande så kallade “continuous delivery”-principen, är det möjligt att använda sig av Docker-containers som kapslar in konfigurationer för att underlätta leverans och körbarhet till IT-systemet.

E-ons hundrakollprojekt är ett bra exempel på en tjänst som ställer höga krav på att konfigurationer samt mjukvara levereras felfritt och kontinuerligt. Hundrakoll hjälper E-ons kunder att hålla koll på sin elförbrukning; i sitt nuvarande tillstånd tillhandahåller tjänsten ett perfekt tillfälle att undersöka processen mot continuous delivery. I samarbete med IT-konsult-bolaget Data Ductus skapades, med hjälp utav Docker, en lösning för att kunna upprätthålla en kontinuerlig leverans av programvaruändringar.

Under arbetets gång skulle det snart visa sig att målet, continuous delivery, inte uppnås genom att bara ha tillgång till rätt verktyg (d.v.s. Docker). Docker kan dock

underlätta hur konfigurationer och tillhörande körbar kod kan tas från en miljö till en annan. Dock kvarstår frågan om hur miljöerna i sig ska kunna konfigureras per automatik, samt att leda in utvecklarna i rätt tanke-sätt. Det krävs att ny kod testas och integreras med den befintliga koden kontinuerligt.

Projektet inleddes med att de arbetsprocesser som förhindrade Data Ductus från att uppnå continuous delivery identifierades. Docker, i kombination med andra konfigurationshanteringsverktyg, användes sedan för att bryta dessa så kallade “anti-patterns”. Tillvägagångssättet för att kunna göra detta bestod av en implementeringsmetod i fem steg, i vilka containers stod i fokus.

Det första problemet att lösa var att beskriva infrastrukturen med hjälp av programmeringskod. Genom att sedan generalisera och standardisera denna programmeringskod kunde nya uppdateringsstrategier skapas för att uppnå en zero-downtime deployment. Detta gav tillräckliga förutsättningar för att gå vidare mot målet att uppnå continuous delivery.

Nästa steg var att skapa sekventiella steg eller trösklar som den nyintegrerade koden måste genomgå. Dessa steg bildar en så kallad “deployment-pipeline”. Denna deployment-pipeline ger sedan möjligheten att återkoppla information till utvecklarna rörande hur deras kod klarade av integrationsprocessen. Detta ger i sin tur goda förutsättningar att uppnå ett kontinuerligt flöde av nya leveranser av kod med hög kvalitet.