

Сегодня хочу рассказать о том, как я мониторую самодельные бэкапы с помощью zabbix. Подход немного костыльный, но на вопрос отвечает, ниже расскажу в чем его смысл. Я рассмотрю 2 способа, когда у вас бэкапы в виде директорий с оригинальными файлами, а второй — в виде запакованных архивов.

Если у вас есть желание научиться строить и поддерживать высокодоступные и надежные системы, рекомендую познакомиться с **онлайн-курсом «Администратор Linux»** в OTUS. Курс не для новичков, для поступления нужно пройти .

Содержание:

- 1 Введение
 - 1.1 Бэкапы в виде сырых данных в директории (1-й способ)
 - 1.2 Бэкапы в виде запакованных архивов (2-й способ)
- 2 1-й способ
 - 2.1 Скрипт создания проверочного файла
 - 2.2 Настраиваем мониторинг бэкапов через проверку даты изменения файлов
- 3 2-й способ
 - 3.1 Скрипты сбора информации о бэкапах
 - 3.2 Добавляем в zabbix-agent информацию о бэкапах
 - 3.3 Добавляем шаблон с мониторингом бэкапов на сервер
- 4 Заключение

Введение

Если у вас еще нет своего сервера для мониторинга, то рекомендую материалы на эту тему. Для тех, кто предпочитает систему CentOS:

1. Установка CentOS 7.

2. Настройка CentOS 7.
3. Установка и настройка zabbix сервера.

То же самое на Debian 9, если предпочитаете его:

1. Установка Debian 9.
2. Базовая настройка Debian 9.
3. Установка и настройка zabbix на debian.

Бэкапы в виде сырых данных в директории (1-й способ)

У меня много где настроена самодельная система бэкапа, похожая на то, что описано в статье по настройке бэкапа с помощью rsync. Не буду подробно останавливаться на том, почему бэкаплю именно так. Во многих случаях это удобно, так как всегда имеешь под рукой свежую версию данных в исходном виде. В случае повреждения источника, просто меняешь точку монтирования и получаешь практически сразу все данные, без простоя рабочего процесса.

На всякий случай хочется следить за тем, что бэкапы у тебя актуальны и в случае чего ты можешь на них рассчитывать. Делать мы это будет по очень простой схеме. На сервере источнике будет раз в сутки создаваться файл. Во время бэкапа этот файл будет улетать на сервер с резервными данными. Этот сервер подключен к системе мониторинга zabbix с помощью агента. Этот агент будет периодически проверять дату последнего изменения файла. Если эта дата больше заданного интервала, то мы будем получать оповещение о том, что бэкапы не выполняются.

Для настройки описанной схемы нам понадобится выполнить несколько шагов:

1. На серверах источника настроить скрипт, который будет создавать файл и поместить его в планировщик.
2. На сервере zabbix настроить на хосте с бэкапами item и trigger для слежения и оповещения о дате файла.

Бэкапы в виде запакованных архивов (2-й способ)

Если у вас бэкапится, к примеру, какая-то база в дампе или есть просто отдельный файл, то имеет смысл его архивировать и хранить в виде одиночного архива. Для таких бэкапов тоже нужен мониторинг. Чтобы следить за актуальностью бэкапа, я предлагаю мониторить 2 параметра:

1. Размер файла. Если он равен нулю, то срабатывает триггер.
2. Дата создания бэкапа. Если он старше какого-то срока, в моем примере будут 24 часа, то шлем оповещение.

Мониторинг бэкапов будет настроен из расчета, что у вас все бэкапы лежат в одной директории на сервере. В этой директории резервные копии хранятся

для каждого объекта в отдельной папке. Будет настроено автообнаружение папок в директории с бэкапами.

1-й способ

Скрипт создания проверочного файла

Я использую описанную выше схему для бэкапа как windows так и linux серверов. Поэтому скрипта будет 2, для каждой системы. Вот пример такого скрипта для **linux**:

```
# mcedit create-timestamp.sh

#!/bin/sh
echo `date +"%Y-%m-%d_%H-%M" ` > /shares/docs/timestamp
```

Скрипт просто создает файл и записывает в него текущую дату. Нам этого достаточно. Писать туда можно все, что угодно, так как проверять мы будем не содержимое, а дату последнего изменения.

Добавляем этот скрипт в cron:

```
# mcedit /etc/crontab

#Create timestamp for backup monitoring
1 15 * * * root /root/bin/create-timestamp.sh
```

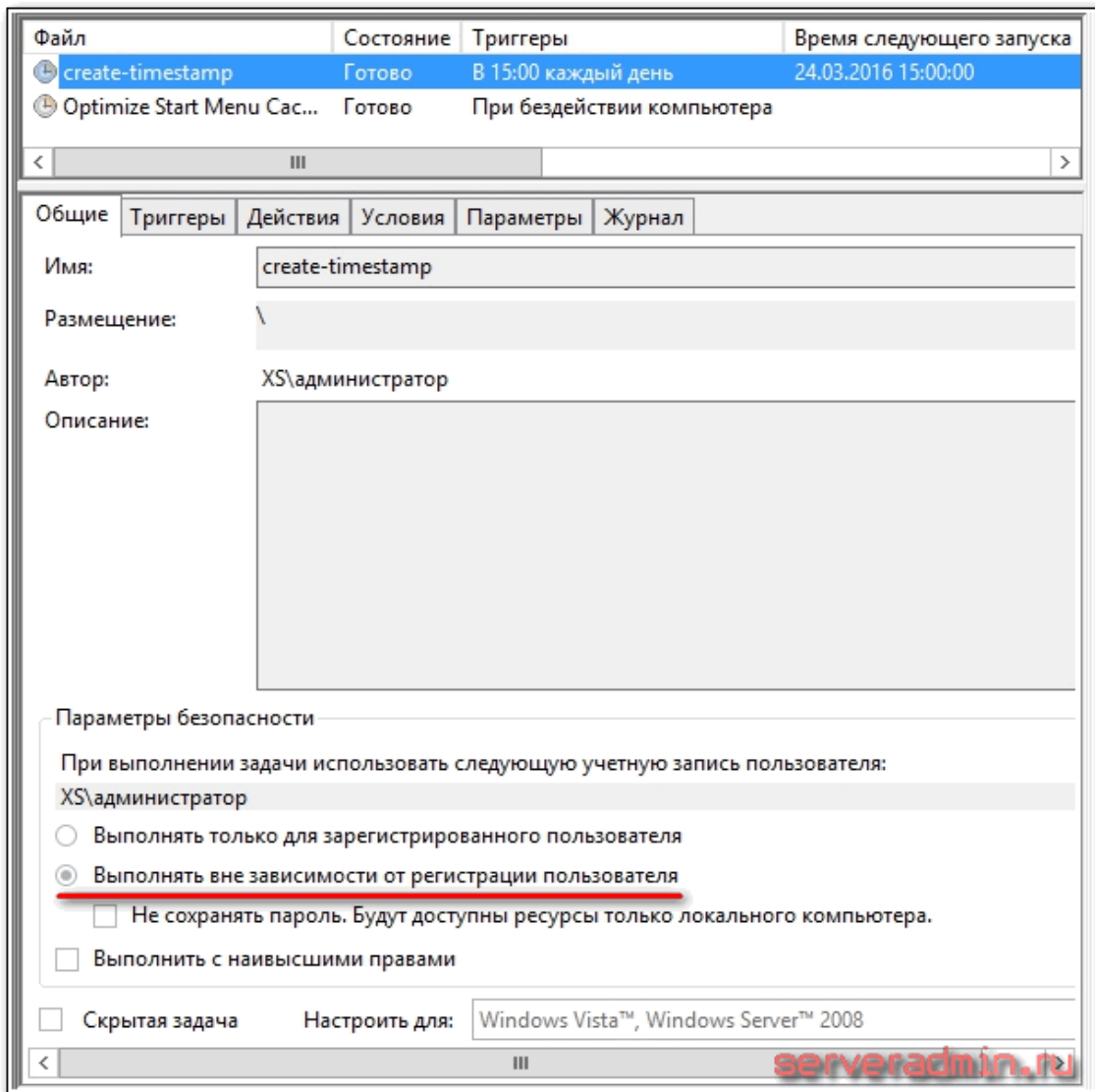
Раз в день в 15:01 скрипт будет создавать файл, перезаписывая предыдущий.

Делаем то же самое на **windows**. Создаем файл *create-timestamp.bat* следующего содержания:

```
echo %date:~-10% > D:\documents\timestamp
```

И добавляем его в планировщик windows. Не забудьте указать, чтобы скрипт запускался вне зависимости от регистрации пользователя, то есть чтобы он

работал, даже если в системе никто не залогинен.



The screenshot shows the Windows Task Scheduler interface. At the top, a table lists tasks:

Файл	Состояние	Триггеры	Время следующего запуска
create-timestamp	Готово	В 15:00 каждый день	24.03.2016 15:00:00
Optimize Start Menu Cas...	Готово	При бездействии компьютера	

Below the table, the 'create-timestamp' task is selected, and its properties are shown in the 'Общие' (General) tab:

- Имя: create-timestamp
- Размещение: \
- Автор: XS\администратор
- Описание: (empty text area)

The 'Параметры безопасности' (Security options) section is expanded, showing the following settings:

- При выполнении задачи использовать следующую учетную запись пользователя: XS\администратор
- Выполнять только для зарегистрированного пользователя
- Выполнять вне зависимости от регистрации пользователя
- Не сохранять пароль. Будут доступны ресурсы только локального компьютера.
- Выполнить с наивысшими правами

At the bottom, the 'Скрытая задача' (Hidden task) checkbox is unchecked, and the 'Настроить для:' (Configure for:) dropdown is set to 'Windows Vista™, Windows Server™ 2008'. A watermark 'serveradmin.ru' is visible in the bottom right corner of the screenshot.

Запустите оба скрипта, чтобы проверить, что все в порядке, и необходимые файлы создаются.

Запустите стандартные скрипты бэкапа, чтобы созданные файлы переместились на резервные сервера. После этого можно приступить к настройке мониторинга за изменением файлов в zabbix.

Настраиваем мониторинг бэкапов через проверку даты изменения файлов

Дальше привычное дело по созданию итемов и триггеров. Идем в панель управления zabbix, открываем раздел **Configuration -> Hosts**, выбираем сервер, на котором у нас хранятся бэкапы и создаем там итем со следующими параметрами:

The screenshot shows the configuration form for a Zabbix agent item. The form is titled 'xb-share-documents'. Key fields include:

- Name: xb-share-documents
- Type: Zabbix agent
- Key: vfs.file.time[/mnt/data/BackUp/xb-share/documents/timesta] Select
- Host interface: 10.1.4.4 : 10050
- Type of information: Numeric (unsigned)
- Data type: Decimal
- Units: unixtime
- Use custom multiplier: 1
- Update interval (in sec): 600
- Flexible intervals: A table with columns 'Interval', 'Period', and 'Action'. It shows 'No flexible intervals defined.'
- New flexible interval: Interval (in sec) 50, Period 1-7,00:00-24:00, Add
- History storage period (in days): 30
- Trend storage period (in days): 90
- Store value: As is
- Show value: As is [show value mappings](#)
- New application: (empty)
- Applications: -None-, Backup, CPU, Filesystems, General, Mdam softraid
- Populates host inventory field: -None-
- Description: (empty text area)
- Enabled:

Buttons at the bottom: Update, Clone, Clear history and trends, Delete, Cancel. A watermark 'serveradmin.ru' is visible at the bottom center.

На скриншот не влезла вся строка параметра **Key**, поэтому привожу ее здесь:

```
vfs.file.time[/mnt/data/BackUp/xb-share/documents/timestamp,modify]
```

/mnt/data/BackUp/xb-share/documents/timestamp	Путь к проверяемому файлу на сервере бэкапов
modify	Время изменения файла. Параметр может принимать значения: access — время последнего доступа, change — время последнего изменения

Не очень понимаю, чем отличается время изменения, от времени последнего изменения. Эта информация из документации zabbix. Для того, чтобы у вас корректно собирались данные, необходимо, чтобы у пользователя zabbix были права на чтение указанного файла. Обязательно проверьте это. Я не сделал это, через одну из папок агент не мог пройти из-за недостатка прав. В итоге получил ошибку:

```
17177:20160321:002008.008 item "xb-share-documents:vfs.file.time[/mnt/data/BackUp/xb-share/documents/timestamp]" became not supported: Not supported by Zabbix Agent
```

Из текста не понятно, в чем проблема. Про права я догадался. Обновление итема установил раз в 10 минут (параметр update interval), чаще не вижу смысла, можно вообще поставить пару раз в сутки, в зависимости от вашего плана архивации данных.

Теперь создадим триггер для этого элемента данных:

Name: Backup DOCS not work on XB-SHARE

Expression: `{xm-backup:vfs.file.time[/mnt/data/BackUp/xb-share/documents/timestamp,modify].now(0)}-{xm-backup:vfs.file.time[/mnt/data/BackUp/xb-share/documents/timestamp,modify].last(0)}>172800`

Expression constructor

Multiple PROBLEM events generation:

Description:

URL:

Severity: Not classified | Information | Warning | Average | **High** | Disaster

Enabled:

Update | Clone | Delete | Cancel

serveradmin.ru

Разберем, что у нас в выражении написано:

```
{xm-backup:vfs.file.time[/mnt/data/BackUp/xb-share/documents/timestamp,modify].now(0)}-{xm-backup:vfs.file.time[/mnt/data/BackUp/xb-share/documents/timestamp,modify].last(0)}>172800
```

xm-backup — сервер, на котором хранятся бэкапы. Мы берем текущее время, вычитаем из него время последнего изменения файла. Если оно больше **172800** секунд (2 суток), то срабатывает триггер. Вы можете сами выбрать подходящий вам интервал времени сравнения в зависимости от плана бэкапа.

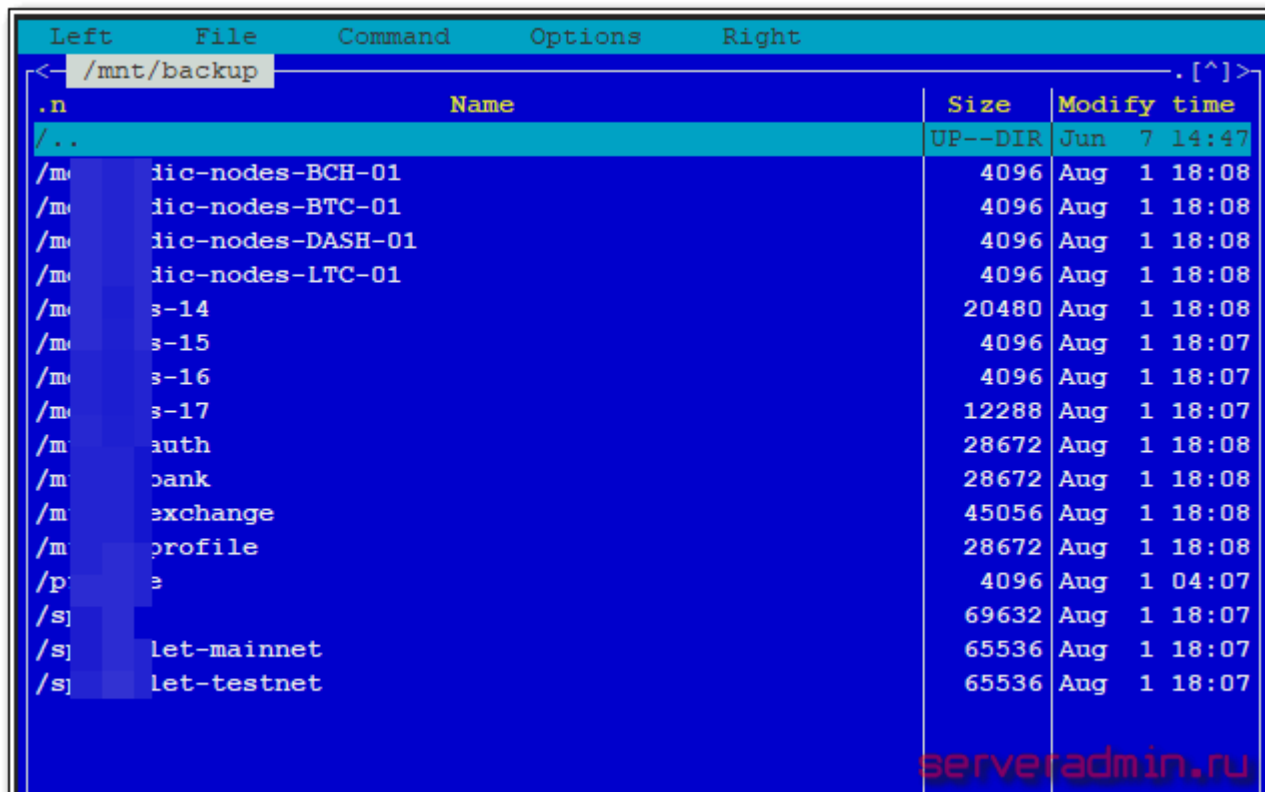
Для тестирования работы оповещений отключите в один из дней скрипты на источниках, создающие проверочный файл. Как только он просрочится, сработает триггер.

На этом все. Мы настроили простейший мониторинг бэкапов с помощью zabbix. Если по какой-то причине файлы перестанут синхронизироваться с сервером резервных копий, вы узнаете об этом и сможете вовремя обнаружить проблему.

2-й способ

Скрипты сбора информации о бэкапах

Во втором случае у нас есть директория с бэкапами `/mnt/backup`, где каждая отдельная папка содержит набор однотипных архивов за разные даты.



Left	File	Command	Options	Right
<	/mnt/backup			. [^]>
.n	Name		Size	Modify time
/. .		UP--DIR	4096	Jun 7 14:47
/mnt	dic-nodes-BCH-01		4096	Aug 1 18:08
/mnt	dic-nodes-BTC-01		4096	Aug 1 18:08
/mnt	dic-nodes-DASH-01		4096	Aug 1 18:08
/mnt	dic-nodes-LTC-01		4096	Aug 1 18:08
/mnt	s-14		20480	Aug 1 18:08
/mnt	s-15		4096	Aug 1 18:07
/mnt	s-16		4096	Aug 1 18:07
/mnt	s-17		12288	Aug 1 18:07
/mnt	auth		28672	Aug 1 18:08
/mnt	bank		28672	Aug 1 18:08
/mnt	exchange		45056	Aug 1 18:08
/mnt	profile		28672	Aug 1 18:08
/mnt	pe		4096	Aug 1 04:07
/mnt	let-mainnet		69632	Aug 1 18:07
/mnt	let-testnet		65536	Aug 1 18:07
/mnt	let-testnet		65536	Aug 1 18:07

Внутри следующее содержимое. Каждый час делает резервная копия базы данных.

2018-07-31_22-05-spw_bd.sql.gz	34116	Jul 31 22:05
2018-07-31_23-05-spw_bd.sql.gz	34114	Jul 31 23:05
2018-08-01_00-05-spw_bd.sql.gz	34115	Aug 1 00:05
2018-08-01_01-05-spw_bd.sql.gz	34114	Aug 1 01:05
2018-08-01_02-05-spw_bd.sql.gz	34115	Aug 1 02:05
2018-08-01_03-05-spw_bd.sql.gz	34115	Aug 1 03:05
2018-08-01_04-05-spw_bd.sql.gz	34116	Aug 1 04:05
2018-08-01_05-05-spw_bd.sql.gz	34116	Aug 1 05:05
2018-08-01_06-05-spw_bd.sql.gz	34115	Aug 1 06:05
2018-08-01_07-05-spw_bd.sql.gz	34115	Aug 1 07:05
2018-08-01_08-05-spw_bd.sql.gz	34116	Aug 1 08:05
2018-08-01_09-05-spw_bd.sql.gz	34116	Aug 1 09:05
2018-08-01_10-05-spw_bd.sql.gz	34116	Aug 1 10:05
2018-08-01_11-05-spw_bd.sql.gz	34115	Aug 1 11:05
2018-08-01_12-05-spw_bd.sql.gz	34115	Aug 1 12:05
2018-08-01_13-05-spw_bd.sql.gz	34114	Aug 1 13:05
2018-08-01_14-05-spw_bd.sql.gz	34115	Aug 1 14:05
2018-08-01_15-05-spw_bd.sql.gz	34115	Aug 1 15:05
2018-08-01_16-05-spw_bd.sql.gz	34115	Aug 1 16:05
2018-08-01_17-05-spw_bd.sql.gz	34114	Aug 1 17:05

serveradmin.ru

Дальнейшее описание способа мониторинга бэкапов будет актуально только для точно такой же структуры бэкапов. У вас есть общая директория, в ней папки, где внутри сжатые архивы формата .gz. Если расширение файлов другое, то можно подредактировать скрипты, которые я покажу дальше.

Обращаю внимание, что приведенная дальше информация требует осмысленного повторения. Не нужно слепо копировать и пробовать. Нужно понять, что я делаю и изменить скрипты и другие настройки под свои нужды. Я просто даю направление, подсказываю способ.

Нам пригодятся несколько скриптов. Первый из них — *backup-info.sh*. Он будет формировать текстовый файл *backup_info.txt*, где будет в одну строку указано имя папки с архивом, размер последнего архива, дата создания архива, время создания архива.

```
#!/bin/bash

BK_DIR=/mnt/backup # директория с бэкапами
SRV_NAME=`ls $BK_DIR` # формируем список папок с архивами
LOG_FILE=/etc/zabbix/scripts/backup_info.txt # выходной файл с информацией

# удаляем предыдущий файл с информацией
rm $LOG_FILE

for a in $SRV_NAME
do
    # определяем размер последнего бэкапа
    bsize=`ls -lt $BK_DIR/$a | grep .gz | awk 'NR == 1{print$5}'`
    if [[ "$bsize" = "" ]]
    then
        bsize=0
    fi
    # определяем имя последнего бэкапа
    bfile=`ls -lt $BK_DIR/$a | grep .gz | awk 'NR == 1{print$9}'`
    # определяем дату создания последнего бэкапа
    btime=`stat $BK_DIR/$a/$bfile | grep Modify | awk '{print $2,$3}' | cut -f1-1 -d"."`
    # записываем информацию в файл
    echo $a $bsize $btime >> $LOG_FILE
done
```

В текстовом файле будет следующее:

```
dedic-nodes-BCH-01 496211 2018-08-01 17:54:58
dedic-nodes-BTC-01 702496 2018-08-01 17:12:46
dedic-nodes-DASH-01 244488 2018-08-01 17:40:51
dedic-nodes-LTC-01 491030 2018-08-01 17:20:23
vps-14 858433 2018-08-01 18:01:02
vps-15 235689258 2018-08-01 18:02:51
```

```
vps-16 235977137 2018-08-01 18:05:54  
vps-17 23983868 2018-08-01 17:09:59
```

Я рекомендую разобрать каждую проверку и вручную ее выполнить в консоли, чтобы убедиться, что она работает как и должна и получает правильное значение. В разных дистрибутивах могут быть разные ключи и выводы команд. Например, команда `ls` может разделять значения одним пробелом, а может несколькими. Так же вывод и порядок столбцов в разных дистрибутивах может быть разным. Я все проверил только в Ubuntu 16. Данные скрипты писались и отлаживались в ней.

Следующий скрипт — `backup-time.sh`. Он берет значения из текстового файла, который формирует предыдущий скрипт, вычисляет разницу в часах между текущей датой и временем создания последнего бэкапа. Полученную информацию записывает в файл `backup_time.txt`.

```
#!/bin/bash  
  
BK_DIR=/mnt/backup # директория с бэкапами  
SRV_NAME=`ls $BK_DIR` # формируем список папок с архивами  
LOG_FILE=/etc/zabbix/scripts/backup_time.txt # выходной файл с информацией  
INFO_FILE=/etc/zabbix/scripts/backup_info.txt # входной файл для парсинга  
  
# удаляем предыдущий файл с информацией  
rm $LOG_FILE  
  
for a in $SRV_NAME  
do  
    # получаем дату создания архива  
    btime=`cat $INFO_FILE | grep $a | awk '{print $3,$4}'`  
    # переводим дату создания архива в unix формат  
    stime=`date --date="$btime" +%s`  
    # получаем текущую дату в unix формате  
    ctime=`date +%s`  
    # сравниваем 2 даты и переводим в часы  
    difftime=$(( $ctime - $stime ) / 60 ]  
    # записываем полученный результат в файл  
    echo $a $difftime >> $LOG_FILE  
done
```

```
done
```

Тестовый файл будет следующего содержания.

```
dedic-nodes-BCH-01 21
dedic-nodes-BTC-01 63
dedic-nodes-DASH-01 35
dedic-nodes-LTC-01 55
vps-14 14
vps-15 13
vps-16 10
vps-17 66
```

Рисуем скрипт *backup-discovery.sh*, который будет выполнять автообнаружение папок с архивами и передавать данные в json формате на zabbix сервер.

```
#!/bin/bash

JSON=$(for i in `ls -l /mnt/backup | sed '1d' | awk '{print $9}'`; do printf "{\"#BACKUP\": \"${i}\",\"; done | sed
's/^\(.*\)$.$/\1/'
printf "{\"data\": [\"
printf \"$JSON\"
printf "]"}
```

Запустите файл и проверьте, что он корректно формирует вывод. Должно быть примерно следующее.

```
{"data": [{"#BACKUP": "dedic-nodes-BCH-01"}, {"#BACKUP": "dedic-nodes-BTC-01"}, {"#BACKUP": "dedic-nodes-
DASH-01"}, {"#BACKUP": "dedic-nodes-
LTC-01"}, {"#BACKUP": "vps-14"}, {"#BACKUP": "vps-15"}, {"#BACKUP": "vps-16"}, {"#BACKUP": "vps-17"}]}
```

И еще 2 скрипта, которые будут непосредственно отдавать информацию заббикс серверу. Первый — *analyze-size.sh*. Он будет передавать серверу информацию о размере архива.

```
#!/bin/bash  
cat /etc/zabbix/scripts/backup_info.txt | grep $1 | awk '{print $2}'
```

И второй — *analyze-time.sh*. Он передает информацию о времени создания бэкапа относительно текущего.

```
#!/bin/bash  
cat /etc/zabbix/scripts/backup_time.txt | grep $1 | awk '{print $2}'
```

В завершении делаем пользователя *zabbix* владельцем всех скриптов и текстовых файлов. Если забыть это сделать, то потом получите ошибку и неактивный итем на сервере.

```
# chown -R zabbix. /etc/zabbix/scripts
```

Добавляем в *zabbix-agent* информацию о бэкапах

Теперь нам нужно добавить новые итемы в агент заббикса через *UserParameter*. Создаем файл */etc/zabbix/zabbix_agentd.d/backup_info.conf* следующего содержания.

```
UserParameter=backup.discovery[*],/etc/zabbix/scripts/backup-discovery.sh  
UserParameter=backup.size[*],/etc/zabbix/scripts/analyze-size.sh $1  
UserParameter=backup.time[*],/etc/zabbix/scripts/analyze-time.sh $1
```

Перезапускаем агента и проверяем. Для начала автообнаружение папок.

```
# zabbix_agentd -t backup.discovery
```

Вы должны увидеть список папок в *json* формате, как при ручном запуске скрипта. Дальше проверим вывод информации о самих бэкапах.

```
# zabbix_agentd -t backup.size[dedic-nodes-BCH-01]  
backup.size[dedic-nodes-BCH-01] [t|496211]
```

```
# zabbix_agentd -t backup.time[dedic-nodes-BCH-01]  
backup.time[dedic-nodes-BCH-01] [t|81]
```

Если получаете актуальную информацию, значит все в порядке. Можно переходить на zabbix-server.

Добавляем шаблон с мониторингом бэкапов на сервер

Здесь вам ничего руками делать не надо будет, так как шаблон я уже сделал и экспортировал в файл — zabbix-backup-info.xml. Экспорт выполнен с версии 3.4. Заработает ли на предыдущих версиях — не знаю, не проверял.

В шаблоне создано одно правило автообнаружения с двумя прототипами итемов и триггеров. Итемы имеют такие настройки:

Все шаблоны / Backup info Список обнаружений / Backup Dirs Discovery Прототипы элементов данных 2 Прототипы триггеров 2 Прототипы графики

Прототип элемента данных [Предобработка](#)

Имя

Тип

Ключ

Тип информации

Единица измерения

Интервал обновления

Период хранения истории

Период хранения динамики изменений

Отображение значения [показать преобразования значений](#)

Новая группа элементов данных

Группы элементов данных

- Нет-
- Backup

Новый прототип группы элементов данных

[serveradmin.ru](#)

Все шаблоны / Backup info Список обнаружений / Backup Dirs Discovery Прототипы элементов данных 2 Прототипы триггеров 2 Прототипы г...

Прототип элемента данных Предобработка

Имя

Тип

Ключ

Тип информации

Единица измерения

Интервал обновления

Период хранения истории

Период хранения динамики изменений

Отображение значения [показать преобразования значений](#)

Новая группа элементов данных

Группы элементов данных

- Нет-
- Backup

[serveradmin.ru](#)

Триггеры такие:

Прототип триггеров [Зависимости](#)

Имя

Важность Не классифицировано Информация Предупреждение Средняя Высокая Чрезвычайная

Выражение проблемы

[Конструктор выражения](#)

Генерация ОК событий Выражение Выражение восстановления Нет

Выражение восстановления

[Конструктор выражения](#)

Режим генерации событий ПРОБЛЕМА Одиночная Множественный

ОК событие закрывает Все проблемы Все проблемы если значения тегов совпадают

Теги

<input type="text" value="тег"/>	<input type="text" value="значение"/>	<input type="button" value="Удалить"/>
----------------------------------	---------------------------------------	--

[Добавить](#)

serveradmin.ru

Прототип триггеров **Зависимости**

Имя

Важность

Выражение проблемы

[Конструктор выражения](#)

Генерация ОК событий

Выражение восстановления

[Конструктор выражения](#)

Режим генерации событий ПРОБЛЕМА

ОК событие закрывает

Теги

[Добавить](#)

Разрешить закрывать вручную

serveradmin.ru

Триггер срабатывает, если архив имеет нулевой размер и если он старше 25 часов. Время переведено в минуты. Можете изменить значение по своему усмотрению. Прикрепляйте шаблон к хосту, где настроили zabbix-agent и ждите поступления данных. Обновляются они раз в час. Получите примерно такую картинку.

<input type="checkbox"/>	Backup time	-auth	01.08.2018 18:45:29	10	-61
<input type="checkbox"/>	Backup time	/ps-17	01.08.2018 18:45:29	66	-60
<input type="checkbox"/>	Backup time	/ps-16	01.08.2018 18:45:29	10	-60
<input type="checkbox"/>	Backup time	/ps-15	01.08.2018 18:45:29	13	-60
<input type="checkbox"/>	Backup time	/ps-14	01.08.2018 18:45:29	14	-60
<input type="checkbox"/>	Backup time	dedic-nodes-LTC-01	01.08.2018 18:45:29	55	-59
<input type="checkbox"/>	Backup time	dedic-nodes-DASH-01	01.08.2018 18:45:29	35	-60
<input type="checkbox"/>	Backup time	dedic-nodes-BTC-01	01.08.2018 18:45:29	63	-12
<input type="checkbox"/>	Backup time	dedic-nodes-BCH-01	01.08.2018 18:45:29	21	-60
<input type="checkbox"/>	Backup size	llet-testnet	01.08.2018 18:41:31	711.26 KB	
<input type="checkbox"/>	Backup size	llet-mainnet	01.08.2018 18:41:31	702.76 KB	+5 B
<input type="checkbox"/>	Backup size		01.08.2018 18:41:31	33.32 KB	+1 B
<input type="checkbox"/>	Backup size	le	01.08.2018 18:41:31	87.79 MB	
<input type="checkbox"/>	Backup size	-profile	01.08.2018 18:41:30	79.12 KB	+51 B
<input type="checkbox"/>	Backup size	-exchange	01.08.2018 18:41:30	3.71 KB	+15 B
<input type="checkbox"/>	Backup size	-bank	01.08.2018 18:41:30	1.09 MB	+4.65 KB
<input type="checkbox"/>	Backup size	-auth	01.08.2018 18:41:30	628.91 KB	+1.13 KB
<input type="checkbox"/>	Backup size	/ps-17	01.08.2018 18:41:30	22.87 MB	

serveradmin.ru

Заключение

Не понравилась статья и хочешь научить меня администрировать? Пожалуйста, я люблю учиться. Комментарии в твоём распоряжении.

Расскажи, как сделать правильно!

Данный функционал можно использовать не только для мониторинга бэкапов, но и других актуальных данных. Например, какая-то программа должна выгружать данные с определенной периодичностью. Мы можем следить за тем, как она это делает. В данной статье мы рассмотрели несколько параметров, по которым заббикс может анализировать файлы и каталоги. Но таких возможностей много. Он может, к примеру, проверять конкретный размер файла и предупреждать, если он сильно меньше или больше предыдущей версии. Настраивается это примерно так же, как здесь.

[Заказать настройку Zabbix от 500 р.](#)

Онлайн курс "Администратор Linux"

Если у вас есть желание научиться строить и поддерживать высокодоступные и надежные системы, рекомендую познакомиться с **онлайн-курсом «Администратор Linux»** в OTUS. Курс не для новичков, для поступления нужны базовые знания по сетям и установке Linux на виртуалку. Обучение длится 5 месяцев, после чего успешные выпускники курса смогут пройти собеседования у партнеров. Проверьте себя на вступительном тесте и смотрите программу подробнее по .

Помогла статья? Есть возможность отблагодарить автора