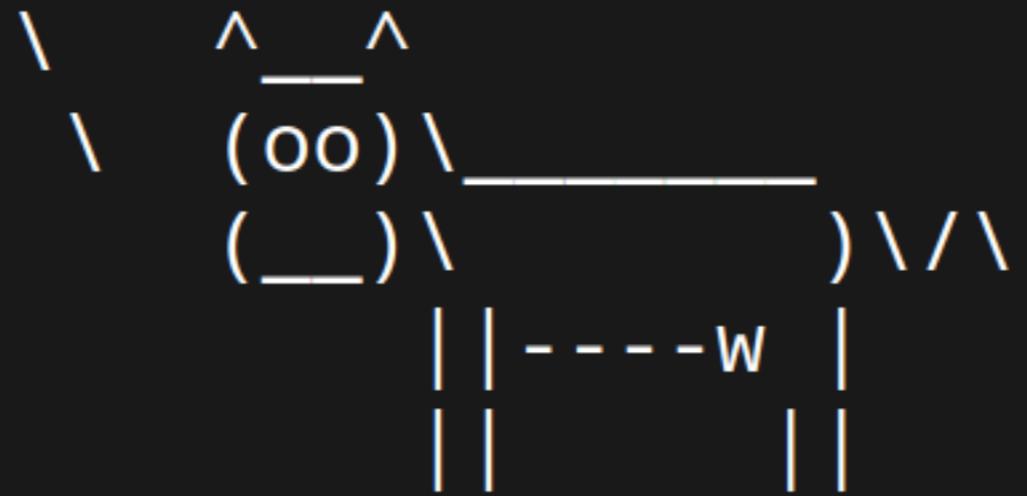


ANSIBLE METRICS IN PROMETHEUS

< Why not ? >



CONFIGURATION MANAGEMENT CAMP 2024

DAVID MOREAU SIMARD && DANIEL MELLADO

\$ cat dmsimard.yml

```
---
- name: David Moreau Simard
  hosts:
    - dmsimard:matrix.org
    - fosstodon.org/@rfc2549
  vars:
    profile: ['sysadmin', 'dev/ops', 'CI/CD', 'SRE']
  roles:
    - Ansible user since version ~1.8 (2014?)
    - Created ara in 2016 to make my life easier
    - Was in the Ansible community team at Red Hat
    - Part time contributor (between dev ops and dad ops)
```

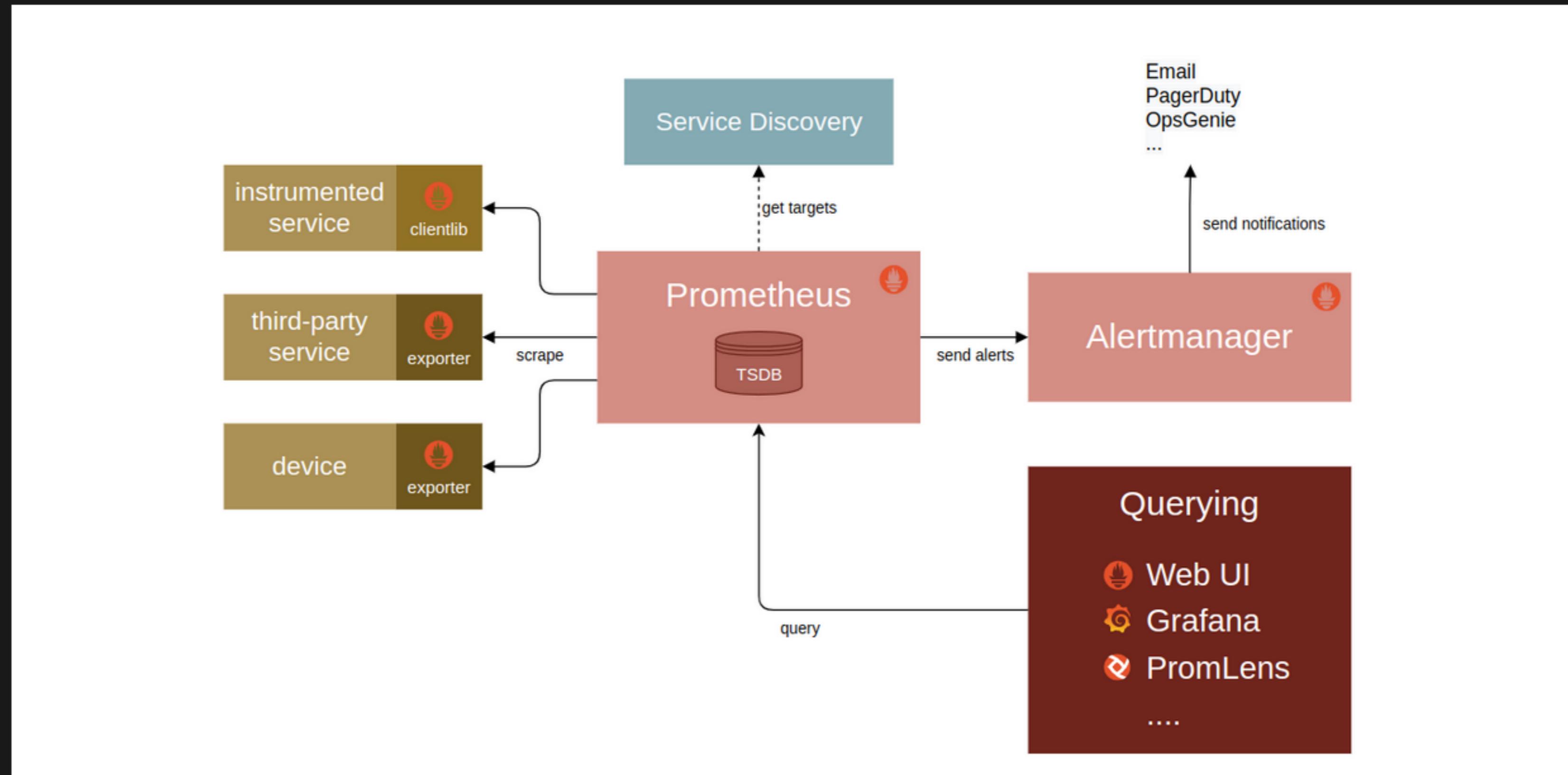
\$ cat dmellado.yml

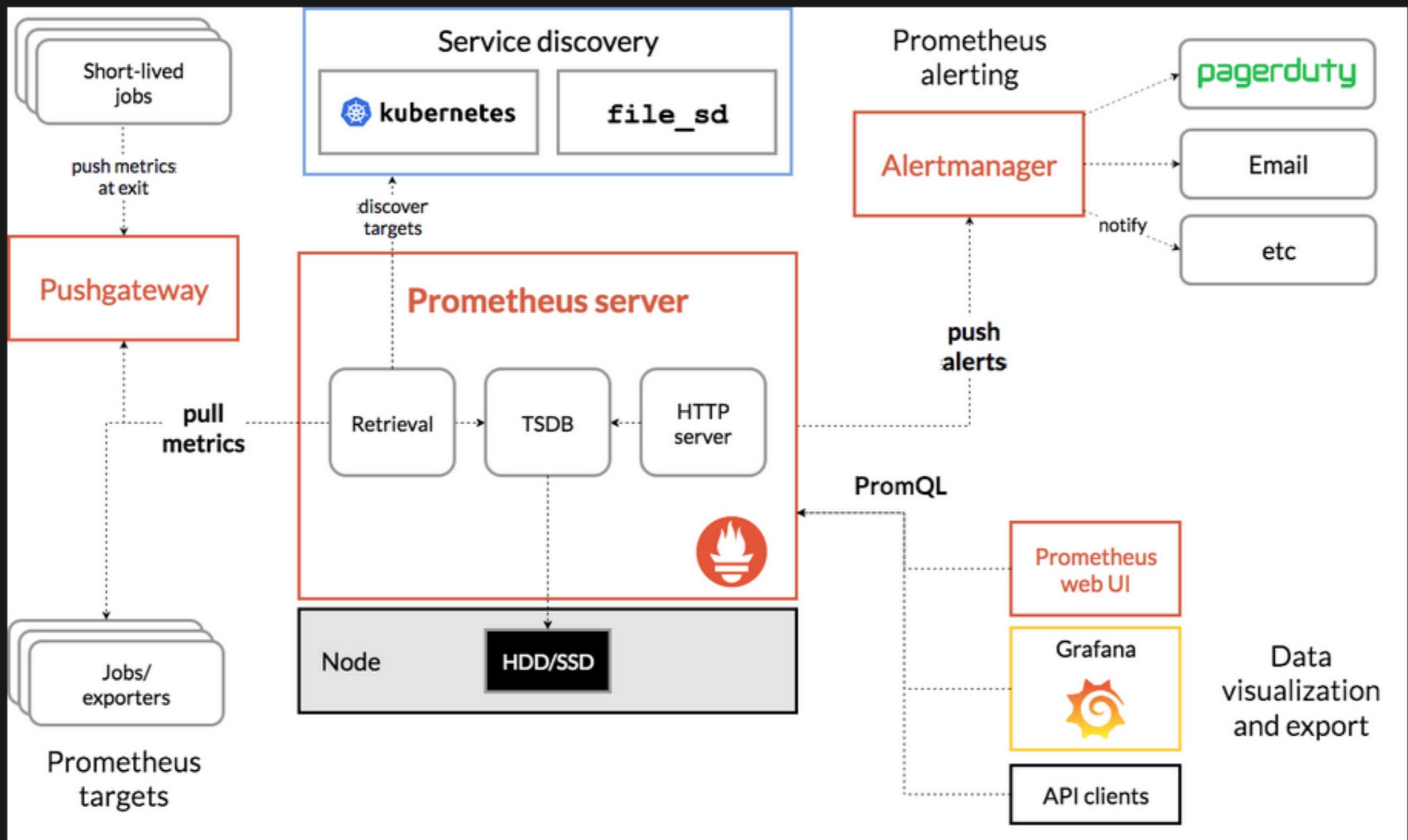
```
- name: Daniel Mellado
  hosts:
    - dmellado:matrix.org
    - dmellado@fedoraproject.org
  vars:
    profile: ['network', 'cloud', 'observability', 'fedora']
  roles:
    - Fedora eBPF SIG group coordinator
    - Long time stacker
    - Prometheus Operator Contributor
```

Do you use Prometheus or Grafana ?

o/

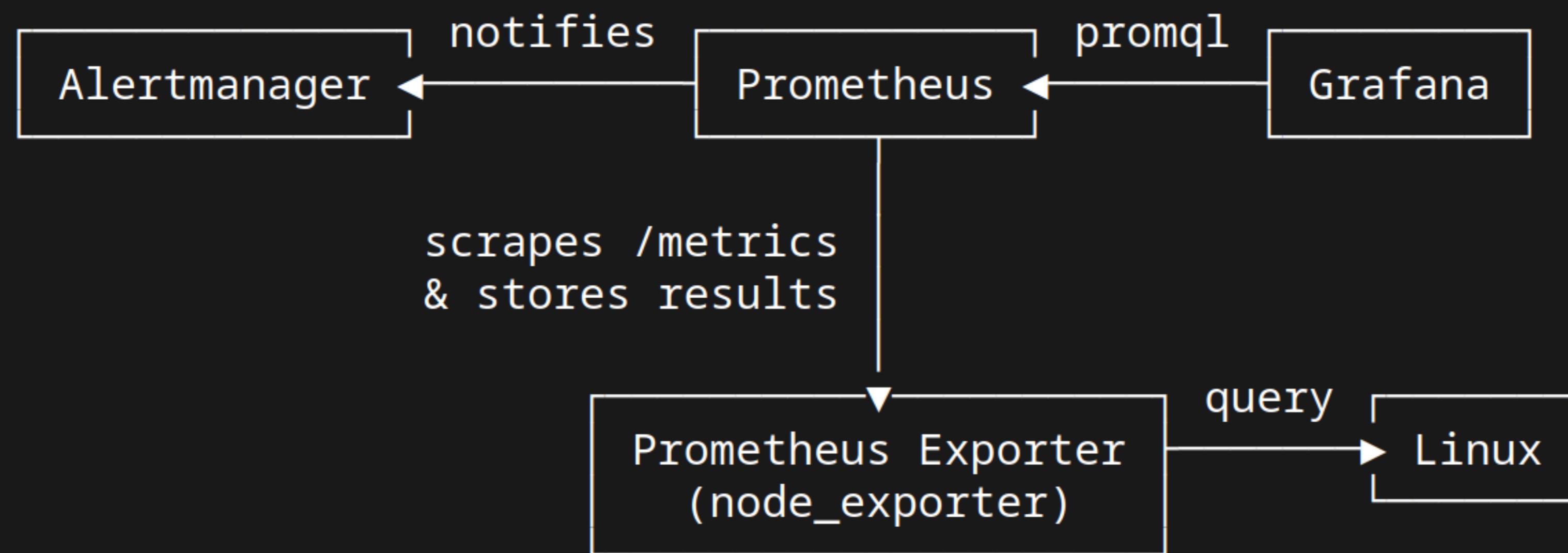
PROMETHEUS



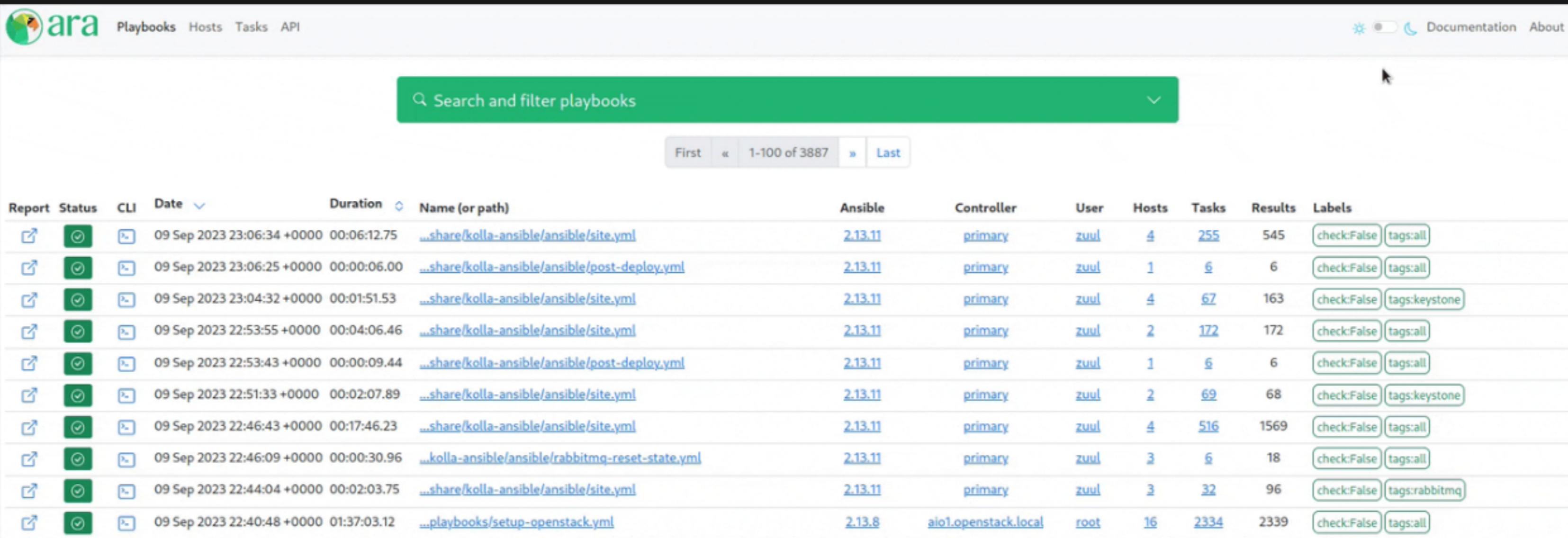


<https://prometheus.io/docs/introduction/overview/>

TL;DR



WITH ARA ?

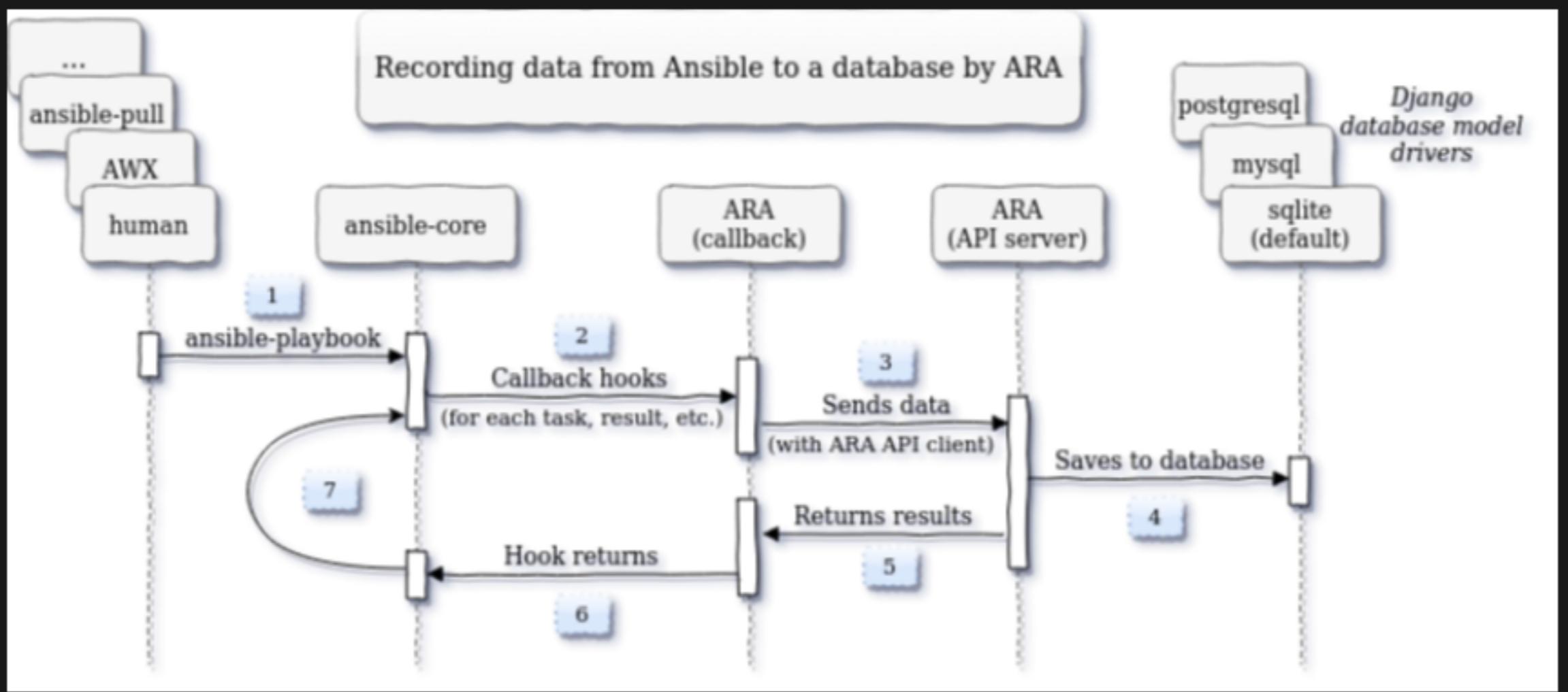


The screenshot shows the Ara web interface, a tool for monitoring Ansible playbooks. The page has a header with the Ara logo, navigation links for Playbooks, Hosts, Tasks, and API, and a documentation/about link. Below the header is a search bar with the placeholder "Search and filter playbooks". Underneath is a table with 100 rows of playbook data, with a "First" button, a page number "1-100 of 3887", and a "Last" button. The table columns are: Report, Status, CLI, Date, Duration, Name (or path), Ansible, Controller, User, Hosts, Tasks, Results, and Labels. The data in the table includes various Ansible playbooks like "site.yml" and "post-deploy.yml" from "share/kolla-ansible/ansible", run by "zuul" user on "primary" controller, with results ranging from 6 to 1569 tasks.

Report	Status	CLI	Date	Duration	Name (or path)	Ansible	Controller	User	Hosts	Tasks	Results	Labels
🔗	🕒	⬇️	09 Sep 2023 23:06:34 +0000	00:06:12.75	...share/kolla-ansible/ansible/site.yml	2.13.11	primary	zuul	4	255	545	check:False tags:all
🔗	🕒	⬇️	09 Sep 2023 23:06:25 +0000	00:00:06.00	...share/kolla-ansible/ansible/post-deploy.yml	2.13.11	primary	zuul	1	6	6	check:False tags:all
🔗	🕒	⬇️	09 Sep 2023 23:04:32 +0000	00:01:51.53	...share/kolla-ansible/ansible/site.yml	2.13.11	primary	zuul	4	67	163	check:False tags:keystone
🔗	🕒	⬇️	09 Sep 2023 22:53:55 +0000	00:04:06.46	...share/kolla-ansible/ansible/site.yml	2.13.11	primary	zuul	2	172	172	check:False tags:all
🔗	🕒	⬇️	09 Sep 2023 22:53:43 +0000	00:00:09.44	...share/kolla-ansible/ansible/post-deploy.yml	2.13.11	primary	zuul	1	6	6	check:False tags:all
🔗	🕒	⬇️	09 Sep 2023 22:51:33 +0000	00:02:07.89	...share/kolla-ansible/ansible/site.yml	2.13.11	primary	zuul	2	69	68	check:False tags:keystone
🔗	🕒	⬇️	09 Sep 2023 22:46:43 +0000	00:17:46.23	...share/kolla-ansible/ansible/site.yml	2.13.11	primary	zuul	4	516	1569	check:False tags:all
🔗	🕒	⬇️	09 Sep 2023 22:46:09 +0000	00:00:30.96	...kolla-ansible/ansible/rabbitmq-reset-state.yml	2.13.11	primary	zuul	3	6	18	check:False tags:all
🔗	🕒	⬇️	09 Sep 2023 22:44:04 +0000	00:02:03.75	...share/kolla-ansible/ansible/site.yml	2.13.11	primary	zuul	3	32	96	check:False tags:rabbitmq
🔗	🕒	⬇️	09 Sep 2023 22:40:48 +0000	01:37:03.12	...playbooks/setup-openstack.yml	2.13.8	alo1.openstack.local	root	16	2334	2339	check:False tags:all



HOW DOES IT WORK ?



```
# Install ansible (or ansible-core) with ara (including API server dependencies)
python3 -m pip install --user ansible "ara[server]"

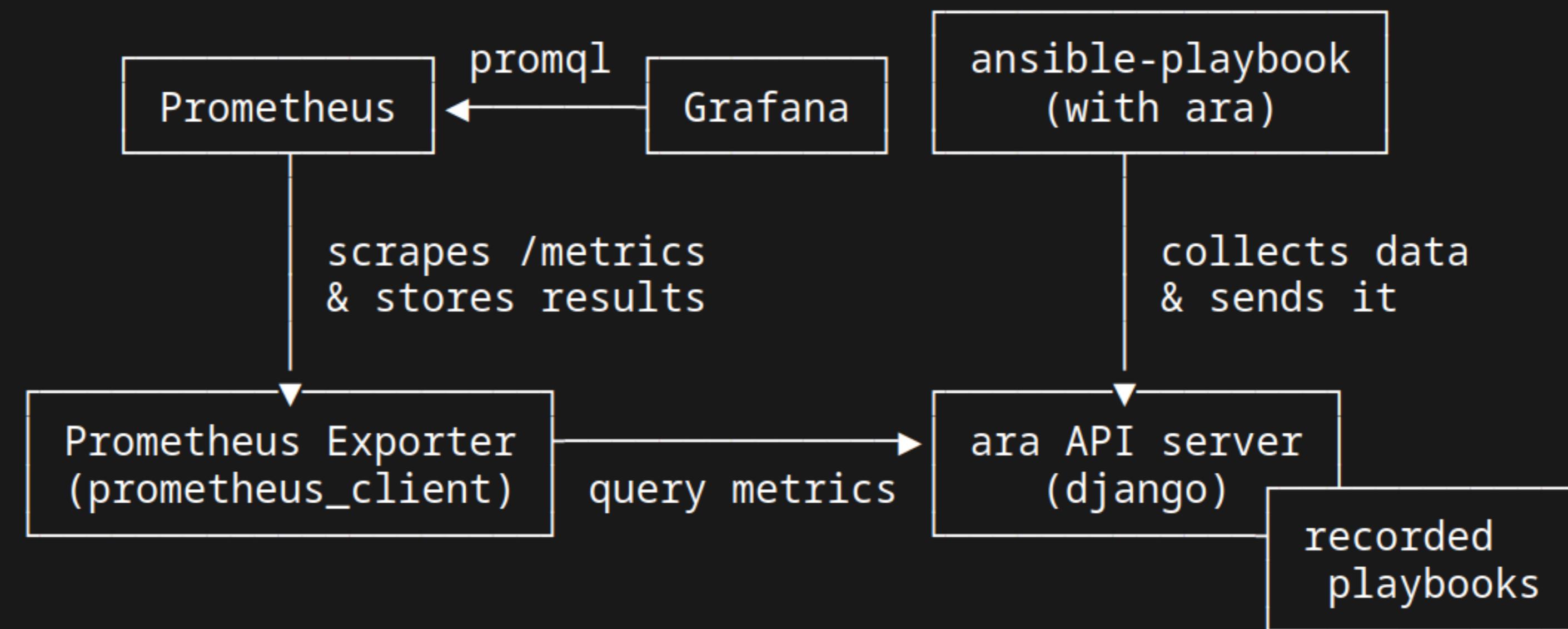
# Configure Ansible to enable ara
export ANSIBLE_CALLBACK_PLUGINS=$(python3 -m ara.setup.callback_plugins)

# Run an Ansible playbook as usual
ansible-playbook playbook.yml

# Check out the CLI
ara playbook list
ara host list

# or the UI at http://127.0.0.1:8000
ara-manage runserver
```

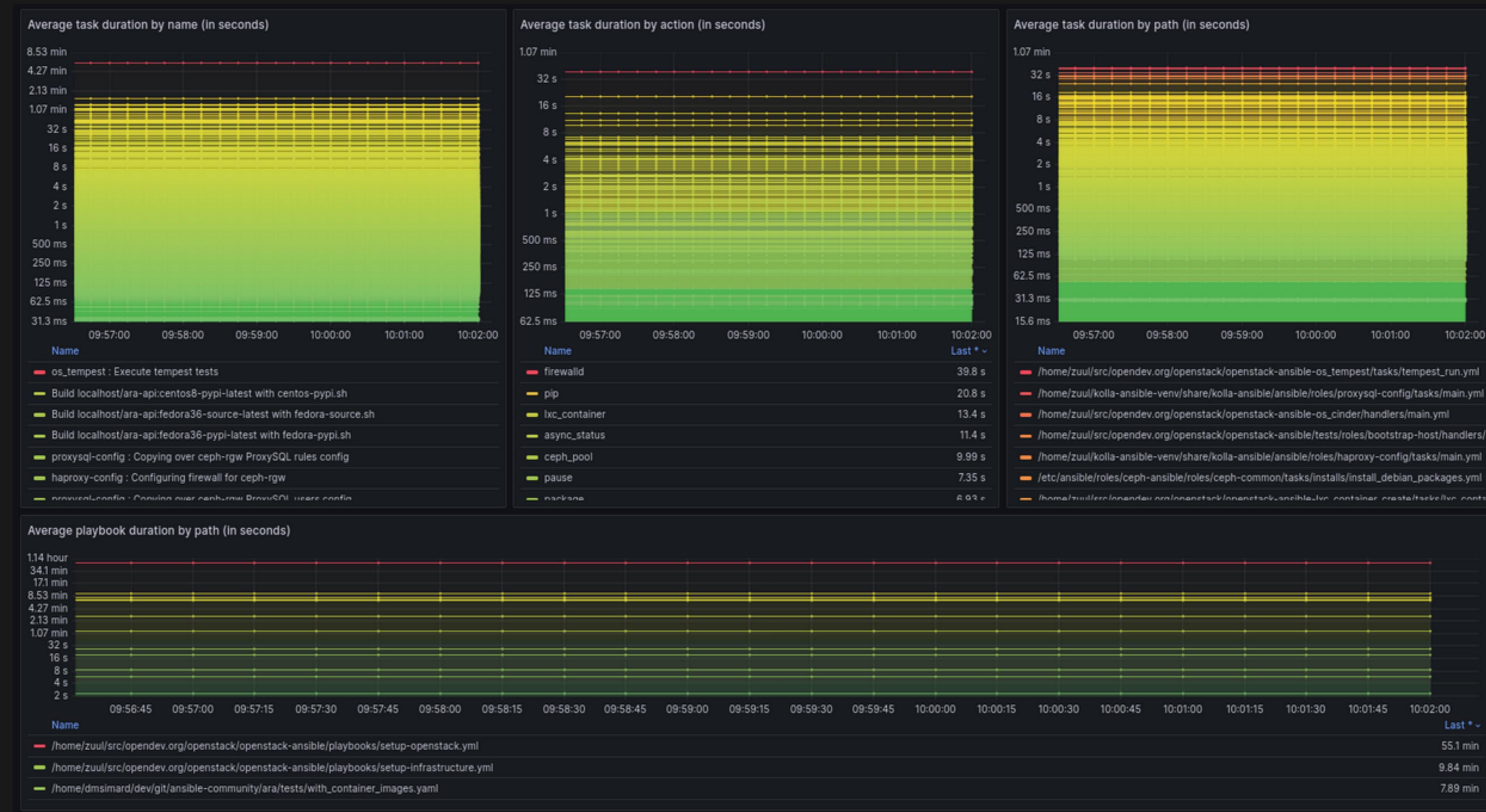
FOR ANSIBLE METRICS ?



DEMO !

<https://github.com/ansible-community/ara/pull/483>

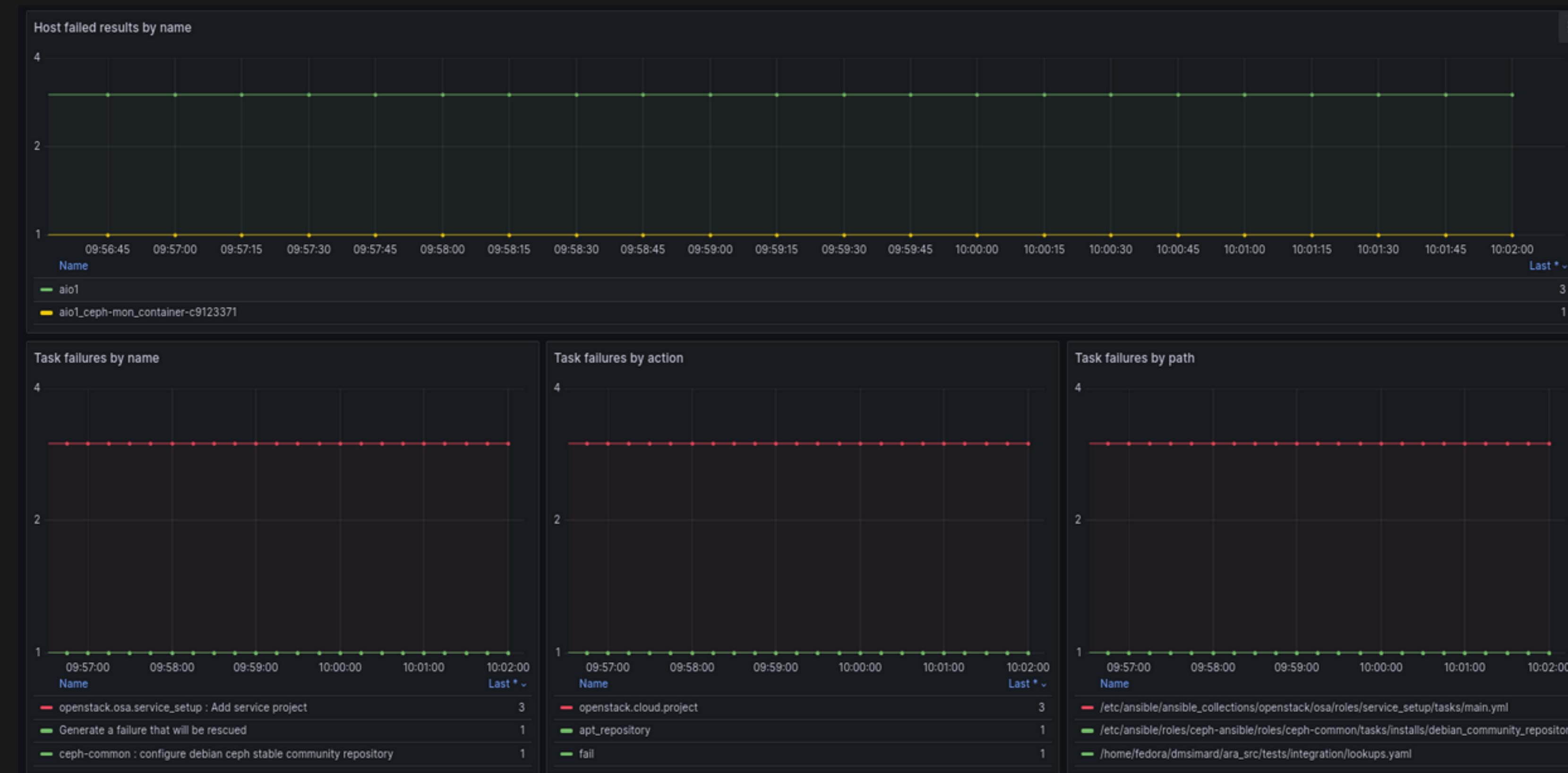
PERFORMANCE METRICS



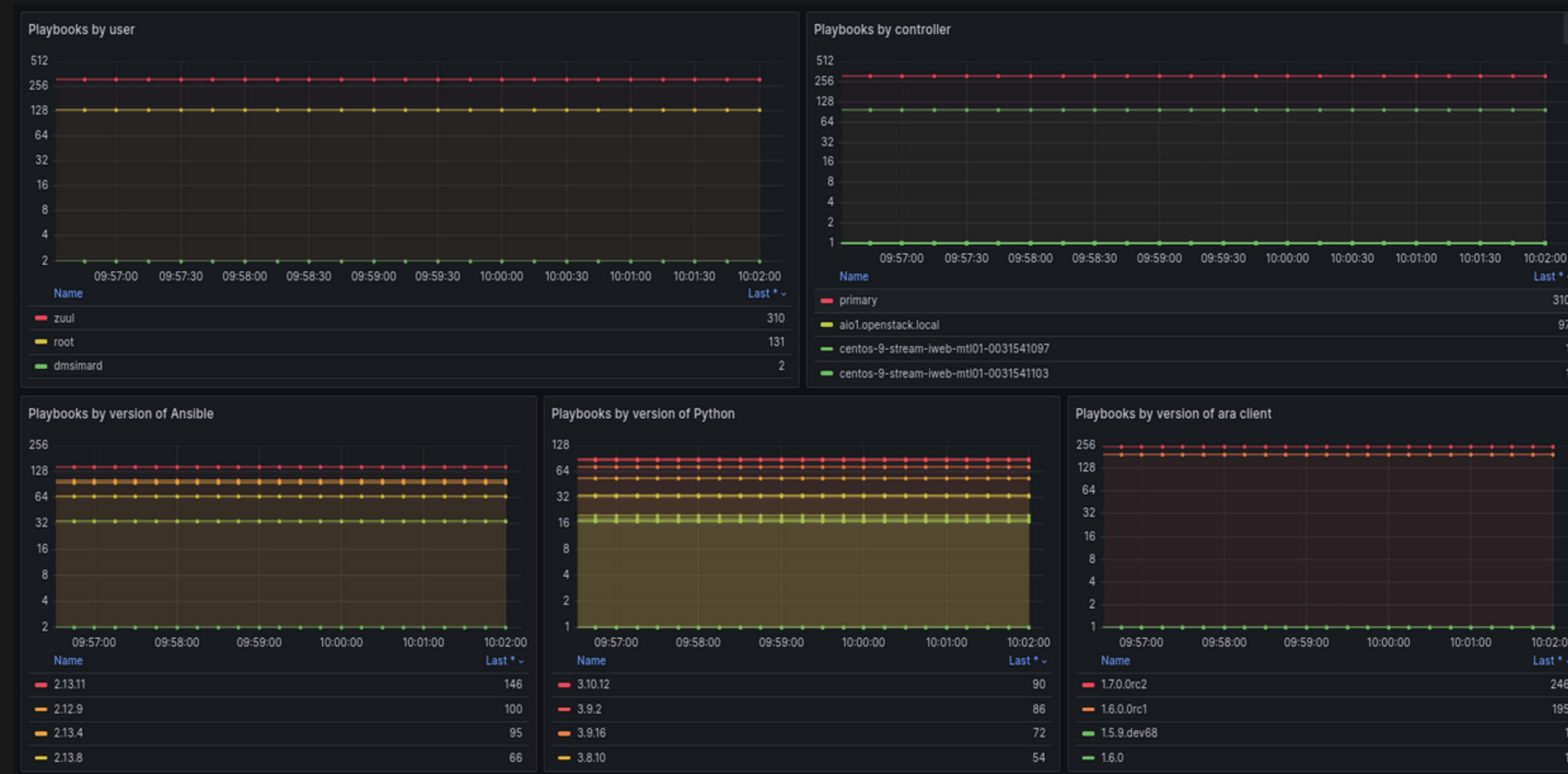
STATUS METRICS



FAILURE METRICS

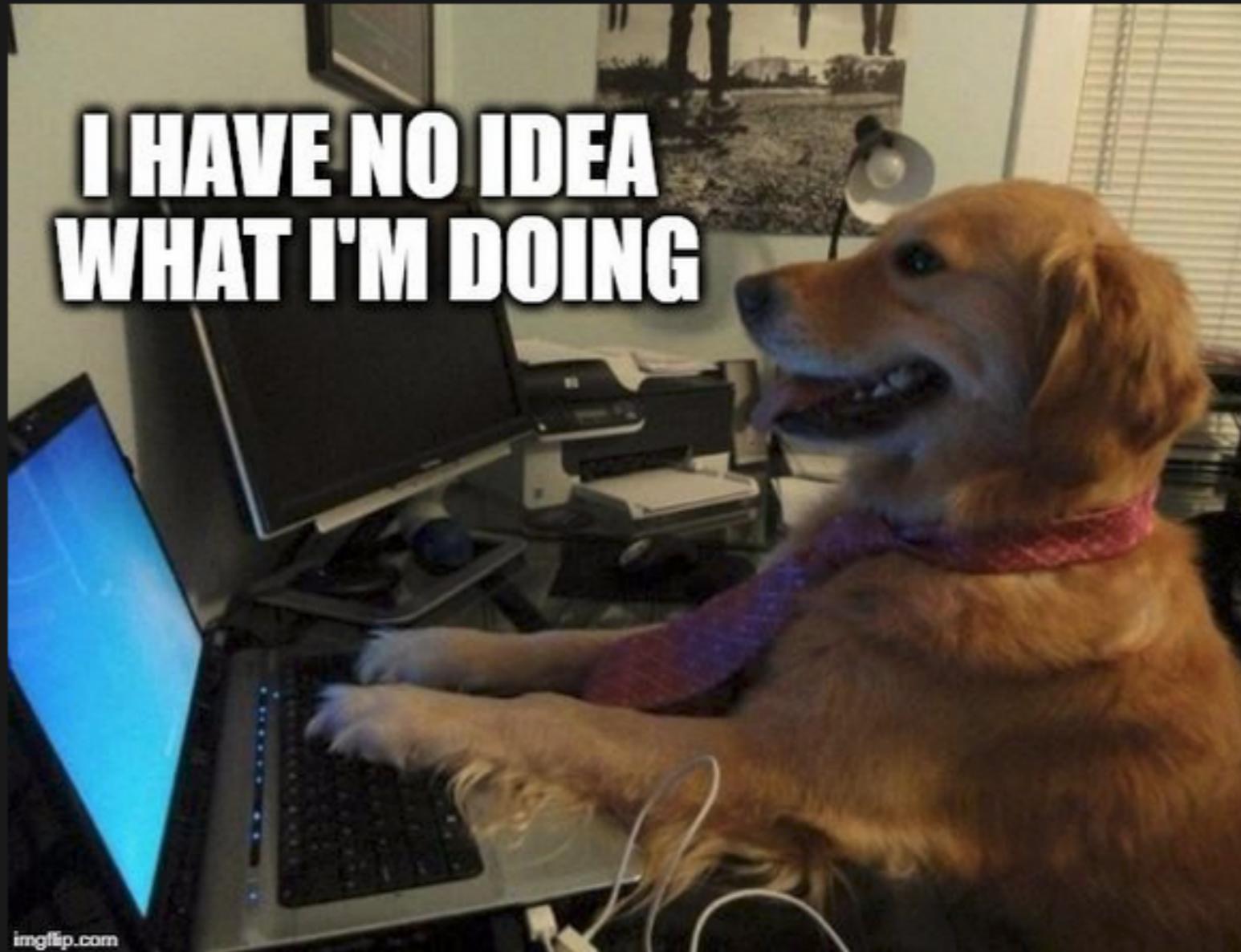


OTHER METRICS



Kind of works, but...

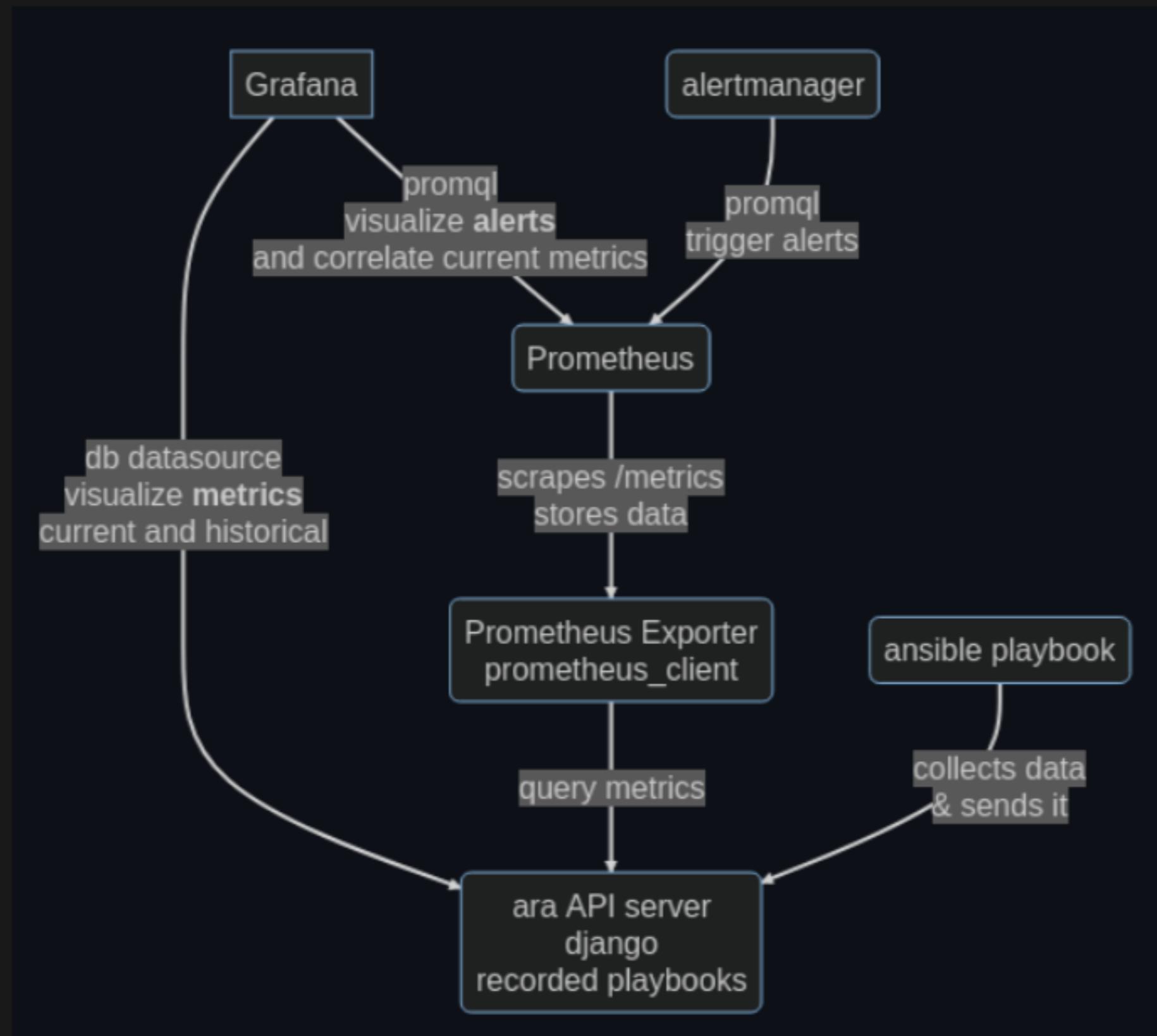
It's not so simple.



- Erroneously duplicated metrics
- Probably not the right metric formats
- Probably not the right labels (cardinality?)

Can we make this work ?

WITH GRAFANA ?



<https://github.com/ansible-community/ara/pull/483#issuecomment-1824480464>

QUERYING MYSQL WITH GRAFANA

Playbook status

status	COUNT(status)
completed	3225
expired	157
failed	266
running	2

Query 1 Transform 0

Data source ara-mysql MD = auto = 440 Interval = Query inspector

A (ara-mysql) Run query Builder Code

Format: Table

```
1 SELECT status, COUNT(status) FROM ara.playbooks GROUP BY status
```

The screenshot shows the Grafana MySQL Query editor interface. At the top, there's a table titled "Playbook status" with four rows: completed (3225), expired (157), failed (266), and running (2). Below the table are tabs for "Query" (which is selected, showing value 1) and "Transform" (value 0). Underneath the tabs is a "Data source" dropdown set to "ara-mysql". To the right of the dropdown are "MD" and "Interval" settings. A "Query inspector" button is also present. Below these settings is a section labeled "A (ara-mysql)" containing a "Run query" button, a "Builder" tab, and a "Code" tab. The "Code" tab is active, displaying the SQL query: "SELECT status, COUNT(status) FROM ara.playbooks GROUP BY status". The entire interface has a dark theme.

PLENTY OF TABULAR DATA

Playbook status	
status	COUNT(status)
completed	3225
expired	157
failed	266
running	2

Ansible versions	
ansible_version	COUNT(ansible_version)
2.10.1	50
2.10.10	62
2.10.12	89
2.10.2	7
2.10.3	61
2.10.4	64

Controller	
controller	COUNT(controller)
0ee3152472c4	68
3386fe471fd4	97
4be96a5e15e9	130
4d0a1c32cabd	3
50f7563353ec	184
7d1a8b301730	77

Task status	
status	COUNT(status)
completed	557034
expired	153
failed	523
running	2

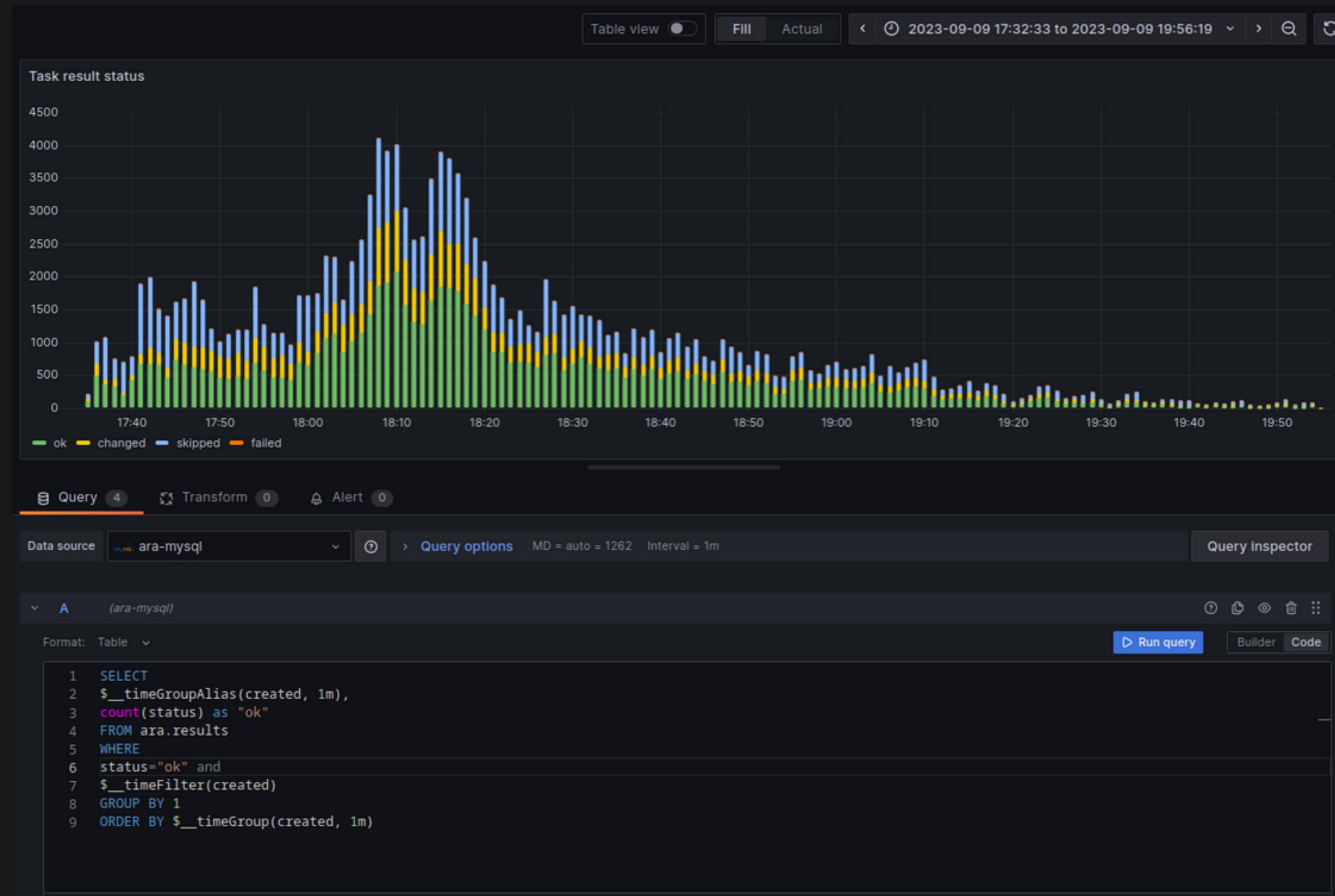
Task action	
action	COUNT(action)
a	4
add_host	686
alias_ping	2
alternatives	161
ansible.builtin.apt	24
ansible.builtin.assert	251

Hosts	
name	COUNT(name)
42	12
A	38
aio1	373
aio1_ceph-mds_container-...	2

Result status	
status	COUNT(status)
failed	2124
ok	573699
skipped	252418
unreachable	30



WHAT ABOUT TIME SERIES ?



<https://grafana.com/blog/2023/07/07/how-to-visualize-time-series-from-sql-databases-with-grafana/>



- It works (!)
- Doesn't require prometheus
- Grafana supports alerting (if you're into that)
- Converting to time series is not trivial
- Requires knowledge of database schema
- Not compatible between database backends

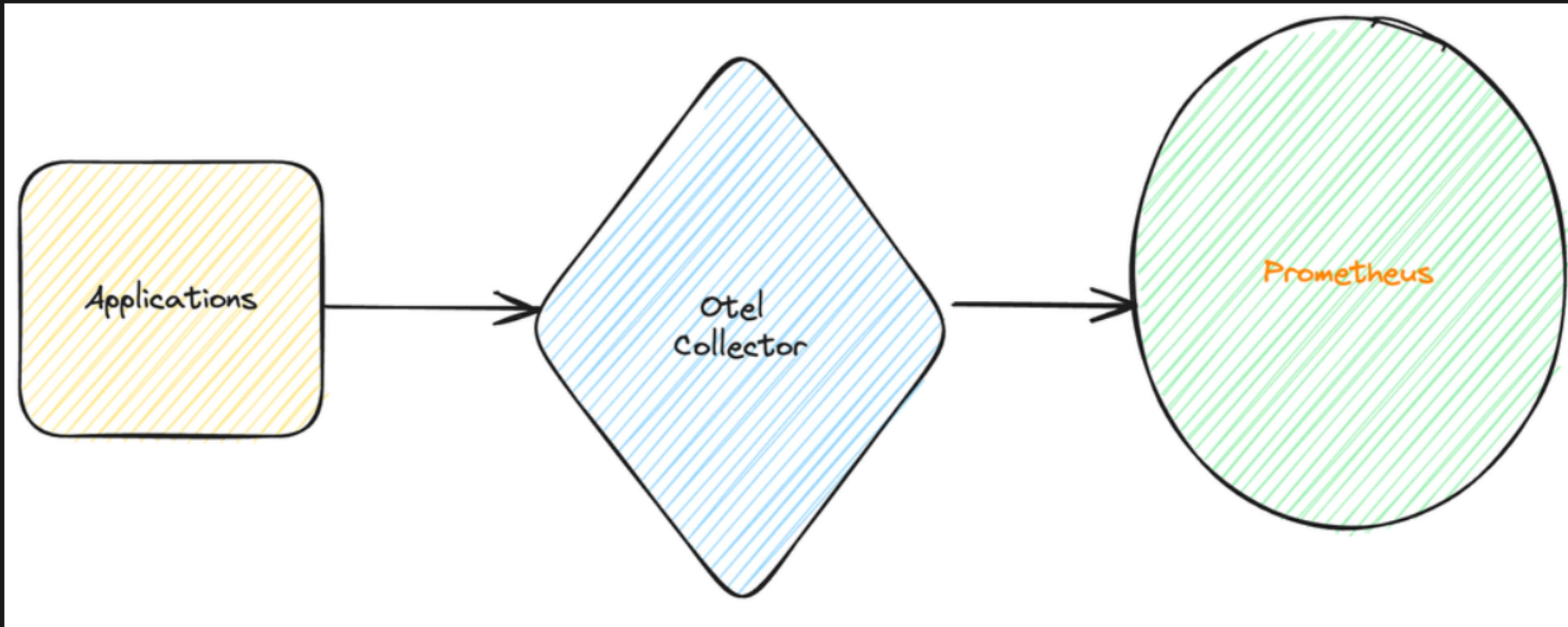
SOME ISSUES...

- First iteration of a prometheus exporter for ara
- Any label which looks like a timestamp is likely not a good choice

ara_hosts_ok{name="localhost",playbook="30",updated="2023-06-0

- When the updated value changes it will create a new series which means increased cardinality
- Pushing an anti-pattern for a prometheus lib

OTEL COLLECTOR



- Prometheus is integrating OpenTelemetry natively, starting with the next release
- New feature flag: `otlp-write-receiver`
- OpenTelemetry metrics can be sent on `/otlp/v1/metrics` the endpoint and ingested natively
- OTEL Python SDK

```
1 +## Copying from opentelemetry/opentelemetry-collector-contrib
2 +
3 + This files in the `prometheus/` and `prometheuseremotewrite/` are copied from the OpenTelemetry
4 + Project[^1].
5 + This is done instead of adding a go.mod dependency because OpenTelemetry depends on
   `prometheus/prometheus` and a cyclic dependency will be created. This is just a temporary
   solution and the long-term solution is to move the required packages from OpenTelemetry into
   `prometheus/prometheus`.
```

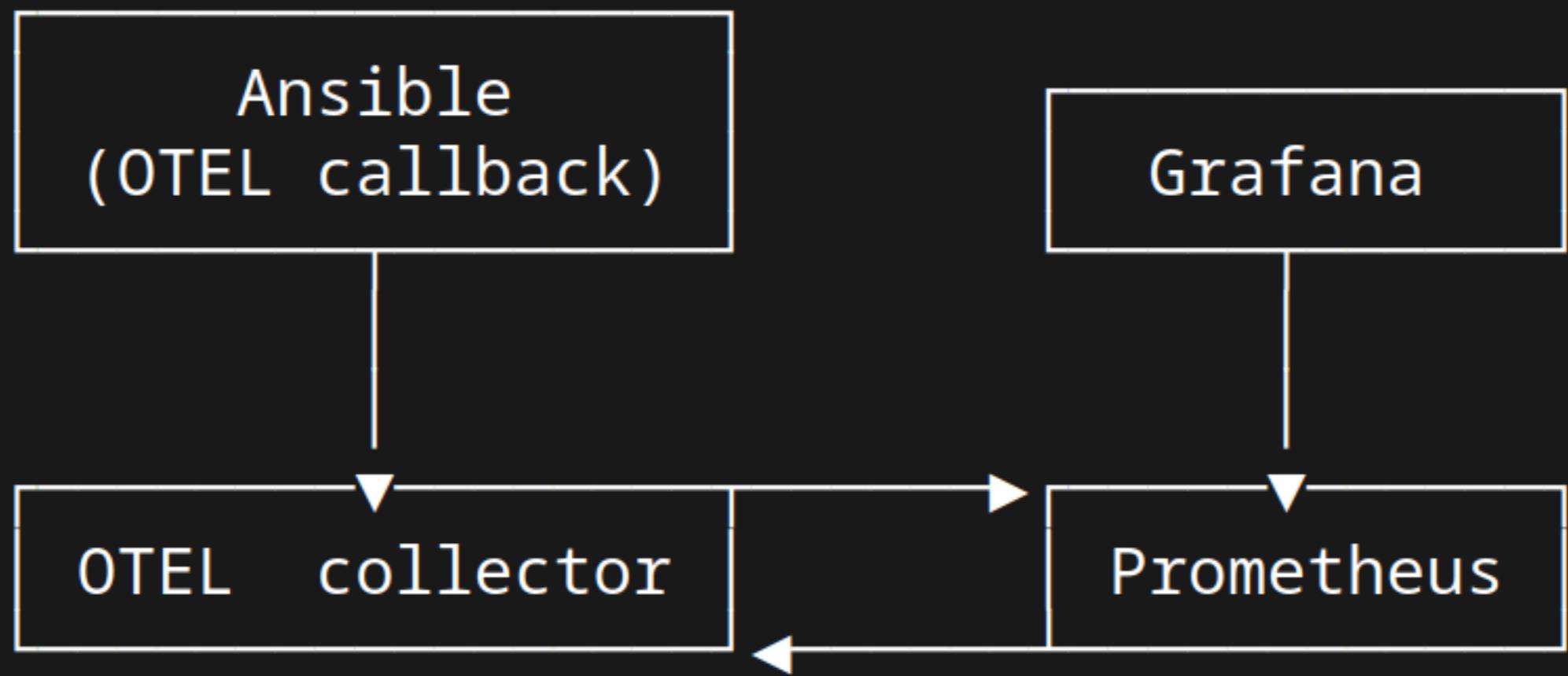
CALLBACK PLUGINS (BESIDES ARA)

- community.general.opentelemetry callback plugin
- part of the community.general collection (version 8.3.0).

```
examples: |
  Enable the plugin in ansible.cfg:
  [defaults]
  callbacks_enabled = community.general.opentelemetry
  [callback_opentelemetry]
  enable_from_environment = ANSIBLE_OPENTELEMETRY_ENABLED

  Set the environment variable:
  export OTEL_EXPORTER_OTLP_ENDPOINT=<your endpoint (OTLP/HTTP)>
  export OTEL_EXPORTER_OTLP_HEADERS="authorization=Bearer your_otel_token"
  export OTEL_SERVICE_NAME=your_service_name
  export ANSIBLE_OPENTELEMETRY_ENABLED=true
```

COMMUNITY.GENERAL.OPENTELEMETRY



https://docs.ansible.com/ansible/latest/collections/community/general/opentelemetry_callback.html

<https://github.com/ansible-collections/community.general/blob/main/plugins/callback/opentelemetry.py>

<https://opentelemetry.io/docs/collector/>

TRACING WITH OPENTELEMETRY

```
#...
Span #5
  Trace ID      : 634741b983a374f1661d92a93ff48921
  Parent ID     : b7cd0871ef
  ID            : 3004acb296337744
  Name          : Ping
  Kind          : Internal
  Start time    : 2023-10-21 18:00:49.809834256 +0000 UTC
  End time      : 2023-10-21 18:00:50.067565576 +0000 UTC
  Status code   : Ok
  Status message:
Attributes:
  -> ansible.task.module: Str(ping)
  -> ansible.task.message: Str(success)
  -> ansible.task.name: Str([host1] ARA Hosts test play: Ping)
  -> ansible.task.result: Int(0)
  -> ansible.task.host.name: Str(host1)
  -> ansible.task.host.status: Str(ok)
  -> ansible.task.args.name: Slice([])
  -> ansible.task.args.value: Slice([])
Events:
SpanEvent #0
  -> Name: {"changed": false, "ping": "pong"}  
# ...
```



<https://www.youtube.com/watch?v=5qKkTwYg15I>



TAKEAWAYS

- Opportunities to improve the proof of concept
- There's still work to do (you can help!)
- Grafana works by itself without prometheus
- Need to determine whether OTEL or Prometheus remote write are viable options
- Consider whether ara is adequate as a source of metrics

THANK YOU !

QUESTIONS ?



For more information:

- <https://github.com/ansible-community/ara>
- <https://ara.recordsansible.org/blog>
- <https://fosstodon.org/@ara>
- Ansible Contributor Summit @ Wednesday, B.1.017

BACKUP SLIDES

PROMETHEUS

Configuration in `prometheus.yml`:

```
global:  
  scrape_interval: 15s  
scrape_configs:  
  - job_name: node  
    static_configs:  
      - targets: ['localhost:9100']
```

Start a Prometheus container:

```
{podman,docker} run -d --name prometheus \  
-p 9090:9090 \  
-v prometheus.yml:/etc/prometheus/prometheus.yml \  
quay.io/prom/prometheus
```

NODE EXPORTER

Start a node exporter container:

```
{podman,docker} run -d --name node-exporter \
-p 9100:9100 \
--pid="host" \
-v "/:/host:ro,rslave" \
quay.io/prometheus/node-exporter:latest \
--path.rootfs=/host
```

Validate that metrics are available:

```
> curl -s 127.0.0.1:9100/metrics
# HELP go_gc_duration_seconds A summary of the pause duration
# TYPE go_gc_duration_seconds summary
go_gc_duration_seconds{quantile="0"} 2.12e-05
go_gc_duration_seconds{quantile="0.25"} 3.3452e-05
go_gc_duration_seconds{quantile="0.5"} 3.7511e-05
go_gc_duration_seconds{quantile="0.75"} 5.0224e-05
go_gc_duration_seconds{quantile="1"} 8.1453e-05
# ...
```

GRAFANA

Start a Grafana container:

```
{podman, docker} run -d --name=grafana \
-p 3000:3000 \
docker.io/grafana/grafana
```

Add a Prometheus data source:

The screenshot shows the Grafana interface with the following details:

- Header:** Search bar with placeholder "Search or jump to..." and keyboard shortcut "ctrl+k".
- Breadcrumbs:** Home > Connections > Data sources > Prometheus
- Sidebar:** Connections (disabled), Add new connection, Data sources (selected).
- Panel:** Prometheus data source configuration.
 - Title:** Prometheus
 - Type:** Prometheus
 - Buttons:** Explore data, Settings (selected), Dashboards.
 - Status:** Alerting supported
 - Fields:**
 - Name: Prometheus
 - Default: On
 - HTTP Section:**
 - Prometheus server URL: http://127.0.0.1:9090
 - Allowed cookies: New tag (enter key to add) Add
 - Timeout: Timeout in seconds

DEMO: TRYING OUT THE EXPORTER

```
# Install and run a prometheus exporter with metrics from ara
# https://github.com/ansible-community/ara/issues/177
# https://github.com/ansible-community/ara/pull/483
git clone https://github.com/dmsimard/ara
cd ara
git checkout prometheus_exporter

# Set up a virtualenv with ansible, ara and prometheus-client
tox -e ansible-integration --notest
source .tox/ansible-integration/bin/activate
pip install prometheus-client

# Metrics from localhost without needing to run a server
ara prometheus --max-days 1

# Metrics from a remote server running somewhere
ara prometheus --client http --server http://127.0.0.1:8000 --
```

