Dremio Query Analyzer on Kubernetes (Azure, AWS, GCP)

Purpose of Dremio Query Analyzer

The Query Analyzer provides capabilities to analyze queries which run in Dremio. This allows the administrator to:

- Analyze slow queries
- Identify queries which could be reflected
- User generating most load on the system
- Optimize Workload Management settings in Dremio
- Get a kind of audit log for queries
- Many more

Required Images

The following Docker images will be required:

Image	Description
dremiops.azurecr.io/dremio-query-analyzer:1.20	Runs the job to fetch queries.json from coordinator node regularly
dremiops.azurecr.io/dremio-query-analyzer-vdscreator:1.4	This imports the VDS definitions into Dremio, which are required for the analyzes. **This is a one time job for the setup process!**

Prerequisites

It is required that logging is enabled on the coordinator node. The `queries.json` file needs to be available under:

```
/opt/dremio/data/log/queries.json
```

The logging can be enabled by adding the following to the `values.yml` file:

```
coordinator:
  extraStartParams: >-
   -Ddremio.log.path=/opt/dremio/data/log
```

Then run the command:

```
$ helm upgrade dremio -f values.yml
```

WARNING: With Dremio 20, the latest Helm Charts need to be used, otherwise the logging is broken for `queries.json`.

Setup Query Analyzer

Setup Kubernetes Service Account

Dremio Query Analyzer needs to access the Dremio Coordinator pod remotely so that it will be able to copy the `queries.json` file into the pod of the Query Analyzer pod. This requires a service account and some permissions:

```
apiVersion: rbac.authorization.k8s.io/v1
kind: Role
metadata:
  name: dremio-jobs-role
rules:
- apiGroups: [""]
  resources: ["pods", "pods/exec"]
  verbs: ["get", "list", "watch", "create"]

---

apiVersion: v1
kind: ServiceAccount
```

```
metadata:
  name: dremio-jobs

---

apiVersion: rbac.authorization.k8s.io/v1
kind: RoleBinding
metadata:
  name: dremio-jobs
subjects:
- kind: ServiceAccount
  name: dremio-jobs
roleRef:
  kind: Role #this must be Role or ClusterRole
  name: dremio-jobs-role
  apiGroup: rbac.authorization.k8s.io
```

Put the yaml specification into a file called `dremio-jobs-serviceaccount.yml`. No changes need to be applied to the definition. Then run the following command:

```
$ kubectl apply -f dremio-jobs-serviceaccount.yml
```

Create a Dedicated ADLS Container (Azure)

This step is only for Azure and ADLS.

Go to the Azure console and select your storage account, then add a container:

Type in the container name “dremio-query-analyzer” and press “Create”:

Check for successful creation:

Generate SAS URL

The SAS Url is required in the next step “Setup Kubernetes CronJob”. Select the container “dremio-query-analyzer” and select “Shared Access Token”:

Select “Read, Add, Create, Write, Delete and List” from Permissions list:

Select an expiry date, e.g. in two years and don’t forget to set a reminder to regenerate it:

Press “Generate SAS token and URL” and you get the following screen:

Copy the “Blob SAS URL”, this one is required for the next step.

Setup Kubernetes CronJob for Azure

Then the Query Analyzer CronJob needs to be set up. In the example below, it would run once a day. Replace the following in the Yaml specification:

- `<USERNAME>` Is the Dremio Username
- `<PAT_TOKEN>` Could be the password, but it is more secure to use a PAT token
- `<SAS_URL>` The SAS Url which was generated in the previous step

The Yaml specification for the Query Analyzer CronJob:

```
apiVersion: batch/v1
kind: CronJob
metadata:
 name: dremio-query-analyzer
spec:
 schedule: "10 5 * * *"
 jobTemplate:
   spec:
     template:
       spec:
         serviceAccountName: dremio-jobs
         automountServiceAccountToken: true
         containers:
         - name: dremio-query-analyzer
```

```
          image:  dremiops.azurecr.io/dremio-query-analyzer:1.20
          imagePullPolicy: Always
          env:
            - name: DREMIO_STORAGE_TYPE
              value: adls
            - name: DREMIO_USERNAME
              value: "<USERNAME>"
            - name: DREMIO_PASSWORD
              value: "<PAT_TOKEN>"
            - name: AZURE_SAS_URL
              value: "<SAS_URL>"
        restartPolicy: OnFailure
```

Replace the fields as described above and put the specification into a file called *dremio-query-analyzer.yml*. Then run the following command:

```
$ kubectl apply -f dremio-query-analyzer.yml
```

WARNING: Do not run the job too often, since it can put load on the coordinator node. Probably, 4 times a day should be the max.

Setup Kubernetes CronJob for AWS

Then the Query Analyzer CronJob needs to be set up. In the example below, it would run once a day. Replace the following in the Yaml specification:

- `<USERNAME>` Is the Dremio Username
- `<PAT_TOKEN>` Could be the password, but it is more secure to use a PAT token
- `<S3_ACCESS_KEY_ID>` S3 Access Key Id
- `<S3_SECRET_ACCESS_KEY>` S3 Secret Access Key
- `<S3_BUCKET_NAME>` S3 Bucket Name

The Yaml specification for the Query Analyzer CronJob:

```
apiVersion: batch/v1
kind: CronJob
metadata:
```

```
  name: dremio-query-analyzer
spec:
  schedule: "10 5 * * *"
  jobTemplate:
    spec:
      template:
        spec:
          serviceAccountName: dremio-jobs
          automountServiceAccountToken: true
          containers:
          - name: dremio-query-analyzer
            image:  dremiops.azurecr.io/dremio-query-analyzer:1.20
            imagePullPolicy: Always
            env:
              - name: DREMIO_STORAGE_TYPE
                value: s3
              - name: DREMIO_USERNAME
                value: "<USERNAME>"
              - name: DREMIO_PASSWORD
                value: "<PAT_TOKEN>"
              - name: AWS_ACCESS_KEY_ID
                value: "<S3_ACCESS_KEY_ID>"
              - name: AWS_SECRET_ACCESS_KEY
                value: "<S3_SECRET_ACCESS_KEY>"
              - name: DREMIO_STORAGE_PATH
                value: s3://<S3_BUCKET_NAME>/queryanalyzer
          restartPolicy: OnFailure
```

Replace the fields as described above and put the specification into a file called *dremio-query-analyzer.yml*. Then run the following command:

```
$ kubectl apply -f dremio-query-analyzer.yml
```

WARNING: Do not run the job too often, since it can put load on the coordinator node. Probably, 4 times a day should be the max.

Setup Kubernetes CronJob for Google Cloud (GCP)

Before the Query Analyzer gets setup in GCP, it is required to create a secret which contains the private keys for the GCP service principal. The key can be downloaded from the IAM page and then run the following command:

```
kubectl create secret generic gcp-service-principal-secret --from-file=gcp-service-principal.json=dremio-ps-f209e5d75609.
```

Then the Query Analyzer CronJob needs to be set up. In the example below, it would run once a day. Replace the following in the Yaml specification:

- `<USERNAME>` Is the Dremio Username
- `<PAT_TOKEN>` Could be the password, but it is more secure to use a PAT token
- `<SERVICE_ACCOUNT>` Service account name
- `<STORAGE_PATH>` Storage path for Query Analyzer

The Yaml specification for the Query Analyzer CronJob:

```
apiVersion: batch/v1
kind: CronJob
metadata:
  name: dremio-query-analyzer-gcp
spec:
  schedule: "10 5 * * *"
  jobTemplate:
    spec:
      template:
        spec:
          serviceAccountName: dremio-jobs
          automountServiceAccountToken: true
          containers:
          - name: dremio-query-analyzer
            image:  dremiops.azurecr.io/dremio-query-analyzer:1.20
            volumeMounts:
            - mountPath: /opt/dremio/conf
              name: gcp-service-principal-secret
              readOnly: true
            imagePullPolicy: Always
            env:
              - name: DREMIO_STORAGE_TYPE
                value: gcs
              - name: DREMIO_USERNAME
                value: "<USERNAME>"
              - name: DREMIO_PASSWORD
```

```
            value: "<PAT_TOKEN>"
          - name: GCS_SERVICE_PRINCIPAL
            value: <SERVICE_ACCOUNT>@dremio-ps.iam.gserviceaccount.com
          - name: DREMIO_STORAGE_PATH
            value: "gs://<STORAGE_PATH>"
      restartPolicy: OnFailure
      volumes:
        - name: gcp-service-principal-secret
          secret:
            secretName: gcp-service-principal-secret
```

Replace the fields as described above and put the specification into a file called *dremio-query-analyzer.yml*. Then run the following command:

```
$ kubectl apply -f dremio-query-analyzer.yml
```

WARNING: Do not run the job too often, since it can put load on the coordinator node. Probably, 4 times a day should be the max.

Trigger the CronJob for the first run

Since the CronJob is set to 5:10 am, it won't run immediately. It might be required to trigger the CronJob a first time, so that it generates initial data that we can access. This step is also required to run the VDS creator and execute the next step.

Run the following command:

```
$ kubectl create job --from=cronjob/dremio-query-analyzer dremio-qa-initial
```

This is a one time action.

Create Dremio Source and Promote PDS

Create QueriesJson Source

Create a new Source called `QueriesJson`. The name must be accurate and is case sensitive:

Set the "Root Path" in "Advanced Options" to "/dremio-query-analyzer" and press save:

Promote Query Analyzer PDS

Next step is to promote all folders which are located in the Source `QueriesJson`. This happens by pressing the marked folders icon on the right side:

Then you receive a dialog like below. Select JSON format, if not already selected and press “Save”:

Repeat the steps for

- chunks
- errormessages
- results
- badrows (optional)

After the promotion the folders should have received a purple icon which indicates that they have been promoted successfully:

Run VDS Creator

After the Source was created and the three VDS have been promoted, the VDS can be imported.

Replace the following in the Yaml specification:

- `<USERNAME>` Is the Dremio Username
- `<PAT_TOKEN>` Could be the password, but it is more secure to use a PAT token

The Yaml specification for the VDS importer:

```
apiVersion: batch/v1
kind: Job
metadata:
 name: dremio-vdscreator
spec:
 template:
   spec:
     containers:
```

```
      - name: job
        image: dremiops.azurecr.io/dremio-query-analyzer-vdscreator:1.4
        imagePullPolicy: IfNotPresent
        env:
        - name: DREMIO_ENDPOINT
          value: "http://dremio-client:9047"
        - name: DREMIO_USERNAME
          value: "<DREMIO_USERNAME>"
        - name: DREMIO_PASSWORD
          value: "<DREMIO_PATTOKEN>"
    restartPolicy: Never
backoffLimit: 4
```

Replace the fields as described above and put the specification into a file called `dremio-vds-creator.yml`. Then run the following command:

```
$ kubectl apply -f dremio-vds-creator.yml
```

NOTE: This is a one time job to create the structures in Dremio. After this is done it does not need to run anymore.

Dremio Query Analyzer VDS definitions

After logging in into Dremio you should see a new Space called `QueryAnalysis`. Also add an access key or alternatively an Azure Active Directory Configuration:

In the folder Application there should be many helpful VDS definitions available: