Release H: Minio Events

Introduction

MinIO bucket notifications allows user to send notifications on bucket events to external services

Notifications to Postgres

To send bucket notifications to Postgres you first need to setup the following environment variables:

Minio ENV

```
        - name: MINIO_NOTIFY_POSTGRES_ENABLE_PRIMARY
          value: "on"
        - name: MINIO_NOTIFY_POSTGRES_CONNECTION_STRING_PRIMARY
          value: "postgresql://minio:minio@postgres.default:5432/minio?sslmode=disable"
        - name: MINIO_NOTIFY_POSTGRES_TABLE_PRIMARY
          value: "minioevents"
        - name: MINIO_NOTIFY_POSTGRES_FORMAT_PRIMARY
          value: "namespace"
        - name: MINIO_NOTIFY_POSTGRES_MAX_OPEN_CONNECTIONS_PRIMARY
          value: "2"
        - name: MINIO_NOTIFY_POSTGRES_QUEUE_DIR_PRIMARY
          value: "/opt/minio/events"
        - name: MINIO_NOTIFY_POSTGRES_QUEUE_LIMIT_PRIMARY
          value: "100000"
        - name: MINIO_NOTIFY_POSTGRES_COMMENT_PRIMARY
          value: "PostgreSQL Notification Event Logging for MinIO"
```

In this example I have used an identifier of PRIMARY but you can use whatever you prefer.

Note: I included the "sslmode=disable" parameter in the Postgres connection string as I'm not running Postgres over HTTPS.

When Minio starts you should see the following line in your log: SQS ARNs: arn:minio:sqs::PRIMARY:postgresql.

This indicates event notifications to Postgres are available.

The next thing you need to do is enable eventing on whatever bucket you want to monitor.

```
 /minio$ mc event add myminio/py-bucket arn:minio:sqs::PRIMARY:postgresql
Successfully added arn:minio:sqs::PRIMARY:postgresql


/minio$ mc event list  myminio/py-bucket arn:minio:sqs::PRIMARY:postgresql
arn:minio:sqs::PRIMARY:postgresql   s3:ObjectCreated:*,s3:ObjectRemoved:*,s3:ObjectAccessed:*   Filter:
```

The second command lists the events Minio will notify on.

When you upload a file:

```
/minio$ mc cp test.txt myminio/py-bucket
.../go/minio/test.txt:  13 B / 13 B  687 B/s 0s
```

You see a new record in the minioevents table in your Postgres database

Key	Value

py-bucket/test.txt	{"Records": [{"s3": {"bucket": {"arn": "arn:aws:s3:::py-bucket", "name": "py-bucket", "ownerIdentity": {"principalId": "minio"}}, "object": {"key": "test.txt", "eTag": "8ddd8be4b179a529afa5f2ffae4b9858", "size": 13, "sequencer": "17552EACF7D25CF0", "contentType": "text/plain", "userMetadata": {"content-type": "text/plain"}}, "configurationId": "Config", "s3SchemaVersion": "1.0"}, "source": {"host": "172.17.0.1", "port": "", "userAgent": "MinIO (linux; amd64) minio-go/v7.0.31 mc/RELEASE.2022-07-24T02-25-13Z"}, "awsRegion": "", "eventName": "s3:ObjectCreated:Put", "eventTime": "2023-04-12T12:09:16.319Z", "eventSource": "minio:s3", "eventVersion": "2.0", "userIdentity": {"principalId": "minio"}, "responseElements": {"content-length": "0", "x-amz-request-id": "17552EACF779E1C4", "x-minio-deployment-id": "689c3baf-ebc6-4df8-ae4e-a09b333ed9b3", "x-minio-origin-endpoint": "http://172.17.0.12:9000"}, "requestParameters": {"region": "", "principalId": "minio", "sourceIPAddress": "172.17.0.1"}}]}

Note: The minioevents table will be created if it does not already exists

Notifications to Kafka

To send bucket notifications to Kafka you first need to setup the following environment variables (if you are using tls authentication):

Minio ENV

```
        - name: MINIO_NOTIFY_KAFKA_ENABLE_PRIMARY
          value: "on"
        - name: MINIO_NOTIFY_KAFKA_BROKERS_PRIMARY
          value: "my-cluster-kafka-external-0.kafka:9098"
        - name: MINIO_NOTIFY_KAFKA_TOPIC_PRIMARY
          value: "my-topic"
        - name: MINIO_NOTIFY_KAFKA_TLS_PRIMARY
          value: "on"
        - name: MINIO_NOTIFY_KAFKA_TLS_SKIP_VERIFY_PRIMARY
          value: "on"
        - name: MINIO_NOTIFY_KAFKA_CLIENT_TLS_CERT_PRIMARY
          value: "/etc/kafka/ssl/user.crt"
        - name: MINIO_NOTIFY_KAFKA_CLIENT_TLS_KEY_PRIMARY
          value: "/etc/kafka/ssl/user.key"
```

Note: You will also need to copy the required tls certs into a secret and mount them somewhere in the Minio pod.

In this example I have used an identifier of PRIMARY but you can use whatever you prefer.

When Minio starts you should see the following line in your log: SQS ARNs: arn:minio:sqs::PRIMARY:kafka

This indicates event notifications to Kafka are available.

The next thing you need to do is enable eventing on whatever bucket you want to monitor.

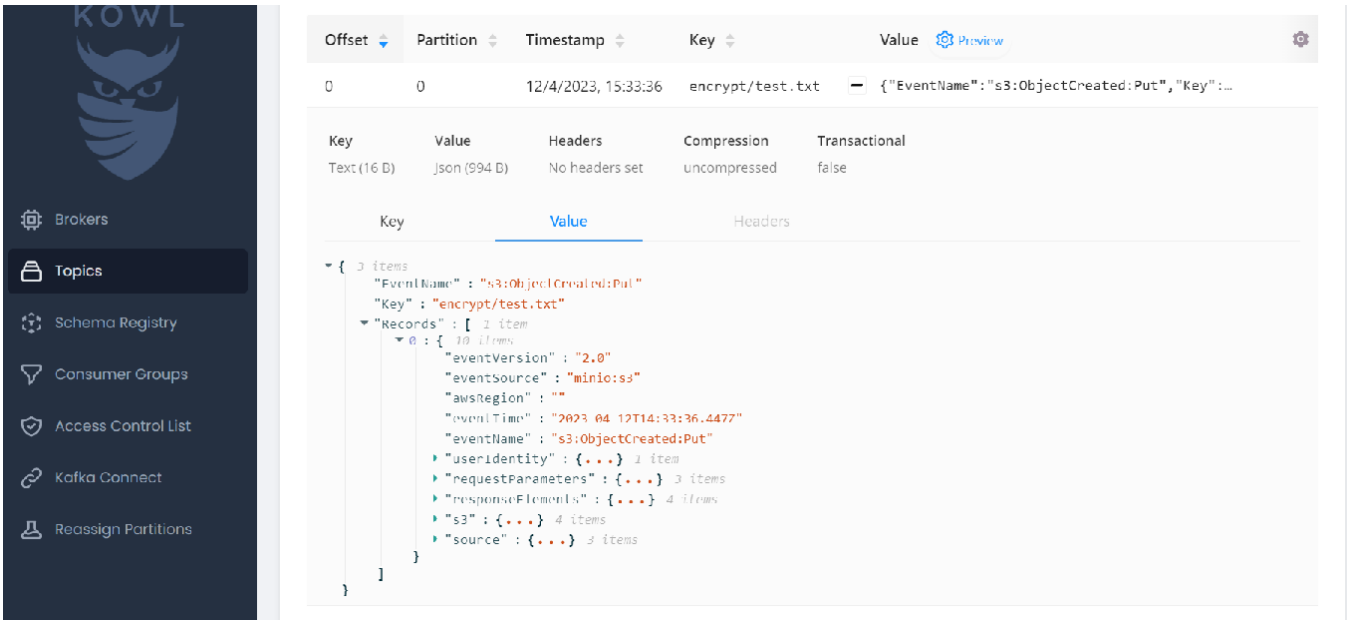
```
 /minio$ mc event add myminio/encrypt arn:minio:sqs::PRIMARY:kafka
Successfully added arn:minio:sqs::PRIMARY:kafka

/minio$ mc event list myminio/encrypt arn:minio:sqs::PRIMARY:kafka
arn:minio:sqs::PRIMARY:kafka   s3:ObjectCreated:*,s3:ObjectRemoved:*,s3:ObjectAccessed:*   Filter:
```

The second command lists the events Minio will notify on.

When you upload a file:

```
/minio$ mc cp test.txt myminio/encrypt
.../go/minio/test.txt:  13 B / 13 B  78 B/s 0s
```

You see a new record in "my-topic"



Links

Monitoring Bucket and Object Events

MinIO Bucket Notification Guide