How to configure the central Grafana to access new data source

1. Open Grafana https://YOUR_GRAFANA_HOST/grafana/
2. Let's check what credentials to use to login to Grafana configuration section:
   2.1. Use local CLI (as root) in your Grafana host: `# psconfig agentctl grafana grafana-password`
   2.2. Username: `admin`
3. Login to Grafana
4. Let's define (new) data source
   4.1. Go to **Data Sources**
   4.2. Click **Add new data source**
   4.3. Search for **OpenSearch** type
   4.4. Fill:
      4.4.1. Name: SOME_NAME
      4.4.2. URL: https://YOUR_MA_HOST/opensearch/
      4.4.3. Turn on **Skip TLS Verify**
      4.4.4. Index name: `pscheduler*`
      4.4.5. Time field name: `pscheduler.start_time`
      4.4.6. Click **Get Version and Save**
      4.4.7. Click **Save & test**
5. Let's verify if we can access the data
   5.1. Go to **Explore**
      5.1.1. Select proper data source (e.g. previously defined) if not selected
      5.1.2. Metric: **Logs**
      5.1.3. Click **Run query**
      5.1.4. You get a list and graph of all records (dokuments) stored at each time period with all details. Look for different fields and familiarize
      5.1.5. Let's use some filtering with the Garafana query language Lucene
         5.1.5.1. Into **Query** field put: `test.type.keyword: throughput`
         5.1.5.2. Click **Run query**
         5.1.5.3. Explore other filters and concatenations with AND (letters must be capitalized!)

How to build a simple dashboard in the central Grafana

1. Let's Build a simple dashboard with some data source
   1.1. Create a new dashboard
      1.1.1. Go to **My Dashboards**
      1.1.2. Select **New -> New Dasboards**
      1.1.3. Save it under some name. Clik on the little floppy Save icon
         1.1.3.1. Put recognizable title for your dashboard
         1.1.3.2. Check if it is listed under **My perfSONAR Dashboards** folder

1.2. We need to add some graph. Let's say we want to graph throughput
    1.2.1. Click **Add visualization**
        1.2.1.1. Note there may be a default Grafana dashboard shown now with some random data
    1.2.2. Select you data source
    1.2.3. The default Time series graph shows count of data so we want to change it now
1.3. Let's add variables for dynamic selection of source and destination hosts
    1.3.1. Click on the small  icon on top of the dashboard page and go to dashboard settings
    1.3.2. Click **Variables** tab
        1.3.2.1. Click New variable
        1.3.2.2. Enter **Name:** `srchost`
        1.3.2.3. Enter **Label** "Source host"
        1.3.2.4. Choose **Data source**
        1.3.2.5. In the **Query** field enter: `{"find": "terms", "field": "meta.source.hostname.keyword"}`
        1.3.2.6. Choose **Refresh**: On time range change
        1.3.2.7. Click **Run query** to see results down the page
        1.3.2.8. Click **Apply**
    1.3.3. Create second variable for destination but add another condition to only show destinations that have data from source
        1.3.3.1. With **Query** `{"find": "terms", "field": "meta.destination.hostname.keyword", "query": "meta.source.hostname.keyword: $srchost"}`
    1.3.4. Click **Save dashboard** and **Close**
1.4. Let's create visualization now:
    1.4.1. Click on the three dots icon for you visualization graph in the dashboard to move to configuarion panels of your visualization
    1.4.2. Into **Query Lucene** field put: `test.type.keyword: throughput AND meta.source.hostname.keyword: $srclocal AND meta.destination.hostname.keyword: $dstlocal`
    1.4.3. You **MUST** use **AND** in capital letters
    1.4.4. We need to change **Metric** from **Count** to **Average** and select corresponding field `result.throughput`
    1.4.5. Click **Refresh** icon in the top of the graph
    1.4.6. Units are now in bps so let's change it
        1.4.6.1. Scroll down the right options section of the page to Unit and choose **Data rate -> bits/sec(SI)**
1.5. Play with **Title** or other visualization options
1.6. Click **Apply** button
1.7. Click **Save** icon to save dashboard and **Refresh** dashboard

2. Let's add reverse direction
    2.1. Go again to visualization panel edit section clicking on its three dots

2.2. Below the graph use **Add query** to add second Query (it will be named “B”)
2.3. Into **Query Lucene** field put: `test.type.keyword: throughput AND meta.source.hostname.keyword: $dstlocal AND meta.destination.hostname.keyword: $srclocal`
2.4. Other option will be the same as in Query A
3. Modify series names using variables
3.1. Go to **Overrides** tab
3.2. Choose **Add field override** and define:
3.2.1. From the list choose **Fields returned by query**
3.2.2. Fields: Query A
3.2.3. Click **Add override property**
3.2.3.1. Select **Standard options > Display name**
3.2.3.2. Put: `$srchost -> $dsthost`
3.2.4. Do opposite for Query B and opposite direction

In order to play with latency data use:

`test.type.keyword: latencybg` for latency tests. In the Lucene query options change **Metric** to **Min** and select field `result.latency.min` or **Max** and select field `result.latency.max`