

Restore Log Data from AWS S3 into Loki

Docker Compose

```
version: '3.8'
services:
  s3-downloader:
    build: .
    volumes:
      - ./logs:/logs
      - ./s3cfg:/root/.s3cfg
    entrypoint: ["/usr/local/bin/download_logs.sh"]
```

Dockerfile

```
# Use an official lightweight image with bash
FROM alpine:3.17

# Install necessary packages (bash, s3cmd, etc.)
RUN apk add --no-cache bash py3-pip && pip install s3cmd

# Copy the Bash script into the container
COPY download_logs.sh /usr/local/bin/download_logs.sh

# Set executable permissions on the script
RUN chmod +x /usr/local/bin/download_logs.sh

# Set the default command to run the Bash script
# CMD ["/usr/local/bin/download_logs.sh"]
```

AWS S3 Credential

```
[default]
access_key = xyz
secret_key = ***
```

Log Data Extractor

```
#!/bin/bash
set -e

# Configure s3cmd if not already configured
if [ ! -f "$HOME/.s3cfg" ]; then
    echo "s3cmd configuration not found."
    exit 1
fi

# S3 Bucket and path
LOG_DATE="20241118"
NAMESPACE="app1-ns"
BUCKET_NAME="log-data-bucket"
REMOTE_LOGS_PATH="fluentd_log/$LOG_DATE/$NAMESPACE"
LOCAL_DESTINATION="/logs"

# Array of patterns to include
PATTERNS=(
    "metadata-server-*.tar.gz"
    "admin-server-*.tar.gz"
)

# Create local destination folder if it doesn't exist
mkdir -p "$LOCAL_DESTINATION"

# Build the include and exclude rules dynamically
INCLUDE_EXCLUDE=""
for PATTERN in "${PATTERNS[@]}; do
    INCLUDE_EXCLUDE+="--include=\"$PATTERN\" "
```

```
done
INCLUDE_EXCLUDE+="--exclude=\"*\"\"
echo $INCLUDE_EXCLUDE

# Logic to download logs
echo "Downloading log files from s3://$BUCKET_NAME/$REMOTE_LOGS_PATH to $LOCAL_DESTINATION"
# eval s3cmd sync -v $INCLUDE_EXCLUDE s3://$BUCKET_NAME/$REMOTE_LOGS_PATH "$LOCAL_DESTINATION"

echo "Log files downloaded successfully to $LOCAL_DESTINATION"

# Extract each .tar.gz file into its own folder
echo "Extracting downloaded files..."
for FILE in "$LOCAL_DESTINATION/$NAMESPACE"/*.tar.gz; do
    if [ -f "$FILE" ]; then
        # Create a folder named after the file (without .tar.gz)
        FOLDER_NAME="${FILE%.tar.gz}"
        mkdir -p "$FOLDER_NAME"

        # Get list of files in the tar.gz archive
        ARCHIVE_CONTENTS=$(tar -tzf "$FILE" | grep -v '/$')
        SUCCESS=true

        # Check each file in the archive exists in the extraction folder
        for ARCHIVE_FILE in $ARCHIVE_CONTENTS; do
            FULL_PATH="$LOCAL_DESTINATION/$NAMESPACE/$ARCHIVE_FILE"
            if [ ! -f $FULL_PATH ]; then
                echo "Error: File $ARCHIVE_FILE is missing in $FOLDER_NAME"
                SUCCESS=false
                break
            fi
        done

        if $SUCCESS; then
            echo "All files successfully extracted for $FILE."
        else
            # Extract the tar.gz file into the folder
            tar -xzf "$FILE" -C "$LOCAL_DESTINATION/$NAMESPACE"

            echo "Extracted $FILE into $FOLDER_NAME"
        fi
    fi
fi
```

done

echo "All files downloaded and extracted successfully!"

Revision #1

Created 26 November 2024 15:04:50 by Ahmad

Updated 26 November 2024 15:09:40 by Ahmad