Lesson 03 Demo 03

Automating Metric Pushing with a Cron Job

Objective: To automate metrics pushing from short-lived jobs to Pushgateway using a cron job for enabling metrics collection and visualization through Prometheus

Tools required: Linux operating system

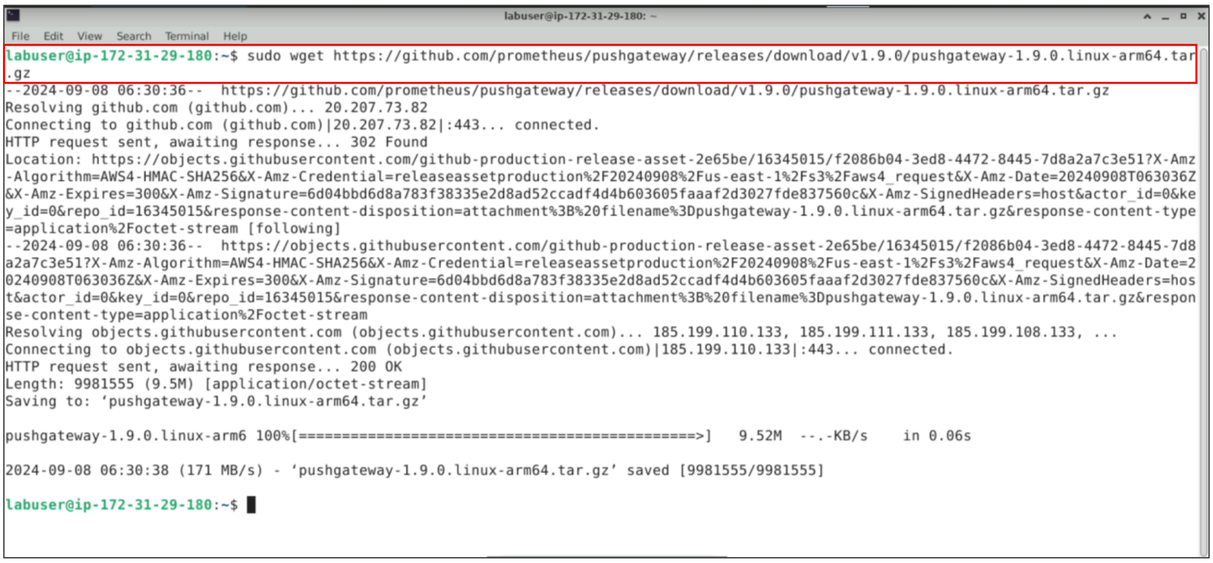
Prerequisites: Refer to Demo 02 of Lesson 01 for setting up a Prometheus server

Steps to be followed:

1. Download and initialize Pushgateway
2. Configure Prometheus for Pushgateway integration
3. Set up a bash script to push metrics to Pushgateway
4. Verify Pushgateway and Prometheus functionality

Step 1: Download and initialize Pushgateway

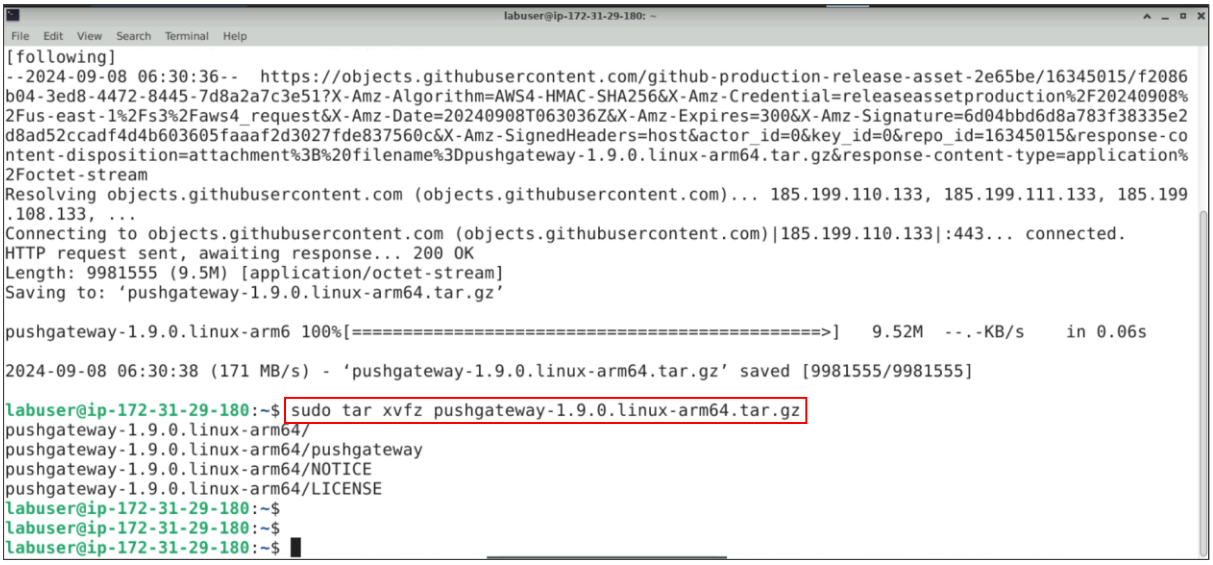
1.1 Navigate to the terminal in the system and run the following command to download the Pushgateway binary:
sudo wget https://github.com/prometheus/pushgateway/releases/download/v1.9.0/pushgateway-1.9.0.linux-arm64.tar.gz



labuser@ip-172-31-29-180: ~
File Edit View Search Terminal Help

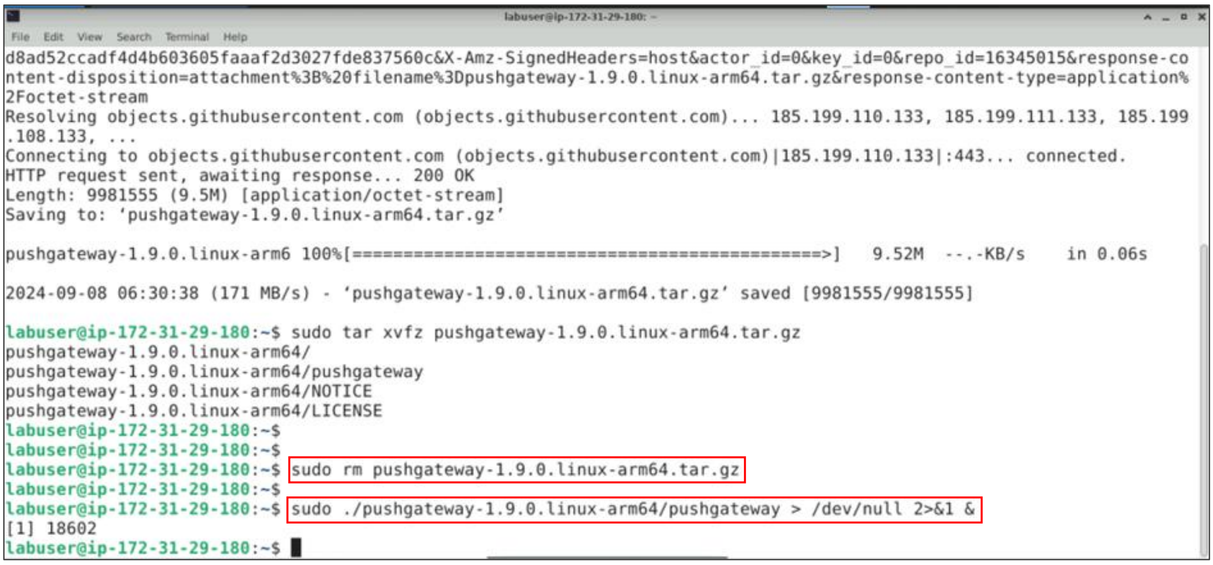
```
labuser@ip-172-31-29-180:~$ sudo wget https://github.com/prometheus/pushgateway/releases/download/v1.9.0/pushgateway-1.9.0.linux-arm64.tar
.gz
--2024-09-08 06:30:36--  https://github.com/prometheus/pushgateway/releases/download/v1.9.0/pushgateway-1.9.0.linux-arm64.tar.gz
Resolving github.com (github.com)... 20.207.73.82
Connecting to github.com (github.com)|20.207.73.82|:443... connected.
HTTP request sent, awaiting response... 302 Found
Location: https://objects.githubusercontent.com/github-production-release-asset-2e65be/16345015/f2086b04-3ed8-4472-8445-7d8a2a7c3e51?X-Amz
-Algorithm=AWS4-HMAC-SHA256&X-Amz-Credential=releaseassetproduction%2F20240908%2Fus-east-1%2Fs3%2Faws4_request&X-Amz-Date=20240908T063036Z
&X-Amz-Expires=300&X-Amz-Signature=6d04bbd6d8a783f38335e2d8ad52ccadf4d4b603605faaaf2d3027fde837560c&X-Amz-SignedHeaders=host&actor_id=0&ke
y_id=0&repo_id=16345015&response-content-disposition=attachment%3B%20filename%3Dpushgateway-1.9.0.linux-arm64.tar.gz&response-content-type
=application%2Foctet-stream [following]
--2024-09-08 06:30:36--  https://objects.githubusercontent.com/github-production-release-asset-2e65be/16345015/f2086b04-3ed8-4472-8445-7d8
a2a7c3e51?X-Amz-Algorithm=AWS4-HMAC-SHA256&X-Amz-Credential=releaseassetproduction%2F20240908%2Fus-east-1%2Fs3%2Faws4_request&X-Amz-Date=2
0240908T063036Z&X-Amz-Expires=300&X-Amz-Signature=6d04bbd6d8a783f38335e2d8ad52ccadf4d4b603605faaaf2d3027fde837560c&X-Amz-SignedHeaders=hos
t&actor_id=0&key_id=0&repo_id=16345015&response-content-disposition=attachment%3B%20filename%3Dpushgateway-1.9.0.linux-arm64.tar.gz&respon
se-content-type=application%2Foctet-stream
Resolving objects.githubusercontent.com (objects.githubusercontent.com)... 185.199.110.133, 185.199.111.133, 185.199.108.133, ...
Connecting to objects.githubusercontent.com (objects.githubusercontent.com)|185.199.110.133|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 9981555 (9.5M) [application/octet-stream]
Saving to: ‘pushgateway-1.9.0.linux-arm64.tar.gz’

pushgateway-1.9.0.linux-arm6 100%[=============================================>]   9.52M  --.-KB/s    in 0.06s

2024-09-08 06:30:38 (171 MB/s) - ‘pushgateway-1.9.0.linux-arm64.tar.gz’ saved [9981555/9981555]

labuser@ip-172-31-29-180:~$ 
```

1.2 Execute the following command to extract the downloaded Pushgateway file:
sudo tar xvfz pushgateway-1.9.0.linux-arm64.tar.gz


labuser@ip-172-31-29-180: ~
File Edit View Search Terminal Help

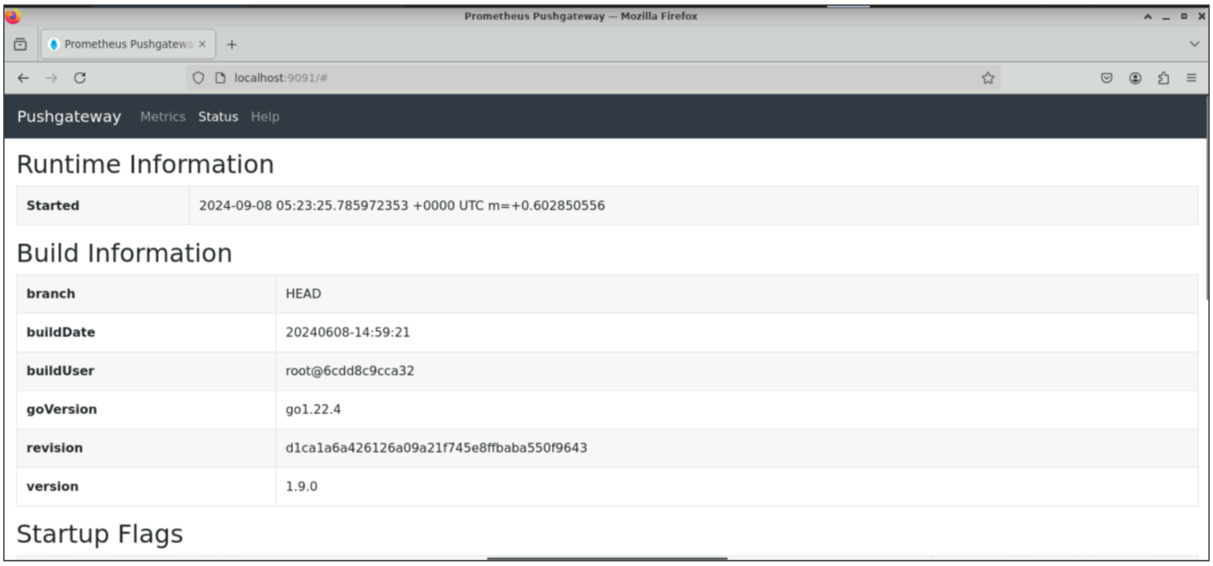
```
[following]
--2024-09-08 06:30:36--  https://objects.githubusercontent.com/github-production-release-asset-2e65be/16345015/f2086
b04-3ed8-4472-8445-7d8a2a7c3e51?X-Amz-Algorithm=AWS4-HMAC-SHA256&X-Amz-Credential=releaseassetproduction%2F20240908%
2Fus-east-1%2Fs3%2Faws4_request&X-Amz-Date=20240908T063036Z&X-Amz-Expires=300&X-Amz-Signature=6d04bbd6d8a783f38335e2
d8ad52ccadf4d4b603605faaaf2d3027fde837560c&X-Amz-SignedHeaders=host&actor_id=0&key_id=0&repo_id=16345015&response-co
ntent-disposition=attachment%3B%20filename%3Dpushgateway-1.9.0.linux-arm64.tar.gz&response-content-type=application%
2Foctet-stream
Resolving objects.githubusercontent.com (objects.githubusercontent.com)... 185.199.110.133, 185.199.111.133, 185.199
.108.133, ...
Connecting to objects.githubusercontent.com (objects.githubusercontent.com)|185.199.110.133|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 9981555 (9.5M) [application/octet-stream]
Saving to: ‘pushgateway-1.9.0.linux-arm64.tar.gz’

pushgateway-1.9.0.linux-arm6 100%[=============================================>]   9.52M  --.-KB/s    in 0.06s

2024-09-08 06:30:38 (171 MB/s) - ‘pushgateway-1.9.0.linux-arm64.tar.gz’ saved [9981555/9981555]

labuser@ip-172-31-29-180:~$ sudo tar xvfz pushgateway-1.9.0.linux-arm64.tar.gz
pushgateway-1.9.0.linux-arm64/
pushgateway-1.9.0.linux-arm64/pushgateway
pushgateway-1.9.0.linux-arm64/NOTICE
pushgateway-1.9.0.linux-arm64/LICENSE
labuser@ip-172-31-29-180:~$
labuser@ip-172-31-29-180:~$
labuser@ip-172-31-29-180:~$
```

1.3 Delete the downloaded file after extraction and start Pushgateway in the background using the following commands:
sudo rm pushgateway-1.9.0.linux-arm64.tar.gz
sudo ./pushgateway-1.9.0.linux-arm64/pushgateway > /dev/null 2>&1 &


labuser@ip-172-31-29-180: ~
File Edit View Search Terminal Help

```
d8ad52ccadf4d4b603605faaaf2d3027fde837560c&X-Amz-SignedHeaders=host&actor_id=0&key_id=0&repo_id=16345015&response-co
ntent-disposition=attachment%3B%20filename%3Dpushgateway-1.9.0.linux-arm64.tar.gz&response-content-type=application%
2Foctet-stream
Resolving objects.githubusercontent.com (objects.githubusercontent.com)... 185.199.110.133, 185.199.111.133, 185.199
.108.133, ...
Connecting to objects.githubusercontent.com (objects.githubusercontent.com)|185.199.110.133|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 9981555 (9.5M) [application/octet-stream]
Saving to: ‘pushgateway-1.9.0.linux-arm64.tar.gz’

pushgateway-1.9.0.linux-arm6 100%[=============================================>]   9.52M  --.-KB/s    in 0.06s

2024-09-08 06:30:38 (171 MB/s) - ‘pushgateway-1.9.0.linux-arm64.tar.gz’ saved [9981555/9981555]

labuser@ip-172-31-29-180:~$ sudo tar xvfz pushgateway-1.9.0.linux-arm64.tar.gz
pushgateway-1.9.0.linux-arm64/
pushgateway-1.9.0.linux-arm64/pushgateway
pushgateway-1.9.0.linux-arm64/NOTICE
pushgateway-1.9.0.linux-arm64/LICENSE
labuser@ip-172-31-29-180:~$
labuser@ip-172-31-29-180:~$
labuser@ip-172-31-29-180:~$ sudo rm pushgateway-1.9.0.linux-arm64.tar.gz
labuser@ip-172-31-29-180:~$
labuser@ip-172-31-29-180:~$ sudo ./pushgateway-1.9.0.linux-arm64/pushgateway > /dev/null 2>&1 &
[1] 18602
labuser@ip-172-31-29-180:~$
```

1.4 Navigate to the preferred browser and enter the URL **http://localhost:9091/** to confirm that Pushgateway is running successfully



Runtime Information

Started	2024-09-08 05:23:25.785972353 +0000 UTC m=+0.602850556

Build Information

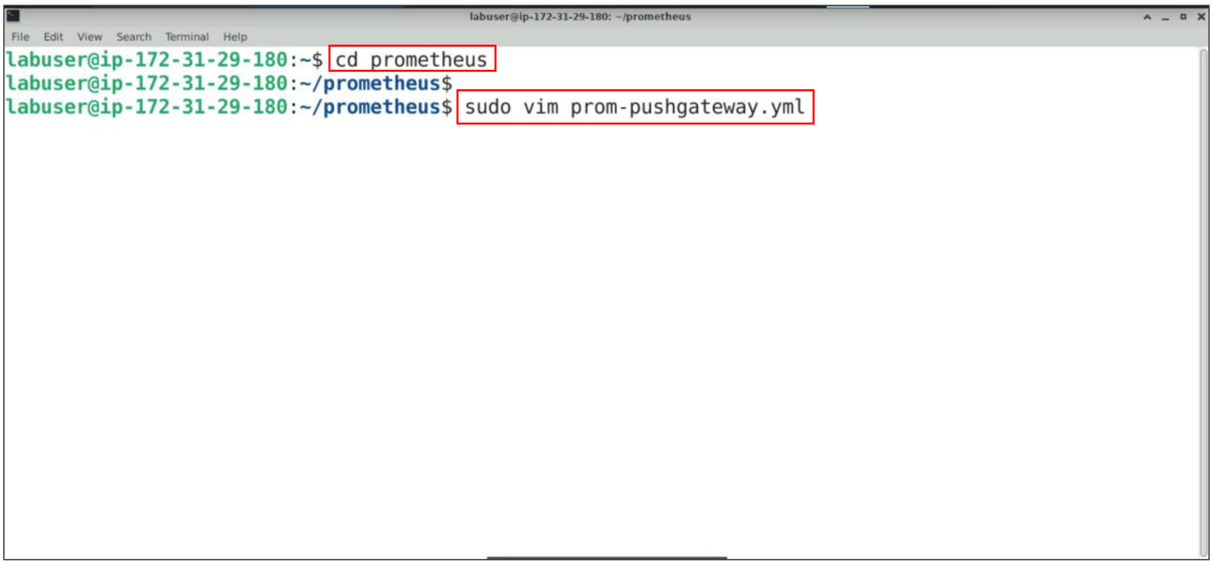
branch	HEAD
buildDate	20240608-14:59:21
buildUser	root@6cdd8c9cca32
goVersion	go1.22.4
revision	d1ca1a6a426126a09a21f745e8ffbaba550f9643
version	1.9.0

Startup Flags

The details in the **Runtime Information** section confirm that Pushgateway is running successfully.

Step 2: Configure Prometheus for Pushgateway integration

2.1 Navigate to the terminal, change the directory to the prometheus installation folder, and then open the **prom-pushgateway.yml** file for editing using the **Vim** editor:
cd prometheus
sudo vim prom-pushgateway.yml



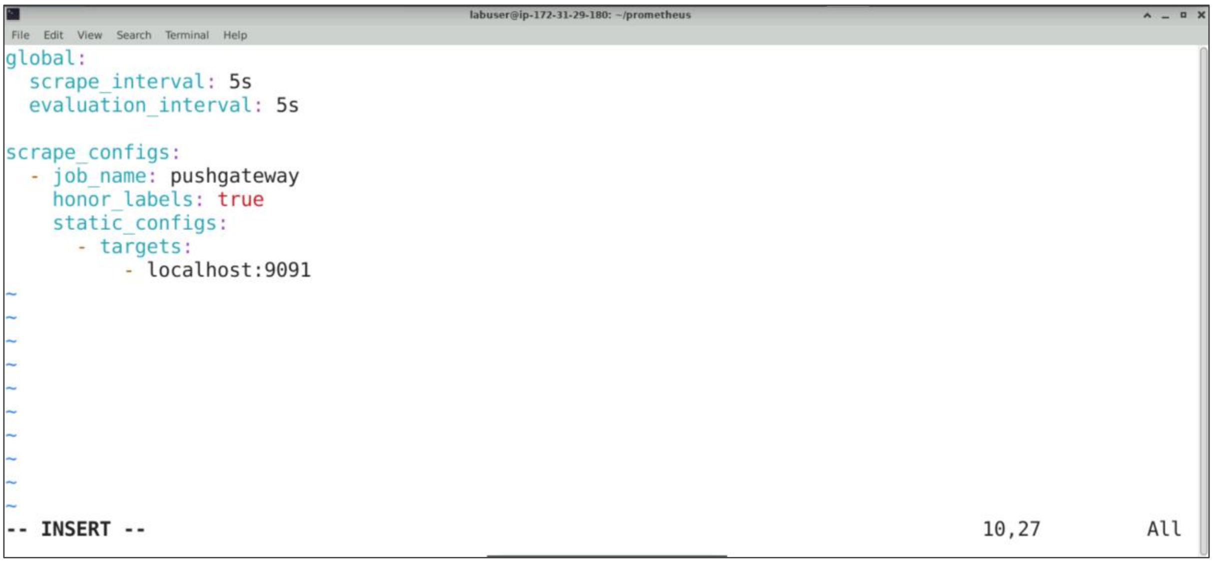
labuser@ip-172-31-29-180: ~/prometheus
File Edit View Search Terminal Help

```
labuser@ip-172-31-29-180:~$ cd prometheus
labuser@ip-172-31-29-180:~/prometheus$
labuser@ip-172-31-29-180:~/prometheus$ sudo vim prom-pushgateway.yml
```

Note: Ensure that Prometheus is already installed and then change the directory to the **Prometheus** installation directory

2.2 Switch to **INSERT** mode by typing **I** and then copy and paste the following job configuration:

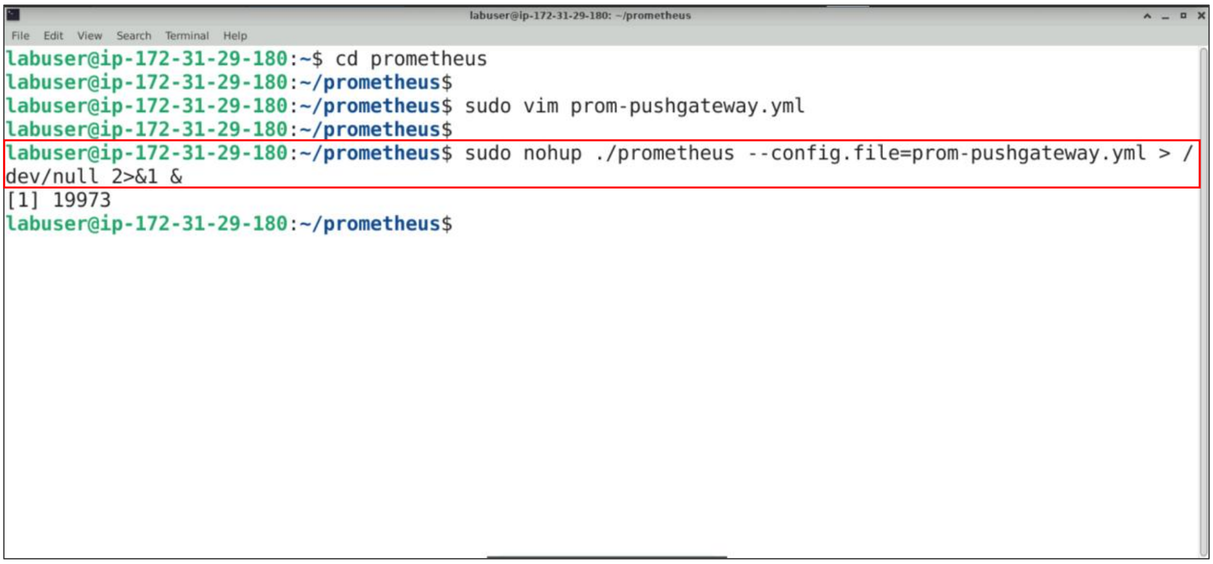
```
global:
  scrape_interval: 5s
  evaluation_interval: 5s

scrape_configs:
  - job_name: pushgateway
    honor_labels: true
    static_configs:
      - targets:
          - localhost:9091
```


labuser@ip-172-31-29-180: ~/prometheus
File Edit View Search Terminal Help

```
global:
  scrape_interval: 5s
  evaluation_interval: 5s

scrape_configs:
  - job_name: pushgateway
    honor_labels: true
    static_configs:
      - targets:
          - localhost:9091
~
~
~
~
~
~
~
~
~
~
-- INSERT --                                                  10,27         All
```

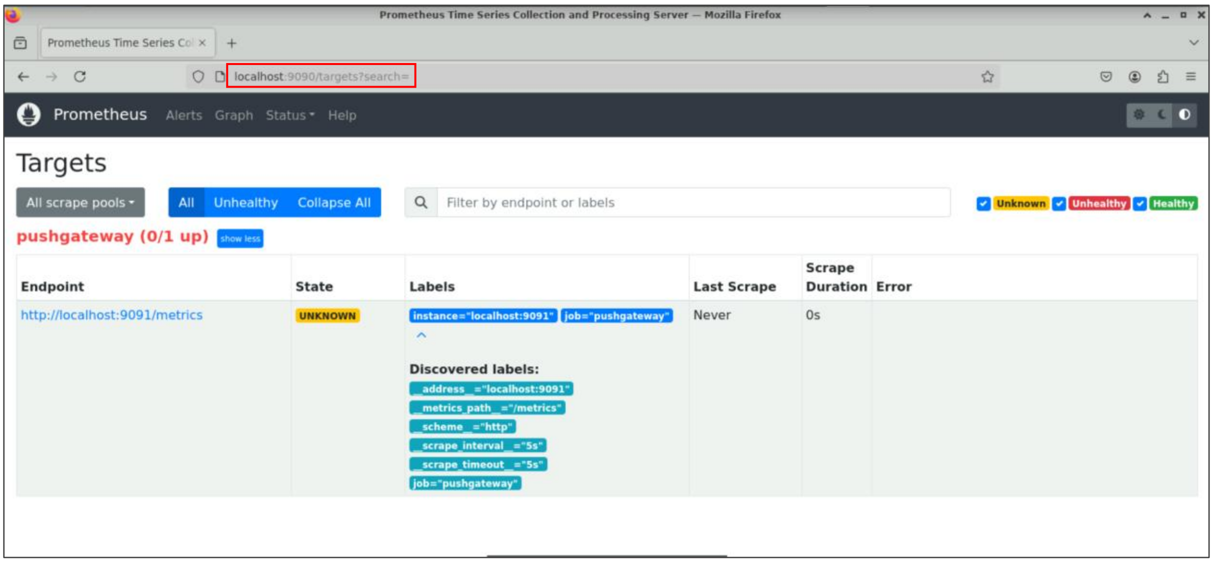
Note: Press **esc** and type **:wq** to save and exit the file

2.3 Run the following command to start Prometheus in the background with the specified configuration file:
sudo nohup ./prometheus --config.file=prom-pushgateway.yml > /dev/null 2>&1 &


labuser@ip-172-31-29-180: ~/prometheus
File Edit View Search Terminal Help

```
labuser@ip-172-31-29-180:~$ cd prometheus
labuser@ip-172-31-29-180:~/prometheus$
labuser@ip-172-31-29-180:~/prometheus$ sudo vim prom-pushgateway.yml
labuser@ip-172-31-29-180:~/prometheus$
labuser@ip-172-31-29-180:~/prometheus$ sudo nohup ./prometheus --config.file=prom-pushgateway.yml > /
dev/null 2>&1 &
[1] 19973
labuser@ip-172-31-29-180:~/prometheus$
```

2.4 Navigate to the browser and verify the job configuration from the Prometheus dashboard by accessing the URL **http://localhost:9090/targets** as shown below:


Prometheus Time Series Collection and Processing Server — Mozilla Firefox
Prometheus Time Series Co ×
localhost:9090/targets?search=
Prometheus
Alerts
Graph
Status
Help
All scrape pools
All
Unhealthy
Collapse All
Filter by endpoint or labels
Unknown
Unhealthy
Healthy
show less

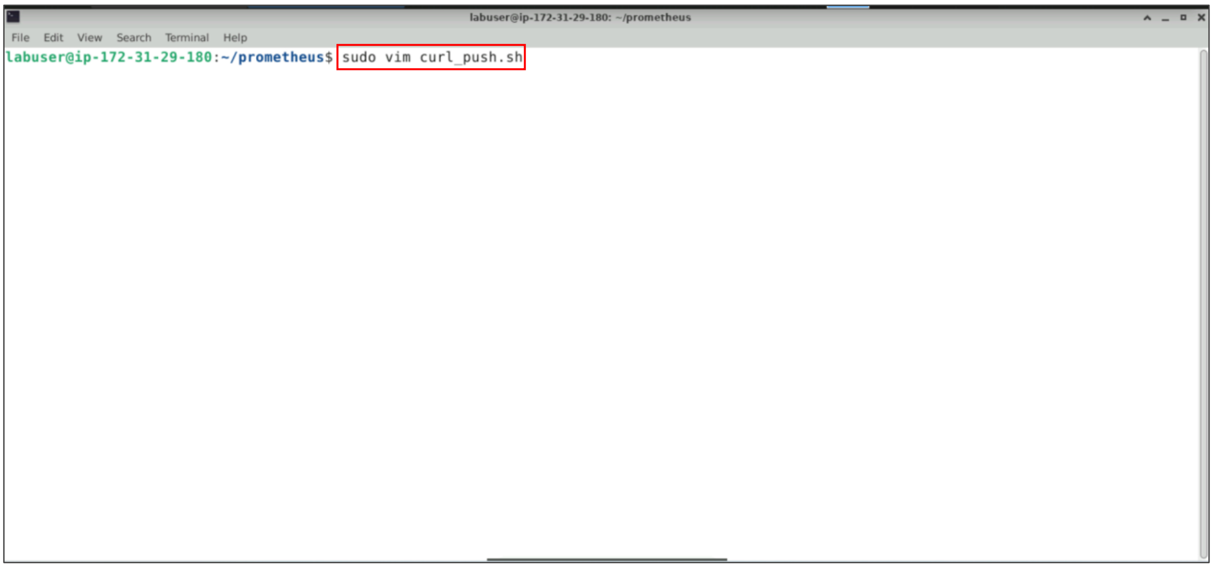
Targets

pushgateway (0/1 up)

Endpoint	State	Labels	Last Scrape	Scrape Duration	Error
http://localhost:9091/metrics	UNKNOWN	instance="localhost:9091" job="pushgateway" Discovered labels: __address__="localhost:9091" __metrics_path__="/metrics" __scheme__="http" __scrape_interval__="5s" __scrape_timeout__="5s" job="pushgateway"	Never	0s	

Step 3: Set up a bash script to push metrics to Pushgateway

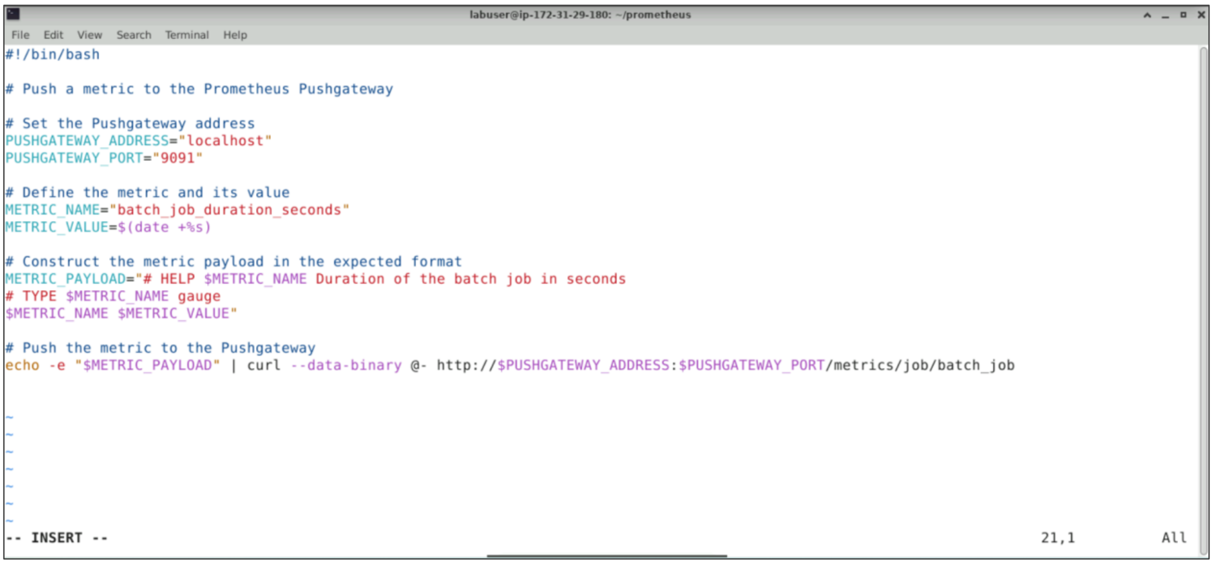
3.1 In the prometheus directory, create a bash file named **curl_push.sh** using the **Vim** editor to push metrics to Pushgateway:
sudo vim curl_push.sh


labuser@ip-172-31-29-180: ~/prometheus
File Edit View Search Terminal Help

```
labuser@ip-172-31-29-180:~/prometheus$ sudo vim curl_push.sh
```

3.2 Enter **I** and then add the following script:

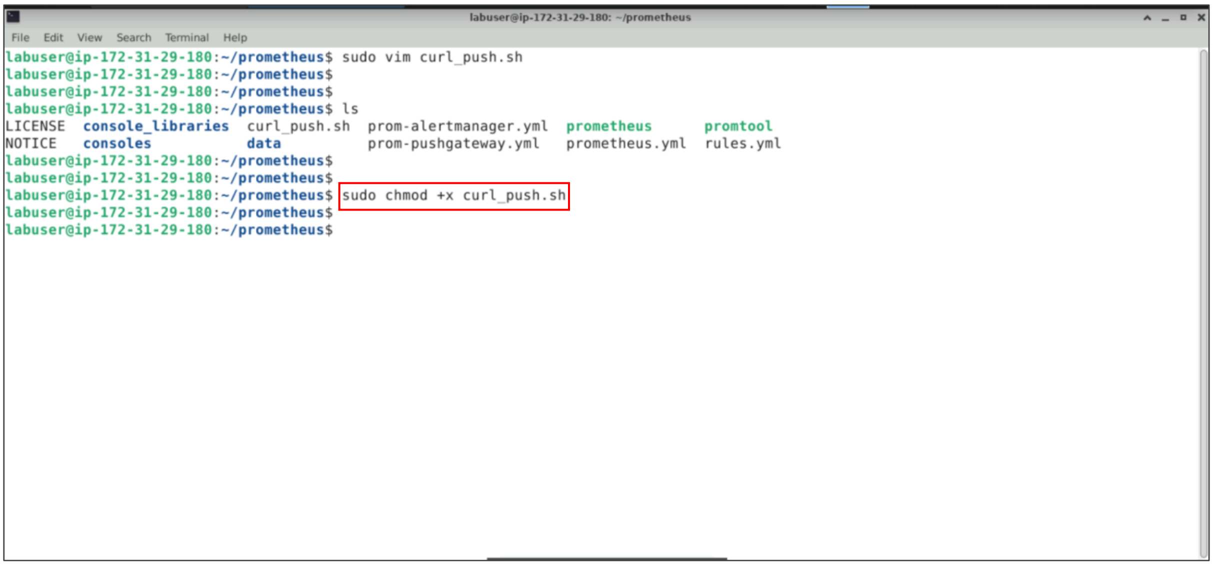
```
#!/bin/bash

# Push a metric to the Prometheus Pushgateway

# Set the Pushgateway address
PUSHGATEWAY_ADDRESS="localhost"
PUSHGATEWAY_PORT="9091"

# Define the metric and its value
METRIC_NAME="batch_job_duration_seconds"
METRIC_VALUE=$(date +%s)

# Construct the metric payload in the expected format
METRIC_PAYLOAD="#HELP $METRIC_NAME Duration of the batch job in seconds
#TYPE $METRIC_NAME gauge
$METRIC_NAME $METRIC_VALUE"

# Push the metric to the Pushgateway
echo -e "$METRIC_PAYLOAD" | curl --data-binary @-
http://$PUSHGATEWAY_ADDRESS:$PUSHGATEWAY_PORT/metrics/job/batch_job
```


labuser@ip-172-31-29-180: ~/prometheus
File Edit View Search Terminal Help

```
#!/bin/bash

# Push a metric to the Prometheus Pushgateway

# Set the Pushgateway address
PUSHGATEWAY_ADDRESS="localhost"
PUSHGATEWAY_PORT="9091"

# Define the metric and its value
METRIC_NAME="batch_job_duration_seconds"
METRIC_VALUE=$(date +%s)

# Construct the metric payload in the expected format
METRIC_PAYLOAD="# HELP $METRIC_NAME Duration of the batch job in seconds
# TYPE $METRIC_NAME gauge
$METRIC_NAME $METRIC_VALUE"

# Push the metric to the Pushgateway
echo -e "$METRIC_PAYLOAD" | curl --data-binary @- http://$PUSHGATEWAY_ADDRESS:$PUSHGATEWAY_PORT/metrics/job/batch_job


~
~
~
~
~
~
~
-- INSERT --                                                                          21,1          All
```

Note: Press **Esc**, then type **:wq** to save and exit the file. This script will track the current time of execution in milliseconds and push the values to Pushgateway. The metric's name is **batch_job_duration_seconds**.

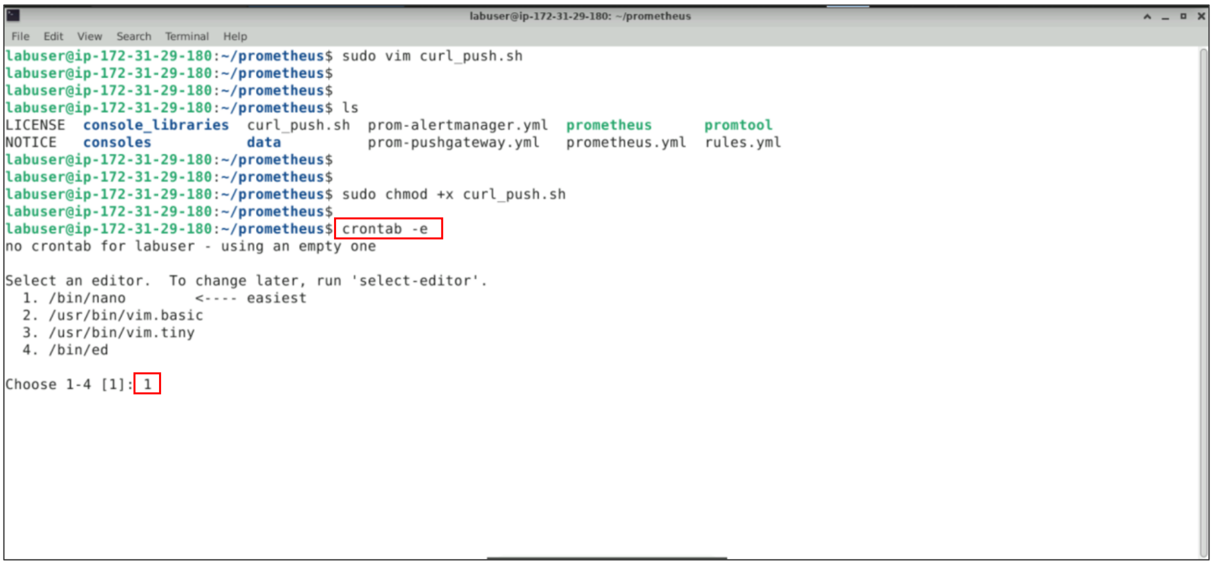
3.3 Run the following command to grant execute permissions to the **curl_push.sh** file:
sudo chmod +x curl_push.sh



labuser@ip-172-31-29-180: ~/prometheus
File Edit View Search Terminal Help

```
labuser@ip-172-31-29-180:~/prometheus$ sudo vim curl_push.sh
labuser@ip-172-31-29-180:~/prometheus$
labuser@ip-172-31-29-180:~/prometheus$
labuser@ip-172-31-29-180:~/prometheus$ ls
LICENSE  console_libraries  curl_push.sh  prom-alertmanager.yml  prometheus      promtool
NOTICE   consoles           data          prom-pushgateway.yml   prometheus.yml  rules.yml
labuser@ip-172-31-29-180:~/prometheus$
labuser@ip-172-31-29-180:~/prometheus$
labuser@ip-172-31-29-180:~/prometheus$ sudo chmod +x curl_push.sh
labuser@ip-172-31-29-180:~/prometheus$
labuser@ip-172-31-29-180:~/prometheus$
```

3.4 Execute the following command to open the crontab file for editing to schedule and manage the cron job, then select **1** to edit it using the nano editor (or any other editors by selecting 2, 3, or 4):
crontab -e

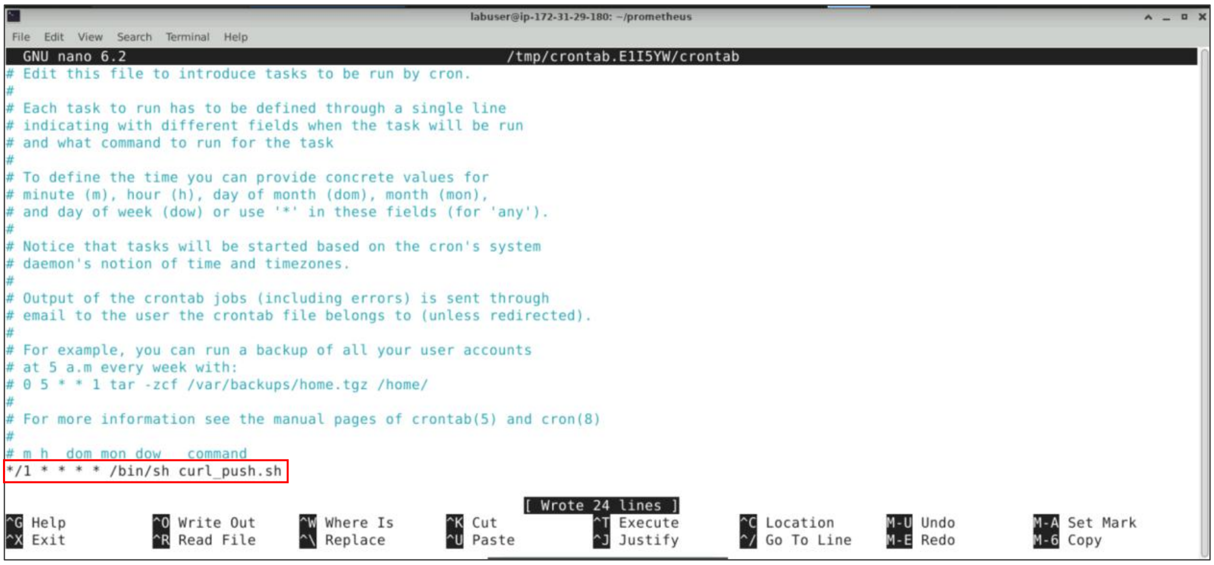


labuser@ip-172-31-29-180: ~/prometheus
File Edit View Search Terminal Help

```
labuser@ip-172-31-29-180:~/prometheus$ sudo vim curl_push.sh
labuser@ip-172-31-29-180:~/prometheus$
labuser@ip-172-31-29-180:~/prometheus$
labuser@ip-172-31-29-180:~/prometheus$ ls
LICENSE  console_libraries  curl_push.sh  prom-alertmanager.yml  prometheus      promtool
NOTICE   consoles           data          prom-pushgateway.yml   prometheus.yml  rules.yml
labuser@ip-172-31-29-180:~/prometheus$
labuser@ip-172-31-29-180:~/prometheus$
labuser@ip-172-31-29-180:~/prometheus$ sudo chmod +x curl_push.sh
labuser@ip-172-31-29-180:~/prometheus$
labuser@ip-172-31-29-180:~/prometheus$ crontab -e
no crontab for labuser - using an empty one

Select an editor.  To change later, run 'select-editor'.
  1. /bin/nano        <---- easiest
  2. /usr/bin/vim.basic
  3. /usr/bin/vim.tiny
  4. /bin/ed

Choose 1-4 [1]: 1
```

3.5 Add the following line at the end of the file:

***/1 * * * * /bin/sh curl_push.sh**

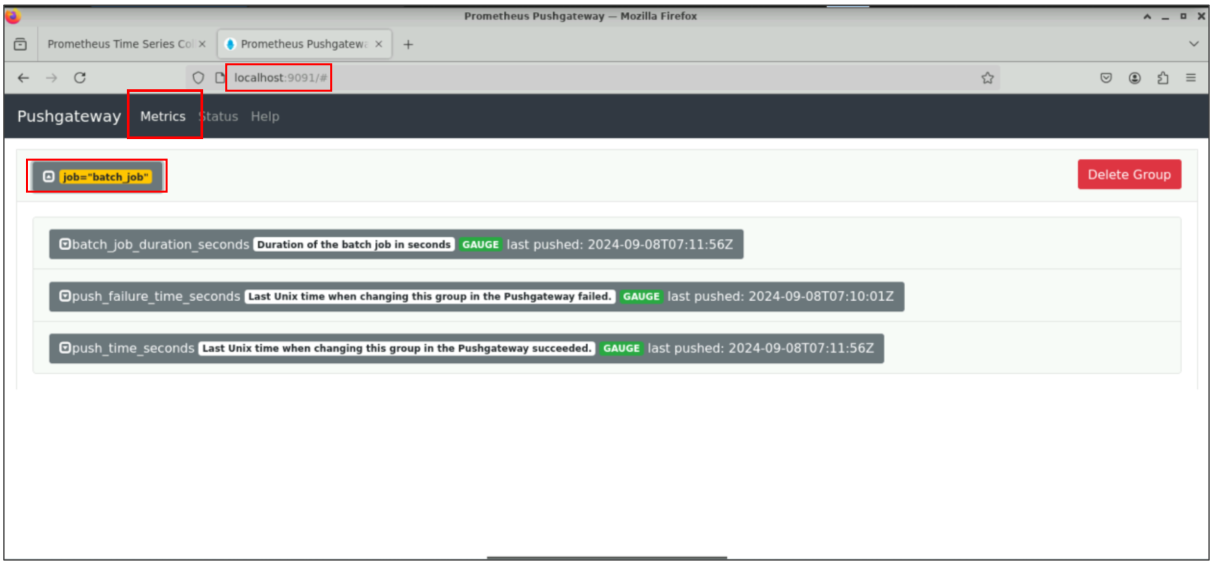

labuser@ip-172-31-29-180: ~/prometheus
File Edit View Search Terminal Help

```
  GNU nano 6.2                                   /tmp/crontab.E1I5YW/crontab
# Edit this file to introduce tasks to be run by cron.
#
# Each task to run has to be defined through a single line
# indicating with different fields when the task will be run
# and what command to run for the task
#
# To define the time you can provide concrete values for
# minute (m), hour (h), day of month (dom), month (mon),
# and day of week (dow) or use '*' in these fields (for 'any').
#
# Notice that tasks will be started based on the cron's system
# daemon's notion of time and timezones.
#
# Output of the crontab jobs (including errors) is sent through
# email to the user the crontab file belongs to (unless redirected).
#
# For example, you can run a backup of all your user accounts
# at 5 a.m every week with:
# 0 5 * * 1 tar -zcf /var/backups/home.tgz /home/
#
# For more information see the manual pages of crontab(5) and cron(8)
#
# m h  dom mon dow   command
*/1 * * * * /bin/sh curl_push.sh

                                                  [ Wrote 24 lines ]
^G Help         ^O Write Out    ^W Where Is     ^K Cut          ^T Execute      ^C Location     M-U Undo        M-A Set Mark
^X Exit         ^R Read File    ^\ Replace      ^U Paste        ^J Justify      ^/ Go To Line   M-E Redo        M-6 Copy
```

The script will automatically run every minute and push metrics to Pushgateway.

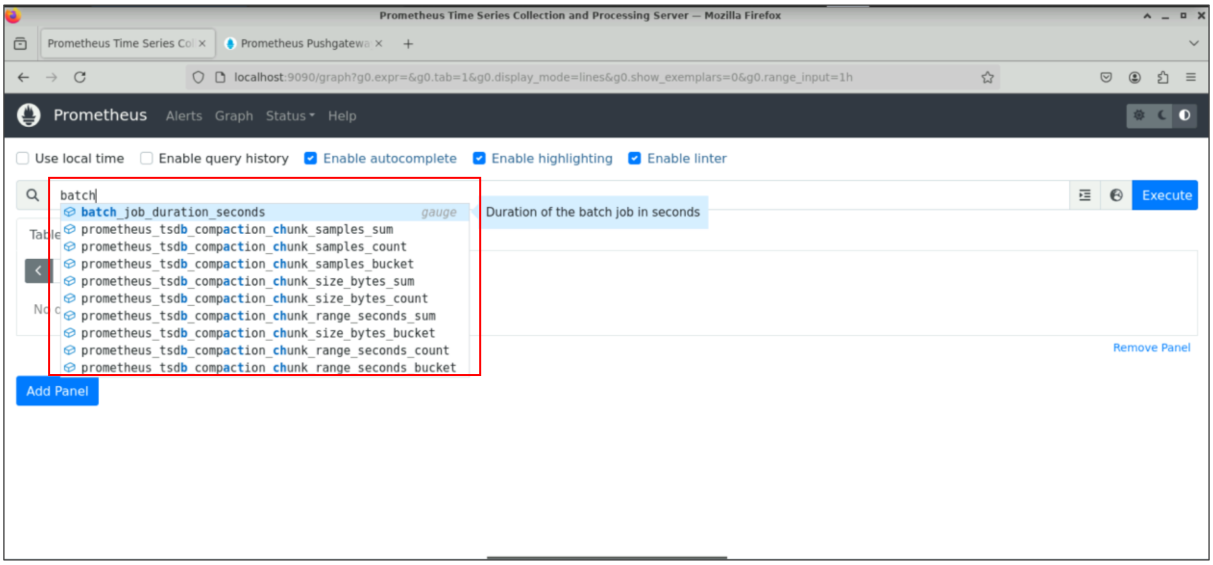
Step 4: Verify Pushgateway and Prometheus functionality

4.1 Navigate to **http://localhost:9091** in the browser to check the Pushgateway dashboard. In the **Metrics** tab, expand **batch_job** by clicking on it as shown below:



```
batch_job_duration_seconds Duration of the batch job in seconds GAUGE last pushed: 2024-09-08T07:11:56Z
push_failure_time_seconds Last Unix time when changing this group in the Pushgateway failed. GAUGE last pushed: 2024-09-08T07:10:01Z
push_time_seconds Last Unix time when changing this group in the Pushgateway succeeded. GAUGE last pushed: 2024-09-08T07:11:56Z
```

4.2 Access the Prometheus UI using **http://localhost:9090/** in the browser. Type **batch** and select the metrics name **batch_job_duration_seconds** in the expression browser

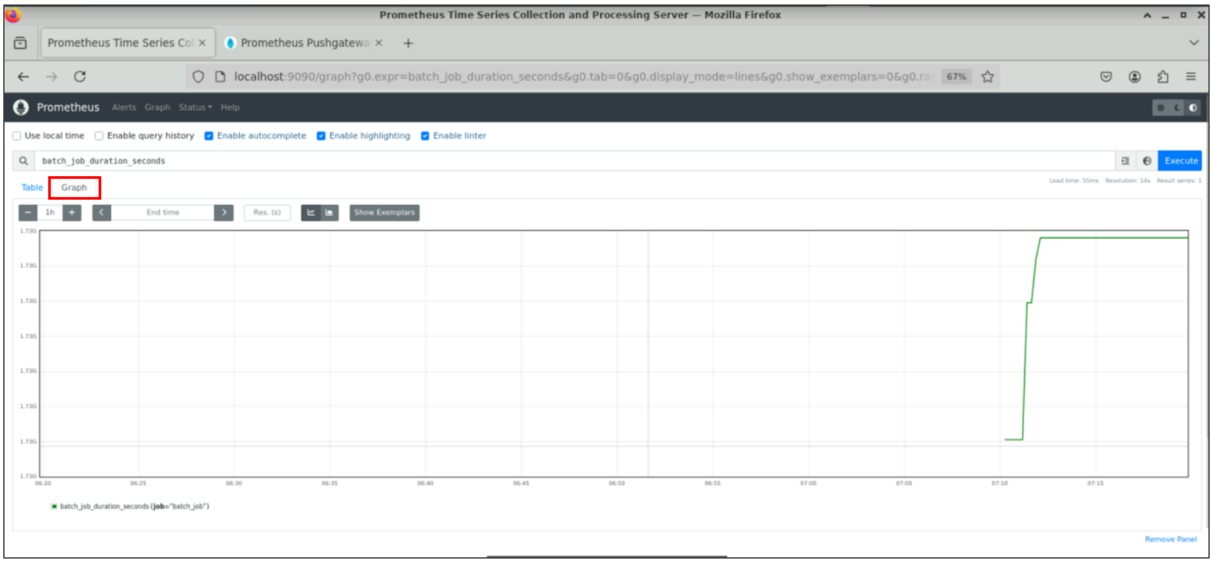


Prometheus Time Series Collection and Processing Server — Mozilla Firefox
Prometheus Time Series Col
Prometheus Pushgatewa
localhost:9090/graph?g0.expr=&g0.tab=1&g0.display_mode=lines&g0.show_exemplars=0&g0.range_input=1h
Prometheus
Alerts
Graph
Status
Help
Execute
Table
Remove Panel
Add Panel

□ Use local time □ Enable query history ☑ Enable autocomplete ☑ Enable highlighting ☑ Enable linter

```
batch
batch_job_duration_seconds                gauge   Duration of the batch job in seconds
prometheus_tsdb_compaction_chunk_samples_sum
prometheus_tsdb_compaction_chunk_samples_count
prometheus_tsdb_compaction_chunk_samples_bucket
prometheus_tsdb_compaction_chunk_size_bytes_sum
prometheus_tsdb_compaction_chunk_size_bytes_count
prometheus_tsdb_compaction_chunk_range_seconds_sum
prometheus_tsdb_compaction_chunk_size_bytes_bucket
prometheus_tsdb_compaction_chunk_range_seconds_count
prometheus_tsdb_compaction_chunk_range_seconds_bucket
```

4.3 Navigate to the **Graph** tab to verify if the Prometheus scrape configuration is functioning correctly



Prometheus Time Series Collection and Processing Server — Mozilla Firefox

localhost:9090/graph?g0.expr=batch_job_duration_seconds&g0.tab=0&g0.display_mode=lines&g0.show_exemplars=0&g0.ra

□ Use local time □ Enable query history ☑ Enable autocomplete ☑ Enable highlighting ☑ Enable linter

```
batch_job_duration_seconds
```

Load time: 55ms Resolution: 14s Result series: 1

The increasing line in the graph indicates successful metric scraping.

By following these steps, you have successfully automated metrics pushing from short-lived jobs to Pushgateway using a cron job to enable metrics collection and visualization through Prometheus.