ИСПОЛЬЗОВАНИЕ СИСТЕМЫ УПРАВЛЕНИЯ КОНФИГУРАЦИЯМИ ANSIBLE КАК ИНСТРУМЕНТА ДЛЯ УПРАВЛЕНИЯ НЕСКОЛЬКИМИ WEB-СЕРВЕРАМИ

О.В. Бобков, Т.А. Пулко, А.О. Хмельницкий
Кафедра защиты информации, Белорусский государственный университет информатики и радиоэлектороники
Минск, Республика Беларусь
E-mail: a.hmelnitsky@itransition.com, olegbobkov@gmail.com, naig@tut.by

Проблема управления большим количеством систем далеко не нова, но особенно острой она стала при распространении кластеров и облачных сервисов. Для ее решения появились разнообразные инструменты, и каждый делает это по-своему. Вот здесь и появляется интерес к системам управления конфигурацией, позволяющим настраивать серверы не руками, а программным способом. Системы настраиваются быстро и с меньшим количеством возможных ошибок, владелец получает отчет о работе системы.

Введение

Информация, получаемая в процессе управления конфигурациями, является фундаментальной для поставки и сопровождения IT-услуг. Данный процесс является ключевым в создании и сопровождении общей модели инфраструктуры, которая может использоваться во всех IT-процессах организации. Система управления конфигурациями - необходимый инструмент для понимания и формирования принципов оказания IT-услуг. Современные системы управления конфигураций по сути стремятся к тому, чтобы в полной мере реализовать принцип Infrastructure-as-a-Code, в соответствеии с которым вся существующая IT-инфраструктура, машины, их конфигурация, связи между ними и так далее могут быть описаны одним или несколькими формальными файлами, а дальше это уже дело системы управления конфигурацией — воплотить описанную конфигурацию в жизнь. За более чем 20 лет активного развития систем управления конфигурацией (первая версия CFEngine, самой ранней системы управления конфигурации из живущих, появилась в 1993 году) было создано множество различных систем управления конфигурацией программного обеспечения, которые хотя и решают одну и ту же задачу, разделяют общие принципы, но вместе с тем сильно различаются между собой.

I. Системы управления конфигурациями

Наиболее популярными на сегодняшний день являются такие системы как Chef, Puppet, SaltStack и Ansible.

Puppet наиболее полон с точки зрения возможных действий, модулей и пользовательских интерфейсов, представляя полную картину ЦОД, охватывая почти каждую операционную систему и предоставляя утилиты для всех основных ОС. Начальная установка относительно проста, требует развертывания головного сервера и клиентских агентов на каждой управляемой системе.

Интерфейс командной строки позволяет загружать и устанавливать модули с помощью команды puppet. Затем требуются изменения в конфигурационных файлах, необходимые для настройки модуля под требуемую задачу, а клиенты, которые должны получить инструкции, получат их при следующем обращении к головному серверу, или через запрос от сервера, инициирующий процесс изменения немедленно.

Chef похож на Puppet с точки зрения общей концепции, в нем также имеется головной сервер и агенты, установленные на управляемых узлах. В дополнение к головному серверу, установка Chef также требует рабочей станции, для управления им. Агенты могут быть установлены с рабочей станции с помощью утилиты knife, которая использует протокол SSH для развертывания, облегчая бремя установки. После этого, управляемые узлы аутентифицируются с головным при помощи сертификатов.

В отличие от Puppet, у Chef пока нет хорошо реализованной функции push, хотя доступна бета-версия кода. Это означает, что агентов должны быть настроены на периодическую синхронизацию с головным сервером, и немедленное применение изменений невозможно. Пользовательский веб-интерфейс функционален, но не предоставляет возможности модифицировать конфигурации. Он не так полон, как веб-интерфейс Puppet Enterprise, уступает в построении отчетов и некоторых других функциях, но позволяет вести учет оборудования и организацию узлов.

Как и у Puppet, у Chef большой набор модулей и рецептов настроек, преимущественно на ruby. По этой причине, Chef хорошо подходит для инфраструктур, ориентированных на разработку.

Salt может связываться с клиентами по протоколу SSH, но масштабируемость значительно расширяется за счет клиентских агентов. Также, Salt включает асинхронный файловый сер-

вер для ускорения обслуживания агентов, позволяя создавать хорошо масштабируемые системы.

Веб-интерфейс Salt слишком новый и не полный, как пользоветельские интерфейсы других систем. С его помощью можно просматривать системные журнал сообщений и статус управляемых узлов, а также имеется возможность выполнять на них команды. Этот иснтурмент активно разрабатывается, и обещает значительные улучшения, но пока это голый «скелет» и содержит много ошибок.

Ansible фокусируется на оптимизации и скорости, и не требует установки агентов на управляемые узлы — все функции производятся по SSH. Ansible написан на python, в отличие от Puppet и Chef, основанных на ruby.

Установка Ansible может быть выполнена путем клонирования Git-репозитория на головной сервер. Вслед за этим, узлы, над которыми требуется управление добавляются в конфигурацию Ansible, и авторизованные ключи SSH «привязываются» к каждому узлу, относясь к пользователю от имени которого будет запускаться Ansible. Как только это сделано, головной сервер может соединяться с узлами по протоколу SSH и выполнять все необходимые задачи. Для работы с системами, не позволящими доступ с правами суперпользователя (root) по SSH, Ansible использует учетные данные, позволяющие выполнять действия от имени суперпользователя с помощью команды sudo.

Ansible может быть запущен из командной строки без использования конфигурационных файлов для простых задач, таких как проверка, что какой-нибудь сервис запущен, или для обновления триггеров и перезагрузки. Для более комплексных задач, конфигурационные файлы создаются с помощью YAML и называются «сценарии» (playbook). В них могут быть использованы шаблоны для расширения функциональности.

В Ansible есть набор модулей, которые могут использоваться для управления различными системами, равно как и «облачными» инфраструктурами, такими как Amazon EC2 и OpenStack. Дополнительные модули могут быть написаны на практически любом языке программирования, при условии, что вывод будет в формате JSON.

II. Принцип работы системы управления конфигурациями Ansible

Конфигурация Ansible имеет древовидную структуру, и при большом количестве хостов главный управляющий сервер может иметь в своем подчинении несколько дополнительных серверов управления благодаря чему можно увеличить скорость работы всей системы. Управляющий сервер должен иметь файлы host inventory, playbooks, ansible config, core modules и custom modules. Файл Host inventory содержит информацию об обслуживаемых узлах, где команды будут исполнены. Файл конфигурации (Ansible config) используется для указания настроек окружения (сервера). Наборы инструкций (playbooks) состоят из одной или более задач, которые описываются с помощью функциональность модуля ядра Ansible (core modules) или сторонних модулей (custom modules), которые могут потребоваться в специфических ситуациях. Сами по себе наборы инструкций — последовательные наборы команд, в которых могут быть проверки условий: если условие не выполняется, определенные команды могут пропускаться. Для создания тестового стенда сперва мы указали настройки окружающей среды в ansible config, указали информацию обсуживаемых узлов в host inventory, далее подключились к управляющему серверу по SSH и запустили набор последовательных инструкций (playbooks) на удаленные узлы (web-сервера). В качестве тестового ПО был установлен web-сервер Tomcat. Использование Ansible показало простоту конфигурирования и администрирования удаленных серверов.

III. Место в информационной безопасности

Благодаря использованию системы управления конфигурациями Ansible совместно с системами контроля версий, такими как Git, Subversion и др., появляется возможность отслеживать историю конфигурационных настроек, которая помогает мониторить поведение систем и оборудования при определенных конфигурациях, после провести анализ и определить слабые места в производительности и безопасности. Наборы инструкций (playbooks) позволяют полностью воспроизводить рабочую систему в тестовом окружение, где потом можно производить детальные тесты. Так же при выходе из строя серверного оборудования или инфраструктуры благодаря наборам инструкций можно в кратчайшие сроки восстановить работоспособность оборудования и их настройки.

1. Trevor A.Roberts, DevOps for VMware Administrators // Pearson. – 2015. – – P. 145–174.
2. Madhurranjan Mohaan, Learning Ansible / Packt Publishing Ltd. – , 2014. -P. 125 .
3. Daniel Hall, Ansible Configuration Management Second Edition // Packt Publishing Ltd. – , 2015. -P. 143 – 172 .