Container 'matrix-synapse'

Ressourcen

- 4 GB RAM
- 4 Cores
- 5 GB HDD (root-fs)
- 25 GB HDD (/var/www)

System

- interne IPs
  - 10.2.0.16, fd00:10:2:0::16
  - 10.3.0.16, fd00:10:3:0::16

Dienste

- matrix-synapse

Installation

- Standard-Template mit Benutzern

Matrix Synapse

1. Matrix-Synapse Repository hinzufügen
   - **sudo apt-get install -y lsb-release wget apt-transport-https**
   - **sudo wget -O /usr/share/keyrings/matrix-org-archive-keyring.gpg https://packages.matrix.org/debian/matrix-org-archive-keyring.gpg**
   - **echo "deb [signed-by=/usr/share/keyrings/matrix-org-archive-keyring.gpg] https://packages.matrix.org/debian/ $(lsb_release -cs) main" | sudo tee /etc/apt/sources.list.d/matrix-org.list**
   - **sudo apt-get update**
2. Matrix-Synapse installieren
   - **sudo apt-get install matrix-synapse-py3**
3. Konfigurationen erstellen

   ```
   FIXME
   ```

4. Dienst aktivieren und neustarten
   - **sudo systemctl enable matrix-synapse.service**
   - **sudo systemctl restart matrix-synapse.service**

Cronjobs zur Bereinigung von Matrix Synapse

Media Cleanup

Der Media Cleanup bereinigt Mediendateien, die über Matrix geteilt wurden und ein bestimmtes Alter erreicht haben. Die aktuelle Schwelle liegt bei **2 Jahren**. Mediendateien, die älter sind, werden entfernt.

1. Access Token für Matrix Administrator lokal erzeugen
   - **curl -XPOST -d '{"type":"m.login.password", "user":"USERNAME", "password":"PASSWORD"}' "http://127.0.0.1:8008/_matrix/client/r0/login"**
2. Skript hinzufügen und generierten Token verwenden

   ```
   #!/bin/bash

   TOKEN="XXXX"
   ```

```
PROTOCOL="http"
SERVER="127.0.0.1"
PORT=8008

# Calculate date
TIMESTAMP=$(date +%s000 --date "2 years ago")

# Build API URLs
API_URL_MEDIA_LOCAL="$PROTOCOL://$SERVER:$PORT/_synapse/admin/v1/media/delete?before_ts
=$TIMESTAMP"
API_URL_MEDIA_CACHE="$PROTOCOL://$SERVER:$PORT/_synapse/admin/v1/purge_media_cache?befo
re_ts=$TIMESTAMP"

# Check requiirements
if ! which jq > /dev/null; then
  echo "jq not found. Please install required package!"
  exit 1
fi

# Log
log(){
  >&2 echo "${@:2}"
  echo "${@:2}" | logger -i -t synapse_media_cleanup -p local0.$1
}

# Log error message
logError(){
  log err "$@"
}

# Log info message
logInfo(){
  log info "$@"
}

# Clean up local media
echo "Clean up local media ..."
RESPONSE=$(curl --silent --fail-with-body --header "Authorization: Bearer $TOKEN" -X
POST $API_URL_MEDIA_LOCAL)
if [[ $? -ne 0 ]]; then
  # Curl error
  logError "Request to API failed: ${RESPONSE}"
else
  logInfo "Removed $(echo $RESPONSE | jq '.total') entries from local media"
fi

# Clean up media cache
echo "Clean up media cache ..."
RESPONSE=$(curl --silent --fail-with-body --header "Authorization: Bearer $TOKEN" -X
POST $API_URL_MEDIA_CACHE)
if [[ $? -ne 0 ]]; then
  # Curl error
  logError "Request to API failed: ${RESPONSE}"
else
  logInfo "Removed $(echo $RESPONSE | jq '.deleted') entries from media cache"
fi
```

3. Berechtigungen für Skript anpassen
    - **chmod 700 /usr/local/bin/synapse_media_cleanup.sh**
4. Cronjob erstellen
    - **crontab -e**

```
0 2 * * *       /usr/local/bin/synapse_media_cleanup.sh >
/var/log/synapse_media_cleanup.log 2>&1
```

State Compressor

Der State Compressor reduziert die Einträge in der Tabelle state_groups_state in PostgreSQL und hält so die Datenbank kompakt.

1. Build des State Compressors auf einer externen VM mit gleichem OS wie der Server matrix-synapse nach Anleitung
2. Ordner erstellen
   - **mkdir /usr/local/bin/compressor**
3. Dateien **synapse_auto_compressor/target/debug/synapse_auto_compressor** und **synapse_auto_compressor/target/debug/synapse_compress_state** in den **Ordner /usr/local/bin/compressor** übertragen
4. Skript erstellen

```
#!/bin/bash

# Run compressor with CHUNK_SIZE=1000 and CHUNKS_TO_COMPRESS=10000000
/usr/local/bin/compressor/synapse_auto_compressor -p postgresql://PGSQL-USER:"PGSQL-
PASS"@@10.3.0.102/synapse -c 1000 -n 1000000
```

5. Berechtigungen für Skript anpassen
   - **chmod 700 /usr/local/bin/compressor/synapse_auto_compressor.sh**
6. Cronjob erstellen
   - **crontab -e**

```
0 4 * * *       /usr/local/bin/compressor/synapse_auto_compressor.sh >
/var/log/synapse_auto_compressor.log 2>&1
```

Backup mit Borgmatic

1. Installation siehe mariadb
2. Konfiguration

```
...
location:
    # List of source directories to backup (required). Globs and
    # tildes are expanded.
    source_directories:
        - /etc
        - /home
        - /root
        - /var/log
        - /var/lib/matrix-synapse
...
```

Dauerhafter Link zu diesem Dokument:
https://wiki.technikkultur-erfurt.de/dienste:bytecluster0002:matrix-synapse

Dokument zuletzt bearbeitet am: **03.11.2026 08:47**

Verein zur Förderung von Technikkultur in Erfurt e.V
https://wiki.technikkultur-erfurt.de/