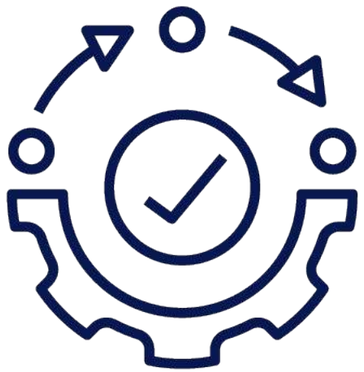
CfgMgmt vs Workflows vs Orchestration

SLAC 2023

Martin Alfke
ma@betadots.de

SLAC 2023
23.–25. Mai 2023 | Berlin



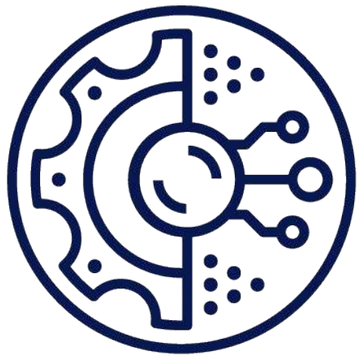
Martin Alfke

CEO betadots

- Consulting, Training and Development
- ~~DevOps~~ Platform Engineering
- tuxmea (Twitter, GitHub, Slack)

SLAC 2023
23.–25. Mai 2023 | Berlin

Jedes Problem braucht eine Lösung



- Konfigurations-Management
- Workflows/Ablaufsteuerung
- Orchestrierung

Konfigurations-Management

Secure Linux Administration Conference - Mai 2023

Früher war alles besser… Oder?

Manuelles Setup von Systemen:

- Ungeplante Änderungen
- Ausfälle
- Neues Setup für neue Produkte
- Nicht nachvollziehbare Änderungen
- Unvollständige Änderungen

Früher war alles besser… Oder?

Wer hat eines der folgenden schon gesehen, mitbekommen oder gemacht?

- Jeder Admin hat seine eigenen Skripte
- Eine Änderung konnte nicht ausgerollt werden, weil ein System in Hardware Wartung war
- Ein Server wurde beim Deployment “vergessen”
- Ein Deployment MUSS über Nacht erfolgen

Das moderne “früher”

Konfigurations-Management

Bcfg2 und CFEngine verwendeten ein Client-Server-Modell und implementierten die erwünschten Änderungen.

Puppet hat die Declarative State Configuration (DSC) eingeführt und den Change in das Versions Control System überführt.

Ansible hat anstelle des Client-Server-Modells eine SSH basierte Push Implementing vorgenommen.

Nächstes Problem - Security

Warum muss man sich auf einen Server mit SSH/WinRM/RDP verbinden?

- Log File Analyse? --> zentrales Logging (ELK, Splunk, Prometheus)
- App Debugging aktivieren? --> CfgMgmt (Puppet, Ansible)

Was wäre, wenn man SSH/WinRM/RDP abschaltet?

Die "neue" Gegenwart

Microservices erlauben schneller Entwicklung
Container erlauben Trennung OS <-> Applikation

Container Betrieb muss sichergestellt werden.
Kubernetes (mit allen Stärken und Schwächen) hat sich als de-facto Standard etabliert.

Die Gegenwart

Viele Tools nutzen das Prinzip von “Infrastructure as Code”.

Code wird in einem Versions-Kontroll-System abgelegt und getestet.

Einige nennen dies DevOps, andere nennen es Platform Engineering.

DevOps: Entwickler bauen und betrieben ihre Anwendungen
Platform Engineering: Aufbau und Betrieb der Infrastruktur für DevOps Automatisierung

Nächstes Problem

Mit Declarative State Configuration (DSC) wird der gewünschte finale Zustand eines Systems beschrieben.

Aber:

- Wie geht man mit widersprüchlichen Definitionen um? (Service stop, …, Service start)
- Wie geht man mit Aktivitäten um, die nicht dauernd ausgeführt werden sollen?

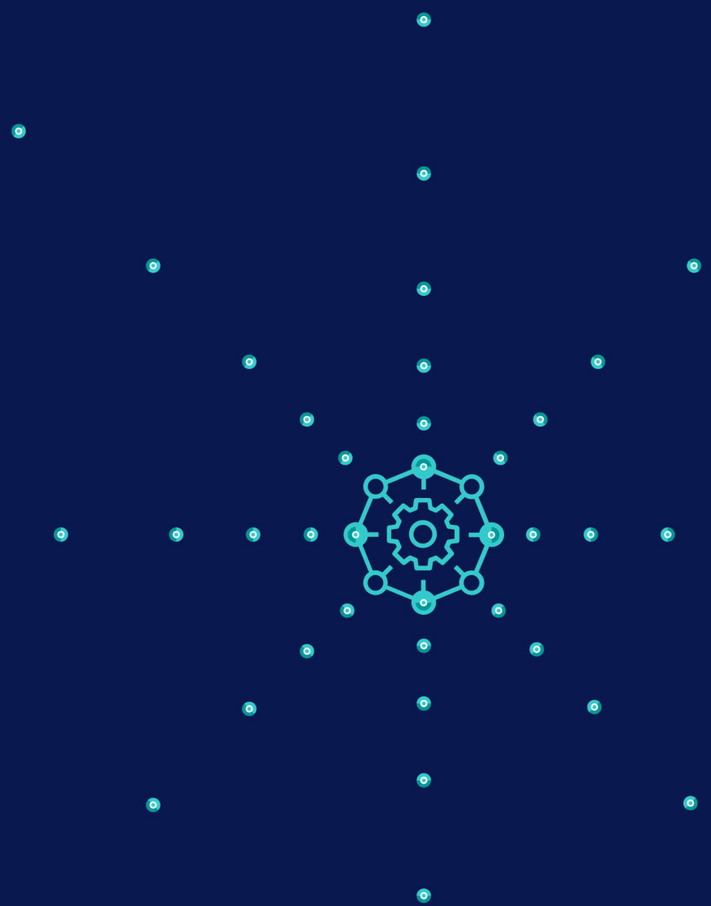
Und wie ist das bei Containern?

Auch im Umfeld von Containern benötigt man CfgMgmt.

Aber wo?

Ein Container beinhaltet eine Anwendung und die Konfiguration.
Anpassungen meistens über Environment Variablen.
Welche Variable wird wo in welcher Config Datei genutzt?

-> Das Dockerfile ist das CfgMfgmt.



Workflows

Ablaufsteuerung

SLAC 2023
23.–25. Mai 2023 | Berlin

Secure Linux Administration Conference - Mai 2023

Was ist eine Ablaufsteuerung?

Alles, was man nicht im Finalen Zustand beschreiben kann, sondern über Skripte oder Kommandos durchgeführt wird, ist eine Ablaufsteuerung.

oder

Jedesmal, wenn man auf einem Server lokal etwas tut, braucht man eine Steuerung, die Aktivitäten protokolliert.

Ablaufsteuerung und DSC

Declarative State Configuration (DSC) beschreibt den gewünschten Finalen Zustand eines Systems.

Jede DSC Lösung vergleicht den Ist-Zustand mit dem gewünschten Zustand.

DSC verbietet widersprüchliche Einstellungen (Service stop, …, Service start).

Implementierung Ablaufsteuerung

Ansible Anwender fügen die Kommandos zu den Playbooks hinzu und lassen diese “on-demand” via SSH laufen.

Puppet Agent arbeitet nach dem “Pull” Prinzip.

Puppet Bolt Tasks arbeiten nach “Push” Prinzip.

Puppet Anwender hinterlegen Skripte (Bash, Python, Ruby, Perl, PowerShell) und eine JSON Datei, in welcher die Parameter und erwarteten Date Typen hinterlegt sind.

Implementierung Ablaufsteuerung ohne SSH/WinRM

Puppet: Puppet Enterprise oder Choria (choria.io) verwenden.
Ansible: Tower oder AWX
Saltstack: Minions

Es wird eine Message Queue anstelle von SSH/WinRM verwendet.

Vorteil: SSH/WinRM/RDP kann abgeschaltet werden!
DevSecOps!

Implementierung Ablaufsteuerung ohne SSH/WinRM

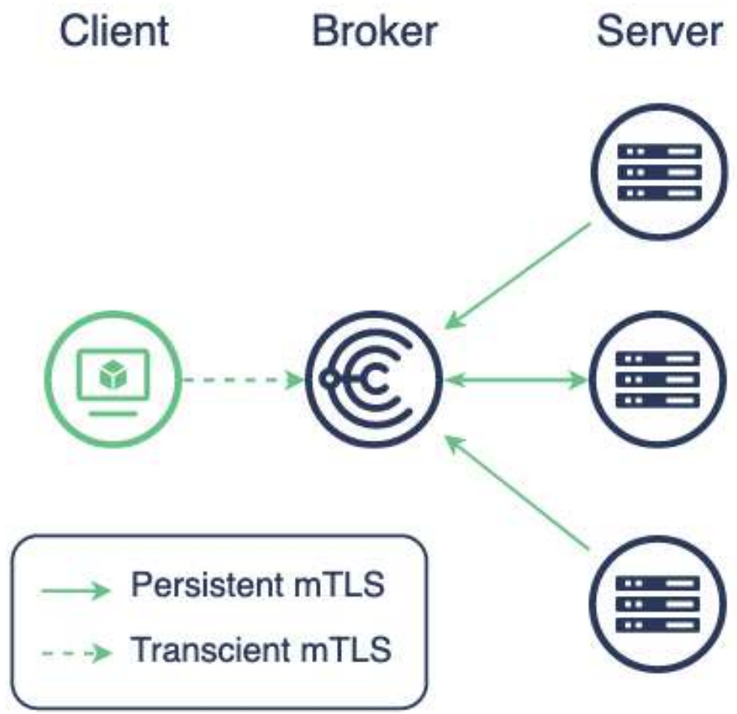


Image: choria.io

Beispiele für Ablaufsteuerung

Aktivitäten, die nicht regelmässig durchgeführt werden, oder die im Widerspruch zu DCS stehen:

- Während eines Updates muss der Cache bei gestoppter Anwendung gelöscht werden
- Aktuelle Konfigurationsdatei(en) als CSV liefern
- …
- Anonymisierter DB Dump von Prod nach Dev??

Aber

- Wie geht man mit Serverübergreifenden Abhängigkeiten um?
- Wie kann man sicherstellen, dass eine Applikation in der richtigen Reihenfolge ausgerollt wird (DB, Middleware, Frontend)?
- Wie kann man Marketing ermöglichen, die neue Version der Webseite selber auszurollen?

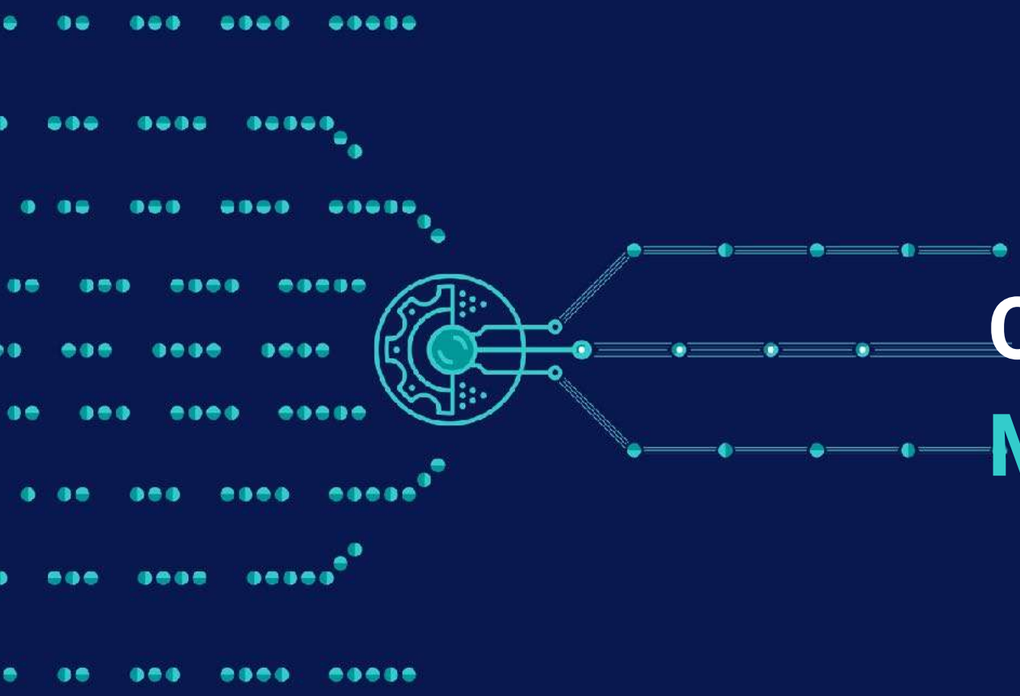
Und wie ist das bei Containern?

Allgemein braucht man bei Containern kein Workflow Management - zumindest nicht im Container.

Maintenance vorbereiten (Node prune)
Anwendungen in richtiger Reihenfolge:

- Docker Compose
- Helm

beladels



Orchestrierung Management

SLAC 2023

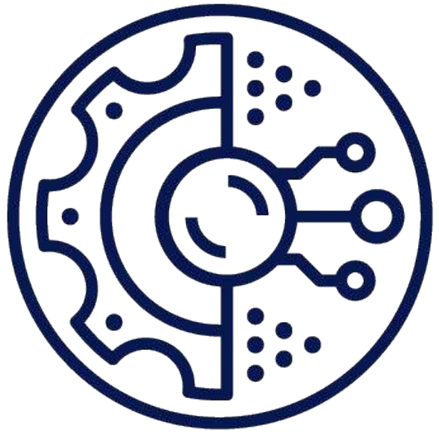
23.–25. Mai 2023 | Berlin

Secure Linux Administration Conference - Mai 2023

Was ist Orchestrierung?

Aktivitäten, die in einer bestimmten Reihenfolge auf einem System oder Systemübergreifend durchgeführt werden müssen.

Beispiel: Ausrollen einer neuen Umgebung:
- VMs
- Basis Konfiguration
- Applikations Deployment

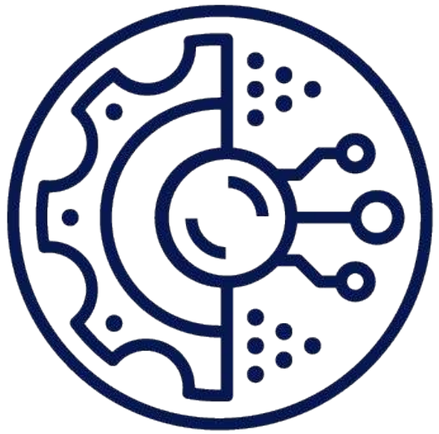


Was ist Orchestrierung?

Aktivitäten, die in einer bestimmten Reihenfolge auf einem System oder Systemübergreifend durchgeführt werden müssen.

Beispiel: Ausrollen einer neuen Anwendung mit Abhängigkeiten:

- Start Service A auf Server A
- Start Service B auf Server B
- Restart Service A auf Server A



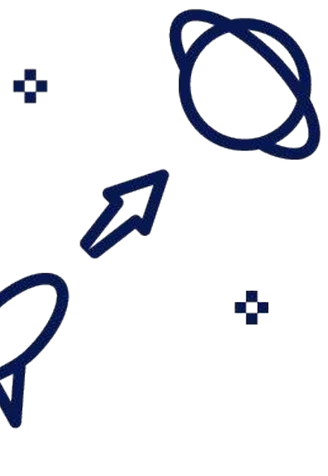
Und wie ist das bei Containern?

Orchestration ist essentiell, auch bei Containern.

Deployment und Verteilung, Aufbau und Abbau

- Docker Swarm
- Kubernetes

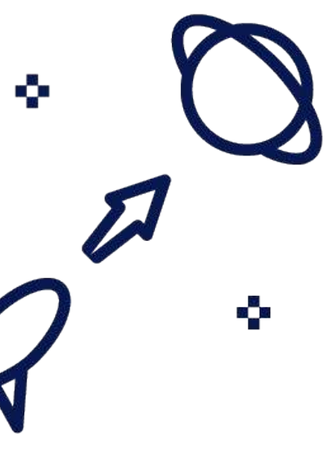
Zusammenfassung



CfgMgmt vs Ablaufsteuerung vs Orchestrierung

CfgMgmt hilft Organisationen bei
- Aufbau und Verwaltung von Plattformen
- Welche einfach deploybar (TTM)
- Welche einfach wieder herstellbar sind (MTTR) und
- Welche einfach an neue Anforderungen angepasst werden können.

CfgMgmt reduziert Aufwand und Zeit

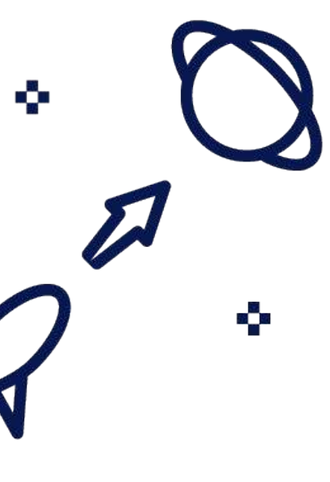


CfgMgmt vs Ablaufsteuerung vs Orchestrierung

Ablaufsteuerung wird benötigt,

- Um einzelne Aufgaben zu automatisieren
- Wenn man Aktivitäten unregelmässig durchführen muss
- Wenn Abläufe im Widerspruch zu DSC stehen

SLAC 2023
23.–25. Mai 2023 | Berlin
Secure Linux Administration Conference - Mai 2023

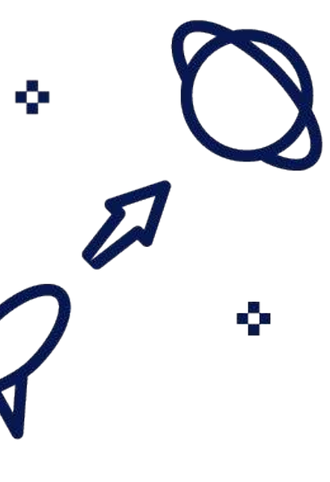


CfgMgmt vs Ablaufsteuerung vs Orchestrierung

Orchestrierung wird benötigt, um die Gesamtansicht einer Plattform abzubilden.

Ausrollen einer Three-Tier Anwendung in richtiger Reihenfolge.
Patchen eines Clusters ohne Downtime (Passiver Node, Switch, 2. Node)

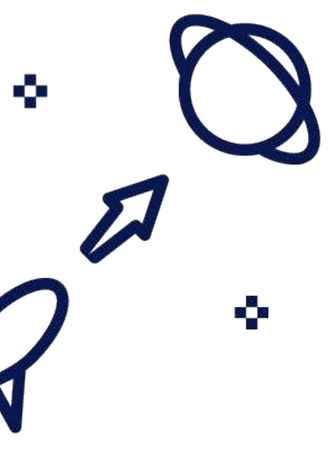
SLAC 2023
23.–25. Mai 2023 | Berlin
Secure Linux Administration Conference - Mai 2023



CfgMgmt mit Ablaufsteuerung und Orchestrierung

Die aktuellen Configuration Management Lösungen implementieren dies.

Bei Ansible ist die Orchestrierung durch das SSH Modell eingebaut.
Bei Puppet nutzt man Tasks und Plans.
Bei Containern wird CfgMimt beim Bau des Container berücksichtigt.

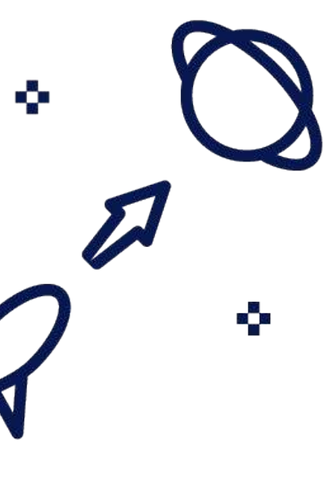


CfgMgmt mit Ablaufsteuerung und Orchestrierung

Die Verwendung eines Message Queue basierten Systems erlaubt das Deaktivieren von SSH/WinRM.

Eine WebUI/CLI mir RBAC erlaubt zentrale Steuerung und Logging und Rechteverwaltung.

SLAC 2023
23.–25. Mai 2023 | Berlin

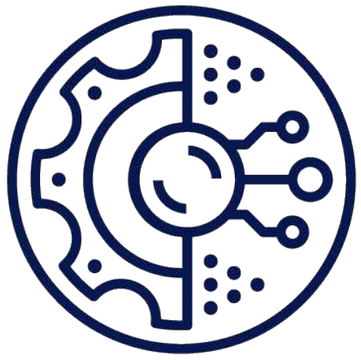


CfgMgmt mit Ablaufsteuerung und Orchestrierung

Eine WebUI mir RBAC erlaubt zentrale Steuerung und Logging.

Applikationsverantwortliche können eine neue Version einer Anwendung ausrollen, ohne das Ops Team bemühen zu müssen.

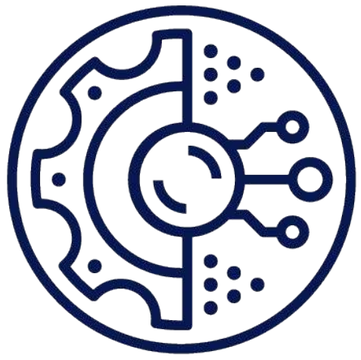
SLAC 2023
23.–25. Mai 2023 | Berlin
Secure Linux Administration Conference - Mai 2023



Jedes Problem ~~braucht~~ hat eine Lösung

- Konfigurations-Management
- Workflows/Ablaufsteuerung
- Orchestrierung

Jede Lösung muss versionierbar und automatisierbar sein.
Über definierte API Schnittstellen müssen Anwender Aktionen durchführen können.
Alle Aktionen müssen im zentralen Logging einsehbar sein.
Für User und Gruppen muss ein Zugriffsystem/-schutz vorhanden sein.



Jedes Problem ~~braucht~~ hat eine Lösung

- Security und Compliance
- Anpassungen an die Anwendungen
- Anpassungen an die Infrastruktur

Deaktivieren von Remote Zugängen - ein System auf dem man sich einloggen muss ist per Definition defekt.

Testen, testen, testen

Was nicht unter Versionskontrolle ist, hat auf einem System nichts verloren.

CfgMgmt vs Workflows vs Orchestration

SLAC 2023

Danke schön!

ma@betadots.de

SLAC 2023
23.–25. Mai 2023 | Berlin