OTUS

ОНЛАЙН-ОБРАЗОВАНИЕ

OTUS

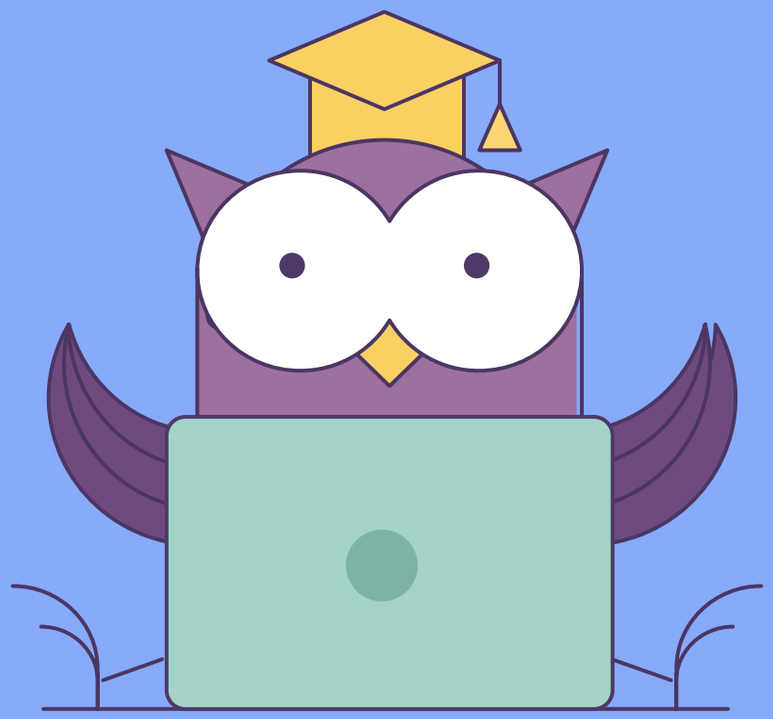
Сбор и анализ логов

Курс «Администратор Linux»

Занятие № 15

Алексей Цыкунов Альбрехт Леонид

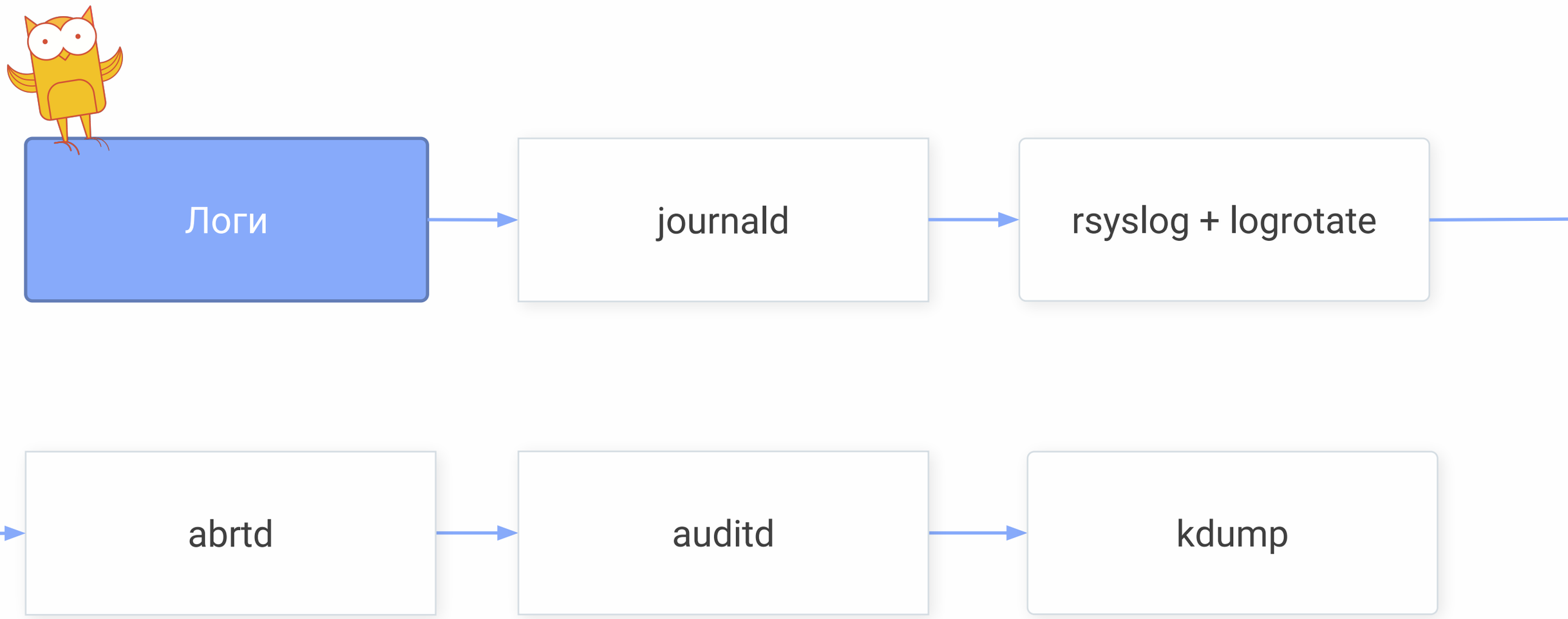
Меня хорошо слышно && видно?



Напишите в чат, если есть проблемы!

Ставьте + если все хорошо
Ставьте - если есть проблемы

Маршрут вебинара



Основные лог файлы

- Приложения
- События
- Службы
- Системные

/var/log

- **/var/log/syslog** или **/var/log/messages** - глобальный системный журнал
- **/var/log/auth.log** или **/var/log/secure** - информация об авторизации пользователей
- **/var/log/dmesg** - драйвера устройств

```
[root@logs ~]# dmesg -l err
[1131424.604352] logs kernel: end_request: I/O error, dev sdc, sector 48
[1131424.604352] logs kernel: Buffer I/O error on device sdc, logical block 6
[1131424.604352] logs kernel: Buffer I/O error on device sdc, logical block 7
```

- **/var/log/anaconda.log** — лог установки системы
- **/var/log/audit** — лог демона auditd.
- **/var/log/boot.log** — лог загрузки системы
- **/var/log/cron** — лог crond

Для каждого дистрибутива будет отдельный журнал менеджера пакетов:

- **/var/log/yum.log** — Для программ установленных с помощью Yum в RedHat Linux.
- /**var/log/emerge.log** — Для ebuild-ов установленных из Portage с помощью emerge в Gentoo Linux.
- **/var/log/dpkg.log** — Для программ установленных с помощью dpkg в Debian Linux и всем семействе родственных дистрибутивах.

/var/log

- **/var/log/mysql/** — Лог базы данных MySQL.
- **/var/log/httpd/** или /var/log/apache2/ — Лог веб сервера Apache, журнал доступа находится в access_log, а ошибки — в error_log.
- **/var/log/nginx/** - логи NGINX-а

Полезные утилиты

- Inav
- tail -f
- cat/less
- zcat

Ротация логов

Logrotate - подсистема для автоматической ротации логов, с возможность сжатия, удаления, перемещения и гибкой настройкой под каждый вид лог файлов

Первое что надо настроить, как часто будет идти проверка совпадения условиям:

- **hourly** - каждый час
- **daily** - каждый день
- **weekly** - каждую неделю
- **monthly** - каждый месяц
- **yearly** - каждый год

logrotate: условия

- **rotate** - указывает сколько старых логов нужно хранить, в параметрах передается количество;
- **dateext** - добавляет дату ротации перед именем лога
- **compress/delaycompress** - сжатие лога и отсрочка сжатия
- **mail** - отправлять Email
- **missingok** - не выдавать ошибки, если лог файла не существует
- **sharedscripts** - запускает скрипт только один раз
- **postrotate/endscript** - запуск скрипта или произвольной команды
- **maxsize** - ротация по размеру файла

logrotate: пример

OTUS

```
[root@logs ~]# cat /etc/logrotate.d/nginx
/var/log/nginx/*.log {
        daily
        missingok
        rotate 52
        compress
        delaycompress
        notifempty
        create 0640 www-data adm
        sharedscripts
        prerotate
                if [ -d /etc/logrotate.d/httpd-prerotate ]; then \
                        run-parts /etc/logrotate.d/httpd-prerotate; \
                fi \
        endscript
        postrotate
                invoke-rc.d nginx rotate >/dev/null 2>&1
        endscript
}
```

Journald: ключевые особенности

- Бинарный формат (защита от подделок, возможность конвертации в другие форматы)
- Не требует специальной настройки
- Структурированные данные (multi-field, multi-line)
- Индексированные данные
- Центральное хранилище логов

Journald: прием логов

- Простые syslog логи
- Логи ядра (kmsg)
- Структурированные данные через Journal API
- Логи и статусы systemd юнитов
- Записи системы аудита

- Сервис - systemd-journald
- /etc/systemd/journald.conf
- /var/log/journal/*
- По дефолту чтение доступно для группы systemd-journald

Journald: конфигурация

- Storage: (volatile, persistent, auto, none) - по умолчанию умеет писать в файл но автоматом не создает директорию
- Compress (yes, no)
- Seal=(yes, no) - накладывает криптографическую печать
- SplitMode (uid, none) - разделение журнальных файлов, по умолчанию uid (для каждого юзера свой журнальный файл)
- RateLimitInterval=, RateLimitBurst= регулировка скорости обработки

Journald: конфигурация

- MaxFileSec - период ротации
- MaxRetentionSec - как долго хранить логи
- SyncIntervalSec - интервал синхронизации журнала с файлами логов
- ForwardToSyslog, ForwardToKMsg, ForwardToConsole, ForwardToWall
  - перенаправление сообщений
- MaxLevelStore=, MaxLevelSyslog=, MaxLevelKMsg=, MaxLevelConsole=, MaxLevelWall=
  - "emerg", "alert", "crit", "err", "warning", "notice", "info", "debug", 0-7

Journald: конфигурация

- SystemMaxUse= максимальный объём, который логи могут занимать на диске;
- SystemKeepFree= объём свободного места, которое должно оставаться на диске после сохранения логов;
- SystemMaxFileSize= объём файла лога, по достижении которого он должен быть удален с диска;
- RuntimeMaxUse= максимальный объём, который логи могут занимать в файловой системе /run;
- RuntimeKeepFree= объём свободного места, которое должно оставаться в файловой системе /run после сохранения логов;
- RuntimeMaxFileSize= объём файла лога, по достижении которого он должен быть удален из файловой системы /run.

journalctl: чтение логов

- journalctl -xe -o json-pretty
- journalctl --list-boots
- journalctl -b -2
- journalctl --since "2017-05-05 00:01" --until "2017-05-06 01:40"
    - YYYY-MM-DD HH:MM:SS
- journalctl --since "10 hours ago"
- journactl -u mysqld.service -f
- journalctl _UID=1001
- journalctl -n 3 -p crit

journalctl: управление

- journalctl --flush
- journalctl --rotate
- journalctl --sync
- journalctl --vacuum-size=1G
- journalctl --vacuum-time=1years
- journalctl --disk-usage

journald: централизованное хранение

В systemd предусмотрены специальные компоненты для решения этой задачи: systemd-journal-remote, systemd-journal-upload и systemd-journal-gatewayd.

С помощью команды systemd-journal-remote можно принимать логи с удалённых хостов и сохранять их (на каждом их этих хостов должен быть запущен демон systemd-journal-gatewayd), например:

```
[root@logs ~]# systemd-journal-remote --url https://some.host:19531/
```

journald: централизованное хранение

В результате выполнения приведённой команды логи с хоста some.host будут сохранены в директории var/log/journal/some.host/remote-some~host.journal.

С помощью команды systemd-journal-remote можно также складывать имеющиеся на локальной машине логи в отдельную директорию, например:

```
[root@logs ~]# journalctl -o export | systemd-journal-remote -o /tmp/dir -
```

Команда systemd-journal-upload используется для загрузки логов с локальной машины в удалённое хранилище:

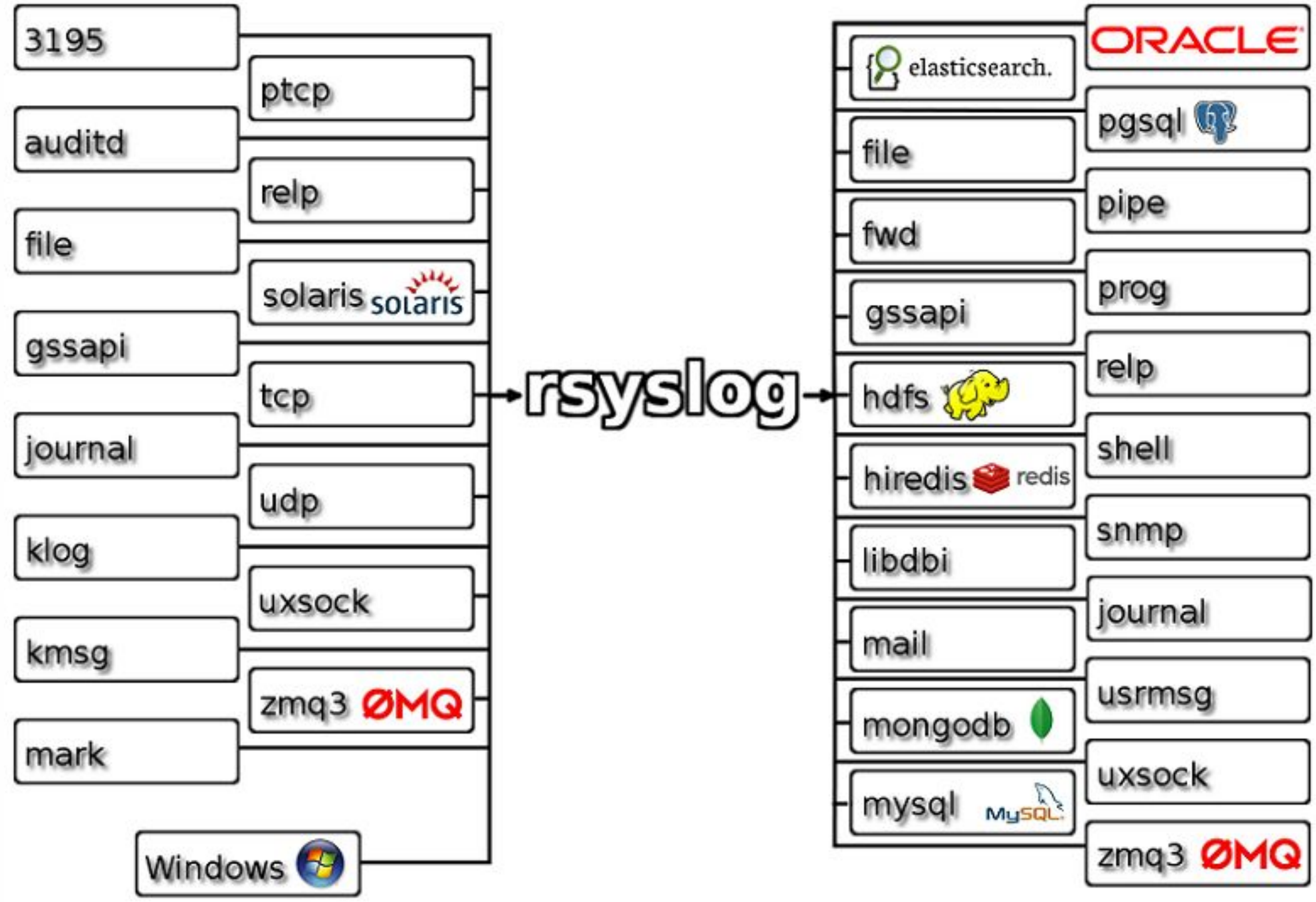
```
[root@logs ~]# systemd-journal-upload --url https://logserver:19531/
```

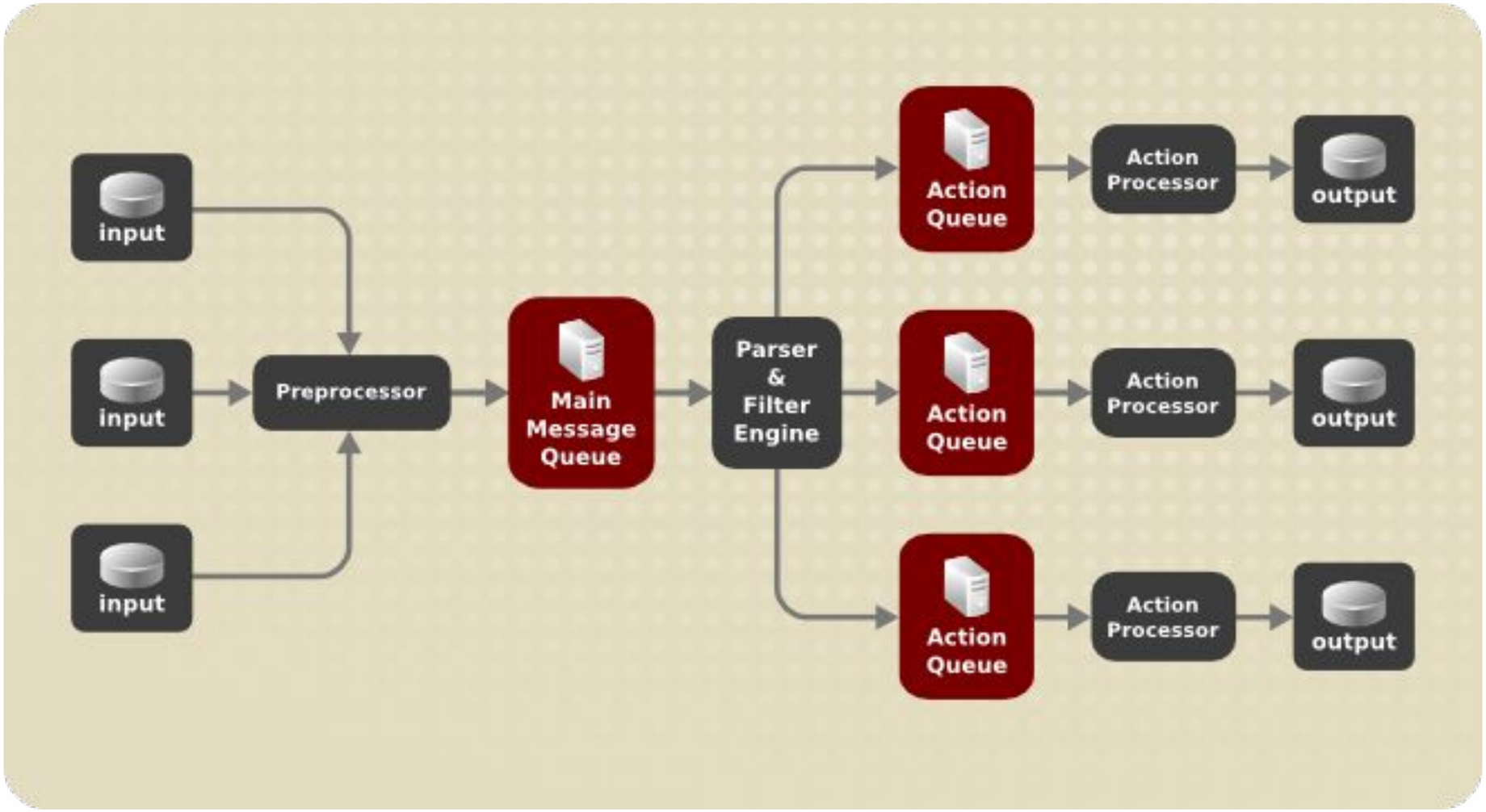

rsyslog

rsyslog

Rsyslog - **R**ocket-fast **Sys**tem for **log** processing.

- Многопоточный
- TCP, SSL, TLS, RELP
- сохранение логов в MySQL, PostgreSQL, Oracle
- Фильтрация по любой части лога
- Полностью настраиваемый формат вывода


3195
auditd
file
gssapi
journal
klog
kmsg
mark
Windows
ptcp
relp
solaris
tcp
udp
uxsock
zmq3
rsyslog
elasticsearch.
file
fwd
gssapi
hdfs
hiredis
redis
libdbi
mail
mongodb
mysql
ORACLE
pgsql
pipe
prog
relp
shell
snmp
journal
usrmsg
uxsock
zmq3


input
input
input
Preprocessor
Main
Message
Queue
Parser
&
Filter
Engine
Action
Queue
Action
Queue
Action
Processor
Action
Processor
output
output

rsyslog: модули

- Output Modules (omelasticsearch, omfile, ommysql,)
- Input Modules (imtcp, imjournal, iudp, imrelp, ...)
- Parser Modules (pmciscoios, pmlastmsg ...)
- Message Modification Modules (mmcount, mmfields ...)
- String Generator Modules (smfile, smfwd, smtradfile, smfwd)

rsyslog: наборы правил

Ruleset - содержит список правил, правило состоит из фильтра и привязанных к нему одного или нескольких Actions

Можно задавать несколько rulesets, тем самым разделяя обработку различных сообщений

rsyslog: фильтры

OTUS

- **FACILITY** - маркировка сообщений по типу: kern(0), user (1), mail (2), daemon (3), auth (4), syslog (5), lpr (6), news (7), uucp (8), cron (9), authpriv (10), ftp (11), and local0 through local7 (16 - 23).
- **PRIORITY** - маркировка сообщений по важности: debug (7), info (6), notice (5), warning (4), err (3), crit (2), alert (1), and emerg (0)

Фильтрация сообщений по свойствам	
:PROPERTY, [!]COMPARE_OPERATION, "STRING"	
• Contains • Isequal • Startwith	• Regex • Ereregex • Isempty
• :msg, !regex, "fatal .* error" • :hostname, isequal, "host1"	

rsyslog: действия

- **Actions** - описывают то, что нужно сделать с сообщением

- Для действий есть следующие типы параметров
  - Общие для всех действий
  - Специфичные для конкретного действия, определяемые модулем
  - Специфичные для конкретной очереди

rsyslog: шаблоны

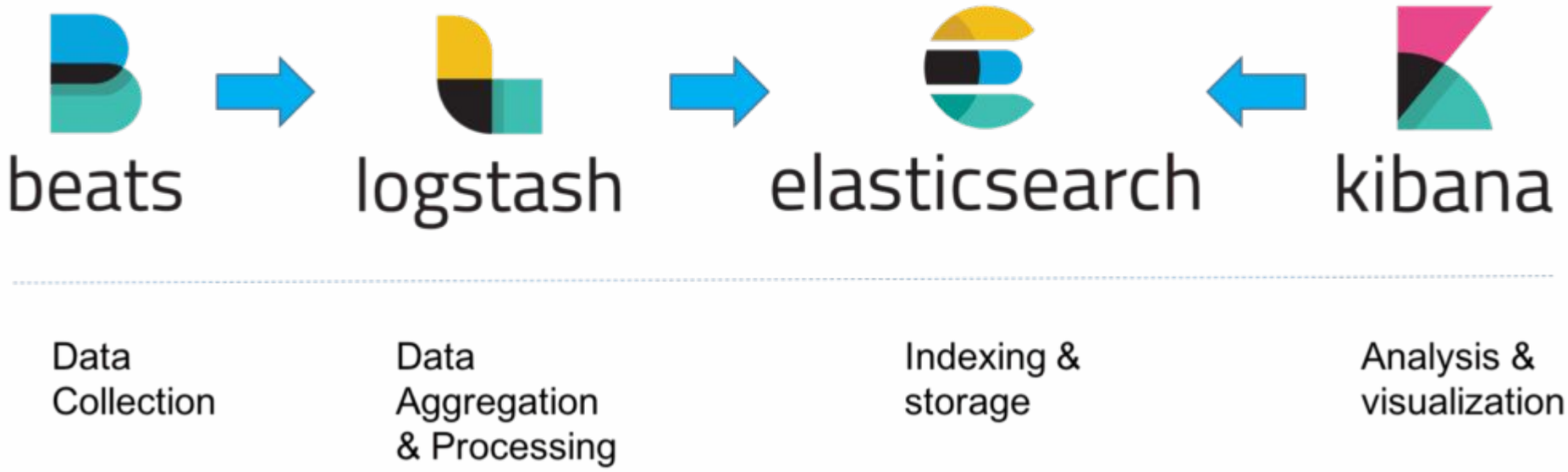
- **Templates** - Ключевая особенность rsyslogd. Позволяют задавать любые шаблоны для динамических имен файлов, содержимого на входе и выходе, SQL для баз данных итд.
- Типы шаблонов: list, subtree, string, plugin

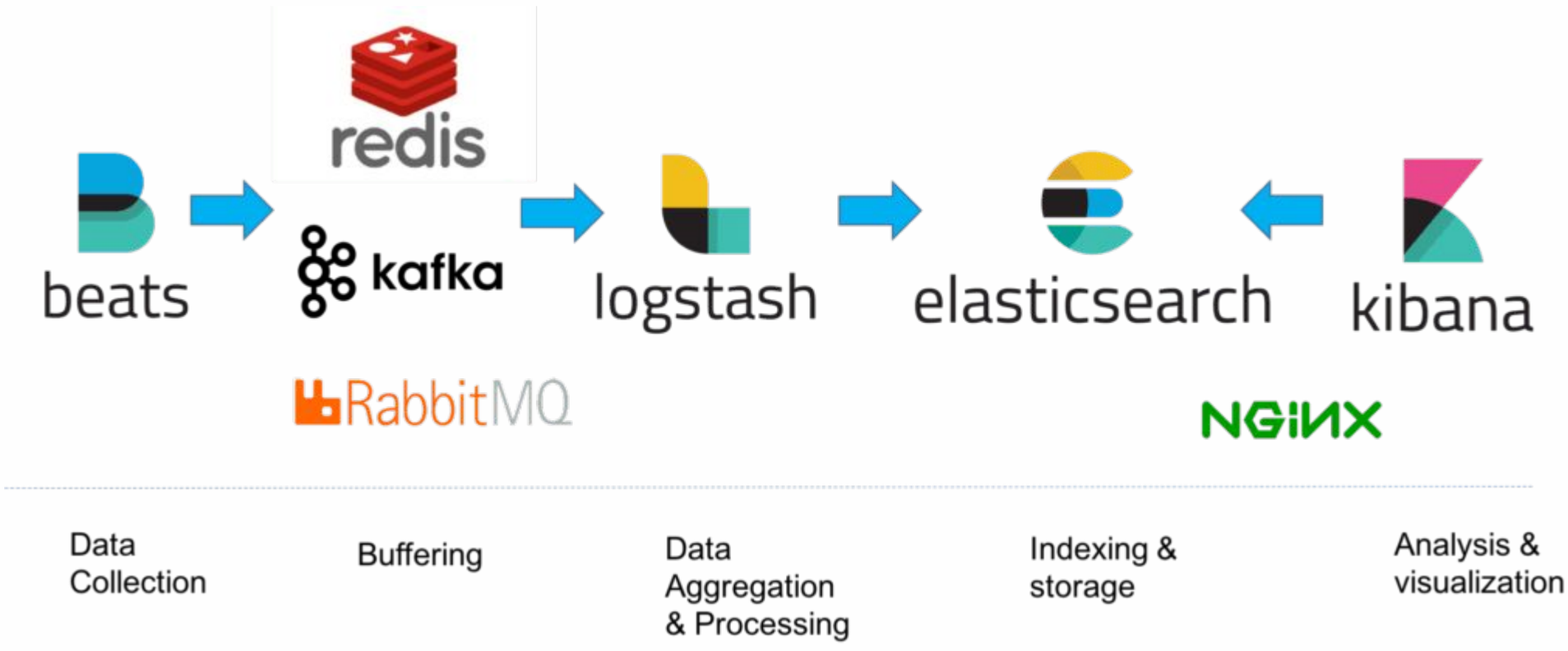
rsyslog: очереди

- Direct queues
- Disk queues
- In-memory queues
- Disk-Assisted Memory queues

Stack

Arch


beats
logstash
elasticsearch
kibana
Data Collection
Data Aggregation & Processing
Indexing & storage
Analysis & visualization


redis
beats
kafka
logstash
elasticsearch
kibana
RabbitMQ
NGINX
Data Collection
Buffering
Data Aggregation & Processing
Indexing & storage
Analysis & visualization

Filebeat

- Сервис для передачи логов приложений на Logstash
- Кроме этого, также доступны Winlogbeat (события Windows Event Logs), Metricbeat (метрики), Packetbeat (сетевая информация) и Heartbeat (uptime)

Logstash

- **Logstash -** сервис для сбора логов и отправки их в Elasticsearch. В самой простой конфигурации можно обойтись без него и отправлять логи напрямую в еластик. Но с logstash это делать удобнее.
- Типичная конфигурация logstash представляет из себя несколько входящих потоков информации (input), несколько фильтров для этой информации (filter) и несколько исходящих потоков (output).

- **Elasticsearch** - используется для хранения, анализа, поиска по логам.
- Изначально, **elasticsearch** – это решение для полнотекстового поиска, построенное поверх Apache Lucene, но с дополнительными удобствами, типа лёгкого масштабирования, репликации.

Kibana

- **Kibana -** позволяет искать/брать данные в Elasticsearch и визуализировать их с помощью графиков, таблиц, карт.
- **Kibana** бесполезна без elasticsearch.

auditd

auditd - это прикладной компонент системы аудита Linux. Он ведёт протокол аудита на диске. Для просмотра протоколов предназначены команды ausearch и aureport. Команда auditctl позволяет настраивать правила аудита. Кроме того, при загрузке загружаются правила из файла /etc/audit.rules.

Некоторые параметры самого демона можно изменить в файле auditd.conf.

auditd

- auditctl -a entry,always -S all -F pid=1005
  - Все системные вызовы с конкретного PID

- auditctl -a exit,always -S open -F auid=510
  - Все открытые файлы конкретным пользователем

- auditctl -a exit,always -S open -F success!=0
  - Все неудачные попытки открытия файлов

- auditctl -w /home/user/test_dir/ -k test_watch
  - Наблюдение за директорий

- auditctl -w /sbin/insmod -p x -k module_insertion
  - Наблюдение за вызовом insmod

auditd

Для диагностики и анализа причин сбоев ядра разработчиками компании RedHat был разработан специализированный инструмент — **kdump**.

Принцип его работы можно кратко описать следующим образом. Создается два ядра: основное и аварийное (именно оно используется для сбора дампа памяти). При загрузке основного ядра под аварийное ядро выделяется определенный размер памяти. При помощи kexec во время паники основного ядра загружается аварийное и собирает дамп. Анализ сбоя анализируется утилитой crash

abrtd

Инструмент для создания отчетов об ошибках. Активизируется при сбоях приложений и собирает дополнительные данные для последующего анализа

- **abrtd**, системный демон,
- **abrt-applet** - аплет работающий в области уведомлений пользователей,
- **abrt-gui** - графический интерфейс, который показывает собранные аварии и позволяет редактировать, сообщать о них и удалить их,
- **abrt-cli** - интерфейс командной строки

Домашнее задание

OTUS

В вагранте поднимаем 2 машины web и log
на web поднимаем nginx
на log настраиваем центральный лог сервер на любой системе на выбор
- journald
- rsyslog
- elk

настраиваем аудит следящий за изменением конфигов nginx

все критичные логи с web должны собираться и локально и удаленно
все логи с nginx должны уходить на удаленный сервер (локально только критичные)
логи аудита уходят ТОЛЬКО на удаленную систему

* развернуть еще машину elk
И таким образом настроить 2 центральных лог системы: elk и какую либо еще
В elk должны уходить только логи NGINX-а
Во вторую систему все остальное

Ваши вопросы?

Заполните, пожалуйста, опрос в ЛК о занятии

Спасибо
за внимание!
До встречи в Slack и на вебинаре