



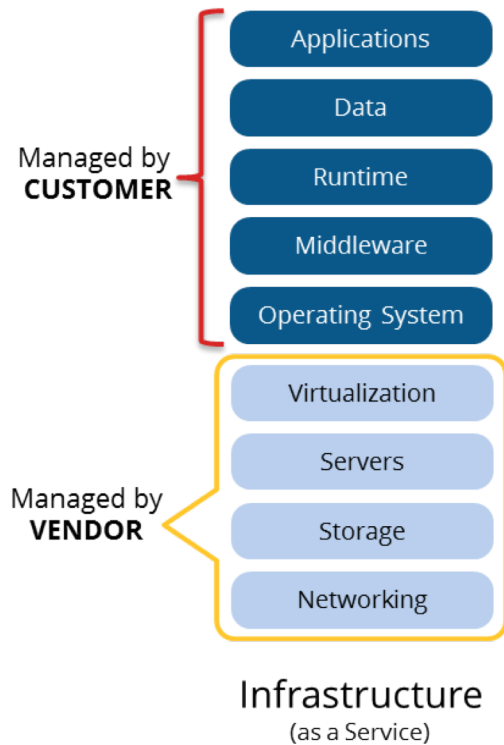
# **Vorlesung „Service and Cloud Computing“**

## **7. Cloud Computing – Platform as a Service (PaaS)**

**Dr.-Ing. Iris Braun**

## Cloud Computing – PaaS

- Platform as a Service – Wiederholung
- PaaS-Anbieter
  - Google App Engine
  - Amazon AWS Elastic Beanstalk
  - Windows Azure
  - Heroku
  - OpenShift
- Alternative Entwicklungen: Serverless



- **Entwicklungs-Plattformen (Application(a)PaaS)**
  - Plattform-Dienste, damit Kunden Ihre eigenen Anwendungen entwickeln, deployen und ausführen können
  - Ausführungsumgebung für verschiedenste Programmiersprachen
  - IDEs für Programmiersprachen, Kopplung an Code Repositories (Git), Continuous Delivery
- **Application Frameworks (Framework as a Service)**
  - anwendungsspezifische Application Frameworks können genutzt und angepasst werden
- **Integrations-Plattformen (iPaaS)**
  - Datenintegration intern und in der Cloud gehosteter Daten
- **Datenbank-Services (DBaaS)**
  - Bereitstellung von Datenbanken und anderen Datenspeichern
- **Business Process Management Services (BPM PaaS)**

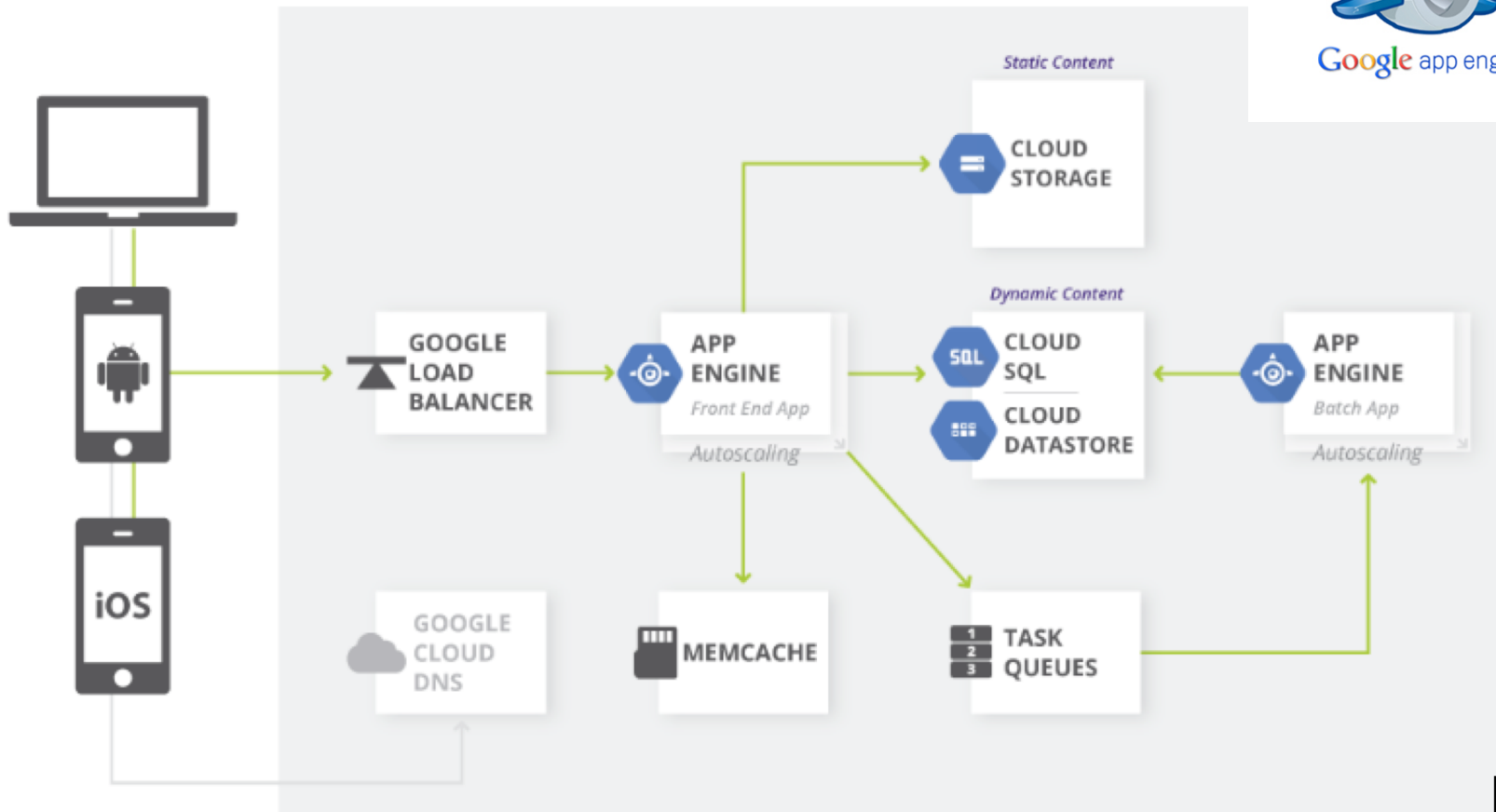
- Nutzer muss nur Anwendungscode entwickeln
- Konfiguration und Management der IaaS-Dienste übernimmt Cloud-Anbieter
- Loadbalancing, Autoscaling, Monitoring durch Plattform
- Multi-Tenant-Unterstützung (mehrere Kunden führen ihre Anwendungen auf gemeinsamer Infrastruktur aus, ohne sich gegenseitig zu beeinflussen)

- Bereitstellung von SQL- und NoSQL-Datenbanken, Überwachungs- und Diagnosetools, Message-Diensten wie Cloud Pub/Sub und einer Reihe von Entwicklungstools
- unterstützt standardmäßig Node.js, Java, Ruby, C#, Go, Python und PHP
- eigener benutzerdefinierter Softwarestack kann mittels Docker-Image bereitgestellt werden
- Versionsverwaltung für Anwendungen, Kopplung an Git
- umfasst Memcache- und Aufgabenwarteschlangen-Dienst
- integriertes Lastenausgleichsmodul (Google Load Balancer)
- Google Stackdriver - Tool zur Anwendungsdiagnose

## Bsp. Bereitstellen einer Webanwendung

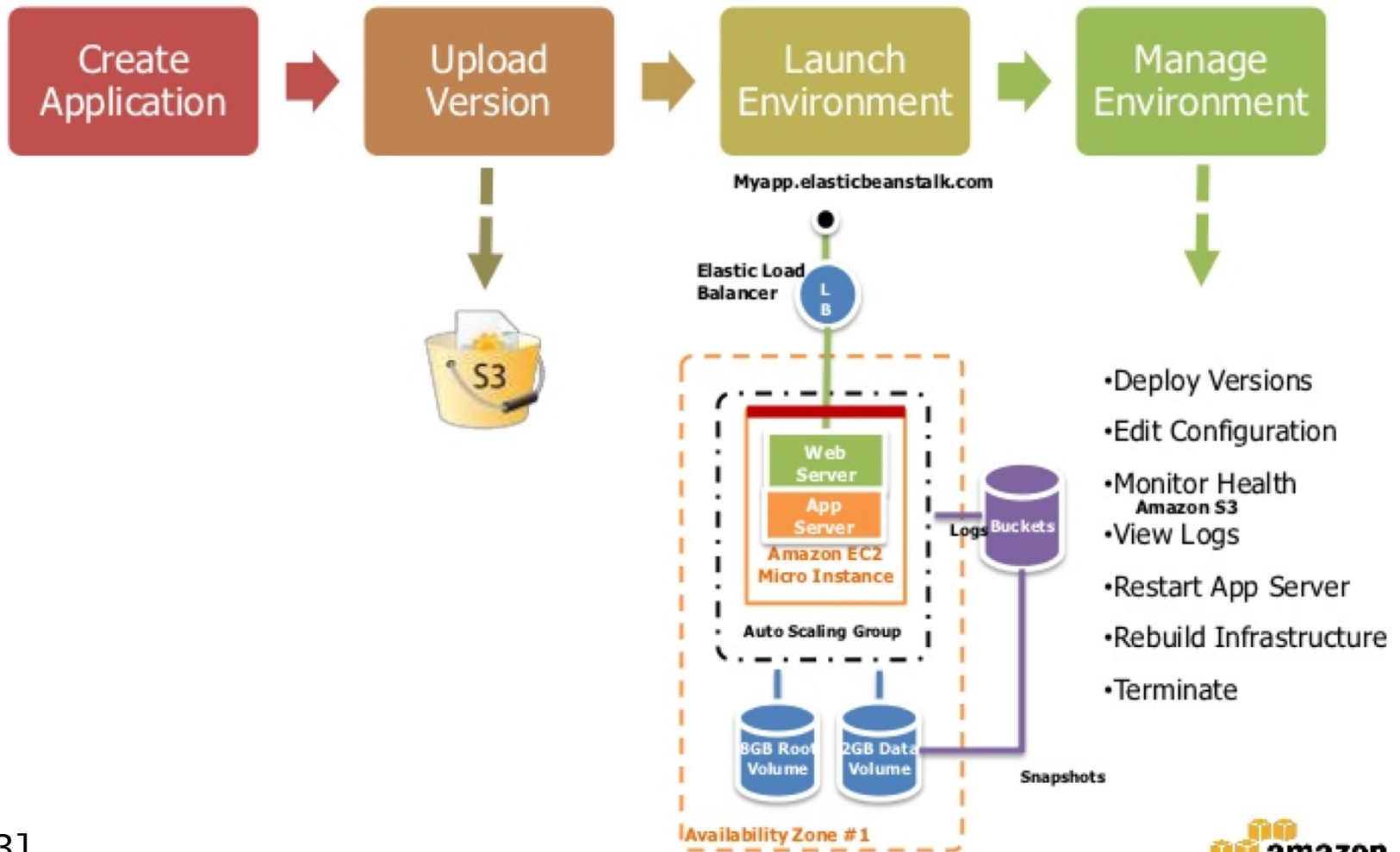


Google app engine



[2]

- Service zum Bereitstellen und Skalieren von Webanwendungen und -Services
- Unterstützte Programmiersprachen: Java, .NET, PHP, Node.js, Python, Ruby, Go
- Bereitstellung von Web-Servern wie Apache, Nginx, Passenger und IIS
- Support von Docker, um eigene Ausführungsumgebung zu nutzen
- Hochladen der Anwendung mit der AWS Management Console, aus einem GIT-Repository oder mittels integrierter Entwicklungsumgebung wie Eclipse oder Visual Studio
- nutzt darunterliegende Services zur Kapazitätsbereitstellung, Lastverteilung und automatischen Skalierung bis zur Statusüberwachung der Anwendung



[3]



## Platform Services

### Security & Management

- Portal
- Active Directory
- Multi-Factor Authentication
- Automation
- Key Vault
- Store / Marketplace
- VM Image Gallery & VM Depot

### Compute

- Cloud Services
- Service Fabric
- Batch
- Remote App

### Web and Mobile

- Web Apps
- API Apps
- API Management
- Mobile Apps
- Logic Apps
- Notification Hubs

### Developer Services

- Visual Studio
- Azure SDK
- Team Project
- Application Insights

### Hybrid Operations

- Azure AD Connect Health
- AD Privileged Identity Management
- Backup
- Operational Insights
- Import/Export
- Site Recovery
- StorSimple

### Integration

- Storage Queues
- Biztalk Services
- Hybrid Connections
- Service Bus

### Analytics & IoT

- HDInsight
- Machine Learning
- Data Factory
- Event Hubs
- Stream Analytics
- Mobile Engagement

### Data

- SQL Database
- SQL Data Warehouse
- Redis Cache
- Search
- DocumentDB
- Tables

### Media & CDN

- Media Services
- Content Delivery Network (CDN)

## Infrastructure Services

### Compute

- Virtual Machines
- Containers

### Storage

- BLOB Storage
- Azure Files
- Premium Storage

### Networking

- Virtual Network
- Load Balancer
- DNS
- Express Route
- Traffic Manager
- VPN Gateway
- Application Gateway

**Datacenter Infrastructure (24 Regions, 19 Online)**

- App Service unterstützt Windows und Linux-Plattformen
- Verschiedene App-Typen:
  - Web-Apps,
  - Mobile Apps,
  - Funktions-Apps,
  - API-Apps
  - Logik-Apps
- Web-Apps-Entwicklung mit .NET, Java, Node.js, PHP, Python oder Ruby
- Continuous Deployment mit Git, Team Foundation Server, GitHub und Visual Studio Team Services
- Integrierte Funktionen für eine automatische Skalierung und Lastenausgleich
- Hohe Verfügbarkeit mit automatischem Patching
- Azure Monitor überwacht Ausführung der Anwendungen



## Verwalten von Daten und API

- Relationale Datenbanken, DBaaS: SQL, PostgreSQL, MySQL
- NoSQL: Azure Cosmos DB

## Speicher:

- Blob Storage, Redis Cache

## AI und Cognitive Services:

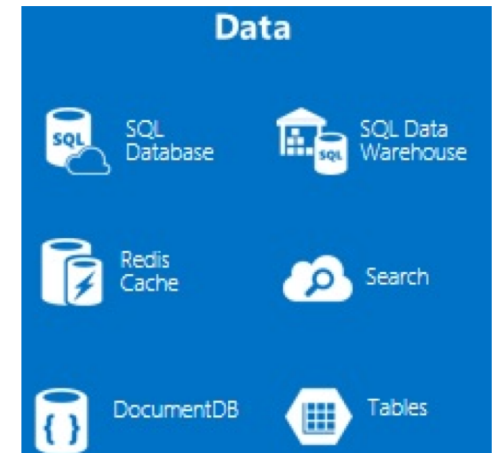
- Suchalgorithmen
- Machine Learning
- Gesichtserkennungs-API

## Analytics und IoT:

- Stream Processing, Event-Hubs

## DevOps:

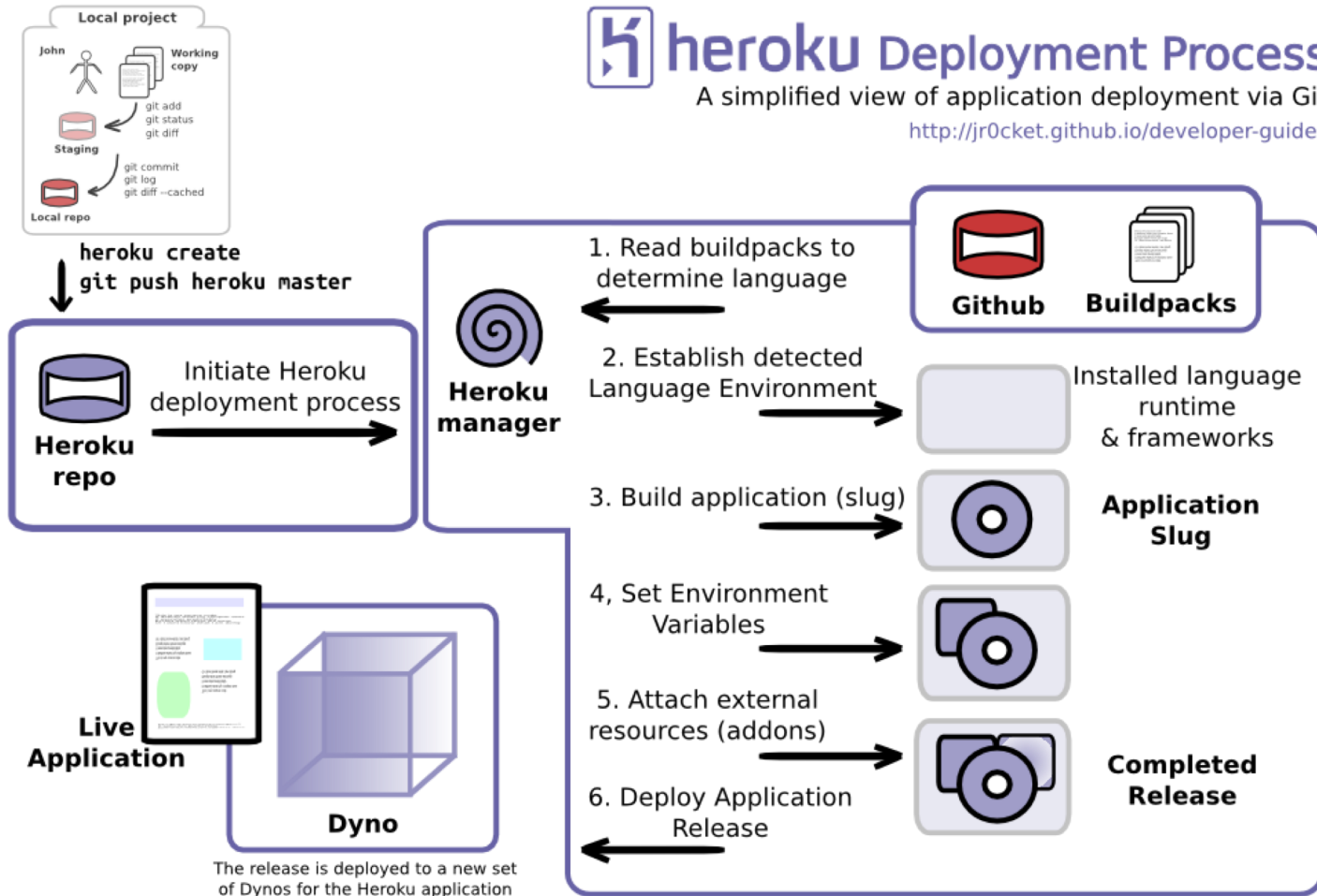
- Jenkins auf virtuellen Azure-Computern



## Cloud-Anwendungsplattform von Salesforce:

- Kombination aus den Ressourcen des zugrunde liegenden Betriebssystems, einer Laufzeitumgebung (je nach Anwendungssprache) und weiterer unterstützender Software in Form von Add-Ons.
- Unterstützt verschiedene Sprachen wie z.B. Ruby, Node.js, Java, Python, Scala, Go und PHP
- Heroku Anwendung als eine Sammlung von Quellcode, Ressourcdateien und Build-Skripten (z.B. Maven für Java).
- Dyno-Manifold - Ausführungsumgebung für Prozesse
- Was benötigt Heroku um eine Anwendung zum Laufen zu bringen?
  - Quellcode der Anwendung
  - Je nach verwendeter Sprache ein Dependency File, um die Abhängigkeiten aufzulösen
  - eine Datei in der definiert wird, welche Teile der Anwendungen ausführbar sind (Procfile)

[5]



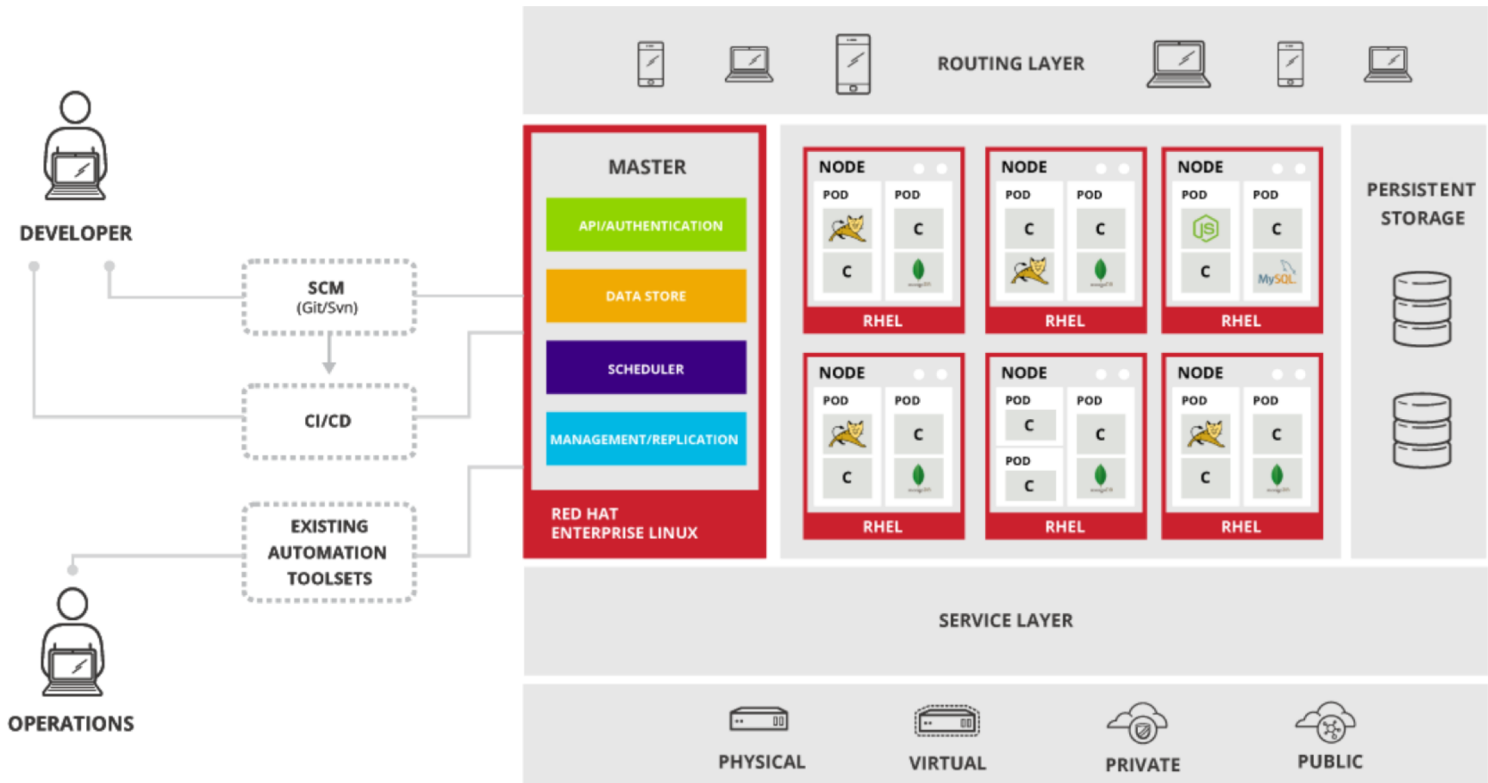
Heroku Developer Guides <https://devcenter.heroku.com/>

Building Scalable apps <http://12factor.net/>

[6]



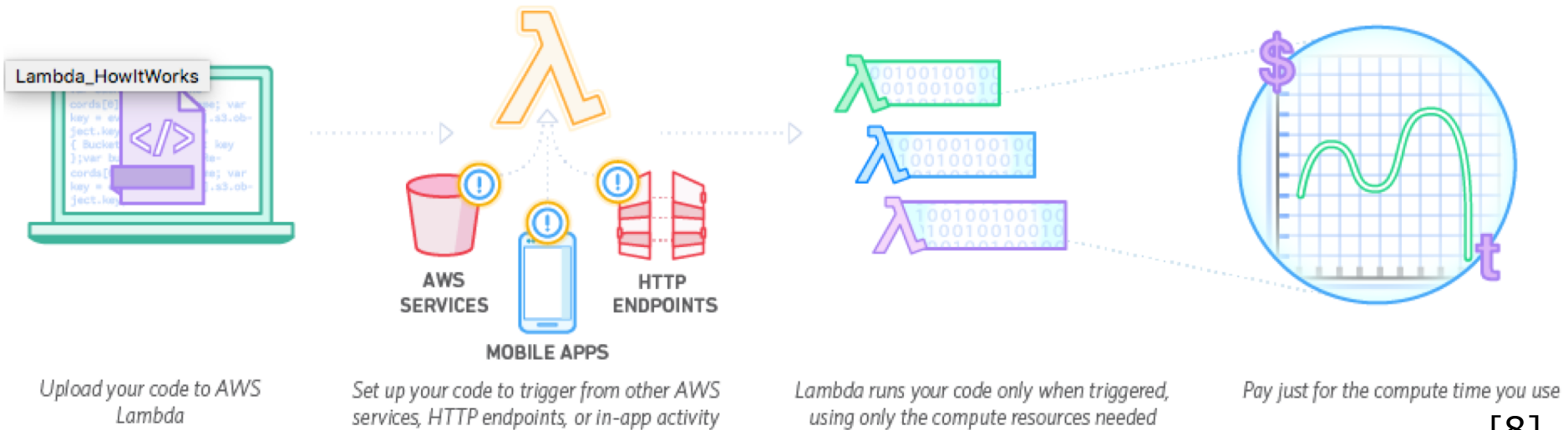
- Open Source Framework (2011 von Red Hat veröffentlicht)
- OpenShift Online, Dedicated und Container Platform
- Basiert auf Docker und Kubernetes



[7]

„Serverless“ bedeutet, dass sich der Entwickler nicht um die Bereitstellung und Konfiguration eines Servers zur Ausführung seiner App kümmern muss

- „Functions as a Service“ - Code wird nur ausgeführt, wenn er getriggert wird und nur unter Verwendung der benötigten Ressourcen (keine Kosten, wenn der Code nicht ausgeführt wird.)
- Erstellung und Betrieb von Echtzeit-Datenverarbeitungssystemen
- Bekanntester Vertreter: Amazon AWS Lambda



Upload your code to AWS  
Lambda

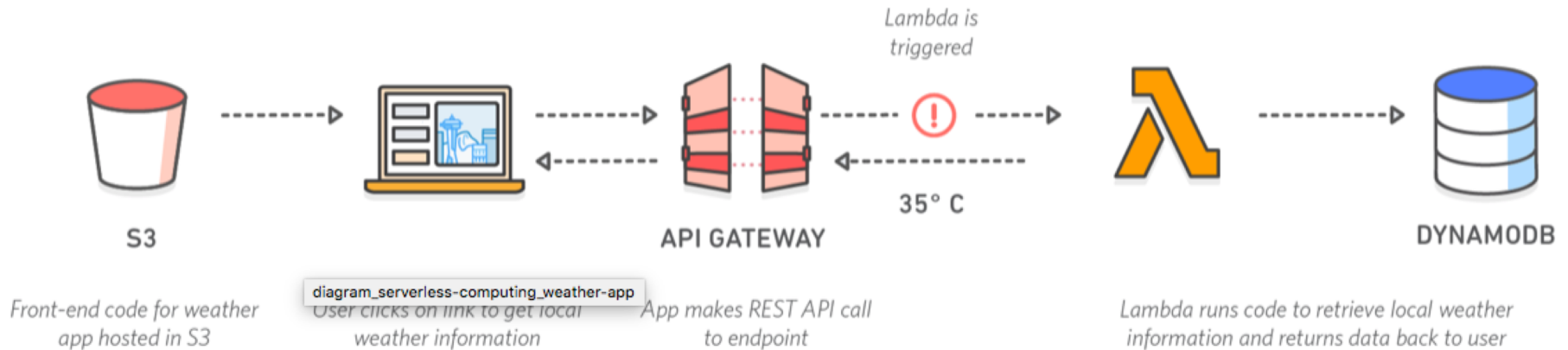
Set up your code to trigger from other AWS  
services, HTTP endpoints, or in-app activity

Lambda runs your code only when triggered,  
using only the compute resources needed

Pay just for the compute time you use

[8]

Example: Weather Application



Amazon API Gateway - Service zum Erstellen, Veröffentlichen, Warten, Überwachen und Sichern von APIs für Entwickler

Amazon Lambda - Ausführung von Funktionen („Functions as a Service“), die von beliebigen AWS Komponenten getriggert werden können (wie S3, DynamoDB, Kinesis, SNS und CloudWatch)

Amazon DynamoDB - flexibler NoSQL-Datenbankservice mit geringer Latenz

**DBaaS** = Datenbank-Software, die Service-Provider über eine Public Cloud als Managed Service bereitstellen

## **Vorteile:**

- vereinfachter und reduzierter Administrationsumfang
- Provider führt Aktualisierungen der Datenbank-Software durch
- verbesserte Skalierbarkeit trotz steigender Datenvolumina durch die höhere Flexibilität der Infrastruktur
- automatisierte Replikation und Synchronisation
- Provider garantiert den Nutzern im Rahmen von Service Level Agreements eine bestimmte Verfügbarkeit und Performance der Datenbanken

**Liste der bekanntesten Anbieter und Dienste: [9]**

- PaaS bieten Grundlage für die Entwicklung und Ausführung von Applikationen in der Cloud
- Setzen auf Infrastrukturdienste auf und stellen Load Balancing, Autoscaling, Monitoring etc. zur Verfügung
- Ausführungsumgebungen für viele Programmiersprachen
- Datenbanken und andere Speicherdienste integriert

### ***Vorteile:***

- Der Entwickler kann sich voll auf die Logik und den Inhalt der eigentlichen Anwendung konzentrieren. Serverkonfiguration oder andere Voraussetzungen für die optimale Lauffähigkeit werden über die Cloud geregelt.
- Cloud Applikationsentwicklung verringert die Kosten für Bereitstellung und Betrieb der Anwendung. Man stellt zusammen, was man braucht und bezahlt nur für das, was man auch verwendet.

1. <https://www.ironsidegroup.com/2015/06/03/driving-analytics-saas-paas-and-iaas-with-managed-services-the-difference-that-experts-make/>
2. <https://cloud.google.com/solutions/architecture/webapp>
3. <https://aws.amazon.com/de/elasticbeanstalk/>
4. <https://msdn.microsoft.com/de-de/magazine/mt573712.aspx>
5. <https://www.heroku.com>
6. John Stevenson: Heroku Overview for Developers,  
<http://jr0cket.co.uk/slides/heroku-for-developers.html#/>
7. <https://docs.openshift.org/latest/architecture/index.html>
8. <https://aws.amazon.com/de/lambda/>
9. <https://www.com-magazin.de/dl/1/4/6/8/8/7/9/Database-as-a-Service.pdf>