# Лабораторная работа №15. Работа с Ansible

Цель работы: формирование практических навыков решения задач с применением Ansible

**Выполнение учебных задач на узлах**

## **1.1. Отключение SELinux и перезагрузка узлов.**

Когда вы устанавливаете **RHEL-системы**, функция **SELinux** включена по умолчанию, из-за этого некоторые приложения в вашей системе могут фактически не поддерживать этот механизм безопасности. Чтобы такие приложения функционировали нормально, вам необходимо отключить **SELinux**. Рамках учебного стенда она нам не нужна.

Посмотрим и убедимся, что на всех узлах, стоит **SELinux**:

# ansible -m shell -a 'rpm -qa | grep selinux' test\_servers -i /root/ansible-inventory/my-test-servers.inventory

В случае, если **SELinux** присутствует в установленном состоянии на всех узлах. Выполним дезактивацию **SELinux** с помощью **Ansible**, без удаления **SELinux** с узлов.

Создадим плейбук для дезактивации **SELinux** на узлах с помощью **Ansible Server**:

# mcedit /root/ansible-playbooks/01-pb-selinux-disable.yml

Записываем в него содержимое:

---

- name: Операции с SELinux

hosts: test\_servers

become\_method: sudo

become\_user: root

tasks:

- name: Отключаем SELinux

selinux:

state: disabled

Проверим синтаксис плейбука 01-pb-selinux-disable.yml специальной утилитой --syntax-check от **Ansible**:

# ansible-playbook /root/ansible-playbooks/01-pb-selinux-disable.yml --syntax-check

Для дополнительной проверки можно воспользоваться командой:

# sestatus

Применим плейбук на узлы и посмотрим что будет с **SELinux**:

# ansible-playbook /root/ansible-playbooks/01-pb-selinux-disable.yml -i /root/ansible-inventory/my-test-servers.inventory

Чтобы завершить дезактивацию **SELinux**, требуется перезагрузить узлы. Перезагрузка узла с помощью **Ansible** не сложная задача.

Предполагаются следующие шаги:

* **SSH** подключение на узел.
* Перезагрузите узел.
* Отключитесь, прежде чем узел закроет ваше **ssh-соединение**.
* Подождите некоторое количество секунд, чтобы убедиться, что узел действительно выключен.
* Попробуйте подключиться по **ssh** к узлу и выполнить команду.
* Повторите попытку **ssh**, пока она не сработает.

В **Ansible** версии **2.7** или выше. **2.7** ввели модуль перезагрузки – reboot.

Теперь все, что вам нужно сделать, это добавить это в свой плейбук:

---

- name: Перезагрузка серверов

hosts: test\_servers

become\_method: sudo

become\_user: root

tasks:

- name: Перезагрузка хостов и ожидание их перезагрузки

reboot:

msg: "Reboot initiated by Ansible"

connect\_timeout: 5

reboot\_timeout: 600

pre\_reboot\_delay: 0

post\_reboot\_delay: 30

test\_command: whoami

Этот плейбук выполнит следующее:

* Перезагрузит узел.
* Подождет 30 секунд.
* Попытается подключиться через **SSH** и запустить whoami.
* Отключится через 5 секунд, если не сработает **ssh**.
* Будет продолжать пытаться подключиться в течение 10 минут (600 секунд).

Применим плейбук:

# ansible-playbook /root/ansible-playbooks/02-pb-reboot.yml -i /root/ansible-inventory/my-test-servers.inventory

**Примечание!** После перезагрузки узлов стоит проверить состояние **SELinux** на узлах:

# sestatus

Как видно **SELinux** отключен на всех узлах дистанционно с помощью **Ansible**.

**Примечание!** В любой команде **Ansible**, которая требует перезагрузки после изменения узел будет перезагружен, когда **playbook** завершится, а затем **Ansible** будет ждать, пока узел не станет доступным, и **ssh** будет работать, прежде чем перейти к следующему плейбуку.

Добавьте директиву:

notify: Reboot host and wait for it to restart

**Примечание!** Если вам нужно перезагрузиться на полпути через **playbook** вы можете заставить все задачи выполниться с помощью команды:

- name: Reboot if necessary

meta: flush\_handlers

Иногда это можно сделать, чтобы что-то изменить, заставить сервер перезагрузиться, а затем проверить, что изменения вступили в силу и все в том же **playbook**.

## **1.2. Обновление операционных систем на узлах.**

Перед выполнением других учебных задач, обновим операционные системы на узловых серверах с помощью **Ansible**.

Создаем файл плейбука:

# mcedit /root/ansible-playbooks/03-pb-upgrade-all-packages.yml

Записываем в него содержимое:

---

- name: Обновление всех пакетов на узлах

hosts: test\_servers

become\_method: sudo

become\_user: root

tasks:

- name: Обновить все пакеты

yum:

name: '\*'

state: latest

Проверим синтаксис плейбука 03-pb-upgrade-all-packages.yml специальной утилитой --syntax-check от **Ansible**:

# ansible-playbook /root/ansible-playbooks/03-pb-upgrade-all-packages.yml --syntax-check

Проверка синтаксиса пройдена.

Применим плейбук на узлы и посмотрим, что будет происходить на узлах в связи с обновлением программного обеспечения:

# ansible-playbook /root/ansible-playbooks/03-pb-upgrade-all-packages.yml -i /root/ansible-inventory/my-test-servers.inventory

Долго-долго ждём, особенно если это самое первое обновление после установки узловых систем, и они ни разу еще не обновлялись до этого.

После некоторого всплеска активности **CPU** на узлах **Ansible** всё завершилось.

Все пакеты на узлах успешно обновлены до последнего актуального состояния из доступных узлов репозиториев.

## **1.3. Установка группы полезных программ.**

По учебному сценарию требуется установить пакет программного обеспечения, который состоит из традиционного набора программ для многих серверов: **mc**, **wget**, **tar**.

Создаем файл плейбука:

# mcedit /root/ansible-playbooks/04-pb-progs-install.yml

Записываем в него содержимое:

---

- name: Установка полезных программ на всех узлах

hosts: test\_servers

become\_method: sudo

become\_user: root

tasks:

- name: Установить группу программ

yum:

name: "{{ package }}"

vars:

package:

- mc

- wget

- tar

state: latest

Проверим синтаксис плейбука 04-pb-progs-install.yml специальной утилитой --syntax-check от **Ansible**:

# ansible-playbook /root/ansible-playbooks/04-pb-progs-install.yml --syntax-check

Применим плейбук на узлы и посмотрим как установится группа программного обеспечения:

# ansible-playbook /root/ansible-playbooks/04-pb-progs-install.yml -i /root/ansible-inventory/my-test-servers.inventory

Все пакеты на узлах успешно установлены программы **mc**, **wget**, **tar** из доступных узлов репозиториев.

## **1.4. Настройка межсетевого экрана.**

Для успешной и безопасной работы любого сервера первым делом требуется настраивать его сетевую безопасность. Настроим разрешение обращаться к узлам по **80 порту** и **443 порту**.

Отдельно установим модуль ansible.posix.firewalld на **Ansible** сервер:

# ansible-galaxy collection install ansible.posix

**Ansible** поменяет настройки в файлах конфигурации **firewalld**. Для того, чтобы новые настройки были задействованы, требуется перезапустить **firewalld** на узлах.

В этом перезапуске службы межсетевого экрана нам поможет модуль ansible.builtin.service от создателей **Ansible**.

Создаем файл плейбука:

# mcedit /root/ansible-playbooks/05-pb-http-https.yml

Записываем в него содержимое:

--

- name: Разрешение 80 и 443 порта - firewalld

hosts: test\_servers

become\_method: sudo

become\_user: root

tasks:

- name: Разрешим 80 порт

ansible.posix.firewalld:

service: http

permanent: yes

state: enabled

- name: Разрешим 443 порт

ansible.posix.firewalld:

service: https

permanent: yes

state: enabled

- name: Перезапустим service firewalld на всех узлах

ansible.builtin.service:

name: firewalld

state: restarted

Проверим синтаксис плейбука 05-pb-http-https.yml специальной утилитой --syntax-check от **Ansible**:

# ansible-playbook /root/ansible-playbooks/05-pb-http-https.yml --syntax-check

Проверка синтаксиса пройдена.

**Примечание!** Перед запуском плейбука проверим состояние межсетевого экрана на любом из узлов.

# firewall-cmd --list-all

Как видно — порты **80** и **443** закрыты, то есть отсутствуют в списке разрешенных.

Применим плейбук на узлы разрешим узлам соединения по **80** и **443** порту:

# ansible-playbook /root/ansible-playbooks/05-pb-http-https.yml -i /root/ansible-inventory/my-test-servers.inventory

**Примечание!** Проверим состояние межсетевого экрана на любом из узлов. Для этого необходимо обратить внимание на строку «services», нас интересует наличие в них «http» и «https».

# firewall-cmd --list-all

Как видно — порты **80** и **443** открыты/ Все узлы успешно получили разрешение на использование **80** и **443** порта.

Чтобы открыть какой-либо конкретный порт, сделайте по аналогии:

---

- name: Разрешение в firewalld порта 10051 для zabbix-agent на узлах

hosts:

cos7client1

cos7client2

cos7client3

cos7gla

become\_method: sudo

become\_user: root

tasks:

- name: Открыть порт 10051 по UDP

firewalld:

port: 10051/udp

permanent: yes

state: enabled

- name: Открыть порт 10051 по TCP

firewalld:

port: 10051/tcp

permanent: yes

state: enabled

- name: Перезапустим firewalld на всех узлах

ansible.builtin.service:

name: firewalld

state: restarted

## **1.5. Установка web-сервера Nginx и тестирование отдачи тестовой страницы.**

Создаем файл плейбука:

# mcedit /root/ansible-playbooks/06-pb-nginx-install.yml

Записываем в него содержимое:

---

- name: Установка web-сервера Nginx на всех узлах

hosts: test\_servers

become\_method: sudo

become\_user: root

tasks:

- name: Создаем файл nginx.repo оригинального репозитория Nginx на узлах

copy:

content: "[nginx-mainline]\nname=nginx mainline repo\nbaseurl= https://nginx.org/packages/mainline/rhel/8/x86\_64/\ngpgcheck=1\nenabled=1\ngpgkey=https://nginx.org/keys/nginx\_signing.key\nmodule\_hotfixes=true"

dest: "/etc/yum.repos.d/nginx.repo"

backup: yes

owner: root

group: root

mode: 0644

- name: Установим Nginx на узлы

yum:

name: nginx

state: latest

- name: Запустим Nginx на узлах

ansible.builtin.service:

name: nginx

state: started

Проверим синтаксис плейбука 06-pb-nginx-install.yml специальной утилитой --syntax-check от **Ansible**:

# ansible-playbook /root/ansible-playbooks/06-pb-nginx-install.yml --syntax-check

Проверка синтаксиса пройдена.

Применим плейбук на узлы и посмотрим как установится web-сервера **Nginx** на узлах:

# ansible-playbook /root/ansible-playbooks/06-pb-nginx-install.yml -i /root/ansible-inventory/my-test-servers.inventory

Проверим доступность демонстрационной страницы **HTML** web-сервера на узлах **Ansible**.

Наберем в браузере адрес какого-нибудь узла:

# http://192.168.0.30

**Ответ на всех узлах будет одинаковый:**



Web-сервер **Nginx** развернут. Можете использовать его в своих целях.

Что еще интересного можно сделать:

* поменять на готовую страничку приветствия, через проброс файла;
* написать свою страничку на **HTML5** и подбросить её в виде текстовой строки через плейбук;
* восстановить копию баз данных, подкинуть файлы хостинга и конфигурации **Nginx**;

Если вам не нравится данная стандартная скучная картинка, то ее всегда можно разнообразить красивой заглушкой на **HTML5**.

По умолчанию index.html — заставка **Nginx**, хранится в каталоге /usr/share/nginx/html на узлах, значит, если её принудительно заменить на что-то другое, то **Nginx** это будет добросовестно показывать.

В копировании содержимого **Ansible** каталога в другой каталог узлов, поможет модуль ansible.builtin.copy.

Создадим каталог /root/ansible-files/nginx/ntml5/zastavka для хранения содержимого заглушки на **HTML5**.

# mkdir -p /root/ansible-files/nginx/html5/zastavka

Заранее скачайте заставку на **HTML5** и распакуйте её в каталог /root/ansible-files/nginx/ntml5/zastavka. Будем копировать содержимое этого каталога на узлы **Ansible**.

Создаем файл плейбука:

# mcedit /root/ansible-playbooks/07-pb-nginx-html5.yml

Записываем в него содержимое:

--

- name: Копирование каталогов Ansible в каталог узлов

hosts: test\_servers

become\_method: sudo

become\_user: root

tasks:

# Копируйте дважды, так как иногда файлы пропускаются (в основном только один файл пропускается из папки, если папка не существует).

- name: Копирование из каталога в каталог

copy:

src: "/root/ansible-files/nginx/html5/zastavka/"

dest: "/usr/share/nginx/html/"

owner: root

group: root

mode: 0644

backup: yes

ignore\_errors: true

**Примечание:** Если вы передаете несколько путей с помощью переменной, тогда:

src: "/ корень / {{элемент}}"

**Примечание:** Если вы передаете путь, используя переменную для разных элементов, тогда:

src: "/ root / {{item.source\_path}}"

Проверим синтаксис плейбука 07-pb-nginx-html5.yml специальной утилитой --syntax-check от **Ansible**:

# ansible-playbook /root/ansible-playbooks/07-pb-nginx-html5.yml --syntax-check

Проверка синтаксиса пройдена.

Применим плейбук на узлы и посмотрим как измениться страница приветствия web-сервера **Nginx** на узлах:

# ansible-playbook /root/ansible-playbooks/07-pb-nginx-html5.yml -i /root/ansible-inventory/my-test-servers.inventory

Проверим доступность новой демонстрационной страницы **HTML** web-сервера на узлах **Ansible**.

Наберем в браузере адрес какого-нибудь узла:

# http://192.168.0.39

**Ответ на всех узлах должен быть одинаковым, с вашим примером.**

Web-сервер получит новую заглушку **HTML5** и **Nginx** поменяет страничку приветствия, которая была по умолчанию.

## **1.6. Добавить пользователей с sudo.**

Для обслуживания тестовых серверов создадим пару учетных записей для самых трудолюбивых сотрудников, которые будут работать не покладая лапок. Учетный записи для пользователей: Мурзик и Барсик.

Создадим на самом сервере **Ansible** двух пользователей: **myrzik** и **barsik**. Генерируем для них ключи. Можно использовать и готовые ключи для установки на узлах, однако Барсик и Мурзик не прислали ключи.

Создание:

# adduser myrzik

# adduser barsik

Установка паролей, чтобы они могли заходить на сервер **Ansible**:

# passwd myrzik

# passwd barsik

Генерируем для них ключи:

# su myrzik

# ssh-keygen

На все вопросы для **myrzik** просто жмём **Enter**.

# su barsik

# ssh-keygen

На все вопросы для **barsik** просто жмём **Enter**.

Теперь у нас на сервере **Ansible** появилось две учетные записи для наших пушистых коллег.

Добавим на узлы с помощью **Ansible** учетные записи пользователей **myrzik** и **barsik** вход по ключам, а так же режим sudo без пароля для них.

В работе **Ansible** с учетными записями пользователей поможет модуль ansible.builtin.user.

Создаем файл плейбука:

# mcedit /root/ansible-playbooks/08-pb-sudo-users.yml

Записываем в него содержимое:

--

- name: Дистанционно добавление произвольных учетных записей с Ansible на узлы

hosts: test\_servers

become\_method: sudo

become\_user: root

tasks:

- name: Убедимся, что у нас на узлах есть группа "wheel"

group:

name: wheel

state: present

- name: Разрешим пользователям группы "wheel" иметь sudo без пароля

lineinfile:

dest: /etc/sudoers

state: present

regexp: '^%wheel'

line: '%wheel ALL=(ALL) NOPASSWD: ALL'

validate: visudo -cf %s

- name: Создадим пользователя myrzik и поместим его в группу "wheel"

ansible.builtin.user:

name: myrzik

comment: Cat Myrzik

group: wheel

- name: Пробросим публичный ключ пользователя myrzik с Ansible

authorized\_key:

user: myrzik

state: present

key: "{{ lookup('file', '/home/myrzik/.ssh/id\_rsa.pub') }}"

path: /home/myrzik/.ssh/authorized\_keys

- name: Создадим пользователя barsik и поместим его в группу "wheel"

ansible.builtin.user:

name: barsik

comment: Cat Barsik

group: wheel

- name: Пробросим публичный ключ пользователя barsik с Ansible

authorized\_key:

user: barsik

state: present

key: "{{ lookup('file', '/home/barsik/.ssh/id\_rsa.pub') }}"

path: /home/barsik/.ssh/authorized\_keys

Проверим синтаксис плейбука 08-pb-sudo-users.yml специальной утилитой --syntax-check от **Ansible**:

# ansible-playbook /root/ansible-playbooks/08-pb-sudo-users.yml --syntax-check

Проверка синтаксиса пройдена.

Применим плейбук на узлы и посмотрим, что будет с обновление программного обеспечения:

# ansible-playbook /root/ansible-playbooks/08-pb-sudo-users.yml -i /root/ansible-inventory/my-test-servers.inventory

На все узлы были добавлены учетные записи **myrzik** и **barsik** с входом по ключу и sudo возможностями.

Прогуляемся самостоятельно учетной записью **myrzik** по узлам, потому что у него лапки, и проверим доступ к **sudo**.

# su myrzik

# ssh myrzik@192.168.0.39

# sudo su

Мы стали **root**.

Вернёмся обратно на **Ansible** сервер при помощи череды команд:

# exit

# exit

# exit

Для учетной записи **barsik** аналогично, ведь у него тоже лапки**!**

## **1.7. Удаление учетных записей пользователей.**

Так как пользователи Мурзик и Барсик, начали хулиганить на узлах и ронять сервера своими лапками, срочно удалим их учетные записи со всех узлов!

Создаем файл плейбука:

# mcedit /root/ansible-playbooks/09-pb-users-del.yml

Записываем в него содержимое:

---

- name: Дистанционно удалим произвольные учетные записи на всех узлах

hosts: test\_servers

become\_method: sudo

become\_user: root

tasks:

- name: Удалим учетную запись myrzik на узлах

ansible.builtin.user:

name: myrzik

state: absent

remove: yes

- name: Удалим учетную запись barsik на узлах

ansible.builtin.user:

name: barsik

state: absent

remove: yes

Проверим синтаксис плейбука 09-pb-users-del.yml специальной утилитой --syntax-check от **Ansible**:

# ansible-playbook /root/ansible-playbooks/09-pb-users-del.yml --syntax-check

Проверка синтаксиса пройдена.

Применим плейбук на узлы и посмотрим как удалятся учетные записи:

# ansible-playbook /root/ansible-playbooks/09-pb-users-del.yml -i /root/ansible-inventory/my-test-servers.inventory

Указанные в плейбуке учетные на узлах **Ansible** должны быть удалены.

## **1.8. Запуск консольных команд.**

Иногда на узлах требуется запускать простые банальные консольные команды.

Воспользуемся модулем ansible.builtin.shell от создателей **Ansible**.

Создаем файл плейбука:

# mcedit /root/ansible-playbooks/10-pb-shell.yml

Записываем в него содержимое:

---

- name: Запустим с помощью Ansible команду для командной строки на узлах

hosts: test\_servers

become\_method: sudo

become\_user: root

tasks:

- name: Запустим дистанционно запись генерированной строки в файл

ansible.builtin.shell: openssl rand -base64 32 >> somefile.txt

args:

chdir: /root

creates: somefile.txt

- name: Скопируем полученный файл внутри каталога

ansible.builtin.shell: cp /root/somefile.txt /root/somefile2.txt

Проверим синтаксис плейбука 10-pb-shell.yml специальной утилитой --syntax-check от **Ansible**:

# ansible-playbook /root/ansible-playbooks/10-pb-shell.yml --syntax-check

Применим плейбук на узлы и посмотрим, как удалятся учетные записи:

# ansible-playbook /root/ansible-playbooks/10-pb-shell.yml -i /root/ansible-inventory/my-test-servers.inventory

Команда для командной строки на узлах **Ansible** исполнена.

Есть еще способы запустить консольную команду:

---

- name: Установка zabbix-agent на узлах

hosts:

cos7client1

cos7client2

cos7client3

become\_method: sudo

become\_user: root

tasks:

- name: Установка репозитория docker и установка docker на узлы

ansible.builtin.shell: "curl -fsSL https://get.docker.com/ | sh"

- name: Поставим в автозапуск docker на узлах

ansible.builtin.service:

name: docker

enabled: yes

- name: Запустим docker на узлах

ansible.builtin.service:

name: docker

state: started

- name: Версия docker на узлах

ansible.builtin.shell: docker info

- name: Executing a Command Using Shell Module

shell: docker info

- name: Executing a command using command module

command: docker info

Для того, что бы увидеть что выводит команда в консоль узла, введите ключ -vvv:

# ansible-playbook /ansible/roles/docker-install.yml -i /ansible/hosts/pb-zs-stend.yaml -l cos7client1 -vvv

## **1.9. Объединение плейбуков в Основную книгу.**

Если ваши плейбуки должны работать в определённом порядке и если все они являются обязательными, то создайте **Основную книгу** и включите файлы с задачами с помощью операторов import\_playbook.

Например:

---

- import\_playbook: A-systemd-networkd.yml

- import\_playbook: B-fail2ban-ssh.yml

- import\_playbook: C-enable-watchdog.yml

Создадим **Основную книгу**:

# mcedit /root/ansible-playbooks/my-main-book.yml

Вот такой вот будет **Основная книга** нашем в учебном варианте:

---

- name: Запустим с помощью Ansible сборник плейбуков - Основная книга

hosts: test\_servers

become\_method: sudo

become\_user: root

- import\_playbook: 01-pb-selinux-disable.yml

- import\_playbook: 02-pb-reboot.yml

- import\_playbook: 03-pb-upgrade-all-packages.yml

- import\_playbook: 04-pb-progs-install.yml

- import\_playbook: 05-pb-http-https.yml

- import\_playbook: 06-pb-nginx-install.yml

- import\_playbook: 07-pb-nginx-html5.yml

- import\_playbook: 08-pb-sudo-users.yml

- import\_playbook: 09-pb-users-del.yml

- import\_playbook: 10-pb-shell.yml

Проверка синтаксиса плейбука:

# ansible-playbook /root/ansible-playbooks/my-main-book.yml --syntax-check

Применим **Основную книгу** на узлы и посмотрим, как по экрану поползет длинная вереница событий на узлах:

# ansible-playbook /root/ansible-playbooks/my-main-book.yml -i /root/ansible-inventory/my-test-servers.inventory

В отчете обязательно приведите:

* результат работы каждого плейбука
* уникальный пример творческой работы в разделе 1.5