

Autoren:

Markus Sümmchen
Nico Meisenzahl



Cloud Native Development für Software- hersteller

1. Vorwort	3
2. Was bedeutet Cloud-Native?	4
3. Wie Cloud-Native Sie unterstützt?	7
4. Mehrwerte für Sie und Ihre Endkunden	8
5. Ihre Etappen zu Cloud-Native	8
6. Wie Sie die Cloud-Native Herausforderungen meistern können?	11
7. Cloud-Native und Azure	12
8. Wie wir Ihnen helfen können?	14

1. Vorwort

Liebe Softwarehersteller,
mit diesem Whitepaper möchten wir Ihnen einen Überblick zu Cloud-Native Software und deren Entwicklung auf der Basis unserer Projekterfahrungen geben. Dies tun wir mit Empfehlungen und Best Practices rund um die Konzeption und Entwicklung von Cloud-Native Anwendungen und stellen für Sie den Nutzen von Azure als Plattform für Cloud-Native Anwendungen dar. Jedes Kapitel betrachten wir aus der Business-Sicht (durch Markus) und auf der technischen Ebene (durch Nico), hierbei immer stark geprägt durch unsere eigenen Projekterfahrungen.

Zu den Autoren:



Markus Sümmchen
CEO & Founder, white duck GmbH

Twitter: @msuemmchen
LinkedIn: linkedin.com/in/msuemmchen

Den ersten Teil meiner beruflichen Laufbahn habe ich wie Sie auf der Seite der Softwarehersteller verbracht. Nach dem Studium der Informatik in Rosenheim stieg ich bei einem lokalen und aufstrebenden Softwarehersteller aus der Business Intelligence Branche als Softwareentwickler ein. Über die folgenden 13 Jahre durchlief ich daraufhin die Positionen als Teamleiter, Entwicklungsleiter bis hin zum Mitglied der Geschäftsleitung mit der Verantwortung für die Bereiche Entwicklung, QS, Support und Produktmanagement. Bedingt durch diesen Werdegang bin ich sehr gut damit vertraut, was es bedeutet als Hersteller ein Softwareprodukt zu entwickeln, dies für die eigene Kundenbasis zu warten, funktional auszubauen und technologisch weiter zu entwickeln und auf einem modernen Stand zu halten.

Im zweiten Teil meiner beruflichen Laufbahn bin ich nun im 9. Jahr als Gründer und Geschäftsführer des Software- und Cloud-Engineering Dienstleister und Azure Spezialisten white duck in beratender und unterstützender Form mit vielen Softwareherstellern im engen Austausch. Immer wieder geht es um die Frage in welchen Schritten man mit einem bestehenden Produkt oder Produkt-Portfolio in die Cloud kommen und ein Cloud-basiertes Geschäftsmodell dafür aufsetzen und etablieren kann. Oftmals geht es dabei um Produkte in die 15-20 Jahre Entwicklung investiert wurde und welche zahlreiche fachliche Funktionalitäten bieten, aber leider auf der Basis veralteter Technologien.



Nico Meisenzahl
Senior Cloud & DevOps Consultant, Cloud & Data Management MVP, white duck GmbH

Twitter: @nmeisenzahl
LinkedIn: linkedin.com/in/nicomeisenzahl

Meine berufliche Laufbahn startete vor 12 Jahren mit einer Ausbildung zum Fachinformatiker im Bereich Systemintegration. Mit dem Abschluss meiner Ausbildung startete ich als Berater und unterstützte Kunden im Aufbau und Betrieb eigener, sowie Third-Party Lösungen. Seit mehreren Jahren sind hierbei Cloud, Open-source, Container und Kubernetes ein wichtiger Bestandteil meiner täglichen Arbeit.

Seit gut einem Jahr unterstütze ich unsere Kunden als Senior Cloud & DevOps Consultant auf ihrer Cloud-Native Reise. Hierbei berate ich im Cloud-Native Anwendungs- und Infrastruktur-Design, Containerisierung sowie der Cloud Automatisierung. Des Weiteren helfe ich unseren Kunden bei der Implementierung von DevOps Methodiken und Best Practices und fungiere hierbei auch als Bindeglied zwischen Entwicklung und Betrieb.

Als Microsoft MVP, Docker Community Leader und GitLab Hero bin ich seit Jahren ein aktives Community-Mitglied und teile sehr gerne mein Wissen und Erfahrung auf Konferenzen, Meetups und weiteren Plattformen.

2. Was bedeutet Cloud-Native?

Markus:

Cloud-Native ist die logische Folgerung aus den Entwicklungen der letzten 10-15 Jahren im Bereich der Softwareentwicklung. Von SOA (Serviceorientierte Architektur) bis hin zu Microservices, von Virtualisierung bis hin zu Containerisierung und von Hosting in Rechenzentren bis hin zu Cloud Computing ist Cloud-Native die Essenz einer langen Entwicklung. Diese ermöglicht uns nun skalierbare, wartbare und mehrmandantenfähige Cloud-Lösungen nach definierten Architektur- und Betriebsmustern in einem durchgängigen Prozess (DevOps) zu entwickeln und auch zu betreiben. Cloud-Native beschreibt also ein Gesamtkonzept, sowohl technisch als auch methodisch für den gesamten Lebenszyklus von modernen Cloud Anwendungen. Für Sie als Softwarehersteller ändert sich damit Ihr Entwicklungsprozess aber auch die Art und Weise wie Sie Ihre Software bereitstellen. Denn Cloud-Native ermöglicht Ihnen die Erweiterung Ihres Geschäftsmodells um den Betrieb Ihrer Softwarelösungen für Ihre Endkunden. Sie sind nicht mehr nur Softwarelieferant sondern auch Softwarebetreiber, mit all seinen Vorteilen, Nachteilen und Möglichkeiten, die ein erweitertes Geschäftsmodell so bietet.

Nico:

Der Begriff „Cloud-Native“ wurde von der Cloud-Native Computing Foundation ins Leben gerufen. Unter Cloud-Native versteht man Anwendungen die basierend auf modernen, dynamischen und skalierbaren public, private oder hybrid Cloud-Lösungen entwickelt werden. Der Einsatz von Containern, Microservices, deklarative APIs, Automatisierung sowie unveränderliche Infrastrukturen ist hierbei eine Kernkomponente, um lose gekoppelte, belastbare und einfach verwaltbare Anwendungen zu entwickeln und zu betreiben.

Cloud-Native setzt hierbei zudem einen Schwerpunkt auf eine agile Arbeitsweise, um neue Funktionen schnell und mit minimalem Arbeitsaufwand bereitstellen zu können.

Die Cloud-Native Computing Foundation versucht hierbei die Verbreitung von Cloud-Native voranzutreiben, indem sie ein Ökosystem von herstellerneutralen Open-Source-Projekten fördert und aufrechterhält.

Quelle: <https://github.com/cncf/foundation/blob/master/charter.md>

Die Cloud-Native Trail Map beschreibt hierbei übersichtlich die wichtigen Eckpunkte, von Cloud-Native:

- Containerisierung**
 Container helfen die Anwendung und ihre Abhängigkeiten deklarativ zu bündeln. Hierbei sollte Ihre Anwendung nach funktionellen Schwerpunkten in einzelne Microservices unterteilt werden.
- Continuous Integration / Continuous Delivery**
 Automatisierung ist hierbei eine Kernkomponente und sollte auf Ihrer Cloud-Native Reise von Anfang an betrachtet werden. Durch Automatisierung können neue Funktionen schnell und mit geringen Aufwänden bereitgestellt werden.
- Orchestrierung**
 Eine Orchestrierung der Microservices ist essenziell, um einen fehlerfreien Betrieb der Anwendung sicherzustellen. Die Auswahl der Plattform sollte basierend auf den jeweiligen Voraussetzungen erfolgen. Kubernetes ist hierbei ein beliebter Orchestrator jedoch nicht der Einzige.
- Beobachtbarkeit & Analyse**
 Monitoring, Logging und Tracing sind die drei Säulen der Beobachtbarkeit. Diese helfen Ihnen Ihre Anwendung erfolgreich zu betreiben und können Ihnen zudem tiefreichenden Business Metriken bereitstellen.

Neben diesen Kernthemen beschreibt die Cloud-Native Trail Map weitere wichtige Teilbereiche, die für verteilte Cloud-Native Anwendung essenziell sind.

CLOUD NATIVE COMPUTING FOUNDATION

CLOUD NATIVE TRAIL MAP

The Cloud Native Computing Foundation has a large number of options. This Cloud Native Trail Map is a recommended process for leveraging open source cloud native technologies. At each step, you can choose a vendor-supported offering or do it yourself, and everything after step #3 is optional based on your circumstances.

HELP ALONG THE WAY

A. Training and Certification
 Consider training offerings from CNCF and then take the exam to become a Certified Kubernetes Administrator or a Certified Kubernetes Application Developer cncf.io/certification

B. Consulting Help
 If you want assistance with Kubernetes and the surrounding ecosystem, consider leveraging a Kubernetes Certified Service Provider cncf.io/cscsp

C. Join CNCF's End User Community
 For companies that don't offer cloud native services externally cncf.io/end-user

WHAT IS CLOUD NATIVE?

Cloud native technologies empower organizations to build and run scalable applications in modern, dynamic environments such as public, private, and hybrid clouds. Containers, service meshes, microservices, immutable infrastructure, and declarative APIs exemplify this approach.

These techniques enable loosely-coupled systems that are resilient, manageable, and observable. Combined with robust automation, they allow engineers to make high-impact changes frequently and predictably with minimal tooling.

The Cloud Native Computing Foundation seeks to drive adoption of this paradigm by fostering and sustaining an ecosystem of open source, vendor-neutral projects. We demonstrate state-of-the-art patterns to make these innovations accessible for everyone.

1. CONTAINERIZATION

- Commonly done with Docker containers
- Any file operations and dependencies (even PHP 1.1 code running on an emulator) can be containerized
- Over time, you should expect benefits (easier to update, applications and writing future functionality as microservices)

2. CI/CD

- Setup Continuous Integration/Continuous Delivery (CI/CD) so that changes to your source code automatically result in a new container being built, tested, and deployed to staging and eventually, perhaps, to production
- Setup automated builds, unit tests and testing
- Argo is a set of Kubernetes native tools for defining and running jobs, workflows, workflows, and events using GitOps paradigms such as continuous and progressive delivery and ML ops

3. ORCHESTRATION & APPLICATION DEFINITION

- Kubernetes is the most leading container orchestration solution
- You should select a Certified Kubernetes Distribution
- Hosted Platform, or Installer: cncf.io/certified
- Helps Onboard you define, install, and upgrade even the most complex Kubernetes application

4. OBSERVABILITY & ANALYSIS

- Pick solutions for monitoring, logging and tracing
- Consider CNCF projects Prometheus for monitoring, fluentd for logging and Jaeger for tracing
- For tracing, look for an OpenTracing-compatible implementation like Jaeger

5. SERVICE PROXY, DISCOVERY, & MESH

- Consul is a fast and flexible tool that is useful for service discovery
- Envoy and Linkerd each enable service mesh architectures
- They offer health checking, routing, and load balancing

6. NETWORKING, POLICY, & SECURITY

To enable more flexible networking, use a CNF-compliant network project like Calico, Flannel, or Weave Net. Open Policy Agent (OPA) is a general purpose policy engine with logging, logging, authorization and admission control for user filtering. Falco is an anomaly detection engine for cloud native.

7. DISTRIBUTED DATABASE & STORAGE

These tools need more resiliency and scalability than you can get from a single database. Vertica is a good option for running MySQL at scale through sharding. CockroachDB is a storage technology that organizes a diverse set of storage solutions into Kubernetes. Serving as the "brain" of Kubernetes, etcd provides a reliable way to store data across a cluster of machines. TiKV is a high-performance distributed transactional key-value store written in Rust.

8. STREAMING & MESSAGING

When you need higher performance than J2EE REST, consider using gRPC as a back-end. gRPC is a universal RPC framework. NATS is a multi-modal messaging system that includes request-reply, publish and last-in/first-out queues. CloudEvents is a specification for describing event data in common ways.

9. CONTAINER REGISTRY & RUNTIME

Helm is a registry that stores, signs, and scales content. You can use alternative container runtimes. The most common, both of which are OCI-compliant, are containerd and CRIO.

10. SOFTWARE DISTRIBUTION

If you need to do secure software distribution, enable history, an implementation of the Update Framework.

cncf.io
v20200507

Für eine größere Ansicht der Grafik bitte anklicken.

3. Wie Cloud-Native Sie unterstützt?

Markus:

Mit den einzelnen Cloud-Native Paradigmen wie DevOps, Microservice Architektur, Containerisierung und Continuous Integration / Continuous Delivery (CI/CD) bekommen Sie als Softwarehersteller reife Methoden und reife Technologien an die Hand, um Ihre digitalen, cloud-basierten Produkte zu entwickeln und zu vermarkten. Auf der einen Seite sind die Techniken inzwischen ausgereift, die Ecosysteme lebendig und mit Open-Source erhalten diese eine nie da gewesene Dynamik. Auf der anderen Seite sind die Public Cloud Plattformen, wie die Microsoft Azure Cloud, optimal für die Cloud-Native Entwicklung ausgerichtet und bieten alle Services, Frameworks und Tools, um erfolgreich Cloud-Native Anwendungen zu entwickeln.

Mit modernen Entwicklungsparadigmen, Tools und Frameworks erzielen Sie Geschwindigkeit in der Implementierung, da oftmals ein höherer Vorfertigungsgrad genutzt werden kann und die neuen Techniken besser zu den neuen Anwendungsfällen wie Cloud, Software-as-a-Service (SaaS) und Mobile passen. Des Weiteren ist es für die Motivation des eigenen Entwicklungsteams enorm wichtig, moderne Entwicklungsparadigmen wie Cloud-Native zu nutzen, um die Mitarbeiter und Kollegen auf einem aktuellen Wissensstand zu halten, aber auch um neue Entwickler am Markt zu gewinnen. Softwareentwicklung auf einem alten Technologie-Stack ist gerade bei jüngeren Menschen nicht mehr gefragt.

Nico:

Die Entwicklung von Cloud-Native Anwendungen bringt Ihnen weitreichende Vorteile. Microservices-Architekturen werden als ein System von miteinander verbundenen Microservices aufgebaut, was Ihnen erlaubt einzelnen Funktionen Ihrer Anwendung unabhängig zu betrachten. Dabei sind die einzelnen Microservices nicht nur von einander entkoppelt, sondern auch von der Infrastruktur abstrahiert. Diese Entkopplung hilft Ihnen zudem auf notwendige Anpassungen schnell zu reagieren, um zum Beispiel eine neue Funktion auszuliefern und sie an einer bestimmten Kundengruppe zu testen oder aber um einzelne Teilbereiche Ihrer Anwendung zu skalieren.

Dies und weitere Vorteile von Cloud-Native hilft Ihnen Komplexität zu reduzieren und unterstützt Sie somit schneller Mehrwerte zu generieren. Die Azure Cloud sowie verschiedenste Open-Source Projekte bieten Ihnen hier die perfekte Grundlage.

Mit Cloud-Native Anwendungen machen Sie Ihr Unternehmen fit für die Zukunft. Es verschafft Ihnen einen Wettbewerbsvorteil, da sie sich mit einer skalierbaren und flexiblen Architektur mehr auf die Differenzierung Ihres zugrunde liegenden Geschäftsmodells konzentrieren können.

4. Mehrwerte für Sie und Ihre Endkunden

Markus & Nico:

Mit Cloud-Native hält in der Welt der Softwareentwicklung ein neuer Technologie-Stack und eine neue Methodik der Entwicklung und der Bereitstellung von Software Einzug. Durch diese Veränderung schreiten wir auf der einen Seite in das nächste Level der Softwareentwicklung. Auf der anderen Seite werden der Rollout und die Bereitstellung von Software

über die Cloud und als SaaS-Angebot noch einmal mehr forciert. Die Konsequenz daraus ist, dass mehr und mehr Cloud-basierte Softwarelösungen und -produkte entstehen.

Für Sie als Softwarehersteller besteht der Mehrwert darin die neuen Distributionswege von Software effektiv zu nutzen und somit in Abhängigkeit Ihres Produkts im besten Falle die gesamte Welt als Markt zu adressieren. Des Weiteren werden Sie durch die Etablierung des Cloud-Native Paradigmas ein interessanter Arbeitgeber im IT-Markt. Aber auch auf die Bewertung Ihres Unternehmens wirkt sich eine Cloud-Native Strategie positiv aus, da Sie sich damit zukunftsfähig aufstellen und für zukünftige Cloud-basierte Geschäftsmodelle gerüstet sind.

Aus der Sicht Ihrer Endkunden entwickeln Sie sich mit einer Cloud-Native Agenda zu einem Softwarelieferanten, der auf der Höhe der Zeit agiert und auch in Zukunft ein verlässlicher Partner sein wird. Denn Ihre Kunden wollen Software einfach und unkompliziert konsumieren, sowie sie es aus der Welt der Smartphone-Apps seit längerem bereits gewöhnt sind.

In Summe hilft Ihnen Cloud-Native auf die Anforderungen des Marktes und Ihrer Kunden schneller reagieren zu können. Mit der Umsetzung Ihrer Lösung als eine Cloud-Native Anwendung steigern Sie Ihre Flexibilität bei gleichzeitiger Reduzierung Ihrer Aufwände. Dies hilft Ihnen schneller auf Anforderungen Ihrer Endkunden eingehen zu können und notwendige Funktionen mit einer geringen Time-to-Market bereitzustellen. Durch einen hohen Automatisierungsgrad und dezentrale Microservices reduzieren Sie zudem Komplexität und somit auch Aufwände und Kosten in der Produktentwicklung.

5. Ihre Etappen zu Cloud-Native

Markus:

Aus der Sicht eines Softwareherstellers mit Bestandsprodukten gibt es im Wesentlichen zwei Szenarien, um in die Cloud-Native Welt einzusteigen.

Ablöse eines Bestandsprodukts durch eine Neuentwicklung

Die Neuentwicklung mit dem Ziel der Ablöse eines Bestandsprodukts, das über viele Jahre entwickelt und gewachsen ist, ist sicherlich die größte Herausforderung für Softwarehersteller. Bei diesem Szenario muss ein bewährtes und oftmals sehr reifes Bestandsprodukt mit neuen Techniken entwickelt und auch noch besser nachgebaut werden. Die Bestandskunden erwarten mindestens die gleiche Funktionalität, nur moderner und intuitiver in der Bedienung bei einem geringen Migrationsaufwand. Dies sind zwei Hauptanforderungen die schwer erreichbar sind und auf dem Zeitstrahl oftmals einen Entwicklungsaufwand von 5 Jahren und mehr erfordern. Wenn das Bestandsprodukt und das Business Modell, was dahinter steht über diesen Zeitraum noch tragfähig ist und man als Hersteller die Entwicklungskapazitäten personell und wirtschaftlich verfügbar hat, ist dies sicherlich die beste Art und Weise sein Business in die Zukunft zu bringen. In Summe aber auch die Aufwendigste.

Entwicklung eines neuen Produkts für eine neue Zielgruppe

Dem gegenüber steht als einfacherer Ansatz die Entwicklung eines neuen Produkts, das nicht mit dem Bestandsprodukt konkurriert, aber dennoch einen Mehrwert in der Bestandskundenbasis bietet und neue Kundensegmente adressiert, sowie neue Umsatzströme ge-

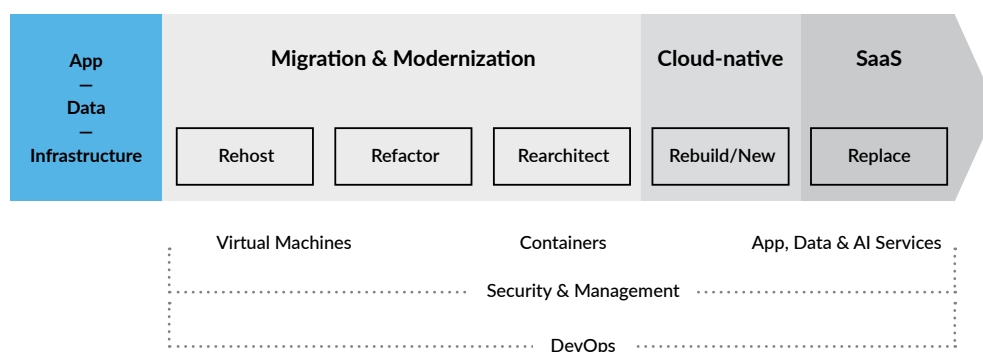
neriert. Optimaler Weise geht dies auch mit einer Transformations-Roadmap des Geschäftsmodells einher. Mit diesem Szenario kann ein kleines Team auf der grünen Wiese beginnen, schnell kleine neue Feature-Sets entwickeln und ausliefern und damit erste Mehrwerte für die Kunden schaffen. Dafür bieten sich Cloud-Native Anwendungen natürlich an.

Für beide Szenarien sollte betrachtet werden, inwieweit das Bestandsprodukt über einen Rehost-, Refactor-, Rearchitect-Ansatz dennoch modernisiert und in die Cloud portiert werden kann. Mit diesem evolutorischen Ansatz kann Zeit gewonnen und das Thema Cloud-Lösung am Markt adressiert werden. Wobei man sich bewußt sein muss, dass dieser Ansatz immer mit Einschränkungen, auf der technischen Seite aufgrund alter Softwarearchitekturen und Techniken, oder auf der wirtschaftlichen Seite bedingt durch höhere Kosten für Cloud-Ressourcen, einhergeht.

Nico:

Sie können Ihre Cloud-Native Reise in einem Schritt gehen, jedoch auch die einzelnen Teilbereiche in Etappen beschreiten. Hierbei kann Sie die Azure Cloud wie folgt unterstützen:

- Rehost, Refactor und Rearchitect beschreiben die Migration in die Cloud, sowie die Modernisierung Ihrer Anwendung
- Rebuild und Replace beschreiben die Transformation zu einer Cloud-Native Anwendung



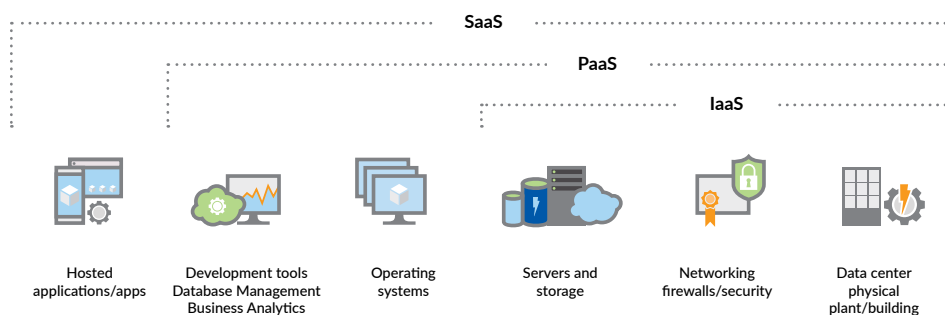
„**Rehost**“ auch bekannt als „Lift & Shift“, ist ein gängiger Ansatz für die Migration von Anwendungen in die Cloud. Sie verlagern Ihre Anwendung mit minimalen Anpassungen mit Hilfe von Infrastructure-as-a-Service (IaaS) in die Cloud.

Bei „**Refactor**“ geht es darum, Anwendung so anzupassen, dass diese mit Hilfe von Platform-as-a-Service (PaaS) Diensten zusammenarbeiten können. Dies erfolgt, um Komplexität und Kosten zu reduzieren, sowie Ihre Geschwindigkeit beim Ausliefern von neuen Funktionen zu erhöhen.

„**Rearchitect**“ ermöglicht es Ihnen, die Anwendung in mehrere funktionale Komponenten zu unterteilen, die individuell angepasst und weiterentwickelt werden können. Hierbei können Architekturentscheidungen mit Hilfe von PaaS-Diensten neu betrachtet und definiert werden. Sie erhalten dadurch mehr Flexibilität, Skalierbarkeit und Kontrolle über Ihre Anwendung und Architektur.

Eine „**Rebuild**“-Migrationsstrategie bedeutet, dass Sie Ihre Anwendung auf einer PaaS-Infrastruktur komplett neu entwickeln - im Gegensatz zu Rearchitect, wo Sie nur Teile der Anwendung modifizieren. Hierbei entwickeln Sie Ihre Anwendung unter Berücksichtigung von Cloud-Native Methoden komplett neu. Dabei haben Sie vollumfängliche Möglichkeiten und können die Cloud-Dienste bestmöglich nutzen.

Bei einer „**Replace**“ Migration wird eine bestehende Anwendung vollständig durch Software-as-a-Service (SaaS) Dienste ersetzt. Sofern bestehende SaaS-Anwendungen Ihnen alle erforderlichen Funktionalitäten bieten, können Sie mit dieser Strategie zukünftig Kosten bei der Weiterentwicklung und dem Betrieb einsparen.



6. Wie Sie die Cloud-Native Herausforderungen meistern können?

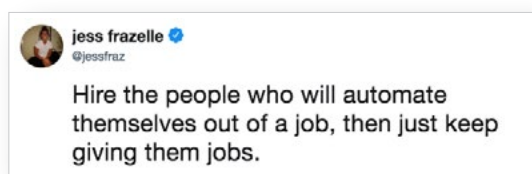
Markus:

Der Schritt zu Cloud-Native Produkten und einem Cloud-Native Geschäftsmodell bringt neben einem Technologiewechsel auch einen Kulturwechsel im Unternehmen mit sich, der von „oben“ gelebt werden muss. Dieser Tatsache sollte sich jeder Unternehmer, CEO oder Entscheider bewusst sein. So ist es als CEO eines Softwareherstellers heutzutage wichtig nahe an der Technik zu sein. D. h. nicht, dass der CEO oder Entscheider alles im Detail verstehen muss oder gar selbst Software entwickeln können muss. Es ist aber von großem Vorteil sich mit seinen Cloud Solution und Software Architekten unterhalten und einschätzen zu können, um was es geht und was der richtige Weg ist. Ist dies nicht der Fall, so sollte beratende Unterstützung hinzugezogen werden ggf. sogar aus dem eigenen Entwicklungsteam. So zeigt die Erfahrung, dass sich die Softwareentwicklungs-Teams meist selbst bereits mit Cloud-Native Entwicklungsthemen beschäftigen und oftmals eine recht konkrete Meinung oder Ideen im Kontext zum eigenen Produkt haben. In der Regel hängen Softwareentwickler mit Herzblut an „Ihrem“ Produkt und haben meist auch sehr konstruktive Vorschläge für dessen Weiterentwicklung oder Neuentwicklung, die nicht überhört bleiben sollten. Die generelle Vorgabe in Richtung Cloud-Native zu marschieren muss, aber vom Management kommen und auch von diesem komplett unterstützt und gefördert werden. Denn hinter Cloud-Native Development stecken nicht nur neue Techniken, sondern auch ein Paradigmenwechsel für das gesamte Unternehmen.

Nico:

Hinter Cloud-Native steht nicht ausschließlich nur die Technologie, sondern auch die Implementierung verschiedenster Methodiken und die Umstellung der täglichen Arbeitsweise. Der wahrscheinlich wichtigste Baustein ist DevOps. DevOps beschreibt eine Denkweise und Methodik, bei der es darum geht Development und Operations in einem Team zu vereinen um daraus Mehrwerte zu generieren. Jedoch wird DevOps häufig fehlverstanden und mit der Einführung einer Toolchain gleichgesetzt. Es geht hier jedoch um weitaus mehr – Einen Paradigmenwechsel. Ohne diesen lassen sich Cloud-Native Anwendungen jedoch nicht erfolgreich umsetzen.

Wie schon angesprochen ist auch die Automatisierung ein weiterer wichtiger Teil. Es muss sichergestellt werden, dass alle wiederkehrenden Aufgaben automatisiert sind, um eine gleichbleibende Qualität und auch ein schnelles ausliefern von Funktionen zu ermöglichen. Ein gutes Beispiel kommt hier von Jessie Frazelle:



Des Weiteren ist die Containerisierung eine wichtige Komponente von Cloud-Native. Container helfen uns unsere Anwendungen mit ihren Anforderungen zu paketieren und somit von der zugrundeliegenden Infrastruktur zu abstrahieren. Dies ist ein wichtiger Bauteil, um Ihre Anwendung skalierbar, flexibel und mit gleichbleibender Qualität zu betreiben. Hierbei stehen Ihnen verschiedenste Plattformen zur Verfügung. Ein Beispiel ist Kubernetes. Kubernetes ist in aller Munde und neben Linux wahrscheinlich das bekannteste und größte Open-Source Projekt. Kubernetes ist für Cloud-Native konzipiert und kann uns somit helfen verschiedenste Hürden in der Cloud-Native Welt zu bewältigen. Jedoch ist es nicht die einzige Plattform. Prüfen Sie, ob Kubernetes Ihnen in Ihrem Anwendungsfall hilft und entscheiden Sie sich aktiv dazu. Implementieren Sie Kubernetes nicht, weil es jeder tut!

7. Cloud-Native und Azure

Markus:

Für die Entwicklung von Cloud nativen Anwendungen bietet Microsoft einen sehr durchgängigen und reifen Baukasten. Mit der Azure Cloud steht, unserer Ansicht nach und aus dem Blickwinkel eines Softwareherstellers, die umfassendste Public Cloud am Markt zur Verfügung. Sämtliche Cloud-Native Technologien wie Container-Orchestrator, Datenbank-Dienste und Identity Provider (Azure AD und Azure AD B2C) stehen ebenfalls in sehr ausgereifter Form zur Verfügung. Es werden alle gängigen Open-Source Tools und Frameworks für die Entwicklung unterstützt und es gibt eine tiefe Integration in Application Lifecycle Management (ALM) Plattformen, wie Azure DevOps und GitHub für die Abbildung von CI/CD Prozessen. Die Entwicklungsumgebungen wie Visual Studio und Visual Studio Code sind ebenfalls tief integriert. Darüber hinaus gibt es ein sehr lebendiges Ecosystem und Communities, welche eine Vielzahl an Code-Beispielen z. B. auf GitHub bereitstellen

oder auch vordefinierte Lösungen über den Azure Marketplace publizieren. Gerade bei der Entwicklung von B2B Lösungen ist Microsoft mit seinen Technologieplattformen ein sehr starker Partner, denn neben der Azure Cloud als Betriebsplattform sind auch Integrationen in die Microsoft 365 Produktivitätsplattform und dessen Marktplätze einfach möglich und erweitern die Zielgruppe für die Cloud-Native Anwendungen. Hervorzuheben als einen zentralen Azure Service ist definitiv das Azure Active Directory (AAD). Hier werden alle notwendigen Identitäten zentral abgelegt und verwaltet um die Authentifizierung und Autorisierung durchgängiger Business Szenarien und Prozesse mit verschiedenen Anwendungen integriert abbilden zu können. Beispielhaft bedeutet dies nur eine Identität für die Authentifizierung und Autorisierung von Microsoft 365, Dynamics CRM und Ihrem eigenen branchenspezifischen Cloud-Produkt. Dies ist ein zentraler Mehrwert für die Nutzung von Azure und dessen Ecosystem für die Entwicklung von Cloud-Native Anwendungen.

Nico:

Ihre Microservices können Sie mit folgenden Azure Services betreiben:

- **Azure Kubernetes Service**
Ein hochverfügbarer, sicherer und vollständig verwalteter Kubernetes-Dienst.
- **Azure Container Instances**
Ein Dienst der das Betreiben von Containern serverless für Batch und event-basierte Workloads ermöglicht.
- **Azure Functions**
Beschleunigen und Vereinfachen der Anwendungsentwicklung mit serverless und event-basierten Functions.
- **Azure Red Hat OpenShift**
Eine vollständig verwaltete OpenShift Umgebung, die gemeinsam von Red Hat und Microsoft betrieben wird.

Azure bietet mit der **Azure Container Registry** einen verwalteten Dienst zum Verwalten und Erstellen eigener Container Images.

Continuous Integration / Continuous Delivery können über **Azure DevOps Pipelines** oder **GitHub Actions** umgesetzt werden.

Identitäts- und Zugriffsverwaltung kann über ein vollständig verwaltetes **Azure Active Directory** (AAD und AAD B2C) implementiert werden.

Azure Storage ermöglicht die Ablage und das Management von Dateien über CLI und REST-API.

Die Azure Cloud bietet zudem eine Vielzahl an verwalteten Datenbanken. Dazu zählen:

- **Azure CosmosDB**
Ein vollständig verwalteter NoSQL-Datenbankdienst für die moderne App-Entwicklung. Sie ist schnell, flexibel nutzbar und skalierbar
- **Azure SQL Database**
Eine verwaltete und intelligente SQL Datenbank, die mit Ihren Anforderungen skaliert
Azure Databases für MariaDB, PostgreSQL und MySQL
Verwaltete open-source Datenbanken für jeden Use-Case
- **Azure Cache for Redis**
Eine vollständig verwaltete Version des beliebten Open-Source Redis Servers

Mit **Azure Monitor** bekommen sie eine verwaltete Beobachtungs-Umgebung, mit der Sie ihre Logs und Metriken verwalten können. Des Weiteren können Sie mit **Application Insights** Probleme über Anwendungen und Abhängigkeiten hinweg erkennen und diagnostizieren.

Verteilte Microservices können mit Hilfe von **Event Grid, Service Bus und API Management** mit einander integriert werden.

Des Weiteren stellt Azure hilfreiche Developer-Tools und Open-Source Projekte für die Cloud-Native Community bereit. Einige davon sind:

- **Dapr**
Eine ereignisgesteuerte, portable Laufzeitumgebung für den Aufbau von Microservices in der Cloud und Edge
- **Keda**
Eine ereignisgesteuerte Autoskalierung für Kubernetes Workloads
- **GitHub Codespaces**
Eine Online-Entwicklungsumgebung mit der Sie vollständig in der Cloud entwickeln können

8. Wie wir Ihnen helfen können?

Markus & Nico:

Wir können Sie als Softwarehersteller unterstützen Ihre Cloud-Native Reise schneller und effektiver zu beschreiten. Mit uns als Partner können Sie schneller die Vorteile von Cloud-Native nutzen und somit schneller und flexibler auf Ihre Kundenanforderungen reagieren.

Wir sind ein auf Software- und Cloud Native-Engineering spezialisiertes Technologiehaus. Mit Hilfe unserer Cloud-Native Journey können wir Sie in den verschiedensten Situationen und Stadien, in denen Sie sich befinden, unterstützen. Unsere Kunden vertreten alle Branchen und sind Unternehmen vom kleinen Start-up mit einer Cloud-basierten Geschäftsidee, über Softwarehersteller mit komplexen Softwareprodukten bis hin zu Großunternehmen mit einer Digitalisierungsagenda, die wir bei der Transformation ihrer Geschäftsprozesse in die Cloud unterstützen. Dafür steht ein hochspezialisiertes Team von Cloud Solution Architects, Software Engineers, DevOps Engineers und Software Testern bereit, welches die Cloud-Lösungen für unsere Kunden bei Bedarf durchgängig und aus einer Hand konzeptioniert, entwickelt und betreibt oder Sie einfach nur mit Beratung und Coaching unterstützt.

Wir können Sie bei dem Rebuild oder Replace Ihrer bestehenden Lösung mit unserem Know-How unterstützen. In diesem Kontext analysieren wir zusammen mit Ihrem Entwicklungsteam Ihre bestehende Anwendung und Anforderungen und definieren die für Ihren Use-Case passende Cloud-Native Architektur. Hierbei greifen wir auf die PaaS und SaaS Dienste der Azure Cloud als Baukasten zurück. Im Anschluss stehen unsere Software und DevOps Engineers Ihrem Team zur Verfügung um diese bei der Entwicklung sowie der Betriebsvorbereitung zu unterstützen und zu beraten.

Entscheiden Sie sich statt einem Rebuild und Replace für die Entwicklung eines neuen Produkts und / oder Business-Cases empfehlen wir dies mit Hilfe eines Minimum Viable Product (MVP) zu beginnen. Hierbei definieren und entwickeln Sie im ersten Schritt die minimal funktionsfähige Version Ihres Produkts, um mit minimalem Aufwand den Kunden-, Markt- oder Funktionsbedarf zu decken. Auch in diesem Fall steht Ihnen unser Team von Experten zur Verfügung, um Sie bei der Konzeption, Entwicklung und dem Betrieb zu unterstützen.

Unser Angebot für Softwarehersteller:

- **Cloud Native Journey:** whiteduck.de/cloud-native-journey
- **Azure Workshop:** whiteduck.de/workshops
- **Azure für ISVs:** whiteduck.de/microsoft-azure-kaufen

Stichwortverzeichnis

A

Azure 3
Azure Active Directory (AAD) 11
Azure API Management 12
Azure Cache for Redis 12
Azure Container Instance 11
Azure Container Registry 11
Azure CosmosDB 12
Azure DevOps Pipelines 11
Azure Event Grid 12
Azure Functions 11
Azure Marketplace 11
Azure Monitor 12
Azure Red Hat OpenShift 11
Azure Service Bus 12
Azure SQL Database 12
Azure Storage 11

C

Cloud-Native 4
Containerisierung 5
Continuous Delivery 5
Continuous Integration 5

D

Dapr 12

G

GitHub Actions 11
GitHub Codespace 12

I

Infrastructure as a Service (IaaS) 8

K

Keda 12
Kubernetes 10
Kubernetes Service 11

M

Minimum Viable Product (MVP) 13

O

Orchestrierung 5

P

Platform as a Service (PaaS) 8

R

Rearchitect 8
Rebuild 9
Refactor 8
Rehost 8
Replace 8

S

Software as a Service (SaaS) 6

V

Visual Studio / Visual Studio Code 11