Комбинированная архитектура системы конфигурационного управления вычислительной инфраструктурой

Ю.А. Воронцов, Е.К. Михайлова

Аннотация **– Информационные системы и сервисы любой вычислительной инфраструктуры требуют конфигурации в соответствии с текущими задачами, которые на них возлагаются. В настоящее время для настройки узлов инфраструктуры применяются системы конфигурационного управления, дающие возможность автоматизировать данный процесс. Одним из основных принципов данных систем является наличие управляющего узла в инфраструктуре, с которого будет производится управление конфигурациями, на котором также будут располагаться сами файлы конфигурации. Основное различие заключается в принципе взаимодействия с управляемыми узлами. В одном случае за инициализацию конфигурации узла отвечает управляющий узел, отправляющий конфигурационные файлы на управляемые машины, которые затем будут настроены в соответствии с данной конфигурацией, данная модель управления называется push-моделью. В другом случае применяется pull-модель, в которой предусмотрено, что за инициализацию запроса отвечает управляемый узел. Он самостоятельно опрашивает управляющий узел на предмет изменения своей конфигурации. Обычно все узлы инфраструктуры обладают различными политиками сетевого доступа: одни доступны постоянно, и их состояние известно системному администратору, другие же находятся под полным управлением их пользователей и нельзя уверенно сказать, когда они появятся в сети и получат конфигурацию. В соответствии с этим предлагается архитектура системы, реализующая обе модели управления, поскольку в таком случае не будет необходимости использовать несколько систем конфигурационного управления для настройки всех узлов в инфраструктуре.**

Ключевые слова **– система конфигурационного управления, конфигурационное управление, вычислительная инфраструктура, архитектура системы, модели конфигурационного управления.**

I. ВВЕДЕНИЕ

Современная вычислительная инфраструктура включает в себя широкий набор взаимосвязанных информационных систем и сервисов, которые предназначены для обеспечения функционирования средств информационного взаимодействия организации [1].

Статья получена 05.08.2021 г.
Ю.А. Воронцов, МИРЭА (e-mail: voroncov_yu@mirea.ru),
Е.К. Михайлова, МИРЭА (e-mail: mihajlova_e@mirea.ru).

Управление современной вычислительной инфраструктурой включает в себя использование систем конфигурационного управления. Данные системы позволяют облегчить процесс настройки узлов инфраструктуры. Однако в состав вычислительной инфраструктуры входят узлы с различной политикой сетевого доступа, доступность одних узлов в сети может контролироваться администраторами вычислительной инфраструктуры, другие же находятся под управлением их пользователей и предсказать доступность таких узлов в сети является невозможным. Данная особенность не позволяет управлять всеми узлами вычислительной инфраструктуры при помощи одной системы конфигурационного управления.

Это приводит к необходимости использования в составе одной вычислительной инфраструктуры нескольких систем конфигурационного управления. Однако, такой подход повышает сложность задач конфигурационного управления, снижает устойчивость инфраструктуры, а также приводит к дополнительным эксплуатационным расходам. Предлагается реализовать архитектуру системы конфигурационного управления, позволяющую в рамках одной инфраструктуры управлять узлами с разной политикой сетевого доступа.

II. КОНФИГУРАЦИОННОЕ УПРАВЛЕНИЕ

Под конфигурационным управлением понимается процесс поддержания компонентов вычислительной инфраструктуры в требуемом состоянии. Такой метод позволяет отслеживать состояние инфраструктуры и её работу при внесении изменений в её конфигурацию.

Конфигурационное управление включает в себя определение требуемого состояния системы с последующим созданием и сопровождением созданной системы. Управление конфигурациями связано с понятиями анализа конфигурации и анализом отклонений, они используются для определения состояния систем, нуждающихся в обновлении, перенастройке или исправлении. Конфигурационное управление направлено на снижение сложности изменения настроек узлов вычислительной инфраструктуры и установленного на них программного обеспечения [2].

Системы конфигурационного управления позволяют автоматизировать процессы получения информации о конфигурации, изменения конфигурации, а также применения изменённой конфигурации.

В основе любой системы конфигурационного управления лежит внедрение в вычислительную инфраструктуру одного или нескольких управляющих узлов. С данных узлов будет производиться управление параметрами остальных узлов в вычислительной инфраструктуре. Управляющие узлы отвечают за хранение файлов конфигураций, также на данных узлах производятся все операции по управлению конфигурационными файлами. Различаются модели распространения конфигурации на управляемые узлы.

III. Модели конфигурационного управления

Всего существует 2 модели: push-модель и pull-модель.

Push-модель подразумевает полное управление конфигурацией с управляющего узла. Распространение конфигурации на управляемые узлы инициируется с управляющего узла администратором системы или при наступлении какого-либо события [3]. Данная модель не подразумевает обязательного наличия клиентской составляющей на управляемых узлах.

Одним из представителей систем, реализующих push-модель, является система Ansible. Данная система не использует в своей архитектуре клиентов. Для управления и распространения конфигураций используется только управляющий узел и конфигурационный репозиторий.

Репозиторий системы Ansible содержит в себе набор конфигурационных файлов, содержащих как настройки самой системы, так и конфигурации узлов инфраструктуры.

Одним из важных элементов репозитория является inventory-файлы с информацией о хостах, конфигурируемых при помощи Ansible. Данные файлы содержат в себе параметры подключения к управляемым узлам, к примеру, их IP-адреса, имена пользователей для подключения и т. д.

Репозиторий содержит в себе сами файлы конфигураций, написанные с использованием языка разметки YAML. Данный язык позволяет легко описывать требуемые конфигурации, а также является простым для человеческого восприятия.

По умолчанию конфигурационный репозиторий расположен на управляющей машине Ansible. Помимо централизованного возможно создание децентрализованного репозитория, используя распределенные системы контроля версий. Это, в свою очередь, даст преимущество при необходимости воссоздания конфигураций вычислительной инфраструктуры. Также применение данного инструмента позволит реализовать децентрализованное внесение изменений.

Управляющий узел Ansible отвечает за распространение конфигурации на управляемые машины, основываясь на заданном в репозитории inventory-файле и paybook'ах. Вся передача модулей на управляемые машины происходит посредством протокола SSH. Используя данный протокол, Ansible подключается к управляемым хостам, передаёт им необходимые модули, а затем получает обратно статистику выполнения модулей. При этом вся настройка на управляемых узлах происходит через интерпретатор Python [4,5].

Архитектура Ansible представлена на рисунке 1.

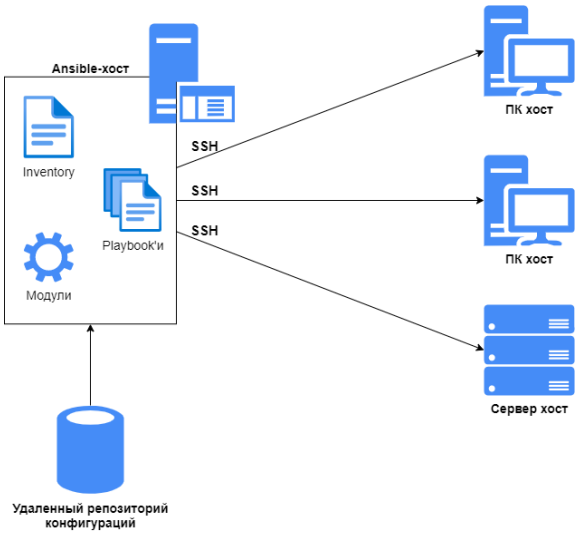

Рис. 1 – Архитектура системы Ansible

Pull-модель управления подразумевает инициацию изменений самими управляемыми узлами. Через определённый промежуток времени узлы запрашивают конфигурацию с управляющего сервера и выполняют необходимые изменения для приведения системы в необходимое состояние. В таком случае применение изменений производят именно управляемые узлы, а не управляющий [3].

Соответственно для реализации pull-модели на управляемом узле должно быть установлено дополнительное клиентское программное обеспечение, которое будет выполнять запросы конфигурационных файлов с управляющего узла и последующую настройку управляемого узла в соответствии с полученной конфигурацией.

Представителем pull-модели является система Puppet, базирующаяся на клиент-серверной архитектуре. В системе присутствует Puppet-сервер, Puppet-клиент, который в терминологии системы называется Puppet-агент, а также репозиторий конфигураций.

Репозиторий системы Puppet представляет из себя набор специальных директорий – окружений – независящих друг от друга наборов конфигураций. Каждое окружение в свою очередь содержит набор корневых директорий, в которых находятся Puppet-конфигурации для конкретного узла (node) вычислительной инфраструктуры. Конфигурация узлов описывается в специальных файлах-манифестах, написанных на языке Puppet DSL.

Репозиторий может быть расположен локально на Puppet-сервере в предназначенной для этого директории с конфигурационными файлами. Такой вариант размещения подразумевает централизованное размещение репозитория. Второй вариант размещения –

создание репозитория, используя систему контроля версий.

Сервер системы Puppet предназначен для предоставления конфигурационных файлов из репозитория конфигураций управляемым хостам (нодам). Сервер является основным хранилищем текущих конфигураций.

Взаимодействие с клиентами строится на базе протоколов HTTPS и TSL. Задача сертификации также ложится на Puppet-сервер.

Puppet-агент отвечает за запрос конфигурации с сервера и приводит управляемый им узел в состояние, описанное в конфигурационном файле. Помимо этого, агент занимается отправкой статистических данных на Puppet-сервер [6].

Архитектура системы Puppet представлена на рисунке 2.

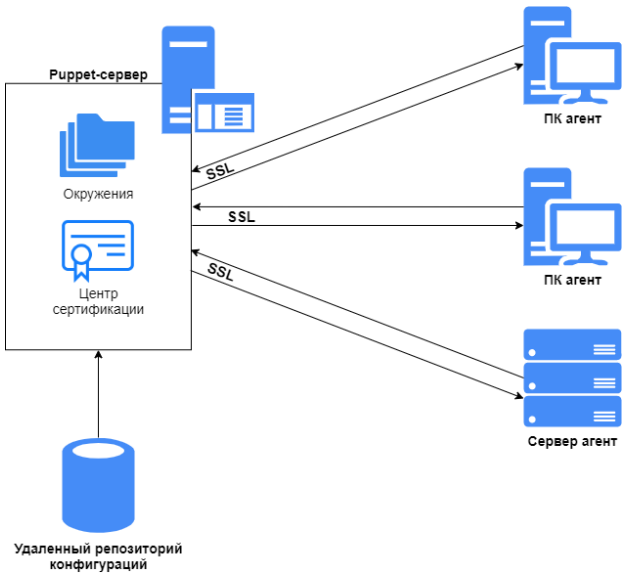

Рис. 2 – Архитектура системы Puppet

IV. Комбинированная архитектура

Рассмотренные примеры систем конфигурационного управления базируются только на одной модели, каждая из которых подходит для управления узлами с одной политикой сетевого доступа. Push-модель больше подходит для конфигурации узлов, состояние которых всегда известно и находится под управлением системного администратора. То есть push-модель больше подходит для управления серверами и сетевым оборудованием. Pull-модель, напротив, не преследует цель моментального применения конфигурации, а направлена на гарантированную доставку конфигурационных файлов до управляемых узлов, в каком бы состоянии на момент изменения этих самых файлов они бы не находились. Следовательно, системы с данной моделью больше подходят для настройки, к примеру, персональных машин пользователей.

В крупных вычислительных инфраструктурах присутствует большое количество узлов с обеими сетевыми моделями, соответственно необходимо использовать как push-модель, так и pull-модель для корректного управления и применения конфигураций. К примеру, при использовании системы с pull-моделью будет невозможно своевременно применить конфигурацию для серверов или сетевого оборудования, а при использовании системы только с push-моделью невозможно будет гарантировать доставку конфигурации к узлам с изменчивым состоянием, что не гарантирует обновление их конфигурации при отправке команды с управляющего узла.

На основании выявленных особенностей можно сделать вывод, что для управления сразу всей инфраструктурой через единую систему, данная система должна реализовывать обе модели конфигурационного управления. Предлагается комбинированная архитектура системы конфигурационного управления вычислительной инфраструктурой, которая позволит реализовать управление всеми возможными компонентами вычислительной инфраструктуры через единый интерфейс вне зависимости от состояния, в котором они находятся. Итоговая архитектура системы конфигурационного управления представлена на рисунке 3.

Таким образом, узлы с неопределённым состоянием, к примеру, персональные компьютеры пользователей, будут управляться при помощи pull-модели, а узлы с известным состоянием, а также требующие моментального применения конфигурационных изменений, такие как сетевые узлы, а также серверные машины, будут управляться при помощи push-модели.

Конфигурационный репозиторий, предназначенный для хранения файлов конфигурации, наиболее удобно будет реализовать по схеме построения, схожей с системой Puppet. Репозиторий будет содержать окружения для каждого из вариантов настройки инфраструктуры, к примеру, тестового, dev-окружения или production. Внутри каждого окружения будет присутствовать несколько директорий для описания конфигураций каждого узла в инфраструктуре или группы узлов, что позволит отделить конфигурации друг от друга и удобно ими управлять по-отдельности.

Производить описание непосредственно файлов конфигураций предлагается при помощи языка разметки данных YAML, позволяющего реализовывать конфигурации в декларативном стиле. YAML обладает удобной и понятно структурой, нацеленной именно на данные, а не на формат документов, что и определяет сферу его использования, а именно формирование конфигурационных файлов различного назначения. Предлагается использовать схожую с Ansible схему описания конфигураций, в которой команды и параметры к ним явно отражают необходимые команды и параметры, реализуемые на управляемом узле.

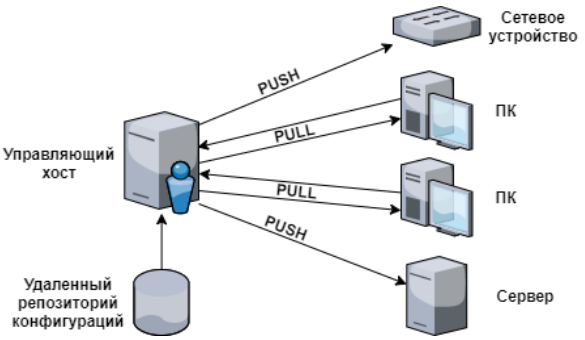

Рис. 3 – Комбинированная архитектура системы конфигурационного управления

Передавать данные между управляемым и управляющими узлами предлагается при помощи нескольких протоколов. Для узлов с определённым состоянием можно использовать протокол SSH. Обусловлено это его высокой распространенностью и широкой поддержкой со стороны различных операционных систем, таких как операционная система сервера и операционная система сетевого оборудования. Вне зависимости от устройства протокол SSH позволит установить подключение к нему и произвести настройку. Для управления узлами с изменяющимся состоянием предлагается использовать протокол HTTPS в связке с TSL, позволяющий устанавливать защищённые соединения, гарантирующие безопасную передачу данных от управляемого узла до управляющего. Как и в случае с Puppet сессия такого протокола будет инициироваться клиентским узлом, когда он будет находиться в состоянии, которое будет позволять изменять конфигурацию с целью её приведения к необходимому состоянию. Соответственно, подразумевается использование системы как с установленным клиентом на управляемых узлах, в случае с применением pull-модели для управления конфигурацией, так и без клиентской части, в случае управления узлами с известным состоянием при помощи push-модели. Однако, клиент может быть использован и для push-модели, в таком случае он будет получать всю необходимую конфигурацию, в соответствии с которой необходимо будет настроить управляемый узел, а на управляющий узел будет отдаваться только статистика об успешном или неуспешном применении конфигурации.

Заключение

Предлагаемая комбинированная архитектура системы позволит более эффективно управлять конфигурацией вычислительной инфраструктуры, поскольку не будет необходимости разворачивать несколько систем конфигурационного управления для узлов с различной сетевой политикой. Всеми узлами можно будет управлять при помощи единого интерфейса и единого набора команд, что повысит надёжность конфигурационного управления.

БИБЛИОГРАФИЯ

[1] Старцев М. В. ИТ-инфраструктура: категориальный анализ. // Информационные технологии в экономике, бизнесе и управлении. 2016. Материалы III международной научно-практической конференции. С. 122-128.

[2] Red Hat, What is configuration management? [Электронный ресурс] // Блог компании Red Hat. URL: https://www.redhat.com/en/topics/automation/what-is-configuration-management (Дата обращения: 24.07.2021)

[3] Gayatri S Ajith, Beginner Fundamentals: Push & Pull Configuration Management Tools [Электронный ресурс] // Medium — платформа для социальной журналистики. 3 сентября 2019. URL: https://medium.com/@gayatrisajith/beginner-fundamentals-push-pull-configuration-management-tools-85eff1b41447 (Дата обращения: 24.07.2021)

[4] Документация на систему Ansible [Электронный ресурс] – URL: https://docs.ansible.com/ansible/latest/index.html (Дата обращения: 25.07.2021).

[5] Архитектура системы Ansible [Электронный ресурс] – URL: https://docs.ansible.com/ansible/latest/dev_guide/overview_architecture.html (Дата обращения: 25.07.2021).

[6] Документация на систему Puppet [Электронный ресурс] – URL: https://puppet.com/docs/puppet/7/puppet_index.html (Дата обращения: 26.07.2021).

[7] Serrano, João & Pereira, Rúben. (2020). Improvement of IT Infrastructure Management by Using Configuration Management and Maturity Models: A Systematic Literature Review and a Critical Analysis. Organizacija. 53. 3-19. 10.2478/orga-2020-0001.

Combined architecture of the configuration management system for computing infrastructure

Y.A. Vorontsov, E.K. Mikhailova

***Abstract* - Information systems and services of any computing infrastructure require configuration in accordance with the current tasks assigned to them. Currently, configuration management systems are used to configure infrastructure nodes, which make it possible to automate this process. One of the basic principles of these systems is the presence of a control node in the infrastructure, from which configuration management will be performed, on which the configuration files themselves will also be located. The main difference lies in the principle of interaction with managed nodes. In one case, the management node is responsible for initializing the node configuration, sending configuration files to managed machines, which will then be configured according to this configuration, this management model is called the push model. In another case, the pull model is used, which provides that the managed node is responsible for initializing the request. It independently polls the control node for changes in its configuration. Usually, all infrastructure nodes have different network access policies: some are constantly available and their status is known to the system administrator, while others are under the full control of their users and it is impossible to say with certainty when they will appear on the network and receive the configuration. In accordance with this, a system architecture is proposed that implements both management models, since in this case it will not be necessary to use several configuration management systems to configure all nodes in the infrastructure.**

***Keywords* - configuration management system, configuration management, computing infrastructure, system architecture, configuration management models.**

REFERENCE

[1] Starcev M. V. IT-infrastruktura: kategorial'nyj analiz. // Informacionnye tehnologii v jekonomike, biznese i upravlenii. 2016. Materialy III mezhdunarodnoj nauchno-prakticheskoj konferencii. C. 122-128.

[2] Red Hat, What is configuration management? [Jelektronnyj resurs] // Blog kompanii Red Hat. URL: https://www.redhat.com/en/topics/automation/what-is-configuration-management (Data obrashhenija: 24.07.2021)

[3] Gayatri S Ajith, Beginner Fundamentals: Push & Pull Configuration Management Tools [Jelektronnyj resurs] // Medium — platforma dlja social'noj zhurnalistiki. 3 sentjabrja 2019. URL: https://medium.com/@gayatrisajith/beginner-fundamentals-push-pull-configuration-management-tools-85eff1b41447 (Data obrashhenija: 24.07.2021)

[4] Dokumentacija na sistemu Ansible [Jelektronnyj resurs] – URL: https://docs.ansible.com/ansible/latest/index.html (Data obrashhenija: 25.07.2021).

[5] Arhitektura sistemy Ansible [Jelektronnyj resurc] – URL: https://docs.ansible.com/ansible/latest/dev_guide/overview_architecture.html (Data obrashhenija: 25.07.2021).

[6] Dokumentacija na sistemu Puppet [Jelektronnyj resurs] – URL: https://puppet.com/docs/puppet/7/puppet_index.html (Data obrashhenija: 26.07.2021).

[7] Serrano, João & Pereira, Rúben. (2020). Improvement of IT Infrastructure Management by Using Configuration Management and Maturity Models: A Systematic Literature Review and a Critical Analysis. Organizacija. 53. 3-19. 10.2478/orga-2020-0001.