ALTCRAFT PLATFORM

Поддержание жизненного цикла ПО

Требования к персоналу

Требования к специалистом по сопровождению платформы на стороне заказчика

Специалисты, осуществляющие мониторинг работоспособности платформы, администрирование и поддержку инфраструктуры заказчика, должны обладать следующими знаниями и навыками:

- техническое образование;
- знание работы протоколов TCP/IP;
- знание основных принципов работы БД;
- навыки администрирования ОС Linux не менее 1-ого года;
- знание функциональных возможностей платформы;
- навыки работы с MongoDB;
- знание средств восстановления баз данных и мониторинга производительности серверов;

Устранение программных ошибок в работе платформы осуществляется только специалистами компании ООО "Альткрафт", после предоставления им собранной диагностической информации.

Процессы сопровождения платформы

Консультации пользователей и администраторов по вопросам эксплуатации

Консультацию пользователей и администраторов платформы осуществляет ООО "Альткрафт".

Консультация может происходить в персональном чате, email или телефону, в зависимости о уровня приобретаемой поддержки.

Регулярные обновления платформы

На весь срок права использования платформой заказчику предоставляется право получения регулярных бесплатных обновлений.

Количество обновлений составляет не менее 4 крупных обновлений в год.

Выход одного обновления не реже чем 1 раз в квартал.

Устранение программных ошибок

Устранение программных ошибок в работе платформы осуществляется только специалистами компании ООО "Альткрафт", после предоставления им собранной диагностической информации.

Вся диагностическая информация может быть передана через персональный чат поддержки или по электронной почте team@altcraft.com.

Модификация платформы

Создание, изменение, модернизация компонентов/сервисов платформы осуществляется только специалистами компании ООО "Альткрафт".

Коллектив разработчиков платформы обладают необходимым набором знаний для работы со всеми компонентами.

У заказчика есть возможность согласовать возможность доработки функционала платформы для решения своих бизнес задач согласовав техническое задание с ООО "Альткрафт".

Для согласования технического задания обратитесь по адресу customers.department@altkraft.com.

Обновление платформы

Обновление в ручном режиме

Для обновления поставляется сжатый файл: *AKD.tar.gz* – управляющий модуль.

Необходимо распаковать архив *AKD.tar.gz*:

```
tar -xzf AKD.tar.gz
```

Изменения в конфигурационных файлах не требуются.

Обновление с помощью самораспаковывающегося архива

Проверьте целостность поставляемого архива при помощи команды *md5sum* способом описанным в разделе "Подготовка сервера к установке" данной инструкции.

Запуск установщика с ключом *-u* или *-update* позволяет немедленно приступить к обновлению:

```
bash AKD_Auto_Installer --update
```

Также, можно сразу указать путь к существующему экземпляру AKD:

```
bash AKD_Auto_Installer --update /opt/akd
```

После запуска будет проверено существование сконфигурированного экземпляра системы, и в случае нахождения такого, будет создана резервная копия директорий, процессы AKD будут остановлены, и после обновления файлов системы запущены снова. Создание резервной копии может занять некоторое время.

При возникновении каких-либо изменений в конфигурации, установщик предложит указать соответствующие настройки, до запуска системы.

Диагностика проблем и восстановление работы после сбоя

На этой странице:

Проверка работоспособности платформы

1) Проверьте работоспособность БД

Одна из часто встречаемых проблем – это недоступность одной из баз данных, необходимых для работы процесса.

```
# MongoDB
systemctl status mongod-account.service
systemctl status mongod-control.service

# SSDB
systemctl status ssdb-actions.service
systemctl status ssdb-hb.service
systemctl status ssdb-notify.service
systemctl status ssdb-stat.service
```

В случае ошибок, убедитесь в наличии свободного места на дисках и изучите следующие логи:

- `/var/log/mongodb/mongod-account.log`
- `/var/log/mongodb/mongod-control.log`

Устраните причину сбоя и перезапустите базу данных.

2) Проверьте доступность брокер сообщений RabbitMQ

```
# RabbitMQ
systemctl status rabbitmq-server.service
```

В случае ошибок, убедитесь в наличии свободного места на дисках и изучите лог файлы в директории /var/log/rabbitmq/

Устраните причину и перезапустите брокер сообщений.

3) Проверьте доступность сервисов платформы

Чтобы проверить общую работоспособность сервисов платформы достаточно перейти по ссылке

https://<url панели управления платформы>/status

> Рекомендуем настроить регулярную HTTP проверку на наличие текста "OK" на странице

Если страница возвращает "OK", значит все в порядке все сервисы подняты.

Если вы видите название сервиса на странице, значит необходимо проверить его лог файл и выяснить причину падения.

Соберите всю информацию и передайте ее в службу технической поддержки платформы в персональный чат или отправьте архив на team@altcraft.com.

Восстановления работоспособности сервиса

Для восстановления работоспособности сервиса попробуйте перезапустить его и проверьте логи заново.

```
./akd onerestart < >
```

4) Для корректной работы после сбоя, перезапустите все сервисы платформы

```
./akd restart
```

Лог-файлы платформы

Лог-файлы сервисов находятся в каталоге `./logs`, относительно каталогу с платформой. Н - .

Уровни логирования могут быть определены в файле конфигурации `main.json`, параметром `LOG_LEVEL`.

Сбор информации для запроса в поддержку

В некоторых случаях, системный администратор может определить проблему самостоятельно при изучении логов системы. Для понимания и воспроизведения более сложной ошибки, нашим специалистам необходимо изучить логи системы более детально.

Получите версиии исполняемы файлов с помощью скрипта ниже, передав первым параметром путь до каталога с платформой. Запишите вывод в файл.

```
#!/bin/bash

basepath=$1
if [ "" = "$basepath" ]; then
        basepath="."
fi

PROCESSES=(`$basepath/akd --processes`)


for process in "${PROCESSES[@]}"; do
        echo "$process:$($basepath/ak/bin/$process --version)"
done
```

Подготовьте архив с журналами ПО:

```
tar -czf "logs-$(date --iso-8601).tar.gz" \
  ./logs \
  ./nginx/logs \
  /var/log/syslog \
  /var/log/mongodb/mongod-account.log \
  /var/log/mongodb/mongod-control.log \
  /var/log/rabbitmq/rabbit@host.log
```

Для дополнительной информации, запустите следующий скрипт с помощью Python 3, передав первым параметром путь до каталога с платформой. Запишите вывод в файл. Убедитесь в наличии в системе дополнительного модуля psutil.

```
import psutil
import collections
import json

PROCESSES = [
"AK:adminwebcontrol",
```

```
"AK:akmtad",
"AK:api",
"AK:cookie_saver",
"AK:job_server",
"AK:node",
"AK:procactions",
"AK:proceventgen",
"AK:prochook",
"AK:procintegras",
"AK:procleadsaver",
"AK:procmandrillev",
"AK:procnotify",
"AK:procpiper",
"AK:procpixel",
"AK:procplmt",
"AK:procpush",
"AK:procresume",
"AK:procrpc",
"AK:procsenderev",
"AK:procsmsev",
"AK:procsmslisten",
"AK:proctask",
"AK:proctrigger",
"AK:procwebver",
"AK:procworkflow",
"AK:pywebcontrol",
"AK:stataggregator",
"AK:tariffcontroller",
"AK:trk_amazon_sns",
"AK:trkaction",
"AK:trkcustom",
"AK:trkimage",
"AK:trklisten",
"AK:trkmandrill",
"AK:trkpush",
"AK:trkread",
"AK:trksms",
"AK:trkwebversion",
"AK:webcontrol",
]

TCP_CONNECTIONS = psutil.net_connections(kind="tcp")


def make_hash():
    return collections.defaultdict(make_hash)


def get_information(process: psutil.Process) -> dict:
    data = collections.defaultdict(make_hash)

    with process.oneshot():
        # https://psutil.readthedocs.io/en/latest/#process-class
        data["pid"] = process.pid
        data["name"] = process.info["cmdline"][0]
        data["exe"] = process.exe()
        data["create_time"] = process.create_time()
        data["status"] = process.status()
        data["username"] = process.username()
        data["nice"] = process.nice()

        data["rlimit"]["RLIMIT_NOFILE"]["soft"] = process.rlimit(psutil.RLIMIT_NOFILE)[0]
        data["rlimit"]["RLIMIT_NOFILE"]["hard"] = process.rlimit(psutil.RLIMIT_NOFILE)[1]

        data["io_counters"] = process.io_counters()._asdict()

        data["num_fds"] = process.num_fds()

        data["num_threads"] = process.num_threads()
        data["cpu_times"] = process.cpu_times()._asdict()
        data["cpu_num"] = process.cpu_num()
```

```
        data["memory_full_info"] = process.memory_full_info()._asdict()
        data["memory_percent"] = process.memory_percent()

        # Network connections
        data["net_connections"]["tcp"]["states"] = collections.Counter(
            {
                "ESTABLISHED": 0,
                "SYN_SENT": 0,
                "SYN_RECV": 0,
                "FIN_WAIT1": 0,
                "FIN_WAIT2": 0,
                "TIME_WAIT": 0,
                "CLOSE": 0,
                "CLOSE_WAIT": 0,
                "LAST_ACK": 0,
                "LISTEN": 0,
                "CLOSING": 0,
                "NONE": 0,
            }
        )
        data["net_connections"]["tcp"]["states"].update(
            collections.Counter(
                connection.status
                for connection in TCP_CONNECTIONS
                if connection.pid == process.pid
            )
        )
        data["net_connections"]["tcp"]["total"] = sum(
            data["net_connections"]["tcp"]["states"].values()
        )

    return data


if __name__ == "__main__":
    result = []
    for process in psutil.process_iter(["name", "cmdline"]):
        if len(process.info["cmdline"]) > 0:
            if process.info["cmdline"][0] in PROCESSES:
                result.append(get_information(process))

    for name in PROCESSES:
        if not any(process["name"] == name for process in result):
            result.append({"name": name, "status": "Not running"})

    print(json.dumps(result, sort_keys=True))
```

Резервное копирование данных платформы

Для полного резервного копирования и восстановления, необходимы следующие данные:

- управляющий модуль
- базы данных MongoDB (`account`, `control`)
- базы данных SSDB (`actions`, `hb`, `notify`, `stat`)

Управляющий модуль

Резервное копирование

Поместите содержимое управляющего модуля в архив, исключая файлы `*.pid` и `*.log`.

```
#!/usr/bin/env bash

tar -czf /backup/altcraft-$(date +%F).tar.gz /opt/altcraft --exclude=*.pid --exclude=*.log
```

Восстановление

Распакуйте архив на место и перезапустите управляющий модуль.

```
#!/usr/bin/env bash

/opt/altcraft/akd restart
```

MongoDB

Методы резервного копирования данных описаны в официальной документации.
https://docs.mongodb.com/v3.4/core/backups/

Резервное копирование

Выполните полный дамп существующих баз и коллекций, используя утилиту `mongodump`.
Используйте сжатие (`--gzip`) на свое усмотрение.

Описание утилиты mongodb.
https://docs.mongodb.com/manual/reference/program/mongodump/

```
#!/usr/bin/env bash

mongodump --host 127.0.0.1 --port 27017 --gzip --archive=/backup/mongodb-control-$(date +%F).tar.gz
mongodump --host 127.0.0.1 --port 27018 --gzip --archive=/backup/mongodb-account-$(date +%F).tar.gz
```

Восстановление

Восстановите данные с помощью утилиты `mongorestore` и перезапустите сервисы `systemd`.

Описание утилиты `mongorestore`.
https://docs.mongodb.com/manual/reference/program/mongorestore/

```
#!/usr/bin/env bash

mongorestore --host 127.0.0.1 --port 27017 --gzip --arcdhive=/backup/mongodb-control-2019-06-01.tar.gz
mongorestore --host 127.0.0.1 --port 27018 --gzip --arcdhive=/backup/mongodb-account-2019-06-01.tar.gz
systemctl restart mongod-control
systemctl restart mongod-account
```

SSDB

Пример размещения баз данных на диске.

```
/var/lib/ssdb
 actions
    data
    meta
 hb
    data
    meta
 notify
    data
    meta
 stat
     data
     meta
```

Резервное копирование

Для каждой базы поместите каталоги, содержащие `data` и `meta`, в архив.

```
#!/usr/bin/env bash

tar -czf /backup/ssdb-actions-$(date +%F).tar.gz /var/lib/ssdb/actions
tar -czf /backup/ssdb-hb-$(date +%F).tar.gz /var/lib/ssdb/hb
tar -czf /backup/ssdb-notify-$(date +%F).tar.gz /var/lib/ssdb/notify
tar -czf /backup/ssdb-stat-$(date +%F).tar.gz /var/lib/ssdb/stat
```

Восстановление

Распакуйте содержимое архивов на место и перезапустите сервисы `systemd`.

```
#!/usr/bin/env bash

systemctl restart ssdb-actions
systemctl restart ssdb-hb
systemctl restart ssdb-notify
systemctl restart ssdb-stat
```

Мониторинг работы платформы

В зависимости от используемой в организации системы мониторинга, специалист должен выбрать способ отслеживания состояния процессов платформы.

На странице описаны все процессы, которые необходимо поддерживать в рабочем состоянии для полноценной работы платформы.

Мы рекомендуем использовать проверку на существование процесса по имени, встроенными в систему мониторинга средствами, либо с помощью утилиты pidof.

В обязательном порядке по хосту (или виртуальной машине) должна собираться такая информация, как:
- потребление и количество свободного ОЗУ;
- потребление и количество свободного времени CPU;
- количество свободного места и информация по утилизации дисков (скорость, задержки, SMART);
- количество открытых файлов и соединений.

Базы данных и сервисы

	IP адрес по умолчанию	Порт по умолчанию	Важность	Уведомление
MongoDB (control) Примечание: базы данных могут быть объединены в одну	127.0.0.1	27017	Чрезвычайная	Невозможно установить TCP соединение
MongoDB (accounts) Примечание: базы данных могут быть объединены в одну	127.0.0.1	27018	Чрезвычайная	Невозможно установить TCP соединение
SSDB (actions)	127.0.0.1	4410	Чрезвычайная	Невозможно установить TCP соединение
SSDB (hb)	127.0.0.1	4420	Чрезвычайная	Невозможно установить TCP соединение
SSDB (notify)	127.0.0.1	4430	Чрезвычайная	Невозможно установить TCP соединение
SSDB (stats)	127.0.0.1	4440	Чрезвычайная	Невозможно установить TCP соединение
RabbitMQ	0.0.0.0	5672	Чрезвычайная	Невозможно установить TCP соединение

Очереди RabbitMQ

Виртуальный хост	Наименование очереди	Поставщик (producer)	Потребитель (consumer)	Описание	Важность	Уведомление
/	oxy_triggers	*	AK:proctrigger	Триггеры (кампании)	Высокая	Минимальное количество сообщений в очереди >= 5000 за 10 минут
/	oxy_triggers_prior	*	AK:proctrigger	Приоритетные триггеры (кампании)	Высокая	Минимальное количество сообщений в очереди >= 1000 за 10 минут
/	trk_*	AK:trk* AK: cookie_saver	procactions procpixel	События, обработанные трекингами	Высокая	Минимальное количество сообщений в очереди >= 1000 за 10 минут
akmta_senders	akmta_*	AK:proctrigger AK:webcontrol campaign AK: procworkflow	AK:akmtad	Сообщения для отправки	Высокая	Количество сообщений на уменьшается в течение 50 минут
akmta_senders	geo_akmta_*	AK:proctrigger AK:webcontrol campaign AK: procworkflow	AK:akmtad	Сообщения, для отправки по часовым поясам	Высокая	Количество сообщений на уменьшается в течение 50 минут

akmta_senders	prior_akmta_*	AK:proctrigger AK:webcontrol campaign AK: procworkflow	AK:akmtad	Приоритетные сообщения для отправки	Высокая	Количество сообщений на уменьшается в течение 50 минут

Процессы AKD

Для проверки работоспособности платформы достаточно убедиться в наличии процесса в виртуальной файловой системе `/proc`.

Полный список процессов можно посмотреть командой – `<BASEDIR>/akd --processes`
Расположение PID-файла процесса, по умолчанию – `<BASEDIR>/pids/<  >.pid`

Наименование исполняемого файла	Наименование процесса	Описание	Важность	Уведомление
`adminwebcontrol`	`AK:adminwebcontrol`	Панель администратора	Средняя	Процесс не найден в виртуальной файловой системе `/proc`
`akmtad`	`AK:akmtad`	Агент пересылки сообщений, AKMTA	Высокая	Процесс не найден в виртуальной файловой системе `/proc`
`api`	`AK:api`	API	Высокая	Процесс не найден в виртуальной файловой системе `/proc`
`cookie_saver`	`AK:cookie_saver`	Работа с пользовательскими cookie	Высокая	Процесс не найден в виртуальной файловой системе `/proc`
`proctask`	`AK:proctask`	Выполнение задач, запуск кампаний	Высокая	Процесс не найден в виртуальной файловой системе `/proc`
`procactions`	`AK:procactions`	Обработка событий для статистики (клики, открытия, подписки и т.п.)	Высокая	Процесс не найден в виртуальной файловой системе `/proc`
`proceventgen`	`AK:proceventgen`	Генератор событий	Средняя	Процесс не найден в виртуальной файловой системе `/proc`
`prochook`	`AK:prochook`	Захват различных событий из платформы	Высокая	Процесс не найден в виртуальной файловой системе `/proc`
`procpixel`	`AK:procpixel`	Обработка событий пикселей	Высокая	Процесс не найден в виртуальной файловой системе `/proc`
`procintegras`	`AK:procintegras`	Интеграции с внешними системами (AppMetrica и т. п.)	Высокая	Процесс не найден в виртуальной файловой системе `/proc`
`procleadsaver`	`AK:procleadsaver`	Обработка статистики импортов	Средняя	Процесс не найден в виртуальной файловой системе `/proc`
`procnotify`	`AK:procnotify`	Обработка уведомлений	Средняя	Процесс не найден в виртуальной файловой системе `/proc`
`procpush`	`AK:procpush`	Обработка событий пушей	Высокая	Процесс не найден в виртуальной файловой системе `/proc`
`procresume`	`AK:procresume`	Обработка и сканирование БД в целях возобновления статуса профилей	Средняя	Процесс не найден в виртуальной файловой системе `/proc`
`procrpc`	`AK:procrpc`	RPC-клиент для обработки RPC-соединений с процессами	Высокая	Процесс не найден в виртуальной файловой системе `/proc`
`procsenderev`	`AK:procsenderev`	Обработка событий	Высокая	Процесс не найден в виртуальной файловой системе `/proc`
`procsmsev`	`AK:procsmsev`	Запрашивает информацию по SMS-отправкам	Высокая	Процесс не найден в виртуальной файловой системе `/proc`
`procsmslisten`	`AK:procsmslisten`	Обработка ответов от SMS-шлюзов	Высокая	Процесс не найден в виртуальной файловой системе `/proc`
`proctrigger`	`AK:proctrigger`	Обработка триггер-кампаний	Высокая	Процесс не найден в виртуальной файловой системе `/proc`
`procwebver`	`AK:procwebver`	Обработка веб-версий	Средняя	Процесс не найден в виртуальной файловой системе `/proc`
`procworkflow`	`AK:procworkflow`	Обработка цепочек	Высокая	Процесс не найден в виртуальной файловой системе `/proc`

`stataggregator`	`AK: stataggregator`	Агрегация событий	Высокая	Процесс не найден в виртуальной файловой системе `/proc`
`tariffcontroller`	`AK: tariffcontroller`	Контроль ограничений на количество отправок (тарифы)	Средняя	Процесс не найден в виртуальной файловой системе `/proc`
`trkaction`	`AK:trkaction`	Регистрация событий по трекингу	Высокая	Процесс не найден в виртуальной файловой системе `/proc`
`trk_amazon_sns`	`AK: trk_amazon_sns`	Регистрация событий по сендеру Amazon	Высокая	Процесс не найден в виртуальной файловой системе `/proc`
`trkmandrill`	`AK:trkmandrill`	Регистрация событий по сендеру Mandrill	Высокая	Процесс не найден в виртуальной файловой системе `/proc`
`trkimage`	`AK:trkimage`	Регистрация событий по пикселям	Высокая	Процесс не найден в виртуальной файловой системе `/proc`
`trkpush`	`AK:trkpush`	Регистрация событий по трекингу пушей	Высокая	Процесс не найден в виртуальной файловой системе `/proc`
`trkread`	`AK:trkread`	Регистрация событий по чтениям email-сообщений	Высокая	Процесс не найден в виртуальной файловой системе `/proc`
`trksms`	`AK:trksms`	Регистрация событий по чтениям SMS-сообщений	Высокая	Процесс не найден в виртуальной файловой системе `/proc`
`trkwebversion`	`AK: trkwebversion`	Регистрация событий по чтениям из веб-версий	Высокая	Процесс не найден в виртуальной файловой системе `/proc`
`webcontrol`	`AK:webcontrol`	Пользовательский веб-интерфейс	Высокая	Процесс не найден в виртуальной файловой системе `/proc`

Дополнительно по каждому процессу

Вы так же можете собирать дополнительные данные по процессам, такие как: потребление памяти процессом, количество открытых файлов и соединений, использование CPU и т.д.. Пример получения дополнительной информации по процессам можно найти на странице Диагностика проблем, в разделе Сбор информации для запроса в поддержку.

Процессорное время

https://en.wikipedia.org/wiki/CPU_time

Тип метрики	Описание	Важность	Уведомление
`CPU system time`	Использование CPU процессом в процентах (`system`)	Информация	-
`CPU iowait time`	Использование CPU процессом в процентах (`iowait`)	Высокая	> 15% * количество ядер CPU
`CPU user time`	Использование CPU процессом в процентах (`user`)	Информация	-
`CPU utilization`	Использование CPU процессом в процентах (`total`)	Высокая	> 50% * количество ядер CPU

Память

Тип метрики	Описание	Важность	Уведомление
`RSS (resident set size)`	https://en.wikipedia.org/wiki/Resident_set_size	Высокая	> 20% от общего количество памяти
`SWAP`	https://en.wikipedia.org/wiki/Paging#Unix_and_Unix-like_systems	Высокая	> 5% от общего объема SWAP

Сеть

Для выявление проблем с сетью, рекомендуется мониторить количество соединений по каждому состоянию.
https://en.wikipedia.org/wiki/Transmission_Control_Protocol#Protocol_operation

Тип метрики	Описание	Важность	Уведомление
`CLOSE`	Закрыт. Сокет не используется.	Информация	-
`CLOSE_WAIT`	Удаленная сторона отключилась; ожидание закрытия сокета.	Средняя	Количество соединений > 2500
`CLOSING`	Сокет закрыт, затем удаленная сторона отключилась; ожидание подтверждения.	Информация	-
`ESTABLISHED`	Соединение установлено.	Средняя	количество соединений > 5000

FIN_WAIT1	Сокет закрыт; отключение соединения.	Информация	-
FIN_WAIT2	Сокет закрыт; ожидание отключения удаленной стороны.	Информация	-
LAST_ACK	Удаленная сторона отключилась, затем сокет закрыт; ожидание подтверждения.	Информация	-
LISTEN	Ожидает входящих соединений.	Информация	-
SYN_RECV	Идет начальная синхронизация соединения.	Информация	-
SYN_SENT	Активно пытается установить соединение.	Высокая	количество соединений > 5000
TIME_WAIT	Сокет закрыт, но ожидает пакеты, ещё находящиеся в сети для обработки	Высокая	количество соединений > 5000

Сбор логов и мониторинг с помощью ELK (Elasticsearch, Logstash, Kibana)

Содержание:

Установка ELK

Следуйте инструкциям по установке из официальной документации:
https://www.elastic.co/guide/en/elastic-stack/current/installing-elastic-stack.html

Конфигурирование Elasticsearch

Документация: https://www.elastic.co/guide/en/elasticsearch/reference/current/index.html

После установки Elasticsearch отредактируйте параметр *node.name* в файле конфигурации */etc/elasticsearch/elasticsearch.yml*.

Небольшая справка:

```
#
curl 'localhost:9200/_cat/templates?v&s=name'
#
curl 'localhost:9200/_cat/indices?v&s=index'
#
curl -XDELETE 'localhost:9200/<index>'
```

Конфигурирование Curator

Документация: https://www.elastic.co/guide/en/elasticsearch/client/curator/current/index.html

Пример конфигурации – *~/.curator/curator.yml*:

```
client:
  hosts:
    - localhost
  port: 9200
  url_prefix:
  use_ssl: False
  certificate:
  client_cert:
  client_key:
  ssl_no_validate: False
  http_auth:
  timeout: 30
  master_only: False

logging:
  loglevel: INFO
  logfile:
  logformat: default
  blacklist: ['elasticsearch', 'urllib3']
```

Пример события для удаления индексов старше 90 дней – *~/.curator/delete_indices.yml*:

```
actions:
  1:
    action: delete_indices
    description: >-
      Delete indices older than 90 days (based on index name), for filebeat-
      prefixed indices. Ignore the error if the filter does not result in an
      actionable list of indices (ignore_empty_list) and exit cleanly.
    options:
      ignore_empty_list: True
      disable_action: False
    filters:
    - filtertype: pattern
      kind: prefix
      value: filebeat-
    - filtertype: age
      source: name
      direction: older
      timestring: '%Y.%m.%d'
      unit: days
      unit_count: 90
```

Проверьте вашу конфигурацию с помощью параметра *--dry-run:*

```
/usr/local/bin/curator ~/.curator/delete_indices.yml --dry-run
```

4) Добавьте следующую строку в */etc/crontab* для периодической очистки индексов:

```
# Curator. Delete indices older than 90 days (filebeat-)
0  0    * * *   root    /usr/local/bin/curator ~/.curator/delete_indices.yml > /dev/null
```

Конфигурирование Logstash

Документация: https://www.elastic.co/guide/en/logstash/current/index.html

Создайте отдельную папку для хранения паттернов регулярных выражений, например – */etc/logstash/patterns*.

Разбейте **pipeline** на несколько файлов, например:

- */etc/logstash/conf.d/10-input.conf*
- */etc/logstash/conf.d/20-filter.conf*
- */etc/logstash/conf.d/30-output.conf*

Содержимое */etc/logstash/conf.d/10-input.conf*:

```
input {
  beats {
    include_codec_tag => false
    port => "5044"
  }
}
```

Содержимое */etc/logstash/conf.d/20-filter.conf*:

```
# Add "file.name" field.
filter {
  grok {
    match => { "source" => "%{UNIXPATH}/%{NOTSPACE:[file][name]}" }
  }
}
```

Содержимое */etc/logstash/conf.d/30-output.conf*:

```
output {
  elasticsearch {
    hosts => "localhost:9200"
    manage_template => false
    index => "%{[@metadata][beat]}-%{[@metadata][version]}-%{+YYYY.MM.dd}"
  }
}
```

Документация по настройке SSL: https://www.elastic.co/guide/en/beats/filebeat/current/configuring-ssl-logstash.html

Захват логов из платформы

Формат логов в платформе имеет следующую структуру:

```
[module] level timestamp file message
```

Пример:

```
[tariffcontroller] LOG 2018/07/02 14:00:00 report_processing.go:30: Awake report process on: 2018-07-02 14:00:
00.000230175 +0000 UTC m=+250432.1215933
[job] WARN 2018/07/02 14:00:15 segment.go:391: Segment counting time: 57.995838503s
```

Настройка паттернов в logstash для захвата логов

1) Поместите следующие паттерны в отдельный файл, например – */etc/logstash/patterns/akd*:

```
AKD_MODULE [\w_]+
AKD_LEVEL (LOG|Log|[Aa]lert|ALERT|[Tt]race|TRACE|[Dd]ebug|DEBUG|[Nn]otice|NOTICE|[Ii]nfo|INFO|[Ww]arn?(?:ing)?
|WARN?(?:ING)?|[Ee]rr?(?:or)?|ERR?(?:OR)?|[Cc]rit?(?:ical)?|CRIT?(?:ICAL)?|[Ff]atal|FATAL|[Ss]evere|SEVERE|EMERG
(?:ENCY)?|[Ee]merg(?:ency)?)
AKD_TIMESTAMP %{YEAR}/%{MONTHNUM}/%{MONTHDAY} %{TIME}
AKD_FILE [\w.:]+
AKD_MESSAGE %{GREEDYDATA}
AKD_LOG \[%{AKD_MODULE:akd_module}\] %{AKD_LEVEL:akd_level} %{AKD_TIMESTAMP:akd_timestamp} %{AKD_FILE:akd_file}
%{AKD_MESSAGE:akd_message}
AKD_MULTILINE_LOG (?<akd_level>[\w\s]+): (?<akd_message>[\w\s\'\".:]+)
```

2) Настройте фильтр

```
filter {
  if [application] == "akd" {
    if "runtime error:" in [message] or "panic:" in [message] {
      grok {
        match => { "message" => "%{AKD_MULTILINE_LOG}" }
      }
      mutate {
        rename => { "akd_level" => "[akd][level]" }
        rename => { "akd_message" => "[akd][message]" }
      }
    } else {
        grok {
          patterns_dir => [ "/etc/logstash/patterns" ]
          match => { "message" => "%{AKD_LOG}" }
        }
        mutate {
          rename => { "akd_module" => "[akd][module]" }
          rename => { "akd_level" => "[akd][level]" }
          rename => { "akd_timestamp" => "[akd]][timestamp]" }
          rename => { "akd_file" => "[akd][file]" }
          rename => { "akd_message" => "[akd][message]" }
      }
    }
  }
}
```

Захват логов из модуля отправки AKMTA

Формат логов в AKMTA имеет следующую структуру:

```
 : timestamp
'info','message_id','sender_id','ISP','sender_ip','from_email','from_domain','recipient','mx_ip','mx_host','resp
onse_status response_message','command'
```

Пример:

```
Jul  2 13:51:22 DS5052 AKMTA-RESPONSES[1987]: 2018/07/02 13:51:22
'5|0|Hotmail|153d914326396fac','w4y4rWfULx6t_2_4v_t_5__6.
2HRfsoKxfZX5eqUSz_5rCL1w','5','Hotmail','10.10.10.10','noreply@example.com','sender.example.com','john@live.
fr','104.47.8.33','eur.olc.protection.outlook.com.','550 5.5.0 Requested action not taken: mailbox unavailable.
[AM5EUR03FT004.eop-EUR03.prod.protection.outlook.com]','RCPT TO'
Jul  2 14:00:45 DS5052 AKMTA-RESPONSES[1987]: 2018/07/02 14:00:45
'2|0|iCloud|153d920d3c347472','w4y4sk4hysyQ_2_5o_2q_5__3.
2HRfuXT3rgE7xYXQR_3QccWj','2','iCloud','10.10.10.10','noreply@example.com','sender.example.com','john@icloud.
com','17.57.8.137','mx4.mail.icloud.com.','550 5.7.1 [CS01] Message rejected due to local policy. Please visit
https://support.apple.com/en-us/HT204137','.CRLF'
```

Настройка паттернов в logstash для захвата логов

1) Поместите следующие паттерны в отдельный файл, например – */etc/logstash/patterns/akmta*:

```
AKMTA_EMAIL_ADDRESS [a-zA-Z0-9_.+-=:]+@%{HOSTNAME}
AKMTA_TIMESTAMP %{YEAR}/%{MONTHNUM}/%{MONTHDAY}%{SPACE}%{TIME}
AKMTA_INFO [\w\s\|.-]+
AKMTA_MESSAGE_ID [\w_.-]+
AKMTA_SENDER_ID %{NUMBER}
AKMTA_ISP [\w\s_.-]+
AKMTA_SENDER_IP %{IP}
AKMTA_FROM_EMAIL %{AKMTA_EMAIL_ADDRESS}
AKMTA_FROM_DOMAIN %{HOSTNAME}
AKMTA_RECIPIENT %{AKMTA_EMAIL_ADDRESS}
AKMTA_MX_IP (%{IP}|[<\w>]+)?
AKMTA_MX_HOST %{HOSTNAME}?
AKMTA_RESPONSE_MESSAGE %{GREEDYDATA}
AKMTA_RESPONSE_STATUS \w+
AKMTA_COMMAND [\w\s._-]+
AKMTA_LOG %{AKMTA_TIMESTAMP:akmta_timestamp} '%{AKMTA_INFO}','%{AKMTA_MESSAGE_ID:akmta_message_id}','%
{AKMTA_SENDER_ID:sender_id}','%{AKMTA_ISP:akmta_isp}','%{AKMTA_SENDER_IP:akmta_sender_ip}','%
{AKMTA_EMAIL_ADDRESS:akmta_from_email}','%{AKMTA_FROM_DOMAIN:akmta_from_domain}','%{AKMTA_RECIPIENT:
akmta_recipient}','%{AKMTA_MX_IP:akmta_mx_ip}','%{AKMTA_MX_HOST:akmta_mx_host}','%{AKMTA_RESPONSE_STATUS:
akmta_response_status} %{AKMTA_RESPONSE_MESSAGE:akmta_response_message}','%{AKMTA_COMMAND:akmta_command}'
```

2) Настройте фильтр

```
filter {
  if [application] == "akmta" {
    grok {
      patterns_dir => [ "/etc/logstash/patterns" ]
      match => { "message" => "%{AKMTA_LOG}" }
      match => { "akmta_response_message" => "%{AKMTA_RESPONSE_STATUS:akmta_response_status}" }
    }
    mutate {
      rename => { "akmta_timestamp" => "[akmta][timestamp]" }
      rename => { "akmta_message_id" => "[akmta][message][id]" }
      rename => { "akmta_sender_id" => "[akmta][sender][id]" }
      rename => { "akmta_isp" => "[akmta][isp]" }
      rename => { "akmta_sender_ip" => "[akmta][sender][ip]" }
      rename => { "akmta_from_email" => "[akmta][from][email]" }
      rename => { "akmta_from_domain" => "[akmta][from][domain]" }
      rename => { "akmta_recipient" => "[akmta][recipient]" }
      rename => { "akmta_mx_ip" => "[akmta][mx][ip]" }
      rename => { "akmta_mx_host" => "[akmta][mx][host]" }
      rename => { "akmta_response_status" => "[akmta][response][status]" }
      rename => { "akmta_response_message" => "[akmta][response][message]" }
      rename => { "akmta_command" => "[akmta][command]" }
    }
  }
}
```

Конфигурирование Filebeat

Документация: https://www.elastic.co/guide/en/beats/filebeat/current/index.html

После установки рекомендуем удалить конфигурацию по умолчанию и выделить отдельную папку под входные данные, например – */etc/filebeat /inputs.d*.

В главной конфигурации – */etc/filebeat/filebeat.yml*, добавьте секции *Configuration* и *Output*:

```
# Configuration
filebeat.config:
  inputs:
    path: ${path.config}/inputs.d/*.yml
    reload.enabled: false
  modules:
    path: ${path.config}/modules.d/*.yml
    reload.enabled: false

# Output
output.logstash:
  hosts: ["<IP  host>:5044"]
```

Документация по настройке SSL: https://www.elastic.co/guide/en/beats/filebeat/current/configuring-ssl-logstash.html

Справка по filebeat.yml: https://www.elastic.co/guide/en/beats/filebeat/current/filebeat-reference-yml.html

Настройка передачи логов из платформы

Пример конфигурации входных данных – */etc/filebeat/inputs.d/akd.yml*:

```
- type: log
  enabled: true
  paths:
    - /opt/akd/logs/*.log
  fields:
    application: akd
  fields_under_root: true
  tail_files: true
  exclude_lines: ['.*DEBUG.*']
  multiline.pattern: '^\[.*\]|^runtime error:|^panic:'
  multiline.negate: true
  multiline.match: after
```

Настройка передачи логов из модуля отправки AKMTA

Пример конфигурации входных данных – */etc/filebeat/inputs.d/akmta.yml*:

```
- type: log
  paths:
    - /var/log/syslog
  include_lines: ['AKMTA-RESPONSES']
  fields:
    application: akmta
  fields_under_root: true
  tail_files: true
```

Конфигурирование Kibana

Документация: https://www.elastic.co/guide/en/kibana/current/index.html

Ограничить доступ к Kibana можно с помощью NGINX и htpasswd:

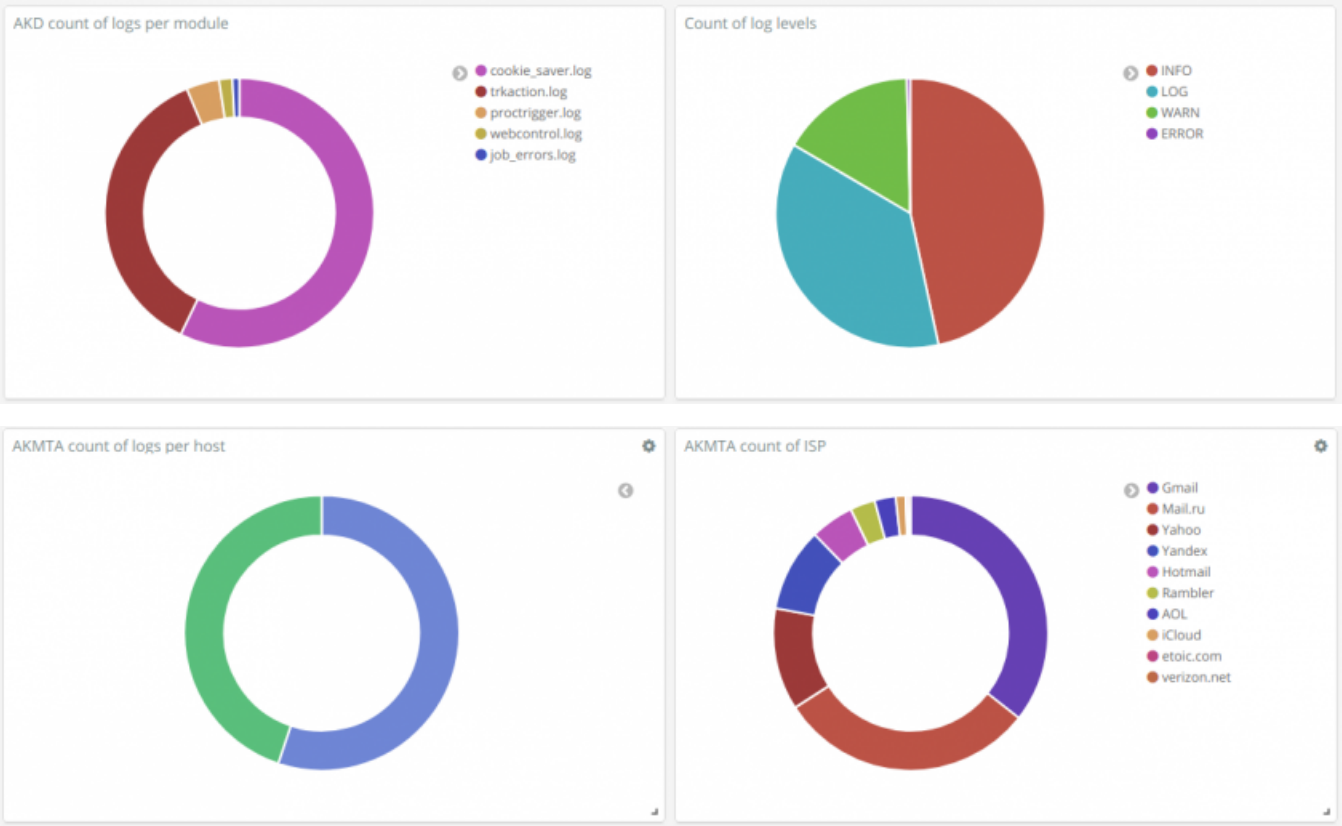
Создайте пару пользователь/пароль:

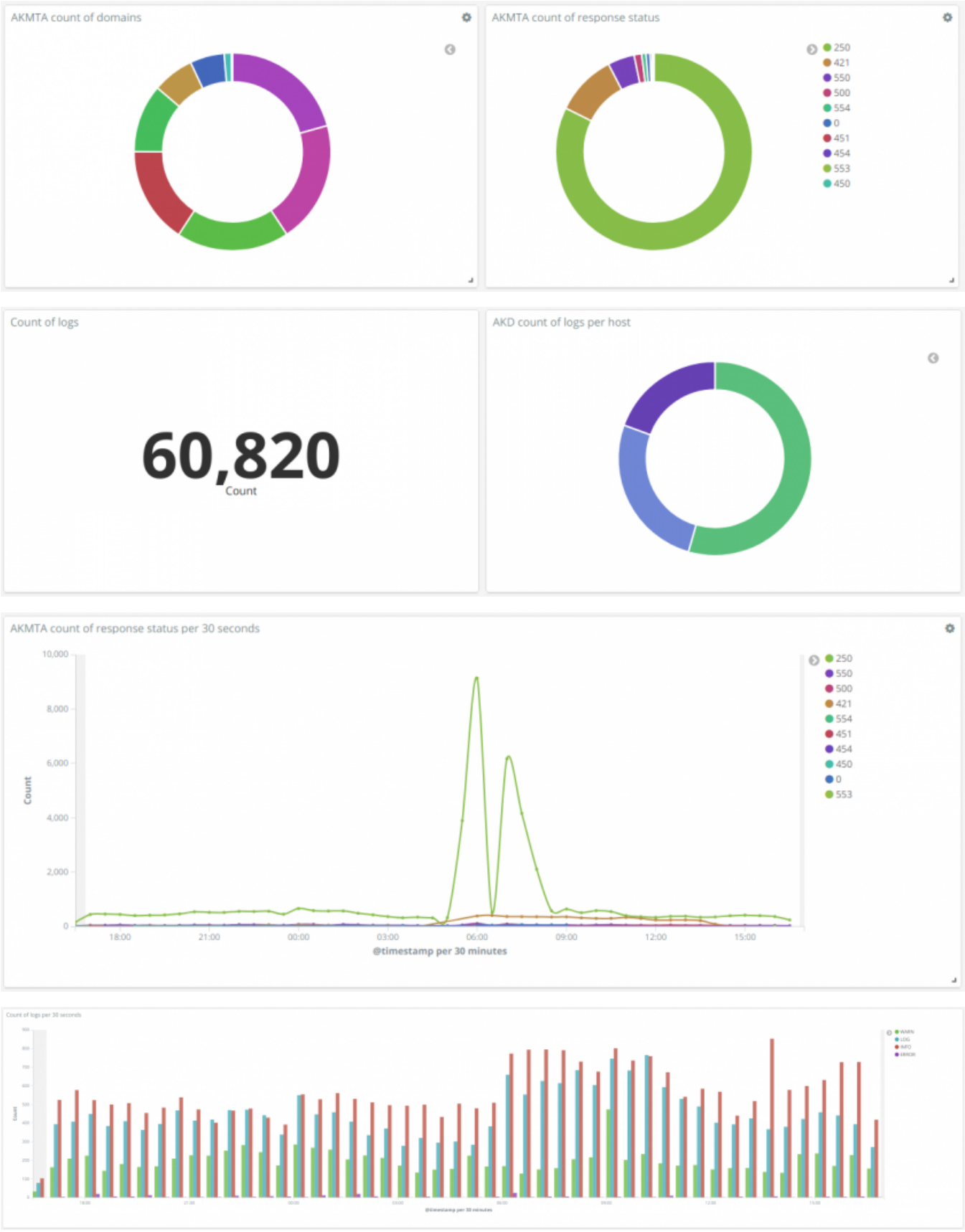
```
htpasswd /etc/nginx/.htpasswd kibana
```

Пример конфигурации NGINX:

```
server {
  listen 80;
  listen 443 ssl;
  server_name kibana.example.com;

  # Redirect non-https traffic to https
  if ($scheme != "https") {
    return 301 https://$host$request_uri;
  }

  ssl_certificate /etc/letsencrypt/live/kibana.example.com/fullchain.pem;
  ssl_certificate_key /etc/letsencrypt/live/kibana.example.com/privkey.pem;
  include /etc/letsencrypt/options-ssl-nginx.conf;
  ssl_dhparam /etc/letsencrypt/ssl-dhparams.pem;

  # allow ...;
  # deny all;

  access_log /var/log/nginx/kibana.example.com.access.log;
  error_log /var/log/nginx/kibana.example.com.error.log;

  location / {
    auth_basic "Restricted";
    auth_basic_user_file /etc/nginx/.htpasswd;
    proxy_pass http://localhost:5601;
  }
}
```

Добавляйте любые на ваш вкус графики в Kibana:





Конфигурирование Elastalert

Документация: http://elastalert.readthedocs.io/en/latest/

1) После установки, рекомендуем создать отдельную папку для хранения конфигураций и правил, например:

- */etc/elastalert*
- */etc/elastalert/rules*

2) Создайте индексы с помощью команды – *elastalert-create-index*.

3) Создайте файл главной конфигурации, например:

```
# /etc/elastalert/config.yaml
# This is the folder that contains the rule yaml files
# Any .yaml file will be loaded as a rule
rules_folder: "/etc/elastalert/rules"

# How often ElastAlert will query Elasticsearch
# The unit can be anything from weeks to seconds
run_every:
  minutes: 1

# ElastAlert will buffer results from the most recent
# period of time, in case some log sources are not in real time
buffer_time:
  minutes: 15

# The Elasticsearch hostname for metadata writeback
# Note that every rule can have its own Elasticsearch host
es_host: 127.0.0.1

# The Elasticsearch port
es_port: 9200

# The index on es_host which is used for metadata storage
# This can be a unmapped index, but it is recommended that you run
# elastalert-create-index to set a mapping
writeback_index: elastalert

# If an alert fails for some reason, ElastAlert will retry
# sending the alert until this time period has elapsed
alert_time_limit:
  days: 2
```

4) Создайте правило, например:

```
# /etc/elastalert/rules/akd_to_dev.yaml
# Rule name, must be unique
name: AKD application logs to Dev channel

# Type of alert.
type: any

# Index to search, wildcard supported
index: filebeat-*

# Ignore repeating alerts for a period of time
realert:
  minutes: 0

# A list of Elasticsearch filters used for find events
# These filters are joined with AND and nested in a filtered query
filter:
- query:
    query_string:
      query: 'application: akd AND (akd.level: panic OR akd.level: "runtime error")'

# The alert is use when a match is found
alert:
- "telegram"
alert_subject: "<subject>"
telegram_room_id: "<id>"
telegram_bot_token: "<token>"
# Example SOCKS5 proxy
# telegram_proxy: "socks5://username:password@host:port"

# The body text of the various types of events
alert_text_type: alert_text_only
alert_text: |
  Host: {0}
  Level: {1}
  Source: {2}
  Message:
  {3}
alert_text_args: ["host.name", "log.level", "source", "message"]
```

5) Для запуска используйте команду:

```
python -m elastalert.elastalert --verbose --config /etc/elastalert/config.yaml
```

Если вы используете systemd, добавьте сервис – */lib/systemd/system/elastalert.service*:

```
[Unit]
Description=elastalert
After=multi-user.target

[Service]
Type=simple
WorkingDirectory=/etc/elastalert
ExecStart=/usr/local/bin/elastalert
StandardOutput=syslog
StandardError=syslog
Restart=on-failure

[Install]
WantedBy=multi-user.target
```

Миграция данных RabbitMQ на новый сервер

Если ваш RabbitMQ начинает потреблять значительное количество ресурсов вашего сервера (CPU или память), возможно его стоит вынести на отдельный сервер.

> Сервер для RabbitMQ желательно разместить рядом с управляющем сервером, для избежания задержек.

Чтобы не оставлять сообщения на старом сервере, это руководство предлагает миграцию данных RabbitMQ с помощью Shovel Plugin.

Шаги

1 Новый сервер RabbitMQ

Создайте виртуальные хосты и установите Management Plugin.

```
#!/usr/bin/env bash

user="rabbit"
password="password"

# Configure
rabbitmqctl add_user "$user" "$password"
rabbitmqctl set_user_tags "$user" administrator
rabbitmqctl set_permissions -p / "$user" ".*" ".*" ".*"

vhosts=(
  "akmta_events"
  "akmta_senders"
  "akmta_stat"
  "amazon_sns_events"
  "dmta_events"
  "dmta_pools"
  "dmta_stat"
  "emaildirect_events"
  "eshark_events"
  "http_api_events"
  "mandrill_events"
  "wz_events"
)

for vhost in "${vhosts[@]}"; do
  rabbitmqctl add_vhost "$vhost"
  rabbitmqctl set_permissions -p "$vhost" "$user" ".*" ".*" ".*"
done

rabbitmq-plugins enable rabbitmq_management
systemctl enable rabbitmq-server.service
systemctl restart rabbitmq-server.service
```

2 Управляющий сервер

Измените данные для подключения к RabbitMQ в файле конфигурации `main.json` и перезапустите AKD.

```
"RABBITMQ_HOST": "192.168.*.*",
"RABBITMQ_PASS": "abcdefghijklmnqrstuvwxyzABCDEFGH",
"RABBITMQ_USER": "rabbit",
```

3 Старый сервер RabbitMQ

Установите Management Plugin и Shovel Plugin.

```
rabbitmq-plugins enable rabbitmq_management
rabbitmq-plugins enable rabbitmq_shovel
rabbitmq-plugins enable rabbitmq_shovel_management
```

В веб-интерфейс RabbitMQ перейдите в раздел "`Admin`" – "`Shovel Management`", добавьте новые лопаты для всех очередей, в которых остались сообщения.

```
Name: deliv
Source: amqp://rabbit:password@rabbitmq-1:5672/akmta_events
Source queue: deliv
Destination: amqp://rabbit:password@rabbitmq-2:5672/akmta_events
Destination queue: deliv

Name: undeliv
Source: amqp://rabbit:password@rabbitmq-1:5672/akmta_events
Source queue: undeliv
Destination: amqp://rabbit:password@rabbitmq-2:5672/akmta_events
Destination queue: undeliv

...
```

Если вы совершаете импорт напрямую в RabbitMQ, добавьте лопату для очередей "database_import" и "database_import_results".

Удалить лопаты можно следующим образом:

```
rabbitmqctl clear_parameter shovel -p "queue"
rabbitmqctl clear_parameter shovel -p "vhost" "queue"
```

Все сообщения в очередях, для которых добавлена лопата – будут уходить в новый RabbitMQ. Дождитесь завершения переноса сообщений и выключите старый инстанс.