DEEPSHORE

DEEPTALK

ConfigMaps und Secrets

2020

Agenda

Recap

Grundlagen und Beispiele aus der Praxis

Ressourcen erzeugen

Zusammenfassung

Recap

- Containerisierung einer App
- Containerbereitstellung über Docker-Registries
- k8s-Ressourcen einer App
- peripher behandelt: ConfigMaps und Secrets

Folge 3: https://deepshore.de/knowledge/_videos/deeptalk-3

Beispiel I: Postgres konfigurieren

- **Ziel**: Aufsetzen und Konfigurieren einer Datenbank-Infrastruktur
- PostgreSQL / Postgres als objektrelationales Datenbankmanagementsystem
- **Vorgehen**: Schrittweises Einbinden von Konfigurationsparametern durch:
  - 1. Umgebungsvariablen
  - 2. ConfigMaps
  - 3. ConfigMaps + Secrets

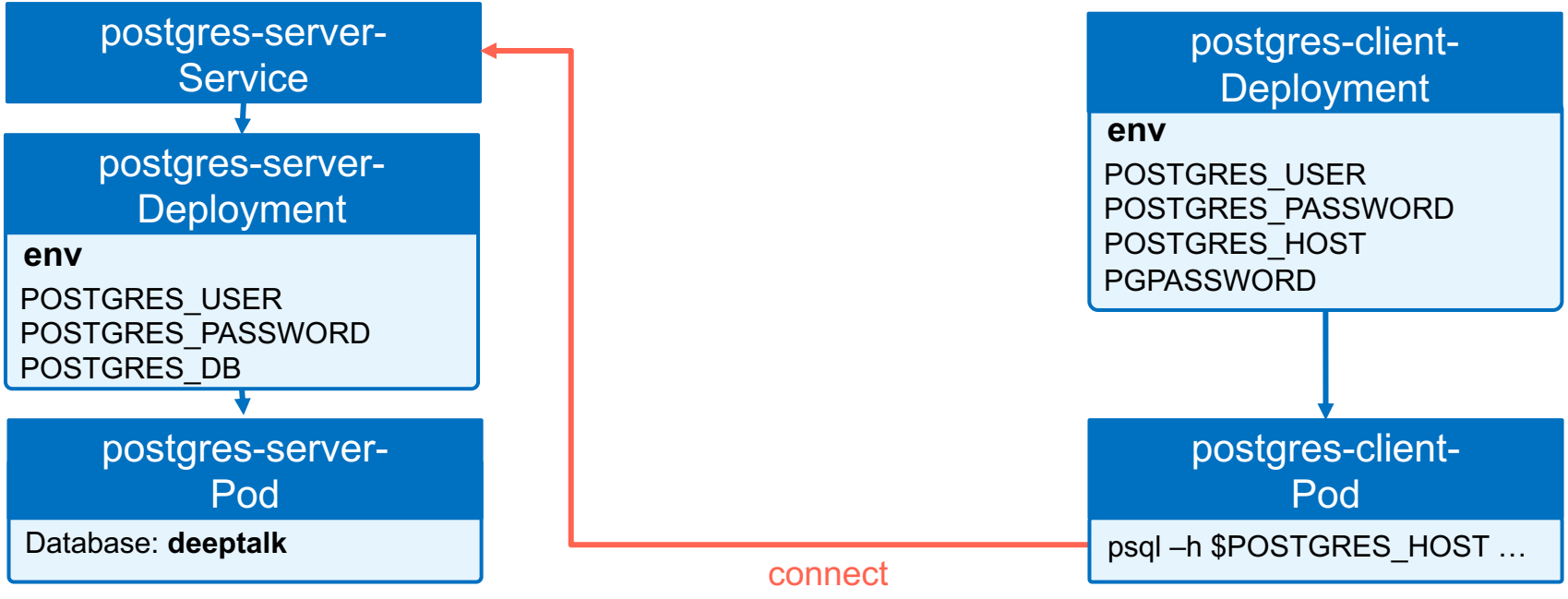
ConfigMaps

- Speichern von nicht vertraulichen Konfigurationsdaten
- Key-Value-Paare
- Trennung von Konfiguration und Anwendungscode
- Umgebungsvariablen, CLI-Argumente, Konfigurationsdateien

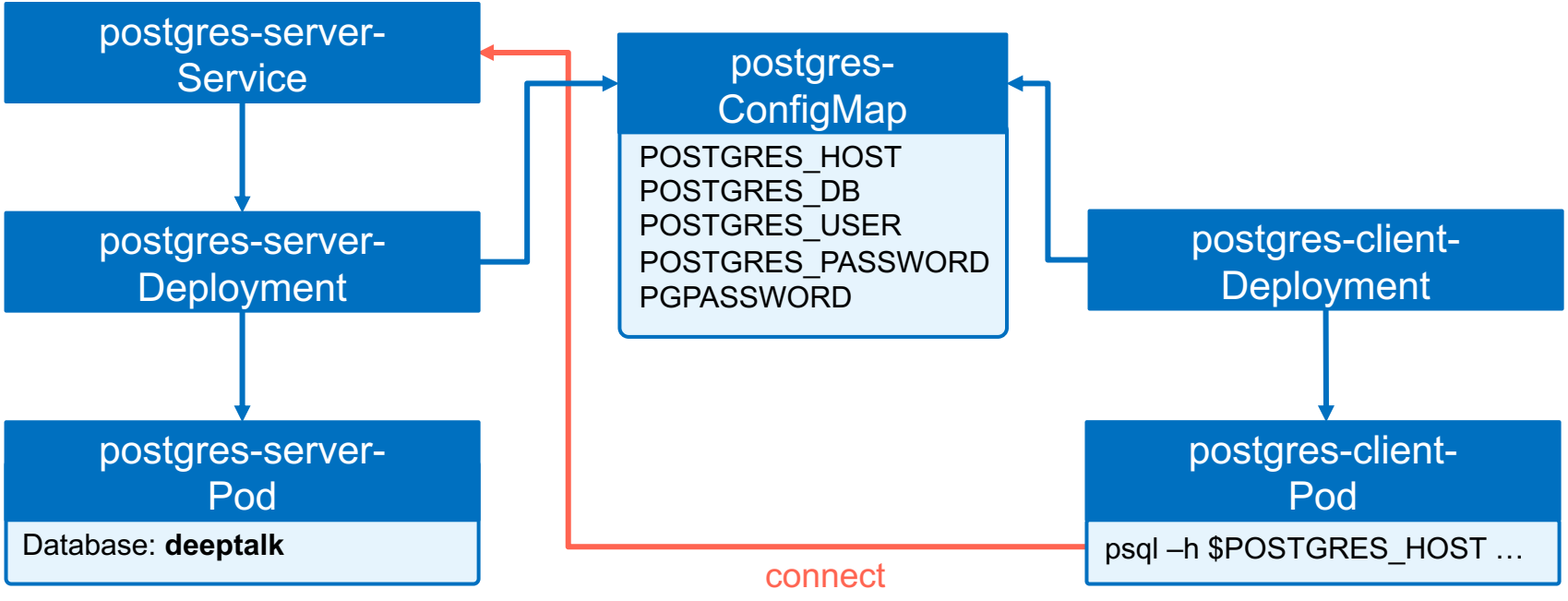
Secrets

- Speichern von vertrauenswürdigen Daten
  - Passwörter, Oauth-Tokens, SSH-Keys
- Key-Value-Paare (Value: base64-encoded)
- Einbindung wie ConfigMaps

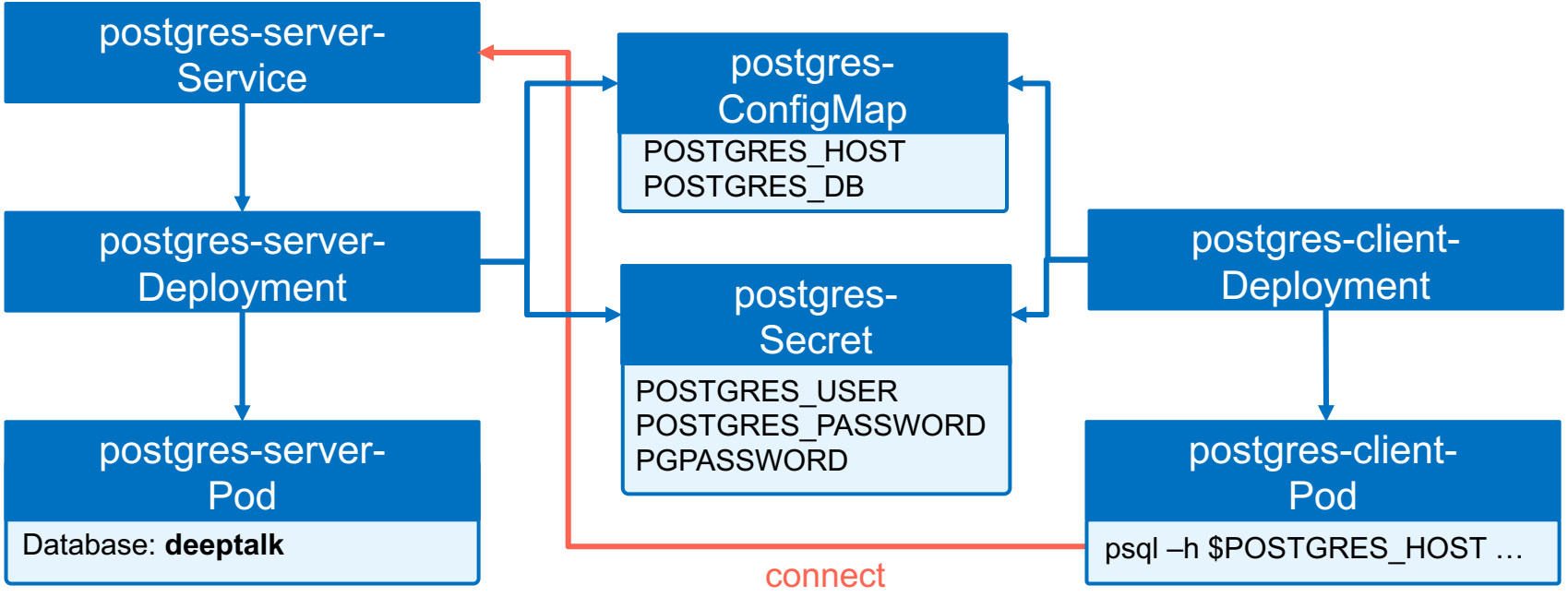
Beispiel I: Umgebungsvariablen



Beispiel I: ConfigMap



Beispiel I:
ConfigMap + Secret



Beispiel II: Grafana konfigurieren

Grafana

Webanwendung für interaktive Visualisierung von Daten

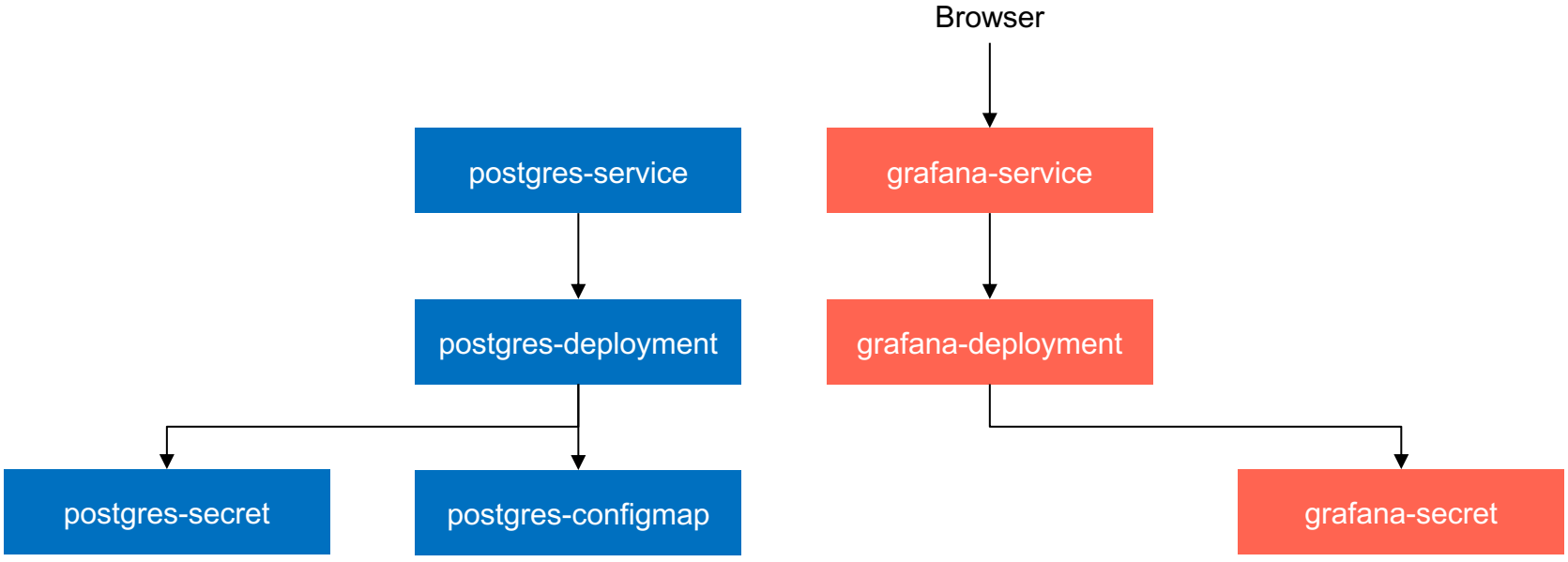
Datenquelle

Postgres

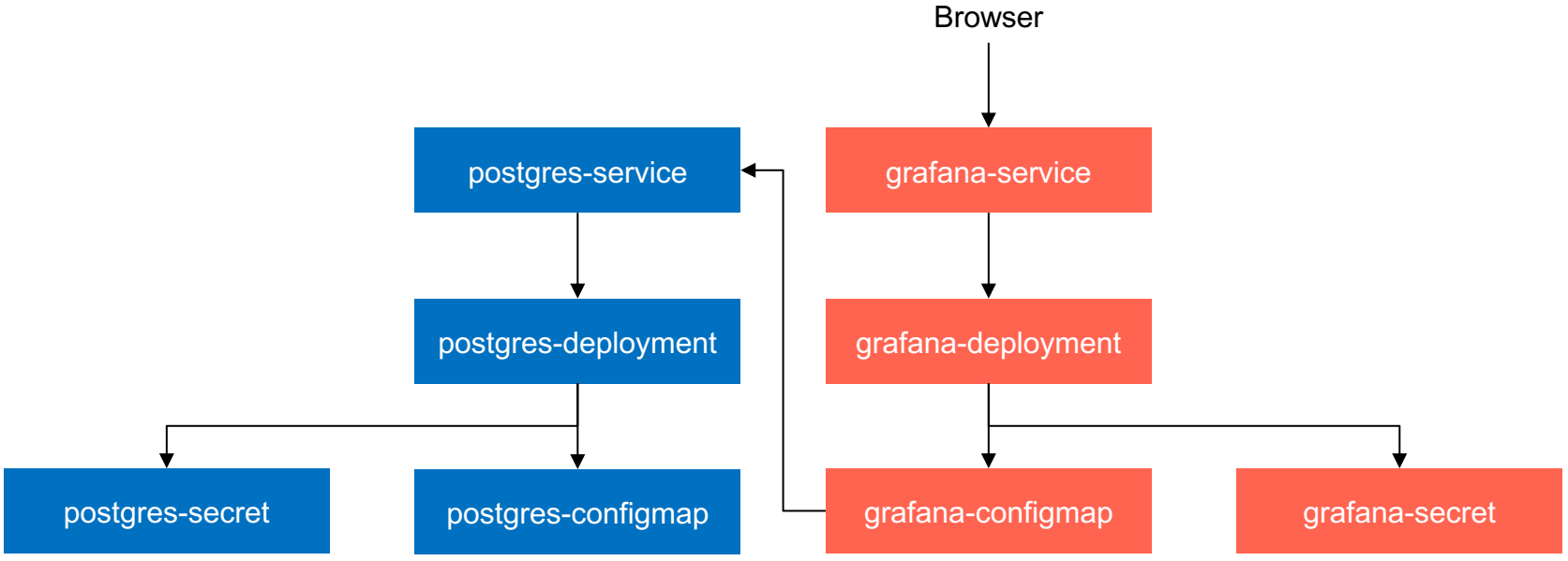
Konfiguration (beim Erzeugen von Pods)

- Credentials
- Datenquellen
- Dashboards

Initiale Credentials festlegen



Datenquelle und Dashboard vorkonfigurieren



Provisioning System (Grafana)

Konfiguration über Config-Files
https://grafana.com/docs/grafana/latest/administration/provisioning

Datasource

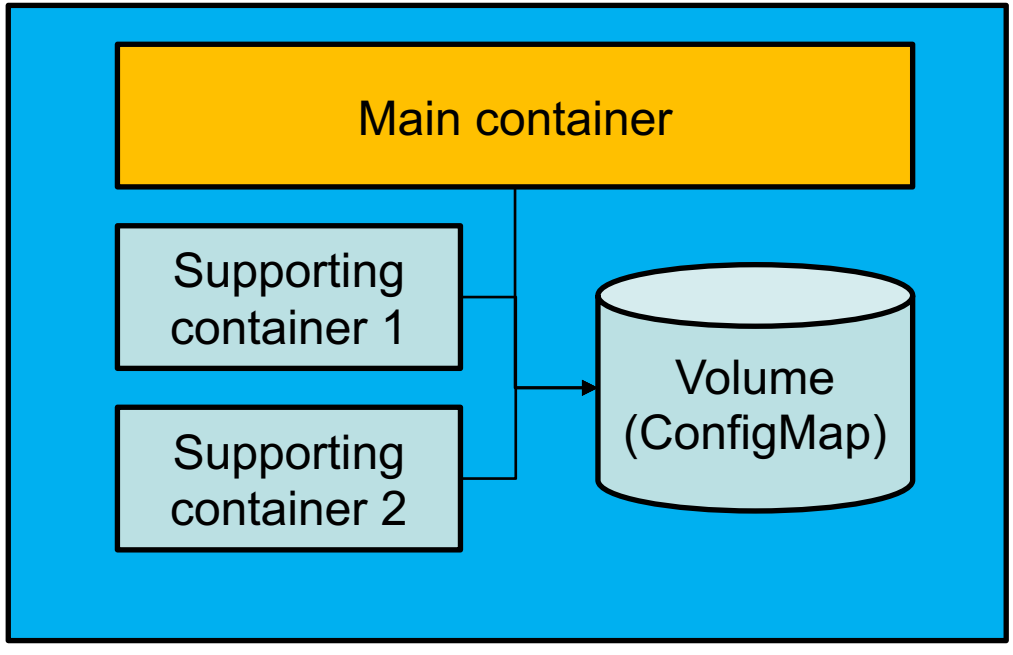
- /etc/grafana/provisioning/datasources/datasource.yml

Dashboard

- /etc/grafana/provisioning/dashboards/dashboard-providers.yml
- /var/lib/grafana/dashboards/dashboard.json

→ ConfigMaps können Config-Files enthalten

ConfigMap als Volume in einem Pod



Bereitstellung von Config-Dateien über ConfigMaps

1. ConfigMap mit Config-Files erstellen
2. ConfigMap als Volume in **Pod** einbinden (key=filename)
3. Dateien aus ConfigMap mittels Key im **Container** mounten

ConfigMaps erzeugen

kubectl create configmap

- from-literal
- from-env-file
- from-file (File, Directory)

Generatoren

- kustomization.yaml

Secrets erzeugen

siehe ConfigMaps

kubectl create secret <type>

- generic: beliebige Keys
- tls: Keys für TLS-Zertifikate
- docker-registry : Keys für Docker-Credentials

Umgang mit Secrets

Secrets enthalten sensible Daten

Dringend empfohlen:

- nicht in VCS hochladen
- nicht in Logs ausgeben
- sorgfältige Vergabe von Zugriffsrechten: z.B. Zugriff auf Pod ermöglicht Auslesen von Secrets

Takeaways

- ConfigMaps und Secrets enthalten Konfigurationsdaten
- Secrets enthalten sensible Daten
- Key-Value-Paare können in Pods eingebracht werden als:
  - Umgebungsvariablen
  - Dateien
- ConfigMaps und Secrets können als Volumes verwendet werden
- Umgang mit Secrets erfordert Sorgfalt

Misc

Materialien zum Talk - NEU

https://github.com/deepshore/deeptalk

Weiterführendes

Postgres: https://www.postgresql.org/

Grafana: https://grafana.com/docs/grafana/latest/administration/provisioning/

Kubernetes: https://kubernetes.io/docs/home/

Feedback , Anregungen, Themenvorschläge

florian.boldt@deepshore.de

malte.groth@deepshore.de

frederic.born@deepshore.de

Ausblick

Problem

Daten sollen erhalten bleiben – Pods sind aber flüchtig

→ Thema: Persistenz in Kubernetes

Anknüpfungspunkte

- Volumes
- Datenbanken

DEEPSHORE

Vielen Dank.