

УДК 004.89

М.С. Лабазов, Н.С. Лесная

Харьковский национальный университет радиоэлектроники, Харьков

МЕТОДЫ И МОДЕЛИ КОНФИГУРАЦИИ СЕРВЕРОВ В CLOUD СИСТЕМАХ ДЛЯ ЗАДАЧ ПРОГНОЗИРОВАНИЯ НАГРУЗКИ ВЫЧИСЛИТЕЛЬНОЙ СЕТИ

В статье проанализировано текущее состояние облачных вычислений, сформулирована задача автоматического конфигурирования серверов Cloud систем. Произведена оценка различных методов для решения этой задачи: SSH, EC2-init, Chef Opscode с точки зрения удобства использования, универсальности подхода и надежности. Для рассмотренных методов оценены преимущества и недостатки. Сделан вывод о необходимости автоматизации конфигурирования серверов в Cloud системах, как обязательного требования развития технологии облачных вычислений.

Ключевые слова: облачные вычисления, методы развертывания, Chef Opscode, EC2-init, SCM, SSH.

Введение и постановка задачи

Множество ИТ экспертов полагают, что в недалеком будущем облачные вычисления войдут в топ-список технологических трендов, что обусловлено имеющимися преимуществами этой технологии.

Во-первых, это высокий уровень виртуализации. Пользователь, развернув свое приложение, не задумывается о таких проблемах, как аппаратные средства, их поддержка, конкретный софт, установленный на машине. Он даже не задумывается о том, на какой именно или даже на каких именно машинах исполняется его приложение.

Возможность запуска множества копий приложения на многих виртуальных машинах представляет преимущества масштабируемости: количество экземпляров приложения способно практически мгновенно увеличиваться по требованию, в зависимости от нагрузки.

И, наконец, дата-центры управляются профессиональными специалистами, обеспечивающими круглосуточную поддержку функционирования виртуальных машин. И даже если физическая машина «фрухнет», благодаря распределению приложения на множество копий оно все равно продолжит свою работу. Это создает определенный высокий уровень надежности и отказоустойчивости функционирования системы.

Облачные вычисления имеют множество преимуществ по сравнению с обычным хостингом сервисов и веб-приложений, но до сих пор не решены некоторые проблемы.

Одна из таких проблем – это автоматическая конфигурация серверов, которые динамически запускаются в облаке. Очень часто приходится выполнять много ручной работы, чтобы установить и развернуть свое приложение на запущенном сервере.

Уже существует довольно большое количество отдельных решений этой проблемы, но по отдельности они не могут выполняться во всех условиях облачных серверов. Например, некоторые

решения применимы для Amazon cloud, но не подходят для Azure, некоторые решения не кроссплатформенны.

Таким образом, создание платформ для облачных вычислений является актуальной задачей. Одна из важных составляющих этой проблемы — конфигурирование серверов. Существует несколько подходов к ее решению: SSH конфигурация, SCM (Software Configuration Management), EC2-init.

Основная часть

Все способы автоматической конфигурации серверов, перечисленные ниже, имеют право на жизнь и могут быть использованы в различных условиях. К сожалению, у всех них есть свои недостатки, но если совместить их использование, можно получить хороший инструмент для решения задач конфигурации.

SSH. SSH (англ. Secure SHell — «безопасная оболочка») — сетевой протокол сеансового уровня, позволяющий производить удалённое управление операционной системой и туннелирование TCP-соединений (например, для передачи файлов). Схож по функциональности с протоколами Telnet и rlogin, но, в отличие от них, шифрует весь трафик, включая и передаваемые пароли. SSH допускает выбор различных алгоритмов шифрования. SSH-клиенты и SSH-серверы доступны для большинства сетевых операционных систем. SSH позволяет безопасно передавать в незащищённой среде практически любой другой сетевой протокол. Таким образом, можно не только удалённо работать на компьютере через командную оболочку, но и передавать по зашифрованному каналу звуковой поток или видео (например, с веб-камеры). Также SSH может использовать сжатие передаваемых данных для последующего их шифрования, что удобно, например, для удалённого запуска клиентов X Window System. Большинство хостинг-провайдеров за определенную плату предоставляют клиентам доступ к их домашнему каталогу по SSH.

Поддержка SSH реализована во всех UNIX-подобных системах, и на большинстве из них в числе стандартных утилит присутствуют клиент и сервер ssh. Существует множество реализаций SSH-клиентов и для не-UNIX ОС. Большую популярность протокол получил после широкого развития анализаторов трафика и способов нарушения работы локальных сетей, как альтернативное небезопасному протоколу Telnet решение для управления важными узлами.

Для работы по SSH нужен SSH-сервер и SSH-клиент. Сервер прослушивает соединения от клиентских машин и при установлении связи производит аутентификацию, после чего начинается обслуживание клиента. Клиент используется для входа на удаленную машину и выполнения команд.

Для соединения сервер и клиент должны создать пары ключей — открытых и закрытых — и обменяться открытыми ключами. Обычно используется также и пароль.

EC2-init. Amazon Elastic Compute Cloud (Amazon EC2) — веб-сервис, который предоставляет вычислительные мощности в облаке. Сервис входит в инфраструктуру Amazon Web Services. Простой веб-интерфейс сервиса позволяет получить доступ к вычислительным мощностям и настроить с минимальными затратами ресурсы. Он предоставляет пользователям полный контроль над вычислительными ресурсами, а также доступную среду для работы. Сервис сокращает время, необходимое для получения и загрузки нового сервера.

Данный сервис имеет свой особый способ конфигурации, который становится все более популярным и уже начал применяться в различных Cloud системах, таких как Nimbula Director.

Способ заключается в том, что при запуске вычислительной машины в облаке, на нее сразу передаются пользовательские данные, с помощью которых происходит конфигурация.

Amazon EC2 ограничивает размер пользовательских данных в 16 Кбайт. Если запуск инструкции больше, чем этот предел, вы можете написать скрипт для пользовательских данных, который загружает их в полном объеме из хранилища в другом месте, как, например, делает S3, и запускает их.

Предоставляемый пользовательский интерфейс представляет собой удобный инструмент написания скриптов для передачи пользовательских данных, а также установки и настройки программного обеспечения. Сценарий может быть написан на любом языке, который поддерживает Shabang (механизм для запуска программы). Этот список включает в себя такие языки как Perl, Python, Ruby, Tcl, AWK, SED, Vim или любой другой язык. Вы можете найти механизм для работы с пользовательскими данными, установленным на образе виртуальной системы.

Если вы хотите использовать другой язык для пользовательских данных, вам необходимо будет создать скрипт, написанный на Bash, который уста-

новит компилятор выбранного языка, а затем запустить этот скрипт.

Создание новой EC2 вычислительной машины часто требует передачи личной информации, такой как EC2 ключи и сертификаты (например, для AWS API вызовов). Если передать секретную информацию в пользовательских данных, полный доступ к ним будет предоставлен для любого пользователя или процесса, работающего на этой машине.

Существует простой способ прочитать пользовательские данные с вычислительной машины после ее запуска, так что любой, кто имеет доступ к ней, может просто отправить запрос на http://имя_сайта.ком/latest/user-data и получить всю информацию, введенную пользователем.

Chef Opscode. Использование SCM в IT инфраструктуре компании помогает избавиться от множества проблем, присущих традиционному подходу, и автоматизировать часть операций, выполняемых IT-отделом, таких как: внесение изменений на группе серверов, возможность вернуться к любому предыдущему состоянию системы в случае неудачного/частично удачного размещения, контроль действиями администраторов и многих других. В данный момент наиболее распространённое SCM программное обеспечение — это Vcfg2, Cfengine, Chef и Puppet.

Chef представляет собой полнофункциональный инструмент управления конфигурацией. Роли и рецепты используются для описания того, как должны быть настроены серверы. Chef работает, позволяя писать рецепты, которые описывают, какие роли сервера (например, Apache, MySQL, или Hadoop) должны быть установлены на этой вычислительной машине. Эти рецепты можно описать с помощью различных инструментов, которые должны выполнять ряд команд.

Недостатки существующих решений. Все вышеперечисленные инструменты позволяют полностью решать задачу конфигурации. Но все эти решения не могут одновременно самостоятельно удовлетворить всем входящим условиям задач конфигурации. Каждый из них может только частично удовлетворить некоторым условиям.

Поэтому на данный момент стоит задача в создании универсального решения, которое сможет выполнять конфигурацию любых серверов в любой Cloud системе.

Есть два пути реализации данного приложения. Первый способ — создать такое приложение с нуля, полностью написав его исходный код. Такой выход займет много времени, а также будут огромные риски с надежностью новой системы.

Второй способ — это создать новую систему, которая будет использовать уже готовые решения. Таким образом, комбинация уже созданных компонентов будет отличаться проверенной надежностью и не займет много времени, чтобы написать такую систему.

Выводы

Проблема автоматической конфигурации до сих пор очень актуальна, потому что без этого будет нарушено одно из шести правил существования Cloud системы. Большинство разработчиков и администраторов стремятся исключить ручную установку программной среды путем автоматизации данного процесса.

На данный момент существует большое количество решений, с помощью которых решается проблема автоматической конфигурации серверов на вычислительных машинах запущенных в Cloud системе.

По отдельности каждый способ представляет собой решение, которое можно использовать только для конкретно поставленных задач. К сожалению, до сих пор не создано универсальное решение, которое можно использовать для широкого круга задач.

Сейчас ведется работа над новым решением, которое объединит в себе несколько уже существующих, и будет пригодно для использования для

большого круга задач в различных операционных средах и Cloud системах.

Список литературы

1. Chang, J. Dean. *A distributed storage system for structured data.* – Seattle, USA, 2006. – Pp. 205-218.
2. Dean, J. and Ghemawat, S. *MapReduce: Simplified data processing on large clusters.* – San Francisco, USA, 2004. – Pp.137-150.
3. Barroso, L. A., Dean, J., and Urs Hölzle, U. *Web search for a planet: The Google cluster architecture.* – Seattle, USA, 2003. – Pp. 22-28.
4. *Hadoop Map/Reduce Tutorial* [Электронный ресурс]. – Режим доступа: http://hadoop.apache.org/common/docs/current/mapred_tutorial.html, свободный.
5. *CloudEra* [Электронный ресурс]. – Режим доступа: <http://www.cloudera.com/>, свободный.

Поступила в редколлегию 3.02.2012

Рецензент: д-р. техн. наук, проф. Е.П. Путятин, Харьковский национальный университет радиоэлектроники, Харьков.

МЕТОДИ І МОДЕЛІ КОНФІГУРАЦІЇ СЕРВЕРІВ У CLOUD СИСТЕМАХ ДЛЯ ЗАДАЧ ПРОГНОЗУВАННЯ НАВАНТАЖЕННЯ ОБЧИСЛЮВАЛЬНОЇ МЕРЕЖІ

М.С. Лабазов, Н.С. Лесна

У статті проаналізовано поточний стан хмарних обчислень, сформульована задача автоматичного конфігурування серверів Cloud систем. Проведено оцінку різних методів для вирішення цієї задачі: SSH, EC2-init, Chef Opscode з точки зору зручності використання, універсальності підходу та надійності. Для розглянутих методів оцінені переваги і недоліки. Зроблено висновок про необхідність автоматизації конфігурування серверів в Cloud системах, як обов'язкової вимоги розвитку технології хмарних обчислень.

Ключові слова: хмарні обчислення, методи розгортання, Chef Opscode, EC2-init, SCM, SSH.

METHODS AND MODEL OF SERVER CONFIGURATION IN CLOUD SYSTEMS FOR LOAD FORECASTING OF COMPUTER NETWORK

M.S. Labazov, N.S. Lesnaya

The article analyzes the current state of cloud computing, formulates the problem of automatic configuration of servers for Cloud systems. Different methods for solving this problem are taken: SSH, EC2-init, Chef Opscode in terms of ease of use, versatility and reliability of the approach. The advantages and disadvantages for selected methods are evaluated. The conclusion about the need to automate the configuration of servers in the Cloud systems as a mandatory requirement of cloud computing is given.

Keywords: cloud computing, deployment methods, Chef Opscode, EC2-init, SCM, SSH.