

- **Infrastructure-as-a-code**
- **skalierbares DB cluster mit OpenShift**



Daniel Kasper
Cloud Engineer

Richard Strnad
Senior Network Engineer

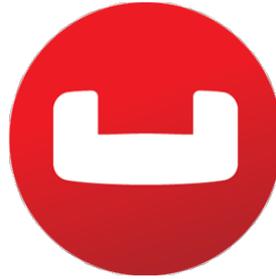
- **Wer sind wir**
- **CouchbaseDB on Openshift**
- **IaC mit Ansible als die eierlegende Wollmilchsau**

**WE'RE
HIRING!**

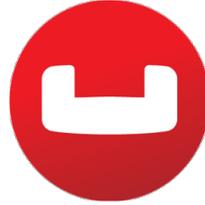
Standorte
Operations Center
Cyber Defence Center
Gründungsjahr
Mitarbeiter
Umsatz 2018
Red Hat
NetApp Partner Level
VMware Partner Level
Besitzverhältnisse

Winterthur, Basel, Bern
Winterthur, Basel, Bern
Winterthur, Bern
1998
177
153 Mio. CHF
CCSP Advanced Partner
Platinum
Enterprise Solution Provider
Zu 100% in Besitz des Managements,
eigenfinanziert

Couchbase DB Cluster on OpenShift



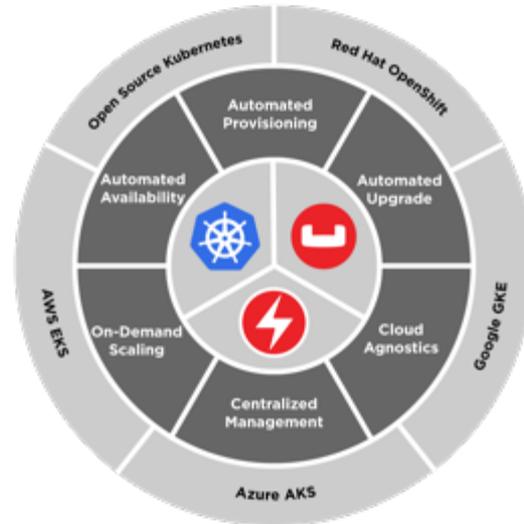
Why Couchbase



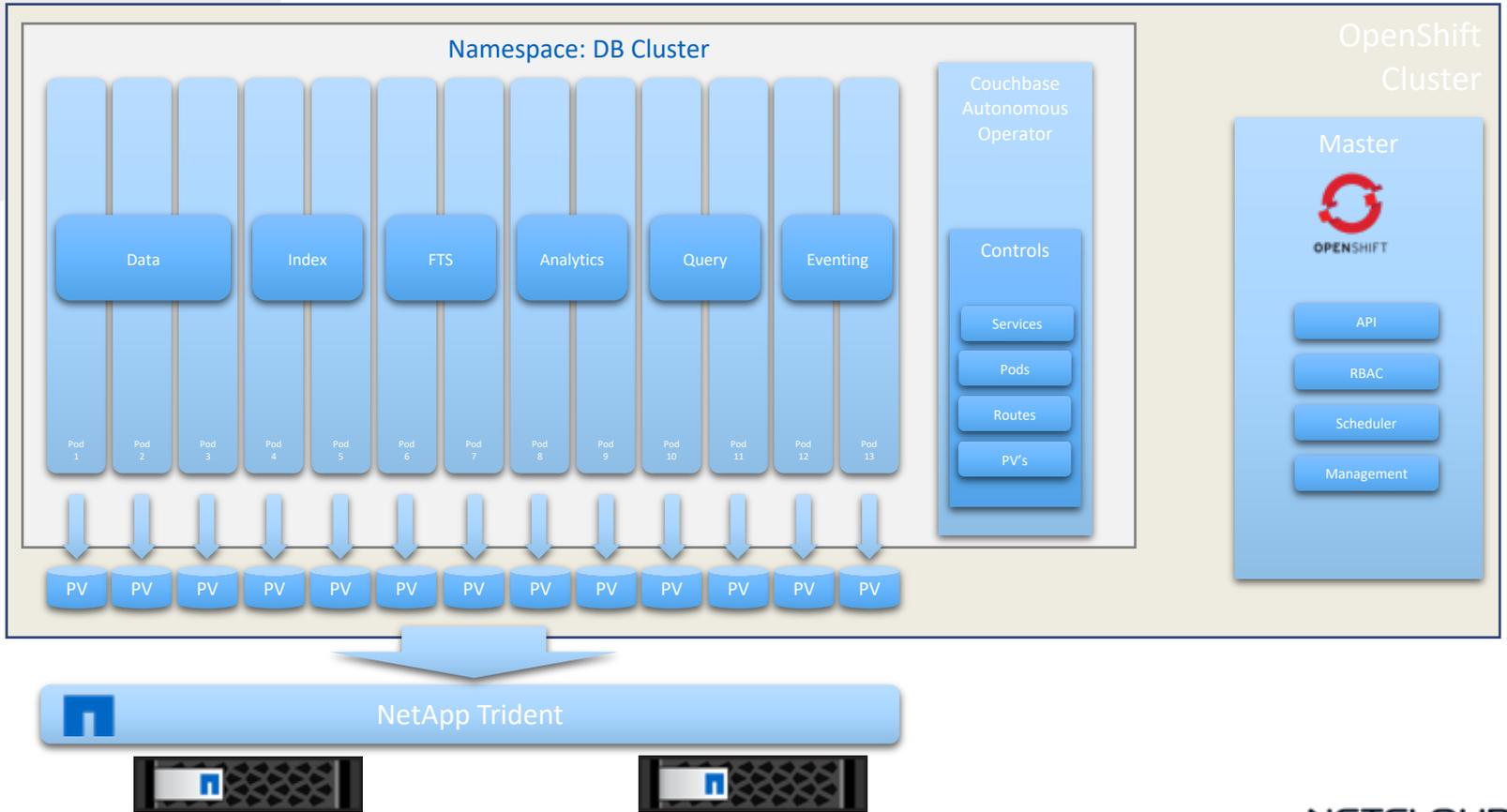
- Unified Programming Interface
- XDCR Replication
- Search
- Query N1QL
- Mobile and IoT
- Analytics
- High Availability
- Core database engine
- Scale-out architecture
- Memory-first architecture
- Big data and SQL integrations
- Full-stack security
- Container and Cloud deployments

Why Couchbase on Containers

- Elastic Scalability
- Cloud- agnostic
- Automated Cluster Provisioning
- Automate operational
- No Vendor lock-in
- Multi Cloud



Architecture



Trident

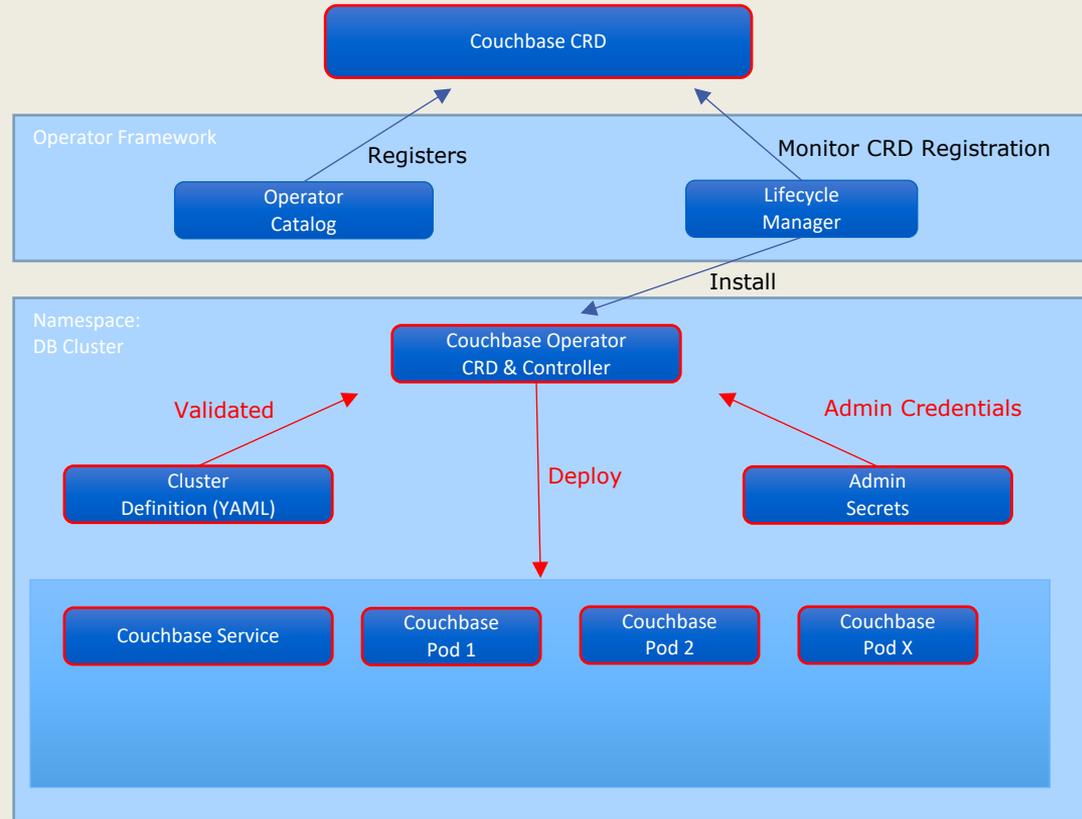
- A dynamic and automated storage provisioner for Kubernetes and Red Hat OpenShift
- Supports ONTAP, SolidFire, and E-Series
- Abstracts back ends into pools of capabilities and retains the ability to differentiate storage
- IOPS, compression, disk type, etc. all able to be specified
- Maps storage requests to storage pools, each backend can contain one or more storage pools

[Download now on github](#)



Autonomous Operator

Couchbase Operator



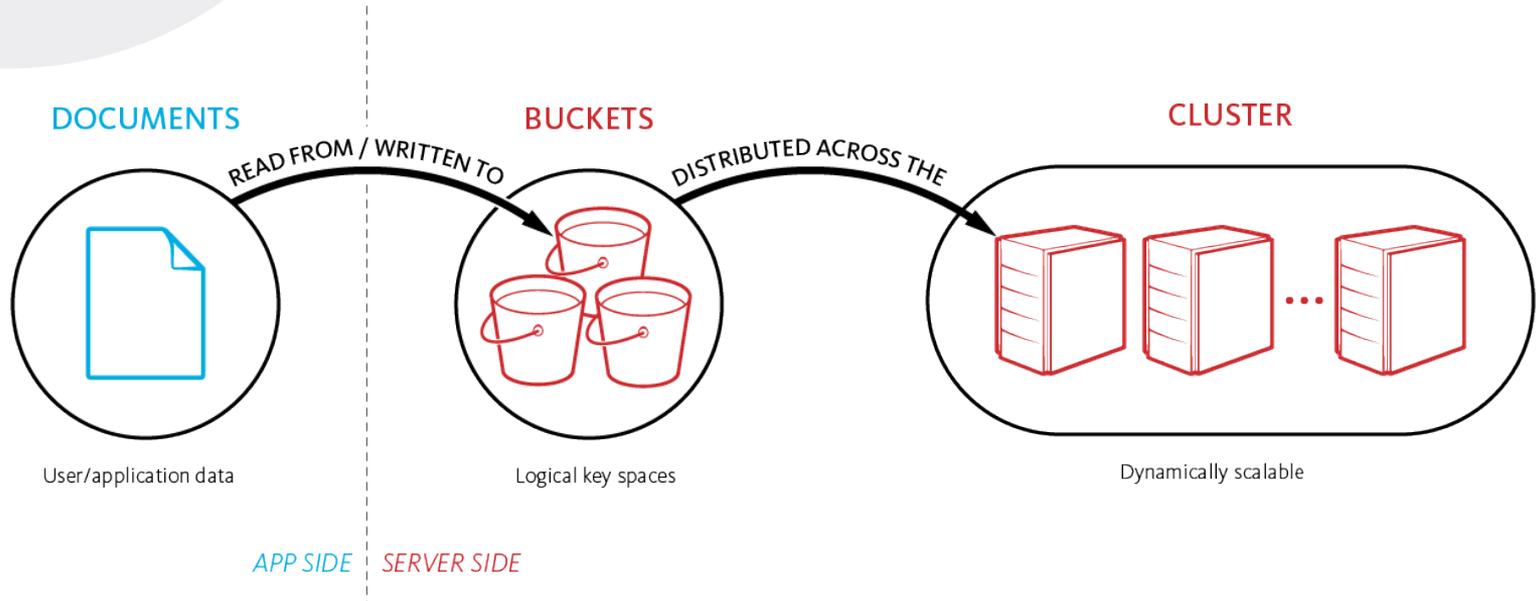
Cluster Definition snippets

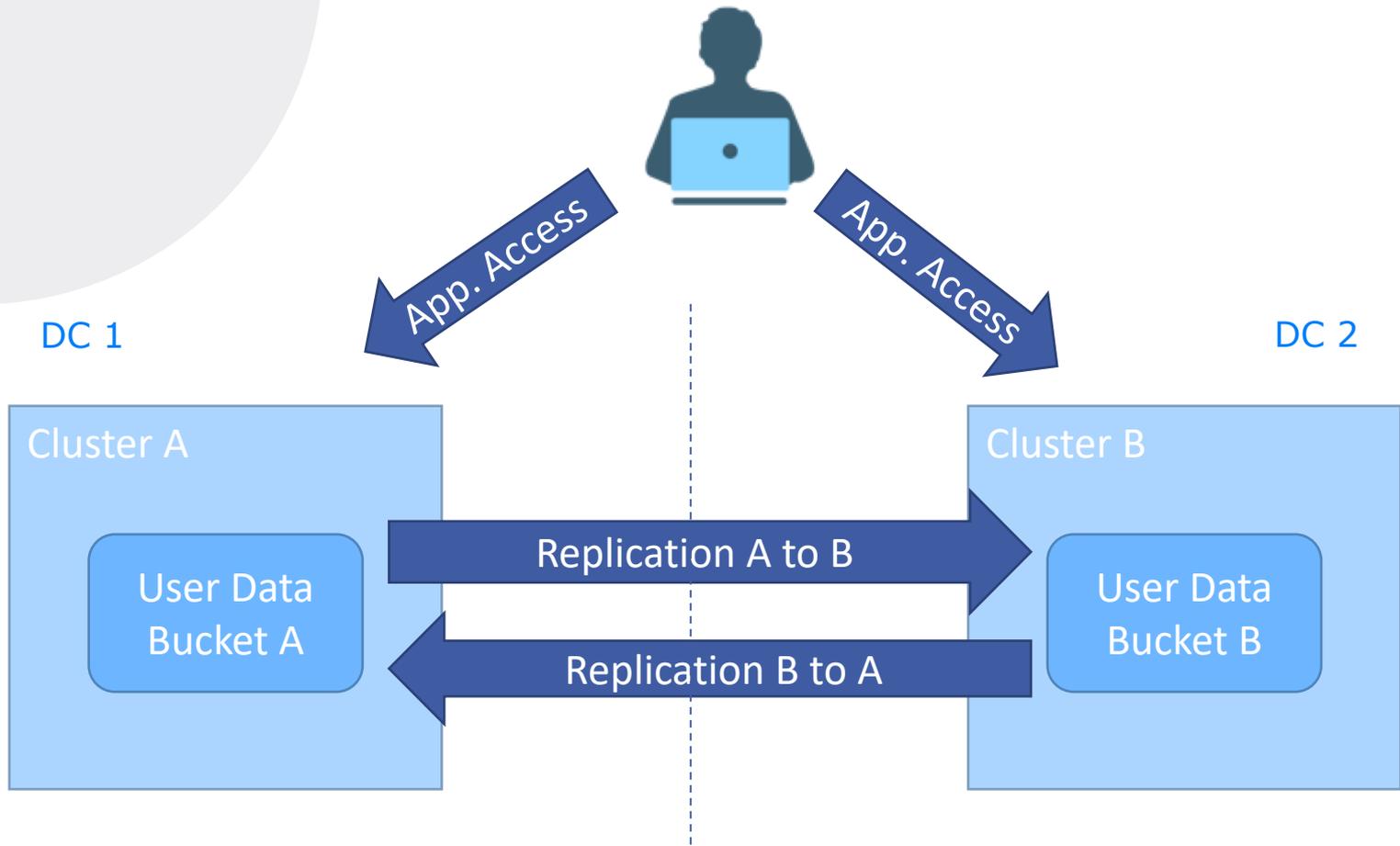
```
disableBucketManagement: true
servers:
  - size: 3
    name: all_services
    services:
      - data
      - index
      - query
      - search
    pod:
      resources:
        limits:
          cpu: 4
          memory: 15Gi
        requests:
          cpu: 3
          memory: 14Gi
    labels:
      couchbase_services: all
    automountServiceAccountToken: false
    volumeMounts:
      default: cb
      data: cb
      index: cb
      logs: cb
```

```
volumeClaimTemplates:
  - metadata:
      name: cb
    spec:
      storageClassName: "bronze"
      resources:
        requests:
          storage: 50Gi
```

```
disableBucketManagement: true
buckets:
  - name: default
    type: couchbase
    memoryQuota: 128
    replicas: 1
    ioPriority: high
    evictionPolicy: fullEviction
    conflictResolution: seqno
    enableFlush: true
    enableIndexReplica: false
```

Documents Buckets Cluster





Infrastructure as Code

Agenda

- **Konfigurationsmanagement von Infrastruktur**
- **IaC und die Rolle von Ansible**
- **Wie kann IaC in der Praxis aussehen**
- **Lessons Learned**

Managed by Hand

- Konfiguration erfolgt direkt auf einer Infrastruktur Komponente
- Dokumentation
 - Visios
 - Konfig Backups
 - Notes

```
R4#conf t
Enter configuration commands, one per line. End with CNTL/Z.
R4(config)#hostname Router-04
Router-04(config)#interface FastEthernet0/0
Router-04(config-if)#ip address 192.168.1.254 255.255.255.0
Router-04(config-if)#no shutdown
Router-04(config-if)#_
```

Infrastructure as Code (Imperativ)

- Definieren von spezifischen Aktionen (Wie)
- Scripts & Tools
- Expect/API/SNMP/...

Infrastructure as Code (Deklarativ)

- Definieren eines Zustandes unabhängig der Infrastruktur (Definition -> Apply) (Was)
- ~Dokumentation im Code
- Versioning der Infrastruktur
- Weniger Fehler
- Effizienz
- Desired State Configuration

- Ansible History
 - Entwickelt von Michael DeHaan, Ansible Inc. wurde von RedHat akquiriert
- Automation für "alles"
 - OS, Application, Infrastructure
- Imperativ und/oder Deklarativ
 - je nach Modul und Ansatz
- Idempotent
- "Workflow Management" mit Ansible Tower

- Use-Case des Kunden (Service Provider)
 - Automatisches Deployment von Applikationen inkl. Umsysteme
 - IPAM
 - LB
 - Netzwerk
 - Server/Storage/Compute
 - OS
 - Applikation

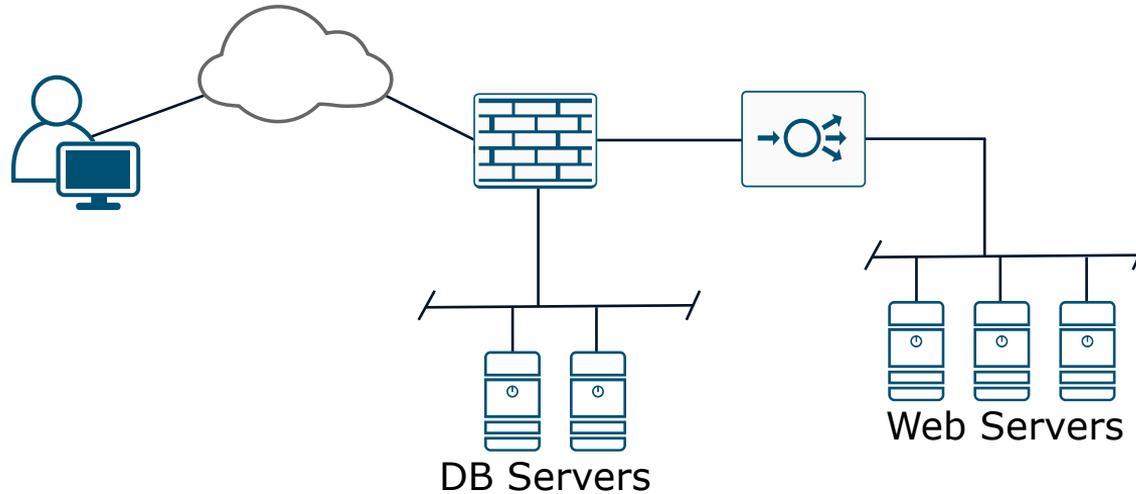
Wie kann IaC in der Praxis aussehen

Lösung

- Modellierung von Applikationen im ITSM (ServiceNow)
 - Service Katalog
- ITSM ist single source of truth
- Überwachung des Configuration Drift
 - Abweichungen vom definierten Zustand!
- End-to-End Deployment mit Ansible

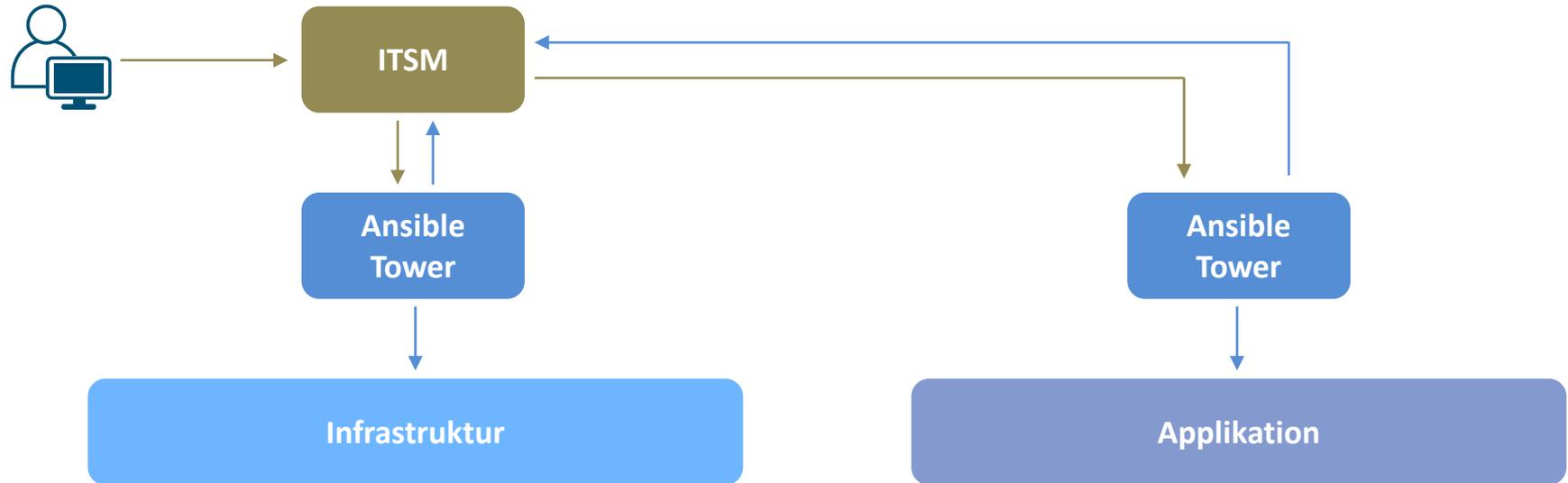
Wie kann IaC in der Praxis aussehen

Design der Beispiel Applikation



Wie kann IaC in der Praxis aussehen

Automation der Beispiel Applikation



**Wie kann IaC in
der Praxis
aussehen**

Bestellung einer Applikation

Wie kann IaC in der Praxis aussehen

Step 1.

Bestellung der Applikation durch Benutzer



Applikations Bestellung



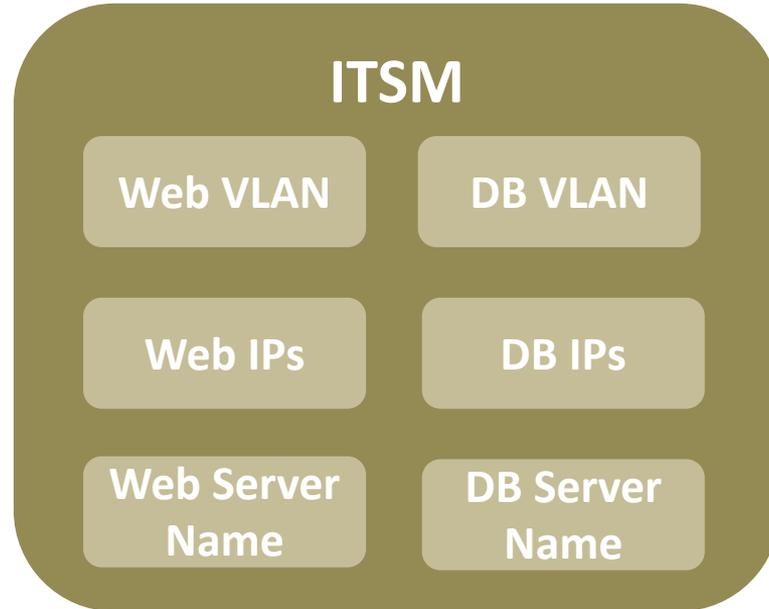
ITSM



Wie kann IaC in der Praxis aussehen

Step 2.

ITSM generiert Parameter



Wie kann IaC in der Praxis aussehen

Step 3.

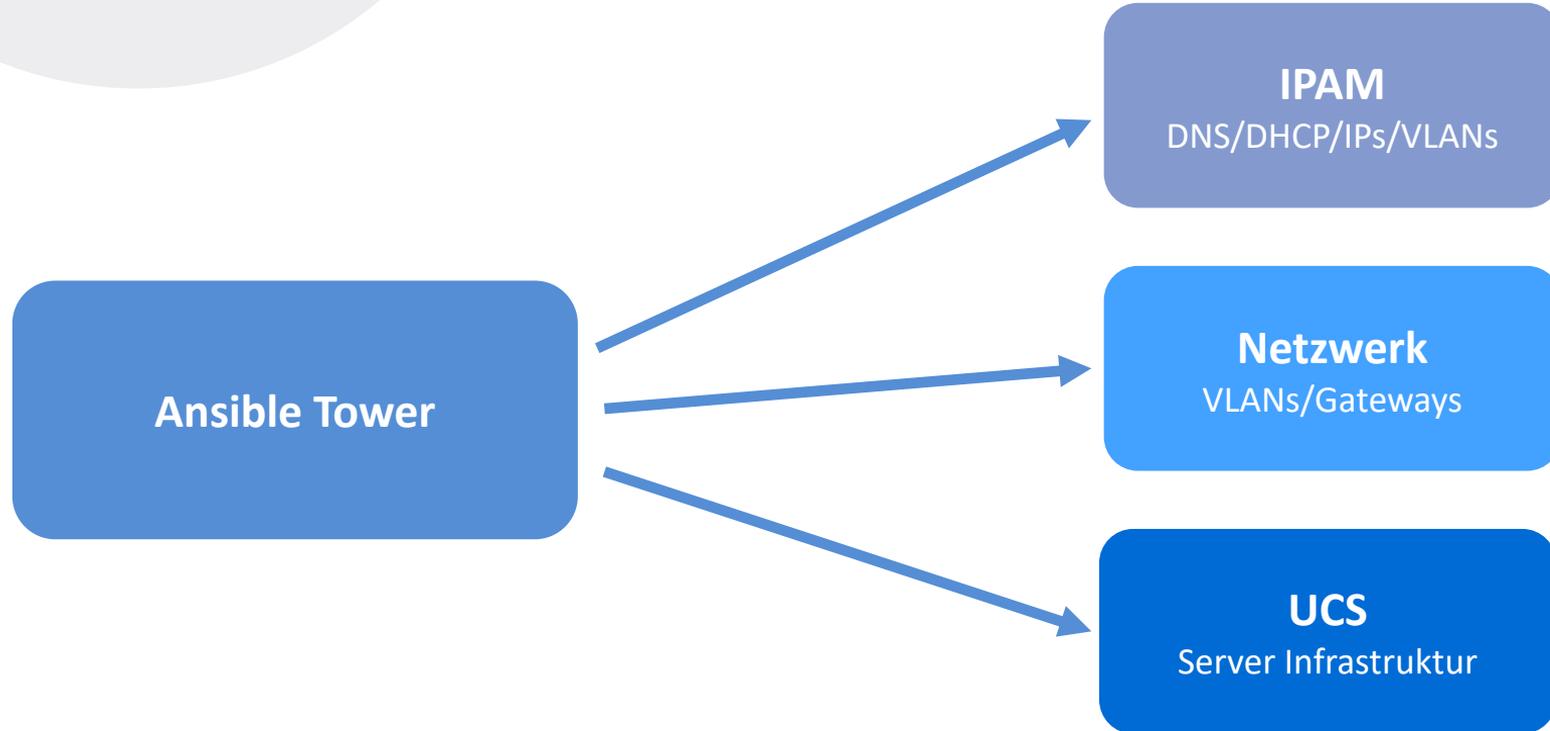
ITSM startet die Provisionierung der Infrastruktur



Wie kann IaC in der Praxis aussehen

Step 4.

Ansible Tower provisioniert die Infrastruktur



Wie kann IaC in der Praxis aussehen

Step 4b.

Ansible meldet den Status an ITSM



Wie kann IaC in der Praxis aussehen

Step 5.

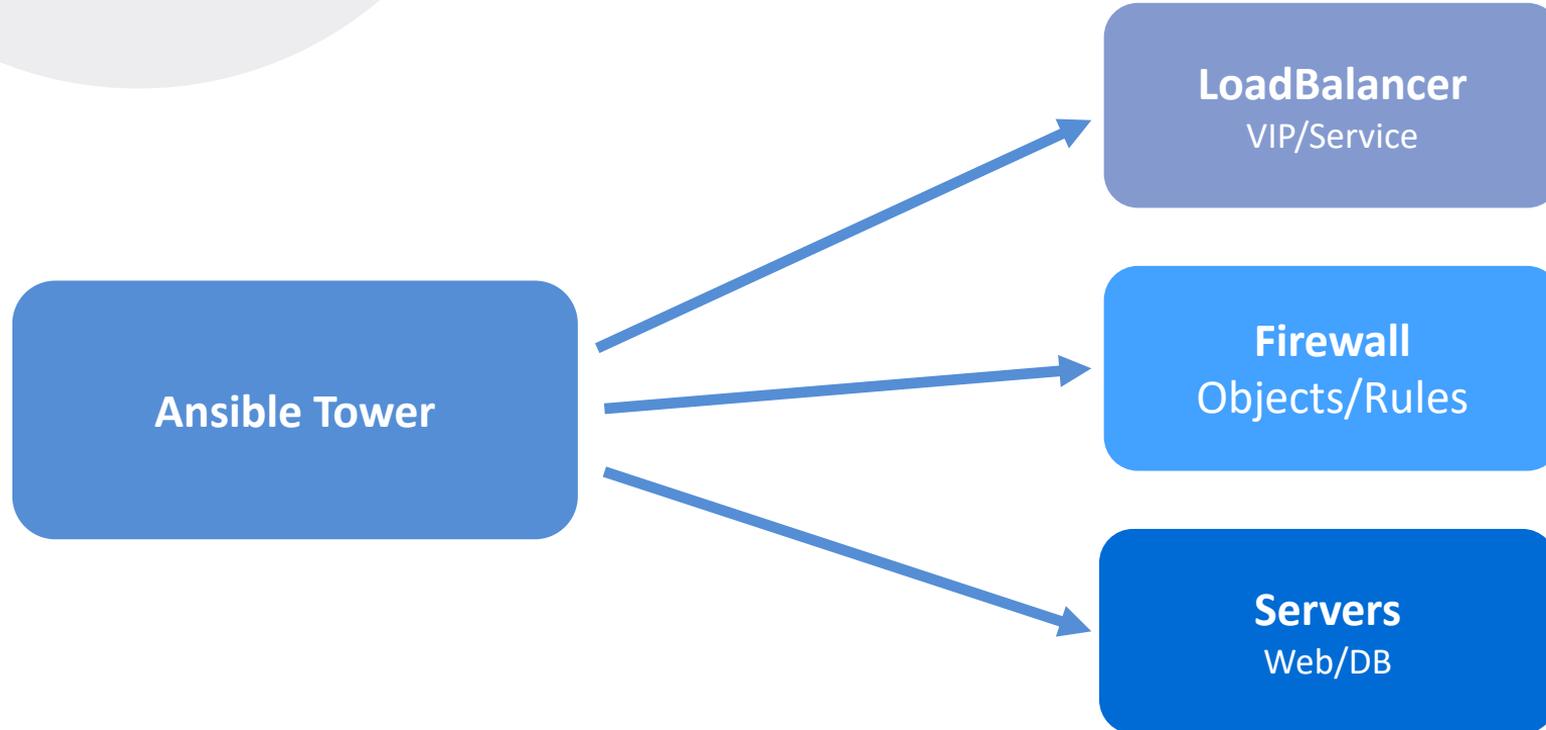
ITSM startet das Deployment der Applikation



Wie kann IaC in der Praxis aussehen

Step 5.

Ansible Tower deployed die Applikation



Wie kann IaC in der Praxis aussehen

Step 5b.

Ansible meldet den Status an ITSM



**Wie kann IaC in
der Praxis
aussehen**

Configuration Drift

Wie kann IaC in der Praxis aussehen

Step 1.

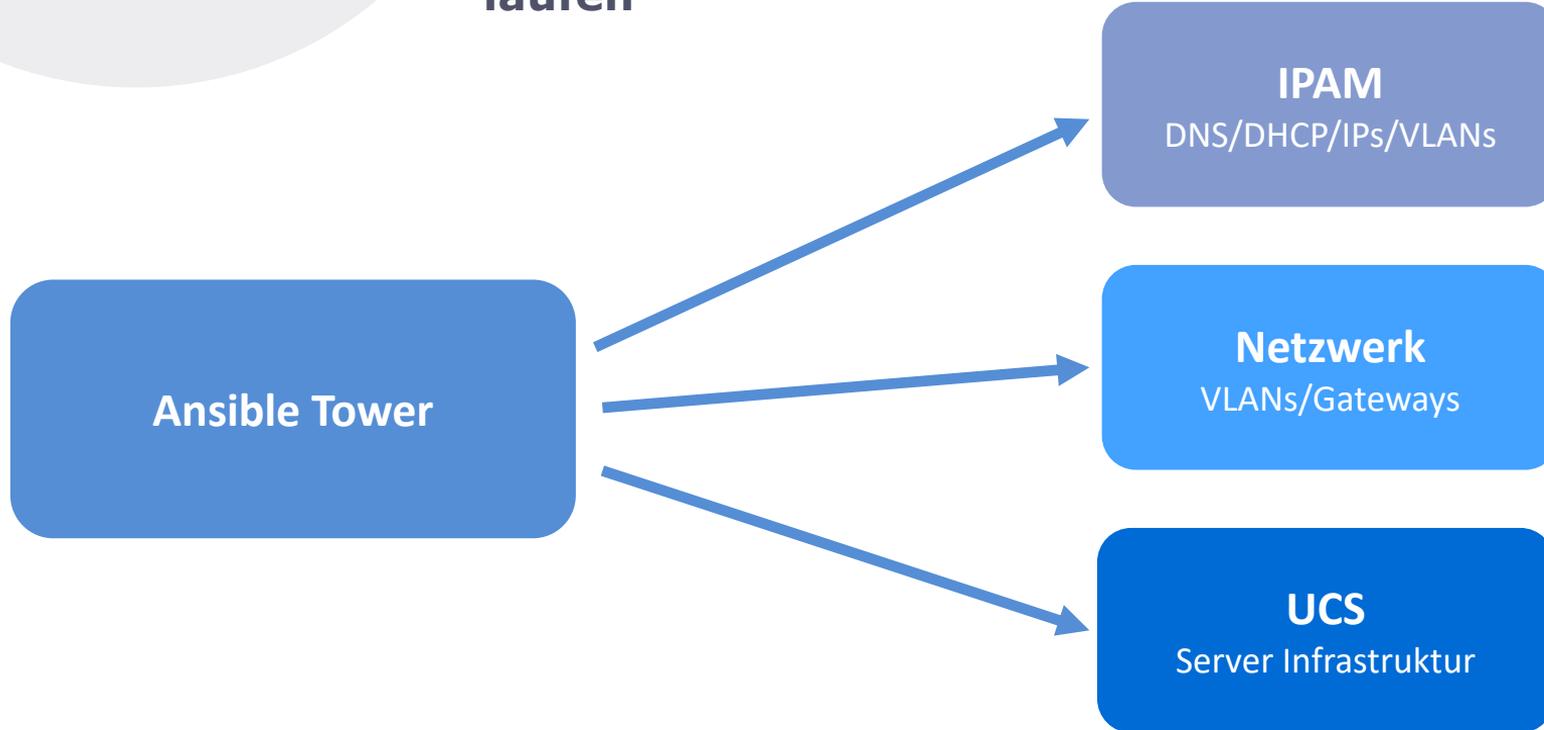
Scheduler in ITSM für Configuration Drift Analyse



Wie kann IaC in der Praxis aussehen

Step 2.

Ansible Tower lässt die Playbooks im Check Mode laufen



Wie kann IaC in der Praxis aussehen

Step 2b.

Ansible meldet status an ITSM, keines der Playbooks sollte „changed“ sein.



Übersicht

- Warum gibt es Ansible erst jetzt? 😊
 - Es ist mehr möglich als man denkt!
- End-to-End nur für Greenfield sinnvoll machbar...
- Ansible auch ohne IaC (Deklarativ) sinnvoll!

- **Trotz allem: Automation ist kein Selbstzweck!**

Q&A

