Infrastructure as Code

GSM - Tobias Schweisfurth
WiSe 18/19

Was ist Infrastruktur?

“Bezeichnet alle materiellen und immateriellen Güter, die den Betrieb von (Anwendungs-) Software ermöglichen.”

Infrastructure as Code

Warum Infrastructure as Code?

- Anpassungsfähigkeit an die sich immer weiterentwickelnde Cloud- und Automatisierungs-Landschaft
- Konsistente und zuverlässige Infrastruktur
- Vermeidung von “Shadow IT”

- Nimmt Angst vor Veränderung der Infrastruktur

Was ist Infrastructure as Code?

- Infrastruktur-Automatisierung basierend auf Praktiken der Softwareentwicklung

- Die “Vercodung” von Infrastruktur

```
server: dbnode
    base_image: centos72
    chef_role: dbnode
    network_segment: prod_db
    allowed_inbound:
        from_segment: prod_app
        port: 1521
    allowed_inbound:
        from_segment: admin
        port: 22
```

Abbildung 1: Beispiel einer Definitionsdatei geschrieben in einer DSL

```
stages:
    - test
    - deploy

before_script:
    - pip install -r requirements.txt
    - pip install -q ansible

generate_config:
    stage: test
    script:
        - ansible-playbook pb.generate.config.yaml

deploy_config:
    stage: deploy
    script:
        - ansible-playbook pb.conf.all.commit.yaml
```

Abbildung 2: Beispiel einer Gitlab-CI, welche über Ansible eine Infrastruktur aufbaut und ausliefert

Ziele von Infrastructure as Code

- Support von Veränderung über kontinuierliche Updates

- Fokus auf Business-Logik anstatt Infrastruktur
- Unabhängige Entwicklerteams
- Einfache und schnell Regeneration bei Fehlern

Prinzipien von Infrastructure as Code

- Reproduzierbarkeit
  - bei Skalierung
  - bei Recovery
- “Wegwerfbarkeit”
  - Systeme sollten zerstörbar, ersetzbar, bewegbar und in der Größe veränderbar sein
- Wiederholbarkeit

Praktiken aus der Softwareentwicklung übertragen auf Infrastruktur

Nutzung von Definitionsdateien

- Grundpfeiler von Infrastructure as Code
- Definition spezifiziert Konfiguration von Infrastruktur(-elementen)
- Werden als Textdateien organisiert (YAML, JSON, XML, DSL)

- Definition wird nicht im Tool selbst angepasst, sondern immer über eine Definitionsdatei

Versionierung in VCS

- Anpassungen der Infrastruktur werden durch Änderungen, die ins VCS committed wurden, angestoßen.

- Vorteile:
  - Historie aller gemachten Änderungen
  - Rollback zu einem vorherigen Zeitpunkt
  - Tagging und Versionsnummern
  - Sichtbarkeit von Änderungen
  - Pipelines: Automatisiertes Ausführungen von Aktionen basierend auf Änderungen im VCS

CI/CD und Testing

- Änderungen können kontinuierlich integriert und validiert werden (CI)
- Änderungen können sicher in die (Produktions-)Umgebungen integriert werden (CD)

- Es findet automatisiertes Testing statt
- Arbeiten mit verschiedenen Entwicklungsumgebungen, ggf. auch Feature-Branches (s. Git-Flow)

Inkrementelle Erweiterung der Infrastruktur

- “Agiles Mindset”: Jedes Inkrement macht das Produkt etwas besser
  - hier: inkrementelle Erweiterung/Anpassung der Infrastruktur

- Kleine anstatt von großen Änderungen
  - Kleine Änderungen sind einfacher zu identifizieren (bei Problemen)
  - Kleine Änderungen sind einfacher rückgängig zu machen
  - Kleine Änderungen sind einfacher zu testen und sicherzustellen, dass sie funktionieren
  - Psychologisch: Fertigstellen von Aufgaben ist motivierend, auch wenn sie klein sind

Continuous Configuration Automation (CCA)

CCA

- Ermöglicht die Automatisierung von Konfigurationseinstellungen sowohl on-premise als auch in Cloud-Umgebungen
- Konfiguration und Provisionierung der Infrastruktur wird von Tasks in CCA-Tools durchgeführt
    - Erweiterung des Infrastructure as Code Ansatzes: Änderungen werden in Form von Code beschrieben (der Task) und Tasks werden vom CCA-Tool abgearbeitet
- Management-Tools bewirken eine hohe Transparenz
    - Sichtbarkeit wird durch UI oder Schnittstelle ermöglicht
    - Konsistenz wird von CCA-Tools erzwungen

Möglichkeiten basierend auf Infrastructure as Code

Job-basierte Infrastruktur / Cluster-as-a-Service

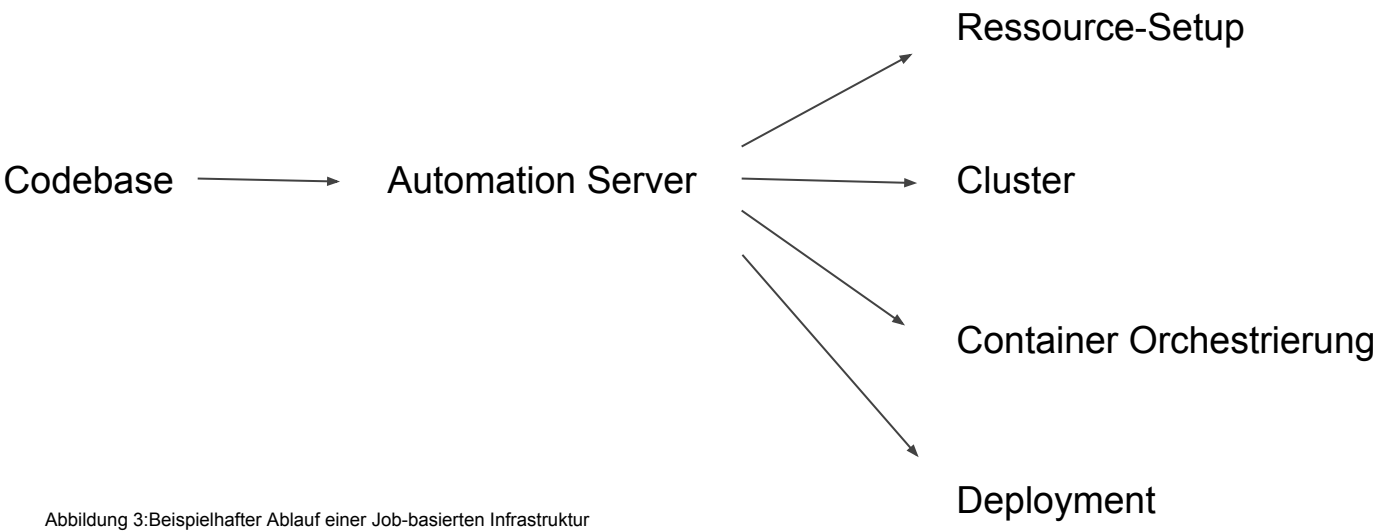
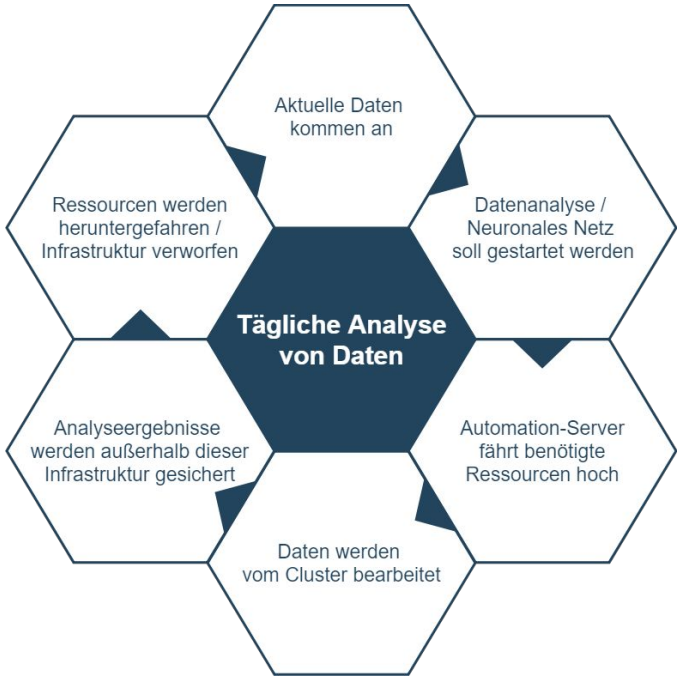


Abbildung 3:Beispielhafter Ablauf einer Job-basierten Infrastruktur

Einweg-Cluster

- Job-basierte Cluster auf die Spitze getrieben
- Wenn der Job beendet wurde, kann das Cluster verworfen werden
- Geeignet für einmalige Aufgaben, Tests oder Proof-of-Concepts



Gedankenexperiment

Abbildung 4: Zyklischer Ablauf eines ausgedachten Anwendungsfalles für ein Cluster-as-a-Service Szenario

Weitere Planung

Cluster-as-a-Service

- Infrastrukturelemente können über Infrastructure as Code gemanaged werden.
- Weitergedacht: Management von Zusammenschlüssen von Infrastrukturelementen
  - z.B. Cluster

- Cluster auf Knopfdruck (vielleicht aus Pipeline)
- Cluster ist direkt infrastrukturell auf die Bedürfnisse des Codes / der Anwendung zugeschnitten

Serverless

- Welche Auswirkungen können Serverless und FaaS auf Infrastructure as Code haben?

- Möglicher Prioritätenwechsel bei Infrastructure as Code
  - Weniger mit Aufsetzen der Infrastruktur beschäftigen
  - Mehr mit Verbinden der Infrastrukturen beschäftigen

Wissenschaft

Konferenzen und wichtige Persönlichkeiten

Persönlichkeiten

- Andrew Clay Shafer und Patrick Debois: Initiatoren der DevOps-Bewegung
- Kief Morris (Autor von “Infrastructure as Code - Managing Servers in the Cloud”)

Konferenzen

- Agile-Development Konferenzen
  - Fokus auf Development und nicht Management
- DevOpsDays-Konferenzen (JUN 26 - 28, 2019 in Amsterdam)
- KubeCon, CloudNativeCon und International Conference on Software Engineering

Abbildungsverzeichnis

Quellenverzeichnis

- Matej Artač, Tadej Borovšak, Elisabetta Di Nitto, Michele Guerriero, and Damian Andrew Tamburri. 2017. DevOps: introducing infrastructure-as-code.
- Akond Rahman, Jonathan Stallings, and Laurie Williams. 2018. Defect prediction metrics for infrastructure as code scripts in DevOps.
- Morris, Kief: *Infrastructure as Code. Managing Servers In The Cloud*. Erste Auflage. O'Reilly
- Sayers, D. (2017). Configuration Management vs. Application Release Automation. DevOps.com. Quelle: https://devops.com/configuration-management-vs-application-release-automation/ [Letzter Zugriff: 5. Nov. 2018]
- Zitat 1. Folie. Übernommen aus Patig, S. (2012). IT-Infrastruktur. Enzyklopädie der Wirtschaftsinformatik Online-Lexikon. Quelle: http://www.enzyklopaedie-der-wirtschaftsinformatik.de/wi-enzyklopaedie/lexikon/daten-wissen/Informationsmanagement/IT-Infrastruktur [Letzter Zugriff: 5. Nov. 2018]

Vielen Dank für die Aufmerksamkeit