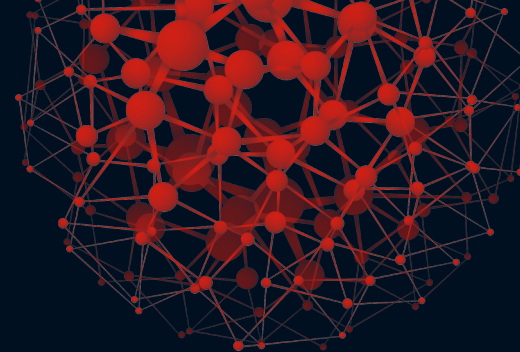


РЕД
СЛЁРМ

+



Southbridge



Самописные скрипты мониторинга. Обзор проверок, работа с ними



Евгений Терешков

Инженер компании Southbridge

План

- Введение
- Некоторые выдающиеся проверки
- Другие проверки

Введение

- Некоторые вещи хочется проверять регулярно
- У всех, проактивно
- Сложные вещи, которые не сильно укладываются в мониторинг

АНАТОМИЯ

- `/srv/southbridge/bin` - скрипты
- `/srv/southbridge/scripts` - групповые скрипты для SCM
- `/srv/southbridge/check` - проверки
- `/srv/southbridge/etc` - настройки
- `/etc/cron.d/_sb_*` - задания

Некоторые выдающиеся проверки



Highload Report

- Запускается `monit-ом/zabbix-ом`
- Собирает данные о:
 - LA/CPU `usr/sys/iowait/steal`
 - посетителях/IP
 - URI
 - блокировках сессий PHP
 - запросах MySQL/PSQL, их статусе
 - потреблении памяти процессами
 - потреблении CPU
 - Внутреннем состоянии Nginx/Apache
 - соединениях
- Умеет аргументы `apache-stop/apache-start/force-restart/чистит семафоры`
- Результат отправляет почтой в задачу вида "HighLoad Report on `$HOSTNAME`"
- Наши действия: проанализировать отчёт, решить что делать (добавить ресурсов/изменить конфигурацию/...), починить
- [КБ](#), исходный код в [Gitlab](#)

RAID

- При первичке SCM устанавливает соответствующие скрипты (в `/srv/southbridge/bin` и крон). Требуется повесить роль (*raid* в Ansible и роли *aacraid/hpraid/megaraid/sas2raid* в Slack)
- Каждый час запускается проверка состояния RAID с выводом результатов в почту/тикет (`/srv/southbridge/bin/RAIDTYPE-check.sh`)
- Наши действия: меняем диски, ребилдим массив
- Для ручной проверки есть скрипты подробного статуса (`/srv/southbridge/bin/RAIDTYPE-status.sh`)

Документация, исходный код в [Gitlab](#)

Групповые отчёты

- Если нужно что-то разово сделать на всех серверах и вывести отчёт (общий или отдельные)
- Делаем MR в `gitlab.slurm.io:slack/centos.git`
- В `"roles/base/files/srv/southbridge/scripts/group_check"` кладём код установки флага/запуска кода для отсылки руту письма с темой `"GROUP_TEMA"` или `"HOST_TEMA"`
- В `"grouptask/TEMA"` пишем документацию к отчёту (в разметке Redmine)
- В `"grouptask/TEMA.tags"` пишем теги, через запятую
- На выходе имеем задачу "Групповой отчёт ТЕМА" (на все хосты) или задачи "ТЕМА on \$HOSTNAME" (на каждый хост)
- Наши действия при получении задачи: чинить на хостах подотчётных группе

[Документация](#)

Примеры групповых отчётов

- yum_list_security - обновления безопасности в yum([gitlab](#))
- autostart_service_check - забыли включить автозагрузку?
- repo-check - белый список репозиториев yum ([KB](#))
- database_backup_check - проверяем бэкапы
mysql/psql/mongo/redis/...
- sudo-check - получают sudo только добавленные через SCM ([KB](#),
[gitlab](#))

Другие проверки



Неполный список

- Открытые порты - сканирует порты "снаружи" ([КБ](#))
- `mysql-table-check.sh` - проверяем таблицы БД на повреждение
- `logsize_check` - не распухли ли логи в `/var/log`, `/home`, `/srv`
- `iptables_check/logrotate_config_check/mysql_config_check/nginx-apache-config-check.sh/postfix_config_check.sh/_sb_sudo_check.sh/_sb_sshd_check.sh` - валидны ли конфиги ?
- `maldet.sh` - отчёты по поиску малвари
- `_sb_rdiff-backup_check.sh` - проверка успешности бэкапов ([КБ](#))
- `_sb_rdiff-backup_check_exclude.sh` - проверка исключений бэкапов ([КБ](#))
- `_sb_systemd_unit_check.sh` - проверяем юниты systemd
- `_sb_vz_backup_config_check.sh` - напоминание о том что не бэкапим
- `kernel-crash-alert.sh` - ловим крахи ядра ([КБ](#))

Вопросы?

