

## Контроль/управление виртуальными машинами на базе мониторинговых событий

*Федотов Андрей Сергеевич*

*Нижегородский государственный университет им. Н.И. Лобачевского  
Магистрант*

*Штанюк Антон Александрович*

*Нижегородский государственный университет им. Н.И. Лобачевского  
Доцент*

### Аннотация

В статье дан краткий обзор и обоснования по выбору технологий и предполагаемый способ решения проблемы в обеспечении работы веб-сервиса, работающего на базе IaaS платформы. Приводятся аргументы и сравнительные характеристики по использованию мониторинговых систем. Приводятся выводы о возможности реализации способа решения проблемы.

**Ключевые слова:** веб-сервис, IaaS, OpenStack, Zabbix, системы мониторинга, системы виртуализации, виртуальные машины, инфраструктура.

### Monitoring/management of virtual machines based on the monitoring events

*Fedotov Andrei*

*Nizhny Novgorod State University  
Student*

*Shtanyuk Anton*

*Nizhny Novgorod State University  
Associate professor*

### Abstract

The article provides short view of technologies and describes the possible way of solving problem to provide stable web service what is working on the IaaS platform. The article includes comparing of monitoring systems and explanation of possibility realizing of the solution.

**Keywords:** web service, IaaS, OpenStack, Zabbix, monitoring systems, virtualization systems, virtual machines, infrastructure.

### Введение

В наши дни все чаще и чаще мы слышим о облачных вычислительных системах, которые имеют многие преимущества по сравнению с традиционными способами хранения и обработки информации, что они

безопасны, отказоустойчивы и склонны решить проблемы традиционного способа организации ИТ инфраструктуры предприятия с "железными" серверами, находящимися на территории организации. Можно выделить несколько основных преимуществ и недостатков подобного рода систем, но они нивелируются тенденциями и требованиями к решению конкретных задач.

С развитием интернет технологий в качестве облачных систем и систем управления облачными инфраструктурами все более популярным становится принцип предоставления инфраструктуры как сервиса (IaaS) [1], на базе мощностей, на которых можно развернуть свою собственную архитектуру под нужды конкретной организации. В результате такого подхода клиент сам волен распоряжаться выделением ресурсов под конкретные нужды. Облачный сервис в этом случае, в основном, отвечает за бесперебойную работу вычислительных ресурсов, но не самих приложений и виртуальных машин, расположенных на нем. В этом случае ответственность за бесперебойную работу приложений возлагается на организацию и влечет за собой решение проблем с пропускной способностью в предоставлении сервиса, утилизации, отказоустойчивости.

Подобного рода задачи в классической инфраструктуре решаются выделением дополнительного сервера для увеличения пропускной способности либо выводом его из эксплуатации. В худшем случае сервер продолжает работать, но нагрузка на него не распределяется, либо распределяется, но очень малая в часы пик. Все эти процедуры требуют вовлечения специалистов со стороны, либо личного штата сотрудников по закупке дополнительного оборудования, настройке и сопровождению как программного обеспечения, так и аппаратного. Процедуры также требуют времени и значительных финансовых затрат для предприятия.

Цель работы состоит в поиске способа автономного контроля/управления виртуальными машинами, предоставляющих программный сервис на базе мониторинговых событий их состояний. Контролируя и своевременно реагируя на те или иные события, можно предотвратить либо уменьшить возможность возникновения экстренной ситуации и тем самым обеспечить бесперебойную работу программного сервиса. Это должно повлечь за собой экономию финансовых затрат и средств организации, а также предоставит гибкость и отзывчивость на изменение количества обрабатываемых операций тем самым увеличит утилизацию вычислительных ресурсов.

В работе рассмотрен способ по сбору и анализу мониторинговых событий, принятие на его базе решений, а также рассмотрены мониторинговые системы. Сделаны выводы о возможности реализации приложения и конфигурации обеспечивающей работу сервиса при определенных условиях.

### **IaaS для исследования**

В настоящее время существует несколько основных представителей платформ по организации IaaS, так или иначе предлагающие примерно схожие возможности [2]. Однако, когда речь заходит о стоимости программного обеспечения OpenStack занимает выигрышную позицию.

Платформа OpenStack – это комплекс программного обеспечения, состоящий из набора компонент/сервисов, реализующий функции облачной платформы с открытым исходным кодом.

OpenStack предоставляет из себя программно-определяемый центр обработки данных с простым и унифицированным доступом к различным вычислительным ресурсам, сетям передачи данных, системам хранения данных, а также дополнительным сервисам, таким как:

- балансировщики нагрузки;
- средства защиты (межсетевой экран, группы безопасности);
- объектное и блочное хранение данных.

OpenStack является представителем IaaS системы и взаимодействует с базовой инфраструктурой через открытые или предоставляемые поставщиком оборудования драйверы, предоставляет управление идентификацией, оркестрацией [3]. Одна из черт OpenStack - это учет ресурсов, управление и его настройка проводимая в той же программной среде (Horizon), либо через API, что очень удобно для человека, мало посвящённого в его работу. В отличии от других систем OpenStack не является гипервизором, но он поддерживает широко используемые системы с коммерческим и открытым исходным кодом [4]:

- Ironic;
- KVM;
- LXC;
- QEMU;
- VMware ESX/ESXi;
- Xen (используя libvirt);
- XenServer;
- Hyper-V;
- PowerVM;
- UML;
- Virtuozzo;
- zVM.

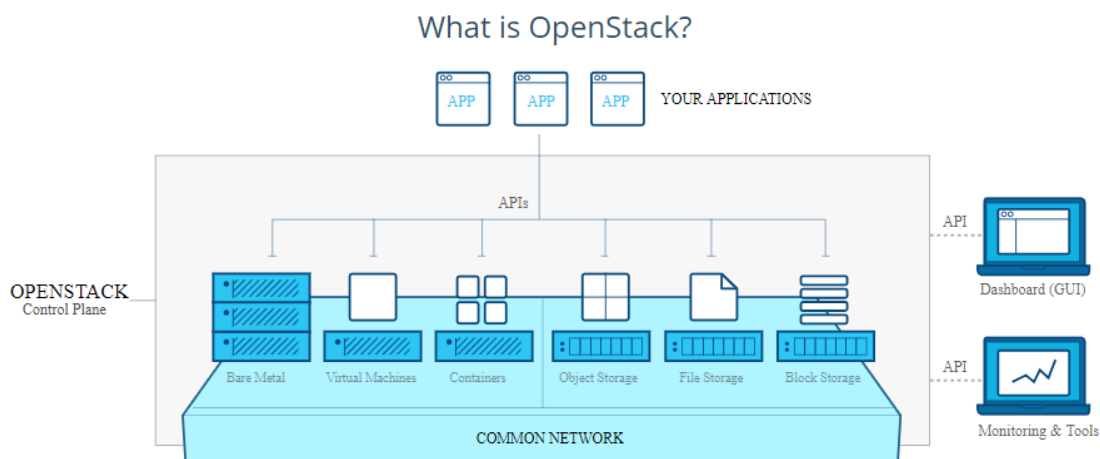


Рисунок 1. Общая архитектура OpenStack

OpenStack существенно сокращает время на развертывание и ввод в эксплуатацию тех или иных приложений. Как следствие, он очень сильно популярен у разработчиков для облачных сред и систем. В последнее время OpenStack становится заметным и в проектах с использованием SDN контроллеров, что в свою очередь, позволяет контролировать и управлять размерами и потоками трафика на программном уровне за счет реализуемых сетевых функций, предоставляемых провайдерами сетевого оборудования.

OpenStack активно развивается, о чем свидетельствует большая вовлеченность в проект не малоизвестных компаний, среди которых: at&t, Ubuntu, Hewlett Packard Enterprise, IBM, Intel, Rackspace, RedHat, SUSE, Ericson, Huawei, Cisco, NEC, ZTE, Deutsche Telekom и многие другие.

Рассмотрим основные компоненты (проекты) OpenStack:

- Nova (Compute) – это проект OpenStack, основная цель которого в предоставлении виртуальных машин: создание и управление. Для обеспечения работы базового функционала требуется наличия запущенных и настроенных проектов Keystone, Glance и Neutron.
- Glance – проект OpenStack для хранения, получения, управления образами виртуальных машин.
- Neutron – проект OpenStack предназначенный для создания, управления сетевыми подключениями и ресурсами. Обеспечивает сетевое подключение между виртуальными сетевыми интерфейсами и устройствами, и сетевыми картами физических серверов, на базе которых виртуальные машины запущены.
- Cinder – проект OpenStack по работе с блочными устройствами, позволяет представить множество блочных устройств в виде одного набора дискового пространства. Основная цель предоставление дискового пространства для проекта Nova.
- Keystone – проект OpenStack выполняющий авторизацию и аутентификацию пользователей и предоставляющий определенный набор ресурсов и средств по работе с OpenStack.

- Horizon – это панель управления по работе с OpenStack. Предоставляет веб интерфейс по управлению и настройке сервисов OpenStack, настройке виртуальных машин, работе с сетевыми устройствами, дисковым пространством и учетом вычислительных ресурсов.

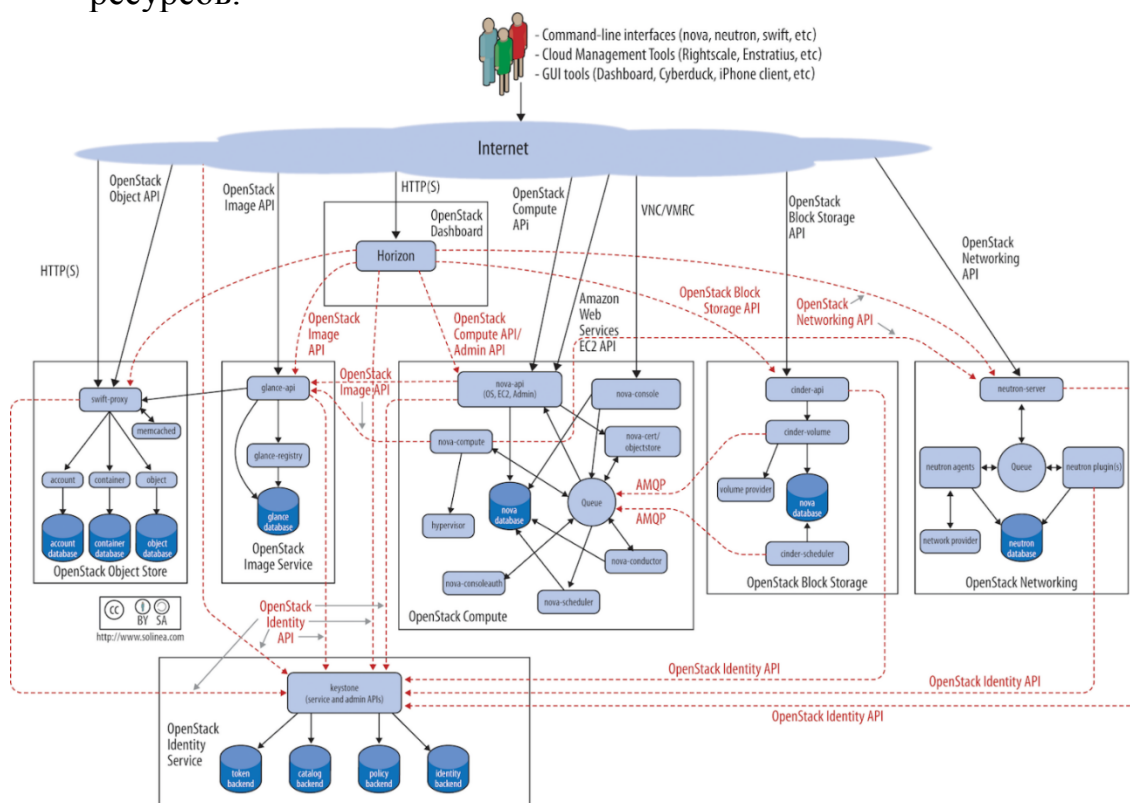


Рисунок 2. Детальная архитектура OpenStack

Использование платформ по типу OpenStack позволяет разумно и своевременно распределять ресурсы всей инфраструктуры в целом. Например, они могут быть расширены за счет добавления новых экземпляров приложений, использующие различные типы виртуальных машин и различные приложения на них. В случае, если эти виртуальные машины предназначены для предоставления одного сервиса, они могут быть объединены и будут предоставлять этот сервис через одну точку доступа при помощи распределителей нагрузки – балансировщиков нагрузки. Подобного рода приложения организуют и предоставляют кластерную/отказоустойчивую архитектуру.

### Системы мониторинга

Для решения проблемы по сбору данных состояний сервиса и его компонентов предполагается использование системы удаленного мониторинга. Системы удаленного мониторинга базируются на принципах клиент-серверного взаимодействия, где взаимодействие осуществляется с помощью передачи данных по сети. Используемые протоколы так же варьируются в зависимости от реализуемой задачи.

Так по примеру использования широко известного протокола SNMP [5] (широко используется для мониторинга инфраструктуры сети) можно реализовать систему обмена сообщениями (событиями) между клиентской и серверной частью системы. Существуют собственные протоколы и способы по сбору данных с наблюдаемых объектов.

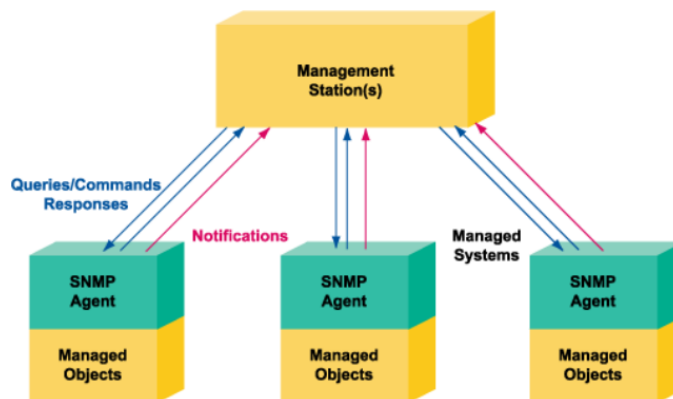


Рисунок 3. Концепт структуры SNMP

Существующие системы мониторинга можно разделить на системы, реализующие активный и пассивный мониторинг.

В случае пассивного мониторинга предполагается использование агентов приложений или сервисов. Агенты устанавливаются на объект мониторинга, локально собирают оперативную информацию и отправляют данные серверу для дальнейшей обработки. Примером может быть сбор данных по загрузке центрального процессора, оперативной памяти, занимаемого места на диске.

В случае активного мониторинга предполагается что сам сервер производит опрос текущего состояния наблюдаемого объекта мониторинга. Как правило это использование протоколов по проверке доступности сервиса, либо опрос состояния балансировщика нагрузки в случае с предоставлением высоконагруженного сервиса.

Таким образом, система мониторинга может выполнять сбор данных как на основе событий так и на основе получаемых параметров состояния компонентов.

Для выбора подходящей системы мониторинга в рамках исследования рассмотрим наиболее популярные системы[6]:

- Nagios;
- Zabbix;
- Shinken;
- Icinga.

Для проведения сравнительной характеристики были выделены основные требования:

1. Способ распространения.
2. Наличие агентной модели сбора данных.
3. Поддержка мониторинга на базе SNMP/ICMP.

4. Поддержка уведомлений.
5. Наличие веб интерфейса для администрирования.
6. Поддержка сценариев на основе событий.

По выработанным требованиям был проведен анализ и выявлено что Shinken и Incinga являются либо потомками Nagios проекта, либо ответвлениями от него и имеют схожие возможности и могут использовать те же расширения.

Таблица 1. Сравнение систем мониторинга.

	Способ распространения	Агентная модель сбора данных	SNMP/ICMP	Уведомления	Веб интерфейс для администрирования	Сценарии/обработчики событий
Nagios	GNU General Public License Version 2	+	+	+	-	+
Zabbix	GNU General Public License Version 2	+	+	+	+	+
Shinken	Affero General Public License	+	+	+	+	+
Incinga	GNU General Public License Version 2	+	+	+	+	+

Анализ показал, что системы мониторинга сходны по выполняемым функциям. Почти все они соответствуют заявленным требованиям и возможностям, имеют реализацию агентной модели сбора данных и поддерживают базовую работу на основе протоколов SNMP и ICMP, имеют механизмы поддержки сценариев.

В результате выбор был сделан в пользу системы мониторинга Zabbix, т.к. она является свободным программным обеспечением, распространяется бесплатно, имеет хорошую документацию и активное сообщество пользователей.

### Выводы

В рамках проведенного анализа, автономный контроль/управление виртуальными машинами, предоставляющих программный сервис на базе мониторинговых событий их состояний, представляется возможным при условиях:

1. Использование внешней системы удаленного мониторинга, настроенной на анализ и сбор метрик с анализируемых виртуальных машин. Данные виртуальные машины будут представлять веб-сервис.
2. Использование платформы IaaS на базе, которой будет поднят веб-сервис. Данная платформа будет предоставлять API для управления состоянием виртуальных машин.
3. Использования отдельного сервиса (приложения) для своевременного реагирования на те или иные события приходящие со стороны системы удаленного мониторинга для автоматического

реагирования и управления виртуальными машинами через API IaaS платформы предоставляющих веб-сервис.

### **Библиографический список**

1. Батура Т.В., Мурзин Ф.А., Семич Д.Ф. Облачные технологии: основные модели, приложения, концепции и тенденции развития // Программные продукты и системы. 2014. №3. С. 107. URL: <https://cyberleninka.ru/article/n/oblachnye-tehnologii-osnovnye-modeli-prilozheniya-kontseptsii-i-tendentsii-razvitiya-1>
2. Официальный сайт RightScale. URL: <https://www.rightscale.com/cloud-comparison-tool>, свободный.
3. ETSI GS NFV-IFA 031. Network Functions Virtualisation (NFV) Release 3; Management and Orchestration; Report on management of NFV-MANO and automated deployment of EM and other OSS functions. - 2018-01.: ETSI, 2018. URL: [https://www.etsi.org/deliver/etsi\\_gs/NFV-IFA/001\\_099/031/03.01.01\\_60/gs\\_NFV-IFA031v030101p.pdf](https://www.etsi.org/deliver/etsi_gs/NFV-IFA/001_099/031/03.01.01_60/gs_NFV-IFA031v030101p.pdf)
4. Официальный сайт системы OpenStack. What is OpenStack. URL: <https://opensource.com/resources/what-is-openstack>, свободный.
5. Сильнов Д. С. Актуальность современных систем удаленного мониторинга вычислительных ресурсов // Известия Российского государственного педагогического университета им. А.И. Герцена. 2011. С. 55-59. URL: <https://cyberleninka.ru/article/n/aktualnost-sovremennyh-sistem-udalennogo-monitoringa-vychislitelnyh-resursov>
6. Краснопер Д.И. Системы мониторинга состояния сети и ее компонентов // Новые информационные технологии в автоматизированных системах. 2010. С. 209–211. URL: <https://cyberleninka.ru/article/v/sistemy-monitoringa-sostoyaniya-seti-i-eyo-komponentov>