Zabbix und Cloud Native Monitoring

Christian Anton

enthus

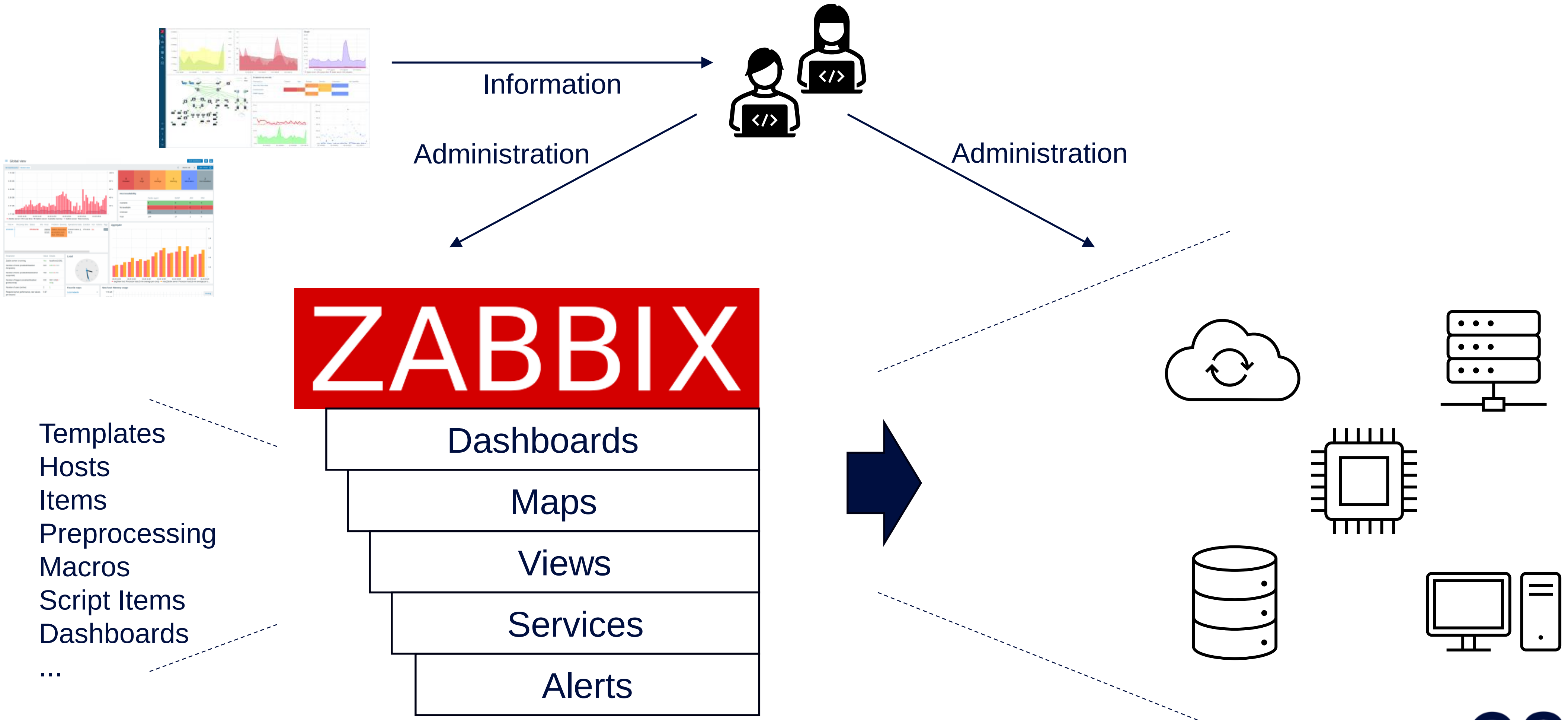
Stärken von Zabbix

- Datensammlungsmethoden
  HTTP, SNMP, MQTT, Scripts,OS Agents, Logs, ...
- Strukturierung von Daten
  Hosts, Items Tags, Low Level Discoveries, Filter, Datentypen
- Verwaltung von Monitoring-Punkten
  Templates
- Visualisierung, Aufbereitung
  Dashboards, Services, Maps, ...
- Automatisierung
  Network Discovery, Active Agent Auto Registration
- Flexibilität
  Frontend Modules, Agent Plugins, User Parameters, ...

All-In-One Monitoring Platform

- Server
- Netzwerkgeräte
- Anwendungen
- APIs
- IoT
- Embedded
- ...

Monitoring Konfigurationszyklus



Information

Administration

Administration

ZABBIX

Dashboards

Maps

Views

Services

Alerts

Templates
Hosts
Items
Preprocessing
Macros
Script Items
Dashboards
...

enthus

Was ist 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 toil.*

CLOUD NATIVE COMPUTING FOUNDATION

Was ist Cloud Native?

- Ansatz für Entwicklung und Bereitstellung von Anwendungen
- Auf den Betrieb in Cloud(artigen) Umgebungen optimiert
- Hochgradig Automatisiert
- DevOps Arbeitsweise

- Container
- Microservices
- Kubernetes
- PaaS Services & Plattformen
- Deklarative APIs
- Immutable Infrastructures

enthus

I NEED TO KNOW WHY MOVING OUR APP TO THE CLOUD DIDN'T AUTOMATICALLY SOLVE ALL OUR PROBLEMS.
YOU WOULDN'T LET ME RE-ARCHITECT THE APP TO BE CLOUD-NATIVE.
JUST PUT IT IN CONTAINERS.
YOU CAN'T SOLVE A PROBLEM JUST BY SAYING TECHY THINGS.
KUBERNETES.
Dilbert.com @ScottAdamsSays
11-08-17 © 2017 Scott Adams, Inc./Dist. by Andrews McMeel

~~Box Moving / Lift-and-Shift~~

gitops

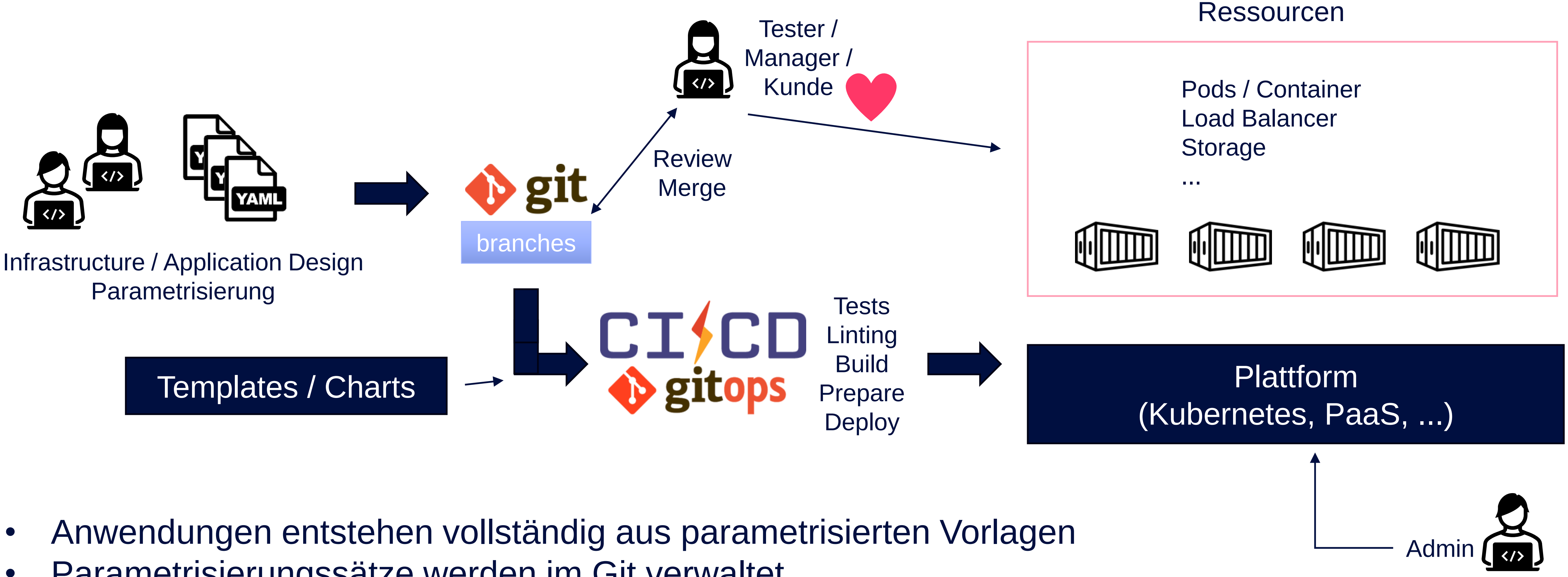
- Forward-Only Automation
- Infrastructure as Code

→

- Wiederholbar
- maschinenlesbar
- selbst-dokumentiert
- nachvollziehbar

enthus

Wie funktioniert das?



Infrastructure / Application Design
Parametrisierung

git
branches

Tester /
Manager /
Kunde

Review
Merge

Templates / Charts

CI/CD
gitops

Tests
Linting
Build
Prepare
Deploy

Ressourcen

Pods / Container
Load Balancer
Storage
...

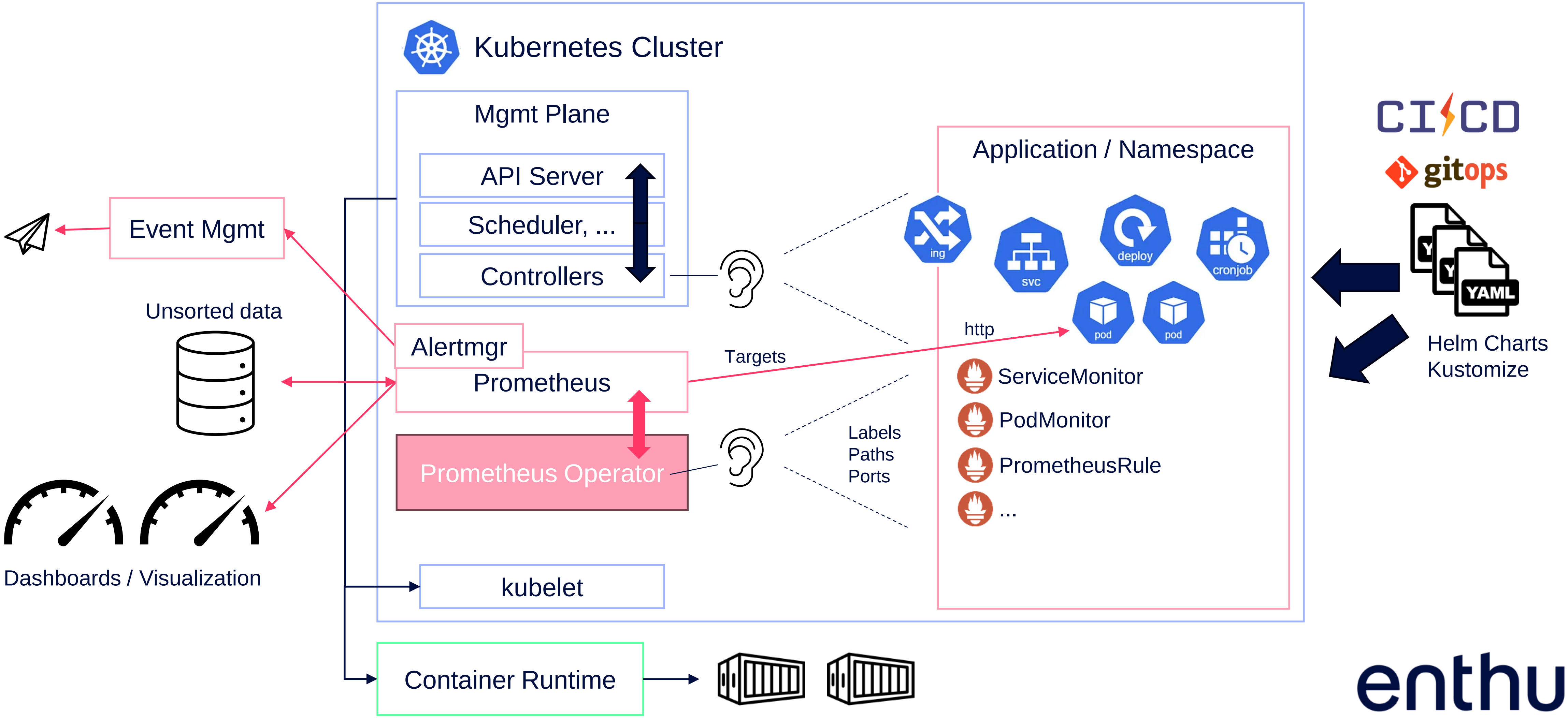
Plattform
(Kubernetes, PaaS, ...)

Admin

- Anwendungen entstehen vollständig aus parametrisierten Vorlagen
- Parametrisierungssätze werden im Git verwaltet
- DevOps Team hat die Verantwortung für alle Ressourcen der Anwendung
- Klassischer IT Admin verantwortet die Plattform (on prem)

enthus

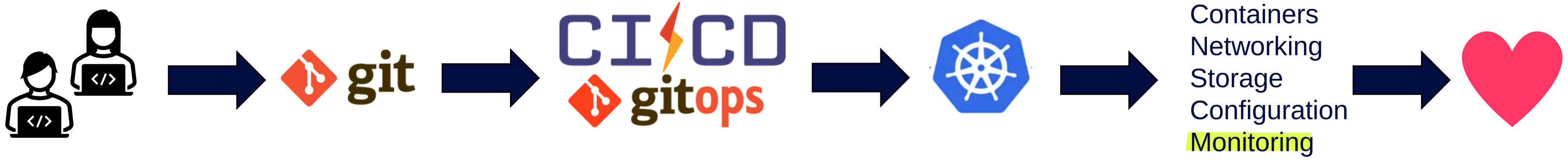
Workload Monitoring (in Kubernetes)



enthus

Zusammenfassung

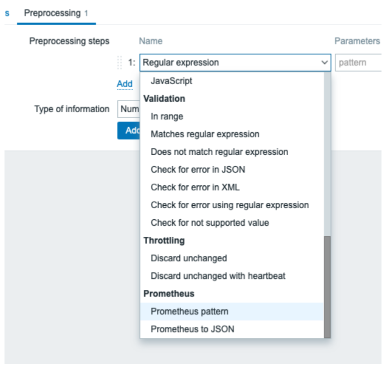
- One-Way Automation auch für Monitoring (Operator Paradigma)
- Etablierter Standard
- Monitoring-Stack bestehend aus Microservices „non-all-in-one“
- Weiterverarbeitung der Daten durch 3rd Party Tools
- Monitoring-Intelligenz in der Anatomie der Anwendung enthalten
- Gleiches Konzept auch für Visualisierung (Dashboards)
- Verantwortung für Anwendung = Verantwortung für Monitoring



Containers
Networking
Storage
Configuration
Monitoring

Und Zabbix?

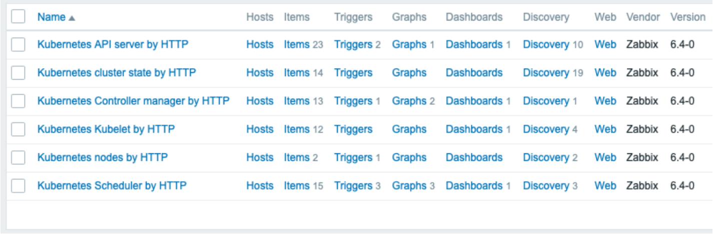
- Abfrage von Metriken von Prometheus Exportern / Endpoints
  - Filterung
  - Aggregation Functions
  - Prometheus selbst besitzt einen `/federate` Endpoint
- Dynamik anhand Low Level Discoveries und Script Items (JavaScript)
- cloudnative Anwendungen sind auch nur Anwendungen
  - Zabbix Templates, z. B. für MySQL, ...
- Zabbix bietet Kubernetes Monitoring Templates
- Es gibt keinen Zabbix Monitoring Operator
- In Zabbix steckt die Monitoring Intelligenz in den Templates



Preprocessing steps — Name: 1: Regular expression — Parameters: pattern

Type of information: Num

- JavaScript
- **Validation**
  - In range
  - Matches regular expression
  - Does not match regular expression
  - Check for error in JSON
  - Check for error in XML
  - Check for error using regular expression
  - Check for not supported value
- **Throttling**
  - Discard unchanged
  - Discard unchanged with heartbeat
- **Prometheus**
  - Prometheus pattern
  - Prometheus to JSON

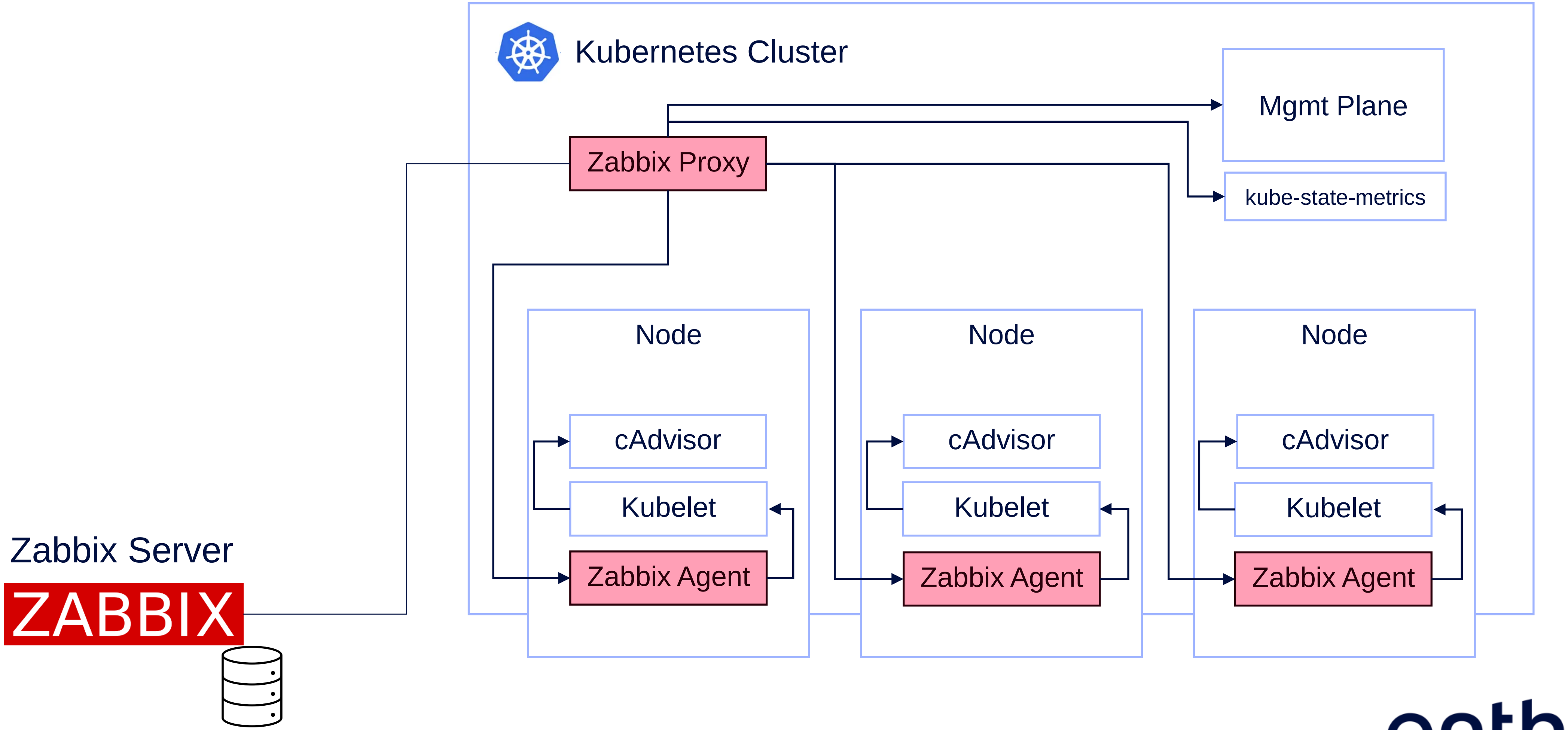


□	Name ▲	Hosts	Items	Triggers	Graphs	Dashboards	Discovery	Web	Vendor	Version
□	Kubernetes API server by HTTP	Hosts	Items 23	Triggers 2	Graphs 1	Dashboards 1	Discovery 10	Web	Zabbix	6.4-0
□	Kubernetes cluster state by HTTP	Hosts	Items 14	Triggers	Graphs	Dashboards	Discovery 19	Web	Zabbix	6.4-0
□	Kubernetes Controller manager by HTTP	Hosts	Items 13	Triggers 1	Graphs 2	Dashboards 1	Discovery 1	Web	Zabbix	6.4-0
□	Kubernetes Kubelet by HTTP	Hosts	Items 12	Triggers	Graphs	Dashboards 1	Discovery 4	Web	Zabbix	6.4-0
□	Kubernetes nodes by HTTP	Hosts	Items 2	Triggers 1	Graphs	Dashboards	Discovery 2	Web	Zabbix	6.4-0
□	Kubernetes Scheduler by HTTP	Hosts	Items 15	Triggers 3	Graphs 3	Dashboards 1	Discovery 3	Web	Zabbix	6.4-0

Best Practices

Z ♥ ☸

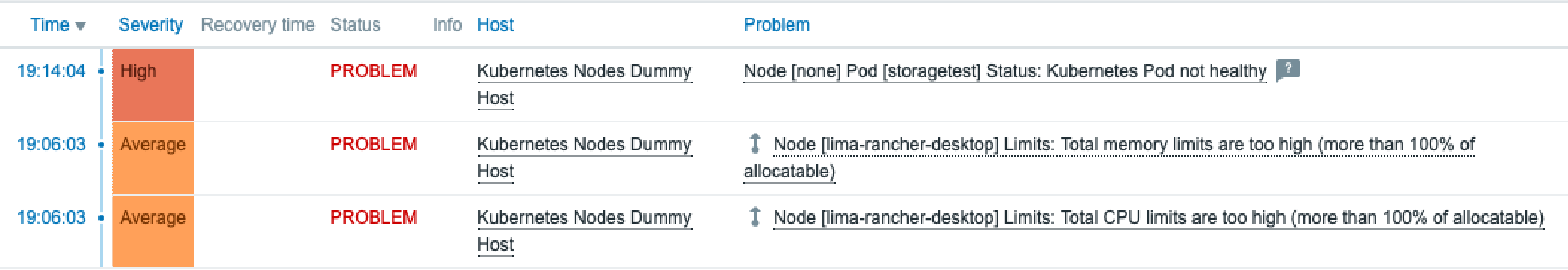
Zabbix Kubernetes Cluster Monitoring



enthus

Zabbix Kubernetes Cluster Monitoring

- Installation via *zabbix helm chart**
- Umfangreiche Metriken, verteilt auf verschiedene Zabbix Hosts
- 29 Metriken pro Pod (CPU Usage, limits, Container phase, Ready, Uptime, ...)
- Nodes Metriken (via APIs und via Zabbix Agent)
- Controller Manager, API Server, Scheduler (pro Node ein Zabbix Host)
- Kubelet (pro Node ein Zabbix Host) – umfasst auch viele Pod-spezifische Metriken
- Trigger / Problems



Time ▾	Severity	Recovery time	Status	Info	Host	Problem
19:14:04	High		PROBLEM		Kubernetes Nodes Dummy Host	Node [none] Pod [storagetest] Status: Kubernetes Pod not healthy ?
19:06:03	Average		PROBLEM		Kubernetes Nodes Dummy Host	↕ Node [lima-rancher-desktop] Limits: Total memory limits are too high (more than 100% of allocatable)
19:06:03	Average		PROBLEM		Kubernetes Nodes Dummy Host	↕ Node [lima-rancher-desktop] Limits: Total CPU limits are too high (more than 100% of allocatable)

*) https://git.zabbix.com/projects/ZT/repos/kubernetes-helm/

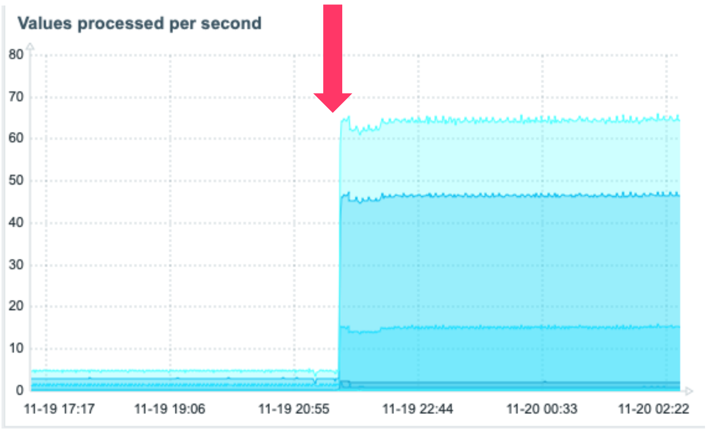
Zabbix Kubernetes Cluster Monitoring

Herausforderungen

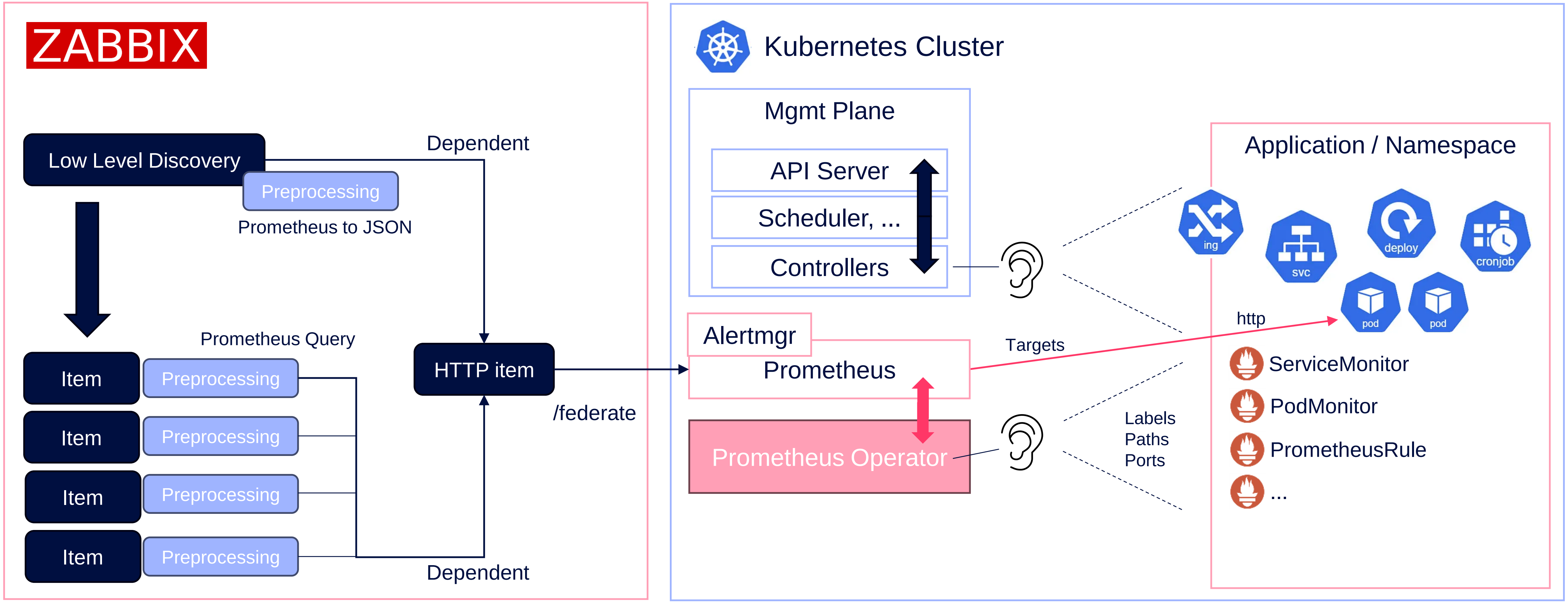
- Sehr viele Metriken
- Keine mitgelieferten Dashboards
- Dummy-Hosts für die verschiedenen Komponenten des Kubernetes Clusters
- Metriken wie Speicherverbrauch pro Pod und Persistent Volume Claims (Füllstände) fehlen, können aber eingerichtet werden
- Komplexe Logik (Preprocessing, ...)

Empfehlungen

- Durchsehen der Metriken und Trigger, einschränken auf notwendige
- Filter für spezifische, Admin – relevante Pod- und Namespace Namen definieren



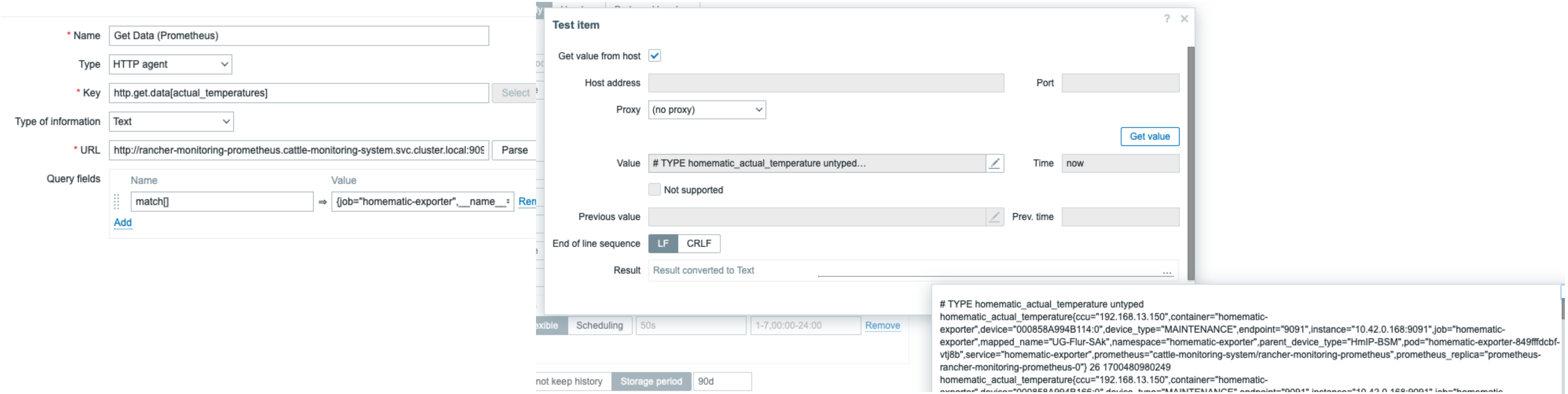
Zabbix Kubernetes Workload Monitoring



enthus

Anwendungsbeispiel

Home Automation Zentrale, überwacht durch Prometheus



* Name	Get Data (Prometheus)
Type	HTTP agent
* Key	http.get.data[actual_temperatures]
Type of information	Text
* URL	http://rancher-monitoring-prometheus.cattle-monitoring-system.svc.cluster.local:90

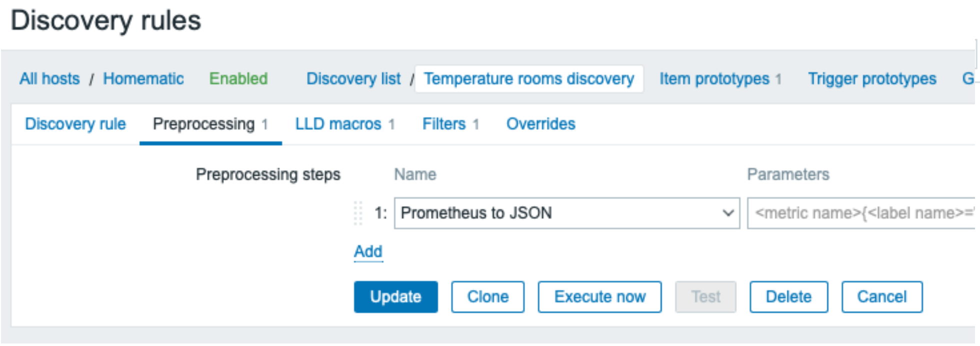
Query fields

Name		Value
match[]	⇒	{job="homematic-exporter",__name__=

Test item

Get value from host	☑		
Host address		Port	
Proxy	(no proxy)		
Value	# TYPE homematic_actual_temperature untyped…	Time	now
	□ Not supported		
Previous value		Prev. time	
End of line sequence	LF CRLF		
Result	Result converted to Text		

```
# TYPE homematic_actual_temperature untyped
homematic_actual_temperature{ccu="192.168.13.150",container="homematic-
exporter",device="000858A994B114:0",device_type="MAINTENANCE",endpoint="9091",instance="10.42.0.168:9091",job="homematic-
exporter",mapped_name="UG-Flur-SAk",namespace="homematic-exporter",parent_device_type="HmIP-BSM",pod="homematic-exporter-849fffdcbf-
vtj8b",service="homematic-exporter",prometheus="cattle-monitoring-system/rancher-monitoring-prometheus",prometheus_replica="prometheus-
rancher-monitoring-prometheus-0"} 26 1700480980249
homematic_actual_temperature{ccu="192.168.13.150",container="homematic-
```



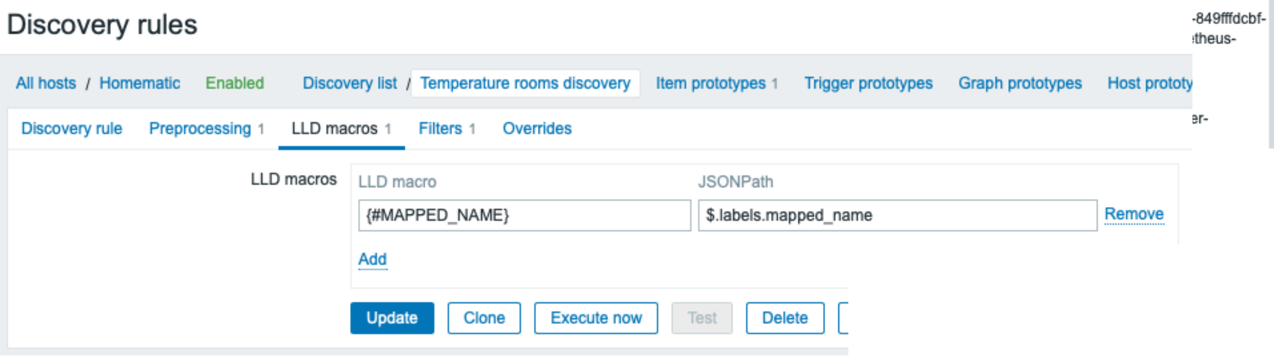
Discovery rules

All hosts / Homematic Enabled Discovery list / Temperature rooms discovery Item prototypes 1 Trigger prototypes G
Discovery rule Preprocessing 1 LLD macros 1 Filters 1 Overrides

Preprocessing steps

Name	Parameters
1: Prometheus to JSON	<metric name>{<label name>=

Add
Update Clone Execute now Test Delete Cancel



Discovery rules

All hosts / Homematic Enabled Discovery list / Temperature rooms discovery Item prototypes 1 Trigger prototypes Graph prototypes Host prototy
Discovery rule Preprocessing 1 LLD macros 1 Filters 1 Overrides
-849fffdcbf-
theus-
er-

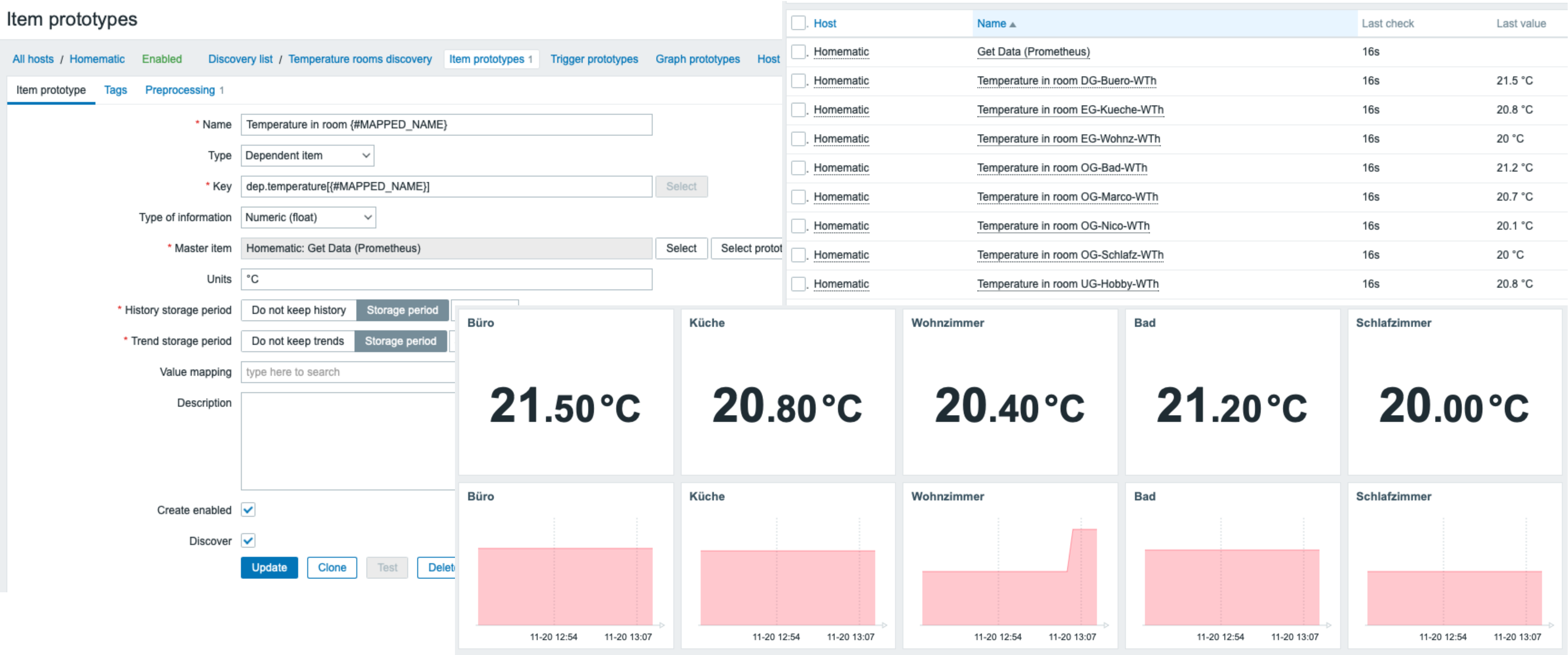
LLD macros

LLD macro	JSONPath
{#MAPPED_NAME}	$.labels.mapped_name

Remove
Add
Update Clone Execute now Test Delete

Anwendungsbeispiel

Home Automation Zentrale, überwacht durch Prometheus



Item prototypes

All hosts / Homematic Enabled Discovery list / Temperature rooms discovery Item prototypes 1 Trigger prototypes Graph prototypes Host

Item prototype Tags Preprocessing 1

* Name	Temperature in room {#MAPPED_NAME}
Type	Dependent item
* Key	dep.temperature[{#MAPPED_NAME}] Select
Type of information	Numeric (float)
* Master item	Homematic: Get Data (Prometheus) Select Select protot
Units	°C
* History storage period	Do not keep history **Storage period**
* Trend storage period	Do not keep trends **Storage period**
Value mapping	type here to search
Description	
Create enabled	☑
Discover	☑

Update Clone Test Delet

□ Host	Name ▲	Last check	Last value
□ Homematic	Get Data (Prometheus)	16s	
□ Homematic	Temperature in room DG-Buero-WTh	16s	21.5 °C
□ Homematic	Temperature in room EG-Kueche-WTh	16s	20.8 °C
□ Homematic	Temperature in room EG-Wohnz-WTh	16s	20 °C
□ Homematic	Temperature in room OG-Bad-WTh	16s	21.2 °C
□ Homematic	Temperature in room OG-Marco-WTh	16s	20.7 °C
□ Homematic	Temperature in room OG-Nico-WTh	16s	20.1 °C
□ Homematic	Temperature in room OG-Schlafz-WTh	16s	20 °C
□ Homematic	Temperature in room UG-Hobby-WTh	16s	20.8 °C

Büro	Küche	Wohnzimmer	Bad	Schlafzimmer
21.50 °C	20.80 °C	20.40 °C	21.20 °C	20.00 °C

Fazit

- Zabbix Cluster Monitoring funktioniert gut!
- Prometheus Preprocessing Funktionen erlauben Integrationen
- Zabbix ist konzeptionell anders als der übliche cloudnative Monitoring Stack
- Empfehlung:
  - Cloudnativer Monitoring Stack: Observability / DevOps
  - Zabbix: Monitoring / System Admin

Z ❤ ☸

Vielen **Dank.**

enthus