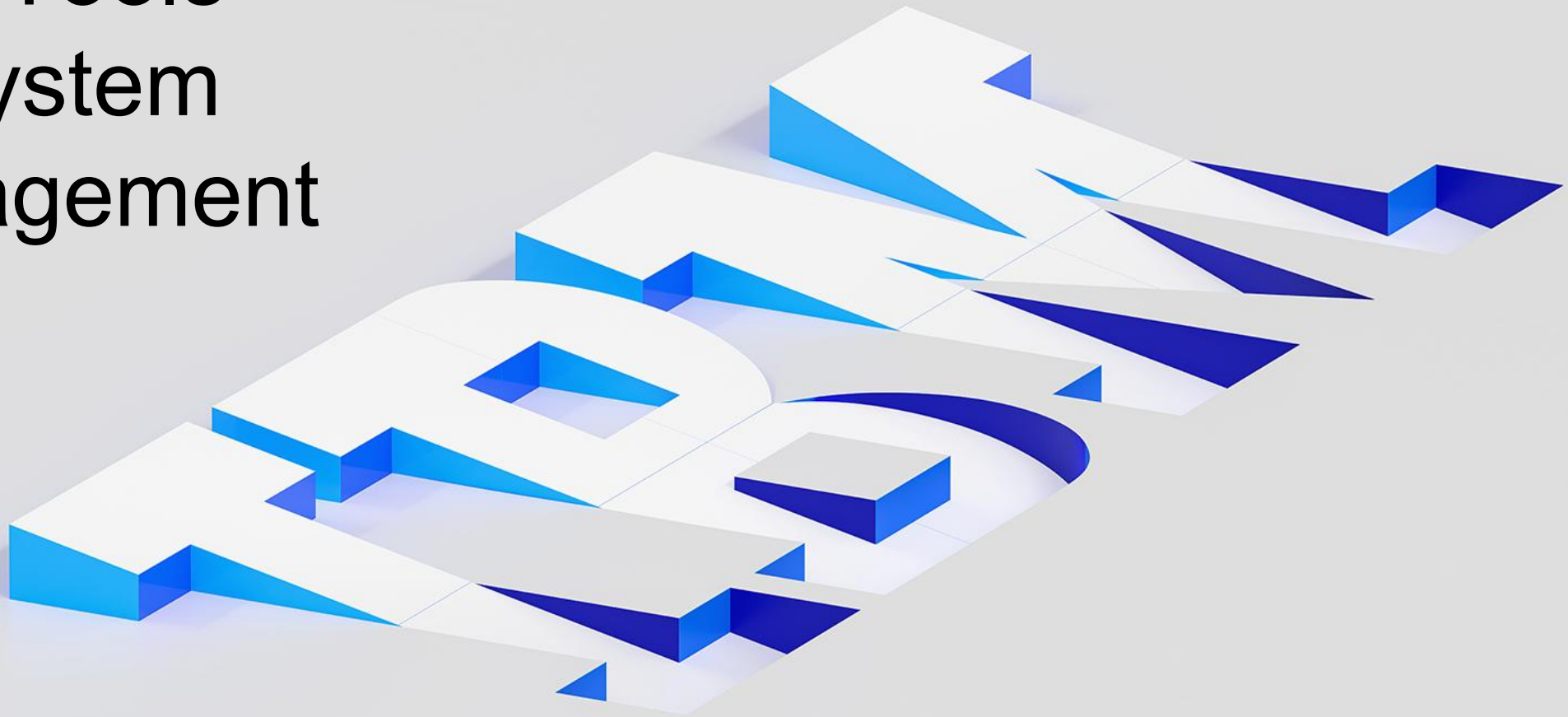
OSS Tools for System Management



Sanjula Ganepola
Software Developer – IBM i App Dev & AI Toolchain

Sanjula.Ganepola@ibm.com
2025 September

© Copyright IBM Corporation 2025

Agenda

- Current State of System Management on IBM i
- Operational Monitoring with Prometheus
- Data Visualization with Grafana
- Event Monitoring with Manzan

© Copyright IBM Corporation 2024

Current State of System Management on IBM i

© Copyright IBM Corporation 2024

What tools are you using to monitor your IBM i systems?

Choose up to 3:

1. Dynatrace
2. Nagios
3. Instana
4. DataDog
5. Control4i
6. Syslog Reporting Manager (SRM)
7. Created your own
8. Other

© Copyright IBM Corporation 2024

Specialty is a collective disadvantage

1. Dynatrace
2. Nagios
3. Instana
4. DataDog
5. Control4i
6. Syslog Reporting Manager (SRM)
7. Created your own
8. Other

Each solutions may have their own…

- Configuration
- Host installation requirements
- Monitoring capabilities
  - Collect system metrics (active jobs, ASP consumption)
  - View sub system information
  - Identify long-running SQL
  - View job queue

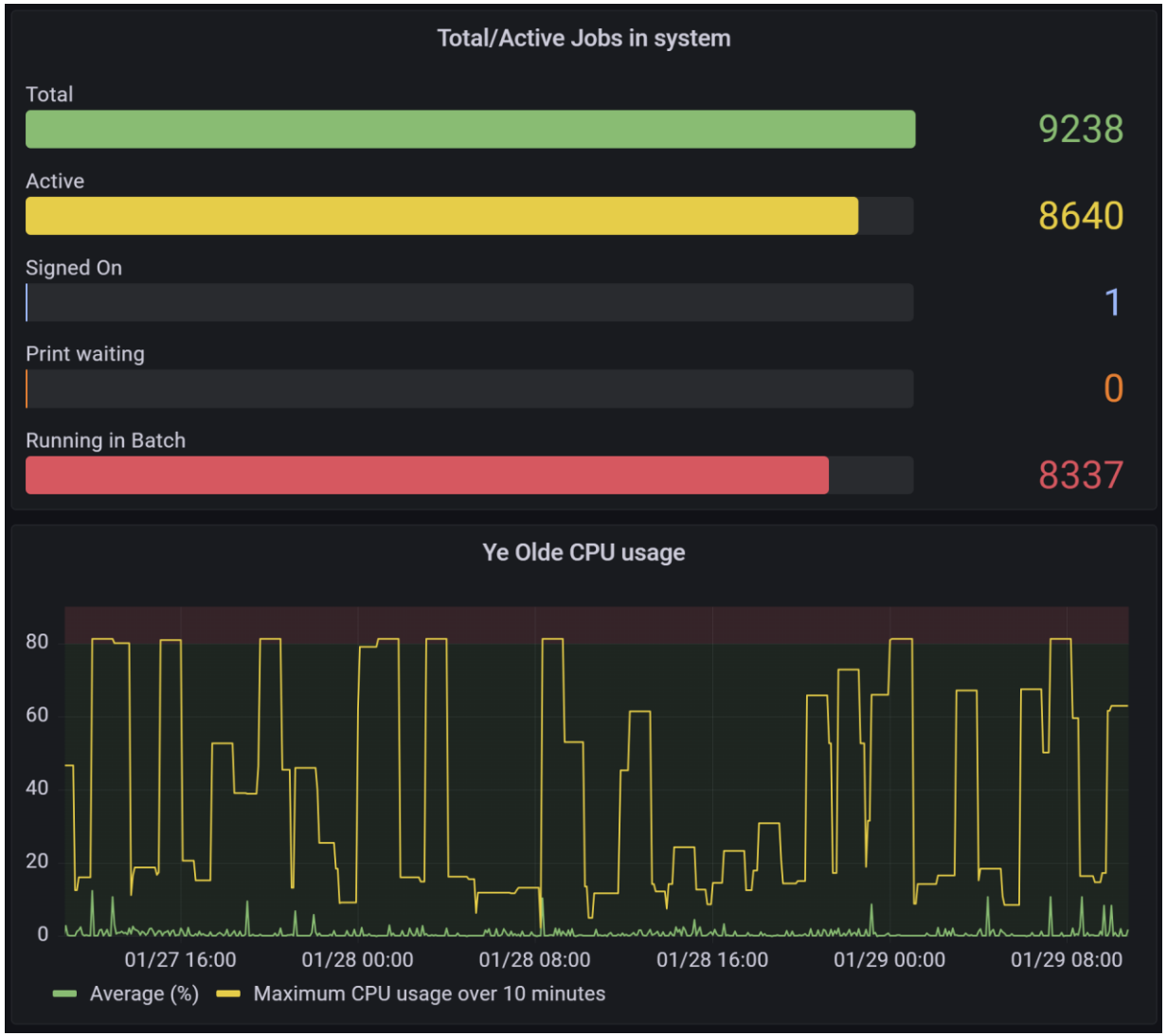
Can you integrate with other tools?

© Copyright IBM Corporation 2024

Do you use Grafana?

Choose 1:

1. Yes, with IBM i
2. Yes, but not with IBM i
3. No, but we want to
4. No, we don’t want to
5. No, don’t know what it is



© Copyright IBM Corporation 2024

Operational Monitoring with Prometheus

© Copyright IBM Corporation 2024

Prometheus Overview

What is it?

- Leading open-source systems monitoring and alerting toolkit
- Collects and stores metrics as timeseries data

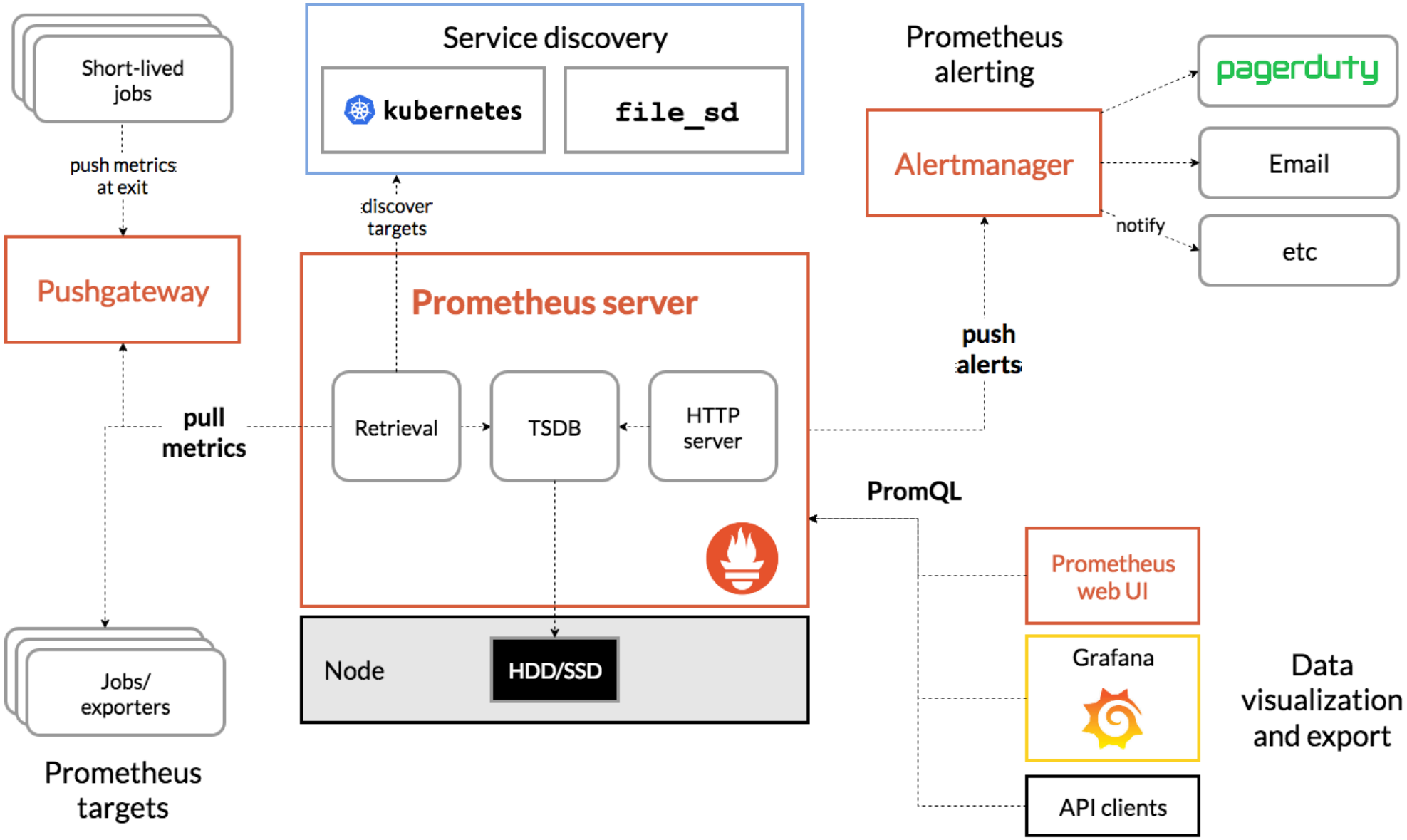
Features

- Multi-dimensional data model with time series data
- Provides a functional query language called PromQL
- No reliance on distributed storage
- Has an alert manager built-in
- Easily paired with Grafana and other monitoring solutions

Prometheus

© Copyright IBM Corporation 2024

Prometheus Architecture

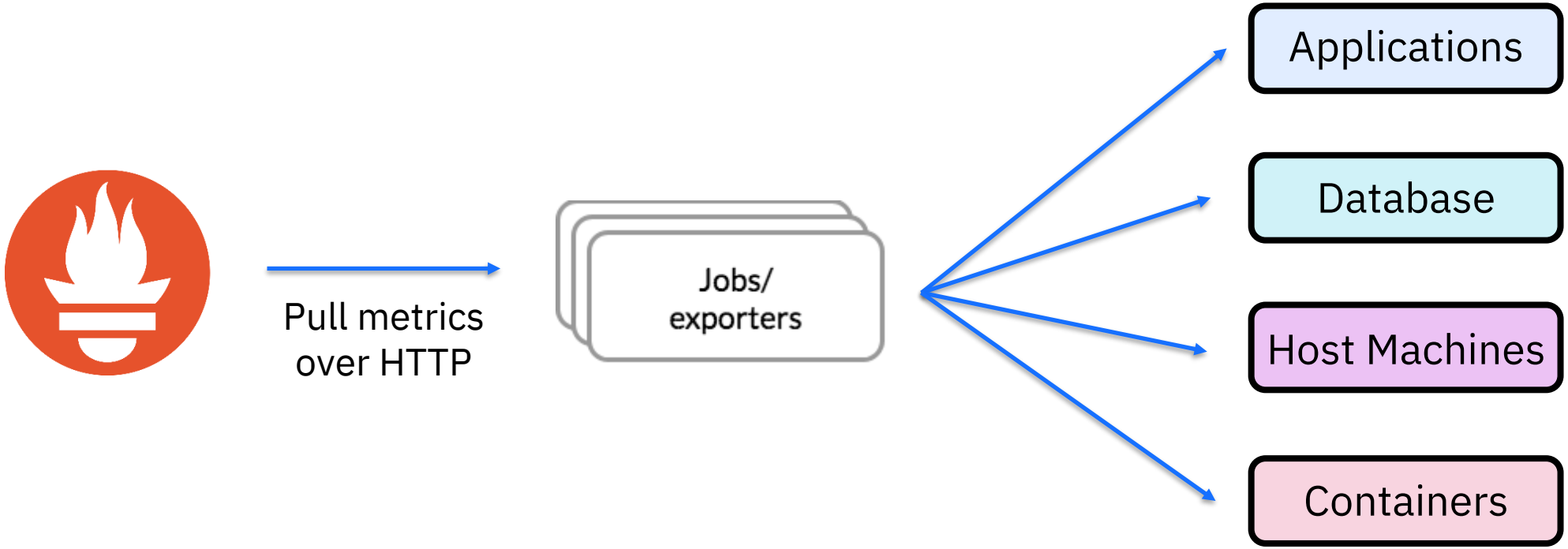


© Copyright IBM Corporation 2024

What do exporters do?

Collection of exporters: https://prometheus.io/docs/instrumenting/exporters/

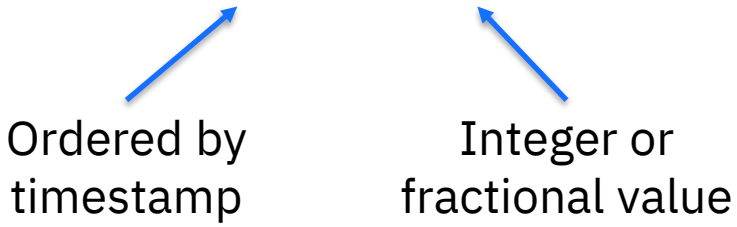
1. Fetch metrics
2. Convert them to Prometheus format
3. Expose data via an HTTP endpoint for it to be scraped



© Copyright IBM Corporation 2024

Data Model for TSDB

- All data is stored as a time series: `(timestamp, value)`



- Each time series has a name (metric name)
  - General feature of a system that is measured

- Each time series can have key/value pairs (metric labels)
  - Identifies a particular dimension of the metric

- Notation: `<metric name>{<label name>=<label value>, ...}`
  - `api_http_requests_total{method="POST", endpoint="/messages"}`

- Time series is uniquely identified by metric name + metric label
  - `temperature{city="Toronto"}`
  - `temperature{city="Rochester"}`

© Copyright IBM Corporation 2024

Understanding Time Series Data

Prometheus Database

http_request_total	GET	/
1683000000	1	
1683000020	2	
1683000040	4	
1683000060	1	
1683000080	6	
1683000100	2	
1683000120	3	

http_request_total	POST	/item
1683000000	26	
1683000020	21	
1683000040	14	
1683000060	3	
1683000080	28	
1683000100	12	
1683000120	22	

http_request_total	PUT	/item
1683000000	4	
1683000020	5	
1683000040	6	
1683000060	1	
1683000080	2	
1683000100	10	
1683000120	7	

© Copyright IBM Corporation 2024

PromQL 101 – Instant Vector

Prometheus Database

http_request_total	GET	/
1683000000	1	
1683000020	2	
1683000040	4	
1683000060	1	
1683000080	6	
1683000100	2	
1683000120	3	

http_request_total	POST	/item
1683000000	26	
1683000020	21	
1683000040	14	
1683000060	3	
1683000080	28	
1683000100	12	
1683000120	22	

http_request_total	PUT	/item
1683000000	4	
1683000020	5	
1683000040	6	
1683000060	1	
1683000080	2	
1683000100	10	
1683000120	7	

🔎 http_request_total →

http_request_total{method="GET", endpoint="/"}	3
http_request_total{method="POST", endpoint="/item"}	22
http_request_total{method="PUT", endpoint="/item"}	7

© Copyright IBM Corporation 2024

PromQL 101 – @ modifier

Prometheus Database

http_request_total	GET	/
1683000000	1	
1683000020	2	
1683000040	4	
1683000060	1	
1683000080	6	
1683000100	2	
1683000120	3	

http_request_total	POST	/item
1683000000	26	
1683000020	21	
1683000040	14	
1683000060	3	
1683000080	28	
1683000100	12	
1683000120	22	

http_request_total	PUT	/item
1683000000	4	
1683000020	5	
1683000040	6	
1683000060	1	
1683000080	2	
1683000100	10	
1683000120	7	

🔎 http_request_total@1683000040 →

http_request_total{method="GET", endpoint="/"}	4
http_request_total{method="POST", endpoint="/item"}	14
http_request_total{method="PUT", endpoint="/item"}	6

© Copyright IBM Corporation 2024

PromQL 101 – Label Selection

Prometheus Database

http_request_total	GET	/
1683000000	1	
1683000020	2	
1683000040	4	
1683000060	1	
1683000080	6	
1683000100	2	
1683000120	3	

http_request_total	POST	/item
1683000000	26	
1683000020	21	
1683000040	14	
1683000060	3	
1683000080	28	
1683000100	12	
1683000120	22	

http_request_total	PUT	/item
1683000000	4	
1683000020	5	
1683000040	6	
1683000060	1	
1683000080	2	
1683000100	10	
1683000120	7	

🔎 http_request_total{endpoint="/item"} → http_request_total{method="POST", endpoint="/item"} 22
http_request_total{method="PUT", endpoint="/item"} 7

© Copyright IBM Corporation 2024

PromQL 101 – Range Vectors

Prometheus Database

http_request_total	GET	/
1683000000	1	
1683000020	2	
1683000040	4	
1683000060	1	
1683000080	6	
1683000100	2	
1683000120	3	

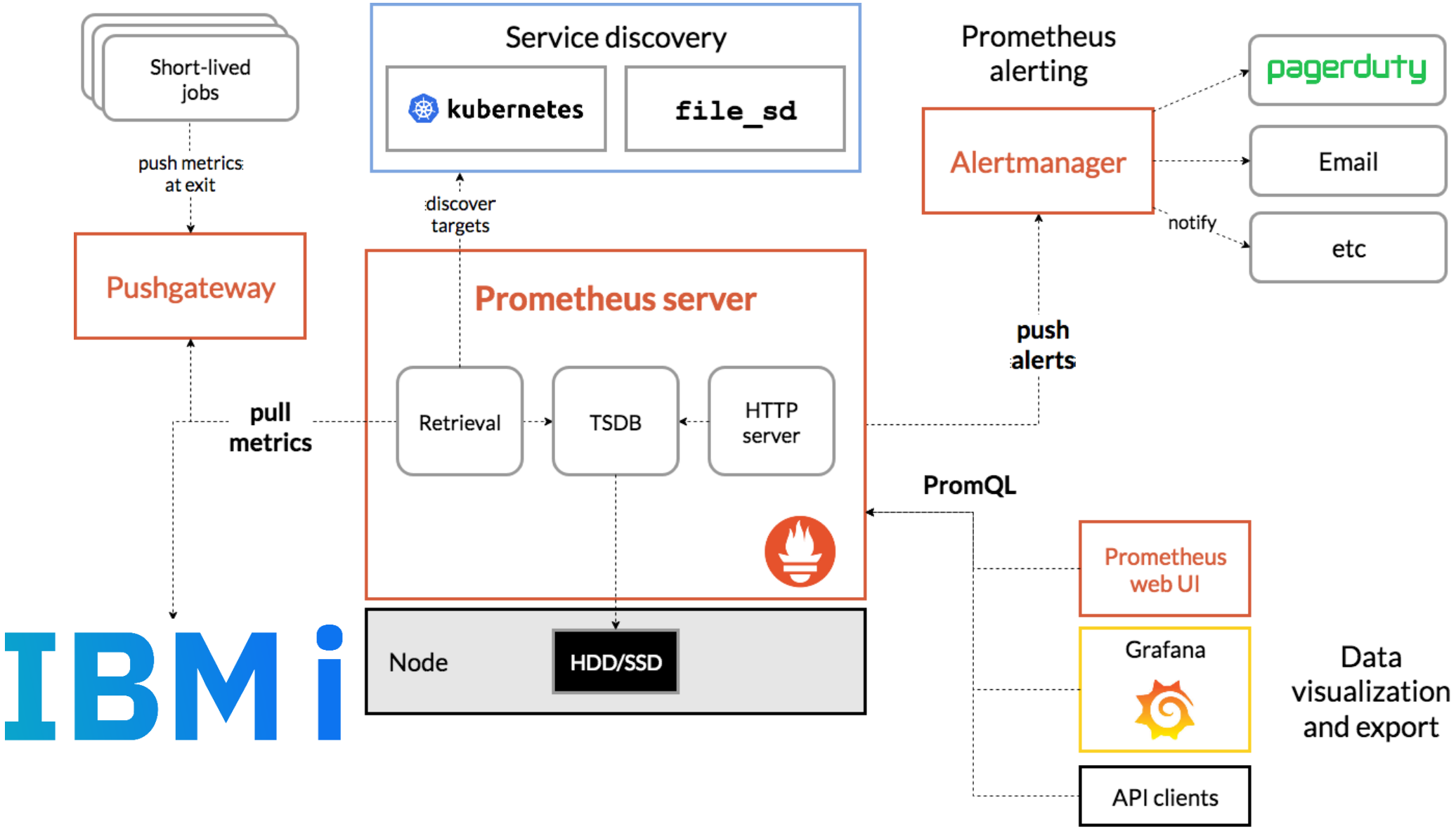
http_request_total	POST	/item
1683000000	26	
1683000020	21	
1683000040	14	
1683000060	3	
1683000080	28	
1683000100	12	
1683000120	22	

http_request_total	PUT	/item
1683000000	4	
1683000020	5	
1683000040	6	
1683000060	1	
1683000080	2	
1683000100	10	
1683000120	7	

🔎 http_request_total{method="GET"}[30s] → http_request_total{method="GET", endpoint="/"} 2@1683000100
3@1683000120

© Copyright IBM Corporation 2024

How to use Prometheus with IBM i



© Copyright IBM Corporation 2024

IBM i Prometheus Exporter

- JDBC-based exporter that leverages SQL to export IBM i metrics
- Exports hundreds of metrics including:
  - System activity levels (such as CPU usage)
  - Memory pool information
  - Apache HTTP server metrics
  - Number of remote TCP connections
- User customizable metrics with SQL

https://github.com/ThePrez/prometheus-exporter-jdbc

config.json

home > SANJULA > jar > prometheus > config.json > ...

```
{
  "port": 9853,

  "queries": [{
      "name": "System Statistics",
      "interval": 60,
      "enabled": true,
      "prefix": "STATS",
      "sql": "SELECT * FROM TABLE(QSYS2.SYSTEM_STATUS(RESET_STATISTICS=>'YES',DETAILED_INFO=>'ALL')) X"
    },
    {
      "name": "System Activity",
      "interval": 20,
      "prefix": "SYSACT",
      "enabled": true,
      "sql": "SELECT * FROM TABLE(QSYS2.SYSTEM_ACTIVITY_INFO())"
    },
    {
      "name": "number of remote connections",
      "interval": 30,
      "enabled": true,
      "sql": "select COUNT(REMOTE_ADDRESS) as REMOTE_CONNECTIONS from qsys2.netstat_info where TCP_STATE = 'ESTABLISHED'
      AND REMOTE_ADDRESS != '::1' AND REMOTE_ADDRESS != '127.0.0.1'"
    },
    {
      "name": "Memory Pool Info",
      "interval": 100,
      "enabled": true,
      "multi_row": true,
      "prefix": "MEMPOOL",
      "sql": "SELECT POOL_NAME,CURRENT_SIZE,DEFINED_SIZE,MAXIMUM_ACTIVE_THREADS,CURRENT_THREADS,RESERVED_SIZE FROM TABLE
      (QSYS2.MEMORY_POOL(RESET_STATISTICS=>'YES')) X"
    }
  ]
}
```


© Copyright IBM Corporation 2024

Deploying the IBM i Prometheus Exporter

Video Demo: https://www.youtube.com/watch?v=qIcAx4lJZIA

1. Download latest .jar file from release page and store in the IFS:

```
wget https://github.com/ThePrez/prometheus-exporter-jdbc/releases/latest/download/prom-client-ibmi.jar
```

2. If this is your first time, create a default configuration file (config.json):

```
java -jar prom-client-ibmi.jar
```

3. Respond with y for the following message:

```
Configuration file config.json not found. Would you like to initialize one with defaults? [y]
```

4. Confirm the Prometheus client has started:

```
==============================================================
Successfully started Prometheus client on port 9853
==============================================================
```

5. Create a service commander configuration file (prometheus.yml) and start it:

```
java -jar prom-client-ibmi.jar sc
sc start prometheus.yml
```


© Copyright IBM Corporation 2024

Deploying the Prometheus Instance

1. Download the Prometheus-provided image from Docker Hub to some central location:

```
docker pull prom/prometheus
```

or

```
podman pull docker.io/prom/prometheus
```

2. Create a Prometheus configuration file (prometheus.yml):

prometheus.yml U

prometheus.yml

```yml
scrape_configs:
  - job_name: 'prometheus'
    static_configs:
      - targets: ['9.5.12.241:9853']
```

3. Launch the image in a container named "prometheus":

```
docker run -d --name=prometheus -v /path/to/prometheus.yml:/etc/prometheus/prometheus.yml -p 9090:9090
prom/prometheus
```

or

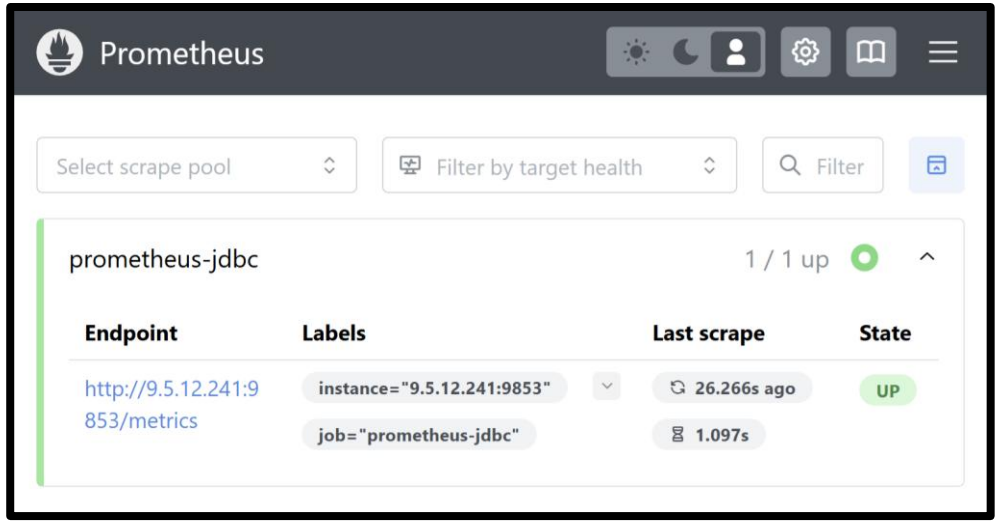
```
podman run -d --name=prometheus -v /path/to/prometheus.yml:/etc/prometheus/prometheus.yml -p 9090:9090
prom/prometheus
```

4. Navigate to Prometheus web UI at http://<container_ip>:9090

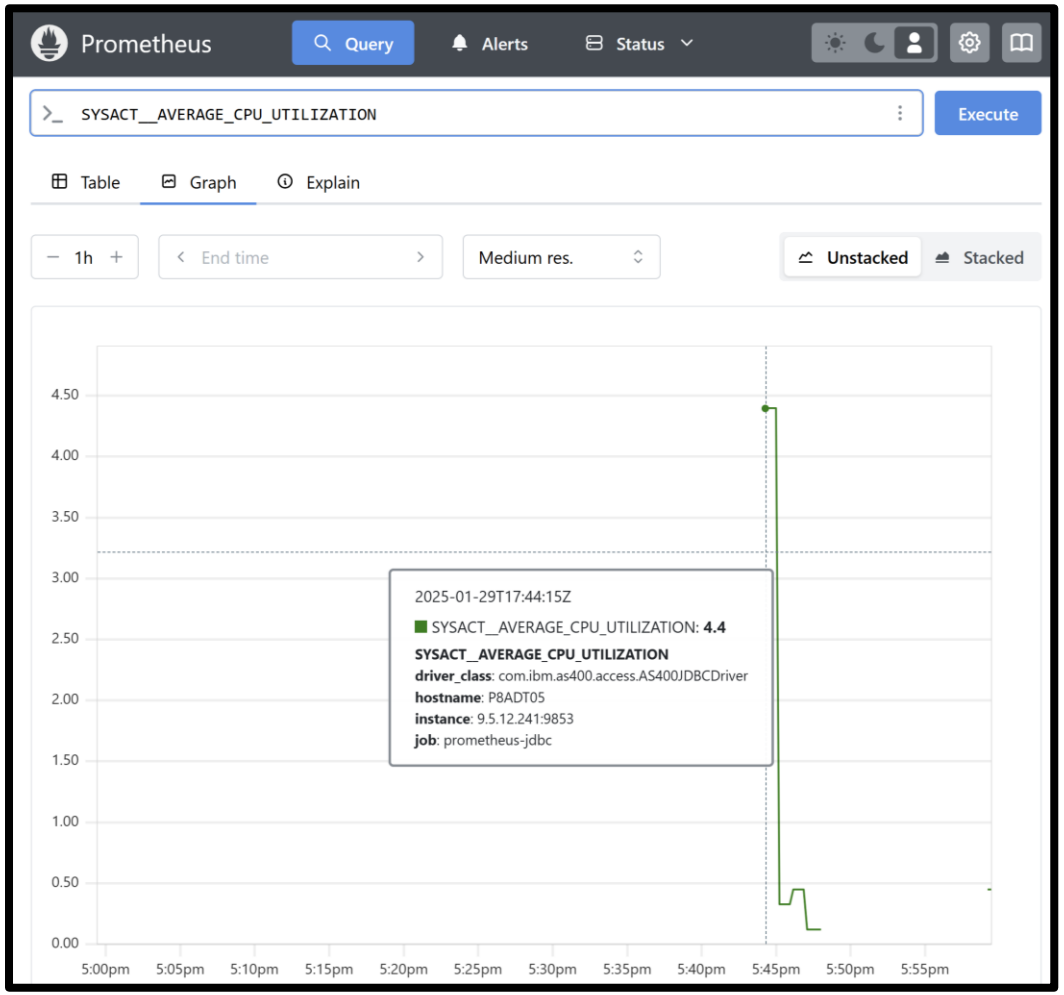
© Copyright IBM Corporation 2024

Prometheus Web UI

Target System Status



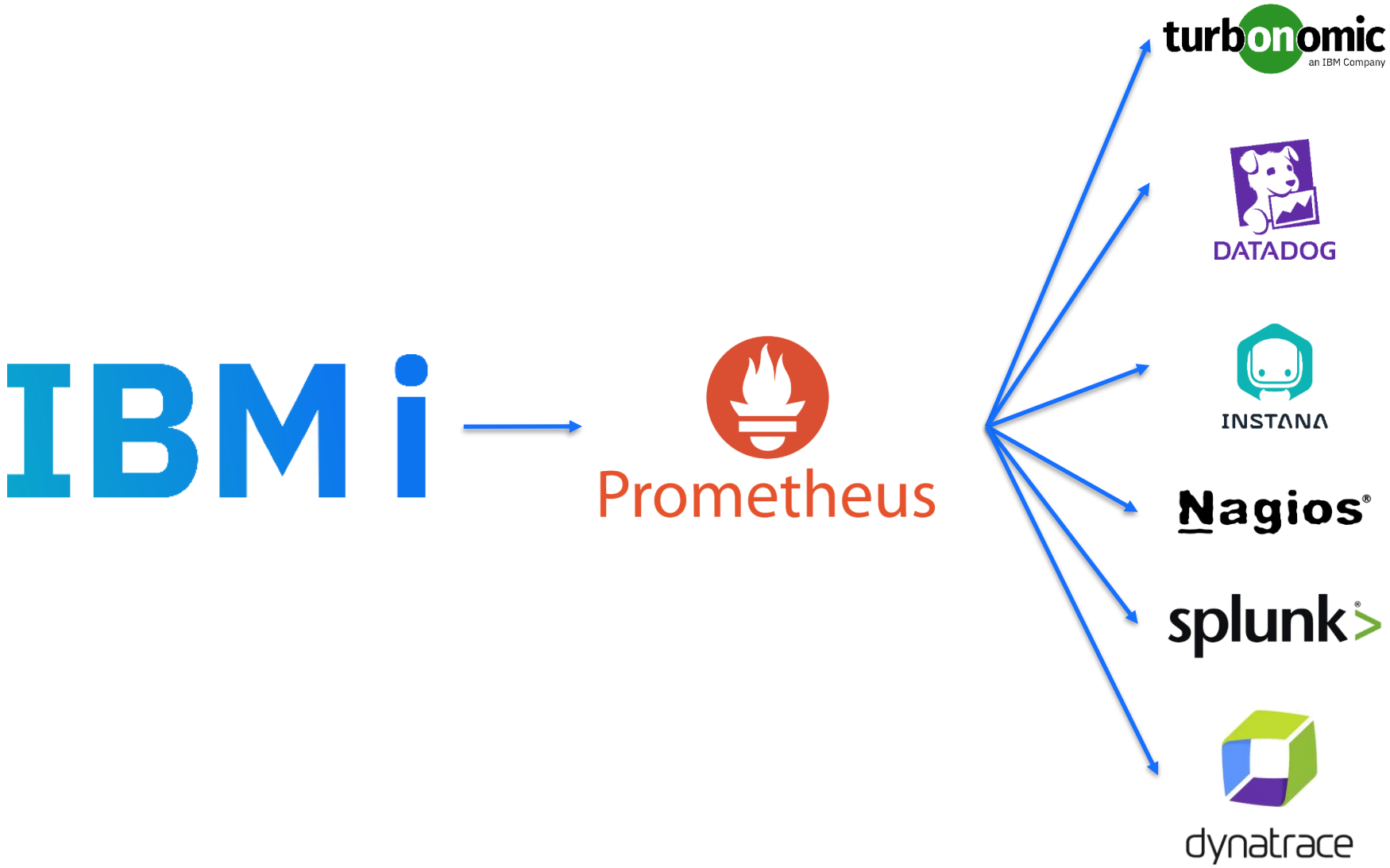
Graphing Historical Metric Data



Blog post by Jesse Gorzinski: "Monitoring IBM i with Prometheus"
https://techchannel.com/open-source-on-ibm-i/monitoring-ibm-i-with-prometheus/

© Copyright IBM Corporation 2024

Prometheus can be the bridge to other solutions



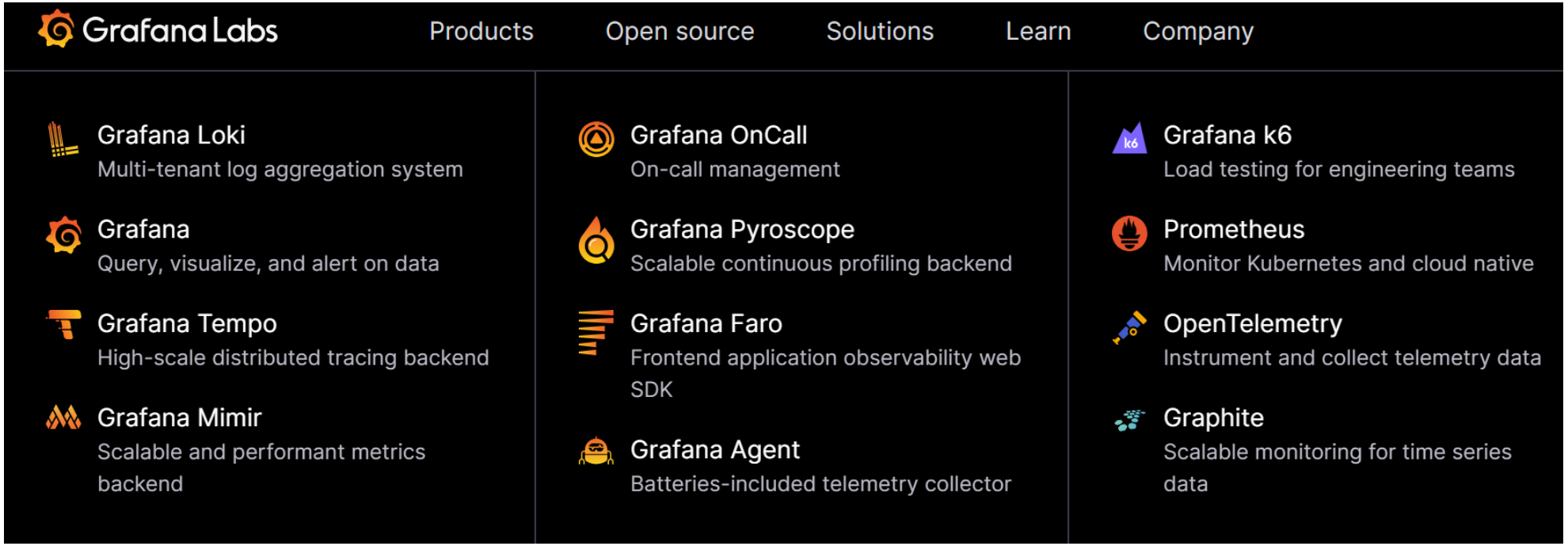
© Copyright IBM Corporation 2024

Data Visualization with Grafana

© Copyright IBM Corporation 2024

Grafana Overview

- Open-source analytics, visualization, and monitoring solution
- Driven by Grafana Labs. Plenty of open-source components. See https://grafana.com/
- Over 300 plugins available



© Copyright IBM Corporation 2024

Grafana Data Sources

Time series databases

Prometheus
Open source time series database & alerting
Core

Graphite
Open source time series database
Core

InfluxDB
Open source time series database
Core

OpenTSDB
Open source time series database
Core

SQL

MySQL
Data source for MySQL databases
Core

Microsoft SQL Server
Data source for Microsoft SQL Server compatible databases
Core

PostgreSQL
Data source for PostgreSQL and compatible databases
Core

Cloud

Azure Monitor
Data source for Microsoft Azure Monitor & Application Insights
Core

CloudWatch
Data source for Amazon AWS monitoring service
Core

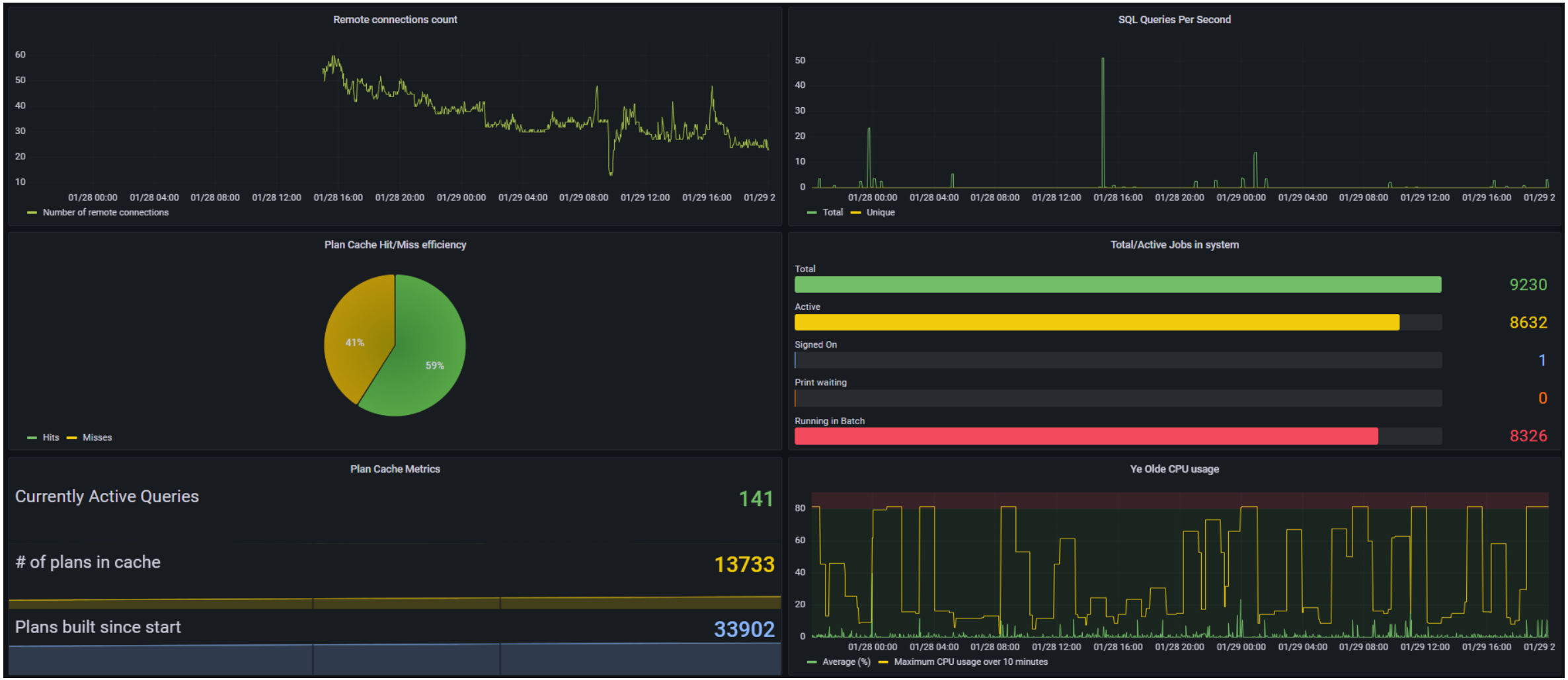
Logging & document databases

Loki
Like Prometheus but for logs. OSS logging solution from Grafana Labs
Core

Elasticsearch
Open source logging & analytics database
Core

© Copyright IBM Corporation 2024

Prometheus Visualization with Grafana



Public COMMON1 Grafana Dashboard: http://ibm.biz/ibmi-prometheus

© Copyright IBM Corporation 2024

Grafana Backend (Without Prometheus)

SERVER_NAME	HTTP_FUNCTION	SERVER_NORMAL_CONNECT	SERVER_ACTIVE_THREADS	SERVER_IDLE_THREADS	BYTES_RECEIVED	BYTES_SENT	NONCACHE_PROCESSING_T	CACHE_PROCESSING_TIME
ADMIN	SERVER HANDLED	1520	0	5	1086170	868072	3.72	0
ADMIN	CGI	1520	0	5	42008	3612	1.63	0
ADMIN	WEBSPHERE	1520	0	5	4129509	28220366	0	3071

Blog post by Jesse Gorzinski: "How to Deploy a Simple Grafana Dashboard“
https://techchannel.com/performance/how-to-deploy-a-simple-grafana-dashboard/

© Copyright IBM Corporation 2024

Grafana with or without Prometheus

	With Prometheus	Straight to Grafana
Persistent storage	Prometheus	Grafana
Persistent storage of unused metrics	Prometheus	--
Metric type	Numerics	Numerics/strings/other
Ecosystem	Extremely Broad	There
Scalability	Excellent	Good
IBM i requisites	None	Node.js
Initial setup	Easier	Easy

© Copyright IBM Corporation 2024

Event Monitoring with Manzan

© Copyright IBM Corporation 2024

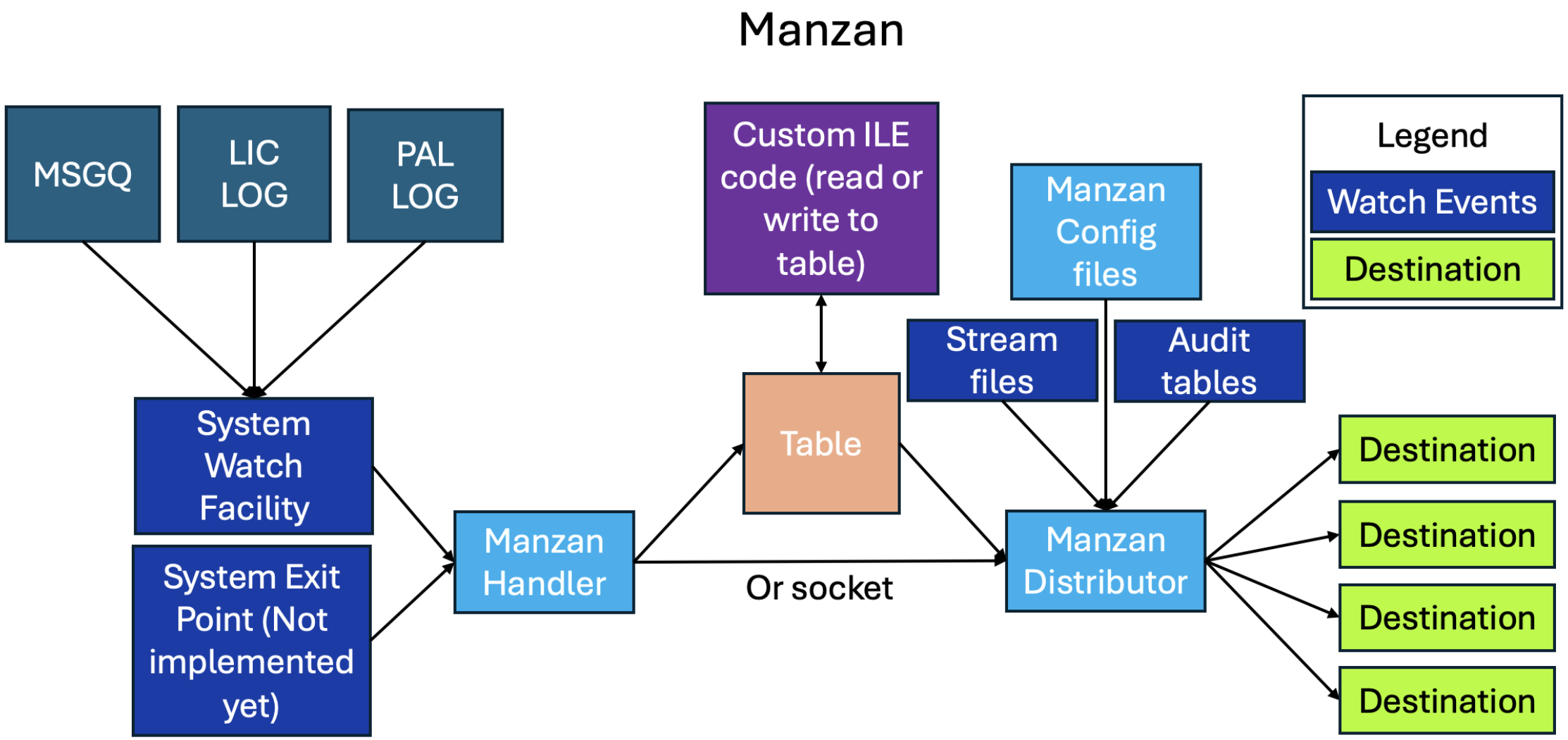
Manzan Overview

- Open-source event handling tool designed to simplify handling of system events
- Serves as a gateway for publishing IBM i events to a variety of endpoints:
  - User applications
  - External resources
  - Open-source technologies
- Example use cases:
  - Monitor system events with a third-party open source or proprietary tool
  - Consolidate logs for auditing/reporting + query for system events
  - Feed real-time system event data into stream processing systems (such as Apache Kafka)
  - Send alerts for system events

Full Documentation: https://theprez.github.io/Manzan/#/

© Copyright IBM Corporation 2024

Understanding the Architecture

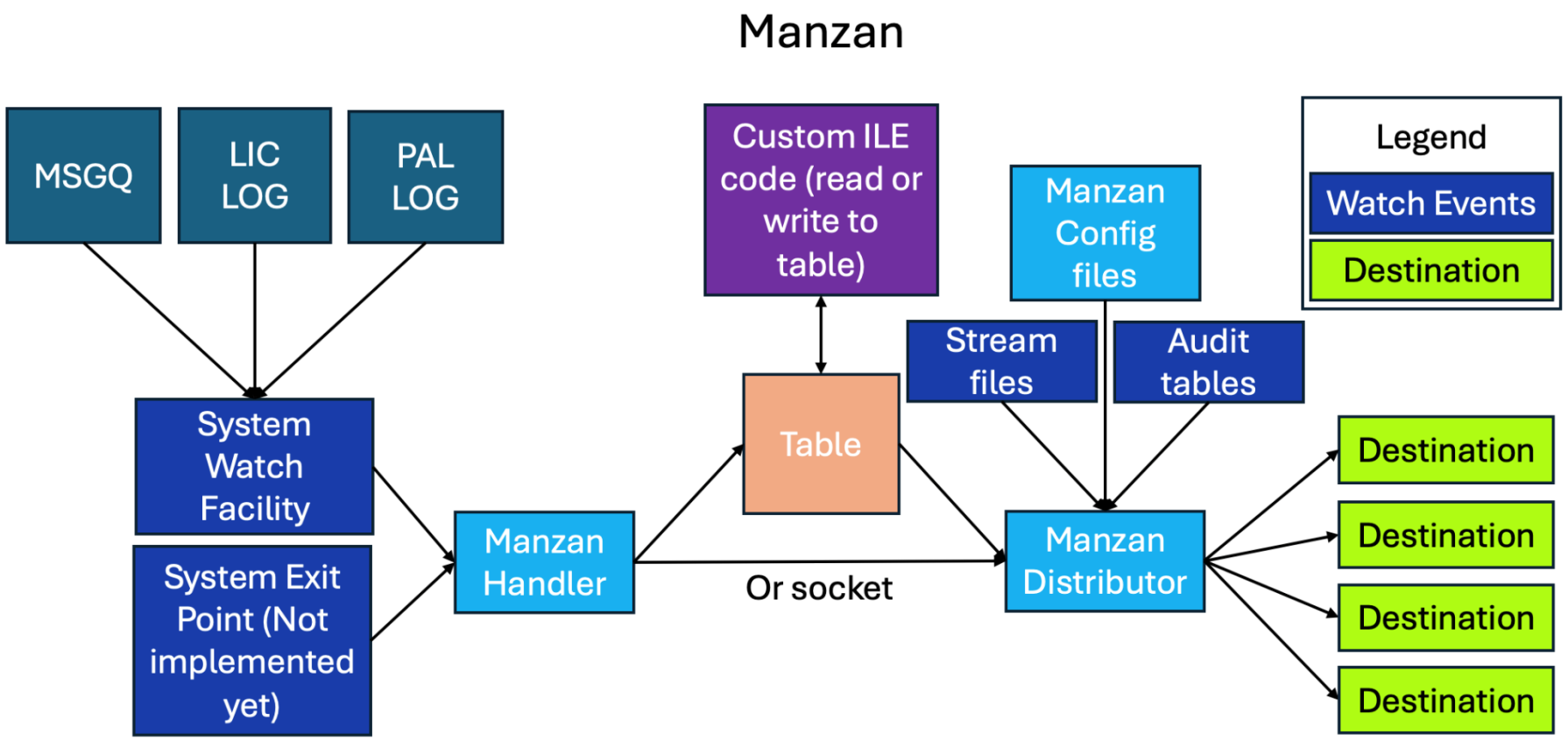


© Copyright IBM Corporation 2024

What are inputs?

Inputs: Sources of your data

- Stream file
- System watch facility (STRWCH)
  - MSGQ
  - LIC logs
  - PAL logs
- Audit journals
- SQL tables
- Arbitrary SQL & CMDs
- System exit points (*coming soon*)



© Copyright IBM Corporation 2024

So practically what can Manzan monitor?

Application crashes	Log data	Specific entries in log data	System Limits alerts
History Log entries	Problem log entries	*SYSOPR messages	Specific job log messages
Audit journal events (future)	PAL entries	VLOGs	Fishy TCP connections (future)

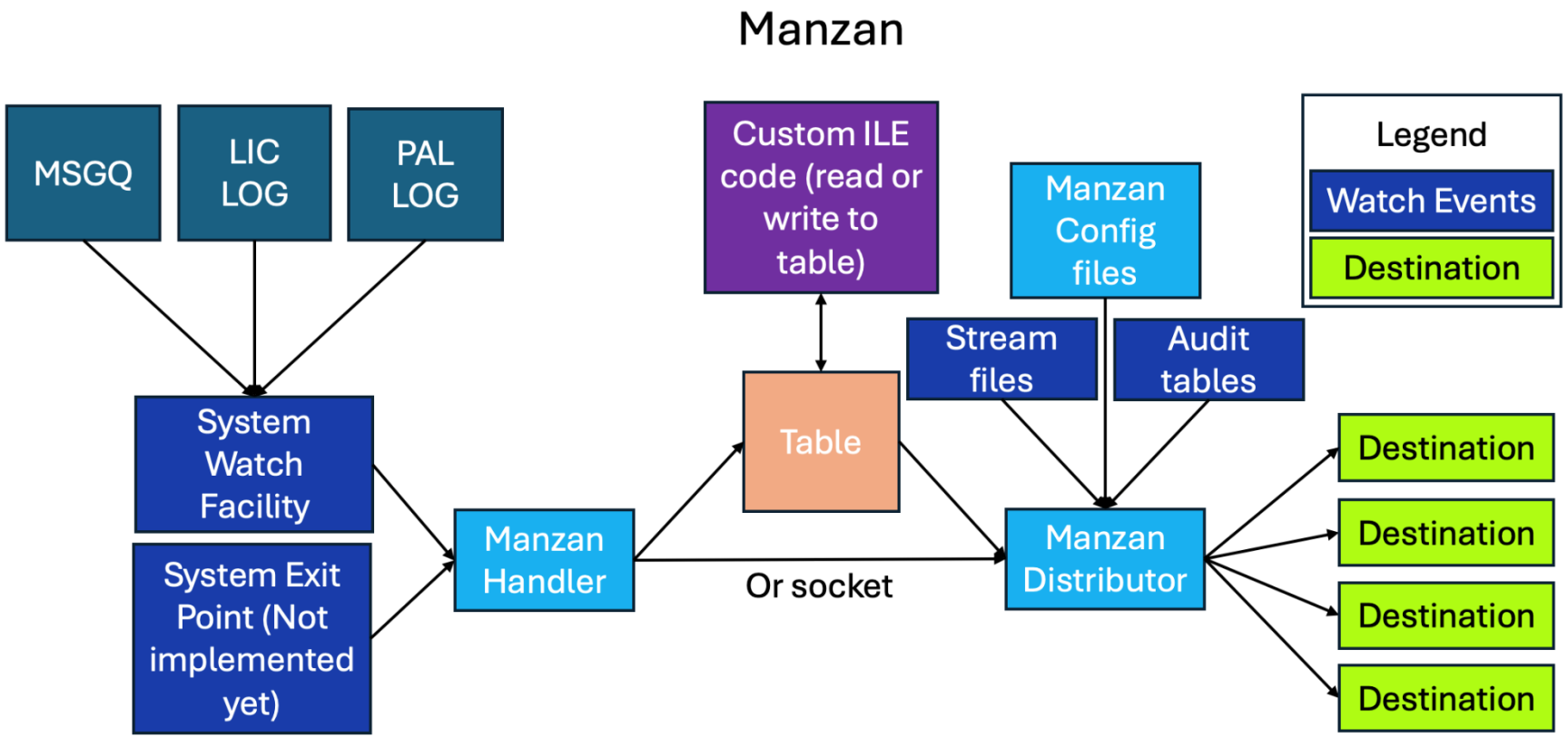
© Copyright IBM Corporation 2024

What are destinations?

Destinations: Locations to send data

- Built-in supported destinations:
  - HTTP/HTTPS endpoints (REST, etc.)
  - Email (SMTP/SMTPS)
  - SMS (via Twilio)
  - Slack
  - FluentD
  - Kafka
  - Sentry
  - Grafana Loki
  - Google Pub/Sub
  - ActiveMQ

- Custom ILE Code



© Copyright IBM Corporation 2024

Can Manzan send data anywhere?

- Many destinations are already working, but there are more to come
- Track supported destinations: https://theprez.github.io/Manzan/#/?id=where-can-i-send-these-events
- Desired target not on the list? Please open an issue to the repository and let us know!

- ActiveMQ ✅
- AWS Simple Email Service (SES) ⏳
- AWS Simple Notification System (SNS) ⏳
- Elasticsearch ✅
- Email (SMTP/SMTPS) ✅
- FluentD ✅
- Google Drive ⏳
- Google Pub/Sub ✅
- Grafana Loki ✅
- HTTP endpoints (REST, etc) ✅
- HTTPS endpoints (REST, etc) ✅
- Internet of Things (mqtt) ⏳
- Kafka ✅
- Mezmo ✅
- Microsoft Teams ⏳
- PagerDuty ✅
- Sentry ✅
- Slack ✅
- SMS (via Twilio) ✅
- Splunk ✅

✅ = implemented 🌓 = partially implemented ⏳ = future

© Copyright IBM Corporation 2024

Understanding the Architecture: Handler and Distributor

Handler

- Start system watches based on config files (data.ini)
- Transform system event data into structured format
- Add structured data to table or send over socket

Distributor

- Fetch data off table or via socket
- Send data to destinations using Apache Camel based on config files (dests.ini)

Data From Message Queue


```
{
    "EVENT_TYPE" : "message",
    "SESSION_ID" : "SANJULA",
    "JOB" : "286747/QSECOFR/QP0ZSPWP",
    "MESSAGE_ID" : "*IMMED",
    "MESSAGE_TYPE" : "*INFO",
    "SEVERITY" : 80,
    "MESSAGE_TIMESTAMP" : "2025-01-29 22:17:23.199",
    "SENDING_USRPRF" : "SANJULA",
    "MESSAGE" : "Hello World!",
    "SENDING_PROGRAM_NAME" : "QCAPCMD",
    "SENDING_MODULE_NAME" : "          ",
    "SENDING_PROCEDURE_NAME" : ""
}
```


Data From Stream File


```
{
    "FILE_NAME" : "my_app_logs.txt",
    "FILE_PATH" : "/home/SANJULA/my_app_logs.txt",
    "FILE_DATA" : "[ERROR] Application crashed due to memory leak"
}
```


© Copyright IBM Corporation 2024

Configuring Inputs and Destinations

Configuration files are stored in /QOpenSys/etc/manzan

app.ini

Used for general Manzan configuration (leave the default contents)

app.ini

config > app.ini

```
[install]
library=MANZAN
```

data.ini

Used for configuring different inputs

data.ini

data.ini

```
[<id>]
# The type of input (file or watch):
type=<type>
# As defined in dests.ini:
destinations=<destinations>
# Optional properties:
format=<format>
enabled=<enabled>
# Other <type> specific properties
# (ie. file, strwch)...
```

dests.ini

Used for configuring different destinations

dests.ini

dests.ini

```
[<id>]
# The type of destination
# (ie. smtp, slack, twilio, sentry)...
type=<type>
# Optional properties:
format=<format>
# Other <type> specific properties
# (ie. server, webhook, from, dsn)...
```


© Copyright IBM Corporation 2024

Example data.ini configurations

Stream File

- Watch a log file (test.txt) and only take action when a line written to the file contains the string “error”

```
[logfile1]
type=file
file=test.txt
destinations=email_it, test_out
filter=error
format=$FILE_DATA$
```

System Watch

- Watch all messages in the IBM i history log and format the message into a human-sensible format

```
[watchout]
type=watch
id=jesse
destinations=test_out, slackme
format=$MESSAGE_ID$ (severity $SEVERITY$): $MESSAGE$
strwch=WCHMSG((*ALL)) WCHMSGQ((*HSTLOG))
```


© Copyright IBM Corporation 2024

More example data.ini configurations

SQL Table

- Watch when data is inserted into a table (company.announcements) and write to slack/kafka

```
[table1]
type=table
table=announcements
schema=company
destinations=slack, kafka
format=$DATA$
```

Audit Journals

- Watch for login failures on the system and write messages to slack

```
[audit1]
type=audit
destinations=slackme
auditType=PASSWORD
fallbackStartTime=1
format=Violation type: $VIOLATION_TYPE_DETAIL$, username: $AUDIT_USER_NAME$
```


© Copyright IBM Corporation 2024

Example dests.ini configurations

Send to Manzan's standard output

```
[test_out]
type=stdout
```

Send an email

```
[email_it]
type=smtps
format=Hey, check out this information!! \n\n$FILE_DATA$
  server = my.smtpserver.com
  subject = Testemail
  from=me@mycompany.com
  to=me@mycompany.com
```


© Copyright IBM Corporation 2024

More example dests.ini configurations

Send message in Slack channel:

```
[slackme]
type=slack
channel=open-source-system-status
webhook=https://hooks.slack.com/services/TA3EF58G4...
```

Send event to Sentry

```
[sentry_out]
type=sentry
dsn=https://k23de06d7bff24f18a86f33178e67291@o1963...
```

Send SMS message with Twilio

```
[twilio_sms]
type=twilio
componentOptions.sid=AC1234567890abcdef1234567890a...
componentOptions.token=8aebaa92e9a4f6d85cf5c0e6ba3...
to=+12345678901
from=+10987654321
```


© Copyright IBM Corporation 2024

Let’s put it together…

dests.ini

```
[slackme]
type=slack
channel=open-source-system-status
webhook=https://hooks.slack.com/services/TA3EF58G4...
```

data.ini

```
[watchout]
type=watch
id=jesse
destinations=slackme
format=$MESSAGE_ID$ (severity $SEVERITY$): $MESSAGE$
strwch=WCHMSG((*ALL)) WCHMSGQ((*HSTLOG))
```


© Copyright IBM Corporation 2024

What does it look like in Slack?

open-source-system-status – Dec 20th, 2022

ibm i system monitor APP 9:01 PM

CPIAD09 (severity 0): User LINUX from client 9.163.40.192 connected to job 492988/QUSER/QZDASOINIT in subsystem QUSRWRK in QSYS on 12/20/22 20:17:06.

© Copyright IBM Corporation 2024

More examples...Sentry!

sentry.io

IBM
Liam

Projects
Issues
Performance
Profiling beta
Releases
Crons beta
User Feedback
Alerts
Discover
Dashboards

Quick Start
5 Remaining tasks

Upgrade Now
Help
What's new

Collapse

Issues

Saved Searches

All Unresolved 12 | For Review | Ignored

manzan | All Envs | 14D | is:unresolved

Resolve | Ignore | Last Seen | GRAPH: 24h 14d | EVENTS | USERS | ASSIGNEE

	EVENTS	USERS
MCH3601: Pointer not set for location referenced.Cause . . . . . : A pointer w 24946 MANZAN-C 1wk ago | 2wk old	385	1
MCH4421: Invalid operation for program.Cause . . . . . : The operation reque 26669 MANZAN-7 1wk ago | 2wk old	47	1
MCH1401: Receiving context contains duplicate object.Cause . . . . . : An at 25195 MANZAN-5 1wk ago | 2wk old	233	2
MCH3601: Pointer not set for location referenced.Cause . . . . . : A pointe 26742 MANZAN-G 1wk ago | 1wk old	1	1
MCH3401: Cannot resolve to object QTMSTRCSVR. Type and Subtype X'19 26987 MANZAN-A 1wk ago | 2wk old	29	1
MCH5801: Dequeue operation not satisfied in 15 seconds for queue &1C		

© Copyright IBM Corporation 2024

More examples...Sentry!

sentry.io

IBM
Liam

- Projects
- Issues
- Performance
- Profiling beta
- Releases
- Crons beta
- User Feedback
- Alerts
- Discover
- Dashboards

5 Quick Start
5 Remaining tasks

- Upgrade Now
- Help
- What's new
- Collapse

Additional Data

Formatted | Raw

HANDLED_TIMESTAMP	2022-12-12T17:21:17.996Z
JOB	293151/QUSER/QZDASOINIT
MESSAGE	Invalid operation for program.Cause . . . . . : The operation requested is not supported for program &1. The system supports distinct "models" of program objects. Not all operations are supported for all models of programs. For example, some program objects cannot be deactivated.
MESSAGE_ID	MCH4421
MESSAGE_TIMESTAMP	2022-12-12T17:21:17.936Z
MESSAGE_TYPE	*ESCAPE
ORDINAL_POSITION	25197
SENDING_MODULE_NAME	QSQRINZ
SENDING_PROCEDURE_NAME	QSQRINZ
SENDING_PROGRAM_NAME	QSQRINZ
SENDING_USRPRF	QUSER
SESSION_ID	JESSE
SEVERITY	40

© Copyright IBM Corporation 2024

More examples…Google Pub/Sub!

my-sub – Pub/Sub – my-projec

console.cloud.google.com/cloudpubsub/subscription/detail/my-sub?project=my-project-438217&tab=messages&pli=1&invt=AbeaQg

Google Cloud | my-project | Search (/) for resources, docs, products, and more | Search

Pub/Sub

- Pub/Sub
  - Topics
  - Subscriptions
  - Snapshots
  - Schemas
- Pub/Sub Lite
  - Lite Reservations
  - Lite Topics
  - Lite Subscriptions
- Release Notes

my-sub | EDIT | CREATE SNAPSHOT | REPLAY MESSAGES | PURGE MESSAGES | DETACH | DELETE | LEARN | SHOW INFO PANEL

Subscription name	projects/my-project-438217/subscriptions/my-sub
Subscription state	active
Topic name	projects/my-project-438217/topics/my-topic

METRICS | DETAILS | MESSAGES

Publish time	Attribute keys	Message body	Ordering key	Ack
Oct 10, 2024, 3:14:09 PM	HANDLED_TIMESTAMP, JOB, MESSAGE, MESSAGE_ID, MESSAGE_TIMESTAMP, MESSAGE_TYPE, ORDINAL_POSITION, SENDING_MODULE_NAME, SENDING_PROCEDURE_NAME, SENDING_PROGRAM_NAME, SENDING_USRPRF, SESSION_ID, SEVERITY	CPIAD0B (severity 0): *SIGNON server job 540281/QUSER/QZSOSIGN processing request for user SANJULA on 10/10/24 19:10:11 in subsystem QUSRWRK in QSYS.Cause . . . . . : The *SIGNON server is processing request 1 for user SANJULA. The types of requests supported are as follows: 1 -- Retrieve Signon Information 2 -- Change Password 3 -- Generate Authentication Token 4 -- Generate Authentication Token on Behalf of Another User 5 -- Start Server	–	ACK
Oct 10, 2024, 3:14:09 PM	HANDLED_TIMESTAMP …	CPIAD09 (severity 0): User SANJULA from client 10.234.201.160 connected to job …	–	ACK
Oct 10, 2024, 3:14:09 PM	HANDLED_TIMESTAMP …	CPIAD0B (severity 0): *SIGNON server job 540179/QUSER/QZSOSIGN processing …	–	ACK
Oct 10, 2024, 3:14:09 PM	HANDLED_TIMESTAMP …	CPIAD0B (severity 0): *SIGNON server job 540329/QUSER/QZSOSIGN processing …	–	ACK
Oct 10, 2024, 3:14:09 PM	HANDLED_TIMESTAMP …	CPIAD09 (severity 0): User SANJULA from client 10.234.201.160 connected to job …	–	ACK
Oct 10, 2024, 3:14:09 PM	HANDLED_TIMESTAMP …	CPIAD09 (severity 0): User SANJULA from client 10.234.201.160 connected to job …	–	ACK
Oct 10, 2024, 3:14:09 PM	HANDLED_TIMESTAMP …	CPIAD09 (severity 0): User SANJULA from client 10.234.201.160 connected to job …	–	ACK
Oct 10, 2024, 3:14:10 PM	HANDLED_TIMESTAMP …	CPIAD09 (severity 0): User SANJULA from client 10.234.201.160 connected to job …	–	ACK

© Copyright IBM Corporation 2024

More examples…Grafana Loki!

Video Demo: https://www.youtube.com/watch?v=3XXjvkchCWc

Explore - grafanacloud-sanjulag

sanjulaganepola.grafana.net/explore?schemaVersion=1&panes=%7B"ei5":%7B"datasource":"grafanacloud-logs","queries":%5B%7B"refId":"A","expr":"%7Bapp%3D%5C"manzan%5C"...

Search or jump to... ctrl+k

Home › Explore › grafanacloud-sanjulaganepola-logs

Home
Bookmarks
Starred
Dashboards
Explore
Metrics
Logs
Traces
Profiles
Alerts & IRM
Testing & synthetics
Cloud provider
Infrastructure
Application
Frontend
Connections
Add new connection
Collector
Data sources
Integrations
Private data source connect
Apps
Administration

Outline | grafanacloud-sanjulaganepol | Split | Add | Live

Queries
Logs

Manzan (grafanacloud-sanjulaganepola-logs) {app="manzan"} |= `

Add query | Query history | Query inspector

Logs volume

Logs | Logs | Table

Time | Unique labels | Wrap lines | Prettify JSON | Deduplication | None | Exact | Numbers | Signature

Display results | Newest first | Oldest first

Common labels: SENDING_MODULE_NAME= app=manzan service_name=manzan

Download

Line limit: 1000 reached, received logs cover 25.42% (15min 15sec) of your selected time range (1h)

Total bytes processed: 4.99 MB

```
2024-10-01 13:44:14.042  JOB=465707/QUSER/QZSOSIGN
                         MESSAGE_ID=CPIAD0B
                         MESSAGE_TYPE=*INFO
                         SENDING_PROGRAM_NAME=QZSOSIGN
                         SENDING_USRPRF=QUSER
                         SESSION_ID=sanjula
                         SEVERITY=0
                         CPIAD0B (severity 0): *SIGNON server job 465707/QUSER/QZSOSIGN processing re
                         quest for user SANJULA on 10/01/24 13:44:14 in subsystem QUSRWRK in QSYS.Cau
                         se . . . . . :   The *SIGNON server is processing request 1 for user SANJUL
                         A.  The types of requests supported are as follows: 1 -- Retrieve Signon Inf
                         ormation 2 -- Change Password 3 -- Generate Authentication Token 4 -- Genera
                         te Authentication Token on Behalf of Another User 5 -- Start Server
2024-10-01 13:44:14.024  JOB=465707/QUSER/QZSOSIGN
                         MESSAGE_ID=CPIAD0B
                         MESSAGE_TYPE=*INFO
                         SENDING_PROGRAM_NAME=QZSOSIGN
                         SENDING_USRPRF=QUSER
                         SESSION_ID=sanjula
                         SEVERITY=0
                         CPIAD0B (severity 0): *SIGNON server job 465707/QUSER/QZSOSIGN processing re
                         quest for user SANJULA on 10/01/24 13:44:14 in subsystem QUSRWRK in QSYS.Cau
                         se . . . . . :   The *SIGNON server is processing request 1 for user SANJUL
                         A.  The types of requests supported are as follows: 1 -- Retrieve Signon Inf
                         ormation 2 -- Change Password 3 -- Generate Authentication Token 4 -- Genera
                         te Authentication Token on Behalf of Another User 5 -- Start Server
2024-10-01 13:44:14.022  JOB=465708/QUSER/QZSOSIGN
                         MESSAGE_ID=CPIAD0B
                         MESSAGE_TYPE=*INFO
                         SENDING_PROGRAM_NAME=QZSOSIGN
                         CPIAD0B (severity 0): *SIGNON server job 465708/QUSER/QZSOSIGN
                         quest for user SANJULA on 10/01/24 13:44:14 in subsystem QUSRWR
                         se . . . . . :   The *SIGNON server is processing request 1 for user SANJUL
                         A.  The types of requests supported are as follows: 1 -- Retrieve Signon Inf
```


© Copyright IBM Corporation 2024

Visualize Logs in Grafana Dashboard

Manzan Dashboard - Dashboar

sanjulaganepola.grafana.net/d/cdz6l0myytywwf/manzan-dashboard?from=2024-10-01T17:42:55.000Z&to=2024-10-01T17:46:05.000Z&timezone=browser

Search or jump to... ctrl+k

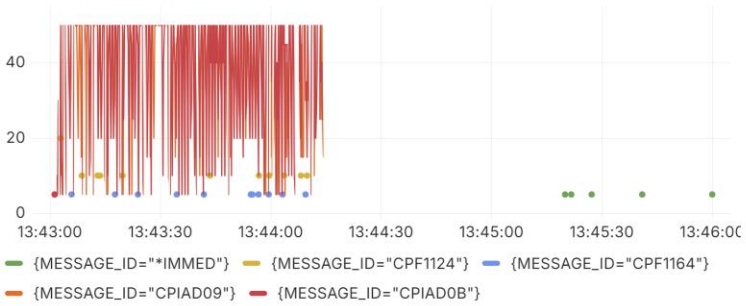
Home › Dashboards › Manzan Dashboard

Share Edit

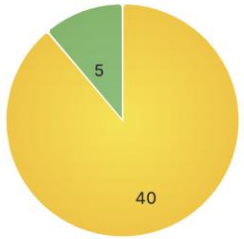
- Home
- Bookmarks
- Starred
- Dashboards
- Explore
  - Metrics
  - Logs
  - Traces
  - Profiles
- Alerts & IRM
- Testing & synthetics
- Cloud provider
- Infrastructure
- Application
- Frontend
- Connections
  - Add new connection
  - Collector
  - Data sources
  - Integrations
  - Private data source connect
- Apps
- Administration

2024-10-01 13:42:55 to 2024-10-01 13:46:05 Refresh

Message Frequency



Log Count By Level



Critical Logs

Common labels: MESSAGE_ID=*IMMED MESSAGE_TYPE=*INFO SENDING_MODULE_NAME= SENDING_PROGRAM_NAME=QCAPCMD SENDING_USRPRF=SANJULA SESSION_ID=sanjula1 SEVERITY=80 app=manzan service_name=manzan

```
› 2024-10-01 13:45:59.924 *IMMED (severity 80): Invalid operation for pro
                          gram
› 2024-10-01 13:45:40.884 *IMMED (severity 80): Pointer not set
› 2024-10-01 13:45:27.173 *IMMED (severity 80): Error in program
2x › 2024-10-01 13:45:19.860 *IMMED (severity 80): Pointer exception
```

All Logs

Common labels: SENDING_MODULE_NAME= app=manzan service_name=manzan

```
› 2024-10-01 13:45:59.924 *IMMED (severity 80): Invalid operation for pro
                          gram
› 2024-10-01 13:45:40.884 *IMMED (severity 80): Pointer not set
› 2024-10-01 13:45:27.173 *IMMED (severity 80): Error in program
2x › 2024-10-01 13:45:19.860 *IMMED (severity 80): Pointer exception
› 2024-10-01 13:44:14.045 CPIAD09 (severity 0): User SANJULA from client
                          127.0.0.1 connected to job 463761/QUSER/QZDASOI
                          NIT in subsystem QUSRWRK in QSYS on 10/01/24 1
                          3:44:14.
› 2024-10-01 13:44:14.043 CPIAD09 (severity 0): User SANJULA from client
                          127.0.0.1 connected to job 463291/QUSER/QZDASOI
                          NIT in subsystem QUSRWRK in QSYS on 10/01/24 1
                          3:44:14.
› 2024-10-01 13:44:14.042 CPIAD09 (severity 0): User SANJULA from client
                          127.0.0.1 connected to job 463530/QUSER/QZDASOI
```


© Copyright IBM Corporation 2024

Any Questions?

© Copyright IBM Corporation 2024

Sanjula Ganepola - OSS Tools For System Management

Please take the last minute of this session to complete the evaluation. A direct link to the evaluation can be found using the QR code to the right.

NAVIGATE 2025 PITTSBURGH

COMMON

IBM i

IBM