Как готовить новый Performance Management

По многочисленным запросам в IRC, будем учиться настраивать нок для сбора данных и построения графиков.

Итак:

1) Необходимо запустить noc-pmwriter. Настраивать ничего не надо, конфиг по-умолчанию достаточно рабочий, но если что, то мой рабочий выглядит так:

```
[main]
logfile = /var/noc/log/noc-pmwriter.log
loglevel = info
logsize = 5000000
logfiles = 9
syslog_host =
pidfile = /var/noc/log/run/noc-pmwriter.pid

[line_listener]
enabled = true
listen = 0.0.0.0
port = 2003
```

Этот демон принимает данные от проб и записывает их в хранилище

2) Необходимо запустить noc-probe. Этот демон собственно и занимается опросом железок, поэтому его стоит размещать поближе к железкам.

Лично у меня возникла такая проблема что на цетральном сервере с NOC проблемы с генерацией большого количества мелких пакетов, поэтому запросы начали теряться, пришлось вынести все на отдельную машину с Linux, соответственно и конфиг стал соответствующим. И еще дополнение, лучше сразу запускать несколько инстансов, нагрузку noc-probe создает заметную, но и параллелится очень хорошо.

Принципиальное значение имеет секция autoconf, там указывается куда noc-probe демон будет подключаться для получения настроек и задач. Имя name определяет конфиг для какого пула будет забирать проба с центрального сервера

```
[autoconf]
name = default
url = http://noc/
user = noc               #     probe     noc-web
passwd = noc                          #     . user  passwd -        NOC     web-
timeout = 30
interval = 300
failed_interval = 10
```

3) Так как в предыдущем шаге встретили настройки для user/passwd, то логично их сейчас и настроить на центральном сервере, чтобы пользователь из настроек иметь достаточно прав для доступа к настройкам для сбора данных

Итак, надо просто создать нового пользователя в main > setup > users. Суперпользователя делать из него не надо из соображений безопасности, достаточно дать права на pm.probe: Config и Read (хотя на счет config я не уверен, возможно достаточно только read, не проверял)

4) продолжаем настройки на центральном сервере в web. Настройка storage. PM > setup > Storages

Name: default (на сколько я понял, очень рекомендуется обозвать его именно default)

Select Policy: Priority (у меня так, понятия не имею на что оно влияет и что будет если выбрать что-то другое)

Write Concern: 1 (если настроено несколько Collectors, то данный параметр задает в скольких из них будут писаться данные, например если задано два коллектора и write concern = 1, то данные будут отправляться в первый рабочий, если write concern = 2, то данные будут отправляться в оба коллектора)

Collectors: line|Y|127.0.0.1|2003|Y (настройка куда и как noc-probe будет отправлять собранные данные, как можно заметить протокол line соответсвует тому что был настроен в первом пункте для noc-pmwriter, тоже самое с номером порта, а вот адрес зависит, если запущены локальные пробы, то будет луппбек, если выносные то это должен быть ip сервера где запущен noc-pmwriter, на этот адрес пробы будут отсылать данные)

Access: graphite|Y|http://noc.example.com (я прописал просто днс имя сервера с ноком)

5) PM > setup > Probes. Настройка проб, если еще нет, то надо создать дефолтовую пробу с именем default.

Name: default

Active: Y

Storage: default (Хранилище из предыдущего пункта)

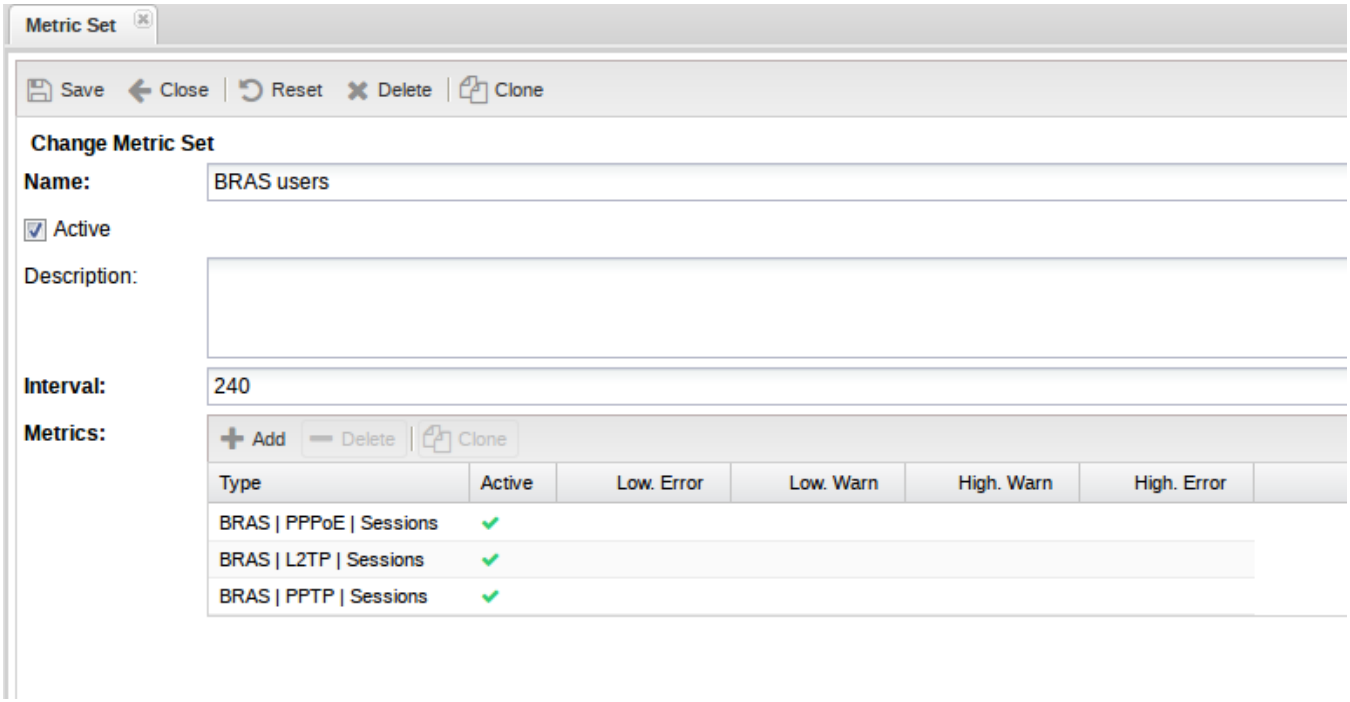
Credentials: noc (юзер который прописан в настройках noc-probe.conf)

Instances: 4 (количество запущенных инстансов noc-probe. Конфиг со списком задач для проб будет разбит на указанное число равных частей, каждый инстанс получит только свою часть, число должно точно соответствовать числу проб, если проб будет меньше, то часть данных не будет собираться, если больше - часть проб будут висеть в воздехе без конфига и ничего не делать)

Все, теперь можно начать назначать, какие данные собирать

6) PM > setup > Metric set

Метрики выбираются из списка доступного в PM > setup > Metric Types. Добавить свои можно в /opt/noc/pm/probes/, например для мультикастовых счетчиков на интерфейсе можно дописать свои метрики в /opt/noc/pm/probes/generic/network/snmp_interface.py



Для МО должен быть включен caps_discovery, noc определит какие возможности поддерживает МО и будет дергать только нужные оид, поэтому в одном сете можно совместить несколько метрик, с pppoe браса будет дергаться только счетчик pppoe, и тд

7) Созданный метрик сет теперь нужно применить по назначению.

Для примера с брасами

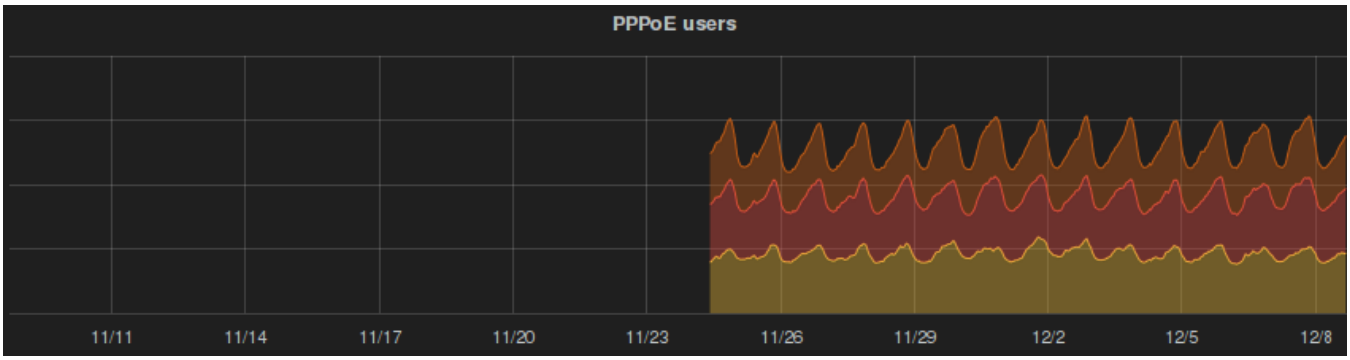
SA > Setup > Managed Object Profiles > Edit (BRAS) > Metrics > Add

Для метрик интерфейсов, metric set надо добавлять в Interface Profile

> **Замечание**
>
> Для сбора данных, noc надо знать какой oid запрашивать на устройстве, для интерфейсов oid формируется из идентификатора ветки и ifindex, который кладется в базу в процессе interface_discovery. Так как interface_discovery очевидно в любом случае будет работать чтобы положить интерфейсы в базу, необходимо озаботиться чтобы script производящий discovery также возвращал ifindex

Все, теперь noc будет собирать данные

Чтобы увидеть что удалось собрать, идем в PM > Dashboards и запускаем Графану, ее расписывать не буду, методом тыка можно и самому разобраться, графики получаются примерно такими



Главный оставшийся вопрос, как получить график в виде файла, потому что js-график никуда не скопировать, не вставить и не приложить

А в остальном достаточно неплохая система получилась, если совместить ее с PyRule из Pyrule для Inventory, который пригодится для FM каждого, то вообще все будет происходить автоматически, надо только сидеть и выбирать какие данные нарисовать на графике, сами данные уже заранее будут лежать в базе

> при дебаге не хило помогает вот эта документация https://www.evernote.com/shard/s57/sh/d8ab1f6e-646d-46f0-9706-d97494819cde/00816532089b33ef67fe7a35ca4b3156