rdatadev dec. 2021

GRAFANA
Loki / Promtail

Use case in SPHERE & SSHADE

Damien ALBERT – OSUG 2021-12

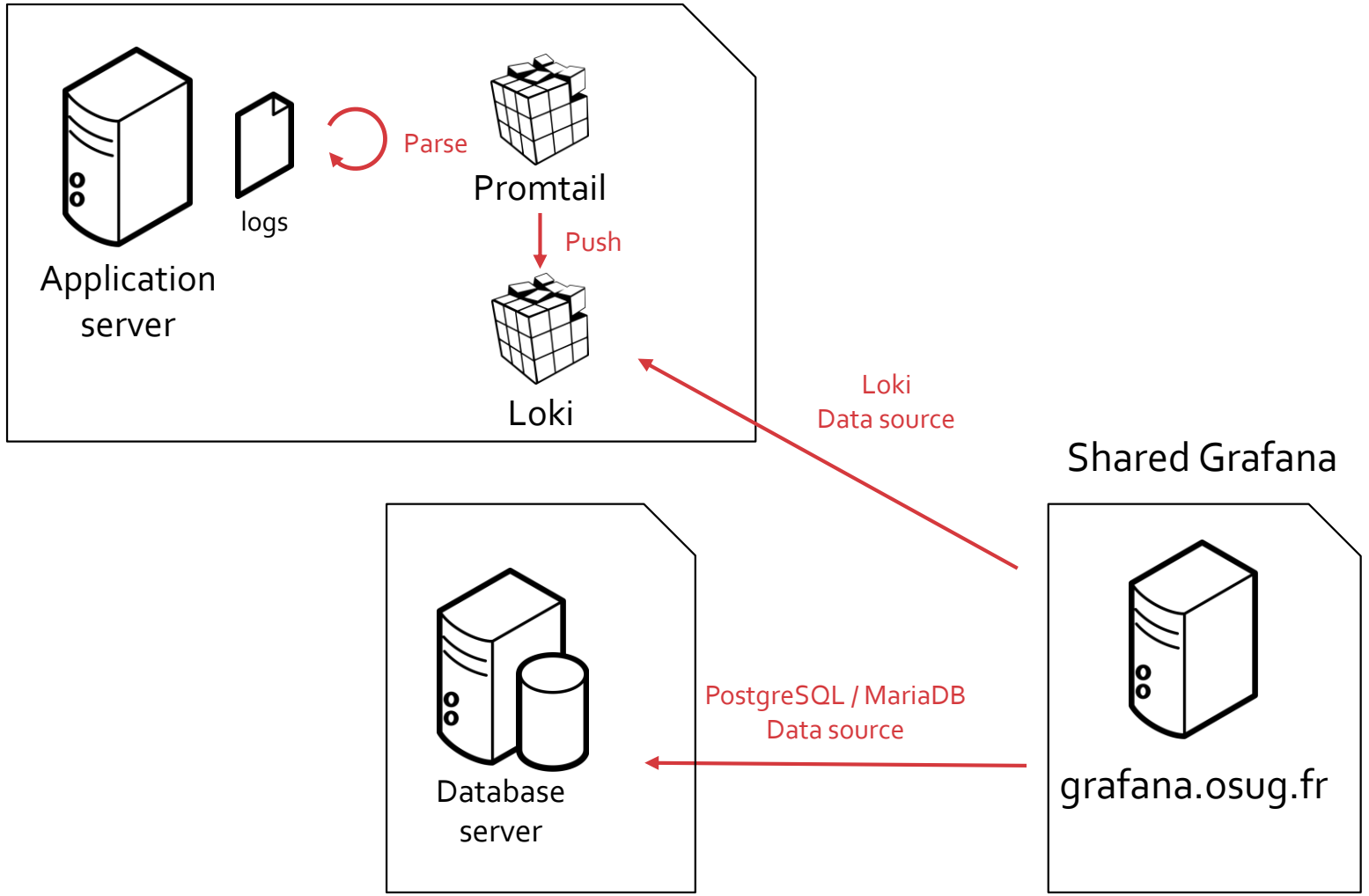
Loki

- https://grafana.com/oss/loki/
- "Loki is a horizontally-scalable, highly-available, multi-tenant log aggregation system inspired by Prometheus. It is designed to be very cost effective and easy to operate. It does not index the contents of the logs, but rather a set of labels for each log stream."

Promtail

- https://grafana.com/docs/loki/latest/clients/promtail/
- "Promtail is an agent which ships the contents of local logs to a private Loki instance or Grafana Cloud. It is usually deployed to every machine that has applications needed to be monitored."

Overview


Application server
logs
Parse
Promtail
Push
Loki
Loki
Data source
Shared Grafana
Database server
PostgreSQL / MariaDB
Data source
grafana.osug.fr

Installation

- https://grafana.com/docs/loki/latest/installation/docker/
- Install Loki and Promtail on local machine
- Customized to deploy services configuration from Ansible

```
services:
  loki:
    image: grafana/loki:latest
    restart: unless-stopped
    ports:
      - "3100:3100"
    volumes:
      - /srv/docker/loki/config/loki:/etc/loki
    command: -config.file=/etc/loki/local-config.yaml
    networks:
      - loki


  promtail:
    image: grafana/promtail:latest
    restart: unless-stopped
    volumes:
      - /var/log:/var/log
      - /srv/docker/loki/config/promtail:/etc/promtail
    command: -config.file=/etc/promtail/config.yml
    networks:
      - loki
```

Loki retention policy

- https://grafana.com/docs/loki/latest/operations/storage/retention/
- 28 days retention

```
chunk_store_config:
  max_look_back_period: 672h

table_manager:
  retention_deletes_enabled: true
  retention_period: 672h
```

- Example of space usage on sphere-dc :

```
spherdwh@sphere-dc ~
$ docker ps --size
CONTAINER ID        IMAGE                          PORTS                    NAMES               SIZE
a34f856829d0        grafana/promtail:latest                                 loki_promtail_1     546B (virtual 168MB)
26ffb818fc24        grafana/loki:latest            0.0.0.0:3100->3100/tcp   loki_loki_1         853MB (virtual 915MB)
```

Promtail - targets

```yaml
- job_name: tomcat
  static_configs:

  # Tomcat logs
  - targets:
      - localhost
    labels:
      job: tomcat
      __path__: /var/log/tomcat8/catalina.out

  # DC application logs
  - targets:
      - localhost
    labels:
      job: application
      __path__: /var/log/tomcat8/{sphere-server,sphere-server-dev,cobrex-server,cobrex-server-dev}.log

  # DB logs
  - targets:
      - localhost
    labels:
      job: db
      __path__: /var/log/tomcat8/{sphere-server,sphere-server-dev,cobrex-server,cobrex-server-dev}-db.log
```

```yaml
- job_name: sshade
  static_configs:

  # UWSGI logs
  - targets:
      - localhost
    labels:
      job: application
      __path__: /var/log/uwsgi/{www,dev,sandbox}.sshade.eu/error.log
  - targets:
      - localhost
    labels:
      job: access
      __path__: /var/log/uwsgi/{www,dev,sandbox}.sshade.eu/access.log
```

Promtail - parsing

```
pipeline_stages:
- match:
    selector: '{job="db"}'
    stages:
    - regex:
        expression: '^(?P<time>\S+ \S+) (?P<level>\S+) \[(?P<logger>\S+)\] - \(?\s*(?P<sql_type>[A-Z][A-Z]+) (?P<log>.*)$'
    - labels:
        sql_type:
- match:
    selector: '{job="tomcat"}'
    stages:
    - multiline:
        firstline: '^\S+[^a][^t]\S'
        max_wait_time: 3s
    - replace:
        expression: '^(.*AssociationRuleOptimization)(?:Error|Exception)[:\n]'
        replace: '{{.Value}}ExcWarning (WARNING) '
    - replace:
        expression: '^(.*ProcessLauncher)(?:Error|Exception): Process output contains an error.'
        replace: '{{.Value}}ExcWarning (WARNING) '
    - replace:
        expression: '^(\s*(?:Internal Exception: |Caused by: |Exception in thread "[^"]+" )?[A-z._\d$]*(?:Error|Exception)[A-z._\d]*)[:\n]'
        replace: '{{.Value}} (ERROR)'
    - regex:
        expression: '^(?P<time>\S+ \S+) (?P<level>\S+) - (?P<log>.*)$'
    - regex:
        expression: '^(?P<time>\S+) \[(?P<thread>\S+)\] (?P<level>\S+) (?P<logger>\S+)? ?- (?P<log>.*)$'
    - regex:
        expression: '^(?P<time>\S+ \S+) (?P<level>\S+) \[(?P<logger>\S+)\] (?P<log>.*)$'
    - regex:
        expression: '^(?P<exception>\s*(?:Internal Exception: )?[A-z._\d$]*(?:Error|Exception|ExcWarning)[A-z._\d]*) \((?P<level>\S+)\)[:\n ]'
    - labels:
        level:
- match:
    selector: '{job="application"}'
    stages:
    - multiline:
        firstline: '^\S+[^a][^t]\S'
        max_wait_time: 3s
    - replace:
        expression: '^(\s*(?:Internal Exception: |Caused by: )?[A-z._\d$]*(?:Error|Exception)[A-z._\d]*)[:\n]'
        replace: '{{.Value}} (ERROR)'
    - regex:
        expression: '^(?P<time>\S+ \S+) (?P<level>\S+) \[(?P<logger>\S+)\] - (?P<log>.*)$'
    - regex:
        expression: '^(?P<exception>\s*(?:Internal Exception: |Caused by: )?[A-z._\d$]*(?:Error|Exception)[A-z._\d]*) \((?P<level>\S+)\)[:\n]'
    - labels:
        level:
        logger:
```

Promtail - parsing

```
  - match:
      selector: '{job="access"}'
      stages:
      - replace:
          expression: '\((HTTP\S+ 200)\)'
          replace: '{{.Value}} INFO'
      - replace:
          expression: '\((HTTP\S+ (404|403|500))\)'
          replace: '{{.Value}} ERROR'
      - replace:
          expression: '\((HTTP\S+ (3\d\d))\)'
          replace: '{{.Value}} TRACE'
      - regex:
          expression: '^\[pid: (?P<pid>\S+)\|app: (?P<app>\d+)\|req: (?P<req_nb>\d+)\/(?P<req_total>\d+)\] (?P<ip>\S+) \(\) \{.*\}
 \[(?P<time>[^\]]+)\] (?P<method>\S+) (?P<url>\S+) => generated (?P<size>\S+) bytes in (?P<duration>\S+) .* \((?P<http_version>HTT
P\S+) (?P<code>\S+) ?(?P<level>\S+)?\) (?P<info>.*)$'
      - pack:
          labels: [size, duration]
      - labels:
          level:
          status:
          method:
          code:
```

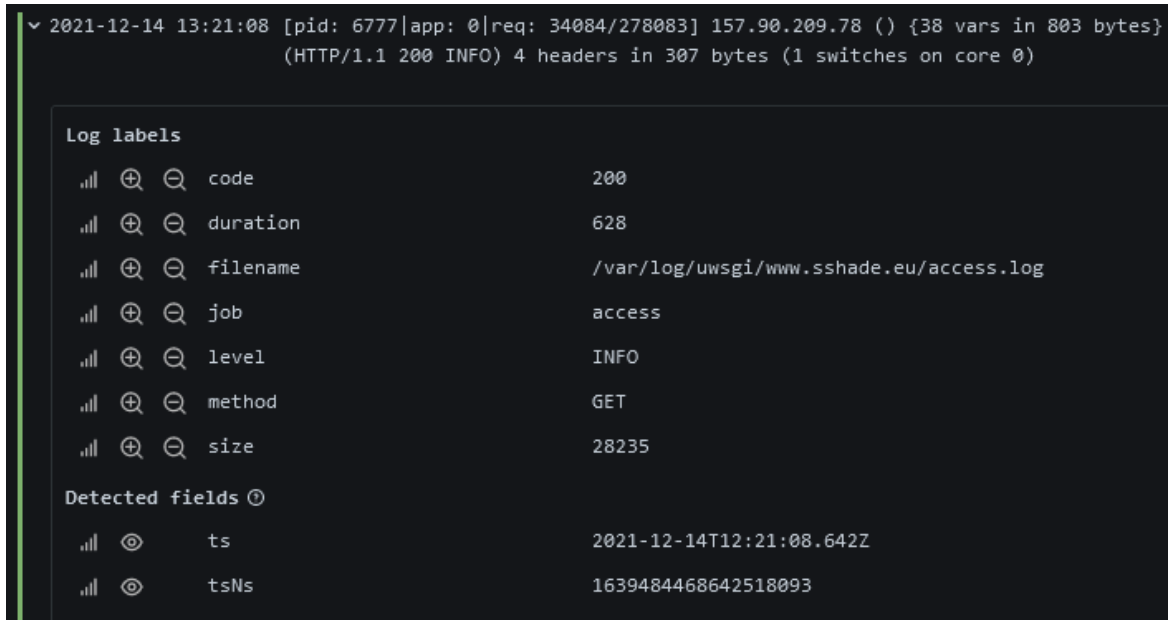
Promtail - parsing

- Use named capturing groups `(?P<name>…)`

```
- regex:
    expression: '^(?P<time>\S+ \S+) (?P<level>\S+) - (?P<log>.*)$'
- regex:
    expression: '^(?P<time>\S+) \[(?P<thread>\S+)\] (?P<level>\S+) (?P<logger>\S+)? ?- (?P<log>.*)$'
- regex:
    expression: '^(?P<time>\S+ \S+) (?P<level>\S+) \[(?P<logger>\S+)\] (?P<log>.*)$'
```

- Pack captured groups as JSON with `pack`

```
      - regex:
          expression: '^\[pid: (?P<pid>\S+)\|app: (?P<app>\d+)\|req: (?P<req_nb>\d+)\/(?P<req_total>\d+)\] (?P<ip>\S+) \(\) \{.*\}
 \[(?P<time>[^\]]+)\] (?P<method>\S+) (?P<url>\S+) => generated (?P<size>\S+) bytes in (?P<duration>\S+) .* \((?P<http_version>HTT
P\S+) (?P<code>\S+) ?(?P<level>\S+)?\) (?P<info>.*)$'
      - pack:
          labels: [size, duration]
```



Promtail - parsing

- Use `replace` to inject a customized log level

```
- replace:
    expression: '^(.*AssociationRuleOptimization)(?:Error|Exception)[:\n]'
    replace: '{{.Value}}ExcWarning (WARNING) '
- replace:
    expression: '^(.*ProcessLauncher)(?:Error|Exception): Process output contains an error.'
    replace: '{{.Value}}ExcWarning (WARNING) '
- replace:
    expression: '^(\s*(?:Internal Exception: |Caused by: |Exception in thread "[^"]+" )?[A-z._\d$]*(?:Error|Exception)[A-z._\d]*)[:\n]'
    replace: '{{.Value}} (ERROR)'
- regex:
    expression: '^(?P<time>\S+ \S+) (?P<level>\S+) - (?P<log>.*)$'
```

```
      - replace:
          expression: '\((HTTP\S+ 200)\)'
          replace: '{{.Value}} INFO'
      - replace:
          expression: '\((HTTP\S+ (404|403|500))\)'
          replace: '{{.Value}} ERROR'
      - replace:
          expression: '\((HTTP\S+ (3\d\d))\)'
          replace: '{{.Value}} TRACE'
      - regex:
          expression: '^\[pid: (?P<pid>\S+)\|app: (?P<app>\d+)\|req: (?P<req_nb>\d+)\/(?P<req_total>\d+)\] (?P<ip>\S+) \(\) \{.*\}
 \[(?P<time>[^\]]+)\] (?P<method>\S+) (?P<url>\S+) => generated (?P<size>\S+) bytes in (?P<duration>\S+) .* \((?P<http_version>HTT
P\S+) (?P<code>\S+) ?(?P<level>\S+)?\) (?P<info>.*)$'
```

- Use `multiline` to group stack traces lines

```
selector: '{job="tomcat"}'
stages:
- multiline:
    firstline: '^\S+[^a][^t]\S'
    max_wait_time: 3s
```

```
selector: '{job="application"}'
stages:
- multiline:
    firstline: '^(Traceback|\S+)'
    max_wait_time: 3s
```

```
› 2021-12-14 06:49:08 Traceback (ERROR) (most recent call last):
                        File "/srv/sshade/www.sshade.eu/sshade/.venv/lib/python3.7/site-packages/pyramid/router.py", line 270, in __call__
                          response = self.execution_policy(environ, self)
                        File "/srv/sshade/www.sshade.eu/sshade/.venv/lib/python3.7/site-packages/pyramid/router.py", line 276, in default_execution_policy
                          return router.invoke_request(request)
                        File "/srv/sshade/www.sshade.eu/sshade/.venv/lib/python3.7/site-packages/pyramid/router.py", line 245, in invoke_request
                          response = handle_request(request)
                        File "/srv/sshade/www.sshade.eu/sshade/.venv/lib/python3.7/site-packages/pyramid/tweens.py", line 43, in excview_tween
                          response = _error_handler(request, exc)
                        File "/srv/sshade/www.sshade.eu/sshade/.venv/lib/python3.7/site-packages/pyramid/tweens.py", line 17, in _error_handler
```

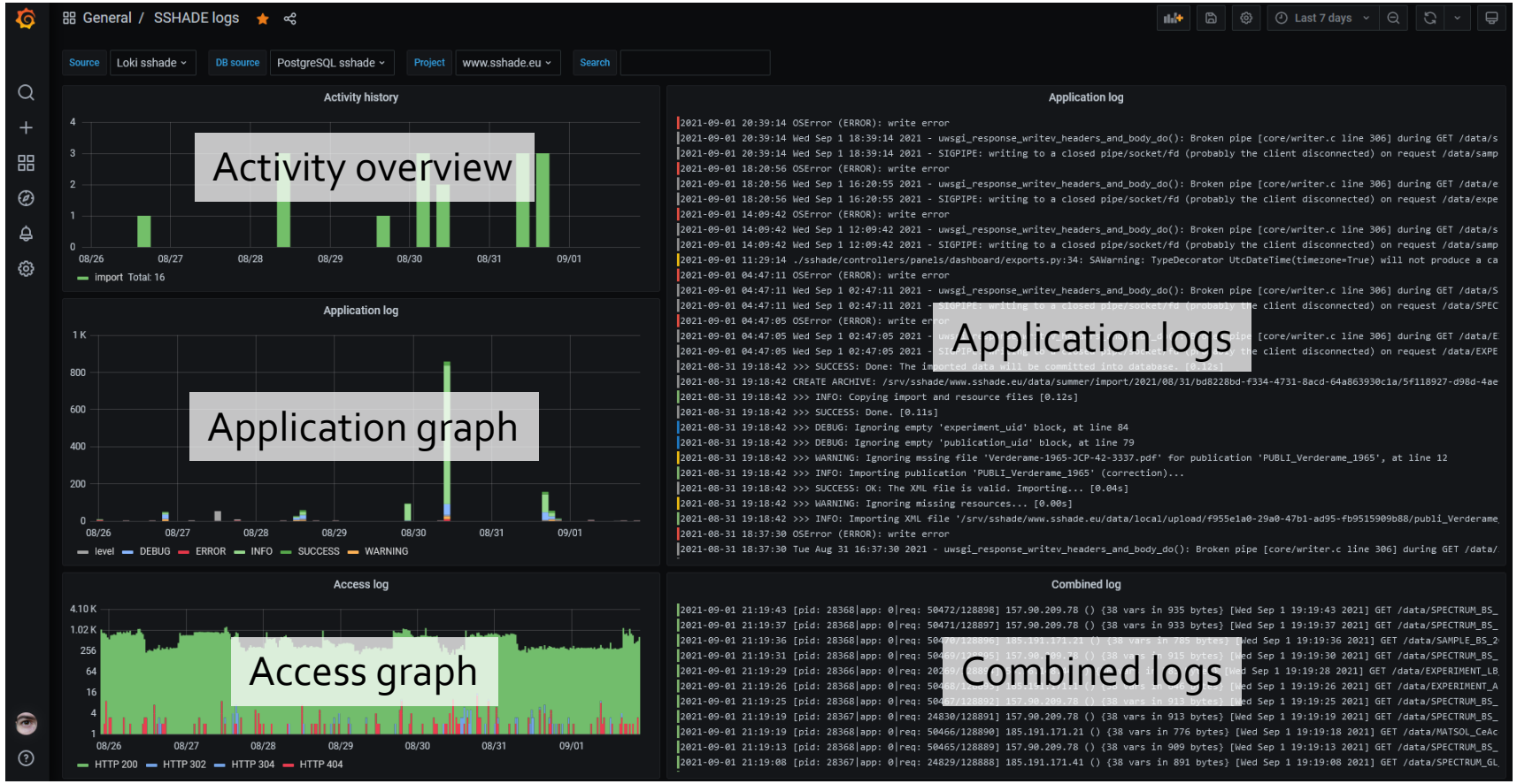
Promtail - parsing

Notes

- Grafana uses Go lang regular expressions, no negative/positive lookahead/lookbehind

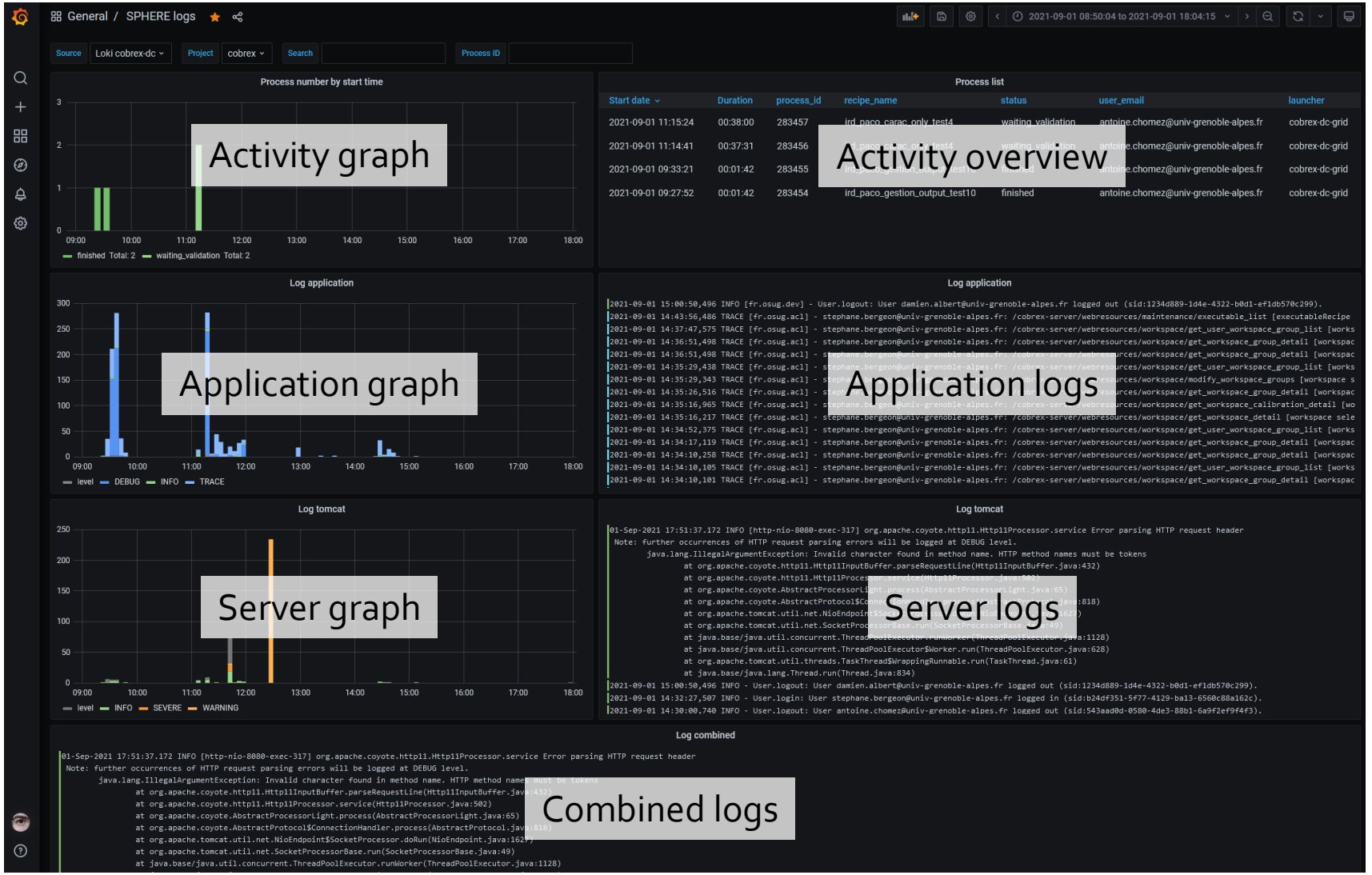
Grafana logs dashboard

SSHADE



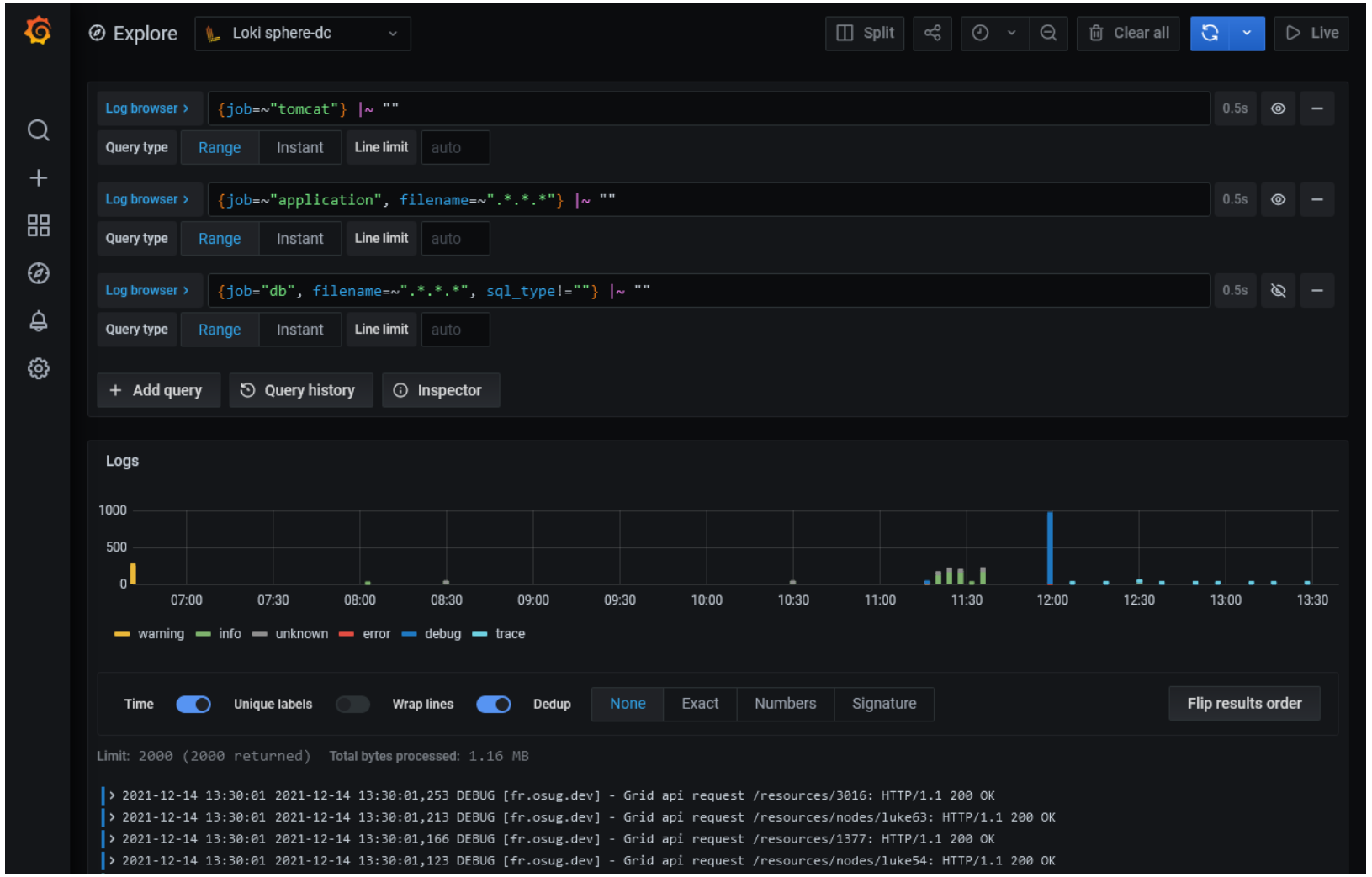
Grafana logs dashboard

SPHERE



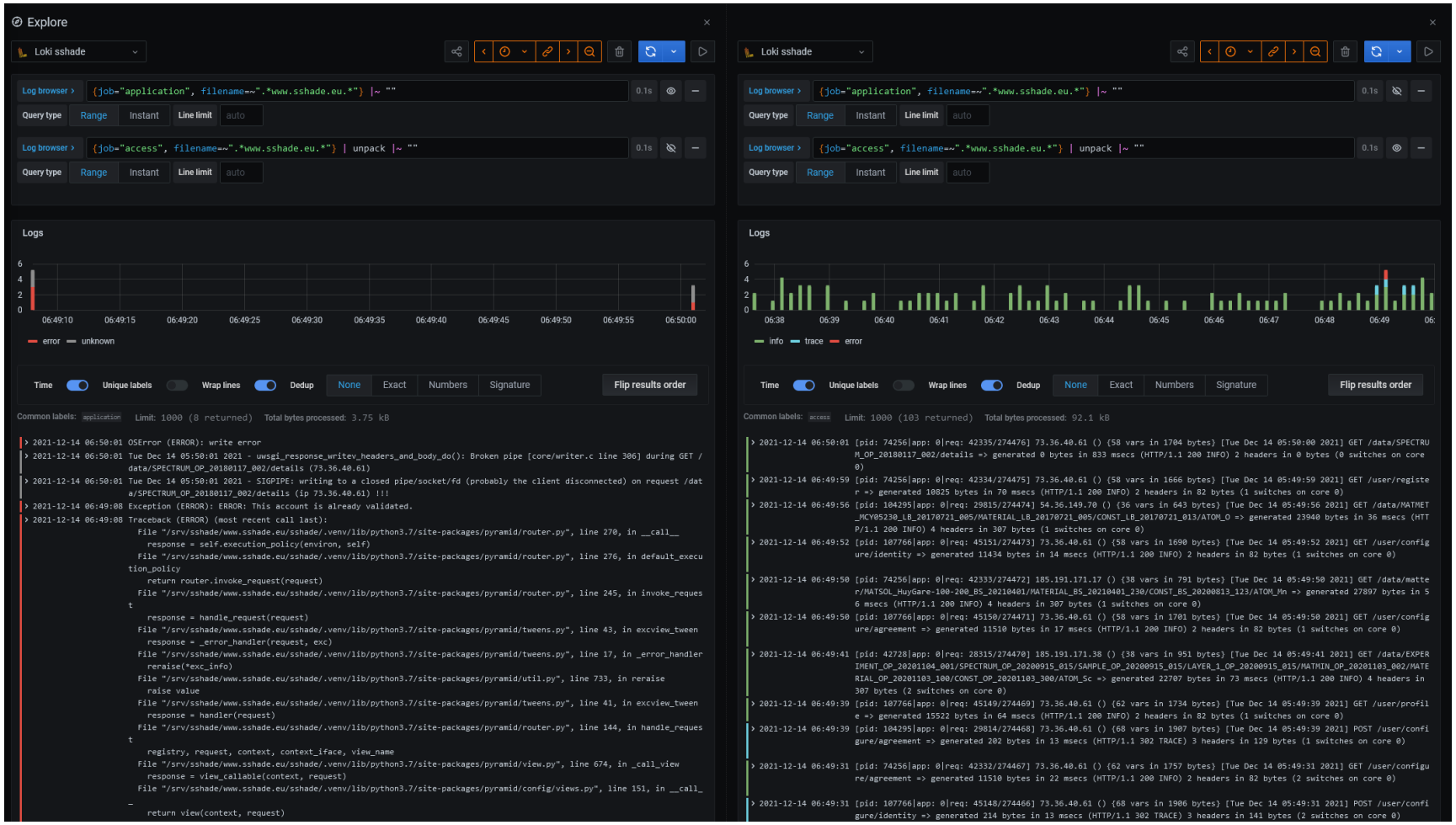
Grafana - logs

- Correlate activity/database requests to logs with combined logs
- Unpack JSON with `| unpack`, refine query with `| var = "value"`
- Explore labels, wrap, deduplication



Grafana - logs

- Explore combined logs : split, link, show/hide

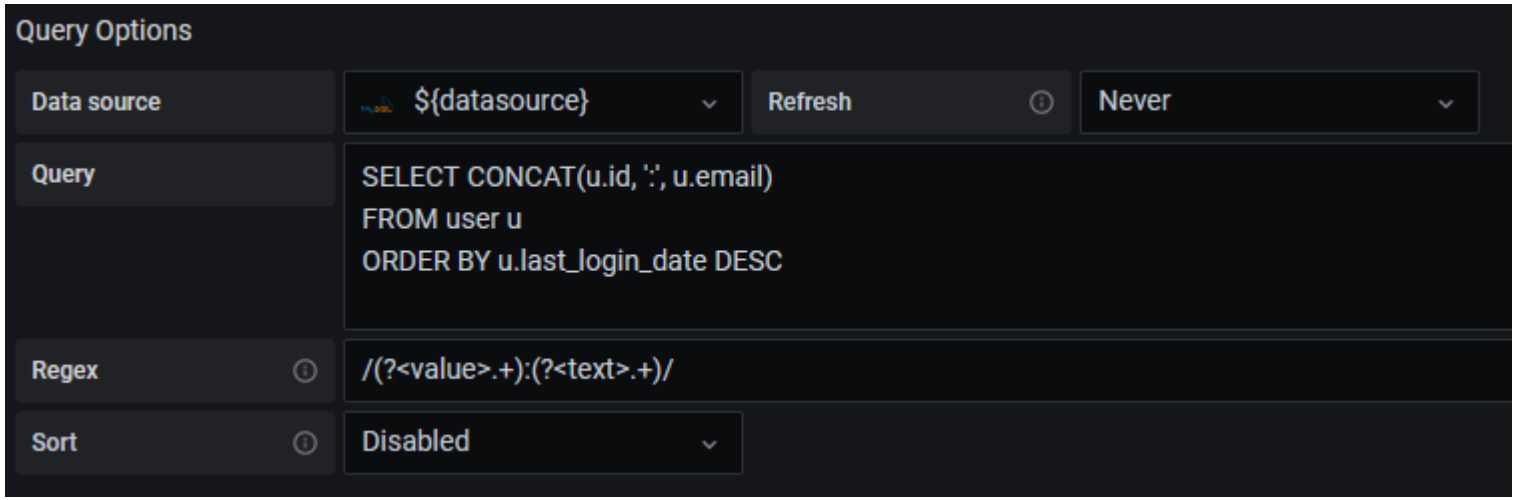


Variables

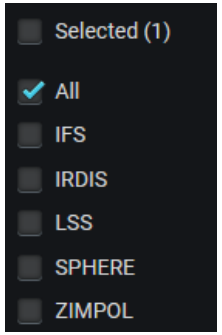
Variable	Definition
datasource	loki
db_datasource	postgres
project	.*,www.sshade.eu,dev.sshade.eu,sandbox.sshade.eu
search	

- Use variables for Loki and database sources
- Restrict logs to project : `filename=~".*$project.*"`
- Search field on all queries : `|~ "$search"`
- Exclude query, either :
  - exclude a non existing string by default (ie. `"_no_exclusion_"`)
  - hack the search field with `"!~ "value`

Variable from SQL

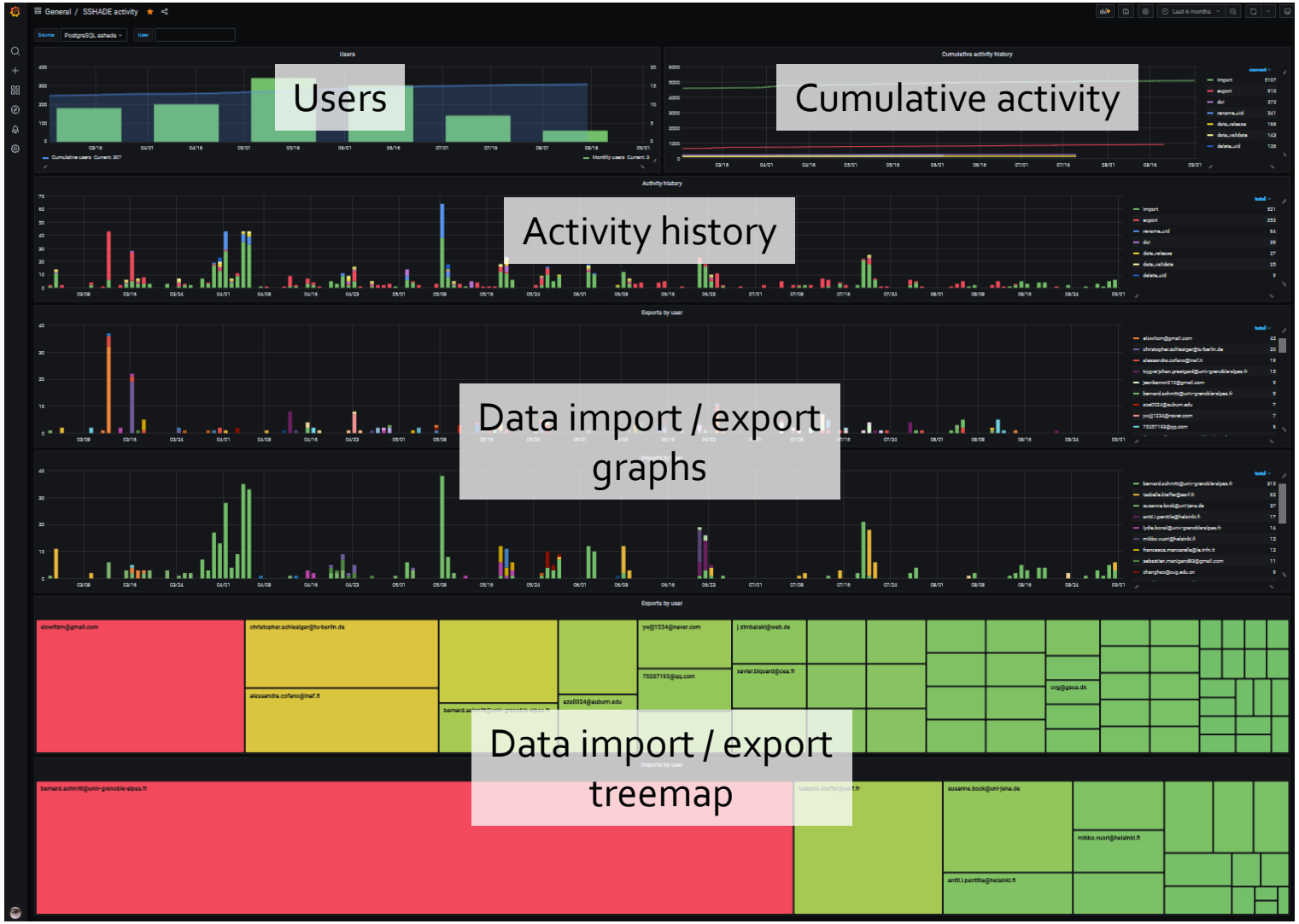


- Use correct datasource
- `CONCAT` multiple columns then split for value/text
- Fine-tune `ORDER` in the query instead of sort
- `All` used as `IN($variable)` can be slow for text, prefer numeric id if available



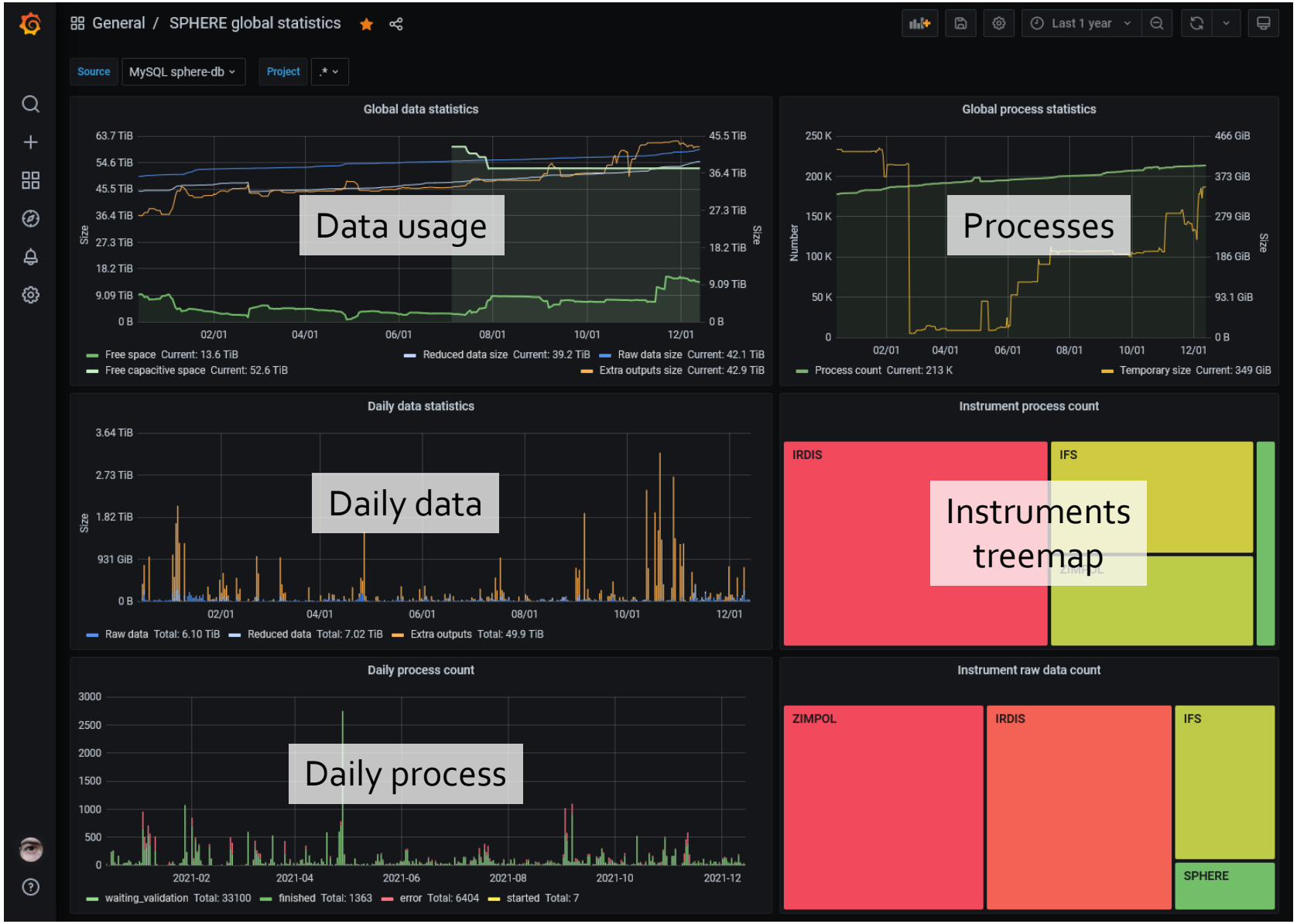
Grafana - activity

SSHADE



Grafana - global activity

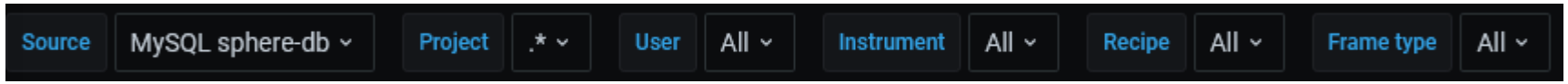
SPHERE



Grafana - detailed activity

SPHERE

Variables



Timeline

Data usage

Treemaps

Tips

- Use query option Max data points, Min interval to improve readability
- Use SQL sum(count(id)) OVER (ORDER BY date) to display cumulative graphs
- Use Legend `{{field}}` to display a field value
- Use Inspect / Panel JSON `"aliasColors"` to override colors consistently