

CISCO *Live!*



#CiscoLive



The bridge to possible

Monitoring Expressway Metrics Using Collectd and API

Luis Garcia

BRKCOL-2056



#CiscoLive

Cisco Webex App

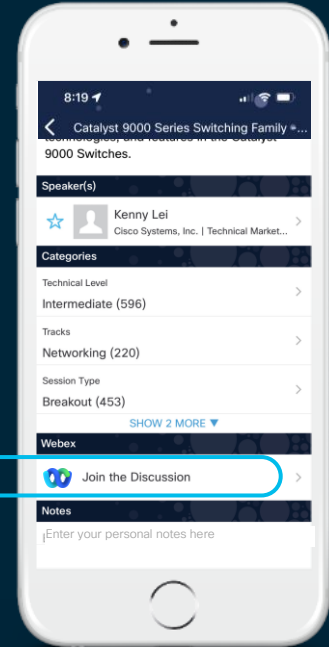
Questions?

Use Cisco Webex App to chat with the speaker after the session

How

- 1 Find this session in the Cisco Live Mobile App
- 2 Click “Join the Discussion”
- 3 Install the Webex App or go directly to the Webex space
- 4 Enter messages/questions in the Webex space

Webex spaces will be moderated by the speaker until June 17, 2022.



<https://ciscolive.ciscoevents.com/ciscolivebot/#BRKXXX-xxxx>



Agenda

- Introduction
- Expressway Collectd
- Grafana Dashboard
- REST API
- Conclusion

Introduction

Why we need metrics?

- In March 2020, there was no MRA registration counter in the web UI.
- You had to use RTMT to review the number of devices connected via Expressways.
- It was difficult to decide if more Expressway servers were required when the employees were moving to work from home.
- X12.6.1 introduced a counter of the MRA registrations in the Exp-E servers. This counter shows the current and max registrations at an Expressway-server.

Expressway Overview Page

- Limited view of the statistics, only current and peak values.
- Multiple clusters need to be reviewed separately, taking more time to do a simple status check.

Overview

Resource usage (last updated: 04:26:18 EDT)

		Total	exp-e01.ucdemolab.com	exp-e02.ucdemolab.com
Registered calls	Current video	0	0	0
	Current audio (SIP)	0	0	0
	Peak video	5	1	4
	Peak audio (SIP)	4	2	2

MRA Registrations	Current	2	2	0
	Peak	10	5	5

Alarm-Based Email Notifications

Email Notifications

Configuration

E-mail notification

On ▾



Source Configuration

Enable SMTP authentication

On ▾



Source E-mail



expwy-alarms@ucdemolab.com



Password



.....



Smtp Server



10.99.101.51



Smtp Port



587



- You can receive emails when a new alarm is raised in your Expressway servers.
- It allows you to customize the notifications, you could disable email notifications for specific alarms.

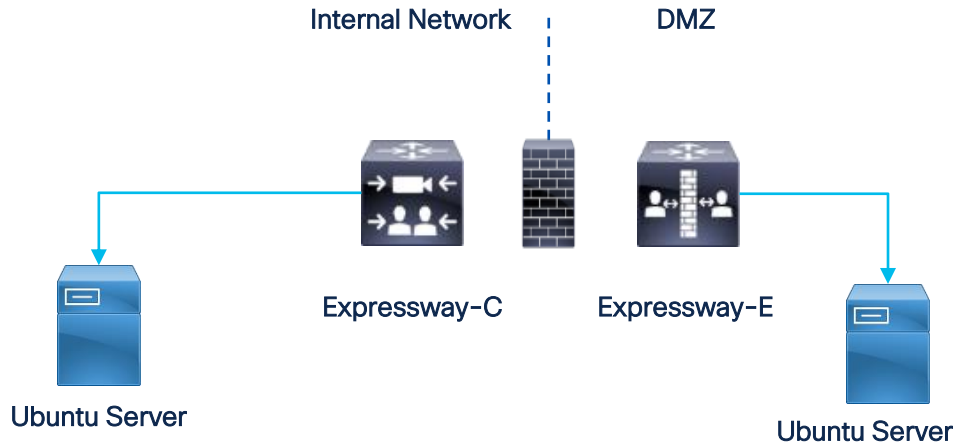
Advantages of Proactive Monitoring

- Detect increased usage before it affects production.
- Identify issues before users create a report.
- Keep track of the system performance for each server in a cluster.
- Simplify the review of metrics using graphics.

Expressway Metrics

Expressway Metrics

- Expressway collects statistics about the performance of the Hardware, OS and application.
- These statistics can be pushed to a remote server running the collectd daemon.



Depending on your security policies, you might need a server installed in the internal and DMZ network.

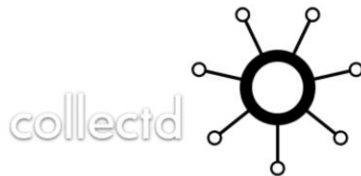
Collectd Plugins in Expressway

Collectd Plugins	
Aggregation	Protocols
CPU	Process
DF	Statsd
Disk	Swap
Exec-app	Uptime
Load	Users
Memory	Network

- Each plugin will provide information we can use to keep track of the Expressway health.
- For example, from the Exec-app plugin we can get the gauge-active_calls.
- We can review the current and max MRA registrations using the Statsd plugin.

Expressway Metrics








- In your remote server you need to install an application that handles collectd information.
- There are plenty of tools that can handle collectd data, but for this example we will use Graphite running in an Ubuntu Server 20.04.



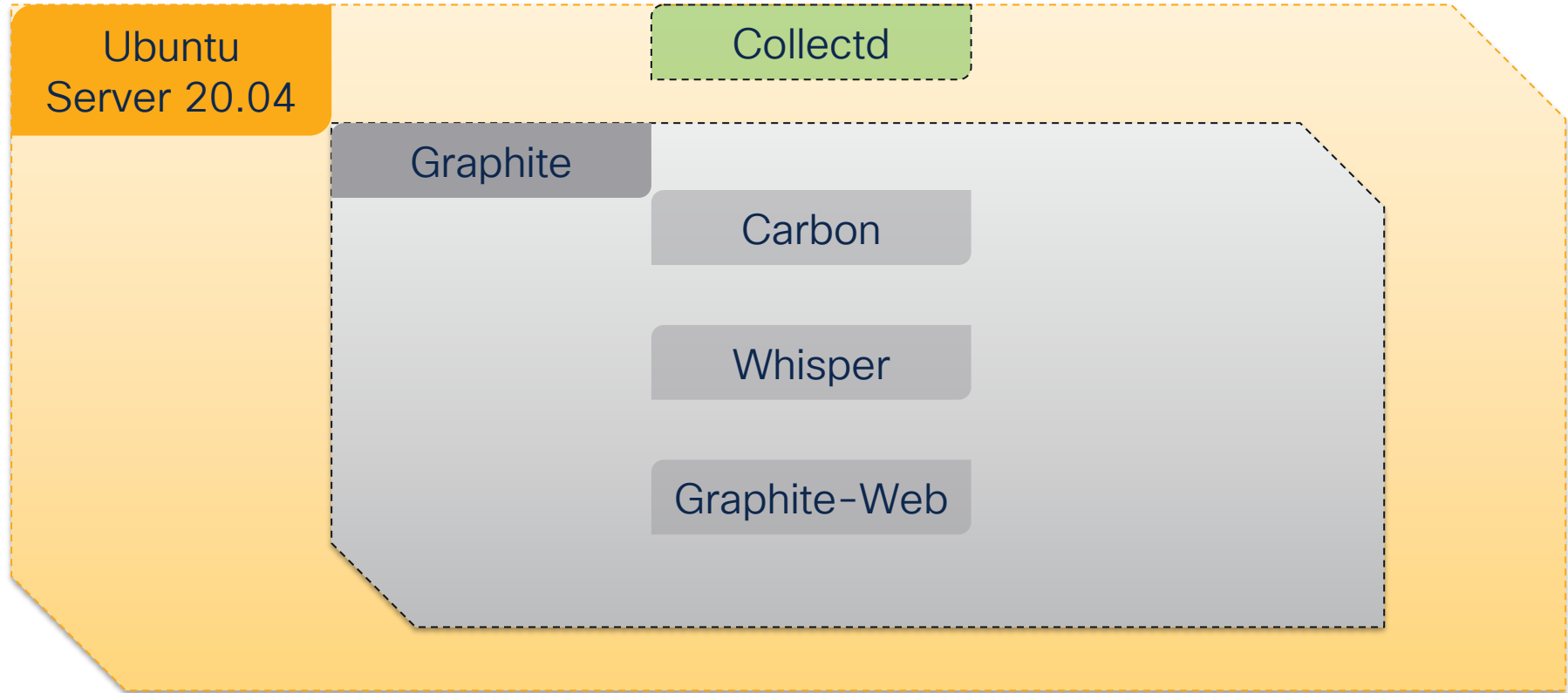
Expressway Metrics

- Configuration in the Expressway server can be done under the Maintenance>Logging page.
- We recommend to keep the interval at 60 seconds to reduce the impact on system performance.

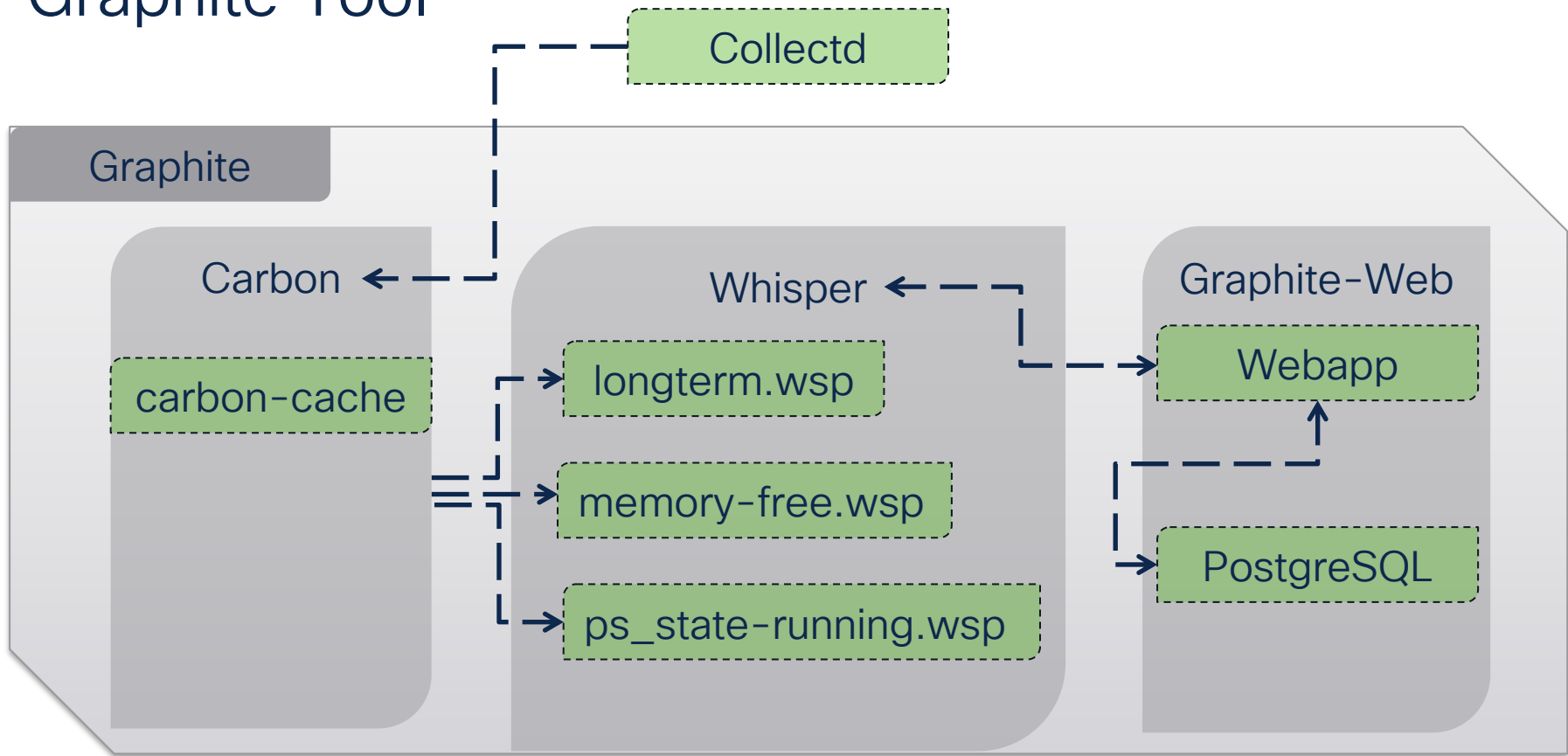
System Metrics

System Metrics Collection	On ▾	
Collection Interval (seconds)	 60	
Collection server address	 10.9.8.80	
Collection server port	 25826	

Graphite Tool



Graphite Tool



Collectd

- After installing the Collectd utility, modify the configuration file to listen for data from collectd clients (Expressway).

```
<Plugin "network">  
    Listen "198.51.100.15"  
</Plugin>
```

Tree

Search

Auto-Completer

gauge-ICEPassthroughMetrics_prrfxprfxcalls

gauge-ICEPassthroughMetrics_relayprrrfxcalls

gauge-ICEPassthroughMetrics_relayrelaycalls

gauge-ICEPassthroughMetrics_srvrfixprrrfxcalls

gauge-ICEPassthroughMetrics_srvrfixrelaycalls

gauge-ICEPassthroughMetrics_srvrfixsrvrfixcalls

gauge-MRARegistrations_number_of_registrations_current

gauge-MRARegistrations_number_of_registrations_max

gauge-siptnsp-0-X_none_stunkeepalive_failureresponses

gauge-siptnsp-0-X_none_stunkeepalive_requestsdropped

gauge-siptnsp-0-X_none_stunkeepalive_requestsforwarded

gauge-siptnsp-0-X_none_stunkeepalive_requestsreceived

gauge-siptnsp-0-X_none_stunkeepalive_responsesdropped

gauge-siptnsp-0-X_none_stunkeepalive_responsesforwarded

gauge-siptnsp-0-X_none_stunkeepalive_responsesreceived

gauge-siptnsp-0-X_none_stunkeepalive_successresponses

swap

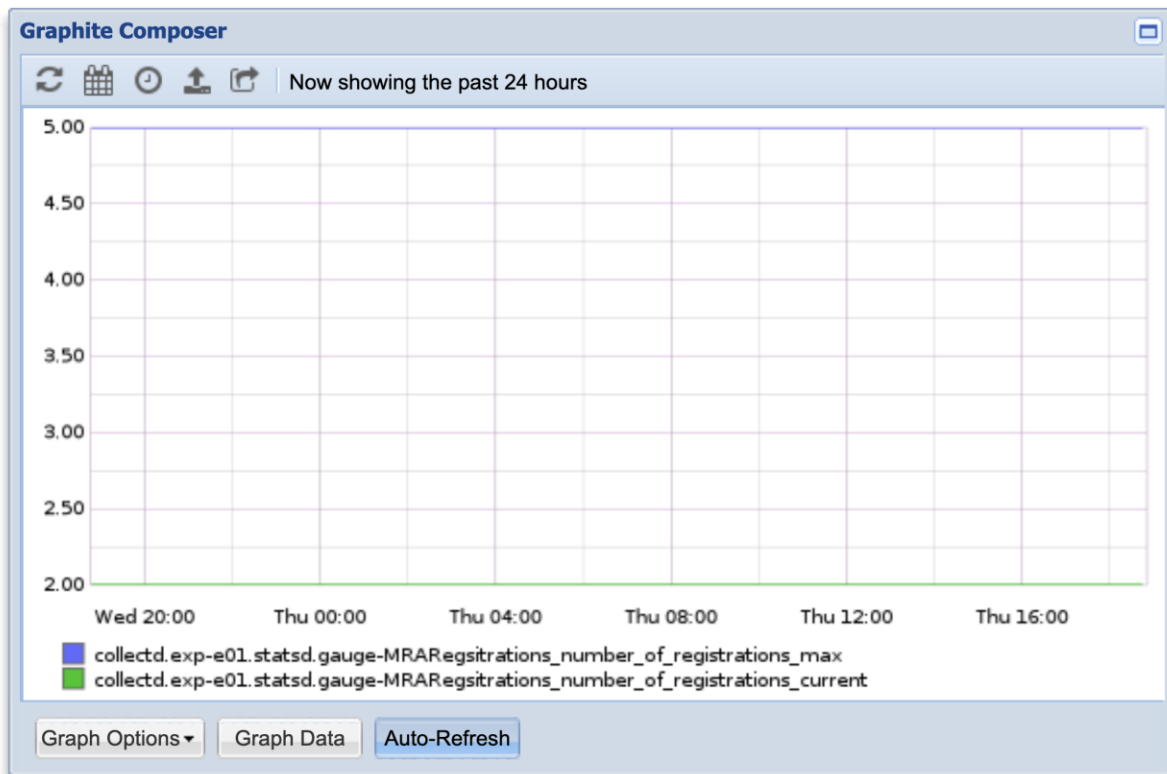
uptime

users

p-e02

p-e03

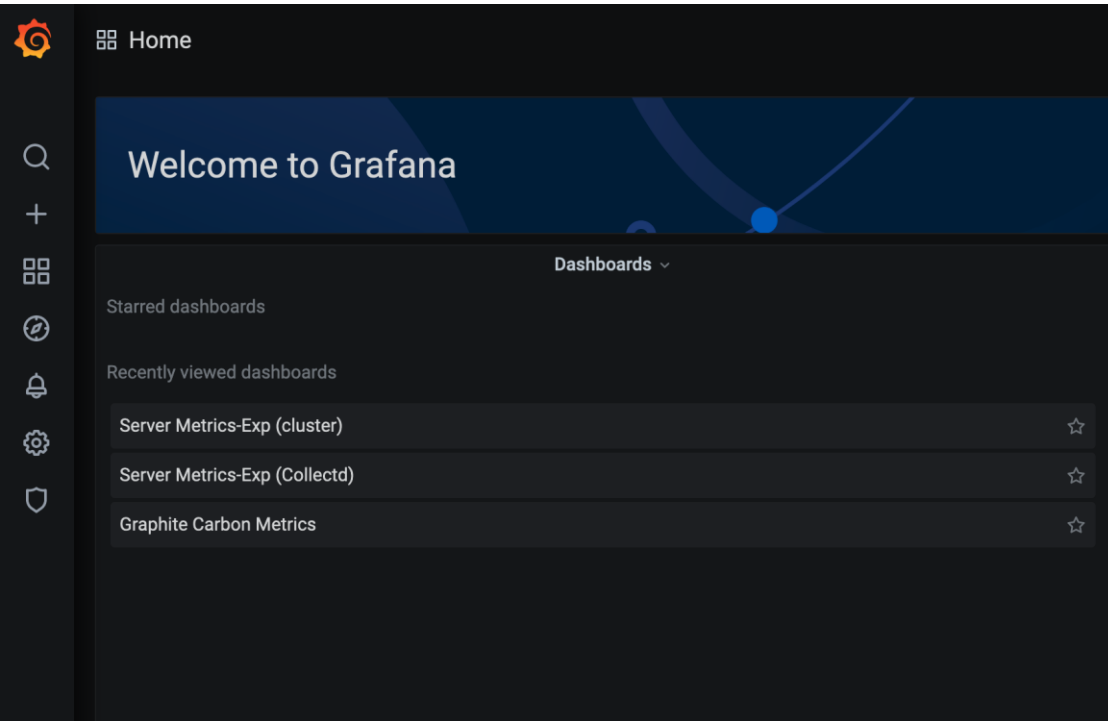
...



Grafana Dashboard



Grafana Dashboard



- Grafana is an analytics tool that can be used to create dashboards to monitor metrics.
- It allows you to monitor several Expressway clusters within the same dashboard.
- For this example, we have Grafana running in an Ubuntu server.

Grafana Dashboard



Add data source

Choose a data source type

graphite



Graphite

Open source time series database

Core

- From the Settings > Data Sources page, select the option Add New Data Source.
- Select Graphite as the type and specify the IP of the server.

Name



Graphite

Default



HTTP

URL



http://127.0.0.1:80

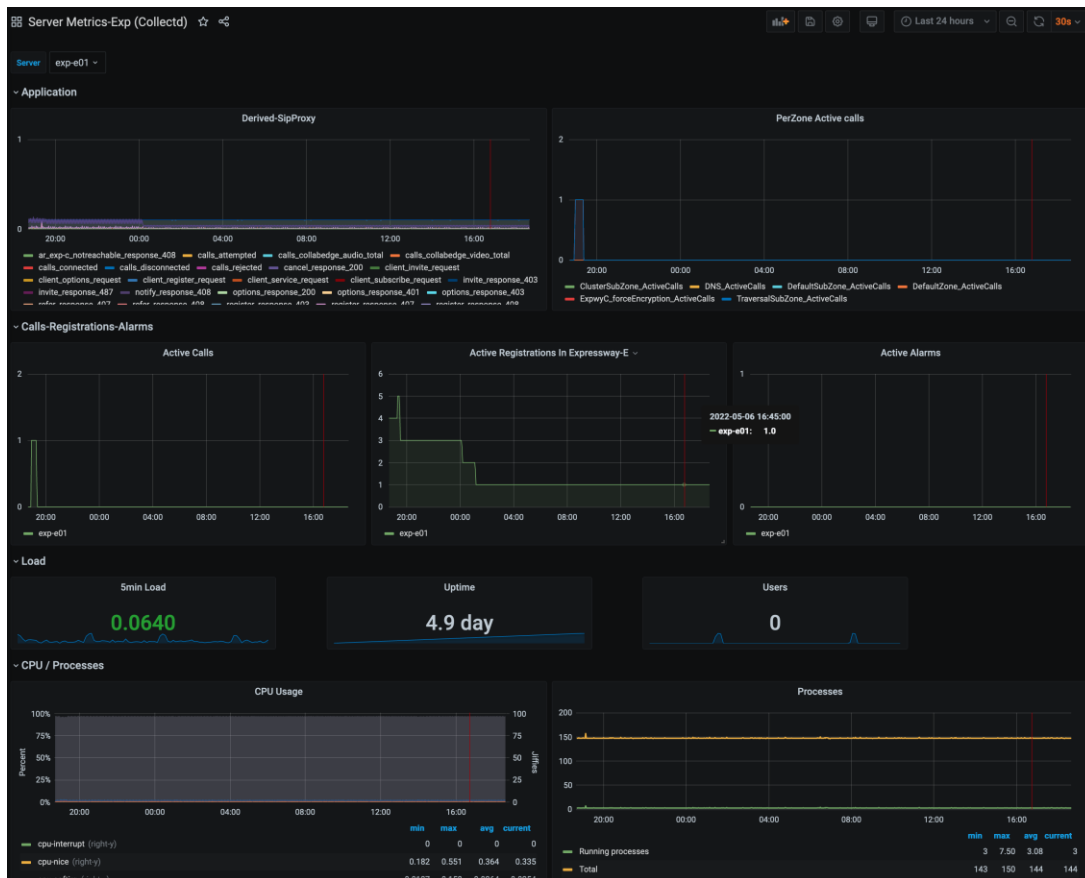
Whitelisted Cookies



Add Name

Add

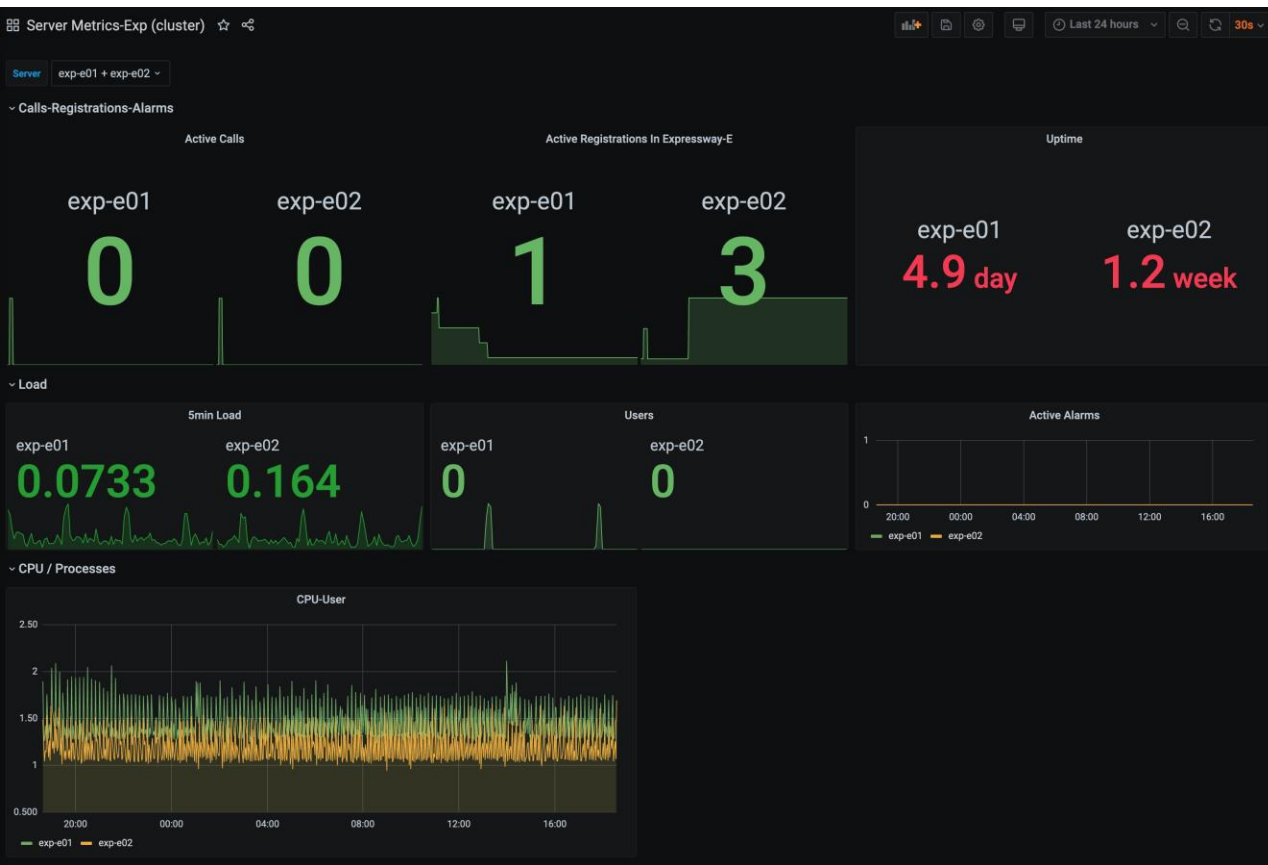
Grafana Dashboard – Server



- The Expressway Metrics guide includes a JSON file to create this dashboard.
- It provides a detailed view per server of most statistics.

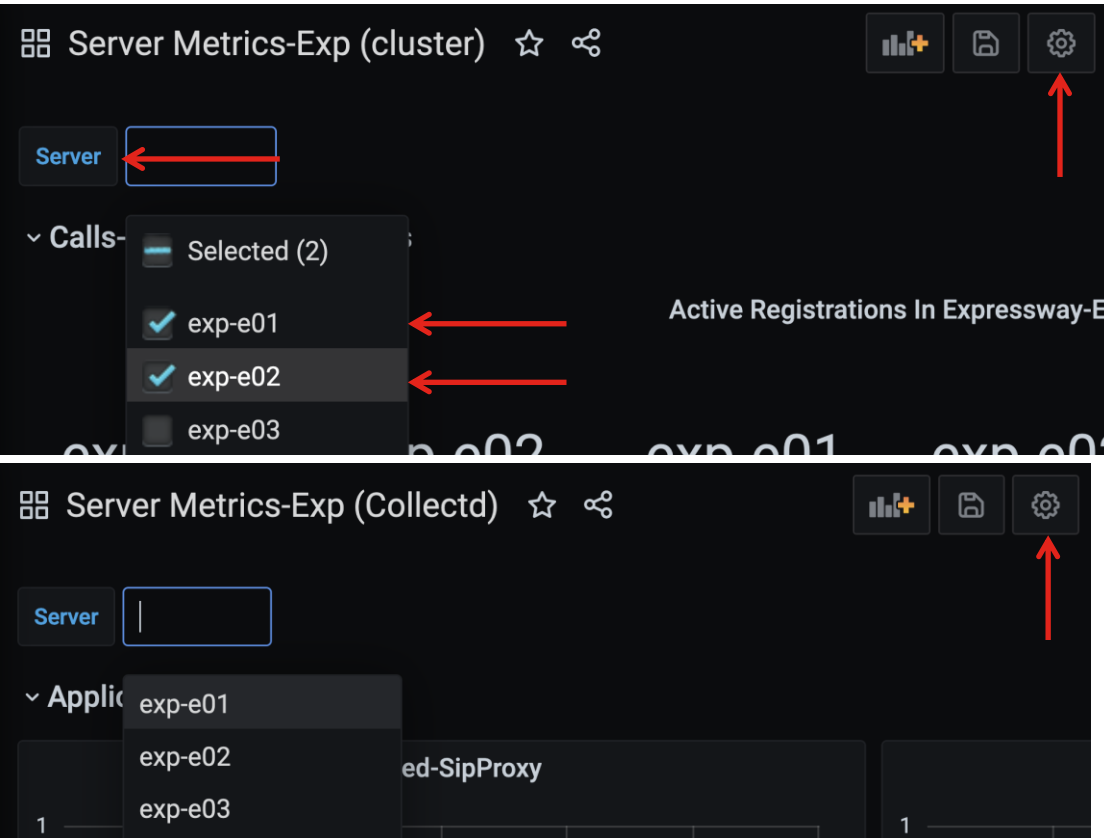
[Expressway Metrics Guide](#)

Grafana Dashboard – Cluster



- This dashboard shows data for all the nodes in the cluster in a single page.
- We are only showing some of the most important metrics to simplify the view.

Grafana Dashboard – Cluster vs Server



- In the Cluster dashboard you can select multiple Expressway servers.
- Servers selected don't need to belong to the same cluster.
- In the Server dashboard you can only see the detailed stats of one server at a time.
- This is controlled in the variable configuration in the dashboard settings.

Grafana Dashboard – Cluster vs Server

The screenshot shows the 'Variables > Edit' page in Grafana. The left sidebar contains navigation links: General, Annotations, Variables (highlighted with a red arrow), Links, Versions, Permissions, and JSON Model. Below these are buttons for 'Save dashboard' and 'Save As...'. The main content area is divided into 'General' and 'Query Options' sections. In the 'General' section, the 'Name' field is set to 'Server' (highlighted with a red arrow), 'Type' is 'Query', 'Label' is 'optional display name', 'Hide' is a dropdown, and 'Description' is 'descriptive text'. In the 'Query Options' section, the 'Data source' is 'collectd' (highlighted with a red arrow), 'Refresh' is 'On Dashboard Load', and the 'Query' is 'collectd.*' (highlighted with a red arrow). A 'Selection Options' panel is open on the right, showing 'Multi-value' and 'Include All option' both with toggle switches turned on (highlighted with a red arrow).

Variables > Edit

General

Name: Server

Type: Query

Label: optional display name

Hide:

Description: descriptive text

Query Options

Data source: collectd

Refresh: On Dashboard Load

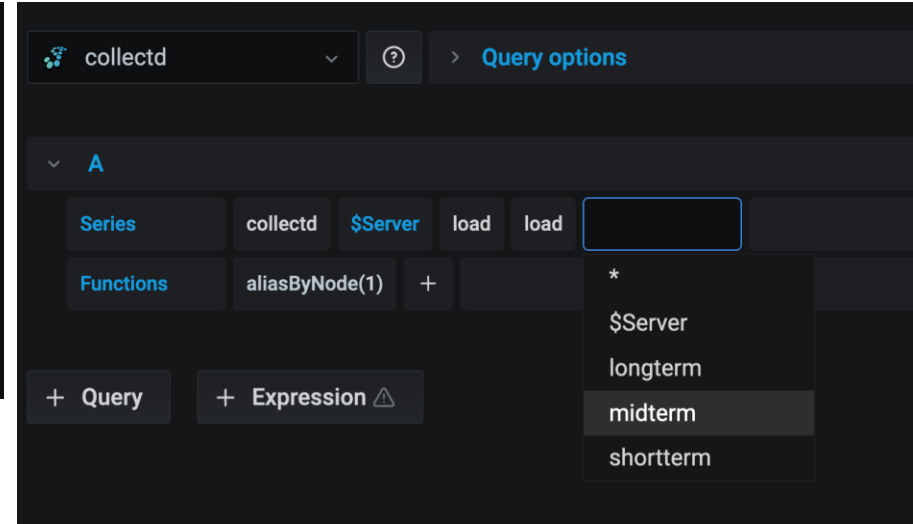
Query: collectd.*

Selection Options

Multi-value: ☒

Include All option: ☒

Grafana Dashboard – Load



- The load panel will show us an overview of the utilization of the system based on the task queue.
- We can keep track of the system load within 1 min, 5 min and 15 min.

Grafana Dashboard – Load

- Alerts can be configured for some of the metrics to get email notifications when the server performance is at risk.

The screenshot shows the Grafana Alerting configuration page. At the top, there are three tabs: 'Query' (1), 'Transform' (0), and 'Alert' (1). The 'Alert' tab is selected. Below the tabs, the 'Rule' section contains a 'Name' field with the value 'Panel Title alert', an 'Evaluate every' field with the value '1m', and a 'For' field with the value '5m'. The 'Conditions' section shows a single condition: 'WHEN avg () OF query (A, 5m, now) IS ABOVE 1.5'. A red arrow points to the '1.5' threshold value. There is a plus icon to add more conditions and a trash icon to delete the current condition.

Query 1 Transform 0 Alert 1

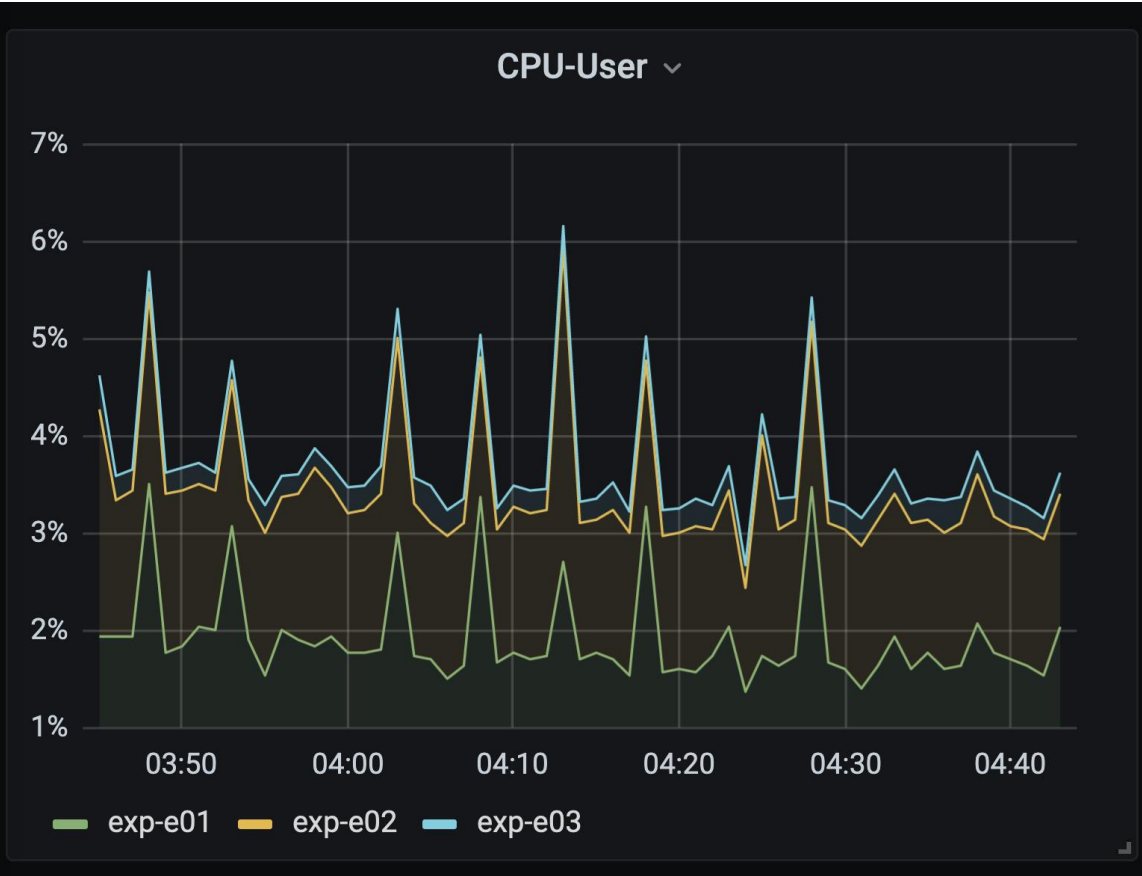
Rule

Name Panel Title alert Evaluate every 1m For 5m

Conditions

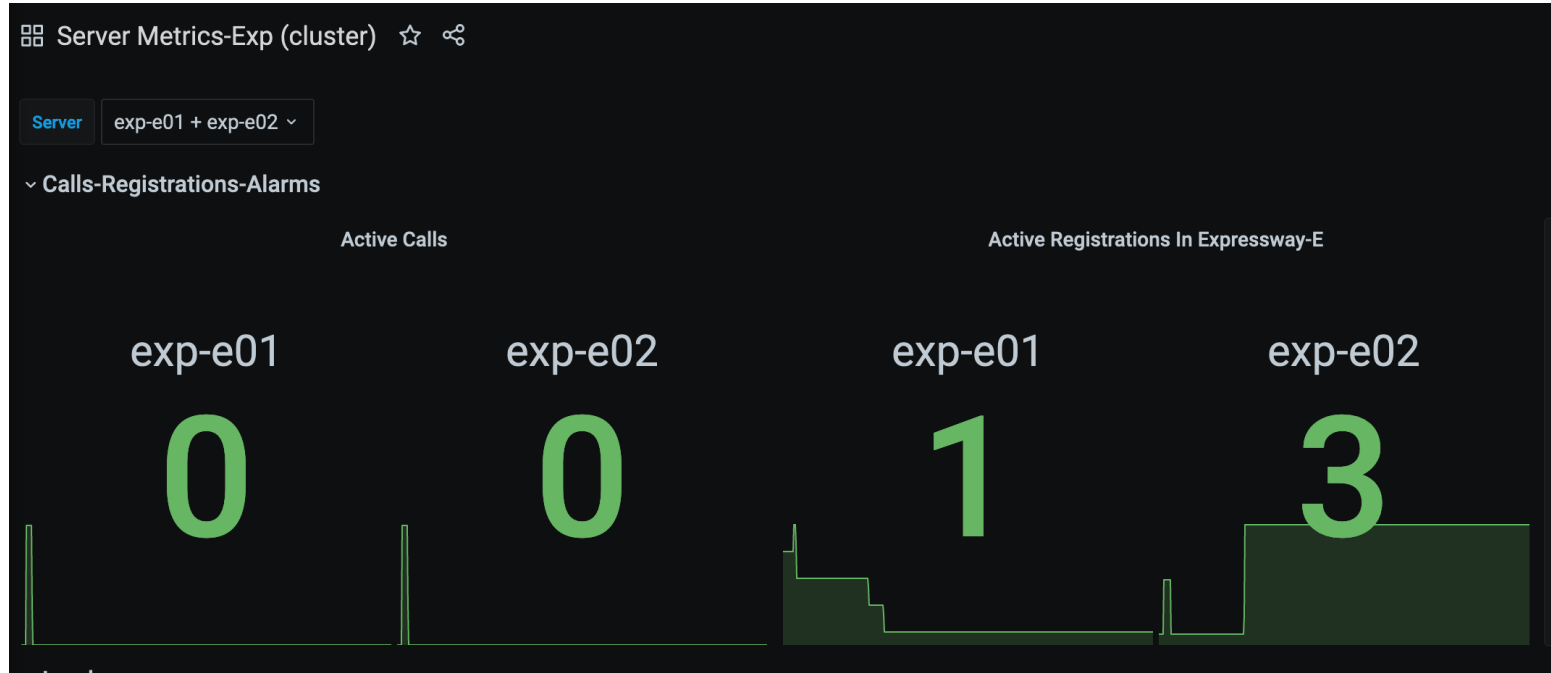
WHEN avg () OF query (A, 5m, now) IS ABOVE 1.5

Grafana Dashboard – CPU-User



- CPU user keeps track of how much CPU is being used by the app.
- CPU utilization should not go above 70%. (cpu-user + cpu-system)

Grafana Dashboard – Cluster



- Active Registrations and Active Calls panels allows administrators decide if more or less Expressways are needed in their infrastructure.

Expressway REST API

REST API

- Expressway's API is self-documented using RAML. This is available at <https://<Expressway FQDN or IP>/api/raml>.
- Requests and Responses use the JSON schema.
- API is accessible via HTTPS using an admin account with API access enabled.
- Base URL: <https://<Expressway FQDN or IP>/api>

IMPORTANT! Database API

- Expressways also have a Database API, but this is only for our support and development teams.
- Changes will be done in X14.2 to have this API disabled by default.

Users > Maintenance > **Experimental**

You are

Database REST API

TXAS links

TXAS docs

XML get/set

Set access

API

Developer log >

Hybrid Services Log >

Network Graph >

Database REST API

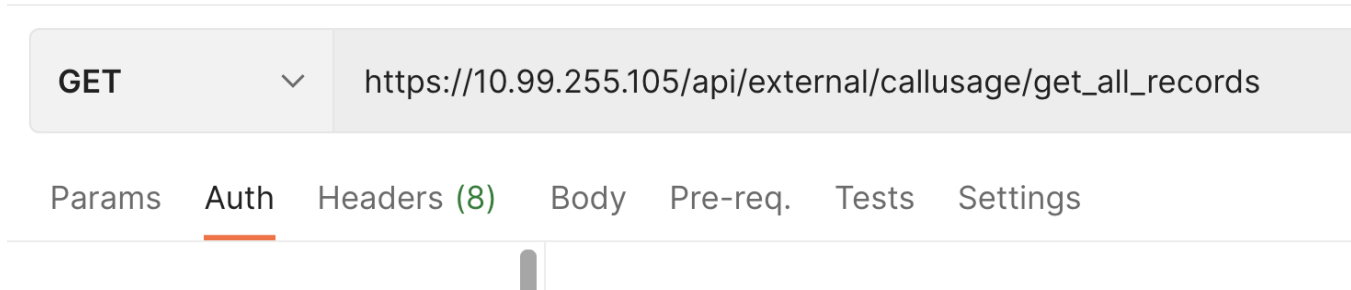
Status API


Table	Replication
status/acmeclustersync	global
status/acmeclustersyncstatus	global
status/acmestate	segmented
status/alarm	segmented
status/authentication/h350	segmented

CDR APIs

Base URL: `https://<Expressway FQDN or IP>/api/external/callusage:`

- `get_all_records`
- `get_records_for_interval`
- `get_records_for_filter`
- `get_all_csv_records`

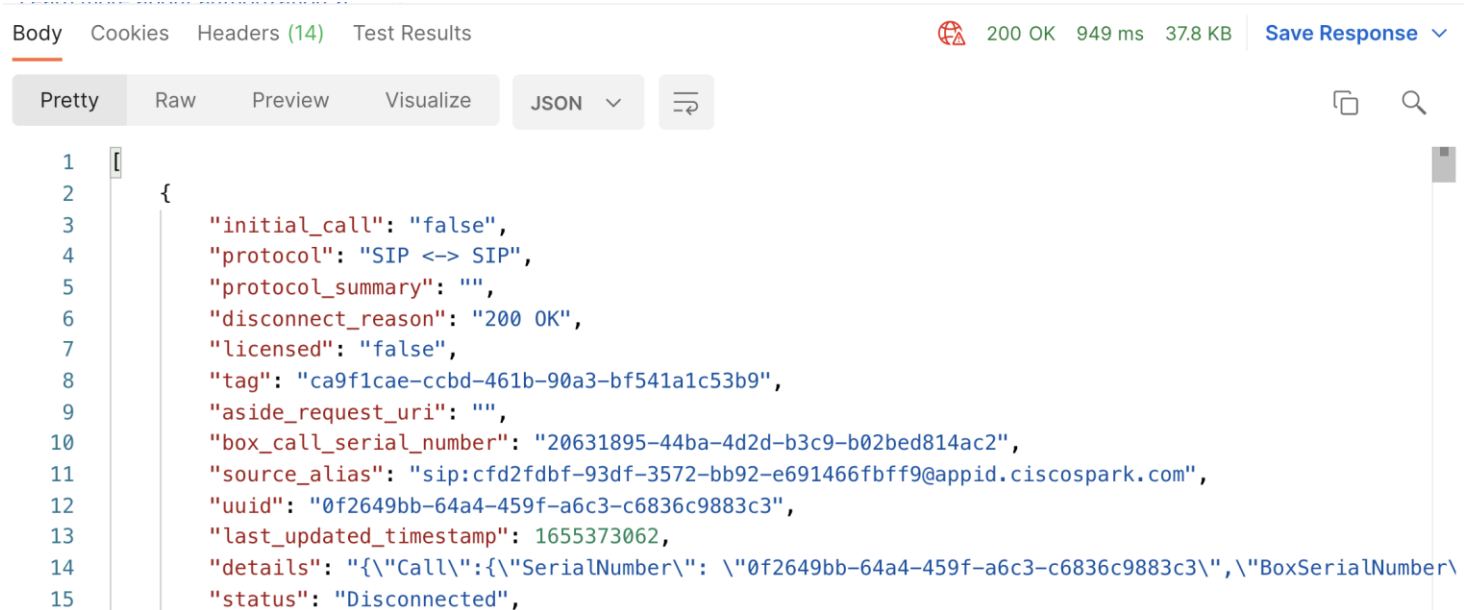


GET  `https://10.99.255.105/api/external/callusage/get_all_records`

Params Auth Headers (8) Body Pre-req. Tests Settings

CDR APIs

Remember to enable CDR before sending the API. This is done from Maintenance > Logging.



The screenshot shows a REST client interface with the following details:

- Body** tab is selected.
- Headers (14)** and **Test Results** tabs are visible.
- Status:** 200 OK, 949 ms, 37.8 KB.
- Save Response** button is available.
- View Options:** Pretty, Raw, Preview, Visualize, JSON (selected), and a copy icon.
- JSON Response:**

```
1 [
2   {
3     "initial_call": "false",
4     "protocol": "SIP <=> SIP",
5     "protocol_summary": "",
6     "disconnect_reason": "200 OK",
7     "licensed": "false",
8     "tag": "ca9f1cae-ccbd-461b-90a3-bf541a1c53b9",
9     "aside_request_uri": "",
10    "box_call_serial_number": "20631895-44ba-4d2d-b3c9-b02bed814ac2",
11    "source_alias": "sip:cfd2fdbf-93df-3572-bb92-e691466fbff9@appid.ciscopark.com",
12    "uuid": "0f2649bb-64a4-459f-a6c3-c6836c9883c3",
13    "last_updated_timestamp": 1655373062,
14    "details": "{\\\"Call\\\":{\\\"SerialNumber\\\": \\\"0f2649bb-64a4-459f-a6c3-c6836c9883c3\\\",\\\"BoxSerialNumber\\\"": \"Disconnected\",
15
```

REST API – Example

Diagnostic Logs

- Sending a GET method we will see the current status of the diagnostic logs.
- Using the PUT method we can Start/Stop/Mark/Collect/Download.

/provisioning/common/diagnosticlogging	
Facilitates updation and read of diagnostic logging using API RAML url /provisioning/common/diagnosticlogging with GET,PUT Methods	
/provisioning/common/diagnosticlogging	GET PUT

REST API – Example

Diagnostic Logs

- Example of a 200 OK response to a GET request.
- We can see that the logs are not running and that a bundle of logs is ready for download.

```
{  
  "LogEndTime": "2022-05-03 09:09:03",  
  "LoggingStatus": "Not Running",  
  "LogInitiator": "luisga@10.99.9.132",  
  "LogStartTime": "2022-05-03 09:03:31",  
  "DownloadStatus": "Ready to download",  
  "TCPDump": "On"  
}
```

REST API – Example

Diagnostic Logs

Expressway / Diagnostic Logging Mode

PUT

https://{{Exp_C1_UCDEMO}}/api/v1/provisioning/common/diagnosticlogging

Params

Authorization

Headers (10)

Body

Pre-request Script

Tests

Settings

none

form-data

x-www-form-urlencoded

raw

binary

GraphQL

JSON

1

{

"Mode": "start",

"TCPDump": "On"

}

2

3

Body

Cookies

Headers (15)

Test Results

Pretty

Raw

Preview

Visualize

JSON

1

{

"Message": "The operation was successful"

}

2

3

- Using Postman, we can send a PUT that includes the “Mode” and “TCPDump” parameters.
- As usual, starting logs in the primary node initiates the logs in all the nodes for that cluster.
- Collect/Download options need to be sent to each node.

REST API – Example

Diagnostic Logs

Expressway / Diagnostic Logging Mode


PUT ▼ https://{{Exp_C1_UCDEMO}}/api/v1/provisioning/common/diagnosticlogging


Params Authorization Headers (10) **Body** Pre-request Script Tests Settings

☐ none ☐ form-data ☐ x-www-form-urlencoded ☒ raw ☐ binary ☐ GraphQL **JSON** ▾

```
1 {"Mode": "stop"}
2 }
```

- We need to send a PUT with the “Mode” value setup to “stop” before collecting the logs.

Body Cookies Headers (15) Test Results  20

Pretty Raw Preview Visualize **JSON** ▼ 

```
1 {
2   "Message": "The operation was successful"
3 }
```

REST API – Example

Diagnostic Logs

What's next? Scripting!!!

```
master_exp = [{"fqdn": "exp-e01.ucdemolab.com", "port": "7443"}]
for master in master_exp:
    → api_peers = "https://" + master["fqdn"] + ":" + master["port"]
    + "/api/v1/provisioning/common/cluster/peers"

    response = (requests.get(api_peers, ←
auth=HTTPBasicAuth(user_exp, passwd_exp))).json()

    for item in response:
        peers.append(item['PeerAddress'] + ":" + master["port"])
```

REST API – Example

Diagnostic Logs

You only need to start/stop logs in the Publisher of each cluster.

```
#Start/Stop logs in the master nodes
```

```
for master in master_exp:
    → diag_log = "https://" + master["fqdn"] + ":" + master["port"]
    + "/api/v1/provisioning/common/diagnosticlogging"
    payload = json.dumps({"Mode": "start"})

    response = (requests.put(diag_log, ← data=payload,
auth=HTTPBasicAuth(user_exp, passwd_exp))).json()
```


REST API – Example

Diagnostic Logs

Now we can Collect/Download from all the nodes in the cluster by simply running a loop.


```
for node in peers:
    → diag_log = "https://" + node +
"/api/v1/provisioning/common/diagnosticlogging"
    payload = json.dumps({"Mode": "collect"})

    response = (requests.put(diag_log, ← data=payload,
auth=HTTPBasicAuth(user_exp, passwd_exp))).json()
```

REST API – Example

Diagnostic Logs

```
def get_filename_from_cd(cd):  
    #Get filename from content-disposition  
    if not cd:  
        return None  
    fname = re.findall('filename="(.)+"', cd)  
    if len(fname) == 0:  
        return None  
    return fname[0]
```



We use this function to find the diagnostic log file name

REST API – Example

Diagnostic Logs

```
#Download logs from the cluster

for node in peers:
    diag_log = "https://" + node +
"/api/v1/provisioning/common/diagnosticlogging"
    payload = json.dumps({"Mode": "download"})

    response = (requests.put(diag_log, data=payload,
auth=HTTPBasicAuth(user_exp, passwd_exp)))
    filename =
get_filename_from_cd(response.headers.get('content-disposition'))
    open(filename, 'wb').write(response.content)
```

Conclusion



Highlights

- Proactive monitoring is important for administrators to make decisions about their infrastructure.
- Using Collectd you can get a good idea of the state of your Expressways.
- Grafana is a powerful tool to represent time series data.
- Expressways REST API can be used to simplify daily tasks.

Technical Session Surveys

- Attendees who fill out a minimum of four session surveys and the overall event survey will get Cisco Live branded socks!
- Attendees will also earn 100 points in the Cisco Live Game for every survey completed.
- These points help you get on the leaderboard and increase your chances of winning daily and grand prizes.



Cisco Learning and Certifications

From technology training and team development to Cisco certifications and learning plans, let us help you empower your business and career. www.cisco.com/go/certs

Pay for Learning with Cisco Learning Credits

(CLCs) are prepaid training vouchers redeemed directly with Cisco.



Learn

Cisco U.

IT learning hub that guides teams and learners toward their goals

Cisco Digital Learning

Subscription-based product, technology, and certification training

Cisco Modeling Labs

Network simulation platform for design, testing, and troubleshooting

Cisco Learning Network

Resource community portal for certifications and learning



Train

Cisco Training Bootcamps

Intensive team & individual automation and technology training programs

Cisco Learning Partner Program

Authorized training partners supporting Cisco technology and career certifications

Cisco Instructor-led and Virtual Instructor-led training

Accelerated curriculum of product, technology, and certification courses



Certify

Cisco Certifications and Specialist Certifications

Award-winning certification program empowers students and IT Professionals to advance their technical careers

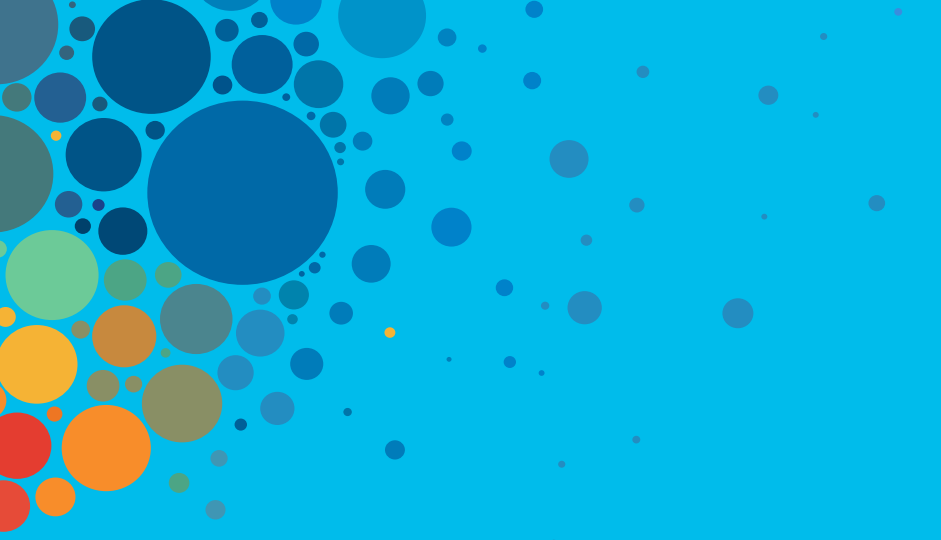
Cisco Guided Study Groups

180-day certification prep program with learning and support

Cisco Continuing Education Program

Recertification training options for Cisco certified individuals

Here at the event? Visit us at **The Learning and Certifications lounge at the World of Solutions**



Continue your education

- Visit the Cisco Showcase for related demos
- Book your one-on-one Meet the Engineer meeting
- Attend the interactive education with DevNet, Capture the Flag, and Walk-in Labs
- Visit the On-Demand Library for more sessions at www.CiscoLive.com/on-demand



The bridge to possible

Thank you

CISCO *Live!*



#CiscoLive