JDriven Tech Radar najaar 2021

Onze kijk op recente ontwikkelingen in het veld

Commit to know.

Develop to grow.

Share to show.

jdriven

Introductie

Onze ervaren specialisten werken dagelijks mee aan tal van softwareprojecten in Nederland en zijn betrokken in wereldwijde communities. Halfjaarlijks komen wij vanuit JDriven bij elkaar om te bespreken wat wij aan nieuwe trends en ontwikkelingen zien. Deze trends proberen wij te vangen in een technologieradar. Elke editie van de radar laat verschuivingen zien t.o.v. een vorige editie. Een verschuiving kan betekenen dat wij een technologie interessanter zien worden waar van toepassing, of juist minder geschikt ongeacht de toepassing. Indien een trend niet meer voorkomt in een latere editie, dan zijn er geen nieuwe ontwikkelingen en/of ervaringen die ons eerdere beeld zouden hebben bijgesteld. In dit document willen we toelichten welke verschuivingen we de afgelopen periode hebben waargenomen, om op basis daarvan weer richting te geven aan wat wij inzetten en aanraden.

Tech Radar

Het idee voor het opstellen van een tech radar komt voort vanuit Thoughtworks. Zij dragen al langer periodiek met een radar hun visie uit op nieuwe trends en ontwikkelingen. Tevens raden zij iedereen aan een eigen radar op te stellen.

Bij JDriven onderschrijven we dat. Het opstellen van een Radar is een leerzame en waardevolle ervaring waarin onderling kennis wordt gedeeld en een technisch bewustzijn wordt gecreëerd. Wij geloven erin dat specialisten zelf in staat moeten zijn om het gereedschap voor hun werkzaamheden samen te stellen. Met het opstellen van een radar faciliteer je discussies over technologie, om als organisatie de juiste balans te vinden in wat voor risico's en voordelen innovatie kan opleveren. Wij kunnen u helpen dit op te starten binnen uw organisatie. Laat uw teams elkaar inspireren tot innovatie en gezamenlijk komen tot een set aan technologieën en technieken die de ontwikkeling in uw bedrijf versnellen.

Indeling

Een radar bestaat uit kwadranten en ringen, met daarbinnen blips om interessante technologieën en technieken aan te duiden.

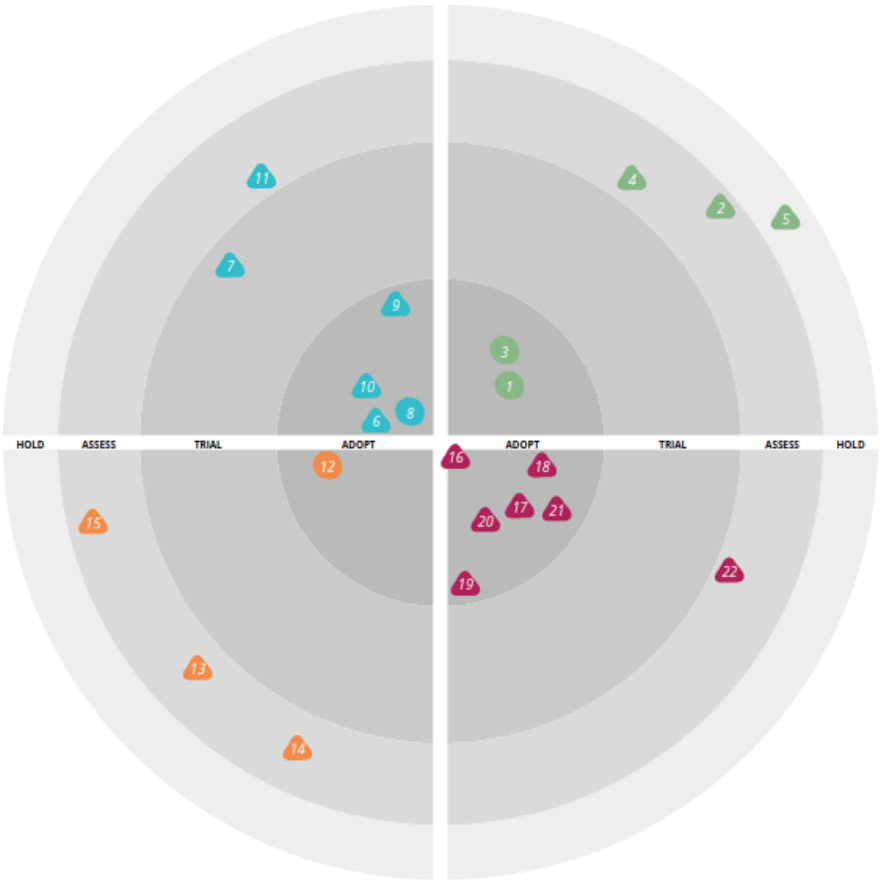
De kwadranten verdelen de verschillende onderwerpen in categorieën.

- **Talen & frameworks** die je ondersteunen bij de ontwikkeling van software
- **Platformen** waarop je software kunt uitvoeren
- **Technieken** die je helpen betere software te maken
- **Tools** ter ondersteuning van je ontwikkel- en delivery-proces

De ringen in elk kwadrant geven aan in welk stadium van adoptie wij denken dat die technologie zich bevindt.

- **Adopt** → Wij raden sterk aan deze technologie te gebruiken, waar van toepassing.
- **Trial** → Interessant om alvast ervaring mee op te doen (in een project dat het risico kan dragen).
- **Assess** → Goed om beter te begrijpen en toekomstige impact in te schatten, maar nog niet om toe te passen.
- **Hold** → Niet (meer) gebruiken.

In de volgende secties zullen we onze kijk op de recente ontwikkelingen per kwadrant toelichten.



Tools	Talen & frameworks
ADOPT ArgoCD Kafka-Connect OpenRewrite Sealed Secrets **TRIAL** Sentry **ASSESS** OpenTelemetry **HOLD** -	**ADOPT** Quarkus ZIO **TRIAL** - **ASSESS** jMolecules WASM serverless **HOLD** Java Persistence API
Platforms	**Technieken**
ADOPT InfluxDB **TRIAL** - **ASSESS** Akka Serverless TimescaleDB Yugabyte **HOLD** -	**ADOPT** Container scanning Defense in Depth Engineering effectiveness Server-side rendering Shift left security Software Composition Analysis **TRIAL** - **ASSESS** Quantum Technology **HOLD** -

Commit. Develop. Share.

Talen & frameworks

ADOPT

1. Quarkus
3. ZIO

ASSESS

2. WASM serverless
4. jMolecules

HOLD

5. Java Persistence API

Quarkus

ADOPT Quarkus is een fijnwerkend framework voor developers die CDI en MicroProfile gewend zijn.

Quarkus is ontwikkeld om het leven voor developers gemakkelijk te maken. Er is maar weinig configuratie nodig om ermee te kunnen werken. Het framework biedt ondersteuning om eenvoudig in development-, test- of productiemodus te draaien. Quarkus gaat efficiënt om met geheugen, is met behulp van extensies eenvoudig uit te breiden, en is een volwassen product geworden. Het werkt native met Kubernetes of OpenShift en start redelijk snel op.

Het gebruik van Smallrye Mutiny als Reactive Streams library maakt het mogelijk om imperatieve stijl naast reactive stijl te laten bestaan. Dit stelt developers in staat om laagdrempelig en stapsgewijs een applicatie volledig non-blocking te maken. Het is zeer geschikt voor HTTP RESTful web development en integreert goed met gangbare message protocols, authentication protocols, datastores en platforms. Dankzij de rijkelijk aangeklede handleidingen is het bovendien relatief eenvouding om Quarkus naar behoefte uit te breiden.

Sinds de vorige JDriven Tech Radar is het maken van native executables al lang niet meer een onderscheidende factor voor Quarkus, maar door de brede ondersteuning heeft Quarkus ons bewezen een serieuze speler te zijn in het aanbod van JVM CDI Frameworks. In het afgelopen jaar werd Quarkus bij diverse klanten succesvol ingezet (o.a. financiële sector). Hiermee heeft het zich als stabiel framework bewezen, waarmee Quarkus in deze editie de ADOPT status krijgt.

ZIO

ADOPT ZIO is een open-source library dat zich focust op puur functioneel, asynchroon programmeren, in Scala. De kern van ZIO is de gelijknamige IO-monad, die, voorzien van een rijke API, puur functioneel programmeren gebruiksvriendelijk wil maken voor elke Scala-ontwikkelaar. Waar de meest bekende libraries voor functioneel programmeren sterk gebaseerd zijn op het zware jargon van het vakgebied, wijkt ZIO hier bewust van af, ten behoeve van de leesbaarheid en gebruiksgemak. Rond de ZIO kernlibrary ontwikkelt zich een rijk ecosysteem van dochterlibraries, allen gebaseerd op het ZIO type, zoals zio-streams, zio-test, zio-nio, zio-kafka, en vele andere. De explosieve groei van dit nieuwe Scala ecosysteem wordt inmiddels gretig afgenomen binnen de community, en de ontwikkelaars onder het bedrijf Ziverge doen stevig aan promotie en ondersteuning via verschillende kanalen.

ZIO biedt een sterke basis voor een puur functionele codebase, wanneer de taalkeuze is gevallen op Scala. De eerste productieversie van de library is gereleased in augustus 2020, en de tweede grote release heeft inmiddels enkele milestones. ZIO draait al enige tijd stabiel in productie bij een van onze klanten en wordt goed onderhouden door de community. Bugs worden snel aangepakt en op online communicatie in hun discord channel wordt vrijwel direct gereageerd. Dit maakt ZIO adoptiewaardig.

jMolecules

ASSESS jMolecules is een set libraries met twee duidelijke doelen. Ten eerste helpt het bij het uitdrukken van DDD-concepten. Wanneer je nadrukkelijk de Ubiquitous Language voor class names gebruikt, wil je deze niet vervuilen met de benaming van de DDD-bouwblokken. jMolecules heeft annotaties om toch te kunnen aangeven welk type bouwblok classes zijn. Als alternatief heeft jMolecules ook Java interfaces gebaseerd op John Sullivan's series "Advancing Enterprise DDD". Hiermee kun je relaties tussen DDD-concepten in een domeinmodel middel het type system van Java vastleggen. Ten tweede kan je met jMolecules expliciet markeren hoe een stuk code (package, class) een concept in softwarearchitectuur vervult. Denk hierbij aan het kunnen aangeven waar een component hoort in een layered- of onion-architecture.

Door het vastleggen van concepten uit DDD of softwarearchitectuur in de code, biedt dit ontwikkelaars naast conventies voor softwaremodellering extra duidelijkheid over de intentie van een stuk code. Omdat concepten in de code vastliggen, is het mogelijk om met geautomatiseerde tooling (ArchUnit, jqassistant) te verifiëren en documenteren of een implementatie voldoet aan de regels en concepten van een gekozen architectuur.

WASM serverless

ASSESS WebAssembly is een nieuwe taal die performance, security en portability biedt. Er zijn meerdere toepassingen voor Wasm, een daarvan is Wasm voor serverless, omdat het zo snel is als native, geen cold start problemen heeft en de developer een brede taalkeuze geeft. Zo kun je bijvoorbeeld Kotlin gebruiken om te compileren naar Wasm.

Momenteel bieden de grote cloud providers nog geen support voor Wasm, maar een Wasm VM (bijv. WasmEdge) kan gemakkelijk naast Node.js draaien in een lambda zonder performanceverlies. Daarmee win je dus wel vrijheid in taalkeuze, betere security en, als Wasm wel direct ondersteund wordt, ook performance winst.

Wasm is nog volop in ontwikkeling, wat het mist aan volwassenheid wordt gecompenseerd door potentie. Vooral de potentie om Kotlin te kunnen schrijven voor serverless functions, met daarbij een performance- en securitywinst, maakt de techniek waardevol om verder te onderzoeken.

Java Persistence API

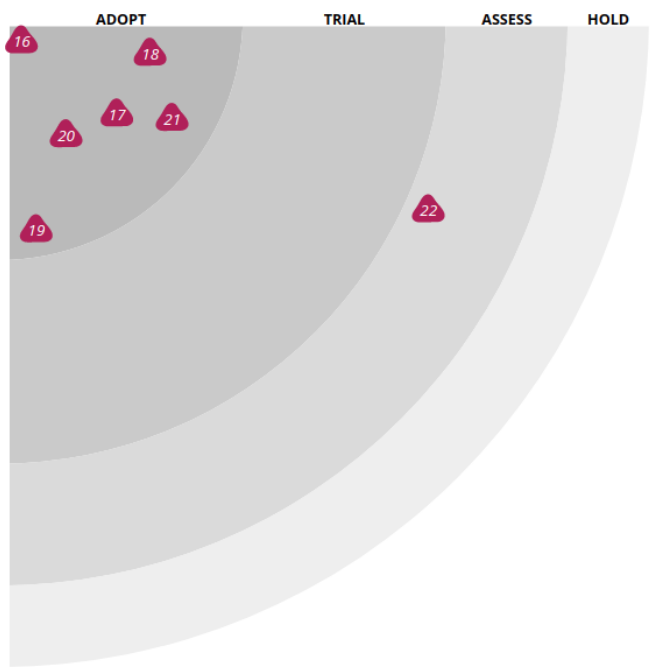
HOLD Hibernate was de eerste populaire ORM voor Java. JPA is een abstractie van de Hibernate API en opgenomen als Java-standaard met inmiddels meerdere implementaties. Het hoofddoel van JPA is het mappen van databasetabellen op Java-objecten en het automatisch uitvoeren van de nodige queries en joins.

Een nadeel van een ORM is dat het correct gebruik ervan de nodige kennis vereist. Voor nieuwe gebruikers is niet altijd vanzelfsprekend wanneer een object opgeslagen wordt, en in welke volgorde inserts, updates en deletes plaatsvinden. JPA maakt zelf de queries, die niet altijd even efficiënt zijn of niet precies doen wat de ontwikkelaar wil. Het fine-tunen hiervan is bovendien niet triviaal. Daarnaast is SQL een krachtige DSL om data op te halen en enkele opties die SQL biedt zijn niet eenvoudig te gebruiken in JPA. Bijvoorbeeld het maken van rapportages is typisch iets waarbij queries/joins vaak meerdere tabellen raken, en slechts enkele velden per tabel nodig hebben. Deze queries zijn duur in JPA. Een ander voorbeeld is bijvoorbeeld een workload verdelen over een cluster van instances die synchroniseren via de database middels "SELECT FOR UPDATE SKIP LOCKED".

Als alternatief zijn er een aantal libraries die SQL niet proberen te verbergen, maar juist een API aanbieden die heel nauw verwant is aan SQL. Het resultaat van die queries kan dan gemapped worden naar een POJO. De volgorde van queries en effecten in de dabase is ook meteen helder. Dit stelt developers in staat om alle krachtige queries van de onderliggende database te kunnen gebruiken. Databaseontwikkelaars en data scientists begrijpen ook welke queries er precies uitgevoerd worden, of hoe iets geoptimaliseerd kan worden omdat iedereen SQL begrijpt.

Een aantal voorbeelden van frameworks zijn JOOQ, Kotlin Exposed en JDBI.

Technieken



ADOPT

16. Container scanning
17. Defense in Depth
18. Engineering effectiveness
19. Serverside rendering
20. Shift left security
21. Software composition Analysis

ASSESS

22. Quantum Technology

Container scanning

ADOPT Tegenwoordig draaien een hoop applicaties in een container in bijvoorbeeld Docker of Kubernetes. Een container draait echter meer dan alleen jouw applicatie, er zit ook software in van de base image die je gebruikt. Deze software kan echter kwetsbaarheden bevatten. Sterker nog, uit een onderzoek vorig jaar bleek dat meer dan de helft van de (toen) 4 miljoen Docker Hub images minstens een kritische kwetsbaarheid had welke ook nog eens misbruikt kan worden.

Reden genoeg om een Container Scanning tool te gaan gebruiken, bij voorkeur direct in je CI/CD pipeline. Een aantal veelgebruikte tools hiervoor zijn Snyk en Trivy (open source).

Defense in Depth

ADOPT Voor een lange tijd werden verdedigingen binnen IT netwerken alleen toegepast bij de poort (firewall) tussen het internet en het interne bedrijfsnetwerk. Tegenwoordig is dit niet meer voldoende, omdat er onder andere ook dreigingen van binnenuit kunnen komen. Daarnaast hoeft er maar één fout gemaakt te worden aan de poort en een aanvaller is binnen. Hieruit is een praktijk gevloeid genaamd "Defense in Depth", waarbij men meerdere lagen aan verdedigingen toepast. Een mooi voorbeeld is hoe de kastelen in de middeleeuwen gebouwd zijn. Naast dat er een hoge muur was, had men ook vaak een gracht om de muren heen, met een ophaalbrug. Ook binnen de muur waren er meerdere verdedigingen beschikbaar.

Wij zijn van mening dat de tijd waar men weg kon komen met slechts één verdedigingslaag ver achter ons ligt, en dat Defense in Depth standaard overal toegepast moet worden wil je bestand zijn tegen cyberaanvallen.

Engineering effectiveness

ADOPT Grote techbedrijven verkondigen al geruime tijd de waarde van Engineering Effectiveness teams voor de gehele organisatie. Naast de focus op het eindproduct zelf wordt bij deze techniek ook aandacht besteed aan hoe dat product tot stand komt. Teams die bezig zijn met het eindproduct worden ondersteund met diepe technische expertise. Hoe groter de organisatie, hoe meer winst er valt te behalen.

Productteams blijven hierbij verantwoordelijk voor het zelf bouwen, onderhouden, draaien en monitoren van hun applicaties, maar worden daarbij ondersteund door een team technische experts dat zorg draagt voor teamoverstijgende zaken. Dit team denkt en helpt vanuit hun expertise actief mee op vlakken als software design, kwaliteit, security scanning, release engineering, infrastructuur, cloudplatformen en reliability engineering. Denk daarbij aan applicatie lifecycle managementzaken als het over de brede lijn realiseren van tijdswinst voor ontwikkelaars, gewenste praktijken uitdragen, geautomatiseerd wegnemen van technical debt, inzetten op standaardisatie en interne tools bouwen en onderhouden.

De grootte van een Engineering effectiveness team zal afhangen van de organisatie en daarbinnen de verscheidenheid aan technologieën. Het team kan het beste gevormd worden door een vaste groep collega's aangevuld met een wisselende groep collega's die vanuit de productteams komt, om de aansluiting met deze teams te optimaliseren.

Ook wij zien dat er bij onze klanten steeds meer winst valt te behalen door in te zetten op standaardisatie, aangevuld met een Engineering effectiveness team om de productiviteit van de gehele organisatie te verhogen. Dit voorkomt dat teams afzonderlijk van elkaar dezelfde problemen verschillend invullen, met technical debt en verminderde productiviteit tot gevolg.

Server-side rendering

ADOPT Voordat single page application (SPA) libraries zoals Angular en React in zwang raakten, was het gebruikelijk om de serverapplicatie de HTML te laten genereren als een bezoeker op een URL van een website kwam. JavaScript werd gebruikt om de complexere interacties te verbeteren, maar ook voor performance, daar roundtrips naar de server konden worden geminimaliseerd. Dit leidde tot een ecosysteem van SPA-frameworks en de trend dat bij de bouw van websites bij voorbaat voor client-side rendering (CSR) wordt gekozen. Dit is ten koste gegaan van SEO, initial page load en progressive enhancement, maar heeft ook geleid tot een grotere afstand tussen backend- en frontend-specialisatie. Er is een groeiend besef dat er situaties zijn, waar de voordelen niet opwegen tegen deze nadelen.

Zodoende is SSR, waarbij de server de HTML genereert en client-side scripts beperkt blijven voor functionele optimalisatie, weer aan een opmars bezig. Onder noemers als Universal of Isomorphic hebben SPA-frameworks hier hun eigen initiatieven voor, opdat je op client en server dezelfde taal en hetzelfde framework kunt gebruiken. Ook zijn er initiatieven zoals de HTMX en Hotwire JavaScript-frameworks, die uitgaan van server-side rendered HTML - via bijv. Java, Spring en Thymeleaf - en ook voor verdere interactie tussen client en server HTML uitwisselen i.p.v. JSON.

Deze renaissance betekent niet het einde van CSR ten gunste van SSR, maar benadrukt dat de client-server interactie voor het bouwen van een website een bewuste keuze moet zijn.

Shift left security

ADOPT Shift left security is het opnemen van security op het vroegst mogelijke moment in de software development lifecycle (SDLC) en is een van de beste manieren om de cyberweerbaarheid van een product of organisatie te vergroten. Bij de traditionele fases van een SDLC van requirements-analyse, design, ontwikkeling, testing en deployment, was security vaak onderdeel van de laatste stap. Vlak voor deployment toegepast en gericht op de buitenkant van een applicatie.

Uit onderzoek door IBM is gebleken dan het aanpakken van security issues tijdens de designfase zes maal goedkoper is dan tijdens de implementatie; tijdens de testfase is dit zelfs 15 maal duurder. Wanneer het product klaar staat om opgeleverd te worden en de kosten van het aanpakken van security issues niet opwegen tegen de cost-of-delay van een uitgestelde release, wordt risico-acceptatie overwogen. Hierbij worden ook zaken als reputatieschade en de kosten van een mogelijk datalek meegenomen.

Shift left security is niet alleen goed voor het reduceren van cyberrisico's, maar ook de kosten die dit met zich meebrengt.

Software Composition Analysis

ADOPT Open-source code wordt overal gebruikt ter ondersteuning bij de implementatie van applicaties. Zo vaak zelfs, dat veel developers zich nauwelijks bewust zijn van alle open-source componenten in hun software. Uit onderzoek is gebleken dat in de afgelopen vijf jaar het gebruik ervan met 259% is toegenomen tot een gemiddelde van 529 open-source componenten per applicatie. Helaas betekent deze groei ook dat het aantal kwetsbaarheden is toegenomen. Het is niet voor niets dat voor het gebruik van kwetsbare en verouderde componenten in de OWASP Top 10 een aparte categorie is toegewezen.

Software Composition Analysis (SCA) is een onderdeel van de application security testing (AST) tool-markt waar specifiek het gebruik van open-source componenten wordt gecontroleerd. SCA-tools voeren een automatische scan uit op de codebase van een applicatie om de gebruikte componenten, hun licentie-compliance en security-kwetsbaarheden in kaart te brengen.

Het detecteren van kwetsbaarheden alleen helpt organisaties niet om het risico te beperken. Hiervoor moet een volwassen SCA security model worden toegepast, waarbij naast detectie ook prioriteren en verhelpen van issues zijn meegenomen.

Quantum Technology

ASSESS Quantum technology houdt zich bezig met het natuurkundige fenomeen, dat sommige deeltjes zich in een zogenaamde superpositie kunnen bevinden. In een dergelijk geval is hun waarde niet alleen uit te drukken in een digitale 0 of 1, maar in waarschijnlijkheden dat ze die waardes hebben. Pas wanneer dit gemeten wordt, nemen ze een waarde aan. Computerbits die dit gedrag vertonen, noemen we quantum bits, of kortweg qubits. Bedrijven als QuTech houden zich bezig met het bouwen van de hardware met en rondom qubits. De kracht hiervan is dat problemen, die met klassieke computers exponentieel moeilijker worden om op te lossen met het introduceren van extra variabelen, met quantumcomputers slechts lineair moeilijker worden.

Terwijl de hardware zich nog ontwikkelt, kunnen mensen zich al voorbereiden op de komst hiervan. De logische circuits die geprogrammeerd moeten worden om dergelijke oplossingen met qubits uit te denken, vereisen een nieuw soort denkwijze. Er zijn programmeertalen zoals Strange en Qiskit waarmee dergelijke circuits al geschreven kunnen worden in o.a. Java en Python, en computers en SaaS diensten die simulatoren of echte quantumcomputers aanbieden om dergelijke programma's te draaien. Het is momenteel vooral zaak om problemen die met quantumcomputers beter (of uitsluitend) opgelost kunnen worden te leren identificeren. Denk aan het traveling salesman probleem.

Een aparte eigenschap van qubits in superpositie is dat ze met elkaar verstrengeld kunnen worden. Wanneer één van de twee dan wordt uitgelezen, ongeacht de afstand tussen de qubits, zal de waarde van de ander automatisch ook direct bepaald zijn. Deze eigenschap heeft impact op de security van nu; met name op het vlak van cryptografie. Dit besef heeft o.a. geleid tot de totstandkoming van veilige quantumnetwerken. Ook hier is de hardware nog in ontwikkeling, en dienen componenten van het OSI model verder uitgewerkt te worden. Te zijner tijd zullen architecturen een plek kunnen gaan geven aan dergelijke netwerken.

Platforms

ADOPT

12. InFluxDB

ASSESS

13. Akka Serverless
14. TimescaleDB
15. Yugabyte

InfluxDB

ADOPT InfluxDB is momenteel één van de populairste time series databases. Influx Data (leverancier) biedt de OSS opensource-variant (enkelvoudige standalone instantie) en een betaalde enterprise-variant (gedistribueerd, schaalbaar). InfluxDB onderscheidt zich van andere time series databases in de benodigde opslag van data en de snelheid waarmee time series kunnen worden opgevraagd. Het product bevat alle benodigde ondersteuning om het systeem onderhoudsvrij in te regelen. Daarnaast biedt het basisondersteuning t.a.v. authenticatie- en autorisatievraagstukken. Voor integratie met andere systemen biedt InfluxDB standaard een (wat rudimentaire) HTTP(S) API en een HTTPS subscriptie-mogelijkheid. Influx Data biedt aanvullend Telegraf met tal van plugins geschreven door de community en libraries in verschillende programmeertalen voor een directe koppeling met InfluxDB.
De eerste production-ready versie 2.0 is een jaar uit en verbeterd. Versie 2.0 biedt nu de volledige ondersteuning van Flux en met aanvullende aggregatiefuncties naast de standaard query language. Flux maakt het nog makkelijker om de data uit een InfluxDB te combineren met andere (JDBC) databronnen. In al zijn eenvoud is InfluxDB een prima keuze als time series database. Deze eenvoud maakt wel dat voor specifieke toepassingen zoals bijvoorbeeld applicatie-monitoring mogelijk aanvullende producten nodig zijn om tot een complete oplossing te komen.

Akka Serverless

ASSESS Akka Serverless is een cloud service die zorgt voor een Akka-omgeving inclusief database. Hiermee kun je eenvoudig technieken zoals Event Sourcing, CQRS en CRDTs gebruiken.

Dit vinden we interessant omdat het een aantal stappen verder gaat in het verzorgen van connectiviteit vergeleken met bekende cloudomgevingen. Het zou daarom in de toekomst een interessante mogelijkheid kunnen worden.

TimescaleDB

ASSESS TimescaleDB is een interessante nieuwe speler op het gebied van timeseries databases. Concreet is het een open-source extensie op PostgreSQL die timeseries capabilities toevoegt aan de database. Een belangrijk aspect daarbij is de belofte dat timeseries data 94% minder diskopslag nodig heeft door het toepassen van op timeseries data toegespitste compressie.

De API van de database blijft op basis van SQL met volledig behoud van alle normale functionaliteit en compatibiliteit met normaal gebruik van PostgreSQL. Daarmee biedt het een unieke all-in-one oplossing voor software met zowel een relationele als timeseries database-behoefte.

De database heeft zijn eigen licentie voor de geavanceerde features (TSL), waarbij eigenlijk alles mag behalve het als eigen SaaS-dienst (DBaaS) doorverkopen. Niet verwonderlijk aangezien het bedrijf achter de database recentelijk 40 miljoen heeft opgehaald in een nieuwe investeringsronde ten behoeve van het opzetten en uitbouwen van hun eigen SaaS-dienst. Deze SaaS-dienst is 5 oktober 2021 live gegaan in drie AWS regions met o.a. auto-scaling van disk en point-in-time recovery. Daarmee biedt het wat je tegenwoordig van een DBaaS mag verwachten. Ook zelf deployen van TimescaleDB is simpel zat doordat verschillende partijen, waaronder TimescaleDB zelf, open-source OCI containers aanbieden met daarin de TimescaleDB extensie.

Wij zien TimescaleDB als een goede optie om te verkennen voor teams die reeds werken met PostgreSQL en een groeiende behoefte hebben aan het opslaan van timeseries data.

Yugabyte

ASSESS YugabyteDB is een horizontaal schaalbare cloud-native database die qua performance beter scoort dan bijvoorbeeld Cassandra. En dat terwijl YugabyteDB wel altijd consistentie garandeert. Vanaf versie 2.0 wordt echter niet alleen YugabyteDB's eigen op Cassandra geïnspireerde YCQL ondersteund maar ook een volledig PostgreSQL-compatible SQL API. Daarmee is het gelijk ook een van de enige horizontaal schaalbare SQL-databases. Anders dan de directe concurrent CockroachDB biedt YugabyteDB wel ondersteuning voor de volledige PostgreSQL query API met zaken als triggers en stored procedures.

De database, inclusief alle enterprise features, is volledig open-source. Dit vanwege de strategie om geld te verdienen met het leveren van DBaaS-diensten (SaaS) en niet met de database zelf. Op dit moment bieden ze zowel de optie tot het managen van een eigen DBaaS-service op eigen infrastructuur als een volledig gemanagede cloud-variant. Ook biedt Yugabyte zelf open-source helm charts en een operator voor deployment op Kubernetes. Dit samen met de horizontale schaalbaarheid maakt dat het ook echt als high available cluster op Kubernetes kan worden ingezet. Anders dan bijvoorbeeld PostgreSQL waarbij high availability met externe tooling moet worden ingeregeld en de schaalbaarheid beperkt blijft.

Wij zien YugabyteDB als goede optie voor een schaalbare database. Zowel self-managed direct op Kubernetes als vanuit de DBaaS-opties vanuit Yugabyte.

Tools

ADOPT

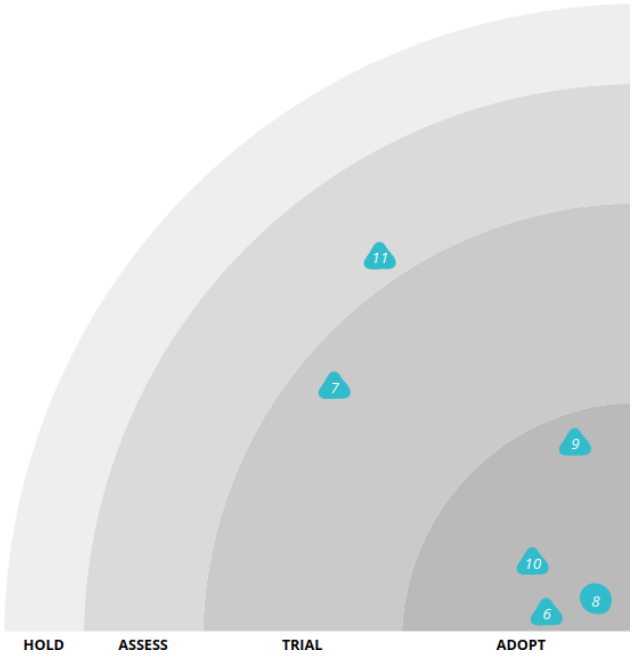
6. ArgoCD
8. Kafka-Connect
9. OpenRewrite
10. Sealed Secrets

TRIAL

7. Sentry

ASSESS

11. OpenTelemetry



ArgoCD

ADOPT ArgoCD is een tool die je in staat stelt om Kubernetes deployments te beheren. ArgoCD is een implementatie van GitOps, waarbij een koppeling gelegd kan worden tussen een Git repository aan de ene kant en een Kubernetes destination aan de andere kant. Zo kun je de state van een Kubernetes cluster geversioneerd beheren.

De belangrijkste kenmerken van ArgoCD zijn als volgt:

- Verschillen detecteren tussen de Git source en Kubernetes-destination en deze te synchroniseren
- Het automatisch synchroniseren naar de gewenste state op basis van een nieuwe commit in de Git repository
- Het monitoren van het Kubernetes cluster, om te bepalen of de applicatie healthy is
- Het automatisch restoren van een Kubernetes cluster, wanneer de actuele state van het cluster is gewijzigd en deze niet meer voldoet aan de gewenste state.

Binnen ArgoCD kunnen de Kubernetes resources op verschillende manieren worden gespecificeerd waaronder kustomize applications, helm charts, ksonnet applications en jsonnet files. Dit maakt het gebruik van de tool makkelijk en laagdrempelig voor degenen die al gewend zijn om met deze tools te werken.

In onze ervaring is ArgoCD een erg geschikte tool voor het uitrollen en beheren van Kubernetes applicaties.

Kafka-Connect

ADOPT Kafka-Connect stelt gebruikers in staat op grote schaal betrouwbaar data uit te wisselen tussen Kafka en andere systemen. Via een van de vele connectors kan data ofwel worden ingeladen naar Kafka topics vanuit externe bronnen, ofwel worden weggeschreven van Kafka topics naar externe systemen. Op die manier komt informatie beschikbaar voor streaming bewerkingen, of kan deze weggeschreven worden voor read optimized systemen of batchgewijze verwerking.

De inzet van Kafka-Connect maakt het eenvoudig patterns toe te passen zoals Change Data Capture, het voorkomen van dual writes, of het toepassen van Command Query Responsibility Segregation. Op die manier kun je complexiteit buiten een (micro) service houden, waardoor die zich kan focussen op kerntaken.

In onze ervaring is het een zeer krachtige tool gebleken voor de integratie van Kafka met andere systemen, zowel bij de inzet als Source als Sink.

OpenRewrite

ADOPT Met OpenRewrite is het eenvoudig om op grote schaal refactorings los te laten op een of meerdere repositories, voor onder andere framework versie-upgrades, om kwetsbaarheden op te lossen, en API's te migreren. OpenRewrite bouwt een Abstract Syntax Tree model op van de code, om aan de hand daarvan gericht aanpassingen door te voeren, met een eerste focus op Java en daaraan gerelateerde technologie. Wat het heel krachtig maakt is de uitgebreide set aan bestaande recepten, welke gezamenlijk in staat zijn complexe migraties uit te voeren zoals:

- JUnit 4 naar JUnit Jupiter migratie,
- Java 8 naar 11 migratie,
- Spring Boot, Micronaut & Quarkus upgrades,
- en het oplossen van CheckStyle issues.

Er zit een erg actieve groep ontwikkelaars achter, welke in rap tempo nieuwe features opleveren. Moderne.io stelt daarnaast grotere organisaties in staat op grote schaal deze refactorings toe te passen.

In onze ervaring, zowel bij de klant als bij opensource-bijdragen, is dit een zeer welkome tool om de taken op te pakken die anders wellicht wat langer blijven liggen, of nodeloos veel tijd kosten. Bijvoorbeeld een JUnit Jupiter migratie is hiermee in minuten uit te voeren, en vrijwel zonder risico op een proefproject los te laten.

Sealed Secrets

ADOPT Sealed Secrets zijn een manier om Kubernetes secrets toch onder versiebeheer te kunnen brengen door het toepassen van asynchrone encryptie. Bitnami heeft hiervoor een custom controller ontwikkeld die de encryptie en decryptie van de zogenaamde "sealed secret" resource regelt. Dit gebeurt in het cluster zelf waardoor de privésleutel het cluster nooit hoeft te verlaten.

De Kubeseal tool kan worden gebruikt om secrets om te zetten in sealed secrets, waarbij de publieke sleutel wordt gebruikt voor de encryptie. Dit bestand kan vervolgens veilig samen met de overige Kubernetes resources onder versiebeheer worden gebracht. Zodra het sealed secret wordt toegepast op het cluster zal de sealed secret controller deze omzetten en er ook een secret voor aanmaken.

Wij zien dit als goede oplossing voor het volledig onder versiebeheer brengen van een Kubernetes-configuratie zoals gebruikelijk is bij het toepassen van GitOps.

Sentry

TRIAL Sentry is een cross-platform applicatiemonitoring-tool met een focus op foutrapportages. Er zijn SDK's voor vrijwel elk platform, of het nou serverless, backend, frontend of mobiele applicaties betreft. Allemaal sturen ze optredende fouten voorzien van relevante context naar een Sentry server backend. Daarbij heeft Sentry intelligente ontdubbeling, en heeft het verscheidene alerting-instellingen en integraties. De backend-server is ofwel hosted af te nemen, zelf te hosten, of geïntegreerd met GitLab te gebruiken.

In onze ervaring is dit een onmisbare tool gebleken om onverwachte fouten aan het licht te brengen en te volgen, ongeacht de technologiestack. Daarnaast is de prijsstelling vriendelijker ten opzichte van bredere monitoring-oplossingen als DataDog, en kun je er eenvoudig op kleine schaal mee beginnen.

OpenTelemetry

ASSESS OpenTelemetry beoogt een eenvoudige, universele, vendor-neutrale, loosely coupled oplossing te zijn voor logging, metrics & tracing. Gesteund door de Cloud Native Compute Foundation en de grote spelers in de observability-wereld, heeft dit goede kans op termijn onze projecten te raken. Er zijn tools, API's en SDK's beschikbaar in verschillende talen voor het instrumenteren van applicaties, verzamelen van meetdata en om deze metrics, logs en traces te exporteren. Daarmee wordt het mogelijk de performance en het gedrag van applicaties te analyseren in een backend naar keuze.

Wij zien vooral de mogelijkheden om middels deze taal- en vendor-onafhankelijke standaard eindelijk één oplossing te hebben die herhaaldelijk is toe te passen, ongeacht de lokale situatie. Voor Java is er al een indrukwekkende set aan integraties met bestaande frameworks, als ook een Java agent om daar snel mee te kunnen starten.

Commit.
Develop.
Share.