GeekBrains

Troubleshooting

CI/CD

Что будет на уроке

1. Мониторинг.
2. Логирование.
3. Графики.
4. Troubleshooting GitLab’a.

Troubleshooting (траблшутинг) — поиск и устранение неисправностей.

Мониторинг

1. **Инфраструктуры**. Выявление технических проблем: выход из строя диска, заканчивающееся место, память и т. п.
2. **Приложения**. Важно получать и измерять показатели работоспособности самого приложения. Это могут быть RPS (request per second — количество запросов в секунду), время ответа, количество ошибок и т. п.

Цели мониторинга

1. Сбор данных.
2. Обработка данных.
3. Представление данных в удобном для анализа виде.

Logging (логирование) — запись информации о работе приложения.

```
2018-06-17 16:56:00.810  INFO 6082 --- [           main] s.w.s.m.m.a.RequestMappingHandlerAdapter : Looking for @ControllerAdvice: org.springframework.boot.web.servlet.context.AnnotationConfigServletWebServerApplicationContext@1d9b7cce: startup date [Sun Jun 17 16:55:55 CEST 2018]; root of context hierarchy
2018-06-17 16:56:01.023  INFO 6082 --- [           main] s.w.s.m.m.a.RequestMappingHandlerMapping : Mapped "{[/]}" onto public java.lang.String com.baeldung.springbootlogging.LoggingController.index()
2018-06-17 16:56:01.044  INFO 6082 --- [           main] s.w.s.m.m.a.RequestMappingHandlerMapping : Mapped "{[/error],produces=[text/html]}" onto public org.springframework.web.servlet.ModelAndView org.springframework.boot.autoconfigure.web.servlet.error.BasicErrorController.errorHtml(javax.servlet.http.HttpServletRequest,javax.servlet.http.HttpServletResponse)
2018-06-17 16:56:01.047  INFO 6082 --- [           main] s.w.s.m.m.a.RequestMappingHandlerMapping : Mapped "{[/error]}" onto public org.springframework.http.ResponseEntity<java.util.Map<java.lang.String, java.lang.Object>> org.springframework.boot.autoconfigure.web.servlet.error.BasicErrorController.error(javax.servlet.http.HttpServletRequest)
2018-06-17 16:56:01.119  INFO 6082 --- [           main] o.s.w.s.handler.SimpleUrlHandlerMapping  : Mapped URL path [/webjars/**] onto handler of type [class org.springframework.web.servlet.resource.ResourceHttpRequestHandler]
2018-06-17 16:56:01.120  INFO 6082 --- [           main] o.s.w.s.handler.SimpleUrlHandlerMapping  : Mapped URL path [/**] onto handler of type [class org.springframework.web.servlet.resource.ResourceHttpRequestHandler]
2018-06-17 16:56:01.398  INFO 6082 --- [           main] o.s.j.e.a.AnnotationMBeanExporter        : Registering beans for JMX exposure on startup
2018-06-17 16:56:01.528  INFO 6082 --- [           main] o.s.b.w.embedded.tomcat.TomcatWebServer  : Tomcat started on port(s): 8080 (http) with context path ''
2018-06-17 16:56:01.538  INFO 6082 --- [           main] c.b.s.SpringBootLoggingApplication       : Started SpringBootLoggingApplication in 7.455 seconds (JVM running for 8.627)
2018-06-17 16:56:03.085  INFO 6082 --- [nio-8080-exec-1] o.a.c.c.C.[Tomcat].[localhost].[/]       : Initializing Spring FrameworkServlet 'dispatcherServlet'
2018-06-17 16:56:03.085  INFO 6082 --- [nio-8080-exec-1] o.s.web.servlet.DispatcherServlet        : FrameworkServlet 'dispatcherServlet': initialization started
2018-06-17 16:56:03.103  INFO 6082 --- [nio-8080-exec-1] o.s.web.servlet.DispatcherServlet        : FrameworkServlet 'dispatcherServlet': initialization completed in 18 ms
2018-06-17 16:56:03.141  INFO 6082 --- [nio-8080-exec-1] c.b.springbootlogging.LoggingController  : An INFO Message
2018-06-17 16:56:03.142  WARN 6082 --- [nio-8080-exec-1] c.b.springbootlogging.LoggingController  : A WARN Message
2018-06-17 16:56:03.142 ERROR 6082 --- [nio-8080-exec-1] c.b.springbootlogging.LoggingController  : An ERROR Message
```

Плюсы логирования

1. Запись в файлы — лёгкий процесс.
2. Операция добавления записи в файл практически не влияет на производительность приложения.
3. Легко искать и анализировать. Такие инструменты, как grep, позволяют фильтровать информацию из файлов.
4. Файлы легко читаются и интерпретируются людьми.
5. Не зависят от конкретной технологии.

Минус логирования

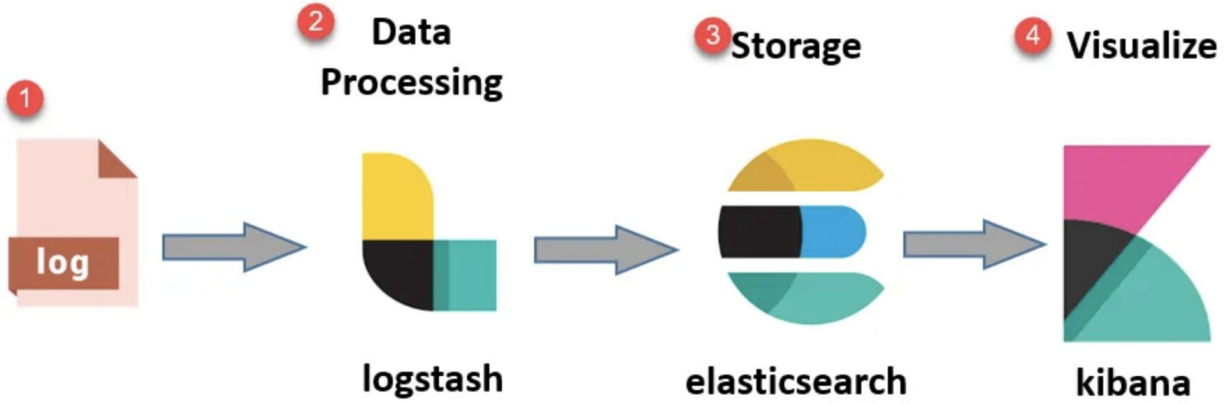
Чем больше файлов логов, тем сложнее их исследовать и проводить анализ.

К примеру, если у нас сервис развёрнут на 10 машинах и на каждой пишется лог, то при возникновении проблем с сервисом нужно будет ходить по всем 10 машинам и смотреть логи. Это не очень удобно.

Уровни логирования

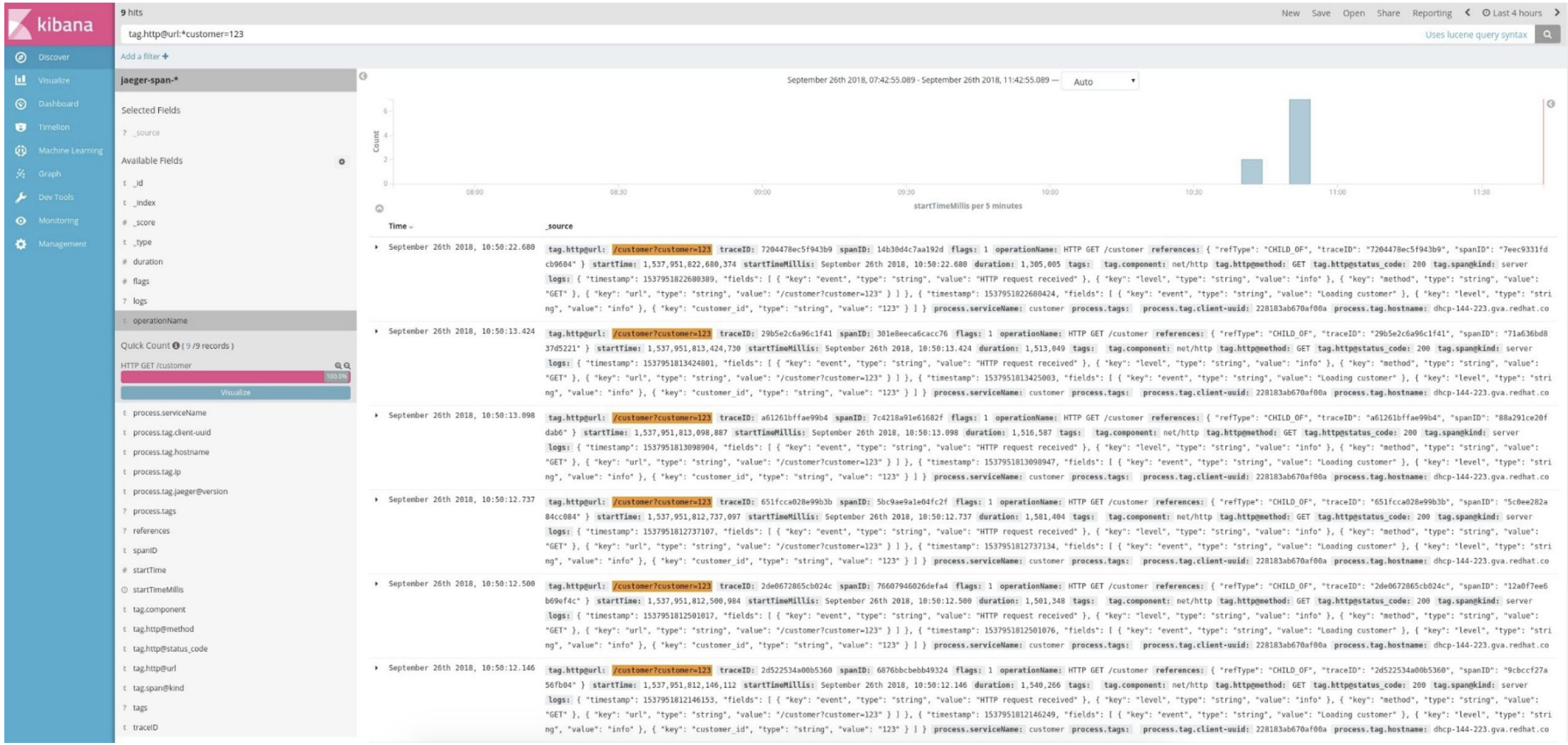
1. **Error.** Для критических ошибок.
2. **Warning.** Для некритических ошибок.
3. **Info.** Для информационных сообщений. Содержит информацию об успешно выполненных прикладных процедурах.
4. **Debug.** Для подробных сообщений, адресованных разработчикам. Не используется на постоянной основе, включается при возникновении проблем для поиска причин возникновения проблемы.

Обработка файлов логов



1. Сбор данных из файлов, которые могут находиться на разных серверах, в одно место.
2. Парсинг файлов.
3. Возможность анализировать данные и, соответственно, идентифицировать ошибки.

Kibana


kibana
9 hits
New Save Open Share Reporting Last 4 hours
tag.http@url:*customer=123
Uses lucene query syntax
Add a filter
Discover
Visualize
Dashboard
Timelion
Machine Learning
Graph
Dev Tools
Monitoring
Management
jaeger-span-*
Selected Fields
_source
Available Fields
_id
_index
_score
_type
duration
flags
logs
operationName
Quick Count (9 /9 records)
HTTP GET /customer
100.0%
Visualize
process.serviceName
process.tag.client-uuid
process.tag.hostname
process.tag.ip
process.tag.jaeger@version
process.tags
references
spanID
startTime
startTimeMillis
tag.component
tag.http@method
tag.http@status_code
tag.http@url
tag.span@kind
tags
traceID
September 26th 2018, 07:42:55.089 - September 26th 2018, 11:42:55.089 — Auto
Count
08:00
08:30
09:00
09:30
10:00
10:30
11:00
11:30
startTimeMillis per 5 minutes
Time
_source
September 26th 2018, 10:50:22.680
September 26th 2018, 10:50:13.424
September 26th 2018, 10:50:13.098
September 26th 2018, 10:50:12.737
September 26th 2018, 10:50:12.500
September 26th 2018, 10:50:12.146

Графики

1. Хранят числовые данные. Например, время отклика, потребление ресурсов.
2. Зачастую данные отображаются веб-приложением: Grafana, Screen в Zabbix.

Grafana


Stats Overview
Last 6 hours
Memory / CPU
memory
cpu
logins
timeshift -1h
logins
logins (-1 hour)
Memory
99 B
Sign ups
271
Logouts
197
Sign outs
272
server requests
2019-06-12 11:32:20
web_server_01: 26.55
web_server_02: 52.30
web_server_03: 78.43
web_server_04: 104.73
web_server_01
web_server_02
web_server_03
web_server_04
client side full page load
38 GB
7 GB
43 GB
19 GB
58 GB
69 GB
22 GB
62 GB
71 GB
55 GB
15 GB
68 GB
89 GB
52 GB
42 GB
5.3 GB
sda1
sda2
sda3
sda4
sda5
sda6
sda7
sda8
sda9
sda10
sda11
sda12
sda13
sda14
sda15
sda16

Возникновение проблемы

1. Восстановить работоспособность системы.
2. Troubleshooting. Поиск и устранение проблемы.

Troubleshooting

1. Определить масштаб проблемы. Проблема аффектит всех пользователей или только малую их часть?
2. В чём конкретно проблема? Не поднимается сервис, не открывается страница в браузере, сервис перестал отвечать на внешние запросы?
3. Установление причины проблемы. Диагностика, исследование. Что в логах, на графиках?

Troubleshooting

4. Воспроизведение проблемы.
5. Восстановление системы.
6. Написание post mortem. Отчёт о том, что привело к поломке, что было сделано для устранения, что можно сделать в дальнейшем, чтобы не допустить повторения проблемы.

Troubleshooting GitLab CI/CD.

GitLab Runner

При использовании локального GitLab и GitLab Runner’ов в случае появления проблем с GitLab Runner’ами можно включить debug-режим.

1. В консоли:

```
gitlab-runner --debug run
```

2. Правка в конфиге: config.toml log_level на debug.

Troubleshooting GitLab CI/CD.

GitLab Container Registry

```
docker push gitlab.example.com/myproject/docs:latest
The push refers to a repository [gitlab.example.com/myproject/docs]
630816f32edb: Preparing
530d5553aec8: Preparing
...
4b0bab9ff599: Waiting
d1c800db26c7: Waiting
42755cf4ee95: Waiting
unauthorized: authentication required
```

Причина

При пуше большого image при превышении 5 минут может возникать эта ошибка, так как по дефолту в настройках GitLab время жизни токена составляет 5 минут.

Troubleshooting GitLab CI/CD.

Ошибки на этапе Git push

При пуше в удалённый репозиторий может возникнуть следующая ошибка:

```
Write failed: Broken pipe
fatal: The remote end hung up unexpectedly
```

Причины

1. Недостаточный размер POST buffer в Git.
2. Ошибки в конфигурации SSH, если push проходит через SSH. Можно попробовать добавить в конфиг файл (~/.ssh/config):

```
Host your-gitlab-instance-url.com
  ServerAliveInterval 60
  ServerAliveCountMax 5
```

Troubleshooting GitLab CI/CD.

Ошибки в job

```
The deployment job is older than
the previously succeeded deployment job...
```

Причина

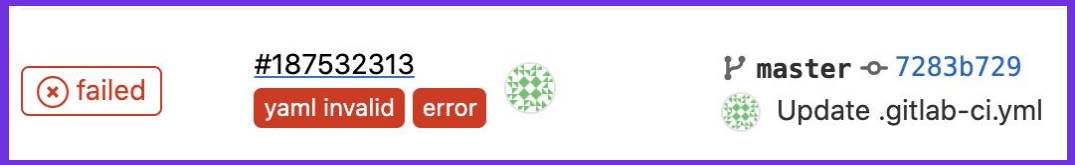
Выставленная галочка в Skip outdated deployment jobs.

На прошлом занятии мы её выставляли, чтобы более старые job’ы, которые выполняются в pipeline, не отрабатывали, если уже отработали более свежие job’ы.

То есть, другими словами, это ожидаемое поведение.

Troubleshooting GitLab CI/CD.

Синтаксические ошибки



Pipeline не запустится, если в файле gitlab_ci.yml будут синтаксические ошибки.

В таком случае необходимо пойти и перепроверить сам файл на ошибки либо использовать встроенный CI Lint:

CI/CD → Pipelines или CI/CD → Jobs.

Troubleshooting GitLab CI/CD.

Верификация переменных

Ключевое в траблшутинге GitLab CI/CD — понять, какие переменные есть в pipeline и какие значения им присвоены.

Для этого можно включить расширенное debug-логирование в самой job.

```
job_name:
  variables:
    CI_DEBUG_TRACE: "true"
```