BRZ

Die Containerplattform des Bundes im BRZ
Einblicke in Planung und Betrieb

Mag. Rupert Zarl (BRZ)
Matthias Rettl (Red Hat)

Vorstellung

Matthias Rettl, Red Hat
Account Solution Architect

Mag. Rupert Zarl, Bundesrechenzentrum
Teamleiter Product Operations
Cloudmanagement und Middleware

Agenda

1. BRZ Containerplattform (PaaS)
   a. Rückblick und Aufbau
2. Erfahrungsbericht
   a. Strategische Erfolgsfaktoren
   b. Betriebserfahrungen
   c. Neue Paradigmen
   d. Skalierungsstrategien
3. Der Grüne Pass
4. Red Hat OpenShift

BRZ

Die BRZ Containerplattform (PaaS)

Entstehung
Status
Erfahrungen

Platform as a Service

Was ist PaaS…?

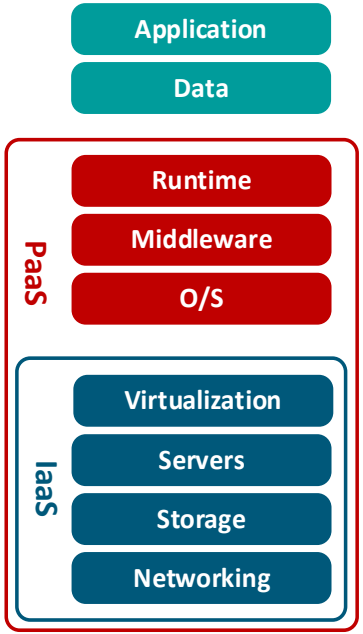
BRZ

> Plattform für die Entwicklung und den Betrieb von webbasierten Anwendungen
  - basiert auf Funktionen/Konzepten von Infrastructure as a Service (IaaS)

> IaaS plus Erweiterungen/Features zur Entwicklungs-/Betriebsplattform
  - dazu zählen Entwicklungs-, Test-, Betriebs- und Auslieferungsprozesse für Anwendungen

> Fokussierung auf effiziente/agile Anwendungsentwicklung
  - durch Bereitstellung von einsatzbereiten Laufzeit- und Entwicklungskomponenten

> Anwendungen nutzen Ressourcen der gesamten Plattform
  - skalierbare PaaS-Plattform dient als Laufzeitumgebung für Anwendungen



Platform as a Service

Warum PaaS…?

BRZ

> Reduktion der Setup-Zeiten

- initiale Aufbauthemen werden durch PaaS stark reduziert
- Infrastruktur wird in PaaS automatisiert und konfigurationsbasiert instanziiert und bereitgestellt

> Effizientere Nutzung von Hardware-Ressourcen

- gezielter Einsatz von Hardware-Ressourcen – Hardware muss nicht exklusiv für Spitzenlasten vorgehalten werden
- Lastspitzen können durch dynamische zeitraum-/lastspezifische horizontale Skalierung abgefangen werden

> Standardisierte und stets aktuelle Plattform

- zentrale Pflege/Wartung von Infrastruktur plus notwendige Komponenten für Anwendungsentwicklung/-betrieb
- Einspielen von Updates und Sicherheits-Patches wird durch Continuous Integration & Delivery automatisiert ermöglicht

Platform as a Service

Warum PaaS…?

BRZ

> Beschleunigung der Anwendungsentwicklung
  - Bereitstellung wiederverwendbarer Anwendungskomponenten
  - Für Entwicklung/Qualitätssicherung/Betrieb werden Laufzeitumgebung zentral und standardisiert zur Verfügung gestellt

> Effiziente Verwaltung des Anwendungslebenszyklus
  - gesamte LifeCycle einer Anwendung wird durch zentrale und standardisierte Funktionen unterstützt
  - rasche Inbetriebnahme neuer Anwendungen bzw. neuer Funktionalitäten

> Erhöhte Fehlertoleranz und Resilienz
  - durch Modularisierung von Anwendungen und deren Funktionen führen Fehler maximal zu Teilausfällen
  - raschere Fehlerbehebung durch effizienten Austausch der betroffenen Komponenten im Regelbetrieb
  - „Selbstheilung“ von Anwendungen durch automatisierten Ersatz/Neustart von ausgefallenen Komponenten

Auswirkungen auf Organisation und Technik

Was ändert sich durch PaaS?

BRZ

Organisation

> Ende-zu-Ende Verantwortung im Anwendungs-Team

> Benutzerzentrierung – Kundenbedürfnisse im Fokus

> agile Methoden – evolutionäre Entwicklung & Bereitstellung

> DevOps – geteilte Zuständigkeiten, autonome Teams

> ReDesign von Anforderungs- und Bereitstellungsprozessen

Technik

> Standardisierung – Konvention vor Konfiguration

> Automatisierung – erhöhte Konsistenz/Reproduzierbarkeit

> Mini- & Microservices – hochmodularer Anwendungsschnitt

> CI/CD Pipelines & dynamische Provisionierung

> API-First – Basis für nachhaltige Wiederverwendung

Containerplattform (PaaS) im BRZ

BRZ

Erste Schritte…

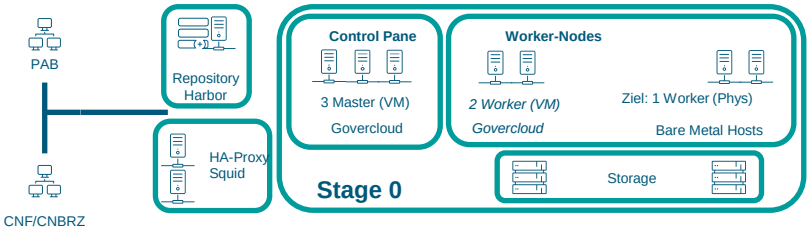
1	**2017:** PoC **2018:** Konzeption **2019:** Technologieauswahl, Ausschreibung, Roadmap **Wir wollen Container!**
2	**2019/2020: Errichtungsprojekt Red Hat Open Shift 4.x**
3	**2020: Start mit Schlüssel-Applikationen** **Strategischer Kapazitätsaufbau**
4	**Kontinuierlicher Kapazitätsaufbau** **06/2020: ca. 190 Cores, in 2 clustern** **12/2020: ca. 540 Cores, in 3 Clustern** **06/2021: ca. 880 Cores, in 3 Clustern**
5	**2021: Die PaaS als strategische Unternehmensplattform**
6	**2022: Fähigkeiten erweitern** **Bedarfsgesteuerte Kapazität** **Multi-Cluster-Management**

→ Stabile Betriebsumgebungen für ein Enterprise Umfeld.

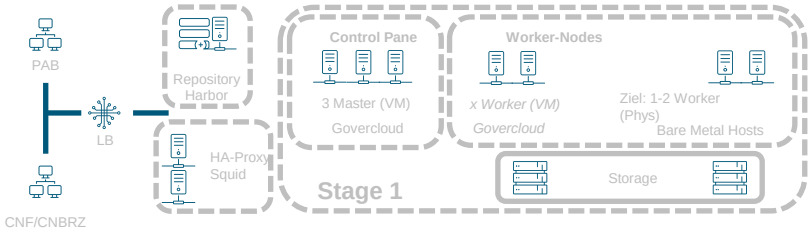
Stufenweise Aufbau der BRZ-PaaS in Stages
Einsatz von Red Hat OpenShift

BRZ

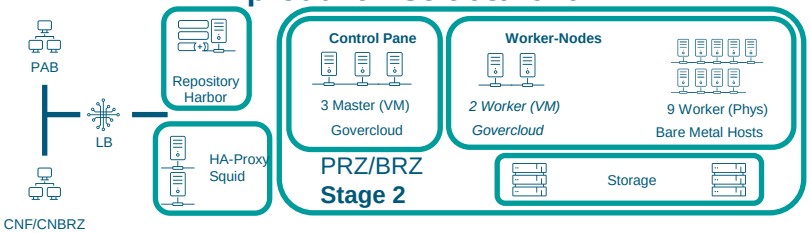
Stage 0 - System Development
produktiv seit 09/2020



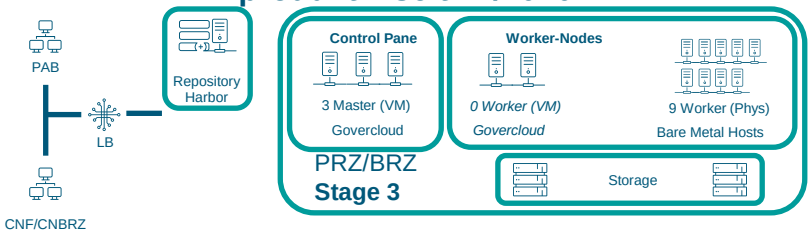
Stage 1 – Application Development & Test
bei Bedarf



Stage 2 – Test and Quality
produktiv seit 05/2020



Stage 3 – Production
produktiv seit 12/2020



Standard-Server als Bare Metal Worker (HP DL380), tw. GPU-Ausstattung

BRZ-intern

BRZ Containerplattform
Erfolgsfaktoren in der Aufbaustrategie

BRZ

Management-Commitment UND Graswurzelbewegung

> Gemeinsame Anstrengung von Architektur, Produktmanagement, Engineering und Operations
> Konkreter Management-Support für strategischen Produktaufbau
> PoC zur technischen Machbarkeit, Commitment der Applikationen
> Das Projekt ist gefordert zu performen.

Applikations-Mix am Start

> CPU-Hungrige Applikationen im Bereich AI, maschinelles Lernen, Datenanalyse
  - Beispiel: Betrugsbekämpfung – die Datenaufbereitung macht die Maschine
  - Kritische Größe für Skaleneffekte
> Strategische Applikationen für die Organisation:
  - Zentraler Jenkins als Entwicklungsunterstützung
  - Moderne Anwendungsarchitekturen (CI/CD, Microservices, Horizontales Skalieren)
> Kapazität für Applikationen, die rasch und flexibel skalieren müssen
  - Grüner Pass

BRZ Containerplattform
Expertenwissen zu Containerplattformen ist rar

BRZ

Kompetenzaufbau und Organisationales Lernen

> Ausbildungsoffensive zum Wissensaufbau in Tiefe und Breite
  - Architektur: Enterprise und Lösungsarchitekten
  - Management, Produktmanagement, Account-Management, Linie
  - Product Engineering und Product Operations: Neudefinition von Rollen und Aufgaben
> Wissensnetzwerk mit Schlüsselpersonen und Communities of Practice (CoP)
  - Interner Personalaufbau rund um Experts

Personal- und Know-How-Strategie

> Strategische Unterstützung: Spezielle Expertise für Architektur und Plattformaufbau
  - Partner: Peter Pfläging
> Operative Unterstützung für Engineering und Operations
  - Breite Kapazität für Development, Plattformbetrieb und Strategie
  - Partner: ATOS mit internationalem Netzwerk

BRZ Containerplattform
Operativer Betrieb

BRZ

Bemerkenswerte Stabilität der Plattform zum operativen Betrieb der Applikationen

> Robust, resilient, selbst heilend, implizite „Hochverfügbarkeit“

Effiziente Erweiterbarkeit mit Standard-Servern

> Remote Administration: Neue Maschinen werden „assimiliert“
> Globale Lieferketten trotz Pandemie und Suez-Kanal?

Betriebssicherheit und Schwachstellenmanagement

> Schwachstellenmanagement muss Regelprozess sein
> Häufige Updates versus fixe Wartungsfenster

Image courtesy of the Earth Science and Remote Sensing Unit, NASA Johnson Space Center
https://eol.jsc.nasa.gov/SearchPhotos/photo.pl?mission=ISS064&roll=E&frame=48480

Personalbedarf im Betriebsteam abhängig vom Supportmodell (7-17 versus 7x24 mit Rufbereitschaft)

> 3 bis 5 Personen mit unterschiedlichen Stärken
> Organisatorische Resilienz benötigt Mindestgröße
> Externe Partner zur Schließung operativer Kapazitätslücken

BRZ Containerplattform
Paradigmenwechsel

BRZ

Downtime und Wartungsfenster

> Es ist ein normaler Vorgang, dass ein Pod beendet und neu gestartet wird
  - „Stateless“ versus „Stateful“
> Plattformupdates werden mit Pufferkapazität rollierend durchgeführt, ein Knoten nach dem anderen, …
> Plattformupdates für die Applikationen „nicht merkbar“ – keine Downtime, keine fixen Wartungsfenster!
> Applikatorische Upgrades
  - Zusammenspiel von Applikation und Plattform
  - Prozessreife erforderlich

Steuern des Ressourcenkonsums durch die Applikation

> Wesentlicher Einfluss auf die Leistungsdichte
> Lesehausübung:
  - Resource Quotas mit Requests.CPU (=Mindestmaß für Pod) und Limits.CPU (=Maximalbedarf für Pod)
  - Quality of Service: „Guaranteed“ / „Burstable“ / „Best Effort“
> Optimierung durch Applikation und Plattform

BRZ Containerplattform
Paradigmenwechsel - Skalierungsstrategien

BRZ

Skalierung ist eine VUCA-Entscheidung (VUCA … Volatility – Uncertainty – Complexity – Ambiguity)

> Pattern of Business Activity für bestehende und neue Applikationen (Büro, Batch, Background, …)
> Erwartete, geschätzte Mengen als Treiber des Ressourcenbedarfs
> Externe Faktoren (nicht steuerbar, z.B. gesetzliche Rahmenbedingungen)
> Ausrichtung an den erwarteten Bedarfsspitzen und dem akzeptablen Leistungsbereich
  - Ideal: keine „Unterperformanz“ und keine „Überausstattung“

Handlungsspielraum Klassisch versus Container:

> Klassisch: Infrastrukturkonzept mit Datenbank-, Backend-Servern, Frontend-Servern, Webservern, …
  - Scale-Up = stärkere Server (CPUs, RAM) … ideal für „Monolithen“
  - Scale-Out = zusätzliche Server, sofern es die Architektur erlaubt
> Container: Namespace mit flexibel zugeteilten Ressourcen (CPU, RAM, Storage) und Pods mit Funktionen
  - Scale-Up … nicht ideal für Microservices
  - Scale-Out = es werden mehr Pods gestartet und nutzen die Ressourcen des Namespace.

Betriebserfahrungen
Skalierung in der Praxis (1/2)

BRZ

Bestandsapplikationen:

> Meine Applikation braucht klassisch X Cores + Y RAM + … => Wie sieht das auf der Containerplattform aus?
  - *„Das kommt darauf an“* … Monolith oder Microservices? Applikationsarchitektur? Pattern of Business Activity ? Grundlast und Spitzenlast? Performanz-Erwartung?

Ressourcenbedarf ermitteln:

> Faustschätzungen: Umrechnungsfaktor CPU (60%-80%) … reicht für grobe Abschätzung
> Schätzmodelle zur Errechnung des Ressourcenbedarfs
  - Lasttreiber und Mengenschätzungen
  - Last- und Performancetests auf Komponenten- / Service- / Applikations-Ebene
  - Methodenmix und Erfahrungsaustausch
> Risikomanagement und Kostenabwägung
  - Szenarien mit externen Faktoren berücksichtigen (erhöhte / zu niedrige Nutzung)
  - Auswirkung von Fehleinschätzungen

Betriebserfahrungen
Skalierung in der Praxis (2/2)

BRZ

Pragmatischer Ansatz

> „Scale-Up“ und „Scale-Down“ in Betriebseinschwingphase:
  - valide Strategien um den realen Ressourcenbedarf zu finden
  - Voraussetzungen:
    - Flexible Reaktionsfähigkeit der Plattform
    - Verfügbare Ressourcen
  - Anpassung ist reiner Konfigurationsvorgang durch Applikations-/Plattformteam
> „Scale-Up“:
  - Die Applikation wird initial mit Mindestressourcen ausgestattet. Bei Bedarf wird erhöht.
  - Eignung für kostensensible Applikationen, deren Performanz keine geschäftskritischen Auswirkungen hat.
  - Aufmerksames Finetuning durch Applikation und Plattform
> „Scale-Down“:
  - Schätzung als Basis für komfortable Ressourcenausstattung
  - Betriebseinschwingphase zeigt den realen Bedarf.
  - Eignung für performanzkritische Applikationen mit strategischer Bedeutung

BRZ

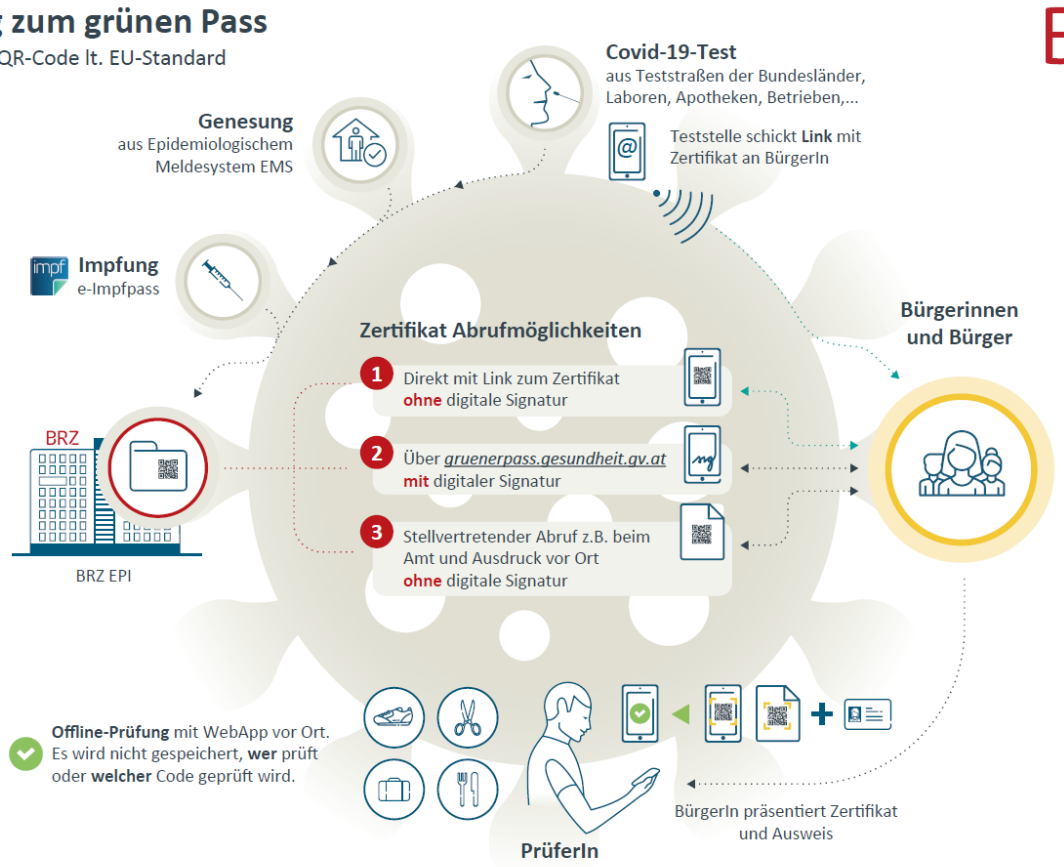
Der Grüne Pass

Erfahrungen aus Engineering und Operations

Der Grüne Pass

Fakten zum Grünen Pass

- 100% EU-konforme Umsetzung
- 2,5 Mio. App-Downloads
- ~8 Mio. Abrufe im Gesundheitsportal
- 30 Mio QR-Codes im BRZ-EPI erstellt
- 300.000 Genesungszertifikate
- 16,3 Mio. Testzertifikate
- 10,4 Mio. Impfzertifikate



https://www.brz.gv.at/presse/GruenerPass-BRZ-EPI.html

Auftraggeber: Bundesrechenzentrum

APA-AUFTRAGSGRAFIK

Der Grüne Pass
… Herausforderungen

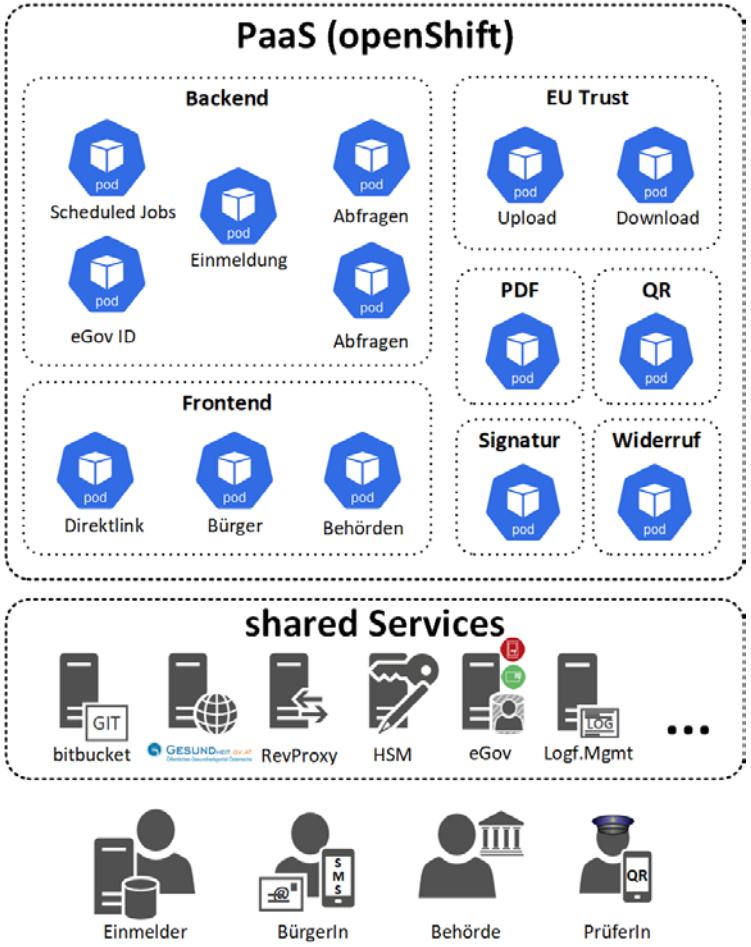
1. hohes öffentliches Interesse
2. sehr hoher Zeitdruck (rapid development)
3. Abhängigkeit von EU Projektfortschritt
4. hohe Sicherheitsanforderungen (Gesundheitsdaten)
5. große Anzahl von Datenlieferanten (Länder, Labore, eImpfpass, EMS)
6. Unbekannte Anzahl von Zugriffen (Skalierung!)

Warum Containerplattform?

BRZ

- Hohe Anzahl von Einmeldungen (Tests, Impfungen, Genesung)
- Anzahl der Abrufe war unbekannt - höchste Flexibilität in Dimensionierung (zB Erhöhung bei initialer Einmeldung von Impfungen)
- Gleiche Software durch Customizing mehrfach lauffähig
- Logfiles von allen pods zentral sammelbar
- CI/CD (wartungsfreies einfaches Deployment)
- durchgängiges Staging (dev -> test -> qs -> prod)
- schnelle Adaptierung der Stages für Lasttests
- Dynamische Zuteilung der Ressourcen (zeitlicher Bedarf der Use Cases)

Architektur „Grüner Pass“



- Nutzung vorhandener Shared Lösungen
  - Sicherheit (IPS, DDoS, Proxys, …)
  - eGov / PvP
  - Source Code Mgmt / Build
- Herzstück „Grüner Pass“ in PaaS
  - Stages als Projects abgebildet
  - ingress/egress-IP je Stage (Firewall)
  - Spezialisierung der services (feature flags mit gleichen Code)
  - Reuse vorhandener Appls
  - Parallele Entwicklung/Test
  - nur „stateless“ Services in PaaS

Dimensionierung durch Last- und Performance-Tests (LuP)

BRZ

> Mengenabschätzungen mit Kunden in Form Use-Cases pro Tag
> Aufteilung der Use Cases in Transaktionen
> Zielwert Transaktionen/Sekunde = Tagesmenge in 2 h (in 8h mit 4-fach Last)

1. pod Optimierung (1 pod, nur PaaS)
   - Optimierung der Software und Test der Stabilität
   - Dimensionierung des einzelnen pods (CPU/RAM)
2. Service Optimierung (mehrere pods, nur PaaS)
   - Horizontale Skalierung der pods auf Zielwert Transaktionen/Sekunde
3. Use Cases Optimierung
   - Skalierung der Services (inkl. Shared Services)

Erfahrungen…

- Eigene ingress/egress-IPs verursachen Zusatzaufwand ermöglichen aber Firewall-Freischaltungen auf IP Ebene (eingehend/ausgehend).
– Last- und Performance-Tests und ihre Validierung fallen in die zeitkritische Schlussphase des Projekts
– Dimensionierung wurde daher manuell (immer fixe CPU,RAM und # pods) gesetzt und nach der Anfangszeit reduziert.
✓ ArgoCD (OpenShift gitOps) hat sich als sehr hilfreich erwiesen.
– Flexibilität beim Einsatz aktuellster Technologien erforderlich: Kurzfristiger Workaround für Ingress-Routen wegen Stör-Effekten zwischen knative, Istio-Ingress und manueller DNS-Namensgebung, die das Hochskalieren verhinderten

BRZ

Red Hat Open Shift

Matthias Rettl,
Account Solution Architect

Red Hat is the
open hybrid cloud technology leader

We are a commercial software and services company with an open source development model

Red Hat

The world's leading provider of open source enterprise IT solutions

More than **90%** of the Fortune **500** use **Red Hat** products and solutions[1]

~16,000 employees

105+ offices

40+ countries

The first **$3 billion** open source company in the world[2]

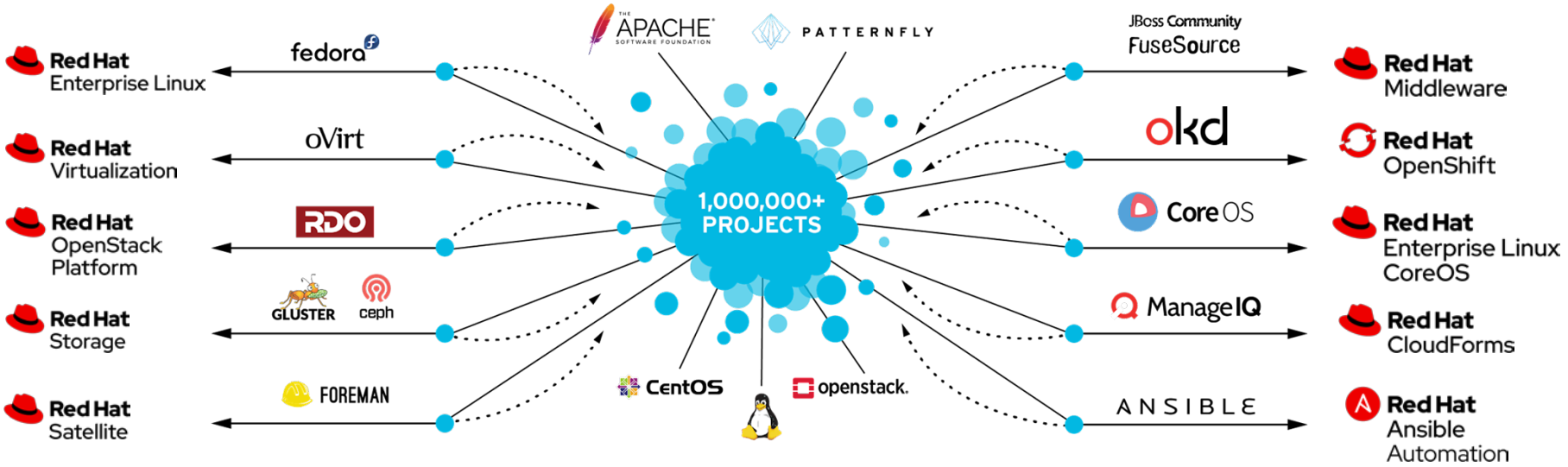
Sources: 1- Red Hat client data and Fortune 500 list, Jul. 2020. 2 - Red Hat SEC filings prior to the acquisition by IBM. Note: Currency in U.S. dollars.

Red Hat

IBM has acquired Red Hat in July 2019 for $34 billion, but

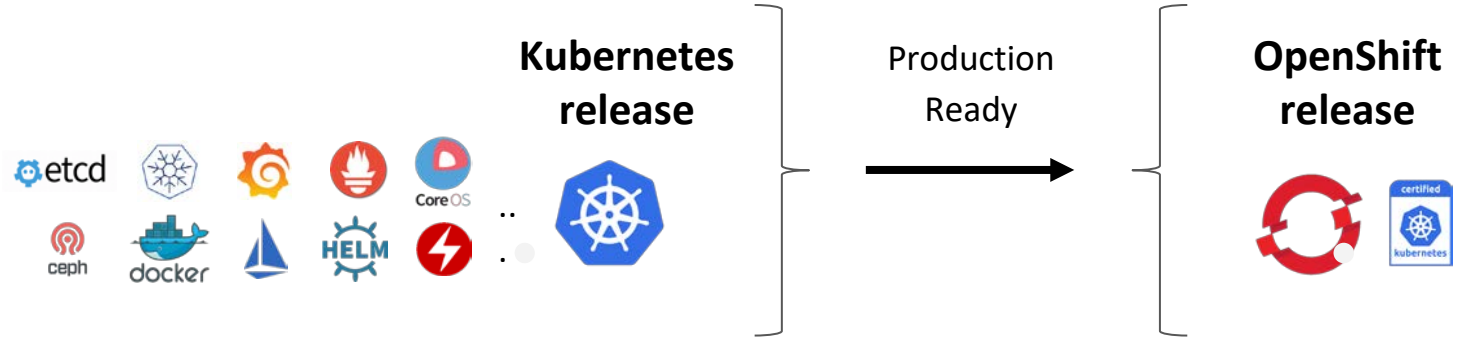
Red Hat is ***still*** Red Hat

From communities to enterprise



Red Hat

OpenShift is trusted enterprise Kubernetes



- Hundreds of defect and performance fixes
- 200+ validated integrations
- Certified container ecosystem
- 9-year enterprise life-cycle management
- Red Hat is a leading Kubernetes contributor since day 1

Red Hat

Supporting hybrid usage and buying patterns

A consistent platform no matter how or where you run

Start quickly, we manage it for you

Managed Red Hat OpenShift services

Red Hat OpenShift Service on AWS[1]	Azure Red Hat OpenShift	Red Hat OpenShift on IBM Cloud[1]	Red Hat OpenShift Dedicated[2]

You manage it, for control and flexibility

Self-managed Red Hat OpenShift

Red Hat OpenShift Platform Plus	Red Hat OpenShift Container Platform	Red Hat OpenShift Kubernetes Engine	On public cloud, or on-premises on physical or virtual infrastructure[3]

Source:
1 In preview as of 1/1/2021. Also available as Red Hat OpenShift Dedicated managed service running on user-supplied AWS infrastructure.
2 Red Hat managed service running on user-supplied GCP infrastructure
3 See docs.openshift.com for supported infrastructure options and configurations

Try Red Hat OpenShift

Start NOW at https://try.openshift.com

Developer Sandbox

Instant access to your own minimal, preconfigured environment for development and testing

Managed Services

Fully managed Red Hat® OpenShift®
Dedicated trial cluster with self-service sign-up and cluster provisioning in your cloud account

Self-managed

Self-managed on Red Hat OpenShift Container Platform, in the cloud, on your computer, or in your datacenter

Red Hat

Fragen?

BRZ

Fragen, die wir schon hatten…

Q: „Container? Wir haben schon virtuelle Server. Ist doch eh dasselbe“
A: Container sind keine VMs, aber auch sie bringen uns weg von der Physik

Q: „Ein Bug? Ich korrigiere das mal direkt im Container“
A: Alles ist automatisiert. Der Fehler wird vorher korrigiert, neu deployed und ein neuer Container gestartet.

Q: „Unsere App läuft schon in Docker. Könnt ihr das eh gleich übernehmen?“
A: Nein. Aber es ist leichter, weil Ihr schon versteht, worum es geht …

Q: Warum ist das so schwierig zu verstehen?
A: Weil es neu und anders ist.

Danke für Ihre Aufmerksamkeit.

> **Robert Bauer**
Produktmanager „BRZ Containerplattform“
Robert.Bauer@brz.gv.at

> **Johann Siegl**
Abteilungsleiter „Platform as a Service“
Johann.Siegl@brz.gv.at

> **Rupert Zarl**
Teamleiter „Cloudmanagement und Middleware“
Rupert.Zarl@brz.gv.at

@brz_gmbh
@Bundesrechenzentrum
@Bundesrechenzentrum
@Bundesrechenzentrum
@Bundesrechenzentrum