DOI 10.37882/2223-2966.2023.6-2.29

ВОЗМОЖНОСТЬ ИСПОЛЬЗОВАНИЯ СИСТЕМЫ УПРАВЛЕНИЯ КОНФИГУРАЦИЯМИ ANSIBLE КАК АНАЛОГ ГРУППОВЫХ ПОЛИТИК WINDOWS В СРЕДАХ НА ОСНОВЕ ЯДРА LINUX

THE ABILITY TO USE THE ANSIBLE CONFIGURATION MANAGEMENT SYSTEM AS AN ANALOGUE OF WINDOWS GROUP POLICIES IN LINUX KERNEL-BASED ENVIRONMENTS

M. Runkov

Summary: The modern enterprise network consists of hundreds or even several thousand personal computers. Roles in such a network are usually divided into client and server. The client can be a host, as is often the case with the Windows operating system installed or Linux-based systems such as Uubuntu. As a rule, the advantage is given to the Microsoft operating system with the ability to manage group policy systems. If we take into account modern realities, the majority of Russian government agencies have switched to the development of the company Basalt SPO system with the name alt, which already has a gpupdate tool out of the box and is an analogue of GPO Windows. Our world is not perfect and often there are questions of using and controlling systems on the Linux kernel in a mixed network where operating systems of both kinds are present. This article demonstrates the capabilities of the Ansible configuration management system for managing.

Keywords: control system, ansible, group policies, host, network, ssh, network, operating systems, software, modules, playbook, inventory file, configuration, configuration file, console, terminal, role, server, yaml, linux, active directory, microsoft, ubuntu.

Рунков Михаил Вячеславович
Национальный исследовательский Мордовский государственный университет им. Н.П. Огарёва, г. Саранск
runkov96@inbox.ru

Аннотация. Современная сеть предприятия состоит из сотен, а то и нескольких тысяч персональных компьютеров. Роли в такой сети как правило подразделяются на клиент и сервер. Клиентом может выступать хост как часто это бывает с установленной операционной системой Windows или базирующейся на ядре Linux такие системы как Uubuntu. Как правило преимущество отдается операционной систему компании Майкрософт с возможностью управления системами групповыми политиками. Если учесть современные реалии, то большинство государственных структур России перешли на разработку компании Базальт СПО систему с названием альт в которой уже сейчас из коробки имеется инструмент gpupdate и является аналогом GPO Windows. Наш мир не идеален и за частую встает вопросы использования и контроля систем на ядре Linux в смешанной сети, где присутствую операционные системы обоих родов. Данная статья демонстрирует возможности системы управления конфигурациями Ansible для управления такими системами как Ubuntu в локальной сети предприятия. Воспользуемся инструментарием данной программы чтобы прийти к выводу о возможности использования ее как некую единую систему контроля групповых политик в Linux. Для демонстрации построен стенд из четырех машин с различными популярными, на сегодняшний день операционным системами с целью тестирования на определенном спектре конфигураций. Статья не затрагивает моменты связанные с установкой и настройкой Ansible и не затрагивает механизм шаблонов Jinja2. Актуальность темы заключается в том, что на сегодняшний момент практически отсутствуют статьи по данной тематике как в английском, так и русского сегменте сети, а использование системы, контролирующей и обеспечивающей единообразие окружение пользователя, например в системе Ubuntu возникает не так редко.

Ключевые слова: система контроля, ansible, групповые политики, хост, сеть, ssh, сеть, операционные системы, программное обеспечение, модули, плейбук, файл инвентаризации, конфигурация, конфигурационный файл, консоль, терминал, роль, сервер, yaml, linux, active directory, microsoft, ubuntu.

Введение

Компания Майкрософт предоставляет широкий спектр программного обеспечения на базе своей операционной системы Windows. На данный момент большинство коммерческих организаций используют системы Windows и Windows Server как основную систему для пользователей и серверов локальной сети предприятия. С точки зрения администрирования локальной сети основанной на Windows подобных системах не возникает проблем т.к. администратору предоставлен определенный инструментарий управления хостами. В распоряжении администратора имеется оснастка Active Directory, DNS, DHCP и GPO(group policy project). Средство управления групповыми политиками. GPO в доменной среде позволяет администратору тонко настраивать локальные машины, организовывать права доступа, подключать сетевые ресурсы на основе прав доступа такие как принтеры, сетевые диски и т.д. Но в разномастной среде средство управления GPO не панацея и присутствие операционной системы на базе ядра Linux вносит неуправляемую переменную в среду однотипных операционных систем. Если возникает такая проблема, есть определенный выход из данной ситуации, использовать Ansible как средство управления групповыми политиками для взаимодей-

ствия с ОС на базе ядра Linux (Ubuntu, Mint, Debian, CentOS, Fedora и т.д).

Историческая справка

Автором платформы является Михаэль Дехан а развитием и коммерциализацией платформы занималась компания Ansible, в последствии поглощенная не мало известной компанией Red Hat. Ansible используется по большей части в контексте подхода CI/CD (Continuous Integration, CD — Continuous Delivery).

Система Ansible разрабатывается с февраля 2012-го и началась с создания простого побочного проекта. Стремительное развитие системы оказалось сюрпризом для создателей. На данный момент над проектом работают тысячи человек.

В полной мере Ansible не сможет удовлетворить все потребности в плане управления в той мере, в которой это могут групповые политики в Windows, но сможет обеспечить администратора неким рычагом воздействия на такие системы как ubuntu и обеспечить однотипность конфигураций и единой точкой управления. Ansible использует только ssh ключи для установления подключения типа сервер, управляемая машина и файл инвентарицзации для указания списка хостов на которых требуется выполнить ту или иную задачу. Пример файла инвентаризации ниже.

```
[Debian]
Ubuntu1 ansible_ssh_host =192.168.4.76
ansible_python_interpreter=/usr/bin/python3

[CentOS]
CentOS1 ansible_ssh_host =192.168.4.75
ansible_python_interpreter=/usr/bin/python3
```

Нельзя не упомянуть что в Ansible есть возможность использования ad hoc команд т.е. это команды которые выполняются непосредственно из терминала на которую установлена система управления конфигурациями. Все команды могут быть описаны в файле playbook что обеспечит воспроизводимость и предсказуемость. Такой подход описания файла конфигурации называется декларативным. Декларативный подход значительно упрощает понимание настроек, а также упрощает внесение изменения. Также стоит упомянуть что в установке клиентской части на удаленные машины нет необходимости т.к. Ansible не использует схему с удаленными агентами.

В настоящий момент сложно встретить человека из сферы информационных технологий который не слышал о системе Ansible. Данная система не несет в себе новые концепции, но использует старые хорошо апробированные решения, разработанные другими специалистами. Используя OpenSSH и устранив демонов управления, операционная система может мгновенно работать без установки на удаленные машины. Ansible как и другие средства управления, предоставляет язык DSL (Domain Specific Language) использующийся для описания состояний серверов. Единая система управления конфигурациями и развертыванием существенно облегчает работу системным администраторам. Файл сценарий описывает на каких удаленных серверах будет выполняться упорядоченный список задач и имеет расширение yml. Основными файлами являются ansible.cfg конфигурация системы, файл инвентаризации hosts в котором перечислены управляемы хосты и файл playbook. yml который представляет из себя список задач, разделенных на этапы. После запуска файла сценария задача выполняется одновременно на всех хостах, указанных в файле инвентаризации. Но надо отметить, что задачи одновременно выполняются только на пяти первых узлах, но данное ограничение можно обойти указав в файле ansible.cfg параметр forks с необходимым значением. Ansible последовательно выполняет каждый этап и ожидает пока команда не будет выполнена на всех хостах. Помимо Ansible существуют такие системы управления конфигурациями как Puppet, Chef, но они используют несколько иной подход и архитектуру.

Puppet клиент-серверное и кроссплатформенное приложение также как и Chef написанное на языке Ruby. В Chef используется понятие «Рецептов» аналог плейбуков Ansible. Рецепты используют описание состояние ресурсов системы, в котором ей следует находится в настоящий момент, наличие или отсутствие файлов, установленные пакеты и запущенные службы. Вышеперечисленные системы отличаются в подходе к управлению системами.

Различаю два подхода в конфигурации системы с использованием системы управления конфигурациями принудительная и добровольная. Такие системы как Chef и Puppet используют подход добровольной настройки. Установленный агенты периодически связываются с управляющей машиной и читают информацию о конфигурации. При запуске файл сценария Ansible принудительно подключится к машине и выполнит задачи. У принудительной настройки есть важное преимущество контролируемое время обновления серверов. Некоторые сторонники добровольной настройки заявляют, что данный подход имеет преимущество в масштабировании на большое число серверов, когда серверы могут появиться в любой момент. Отметим, что Ansible использовался и используется для управления тысячами узлов и результаты с добавлением и удалением узлов были отличные. Еще одной мощной стороной данной системы является набор встроенных модулей. Модули решают такие задачи, как установка приложений, пере-

запуск службы, копирование файлов и предоставление прав использования. Модули являются идемпотентными. Иначе говоря, если в файле сценарии описано создание несуществующего пользователя, то Ansible создаст его, а если он уже присутствует на машине, то просто пропустит данный этап.

Большим преимуществом является использование синтаксиса сценариев основанных на YAML, языке описания данных, который создавался для легкого восприятия человеком. Все что нужно для управления удаленной машиной это SSH и Python версии 2.6 и выше.

Анализ существующих решений

На тот момент времени, когда писалась данная статья по запросу, поисковые системы выдавали результаты запросов связанные с samba или active directory, но по системам управления групповыми политиками для linux систем практически ничего не было. Как правило в поисковой выдачи встречались, групповые политики в дистрибутивах ALT. До недавнего времени мне не приходилось сталкиваться с ALT как в работе, так и в учебе, но почитав документацию, был приятно удивлен широким спектром предлагаемых возможностей. ALT уже сейчас предлагает список объектов которыми администраторы могут управлять по средствам инструмента gpupdate и машин введенных в домен Samba, но многие функции являются еще экспериментальными. Действительно, в IT среде привыкли слышать, как правило про групповые политики в контексте систем, созданных компанией Microsoft. Из всего выше сказанного более или менее рабочий аналог групповых политик из систем Windows, имеет ALT. Установка инструмента групповых политик на систему ALT производится командой apt-get gpupdate и произведя небольшие манипуляции, получаем рабочую схему. Но что, если организация имеет несколько машин на Ubuntu и десятки машин под управлением Windows системы, а желание переходить на ALT отсутствует. С машинами под управлением Windows недолжно возникнуть вопросов, так как реализуется уже давно проверенная схема. Но такая возможность есть и для Ubuntu, под названием Ansible. Описав в файле инвентаризации и добавив ключ на удаленную машину, администратор способен управлять параметрами запуская команды из командной строки на все хосты разом или на отдельно стоящий хост, но и описав в файле playbook.yml способен автоматизировать некоторые рутинные процессы (обновлением системы). Ко всему прочему в Ansible есть возможность удаленного управления Windows системами через WinRM. WinRM является реализацией протокола WS-Management компании Microsoft и представляет из себя протокол простого доступа к объектам. В возможности данной системы также входит управление коммутаторами от компании cisco и mikrotik.

Многие системы управления конфигурацией предоставляют уровень абстракции настолько мощный, что позволяют использовать один и тот же сценарий для работы серверов с различными операционными системами. Кроме того, вместо конкретных диспетчеров пакетов, таких как yum или apt можно использовать абстракцию «пакет», поддерживаемую системой управления конфигурациями. Ansible работает иначе — для установки пакетов в системы, основанные на диспетчере Apt, вы должны использовать диспетчера Apt и um. В практической плоскости это упрощает использование Ansible, хотя на первый взгляд может показаться недостатком.

Модуль является основной единицей, которая используется для повторного использования в Ansible. Однако область использования модуля ограничена операционной системой, это позволяет писать надежные и качественные модули управления. Ansible как открытый проект готов принимать новые модули от сообщества разработчиков

Но Ansible сможет управлять лишь теми серверами, которые известны ему. При помощи файла, который находится в реестре Ansible можно передать информацию о серверах. Для каждого сервера необходимо иметь имя, которое будет идентифицироваться его в Ansible. Для этого можно использовать имя хоста или выбрать иной псевдоним. Также необходимо указать дополнительные параметры, которые будут определять подключение.

Еще одной особенностью вызывающие восхищение, Ansible является масштабирование — вверх и вниз или вертикальное масштабирование. Имеется в виду сложность автоматизируемых задач, а не количество хостов. Простотой разработки отдельных задач характеризуется масштабирование вниз. Масштабирование вверх упрощается благодаря механизмам деления сложных задач на небольшие части. Основным механизмом деления сценария на отдельные файлы в Ansible являются роли. Роли упрощают редактирование сценариев и их переиспользование. В таком контексте каждая роль представляет из себя отдельный хост. В случае такого распределения веб серверу можно присвоить роль webserver или database для баз данных.

Система Ansble поддерживает так называемые плагины обратного вызова, которые могут выполнять некоторые действия в ответ на такие события, как запуск операции или завершение задачи на хаете. Плагины обратного вызова можно использовать, например, для отправки сообщений Slack или для вывода записей в удаленный журнал. Даже информация, которую вы видите в окне терминала во время выполнения сценария Ansible, фактически выводится плагином обратного вызова.

Практическая часть

В данном блоке будут рассмотрены и приведены примеры использования Ansible как средство управления машинами на основе Linux ядра. Демонстрация работы будет производится на виртуальных машинах с применением VirtualBox, а операционные системы не имеют никаких дополнительных установленных пакетов т.е. имеют конфигурацию что называется из коробки. В результате всего проделанного придем к выводу о возможности или не возможности такого подхода в организации управления узлами в локальной сети.

Тестовая схема состоит из четырех виртуальных хостов, один из которых будет выступать в роли основного сервера, остальные в роли ведомых машин соответственно master и slave. Выполнятся команды будут в среде под управлением таких операционных систем как Ubuntu версии 22.04, CentOS 7, Mint 21.1. Выбор таких операционных систем обусловлен их распространенностью как в корпоративной, так и пользовательской среде. Ubuntu является одной из самых популярных операционных систем и имеет многочисленные ответвления такие как Kubuntu. Другие системы в нашей сборке также достаточно часто используются в различных средах и предлагают схожий функционал и отличаются пакетными менеджерами.

Используя модули Ansible (которых более 200) такие как apt, shell имеется возможность управления правами доступа пользователей, добавить идентичную учетную запись администратора на все машины и обеспечить единообразие пользовательского окружения. Под единообразием пользовательского окружения понимается пакеты необходимых программ пользователя, использующиеся на конкретном предприятии. Необходимо добавить, что для различных подразделений пользовательское окружение будет меняться в зависимости от потребностей. В данном случае помощь приходит разбиение хостов на группы, а также возможность использования ролей и различных файлов конфигурации. Разграничив удаленный хосты на группы и подгруппы, мы сможем обеспечить парк конфигураций в зависимости от требований пользователей. В случае наличия большого парка машин такое распределение позволит не запутаться в огромном количестве конфигурационных файлов. Большим плюсом является и то, что используется язык yaml т.к. он легко читается.

Ниже приведен файл playbook.yml для демонстрации этапов, выполняемых на управляемых узлах. Изначально в файле инвентаризации hosts необходимо определить на какие группы будут подразделяться наши узлы. Исходя из всего выше сказанного были определены две группы хостов с названием Debian и CentOS что и отражено в файле конфигурации.

```
— hosts: Debian
 gather_facts: false
 tasks:
 — name: Add user
 user:
 name: Admin
 password:$6$58UJuN45gHE9ODEy$OkxZZjTxxXJkPI6MWtsww.Wib706wbRLDmPh064G0G5wVii83jjJhZTV8uUtxIH4jSXKChYgQVVUKlcWFOw7t0
 shell: /bin/bash
 groups: adm,sudo
 append: yes

 — name: Update and Upgrade
 shell: |
 apt-get update && apt-get upgrade -y

 — name: Install Apps
 shell: |
 wget https://dl.google.com/linux/direct/google-chrome-stable_current_amd64.deb
 dpkg -I –force-depends google-chrome-stable_current_amd64.deb
```

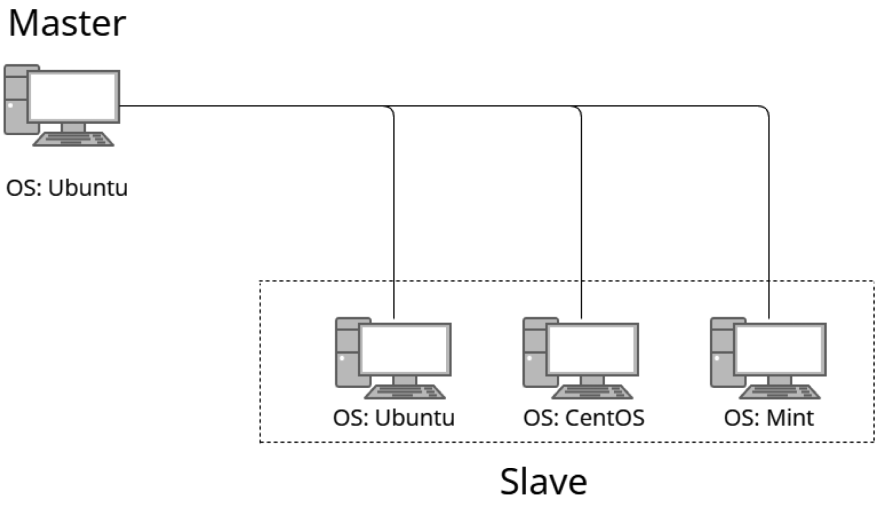

Рис. 1. Схема тестового стенда

```
— hosts: CentOS
gather_facts: false
tasks:
— name: Add user
user:
name: Admin
password:  $6$58UjuN45gHE9ODEy$OkxZZjTxxXJkPI6MWtsww.Wib706wbRLDmPh064G0G5wVii83jjJhZTV8uUtxlH4jSXKChYgQVVUKlcWFOw7t0
shell: /bin/bash
groups: adm,root
append: yes
— name: Update and Upgrade
shell: |
yum update && yum upgrade -y

— name: Install Apps
shell: |
wget https://dl.google.com/linux/direct/google-chrome-stable_current_x86_64.rpm -y
yum localinstall google-chrome-stable_current_x86_64.rpm -y
```

В группе Debian состоят две машины под управлением ОС Mint, Ubuntu и CentOS в аналогичной названию группе. При помощи данного скрипта на хосты, состоящие в выше указанных группах, был добавлен пользователь, произведено обновление системы в соответствии с их менеджерами пакетов и установлен браузера Google Chrome. Это не большой список задач, которые способен выполнить Ansible. Таким образом возможно добавлять правила в файрволл, но перед этим необходимо добавить правила на разрешение подключения по протоколу ssh и активировать файрволл. Иначе при активации файрволл (в случае Ubnutu это ufw) Ansible не сможет подключиться к системе из-за отсутствия разрешающего правила. Для CentOS выполняются аналогичные действия что и для систем на основе Debian, но заменяются менеджер пакетов т.е. yum и сами пакеты, вместо deb используются rpm пакеты. Таким образом можно добавить какое угодно количество пользователей в систему, задать права доступа и удаленно установить необходимые приложения для работы. Проделанные действия в значительной степени сокращают время работы администратора и сводят их к написанию скриптов под конкретные нужды.

На изображении выше приведен результат выполнения написанного плэйбука. Все задачи отработали штатно и в двух последних этапах мы видим, что Ansible внес изменения. Но вместо трех этапов что были описаны мы видим четыре. Gathering facts это скрытая задача запускаемая модулем setup. Эта задача собирает сведения об удаленном узле и пишет их в переменную ansible_facts. Но если вы не используете эти сведения в своем плейбуке то можно отключить ее для ускорения работы указав в playbook.yml gather_facts:False.

Следующим этапом будет добавление сетевого диска. Для этого был заранее подготовлен диск на машине под управлением операционной системой Windows. Ниже приведен этап монтирования диска при помощи /etc/fstab файла. Данные для подключения добавляются на данном этапе через создание файла /root/.smbclient на управляемой машине. После выполняется команда mount -a сокращение от all для монтирования диска по указанному пути.

```
root@ubuntu-server:/etc/ansible# ansible-playbook playbook.yml

PLAY [Debian] ***********************************************************************************************************

TASK [Add user] *********************************************************************************************************
changed: [Ubuntu1]

TASK [Update and Upgrade] ***********************************************************************************************
changed: [Ubuntu1]

TASK [Install Apps] *****************************************************************************************************
changed: [Ubuntu1]

PLAY [CentOS] ***********************************************************************************************************

TASK [Add user] *********************************************************************************************************
ok: [CentOS1]

TASK [Update and Upgrade] ***********************************************************************************************
changed: [CentOS1]

TASK [Install Apps] *****************************************************************************************************
changed: [CentOS1]

PLAY RECAP **************************************************************************************************************
CentOS1                    : ok=3    changed=2    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
Ubuntu1                    : ok=3    changed=3    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
```

Рис. 2. Выполнение файла playbook.yml

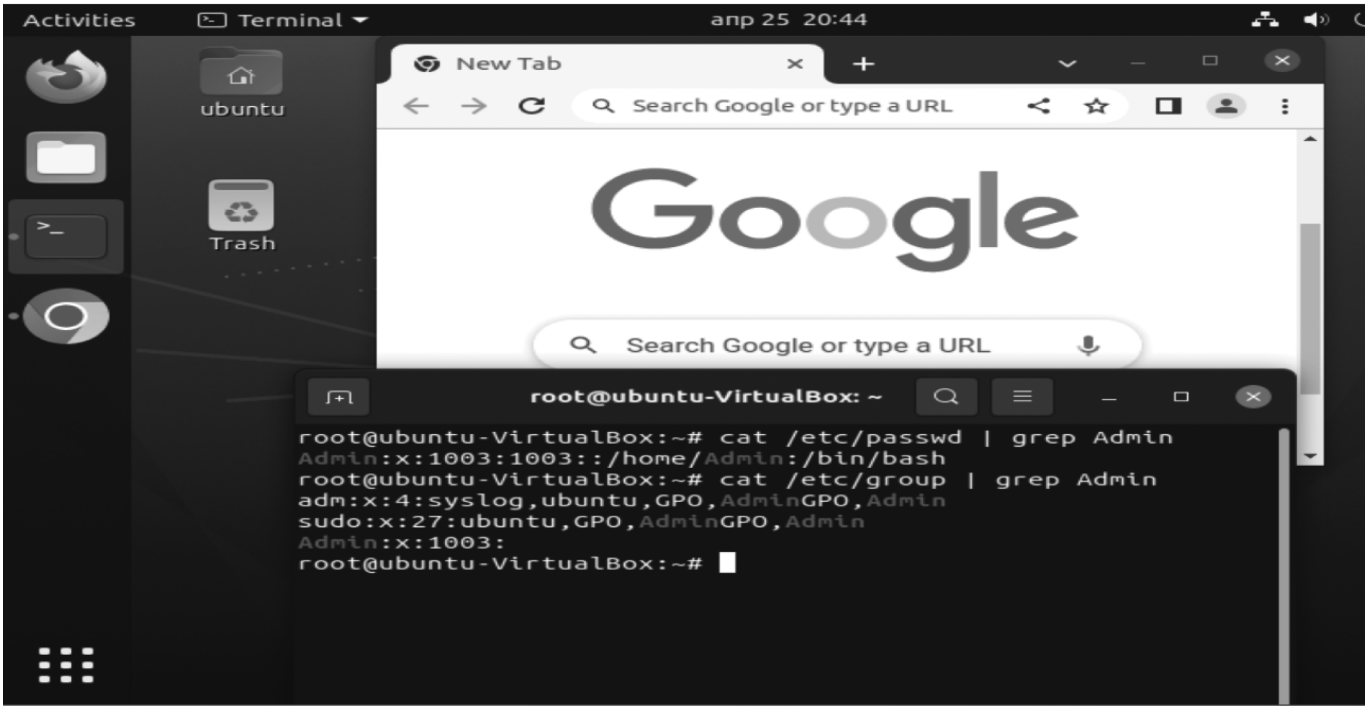

Рис. 3. Результат выполнения файла с задачами

— name: Auto mount disk
shell: |
echo "username=your_username" > /root/.smbclient && echo "password= your_password " >> /root/.smbclient
echo "//your_ip_address/f /mnt/f cifs user,rw,auto,credentials=/root/.smbclient 0 0" >> /etc/fstab
mount -a

Фото с результатом выполнения плейбука выше.

Заключение

Вопросы, связанные с удаленным управлением хостов, остаются актуальными и по сей день, а в особенности управление сетью в которой находятся хосты с различными операционными системами. Данное программное обеспечение поможет избежать некоторого количества рутины в плане администрирования, а в частности настройки и подготовки рабочих станций. Мной были приведены доказательства и примеры использования системы контроля конфигурации Ansible как аналог групповых политик Windows. Описанный исполняемый файл в полной мере продемонстрировал возможности использования программы в таком не тривиальном деле. Использую данную систему на производстве системный администратор упрощает работу, связанную с перенастройкой парка машин которую порой необходимо выполнить в сжатые сроки и позволяет обеспечить различным машинам в локальной сети единое рабочее окружение.

ЛИТЕРАТУРА

1. Мозер Рене., Хоштейн Лорин Запускаем Ansible ДМК Пресс; 2018. 382 с.
2. James Freeman., Jesse Keating Mastering Ansible — Third Edition Packt Publishing; 2019. 412 с.
3. Колесников Денис Николаевич LINUX на примерах. практика, практика и только практика Наука и техника; 2022. 320 с.
4. Уорд Брайан Внутреннее устройство Linux. 3-е изд Питер; 2022. 480 с.
5. Daniel Hall Ansible Configuration Management Packt Publishing; 2013. 92 с.
6. Шоттс Уильям Командная строка Linux. Полное руководство. 2-е межд. изд. Питер; 2022. 544 с.
7. Негус Кристофер Библия Linux. 10-е издание Питер; 2022. 928 с.
8. Шрёдер Карла Linux. Книга рецептов. 2-е изд. Питер; 2022. 592 с.
9. Эви Немет., Гарт Снайдер., Трент Хейн., Бэн Уэйли., Дэн Макин Unix и Linux:руководство системного администратора. 5-е изд. Диалектика-Вильямс; 2020. 1168 с.
10. Барретт Д Linux. Командная строка. Лучшие практики Питер; 2022. 256 с.

Журнал «Современная наука: актуальные проблемы теории и практики»