

Installation

Presto steht als kostenlose Installation zur Verfügung. Im folgenden wird die Installation mittels Docker gezeigt.

Die offizielle Dokumentation kann hier nachgelesen werden: [Deploy Presto From a Docker Image](#)

Für Presto werden neben der Docker Compose Konfigurationsdatei noch weitere Konfigurationen benötigt, deswegen am besten wie immer einen eigenen Ordner erstellen, in den alle Konfigdateien kopiert werden.

Zuerst die folgende docker-compose.yml

```
presto:
  image: prestodb/presto
  container_name: presto
  volumes:
    - /pfad/zu/presto/config.properties:/opt/presto-server/etc/config.properties
    - /pfad/zu/presto/jvm.config:/opt/presto-server/etc/jvm.config
    - /pfad/zu/presto/catalog:/opt/presto-server/etc/catalog
  ports:
    - 8080:8080
  restart: unless-stopped
```

In der Konfiguration sind bereits 2 Dateien und ein Ordner angegeben. In den Konfigurationsdateien sind einige Einstellungen hinterlegt, wie z. B. der Port und über welchen Link Presto erreichbar sein soll. Für ein produktives Setup müssten noch einige Einstellungen mehr vorgenommen werden, aber zum Testen reichen diese bereits aus.

config.properties

```
coordinator=true
node-scheduler.include-coordinator=true
http-server.http.port=8080
discovery-server.enabled=true
discovery.uri=http://localhost:8080
```

jvm.config

```
-server
-Xmx2G
-XX:+UseG1GC
-XX:G1HeapRegionSize=32M
-XX:+UseGCOverheadLimit
-XX:+ExplicitGCInvokesConcurrent
-XX:+HeapDumpOnOutOfMemoryError
-XX:+ExitOnOutOfMemoryError
-Djdk.attach.allowAttachSelf=true
```

Nachdem die beiden Konfigdateien im selben Ordner erstellt wurden wie die Compose Konfiguration, ist noch der Ordner *catalog* anzulegen. In diesen Ordner wird nun für jede Datenquelle eine eigene Konfiguration hinterlegt.

Folgend ein paar Beispiele (die Dateien bitte in dem Ordner *catalog* speichern).

In den folgenden Konfigurationen bitte unbedingt auf Leerzeichen achten. Diese müssen entfernt werden, denn sonst wird z. B. der Connector nicht erkannt, wenn sich hinter dem Namen ein Leerzeichen befindet.

dummy.properties: Diese Konfiguration ist besonders, da sie keine Datenquelle definiert. Aber es können z. B. Views in diesem Katalog erstellt werden, sodass sich Views über mehrere Datenquellen hinterlegen lassen und dann z. B. in Metabase geladen werden können. In Metabase wird dann einfach dieser Dummy Katalog als Datenquelle hinterlegt und Metabase merkt gar nicht, dass die Daten aus verschiedenen Datenbanken stammen.

```
connector.name=memory
```

mysql.properties: Eine Konfiguration für einen MySQL Server, bitte die Daten an die eigene Umgebung anpassen.

```
connector.name=mysql
connection-url=jdbc:mysql://mysql:3306
connection-user=presto_user
connection-password=HierDasPasswort
case-insensitive-name-matching=true
```

sqlserver.properties: Eine Konfiguration für einen Microsoft SQL Server, bitte die Daten an die eigene Umgebung anpassen.

```
connector.name=sqlserver
connection-url=jdbc:sqlserver://sql:1433;databaseName=AdventureWorks2022;trustServerCertificate=true;
```

```
connection-user=presto_user  
connection-password=HierDasPasswort
```

Presto unterstützt natürlich noch viele weitere Datenbanken.

Hinweis: Wenn neue Kataloge hinzugefügt werden, muss Presto neugestartet werden.

Für eine Übersicht sowie Einrichtungshinweise am besten in die offizielle Dokumentation schauen:

Connectors

Sobald alles eingestellt ist, kann der Docker Container erstellt und gestartet werden.

Um die Docker Compose Konfiguration auszuführen, kann am besten in das Verzeichnis der YAML Datei gewechselt werden. Danach wird je nach gewählter Installation `sudo docker-compose up -d` oder `sudo docker compose up -d` (keine Bindestrich zwischen docker und compose) eingegeben, um die Standard Konfiguration `docker-compose.yml` zu starten. Compose erstellt dann die gewünschten Container mit den angegebenen Optionen. Sollten die Container bereits mit dieser Compose Konfiguration erstellt worden sein, so werden die Container in dieser neu erstellt, dessen Konfiguration geändert wurde.

Anschließend ist Presto unter <http://localhost:8080/> erreichbar.

Version #4

Erstellt: 24 Juli 2024 19:21:39 von Marcel

Zuletzt aktualisiert: 24 Juli 2024 21:05:32 von Marcel