



O T U S
ОНЛАЙН-ОБРАЗОВАНИЕ

Онлайн-образование



Меня хорошо видно && слышно?

Ставьте + , если все хорошо
Напишите в чат, если есть проблемы

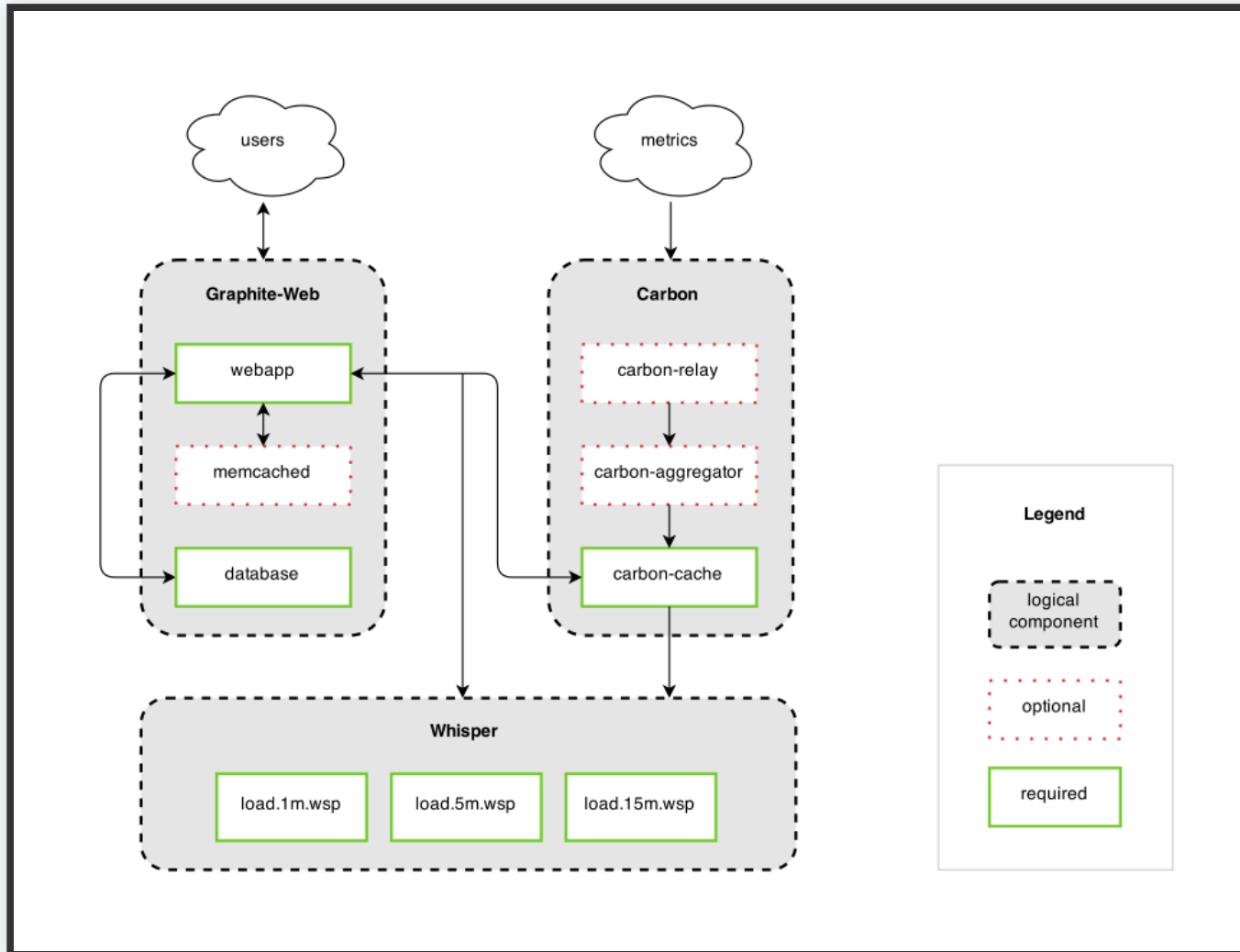
НЕ ЗАБЫТЬ ВКЛЮЧИТЬ
ЗАПИСЬ!!!

Мониторинг на основе Carbon Graphite Telegra

План занятия

- Whisper
 - хранение данных в whisper
 - retention policies в whisper
 - функции агрегации
- Carbon
 - metric API
 - render API
 - серии метрик
 - списки серий
 - функции
- Graphite
- Telegraf
- практический пример сбора и отсылки метрик телеграфом в графит

Архитектура



Запуск стэка в контейнере

```
docker run -d --name graphite --restart=always \  
-p 80:80 -p 2003-2004:2003-2004 -p 2023-2024:2023-2024 \  
-p 8125:8125/udp -p 8126:8126 -p 8080:8080 \  
graphiteapp/graphite-statsd
```

Whisper

- реализовано на python
- цели и назначение как у RRD (round-robin-database)
- хранение числовых данных с временной меткой (до секунды)
- фиксированный размер базы
- быстрая и надежная
- возможности агрегации данных для более длительного хранения
- поддерживает update операции

Whisper: data-points

- big-endian double-precision floats
- каждая метрика хранится в своем файле
- каждый файл имеет фиксированный размер
- размер файл регулируется retention policy
- файл занимает место вне зависимости от наличия в нем данных
- каждое значение имеет свой timestamp

Whisper: retention and resolution

- содержит несколько архивов с разным уровнем агрегации и длительностью хранения
- во все архивы запись идет одновременно
- retentions описываем в формате `frequency:history`
- `storage-schemas.conf`

```
[apache_busyWorkers]
pattern = ^servers\.www.*\.workers\.busyWorkers$
retentions = 15s:7d,1m:21d,15m:5y
```

Whisper: агрегация

- `storage-aggregation.conf`
- `xFilesFactor`
 - от 0 до 1, по умолчанию 0.5
 - указывает какой процент not NULL данных должны быть в предыдущем resolution для агрегации
- `aggregationMethod`
 - **average**, `sum`, `min`, `max`, `last`

```
[all_avg]
pattern = \.avg$
xFilesFactor = 0.1
aggregationMethod = avg
```

```
# пример расчета
' ' ' ' 0 0 1 1 2 3 4 5
avg = 16/8 = 2
```

изменение агрегации

- при изменении функции агрегации, нужно также менять и существующие файлы

```
whisper-set-aggregation-method.py test.wsp max
```

- [whisper scripts](#)

Carbon

- процессы для обработки метрик
 - carbon-cache
 - carbon-relay
 - carbon-aggregator

carbon-cache

- прием метрик по разным протоколам:
 - pickle, AMQP, plain text:
 - `<metric path> <metric value> <metric timestamp>`
- запись на диск
- реализует сервис запросов для in-memory метрик для graphite-web
- при возрастающей нагрузке можно запускать несколько кэшей
- при высокой нагрузке следует увеличить лимит открытых файлов

carbon-cache instances

```
[cache]
```

```
LINE_RECEIVER_INTERFACE = 127.0.0.1
```

```
PICKLE_RECEIVER_INTERFACE = 127.0.0.1
```

```
CACHE_QUERY_INTERFACE = 127.0.0.1
```

```
[cache:1]
```

```
LINE_RECEIVER_PORT = 2013
```

```
PICKLE_RECEIVER_PORT = 2014
```

```
CACHE_QUERY_PORT = 7012
```

```
[cache:2]
```

```
LINE_RECEIVER_PORT = 2023
```

```
PICKLE_RECEIVER_PORT = 2024
```

```
CACHE_QUERY_PORT = 7022
```

carbon-relay

- обеспечивает репликацию и шардинг
- carbon.conf RELAY_METHOD: rules, consistent-hashing
- relay-rules.conf

```
[example]
pattern = ^mydata\.foo\..+
servers = 10.1.2.3, 10.1.2.4:2004, myserver.mydomain.com
```

carbon-relay

- carbon.conf

```
[relay]
LINE_RECEIVER_INTERFACE = 0.0.0.0
LINE_RECEIVER_PORT = 2003
PICKLE_RECEIVER_INTERFACE = 0.0.0.0
PICKLE_RECEIVER_PORT = 2004
RELAY_METHOD = consistent-hashing
REPLICATION_FACTOR = 2
DESTINATIONS = 127.0.0.1:2014:1, 127.0.0.1:20
```

carbon-aggregator

- стоит перед carbon-cache
- буферизация и агрегация метрик перед сохранением в whisper
- aggregation-rules.conf
 - sum, avg, min, max, p50, p75, p80, p90, p95, p99, p999, count
 - output_template (frequency) = method input_pattern

```
<env>.applications.<app>.all.requests (60) = sum <env>.applications.<app>.*.requests  
<env>.applications.<app>.all.latency (60) = avg <env>.applications.<app>.*.latency  
<env>.applications.<app>.all.<app_metric> (60) = sum <env>.applications.<app>.*.<<app_
```

Graphite Api для получения данных

- metrics
- The Metrics API
 - /metrics/find
 - /metrics/expand
 - /metrics/index.json
- The Render API
 - генерация графов
 - /render
 - функции

series, series list

- series
 - metro_east.servers.apache02.system.load_average
- series list

```
metro_east.servers.apache02.system.load_average  
metro_east.servers.apache0{1,2,3}.system.load_average.1_min  
metro_east.servers.apache01.system.load_average.*
```

Graphite Events

- добавлена возможность сохранения текстовых событий в webapp database (SQLite)

```
curl -X POST "http://graphite/events/"
  -d '{ "what": "Event - deploy",
        "tags": ["deploy"],
        "when": 1467844481,
        "data": "deploy of master branch happened at Wed Jul  6 22:34:41 UTC }'
# получение событий
curl -s "http://graphite/render/?target=events('exception')&format=json" |
```

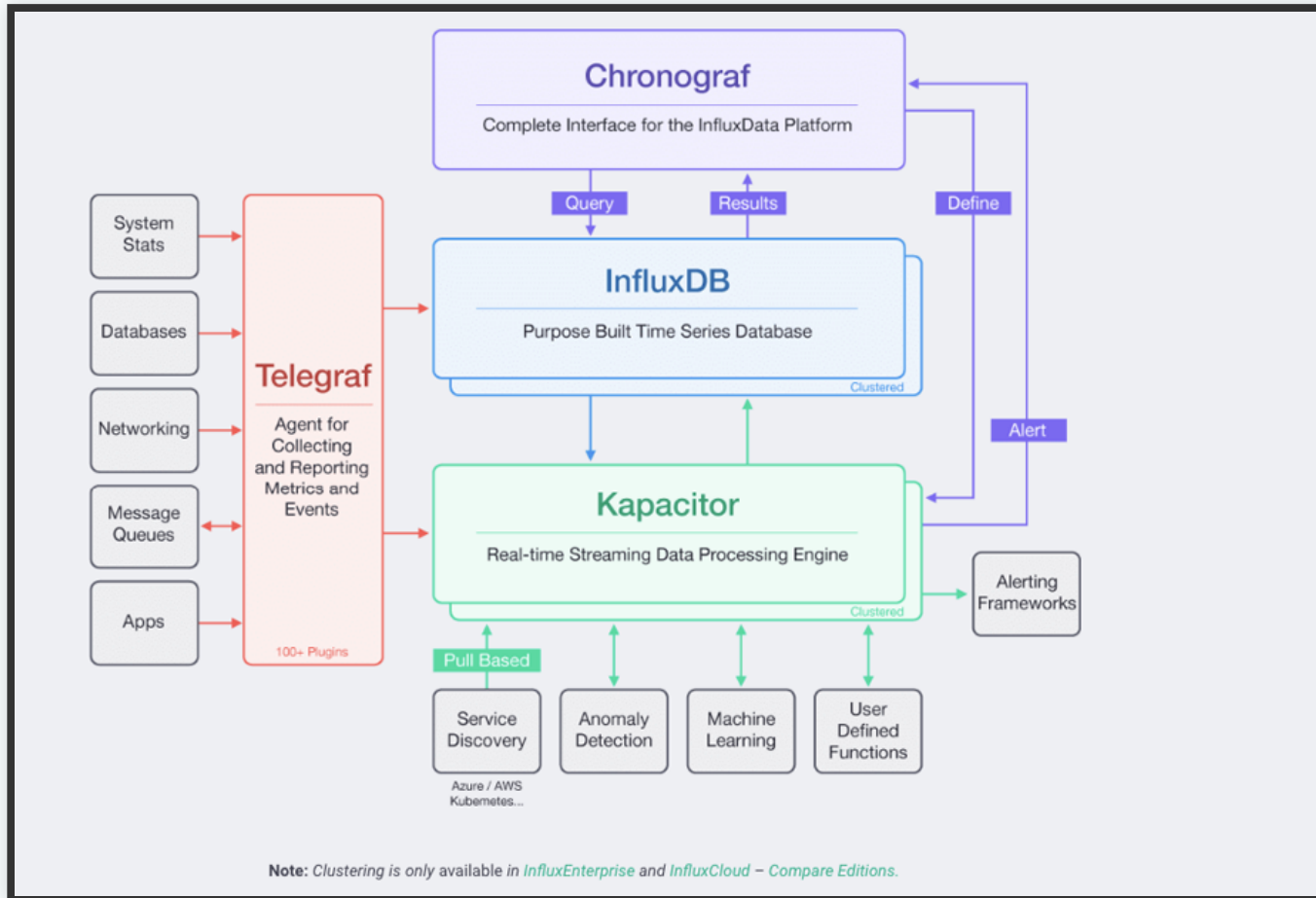
Statsd

- front-end проху для сбора
- агрегирует все метрики и складывает в Graphite только окончательные значения
- буферизирует данные и отправляет данные через указанный таймаут

Collectd

- сборщик метрик с машин
- большой набор плагинов
- `/etc/collectd/collectd.conf`

TICK



Telegraf

- сборщик метрик
- написан на Go
- система плагинов позволяет легко добавлять новые плагины для приема и отправки метрик
- большой набор существующих плагинов
- в том числе для отправки метрик в Graphite
- ставится в том числе под Windows

Конфигурация

- `telegraf config > telegraf.conf`
- документация

Метрики

- название метрики
- тэги: `key/value` для идентификации метрики
- поля: `key/value` для данных метрики
- `timestamp`


Рефлексия



Отметьте 3 пункта, которые вам запомнились с вебинара



Что вы будете применять в работе из сегодняшнего вебинара?

An aerial photograph of a dense city skyline, likely New York City, with numerous skyscrapers. The image is overlaid with a semi-transparent blue and teal gradient. A network of white lines connects various points across the blue area, creating a digital or data network aesthetic. The text is centered in white, bold, sans-serif font.

Заполните, пожалуйста,
опрос о занятии по ссылке в чате