

## Агентный мониторинг ОС Linux

Перейдем к агентному мониторингу серверов с ОС Linux. В роли объекта мониторинга будет выступать виртуальная машина с ОС РЕД ОС и установленным на ней Zabbix-сервером.

Перейдите в консоль виртуальной машины и установите Zabbix-агент.

```
# sudo apt-get install zabbix-agent
```

Откройте конфигурационный файл агента для редактирования.

```
# sudo nano /etc/zabbix/zabbix_agentd.conf
```

Исправьте всего один параметр – предоставить демону права root для сбора данных:

```
AllowRoot=1
```

После этого перезапустите сервис агента.

```
# sudo /etc/init.d/zabbix-agent restart
```

На этом установка Zabbix-агента завершена.

Перейдите в систему мониторинга, и **создайте Шаблон опроса Agent Linux Monitor** (в группе User Templates). В Группе узлов сети Agent Monitoring Servers создайте новый узел сети host.linux.home (с IP-адресом 127.0.0.1), присоедините к нему Шаблон опроса Agent Linux Monitor.

После этого заполните Шаблон элементами данных, триггерами, графиками и обнаружениями. С одной стороны, шаблон для мониторинга ОС Linux похож на шаблон для ОС Windows, и почти все элементы будут повторяться. С другой стороны, возможности Zabbix-агента по предоставлению различных данных для Linux гораздо шире, и элементов данных будет значительно больше.

Начните с элементов данных для инвентаризации (см. Таблицу 7.2):

Таблица 2 - Элементы данных для инвентаризации ОС Linux

Пользовательское название элемента данных	Использованный ключ	Тип информации	Назначение
Архитектура ПО (инвентаризация)	system.sw.arch	Символ	Выполнение команды uname
Время работы системы (инвентаризация)	system.uptime	Целое положительное	Время работы в секундах (используйте Единицу измерения «uptime»)
Идентификация системы (инвентаризация)	system.uname	Символ	Возвращает архитектуру ЦПУ
Информация о ЦПУ (инвентаризация)	system.hw.cpu[0,curfreq]	Целое положительное	Возвращает значение в ГГц (используйте Единицу измерения «Hz»)
Информация о ЦПУ подробно (инвентаризация)	system.hw.cpu[0,full]	Символ	Возвращает полную информацию о ЦПУ
Информация об ОС (инвентаризация)	system.sw.os[full]	Символ	Информация об ОС (подробная информация от разработчиков, размещенная в ОС)

Поскольку при создании элементов данных, не удалось подобрать определения-аналоги в списке «Заполнение поля инвентаря узла сети» - были выбраны либо подходящие под описание, либо просто незадействованные. После создания элементов данных и просмотра инвентаризации, у Вас должно получиться следующее (рис. 18 – 19):

Элементы данных						
Все шаблоны / Agent Linux Monitor   Группы элементов данных <b>Элементы данных 6</b> Триггеры   Графики   Комплексные экраны   Прав						
<input type="checkbox"/>	Мастер	Имя ▲	Ключ	Интервал	История	Динамика
<input type="checkbox"/>	...	Архитектура ПО (инвентаризация)	system.sw.arch	24h	0	
<input type="checkbox"/>	...	Время работы системы (инвентаризация)	system.uptime	15m	0	0
<input type="checkbox"/>	...	Идентификация системы (инвентаризация)	system.uname	24h	0	
<input type="checkbox"/>	...	Информация о ЦПУ (инвентаризация)	system.hw.cpu[0,curfreq]	24h	0	0
<input type="checkbox"/>	...	Информация о ЦПУ подробно (инвентаризация)	system.hw.cpu[0,full]	24h	0	
<input type="checkbox"/>	...	Информация об ОС (инвентаризация)	system.sw.os[full]	24h	0	

Рисунок 18 – Созданные элементы данных для инвентаризации

## Инвентарные данные узла сети

Обзор Детали

Тип Linux zabbix\_machine 4.15.0-34-generic #37-Ubuntu SMP Mon Aug 27

Тип (полная детализация) 2.59 GHz

Псевдоним processor 0: GenuineIntel Intel(R) Core(TM) i5-3230M CPU @ 2.60G

ОС (полная детализация) Linux version 4.15.0-34-generic (buildd@lgw01-amd64-047) (gcc version 7.3.0 (Ubuntu 7.3.0-16ubuntu3)) #37-Ubuntu SMP Mon Aug 27 15:21:48 UTC 2018

ОС (Короткое описание) x86\_64

Этикетка 01:50:04

Отмена

Рисунок 19 – Результат инвентаризации виртуальной машины с ОС Linux

Если необходимо сопоставлять данные от разных серверов, необходимо распределить самые важные опрашиваемые параметры инвентаризации между полями Имя, Тип, ОС, Серийный номер А, Этикетка, MAC адрес А. Zabbix в сводной таблице инвентаризации выводит только эти параметры (Рис. 20):

Инвентарные данные узла сети							
Узел сети	Группа	Имя	Тип	ОС	Серийный номер А	Этикетка	MAC адрес А
host.linux.home	Agent Monitoring Servers		Linux zabbix_machine 4.15.0-34-generic #37-Ubuntu SMP Mon Aug 27			01:50:04	
host.windows.home	Agent Monitoring Servers	BEE-PC		Windows BEE-PC 10.0.17134 Майкрософт Windows 10 Pro x64		2 days, 20:26:38	

Рисунок 20 – Поля инвентаризации, доступные для сравнения

*Важно! Элементы данных, предназначенные для сбора инвентарной информации, должны иметь большой интервал обновления (повторного опроса). Значения, в зависимости от задачи, могут варьироваться от 1 часа до суток, и даже больше. Однако, время хранения таких данных может быть от очень большого (если необходимо отслеживать изменение программно-аппаратной части), до нулевого (если нужно просто владеть текущей информацией).*

**При создании Вашего первого шаблона, в процессе отладки, можно использовать короткие интервалы опроса, например, в 30 или 60 секунд. Впоследствии для инвентарных данных увеличьте интервал опроса. Это снизит и нагрузку на систему мониторинга, и размер базы данных.**

При просмотре инвентарных данных, есть удобный способ сортировать выдачу данных, запрашивая для группы устройств только одно инвентаризационное поле (Инвентаризация – Обзор – Сгруппировать по). Данный подход очень удобен при анализе информации о большой группе однотипных устройств. Например, нужно быстро узнать и предоставить серийные номера или модели коммутаторов, использующихся на предприятии (например, когда их 50-500 единиц). При условии, что сбор такой информации предусмотрен в соответствующих шаблонах (а в Инвентарных данных выводятся собранные сведения со всех опрошенных Узлов сети, независимо от Шаблона опроса), это делается в «несколько кликов мышью».

Перейдем непосредственно к мониторингу системы, созданию элементов данных, триггеров и графиков. Следующая группа элементов данных в таблице 3:

Таблица 3 - Элементы данных для мониторинга ОС Linux

Пользовательское название элемента данных	Использованный ключ	Тип информации	Назначение
Доступность хоста	См. главу 4	Числовой	Проверка доступности хоста
Доступность агента	agent.ping График, триггер	Числовой	Проверка доступности агента (не хоста, не интерфейса сетевой карты)
Использование ЦПУ системой (в %) Использование ЦПУ Zabbix (в %)	proc.cpu.util[,root] proc.cpu.util[zabbix_server,zabbix] Отобразить на одном графике	Числовой (с плавающей точкой)	Использование ЦПУ системой или процессом (не утилизация ЦПУ)
Количество процессов	proc.num График	Числовой	Количество процессов, запущенных в системе
Нагруженность процессора	system.cpu.load[,avg1] system.cpu.load[,avg5] system.cpu.load[,avg15] Отобразить на одном графике, триггер	Числовой (с плавающей точкой)	Load Average
Утилизация процессора	system.cpu.util[,system,avg1] График, триггер	Числовой (с плавающей точкой)	Загруженность, или утилизация процессора (в процентах)
Использование файла подкачки, (в %)	system.swap.size[,pfree] График	Числовой (с плавающей точкой)	Доля свободных ресурсов файла подкачки
Оперативная память, всего / используется	vm.memory.size[total] vm.memory.size[used] Отобразить на одном графике	Числовой	Размер оперативной памяти. Под доступной памятью понимается сумма inactive + cached + free

В результате создания всех элементов данных, графиков и триггеров к ним – у Вас должно получиться следующее (Рис. 21 – 22):

Элементы данных						
Все шаблоны / Agent Linux Monitor   Группы элементов данных <b>Элементы данных 17</b> Триггеры   Графики 6   Комплексные экраны   Правила обнаружения 2						
<input type="checkbox"/>	Мастер	Имя ▲	Триггеры	Ключ	Интервал	История   Динамика
<input type="checkbox"/>	...	Архитектура ПО (инвентаризация)		system.sw.arch	24h	0
<input type="checkbox"/>	...	Время работы системы (инвентаризация)		system.uptime	15m	0   0
<input type="checkbox"/>	...	Доступность агента		agent.ping	1m	60d   60d
<input type="checkbox"/>	...	Доступность узла		icmping[,3,...]	1m	60d   60d
<input type="checkbox"/>	...	Идентификация системы (инвентаризация)		system.uname	24h	0
<input type="checkbox"/>	...	Информация о ЦПУ (инвентаризация)		system.hw.cpu[0,curfreq]	24h	0   0
<input type="checkbox"/>	...	Информация о ЦПУ подробно (инвентаризация)		system.hw.cpu[0,full]	24h	0
<input type="checkbox"/>	...	Информация об ОС (инвентаризация)		system.sw.os[full]	24h	0
<input type="checkbox"/>	...	Использование ЦПУ Zabbix		proc.cpu.util[zabbix_server,zabbix]	1m	60d   60d
<input type="checkbox"/>	...	Использование файла подкачки, (в %)		system.swap.size[,pfree]	1m	60d   60d
<input type="checkbox"/>	...	Количество процессов		proc.num	1m	60d   60d
<input type="checkbox"/>	...	Нагруженность процессора за 1 минуту		system.cpu.load[,avg1]	1m	60d   60d
<input type="checkbox"/>	...	Нагруженность процессора за 5 минут		system.cpu.load[,avg5]	1m	60d   60d
<input type="checkbox"/>	...	Нагруженность процессора за 15 минут		system.cpu.load[,avg15]	1m	60d   60d
<input type="checkbox"/>	...	Оперативная память, всего		vm.memory.size[total]	1m	60d   60d
<input type="checkbox"/>	...	Оперативная память, используется		vm.memory.size[used]	1m	60d   60d
<input type="checkbox"/>	...	Утилизация процессора		system.cpu.util[,system,avg1]	1m	60d   60d

Рисунок 21 – Созданные элементы данных для шаблона опроса ОС Linux

Графики	
Все шаблоны / Agent Linux Monitor   Группы элементов данных   Элементы данных 17   Триггеры <b>Графики 6</b>	
<input type="checkbox"/>	Имя ▲
<input type="checkbox"/>	Load Average CPU
<input type="checkbox"/>	Доступность узла и агента
<input type="checkbox"/>	Использование CPU Zabbix
<input type="checkbox"/>	Количество процессов
<input type="checkbox"/>	Оперативная память
<input type="checkbox"/>	Утилизация процессора

Рисунок 22 – Созданные графики для шаблона опроса ОС Linux

**Создайте триггеры на свое усмотрение**, по аналогии с шаблоном для мониторинга ОС Windows (обязательно используйте и протестируйте комплексный триггер с использованием nodata).

Теперь необходимо **создать два низкоуровневых обнаружения** – для обнаружения жестких дисков и сетевых интерфейсов. Воспользуйтесь шаблоном опроса ОС Windows, но для обнаружения жестких дисков внесите несколько отличий.

Для обнаружения жестких дисков, используйте 2 прототипа элементов данных - `vfs.fs.size[{#FSNAME},used]` и `vfs.fs.size[{#FSNAME},total]`, и поместите их на один прототип графика. Для прототипа триггера, сигнализирующего об отсутствии свободного места на жестком диске, используйте конструкцию

```
{ Agent Linux Monitor:vfs.fs.size[{#FSNAME},used].last()}/{ Agent Linux Monitor:vfs.fs.size[{#FSNAME},total].last()}>0.9
```

По сути, прототип триггера будет вычислять отношение занятого места к общему, и сравнивать с пороговым значением срабатывания 0.9, то есть сигнализировать о том, что занято более 90% свободного места.

После обнаружения жестких дисков, Zabbix отображает все директории корневого каталога, нам же интересна только директория «/». **Создайте фильтр и исключение для всех букв алфавита** (директория / будет обнаруживаться, а папки нет). Регулярное выражение для фильтра – Результат ЛОЖЬ = `[a-z]`

**Шаблон готов** – удостоверьтесь, что обнаружения, прототипы триггеров и графиков работают корректно. Для просмотра доступны графики и инвентарные данные, настроены триггеры и обнаружения – используйте комплексный экран для представления результатов.

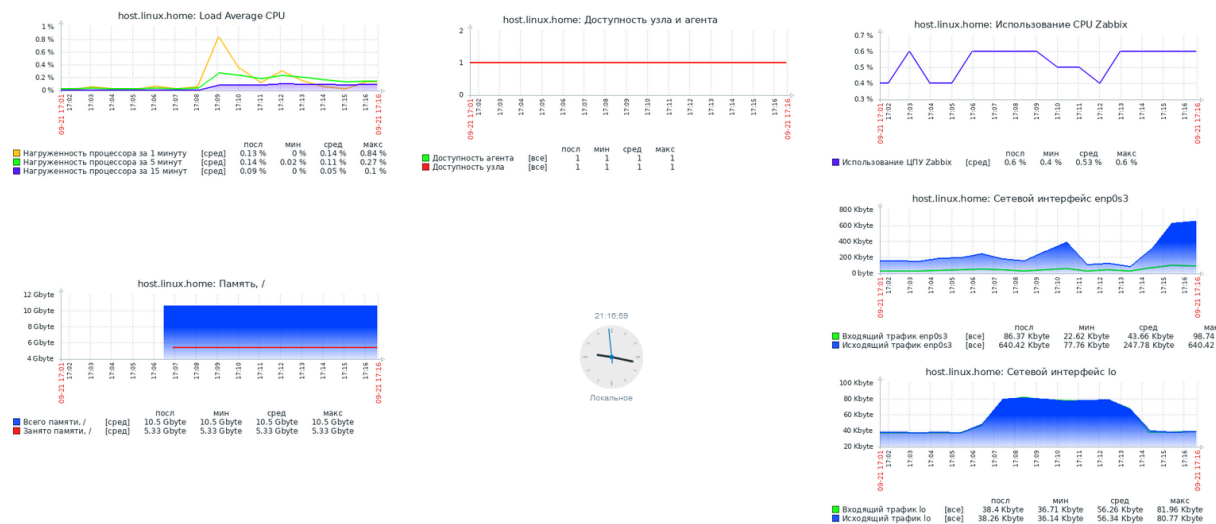


Рисунок 23 – Пример комплексного экрана для хоста host linux.home