

# Die Cloud im Griff mit Consul

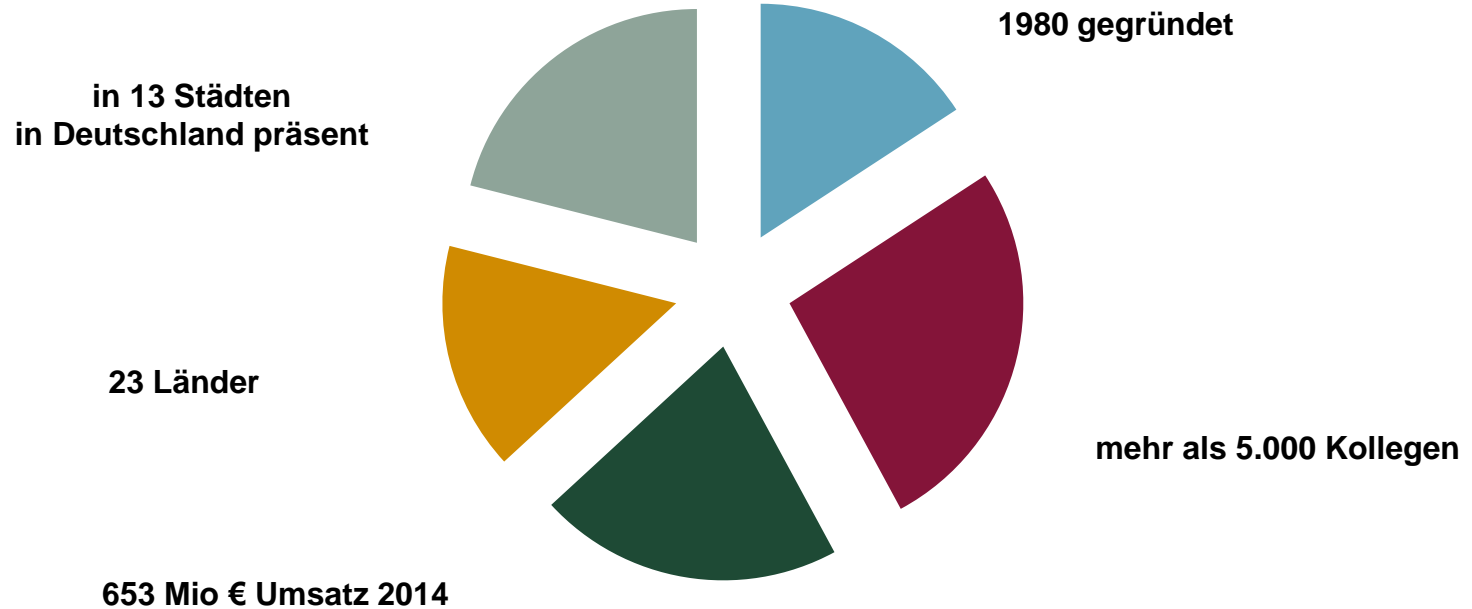
GUUG Frühjahrsfachgespräch – 26. Februar 2016

# Die Cloud im Griff mit Consul

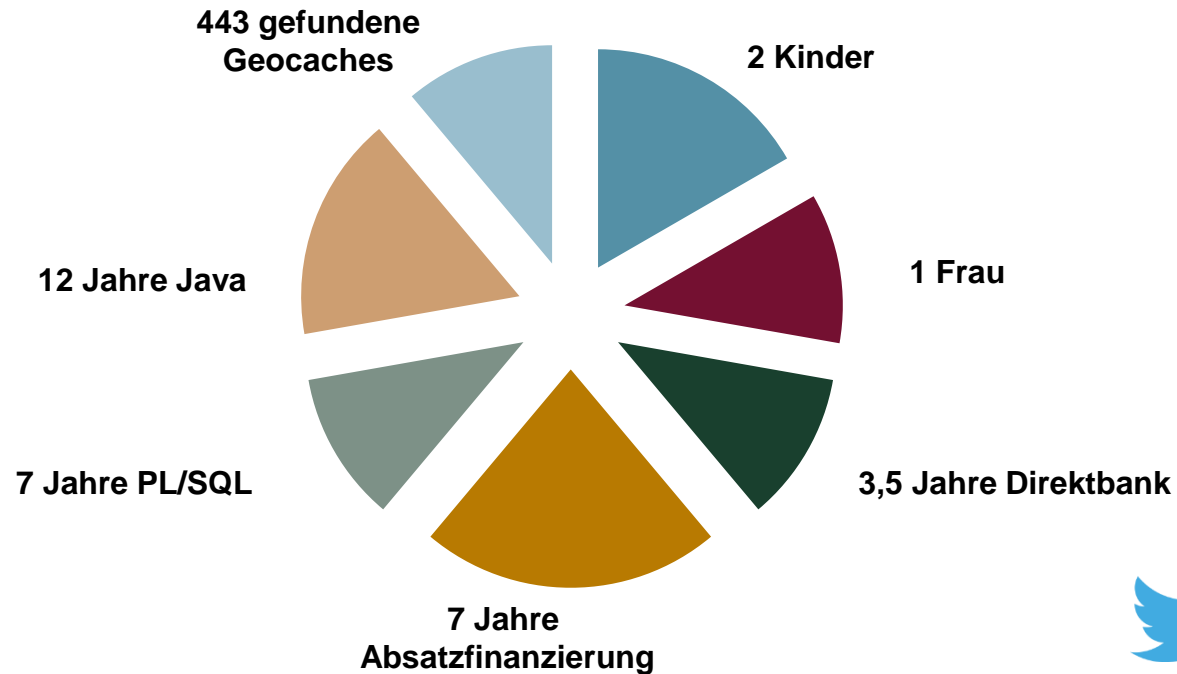
---

- 1 Cloud Setup und Herausforderungen
- 2 Einstieg in Consul
- 3 Services mit Consul
- 4 Key-Value-Store mit Consul
- 5 Fortgeschrittene Anwendungsfälle
- 6 Consul für Cloud und Enterprise

# Mein Sponsor und Arbeitgeber – msg systems ag



# Wer ich bin – Principal IT Consultant im GB Travel & Logistics

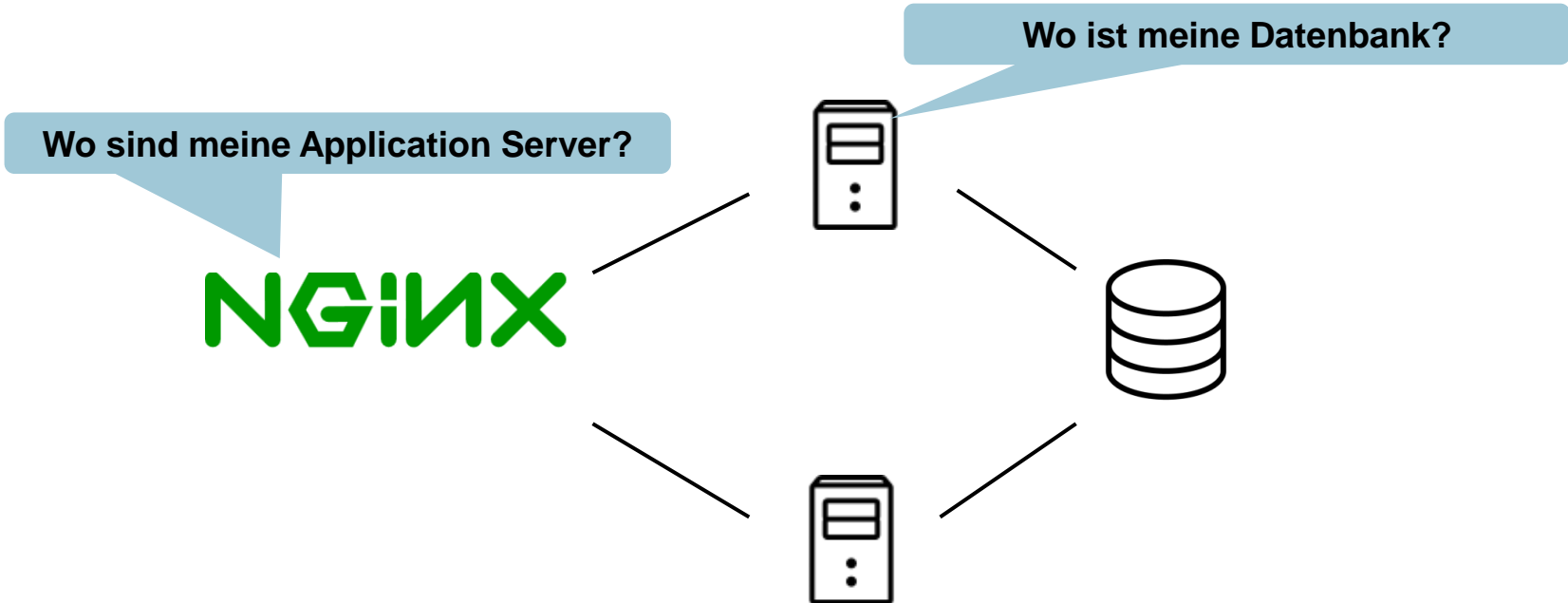


# Die Cloud im Griff mit Consul

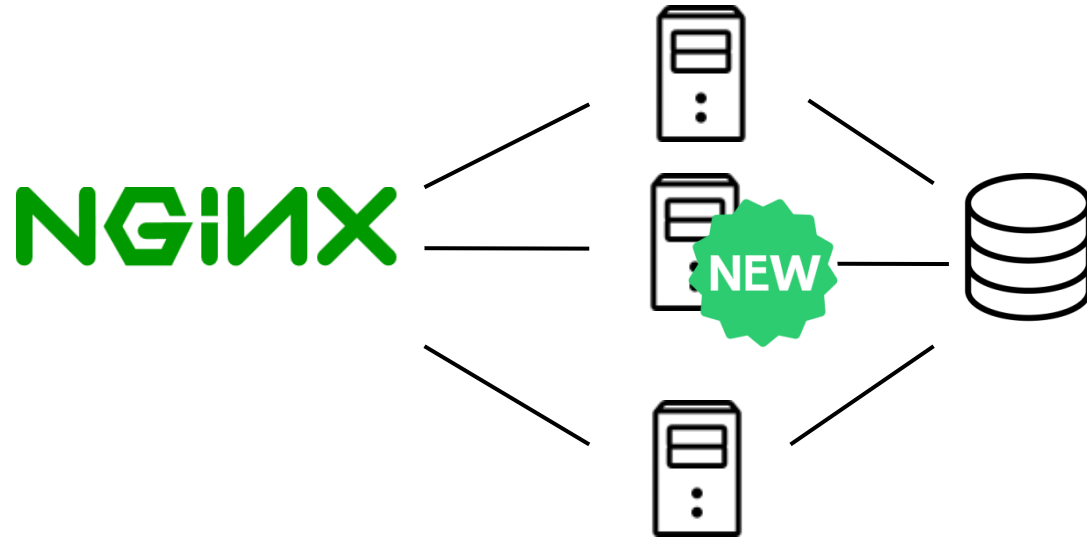
---

- 1 Cloud Setup und Herausforderungen**
- 2 Einstieg in Consul
- 3 Services mit Consul
- 4 Key-Value-Store mit Consul
- 5 Fortgeschrittene Anwendungsfälle
- 6 Consul für Cloud und Enterprise

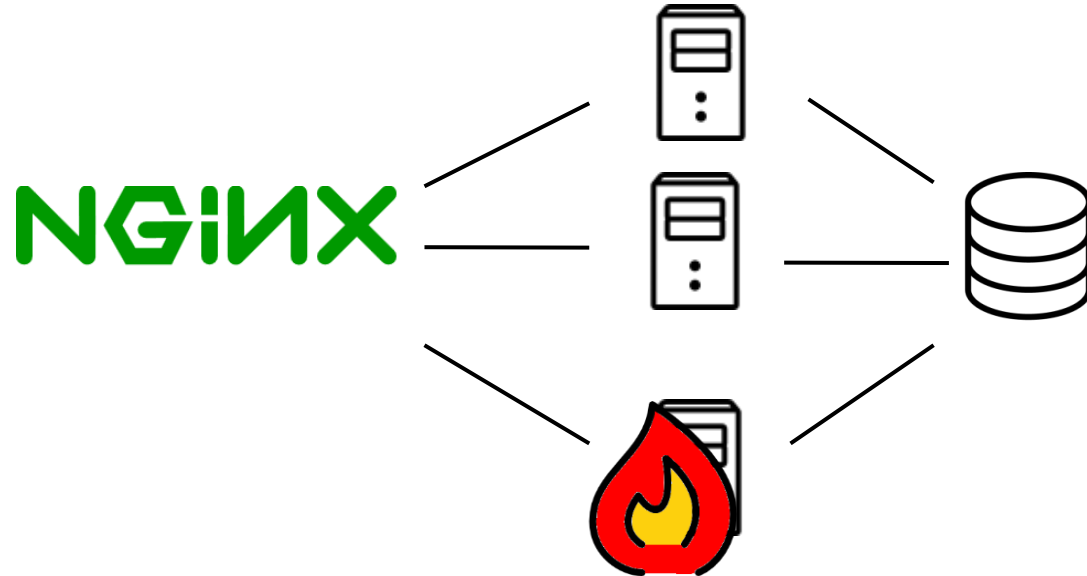
## Ein einfaches Cloud Setup



## Ein einfaches Cloud Setup



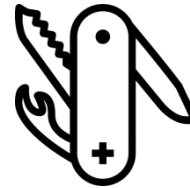
## Ein einfaches Cloud Setup





## Automatisierung bevorzugt

- Händische Konfiguration
  - Funktioniert nur bei wenigen Servern
  - Kann Monitoring-Status nicht berücksichtigen
  - Ungeeignet für dynamische Skalierung
- Configuration Management
  - Umsetzbar mit Puppet/Chef/Ansible/Salt
  - Dynamische Skalierung möglich, aber schwerfällig
  - CM muss Monitoring implementieren
- Eigener Dienst für Konfiguration und Service Discovery
  - HashiCorp Consul
  - Aktualisierung von Konfigurationsdateien
  - Nachschlagen von Diensten per DNS oder REST



# Die Cloud im Griff mit Consul

- 1 Cloud Setup und Herausforderungen
- 2 Einstieg in Consul**
- 3 Services mit Consul
- 4 Key-Value-Store mit Consul
- 5 Fortgeschrittene Anwendungsfälle
- 6 Consul für Cloud und Enterprise

## Steckbrief zu Consul

**Haupt-Sponsor:**

HashiCorp (die, die auch Vagrant machen)

**Homepage:**

<https://www.consul.io/>

**Lizenz:**

Open Source: Mozilla Public License, Version 2.0

**Programmiert in:**

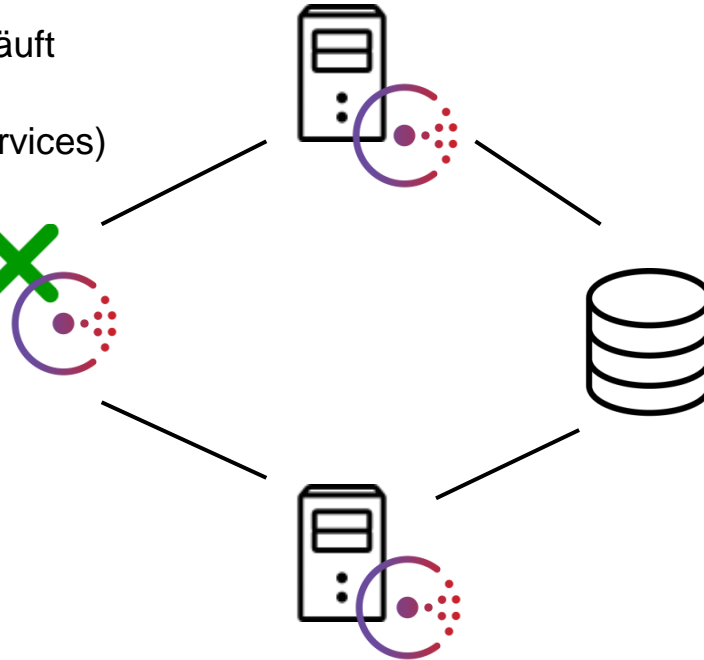
Go



## Infrastruktur für Consul

Auf jedem Knoten läuft  
ein Consul Client  
(für Checks und Services)

**NGINX**



Ein hochverfügbares  
Consul-Server-Ensemble  
(für Key-Value-Store  
und Service-Catalog)



# Die Cloud im Griff mit Consul

- 1 Cloud Setup und Herausforderungen
- 2 Einstieg in Consul
- 3 Services mit Consul**
- 4 Key-Value-Store mit Consul
- 5 Fortgeschrittene Anwendungsfälle
- 6 Consul für Cloud und Enterprise

## Jede Consul Instanz kennt ihre lokalen Services

```
"service": {  
  "name": "web",  
  "tags": ["master"],  
  "port": 80  
}
```

## Jede Consul Instanz prüft sich und ihre lokalen Services

```
"check": {  
  "id": "http",  
  "name": "HTTP API on port 80",  
  "http": "http://localhost:80",  
  "service_id": "web",  
  "interval": "10s",  
  "timeout": "1s"  
}
```

```
"check": {  
  "id": "ttl",  
  "name": "Web Status",  
  "service_id": "web",  
  "ttl": "30s"  
}
```

```
"check": {  
  "id": "script",  
  "name": "Script Check",  
  "script": "/usr/local/bin/check_mem.py",  
  "service_id": "web",  
  "interval": "10s"  
}
```

## Consul Clients und Server sind vernetzt

auf Kommando...

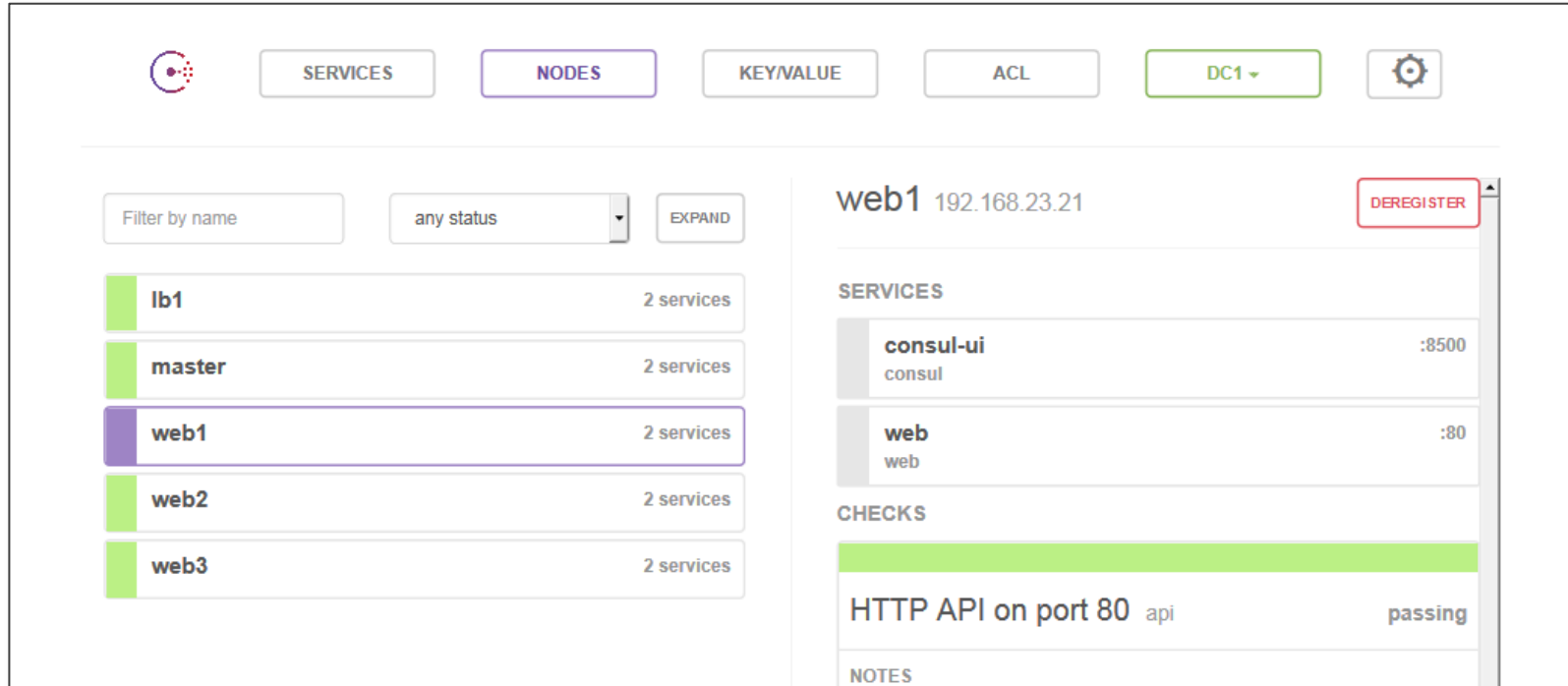
```
$ consul join xx.xx.xx.xx
```

... oder per Konfiguration

```
"start_join": [ "192.168.23.12" ],  
"rejoin_after_leave": true
```



## Consul UI Demo



The screenshot displays the Consul UI interface. At the top, there are navigation tabs: SERVICES, NODES, KEYVALUE, ACL, and DC1 (selected). Below the tabs, there are filters: 'Filter by name' (empty), 'any status' (dropdown), and 'EXPAND' (button). A list of services is shown on the left, with 'web1' selected and highlighted in purple. The right pane shows details for 'web1' at IP 192.168.23.21, including a 'DEREGISTER' button. The 'SERVICES' section lists 'consul-ui' (port 8500) and 'web' (port 80). The 'CHECKS' section shows a green bar and a check 'HTTP API on port 80' with status 'api' and 'passing'. The 'NOTES' section is empty.

Service Name	IP Address	Actions
lb1		2 services
master		2 services
<b>web1</b>	192.168.23.21	2 services <b>DEREGISTER</b>
web2		2 services
web3		2 services

Service Name	Port
consul-ui	:8500
consul	
web	:80
web	

Check Name	Status
HTTP API on port 80	api passing

## Services finden per DNS mit Hostnamenauflösung

```
$ dig @127.0.0.1 -p 8600 web.service.dc1.consul. ANY

web.service.dc1.consul. 0      IN      A       192.168.23.22
web.service.dc1.consul. 0      IN      A       192.168.23.21
```

## Services finden per DNS mit SRV Record

```
$ dig @127.0.0.1 -p 8600 web.service.dc1.consul. SRV

web.service.dc1.consul. 0      IN      SRV     1 1 80 web2.node.dc1.consul.
web.service.dc1.consul. 0      IN      SRV     1 1 80 web1.node.dc1.consul.

web1.node.dc1.consul.   0      IN      A       192.168.23.21
web2.node.dc1.consul.   0      IN      A       192.168.23.22
```

## Consul Template für automatisch aktualisierte Konfigurationsdateien

```
template {
  source = "/etc/nginx/conf.d/upstream.ctmpl"
  destination = "/etc/nginx/conf.d/upstream.conf"
  # adding "true" to allow keep consul-template running
  # even if nginx restart fails
  command = "service nginx restart || true"
}
```

```
# upstream.ctmpl
upstream web {
  {{range service "web"}}
    server {{.Address}};
  {{end}}
}
```

```
upstream web {
  server 192.168.23.21;
  server 192.168.23.22;
}
```

# Die Cloud im Griff mit Consul

- 1 Cloud Setup und Herausforderungen
- 2 Einstieg in Consul
- 3 Services mit Consul
- 4 Key-Value-Store mit Consul**
- 5 Fortgeschrittene Anwendungsfälle
- 6 Consul für Cloud und Enterprise

## Consul Key Value Store GUI

The screenshot displays the Consul Key Value Store GUI. At the top, there is a navigation bar with buttons for SERVICES, NODES, KEY/VALUE (selected), ACL, DC1 (dropdown), and a settings gear icon. Below the navigation bar, the main content area is divided into two sections. On the left, under the heading 'MYAPP/ +', there is a list of keys: 'dburl' (highlighted in purple) and 'maintenance' (highlighted in grey). On the right, a detailed view for the key 'myapp/dburl' is shown. It features a green header bar, the key name 'myapp/dburl', and a text area containing the value 'jdbc:postgresql://db.service.consul:5432/mydb'. At the bottom of this view, there are three buttons: 'UPDATE' (green), 'CANCEL' (grey), and 'DELETE KEY' (red).

## Consul Key Value Store REST API

```
$ curl -X PUT -d 'jdbc:postgresql://db.service.consul:5432/mydb' ↻  
http://localhost:8500/v1/kv/myapp/dburl  
true  
  
$ curl http://localhost:8500/v1/kv/myapp/dburl  
[  
  {  
    "CreateIndex":169,  
    "ModifyIndex":169,  
    "LockIndex":0,  
    "Key":"myapp/dburl",  
    "Flags":0,  
    "Value":"am..Ri"  
  }  
]
```

base64 encoded

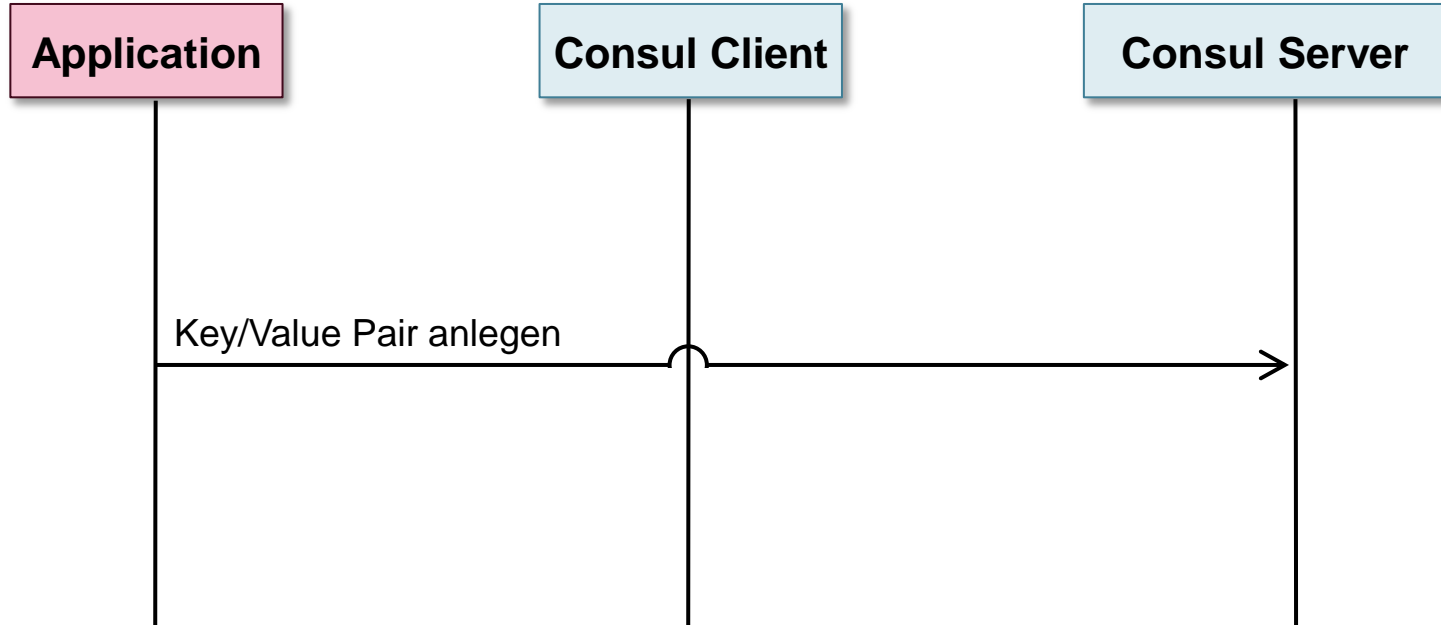
# Die Cloud im Griff mit Consul

---

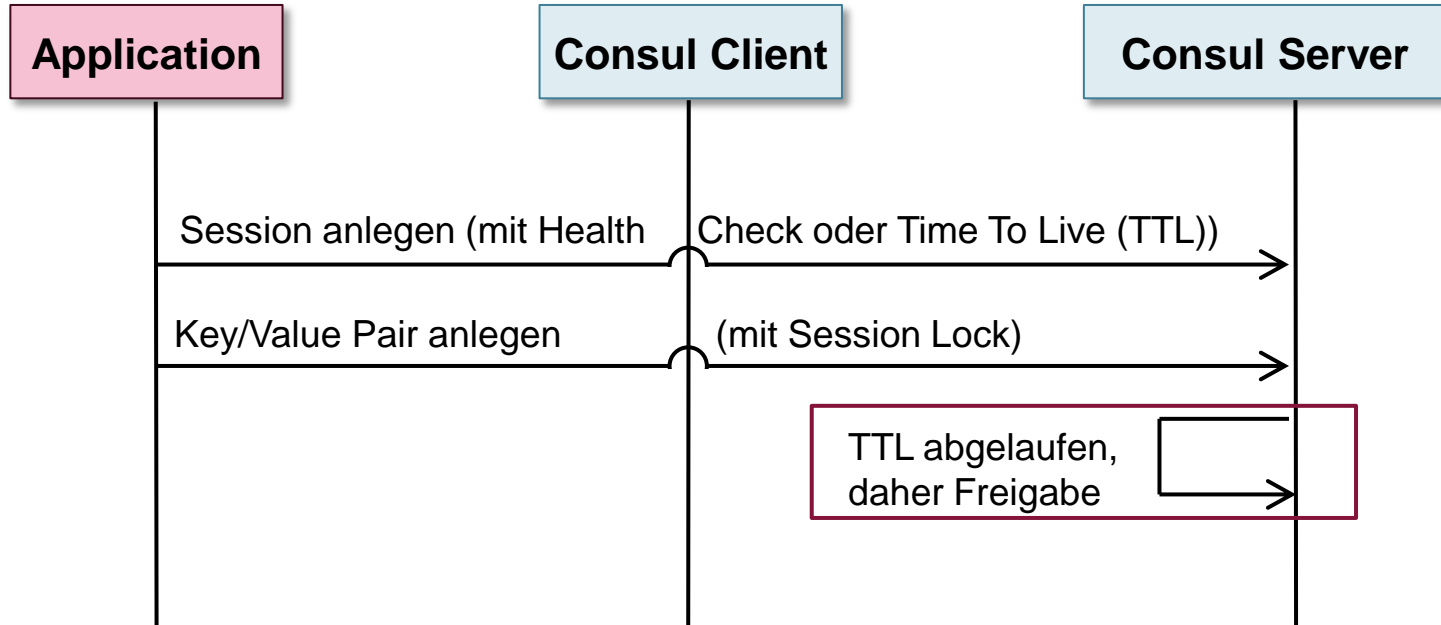
- 1 Cloud Setup und Herausforderungen
- 2 Einstieg in Consul
- 3 Services mit Consul
- 4 Key-Value-Store mit Consul
- 5 Fortgeschrittene Anwendungsfälle**
- 6 Consul für Cloud und Enterprise



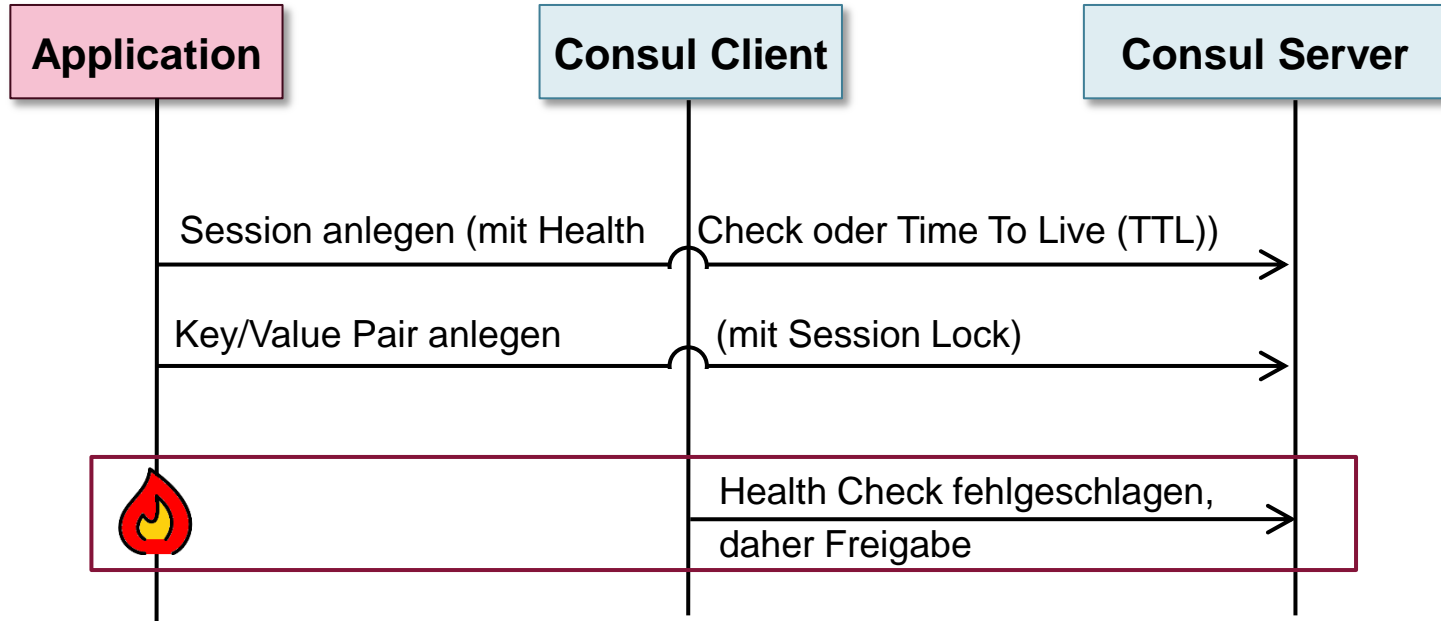
## Consul Sessions



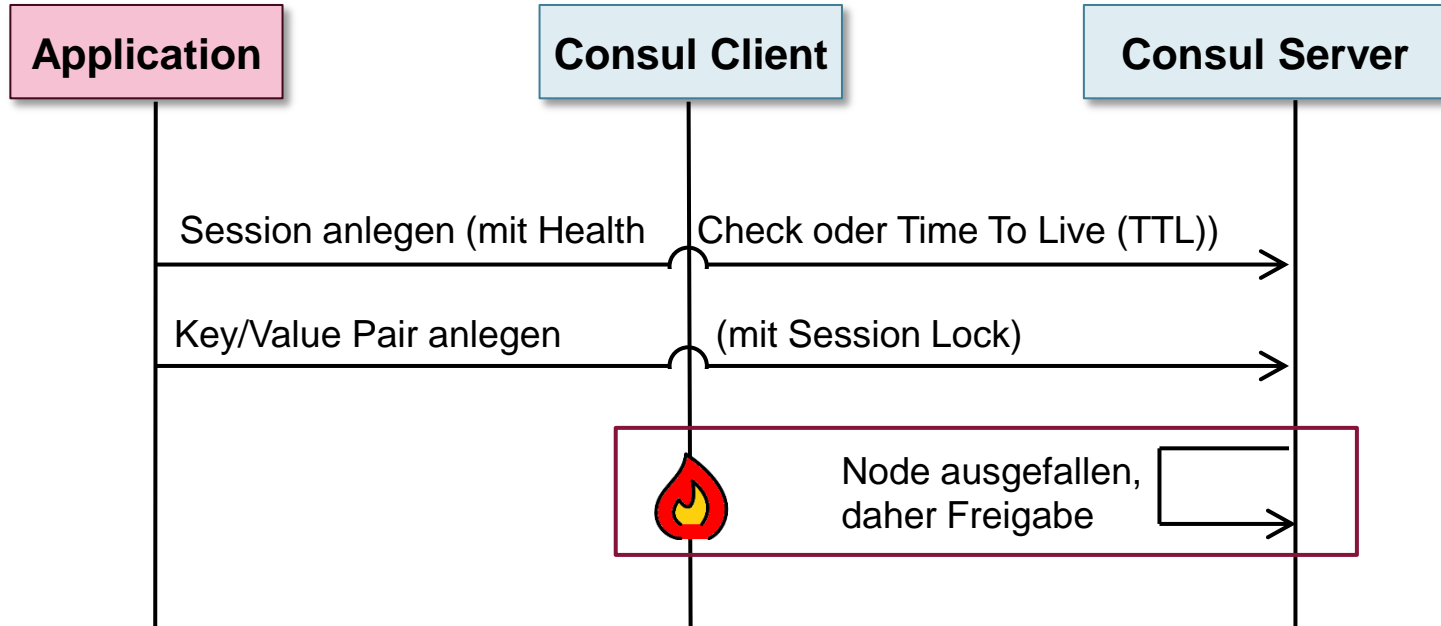
## Consul Sessions



# Consul Sessions



# Consul Sessions



## Consul – was noch dazugehört

---

### **Datacenter**

- Services können in Datacenters gruppiert werden. Damit kann lokationsbezogener Lookup erfolgen

### **Security**

- Serf-Protokoll kann verschlüsselt werden (shared key)
- RPC-Protokoll zwischen Consul-Knoten kann verschlüsselt werden (TLS)
- Access Control Lists für den Lese- und Schreibzugriff (für Services und Keys)
- DNS und REST sind immer lokal (und daher unverschlüsselt)

## Consul – command line goodies

Abo für Benachrichtigungen aus Consul

```
$ consul watch -type [key|keyprefix|services|nodes|service|check|event]
```

Starte ein Kommando genau n-Mal im Datacenter

```
$ consul lock -n 1 <command>
```

# Die Cloud im Griff mit Consul

---

- 1 Cloud Setup und Herausforderungen
- 2 Einstieg in Consul
- 3 Services mit Consul
- 4 Key-Value-Store mit Consul
- 5 Fortgeschrittene Anwendungsfälle
- 6 Consul für Cloud und Enterprise**

## Consul – für Cloud und Enterprise

---

### Ready for the Cloud

- Einfache Verwaltung von vielen Knoten
- Unterstützt dezentrales Monitoring und kurzlebige Knoten (ephemeral nodes)
- Sicherheit ist eingebaut

### Ready for the Enterprise

- Kann genauso im normalen Rechenzentrum laufen
- Ortsbezug von Services auch im Enterprise Rechenzentrum relevant
- Hat eine grafische Web-Oberfläche

### Einfach zu installieren und zu integrieren

- Installation als einfaches Go-Binary
- Konfiguration über JSON-Dateien oder REST-API
- Integration in Anwendungen technologieunabhängig über DNS oder REST-API



@ahus1de



## Links

---

### **Consul**

<http://consul.io/>

### **Artikel zu Consul bei Heise Developer**

<http://heise.de/-3040847>

### **Consul Template**

<https://github.com/hashicorp/consul-template>

### **Beispielprojekt**

<https://github.com/ahus1/saltconsul-examples>



@ahus1de



**Alexander Schwartz**  
Principal IT Consultant

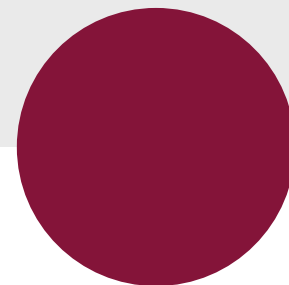
+49 171 5625767  
alexander.schwartz@msg-systems.com



@ahus1de

**msg systems ag** (Headquarters)  
Robert-Buerkle-Str. 1, 85737 Ismaning  
Germany

**[www.msg-systems.com](http://www.msg-systems.com)**



.consulting .solutions .partnership