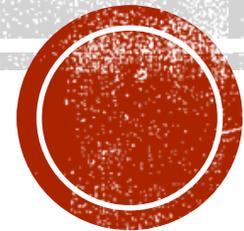


MINIKUBE CLUSTER INFRASTRUKTUR



EINLEITUNG

Die Installation und Konfiguration einer Minikube-Cluster-Infrastruktur mit wichtigen Diensten wie Cert-Manager, Prometheus, Redpanda und RabbitMQ Nach dem Umbau und updates für Redpanda und kubernetes Versionen.

KUBERNETES

Definition:

Kubernetes ist eine Open-Source-Plattform zur Automatisierung der Bereitstellung, Skalierung und Verwaltung von containerisierten Anwendungen.

Hauptkomponenten:

- **Pods:** Kleinste und einfachste Objekte in Kubernetes, die einen oder mehrere Container enthalten.
- **Nodes:** Maschinen (physisch oder virtuell), auf denen Kubernetes läuft und die Pods hosten.
- **Cluster:** Eine Gruppe von Nodes, die zusammenarbeiten.

`USE_KUBERNETES_VERSION = 1.24.0 => USE_KUBERNETES_VERSION = 1.28.3`



REDPANDA

Definition:

Redpanda ist eine Kafka-kompatible Streaming-Plattform, die entwickelt wurde, um eine hohe Leistung und einfache Bedienung zu bieten.

Hauptmerkmale:

- **Kafka-Kompatibilität:** Redpanda verwendet dieselben APIs wie Apache Kafka, sodass bestehende Anwendungen ohne Änderungen verwendet werden können.
- **Leistung:** Redpanda ist in C++ geschrieben und bietet eine hohe Durchsatzleistung und geringe Latenz.

REDPANDA_VERSION="v22.2.6" => REDPANDA_VERSION="v2.1.17-23.3.11"



VORTEILE DER KOMBINATION VON KUBERNETES UND REDPANDA

- **Automatisierung und Skalierbarkeit:**

Das Nutzen vom Kubernetes, um Redpanda-Cluster automatisch zu skalieren und zu verwalten.

- **Hohe Verfügbarkeit:**

Kubernetes sorgt für hohe Verfügbarkeit und Self-Healing von Redpanda-Deployments.

- **Einfache Verwaltung:**

Mit Kubernetes kann man komplexe Infrastruktur einfach deklarativ verwalten.

- **Leistungsstarke Datenverarbeitung:**

Kombination von der hohen Leistung von Redpanda mit der Flexibilität und Skalierbarkeit von Kubernetes.



REDPANDA KONFIGURATIONSDATEI

```
apiVersion: cluster.redpanda.com/v1alpha1
kind: Redpanda
metadata:
  name: msd-redpanda
spec:
  chartRef: {}
  clusterSpec:
    external:
      enabled: false    => Externe Zugriffe sind deaktiviert, was für mehr Sicherheit sorgt
    console:
      enabled: false    => Die Redpanda-Konsole ist deaktiviert, um den Ressourcenverbrauch zu minimieren.
    tls:
      enabled: false    => TLS-Verschlüsselung ist deaktiviert
    statefulset:
      replicas: 1.      => Das Repräsentiert Anzahl der Replicate. In unseren Fall brauchen wir nur 1
    initContainers:
      setDataDirOwnership:
        enabled: true.  => Initialisierungscontainer, der sicherstellt, dass die Besitzrechte des Datenverzeichnisses korrekt gesetzt sind.
  config:
    cluster:
      auto_create_topics_enabled: true
```

Ermöglicht die automatische Erstellung von Topics, was die Verwaltung vereinfacht.



ZUSAMMENFASSUNG DER VORTEILE

Sicherheit und Einfachheit:

Deaktivierte externe
Zugriffe, Konsole und TLS
für einfache und sichere
Konfiguration.

Einfache Verwaltung:
Automatische Topic-
Erstellung erleichtert die
Nutzung.

Initialisierungscontainer:
Stellt sicher, dass die
Dateiberechtigungen
korrekt gesetzt sind, was für
den reibungslosen Betrieb
wichtig ist.



GENERELLE ZUSAMMENFASSUNG

- Helm-Repositories wurden hinzugefügt und aktualisiert
- Cert-Manager und Prometheus wurden installiert
- Redpanda wurde konfiguriert und installiert
- RabbitMQ und der RabbitMQ-Kafka Connector wurden installiert

Das Ganze baut für uns ein gutes Environment für unsere Minikube Cluster.