support-reporting-level1

December 17, 2024

0.1 load data

- `root`: location of the csv files.
- `fromDt` and `toDt` are start and end dates for range

[1]:

```
import sys
sys.path.append('../')

import dataframeLoader as dfl
import pandas as pd

from importlib import reload
reload(dfl)

# Provide csv data location and appliance and timerange information.
root = '../../.dataDir'
fromDt = '2024-11-10'
toDt = '2024-12-10'

# Provide list of prometheus metrics to load.
# metricsArr = ['cpu_used', 'download_workers_count', 'memory_used',␣
 ↪'task_queue_length', 'infra_access_latency', 'pod_cpu_usage',␣
 ↪'pod_memory_usage']
metricsArr = ['cpu_used'
              ,'task_queue_length'
              , 'memory_used'
              ]


daterange=[fromDt, toDt]
df = dfl.loadApplianceTimeSeriesData(root, metricsArr, daterange)
```

```
loading Unstrctured Data from file: SCANPROC-*.csv

loading Strctured Data from file: STRUCTURED-*.csv

processing securiti_appliance_cpu_used-max*.csv

processing securiti_appliance_cpu_used-avg*.csv

processing securiti_appliance_task_queue_length-max*.csv
processing securiti_appliance_task_queue_length-avg*.csv

/Users/amitgupta/code/jupyter-python-local-venv/examples/timeseries
reporting/../dataframeLoader.py:43: ParserWarning: Skipping line 232: expected 7
fields, saw 11

  df = pd.read_csv(os.path.join(path, name), on_bad_lines='warn')

processing securiti_appliance_memory_used-max*.csv

processing securiti_appliance_memory_used-avg*.csv

loading Unstrctured Data from file: UNSTRUCTURED-*.csv
```

[2]:

```
display(df)
print(df.metrics.unique())
```

```
                                 appliance_id                  ts  \
0        01c75278-9c0d-41be-b693-c970b18dbedc 2024-11-10 00:00:00
1        01c75278-9c0d-41be-b693-c970b18dbedc 2024-11-10 00:00:00
2        01c75278-9c0d-41be-b693-c970b18dbedc 2024-11-10 01:00:00
3        01c75278-9c0d-41be-b693-c970b18dbedc 2024-11-10 01:00:00
4        01c75278-9c0d-41be-b693-c970b18dbedc 2024-11-10 02:00:00
…                                         …                 …
3401636  ff0afca2-5070-4f49-8c4d-53a96baee027 2024-12-08 13:00:00
3401637  ff0afca2-5070-4f49-8c4d-53a96baee027 2024-12-09 07:00:00
3401638  ff0afca2-5070-4f49-8c4d-53a96baee027 2024-12-09 07:00:00
3401639  ff0afca2-5070-4f49-8c4d-53a96baee027 2024-12-09 12:00:00
3401640  ff0afca2-5070-4f49-8c4d-53a96baee027 2024-12-09 12:00:00

               legend               metrics      value
0        172.30.9.154           cpu_used_max  63.980000
1        172.30.9.157           cpu_used_max  92.830000
2        172.30.9.154           cpu_used_max  61.170000
3        172.30.9.157           cpu_used_max  98.290000
4        172.30.9.154           cpu_used_max  47.620000
…                 …                     …        …
3401636       s3_2452  fileDownloadTimeInHrs   0.011391
3401637       s3_2359  fileDownloadTimeInHrs   0.003330
3401638       s3_2360  fileDownloadTimeInHrs   0.002996
3401639      nan_2456  fileDownloadTimeInHrs   0.011397
3401640       s3_2452  fileDownloadTimeInHrs   0.009610

[3297712 rows x 5 columns]

['cpu_used_max' 'cpu_used_avg' 'task_queue_length_max'
 'task_queue_length_avg' 'memory_used_max' 'memory_used_avg'
 'dataScannedinGB' 'scanTimeInHrs' 'IdleTimeInHrs' 'numFilesScanned'
 'uniqPodCount' 'avgFileSizeInMB' 'numberOfColsScanned'
 'numberOfChunksScanned' 'fileDownloadTimeInHrs']
```

0.2 Generate plotly report

- `appliance_id`: unique identifier of the appliance.

[3]:

```
reload(dfl)
appliance_id='58e98e10-1b19-4c84-93c0-db2ad5903b80'
fromDate = '2024-11-23'
toDate = '2024-11-29'
dfp = df[(df['ts'].between(fromDate, toDate, inclusive='both'))]
fig  = dfl.plotMetricsFacetForApplianceId(dfp, appliance_id)
fig.show()
```