

Развернуть netdata в тестовом k8s кластере

Цель работы

Нужно развернуть netdata в k8s кластере

Реализация

Развернул netdata в тестовом кластере

Ссылка: <https://netdata.k8stest.eterfund.ru>

<https://learn.netdata.cloud/docs/netdata-agent/installation/kubernetes>

<https://learn.netdata.cloud/docs/netdata-agent/installation/kubernetes-helm-chart-reference>

<https://github.com/netdata/helmchart/blob/master/charts/netdata/values.yaml>

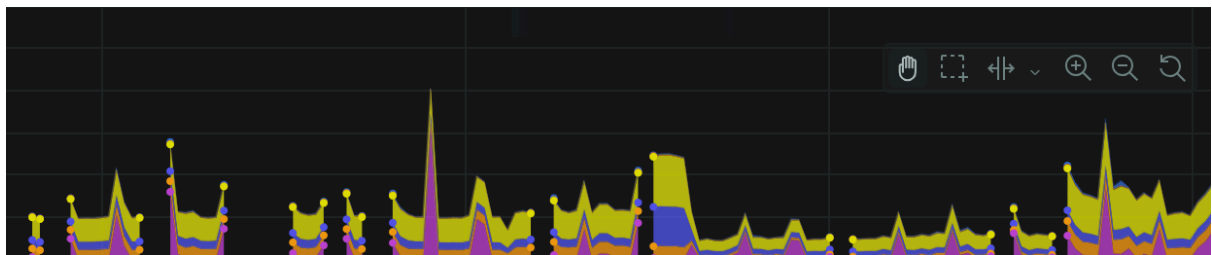
Отключил systemd-journal плагин для нетдаты в конфиге

<https://learn.netdata.cloud/docs/working-with-logs/systemd-journal/>

В UI написано, что логи весят 160 Mi

Файловая система весит 4Гб, после трима уменьшилось до 2х, метрики остались такими же

tier1 метрики показываются некорректно, много пробелов, возможно проблема с хранением или записью tier1 метрик



Удалил имеющиеся метрики, когда соберется метрик на 14+ дней нужно будет проверить корректность отображения tier1 метрик (14 дней - срок хранения tier0 метрик)

Если tier1 метрики будут показываться также как и в скриншоте выше нужно будет искать решение с хранением и записью метрик

Другой вариант - хранить все метрики в tier0, но это будет требовать значительно больше места

Ссылка на официальную документацию netdata с информацией по их движку бд

<https://learn.netdata.cloud/docs/developer-and-contributor-corner/database-engine>

Решил проверить все включенные плагины на предмет потенциально ненужных плагинов

По ссылке <https://netdata.k8stest.eterfund.ru/netdata.conf> можно посмотреть конфиг, дописал ссылки и небольшое объяснение для всех перечисленных тут плагинов

[plugins]

```
logs-management = no
systemd-journal = no
# netdata monitoring extended = no
# idlejitter = yes
```

Вызывает поток выполнения, который отправляет запрос на сон на определенное время, выводит разницу в миллисекундах между запросом и реальным значением

<https://learn.netdata.cloud/docs/collecting-metrics/synthetic-checks/idle-os-jitter>

```
# netdata monitoring = no
# profile = no
# tc = yes
```

Плагин использует команду tc

<https://learn.netdata.cloud/docs/collecting-metrics/linux-systems/network/tc-qos-classes>

```
# diskpace = yes
```

Собирает информацию об занятом/свободном месте на жд. Плагин показывает некорректную информацию (к примеру считает, что на жд 140Гб памяти, когда он всего 32Гб). Поищу какое-то решение, если решения не будет есть смысл отключить этот плагин и поискать альтернативный

<https://learn.netdata.cloud/docs/collecting-metrics/linux-systems/disk-space>

```
# proc = yes
```

Собирает метрики из папок /proc и /sys

<https://github.com/netdata/netdata/blob/master/src/collectors/proc.plugin/README.md>

```
# cgroups = yes
```

Собирает метрики связанные с контейнерами

<https://learn.netdata.cloud/docs/collecting-metrics/containers-and-vms/containers>

```
# timex = yes
```

Monitors the system kernel clock synchronization state (идея понятна, как перевести не очень понятно)

<https://learn.netdata.cloud/docs/collecting-metrics/system-clock-and-ntp/timex>

```
# enable running new plugins = yes
# check for new plugins every = 60
# slabinfo = no
# freeipmi = no
# statsd = yes
```

Собирает statsd метрики

Никакой нормальной ссылки не нашел, если найду, то напишу в другом комментарии

```
# apps = yes
```

Разделяет утилизацию системных ресурсов на процессы, юзеров и группы юзеров

<https://github.com/netdata/netdata/blob/master/src/collectors/apps.plugin/README.md>

```
# charts.d = yes
```

Оркестратор для модулей написанных на bash v4+

<https://learn.netdata.cloud/docs/developer-and-contributor-corner/external-plugins/charts.d.plugin>

```
# debugfs = yes
```

```
# go.d = yes
```

Оркестратор для модулей написанных на go

<https://learn.netdata.cloud/docs/developer-and-contributor-corner/external-plugins/go.d.plugin>

```
# ioping = yes
```

Измеряет задержку диска

<https://learn.netdata.cloud/docs/collecting-metrics/synthetic-checks/ioping>

```
# network-viewer = yes
```

Собирает метрики связанные с входящим/исходящим трафиком

Ссылки не нашел

```
# perf = yes
```

Собирает метрики связанные с производительностью цпу

<https://learn.netdata.cloud/docs/collecting-metrics/linux-systems/cpu-performance>

<https://learn.netdata.cloud/docs/collecting-metrics/linux-systems/cpu-performance>

```
# python.d = yes
```

Оркестратор для модулей написанных на python v2 и v3

<https://learn.netdata.cloud/docs/developer-and-contributor-corner/external-plugins/python.d.plugin>

Метрики связанные с памятью (mem.available) тоже показывают некоректное значение, вместо 8Гб - 16Гб, и так для каждой ноды, хотя все эти ноды - одна вм.

Попробовал отключить child и k8s-state ноды

Это не помогло, нетдата все также считает нода 16 гигаб байт, а не 8Гб

Метрики собираются из /proc/meminfo

Посмотрел значения в файле на вм

```
[root@k8stest ~]# grep Mem /proc/meminfo
```

```
MemTotal:      8133096 kB
```

```
MemFree:       1083608 kB
```

```
MemAvailable:  2926968 kB
```

```
[root@k8stest ~]# grep Com /proc/meminfo
```

```
CommitLimit:   4066548 kB
```

```
Committed_AS:  13726524 kB
```

```
[root@k8stest ~]# grep Swap /proc/meminfo
```

```
SwapCached:    0 kB
```

```
SwapTotal:     0 kB
```

```
SwapFree:      0 kB
```

Проверил логи journalctl

```
journalctl | grep Memory
```

Jul 26 19:10:38 k8stest.eterfund.ru kernel: Memory: 8115412K/8388064K available (14344K kernel code, 2740K rwdatas, 8352K rodata, 2916K init, 16772K bss, 272396K reserved, 0K cma-reserved)

Проверил другую машину с k3s, Committed_As также > RAM

Наконец нашел документацию, в которой написано что показывает Committed_As
<https://www.kernel.org/doc/Documentation/filesystems/proc.txt>

Исходя из документации значение может превышать общую память

Количество места на диске, после отключения child и k8s-state нод, показывается правильно, попробовал развернуть и k8s-state с выключенным плагином diskstats. В необходимости child ноды очень сомневаюсь, так как есть всего 1 нода. Разница между parent и child: parent хранит собранные метрики, а child отправляет собранные метрики в parent.

Отключил все плагины, кроме go.d для ноды k8s-state

Также отключил pvc для k8s-state

Текущие проблемы с плагинами

1) Disk Stats выводит как физические, так и виртуальные диски: вольюмы лонгхорна

В конфиге отключил сбор логов для виртуальных дисков

```
[plugin:proc:/proc/diskstats]  
performance metrics for virtual disks = auto
```

заменив auto на no

Ссылка на доку:

<https://learn.netdata.cloud/docs/collecting-metrics/linux-systems/disk/disk-statistics>

2) Disk Space выводит маунтпоинты для контейнера, а не для виртуалки

К примеру /var/cache/netdata и /var/lib/netdata

Можно собирать метрики только для маунтпоинта /, который соответствует / на виртуалке

<https://learn.netdata.cloud/docs/collecting-metrics/linux-systems/disk-space>

3) /proc/net выводит значения для контейнера, а не для виртуалки

К примеру видит только один интерфейс, ens0, потому что это единственный интерфейс в контейнере

При этом метрики до отключения child ноды были правильными, видно интерфейс ens20

4) Перестали работать cgroup, kubulet и kubeproxy

Вернул child ноду, убрал почти все плагины для parent ноды. Теперь cgroup, kubelet и kubeproxy работают. /proc/net видит интерфейс ens20.

Замена auto на no в первом пункте не помогла, так как это просто отключило сбор метрик для дисков, хранящихся в /sys/devices/virtual/block/%s

В UI есть возможность выбрать лейбл, в том числе модель (нужно QEMU HARDDISK)

Можно задать сбор метрик только для диска sda, а затем добавлять другие диски, если они были созданы, но пока попробую найти более эстетически приятный способ

Маунтпоинты все равно контейнера, но / соответствует / на вм, решил собирать метрики только для этого маунтпоинта

```
[root@k8stest ~]# df -h /  
Filesystem      Size  Used Avail Use% Mounted on  
/dev/sda1       30G   19G   12G   62% /
```

```
root@k8stest:/# df -h /  
Filesystem      Size  Used Avail Use% Mounted on  
overlay         30G   19G   12G   62% /
```

В конфиге добавил /host* /dev*

```
[plugin:proc:diskspace]  
    exclude space metrics on paths = /proc/* /sys/* /var/run/user/* /run/user/* /snap/*  
/var/lib/docker/* /host* /dev*
```

Теперь видны метрики только для /

Удалил плагины statsd и timex для parent ноды
Почистил из файла с конфигами ненужные строки

Развернул netdata в основном кластере

Отключил k8s-state

В результате diskspace все равно некорректно отображает место на машине, поэтому решил отключить этот плагин

Отключил все плагин для parent ноды; теперь за сбор метрик отвечает child нода, а за их хранение - parent

Ссылка на github gist с конфигом netdata.conf для child ноды:

<https://gist.github.com/jack-lull/31a0164f58f2eb60446490e7c79d9452>

Ссылка на github gist с values для netdata:

<https://gist.github.com/jack-lull/ad11ec6a5ed06b053971a0d606d4ccd2>

Ссылка на github gist с helmfile:

<https://gist.github.com/jack-lull/c91b1dce21f95503729ad4ead536e17d>