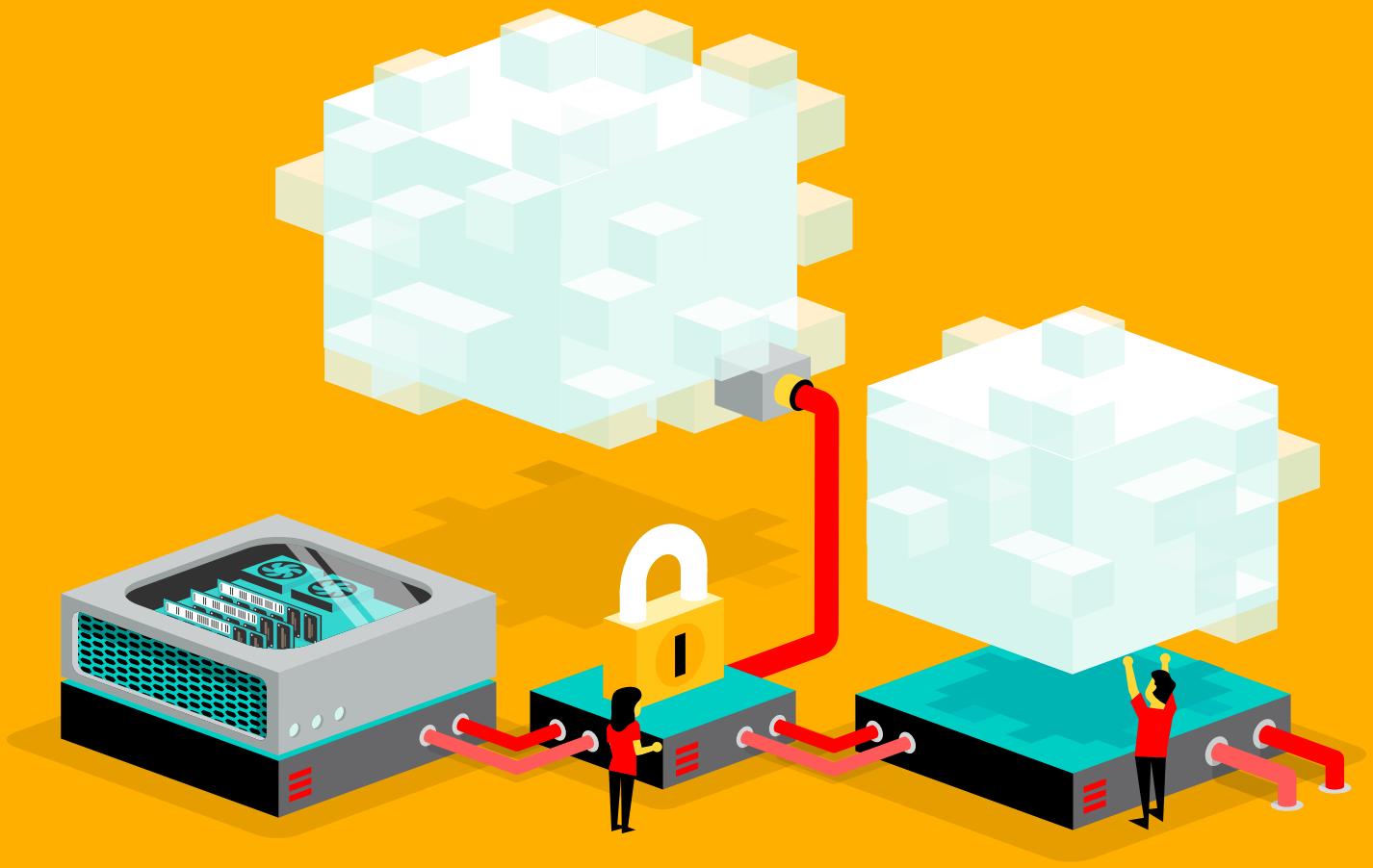
WHITEPAPER

Cloud-native security



Traditionele beveiligingsoplossingen en –maatregelen zijn in veel gevallen niet geschikt voor cloud-native technologie. Niet alleen de ontwikkeling, distributie en implementatie van applicaties vindt op een geheel andere manier plaats. Ook kunnen de microservices - waaruit een cloud-native applicatie is opgebouwd - op uiteenlopende locaties worden gehost. Het beveiligen van cloud-native technologie brengt nieuwe uitdagingen met zich mee, die om een nieuwe aanpak vraagt.

In deze whitepaper zoomen we in op cloud-native security en zetten onder meer uitdagingen, cruciale overwegingen, de belangrijkste vereisten en best practices uiteen. Ook kijken we naar een aantal voorbeelden van praktijktoepassingen.

TRUE

1 De vier fasen van de levenscyclus van een cloud-native applicatie

Wat is cloud-native security precies? De Cloud Native Computing Foundation (CNCF) stelt dat een cloud-native applicatie een viertal fasen kent: ontwikkelen, distribueren, implementeren en runtime. We zoomen in op de rol van cloud-native security in de verschillende fase.

Ontwikkelen

Kenmerkend voor cloud-native security is dat beveiliging in een vroegtijdig stadium in de levenscyclus van de applicatie wordt geïntroduceerd. Gedurende de doorontwikkeling van de applicatie gebruiken ontwikkelaars vaak geautomatiseerde tests die eventuele beveiligingsproblemen aan het licht brengen. Zo zijn beveiligingsproblemen oplosbaar voordat de applicatie naar de volgende fase van zijn levenscyclus gaat.

Distribueren

Bij cloud-native ontwikkeling is het gebruik van Continuous Integration and Continuous Development (CI/CD) populair. Hierbij werken ontwikkelaars met korte ontwikkelcycli, waarbij wijzigingen telkens zo snel mogelijk beschikbaar zijn in de productieversie van een applicatie. Deze werkwijze vraagt om maatregelen voor het bewaken van de integriteit van de workload. Denk hierbij aan het geautomatiseerde scannen van code en containerimages op kwetsbaarheden. Of aan het buiten de deur houden van onveilige ontwikkelpraktijken.

Implementeren

In de implementatiefase is ruimte voor verschillende laatste controles voordat cloud-native applicaties in de runtime-omgeving worden uitgerold. Denk hierbij aan het controleren van compliance van containerimages met vooraf gedefinieerd beveiligingsbeleid. Ook is het belangrijk om aandacht te besteden aan de hoeveelheden kwetsbaarheden die openstaan, dat enkele low risks nog zijn toegestaan en te voorkomen dat bij meerdere medium risks een uitrol wordt tegengehouden. Zo voorkom je onder meer de uitrol van containers die niet aan de eisen voldoen.

Runtime

Ook in de runtimefase zijn er verschillende aandachtspunten op het gebied van security die je in acht kunt houden. Denk hierbij aan geautomatiseerde monitoring van onder meer netwerkactiviteit, systemcalls en processen. Een ander voorbeeld is het periodiek scannen van de runtime-omgeving, onder meer voor het detecteren van onbekende of niet-geautoriseerde workloads en verdachte activiteiten.

2 Wat is cloud-native security?

Cloud-native applicaties kennen een andere structuur dan traditionele applicaties. Cloud-native vraagt dan ook om een andere aanpak op het gebied van security. Cloud-native security is een moderne en pragmatische aanpak voor het beveiligen en veilig implementeren van applicaties op schaal. De aanpak is toegespitst op een cloud-first infrastructuur.

Cloud-native security wijkt af van traditionele beveiliging. We zetten enkele belangrijke aandachtspunten bij cloud-native security op een rij:

Ontwikkelen

Een cloud-native applicatie bestaat in de meeste gevallen uit een verzameling microservices, die met behulp van API's met elkaar communiceren en samenwerken. Deze microservices functioneren ieder als een losstaande applicatie. Die biedt belangrijke voordelen, maar zorgt ook voor een snelle groei van het aantal microservices binnen een organisatie. Overzicht houden over alle microservices kan hierdoor een uitdaging zijn.

Dit inzicht is echter van cruciaal belang. Hoe stel je immers zeker dat er geen ongeautoriseerde workloads of microservices draaien indien je geen volledig zicht heb op je cloud-native omgeving? Denk echter ook aan het tegengaan van configuratiefouten, wat tot grote beveiligingsproblemen en -incidenten kan leiden. Cloud-native security vereist dan ook zeer betrouwbaar inzicht in de cloud-native omgeving, wat ook wel High Fidelity heet.

Mensen, processen en technologie

Cloud-native applicaties functioneren heel anders dan traditionele monolithische applicaties. Onder meer door het gebruik van microservices, waarin individuele functies van een applicatie zijn ondergebracht. De aanpak vraagt om nieuwe kennis, nieuwe tools en een andere aanpak.
Het budget, de tools en de beschikbaarheid van security-experts groeien in de praktijk doorgaans echter niet snel genoeg mee met de omarming van cloud-native technologie.

Organisaties kunnen het security-gat dat voortkomt uit de snel veranderende digitale ecosystemen van vandaag de dag dichten met behulp van een geïntegreerd cloud-native securityplatform. Een dergelijk platform omvat onder meer kunstmatige intelligentie (AI), automatisering, intelligence, threat detection en data-analytics.

Geïntegreerde beveiliging

Kenmerkend voor cloud-native applicaties is hun dynamische karakter. Zo zijn de applicaties continu in ontwikkeling, waarbij ontwikkelaars vaak met behulp van CI/CD-tools in korte cycli updates uitrollen. Deze updates stellen zij doorgaans geautomatiseerd beschikbaar aan eindgebruikers. Het integreren van security in het ontwikkel- en implementatieproces is dan ook cruciaal.

Voorbeelden zijn onder meer geïntegreerde scans op bekende kwetsbaarheden en beveiligingsproblemen. Denk echter ook aan security configuration management (SCM), dat ontwikkelaars helpt bij het voorkomen van configuratiefouten die tot beveiligingsproblemen leiden. Een ander voorbeeld zijn tools voor het snel terugkoppelen van feedback naar het ontwikkelteam, wat een snelle aanpak van eventuele problemen mogelijk maakt.

3 De 4C's van Cloud Native Security

Kubernetes is een bekend orkestratieplatform voor containers. Voor het beveiligen van Kubernetes-implementaties is het Cloud Native Security-model van Kubernetes hanteerbaar. Het model bestaat uit een viertal lagen: code, container, cluster en cloud. Iedere laag bouwt hierbij voort op de onderliggende laag. We zetten de verschillende lagen uiteen.

Cloud

De cloud (of een extern of on-premises data-center) vormt de basis waarop een Kubernetes cluster is uitgerold. Het is dan ook van belang dat de cloudlaag adequaat is beveiligd. Is dit niet het geval? Dan kan je onmogelijk zekerstellen dat de lagen die hierop zijn gebouwd veilig zijn.

Cluster

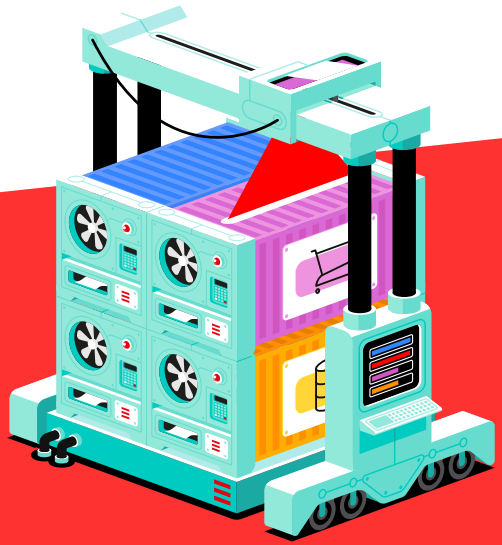
Containers draaien in een cluster, dat bestaat uit verschillende componenten. Het configureren van deze componenten is een belangrijk onderdeel van het beveiligen van de clusterlaag. Denk echter ook aan het beveiligen van de applicaties die in het cluster draaien.

Container

Microservices draaien in containers. Allerlei maatregelen helpen bij het beveiligen van containers. Denk hierbij aan het scannen van containers op bekende kwetsbaarheden, het ondertekenen van containerimages en het adequaat configureren van toegangsrechten.

Code

Iedere applicatie is opgebouwd uit programmeercode. Allerlei keuzes bij het schrijven van deze code zijn van invloed op de beveiliging van de uiteindelijke applicatie. Denk hierbij aan het afschermen van alle poorten die niet essentieel zijn voor bijvoorbeeld communicatie of het verzamelen van meetwaarden. Vergeet daarnaast niet de third-party libraries waarop de applicatie vertrouwd met regelmaat te scannen op kwetsbaarheden. Via dergelijke libraries kunnen beveiligingsproblemen in een applicatie sluipen, iets wat helaas lang niet altijd tijdig wordt opgemerkt.



4 Cloud-native beveiligingsuitdagingen

Veel uitdagingen op het gebied van security die we kennen van traditionele monolithische applicaties komen ook voor bij cloud-native applicaties. Cloud-native brengt echter ook nieuwe uitdagingen met zich meebrengt. We zetten de meest voorkomende op een rij.

Onveilige standaardconfiguratie

Veel cloud-native systemen en bijvoorbeeld containers zijn standaard niet adequaat beveiligd. Hoewel het toepassen van een veilige configuratie mogelijk is, bevat de standaardconfiguratie vaak instellingen die beveiligingsrisico's met zich meebrengen. Je loopt dan ook het risico dat iemand onbewust een container- of serverloze functie implementeert, zonder het hanteren van de juiste configuratie. Constante monitoring van configuratieopties van alle applicaties in iedere implementatieomgeving is dan ook een cruciale stap die configuratiefouten helpt voorkomen.

Dynamische applicatietopologieën

Een cloud-native applicatie is zeer dynamisch. Zo kan een applicatie eenvoudig opschalen met meer containers indien meer capaciteit nodig is. Neemt de capaciteitsvraag later weer af? Dan kan de applicatie net zo eenvoudig weer afschalen. Dit dynamische karakter vraagt om beveiligingsmaatregelen die hierop zijn toegespitst. Zo moet beveiliging meegroeien met bijvoorbeeld de creatie van nieuwe clusters. Maatregelen op dit vlak zijn divers. Denk aan het versleutelen van netwerkverkeer, maar ook aan het ondertekenen van containerimages zodat alleen veilige images kunnen worden uitgerold.

Groter aanvalsoppervlak

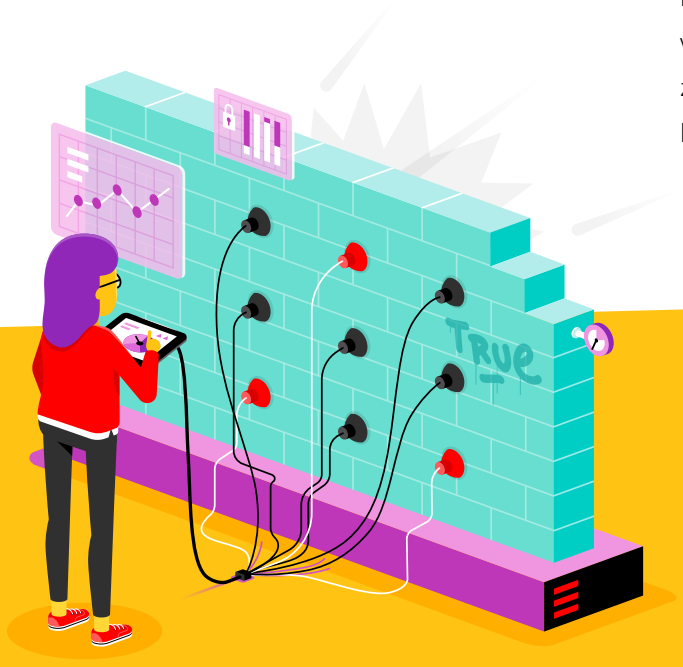
In veel traditionele IT-omgevingen wordt gebruik gemaakt van zogeheten perimeter beveiliging. Het netwerk wordt hierbij afgeschermd en alleen vertrouwde apparaten toegestaan. Deze aanpak is voor cloud-native technologie minder geschikt. Met name doordat cloud-native applicaties al snel bestaan uit honderden of zelfs duizenden individuele componenten. Het hosten van deze elementen kan op diverse locaties, die zich vaak buiten het eigen netwerk bevinden. Ook kunnen al deze componenten kwetsbaarheden bevatten die aanvallers vrij spel geven en soms zelfs als springplank dienen naar de rest van de applicatie. Het aanvalsoppervlak is bij cloud-native technologie dan ook aanzienlijk groter.

5 Wat zijn de beveiligings-overwegingen voor Cloud Native?

Veel bedrijven experimenteren met cloud-native technologie. Het beheren en beveiligen van cloud-native omgevingen krijgt vaak echter minder aandacht. We zetten de implicaties op het gebied van security in een cloud-native omgeving uiteen. En kijken naar de impact hiervan op uw beveiligingsstrategie.

Serverworkloads adequaat beschermen

Traditioneel is IT-beveiliging gericht op het beschermen van de perimeter, waarbij het netwerk wordt afgeschermd. Alleen veilige endpoints zijn toegelaten. Ook is het netwerk vaak gesegmenteerd, zodat eventuele indringers geen vrij spel krijgen. In een cloud-native omgeving is deze beveiligingsaanpak echter minder geschikt. Want hoewel microservices samen een applicatie vormen, zijn kunnen op diverse locaties worden gehost. Cloud-native technologie vraagt daarom om het adequaat beschermen van datacenter- en serverworkloads.



Continue levering vraagt om continue beveiliging

Kenmerkend voor cloud-native applicaties is dat de levering en implementatie van software-updates continu wordt. In de praktijk betekent dit dat ontwikkelaars software-updates in kleine stukjes uitvoeren. Zij implementeren nieuwe code vaak direct in de productieomgeving, zodat gebruikers hiervan zo snel mogelijk gebruik van kunnen maken. Het aantal implementaties kan – afhankelijk van de organisatie – dan ook snel oplopen. Dit brengt nieuwe eisen met zich mee op het gebied van beveiliging. Zo moet security ingebed zijn in toolkits voor het implementeren van nieuwe microservices en containers. En zijn continue beveiligingscontroles van cruciaal belang.

Hybride beveiliging voor de stack

Microservices zijn niet alleen op diverse locaties te hosten, maar kunnen ook op verschillende soorten systemen. Zo draaien sommige microservices op een virtuele server, terwijl andere zijn ondergebracht op een bare-metal Linux-server. Ook dit heeft impact op de beveiligingsvereisten. Zo vraagt de werkwijze om veel inzicht in de servers waarop microservices draaien. Het is bijvoorbeeld van belang dat een container altijd is gehost op een virtuele machine die volledig up-to-date is. Dit kan je echter alleen zekerstellen indien je voldoende zicht hebt op deze virtuele machine. Cloud-native vraagt dan ook om de integratie van inzicht, monitoring en bescherming.

Schaalbare runtime-detectie

Een cloud-native omgeving is zeer dynamisch, onder meer doordat implementaties en upgrades in veel gevallen continu worden uitgerold. End-to-end inzicht en monitoring is hierdoor aanzienlijk complexer dan in een traditionele omgeving. Het is dan ook van belang dat ook het detecteren van aanvallen op dynamische wijze plaatsvindt. Zo moeten detectietools in realtime kunnen schalen en meegroeien met de cloud-native omgeving. Belangrijk hierbij is dat dit de prestaties en stabiliteit van de productieomgeving niet beïnvloedt.

6 Hoe ziet het ideale cloud-native beveiligingsplatform eruit?

Voor het beveiligen van cloud-native applicatie zijn allerlei oplossingen en platforms beschikbaar. Hoe ziet het ideale cloud-native beveiligingsplatform eruit? Dit zijn de belangrijkste vereisten.

Chaos-engineering omarmen

Een cloud-native architectuur bestaat uit containers, waarin microservices zijn ondergebracht. Deze microservices zijn in de runtime met behulp van API's aan elkaar gebonden. Iedere functie van een applicatie kan bestaan uit een reeks van deze microservices. Door de grote hoeveelheid microservices en containers waaruit cloud-native technologie bestaat zijn er veel variaties, die ieder tot andere beveiligingsproblemen en onverwachts situaties kunnen leiden. Chaos engineering biedt uitkomst. Bij deze werkwijze is het uitgangspunt dat beveiligingsproblemen niet voorkomen kunnen worden. Chaos engineering-tools voeren experimenten uit in de productieomgeving en brengen zo problemen aan het licht. Nog voordat zij in de praktijk tot verstoringen of beveiligingsincidenten leiden. LitmusChaos is een voorbeeld van een chaos engineering-tool voor cloud-native technologie.

Hoge mate van automatisering

Een traditionele IT-omgeving is vaak handmatig gemonitord, waarbij eventuele problemen handmatig worden gedetecteerd en opgepakt. Een cloud-native applicatie kan echter bestaan uit een groot aantal microservices, die ieder tot nieuwe problemen kunnen leiden. Handmatige workflows zijn dan ook simpelweg geen optie; het aantal gedetecteerde problemen die om een oplossing vragen is simpelweg te groot om handmatige weg te werken. Geautomatiseerde detectie en respons op schaal is dan ook onmisbaar.

Inzicht in hybride stack

Een cloud-native omgeving bestaat uit een groot aantal componenten. Dit maakt het verkrijgen van inzicht een uitdaging. Zo heb je niet alleen inzicht nodig in de host en eventuele virtuele machines die hierop draaien, maar ook in containers, microservices en API's. Adequaat inzicht in de volledige stack is van groot belang. Zo moet alle vergaarde informatie op verschillende lagen van de infrastructuur centraal zijn verzameld, wat adequaat handelen mogelijk maakt.

Schaalbare monitoring met minimale impact

Door de vele componenten waaruit een cloud-native applicatie bestaat en de API's voor het aan elkaar knopen van microservices zijn er veel elementen die gemonitord en beheerd moeten worden. Het is niet alleen van belang dat security monitoring voldoende kan meeschalen met de ontwikkeling van de cloud-native applicatie. Ook is het essentieel dat de impact van geautomatiseerde monitoring op de prestaties van een applicatie is geminimaliseerd.

Integratie met orkestratie- en automatiseringstools

In een cloud-native omgeving zijn diverse tools inzetbaar voor het orkestreren van containers-workloads, waarvan Kubernetes een voorbeeld is. Voor het automatiseren van implementaties zijn Ansible, Chef en Puppet veelgebruikte tools. Indien security tools met deze componenten zijn geïntegreerd is security direct uitrolbaar met de te beschermen workload. Zo weet je zeker dat iedere workload adequaat is beveiligd.

7 Cloud-Native Security: Waar begin je?

Als je voor het eerst aan de slag gaat met cloud-native security komt er veel op je af. Zo zijn er allerlei aandachtspunten en afwegingen die je aandacht vragen. We bekijken enkele belangrijke overwegingen.

DevSecOps

Voorkomen is beter dan genezen. Dat geldt ook voor beveiligingsproblemen. DevSecOps helpt hierbij. Deze aanpak integreert beveiliging in ieder element van cloud-native technologie, variërend van de code en containers tot de onderliggende infrastructuur. Dit is mogelijk doordat ontwikkelaars, operations en security-experts gedurende het ontwikkelproces nauw met elkaar samenwerken.

Beveiligingsproblemen met API's

Beveiligingsproblemen met API's komen helaas relatief vaak voor. API's verzorgen de communicatie tussen microservices. Gebruik daarom altijd https voor alle communicatie via API's, zodat kwaadwillenden niet kunnen meelezen. Versleutel daarnaast ook gegevens, zodat data ook als deze uitlekken ontoegankelijk blijven. Kies bij voorkeur voor stateless API's. Deze maken gebruik van realtime authentificatie en autorisatie, in plaats van cookies of sessies.

Verantwoordelijkheden delen en samenwerken

De cloud kent een model van gedeelde verantwoordelijkheden. Zo is de cloudprovider bijvoorbeeld verantwoordelijk voor de infrastructuur waarop de cloudomgeving draait. Als klant ben jij zelf verantwoordelijk voor de applicaties en workloads die je in deze omgeving uitrolt. Doordat de cloudprovider de verantwoordelijkheid voor bepaalde componenten van de omgeving op zich neemt, heb je hierover geen controle en hierop minder zicht. Veel cloudproviders maken daarom tools beschikbaar die je inzicht geven in deze componenten. Omarm deze tools. Zo kan je onder meer potentiële beveiligingsincidenten tijdig opmerken en hierop adequaat reageren.

Use cases

Benieuwd hoe andere partijen cloud-native technologie in de praktijk inzetten en welke dit oplevert? We zetten een aantal cases op een rij die een goed beeld geven van de uitdagingen, aandachtspunten en winst bij het omarmen van cloud-native.

Onvoldoende kennis

Een securitymanager bij een wereldwijd actief dierengezondheidsbedrijf moest in korte tijd een cloudinfrastructuur opbouwen in zowel Azure als Amazon Web Services (AWS). In eerste instantie werd een security systems administrator als cloud lead aangewezen. Al snel bleek de systeembeheerder echter over onvoldoende kennis te beschikken. Terugkijkend op het project stelt de securitymanager dat hij de kennis en tijd die nodig is voor het adequaat beveiligen van de cloud onderschatte. Bestaande processen en security tools leken bruikbaar, maar bleken dat in de praktijk niet te zijn.

→ Lees verder bij The New Stack

Samenwerking buiten security

De cloud security director bij een wereldwijde zakelijke dienstverlener stond voor de uitdaging tot het opzetten van een cloudsecurityteam. Ook moest zij een wereldwijde standaard voor cloud security opstellen en samenwerken met IT voor de implementatie van de juiste securitymaatregelen. Niet alleen moest de manager een manier vinden om al deze taken uit te voeren, maar ook een volledig nieuw securityteam opzetten. Samenwerking met belanghebbenden buiten security was de sleutel tot succes. Door overleg met diverse belanghebbenden op het gebied van onder meer IT, risicomanagement, compliance en financiën bouwde zij niet alleen relaties op. Ook creëerde zij een groep met een gedeeld doel: het versnellen van de business, beheersen van risico's, beheren van cloudkosten en compliance met regelgeving. .

→ Lees verder bij The New Stack

Ericsson ondersteunt behoeften 5G-netwerken met Kubernetes

Naarmate de telecomindustrie 5G omarmt ontstaan nieuwe businessmodellen en use cases. Deze use cases vragen om onder meer om betrouwbare netwerken, lage latentie, een hoge bandbreedte, gedistribueerde cloud en kunstmatige intelligentie. Om services efficiënt te kunnen leveren zetten providers in op automatiseren, een lagere total cost of ownership, snellere time-to-market, hoge betrouwbaarheid en hoog veiligheidsniveau. Ericsson ondersteunt 5G-netwerken en zowel de nieuwe businessmodellen als use cases die hierdoor ontstaan met behulp van cloud-native en open source technologie. Het gaat daarbij onder meer om Kubernetes.

→ Lees verder bij de Cloud Native Computing Foundation

Verizon Media kiest cloud-native voor beheer van groeiend aantal mediamerken

Met een groeiend portfolio aan mediamerken werkte Verizon Media met een groot aantal stacks. Zij werden beheerd door diverse teams, die daarvoor ieder hun eigen tools en platforms gebruikten. Geen van allen waren containerized. Ook maakte Verizon Media gebruik van een zelf ontwikkelde on-premises oplossing voor het pushen en uitrollen van code, die veel tijd in beslag nam. Het team is overgestapt op een cloud-native technologie, containers en het orkestratieplatform Kubernetes, wat veel winst oplevert.

→ Lees verder bij de Cloud Native Computing Foundation

Amerikaans ministerie van Defensie versnelt uitrol van software met Kubernetes

Het ontwikkelen en uitrollen van software voor grote wapensystemen kon voorheen bij het Amerikaanse ministerie van Defensie al snel drie tot tien jaar in beslag nemen. Zo maakte teams vooral gebruik van de watervalmethode, en werkte niet met een minimum viable product en incrementele updates. Met het oog op onder meer kunstmatige intelligentie, machine learning en cybersecurity was het versnellen van dit proces voor het ministerie noodzakelijk. Daarom is het DoD Enterprise DevSecOps referentiemodel opgezet. Het model schrijft onder meer het gebruik van CNCF-compliant Kubernetes clusters en andere open source technologie voor binnen het ministerie.

→ Lees verder bij de Cloud Native Computing Foundation