



МИНОБРНАУКИ РОССИИ
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«МИРЭА – Российский технологический университет»
РТУ МИРЭА

Институт информационных технологий
Кафедра Математического обеспечения и стандартизации
информационных технологий

Отчет по практической работе №4

по дисциплине
«Системное программное обеспечение»

Тема практической работы:
«Системы конфигурационного управления»

Выполнил студент группы ИВБО-02-19

К. Ю. Денисов

Проверил ассистент

Ю. А. Вороноцов

Москва 2021

Содержание

1	Подготовка инфраструктуры	3
2	Использование Ansible для конфигурации хостов	4
3	Более сложный playbook	5
4	Роли Ansible	6
5	Индивидуальное задание	7
6	Вывод	9
	ПРИЛОЖЕНИЕ А	10

1 Подготовка инфраструктуры

В ходе данной практической работы мы будем использовать три виртуальные машины под управлением ОС GNU/Linux Debian 8.3.0, одна из которых будет управляющей и будет выполнять конфигурацию и настройку двух управляемых машин.

Настроим ip-адрес управляемых машин. См рисунки 1 и 2 в Приложении А. Установим **ansible** и сервер **ssh** на управляющую машину. См. рисунок 5 и 3.

Настроим inventory-файл:

/home/denilai/Desktop/hosts

```
1 [webservers]
  server-1 ansible_host=192.168.43.101
3 server-2 ansible_host=192.168.43.201
5 [webservers:vars]
  ansible_python_interpreter=/usr/bin/python3
```

После установки данных сервисов, проверим возможность подключения по ssh, предварительно сгенерировав ключи командой **ssh-keygen** и передав их на управляемые машины командами:

```
$ ssh-copy-id root@192.168.43.101
2 $ ssh-copy-id root@192.168.43.201
```

См. рисунок 4.

Выполним следующие команды для проверки работы ansible:

```
$ ansible -i ./hosts -m ping all
2 $ ansible -i ./hosts -m command -a free all
```

См. рисунок 6.

2 Использование Ansible для конфигурации ХОСТОВ

Напишем playbook, который установит веб-сервер Nginx на управляемые хосты.

/home/denilai/Desktop/playbook-nginx.yml

```
2  - name: Install Nginx to Webservers
    hosts: webservers
    become: yes

6  tasks:
    - name: Install Nginx
      apt:
          name=nginx state=present update_cache=yes
10     when:
          ansible_os_family == "Debian"
12     notify:
          - Nginx Systemd

14     handlers:
16     - name: Nginx Systemd
      systemd:
18         name: nginx
          enabled: yes
20         state: started
```

Выполним данную команду для выполнения playbook-a.

```
$ ansible-playbook -i hosts playbook-nginx.yml
```

После выполнения playbook-a, проверим статус **nginx** на первой управляемой машине. См. рисунки **7** и **8**.

3 Более сложный playbook

Приведем содержание более сложного playbook-a:

/home/denilai/Desktop/playbook-param-nginx.yml

```
1  ---
2  - name: Install and config Nginx
3    hosts: webservers
4    become: yes
5
6    vars:
7      html_dir: /usr/share/nginx/html
8      greeting: "Hello Everybody!"
9
10   tasks:
11     - name: Install Nginx
12       apt: name=nginx state=present update_cache=yes
13       when: ansible_os_family == "Debian"
14       notify:
15         - Nginx Systemd
16
17     - name: Delete default HTML files
18       shell: /bin/rm -rf /usr/share/nginx/html/*.html
19       args:
20         warn: false
21
22     - name: Replace config file
23       vars:
24         nginx_user: user
25         worker_processes: 2
26         worker_connections: 256
27       template:
28         src: templates/nginx.conf.j2
29         dest: /etc/nginx/nginx.conf
30         mode: 0644
31       register: result
32       failed_when: result.failed == true
33       notify: Reload Nginx
34
35     - name: Copy index file
36       copy: src=files/index.html dest={{ html_dir }} mode=0644
37       notify: Reload Nginx
38
39     - name: Generate dynamic HTML from template
40       template: src=templates/hello.html.j2 dest={{ html_dir
```

```

    }}/hello.html owner=root mode=0644
41   notify: Reload Nginx

43   handlers:
    - name: Nginx Systemd
45     systemd:
        name: nginx
47         enabled: yes
        state: started
49   - name: Reload Nginx
    systemd: name=nginx state=reloaded

```

Теперь выполним команду для запуска playbook'а с пробным прогоном, который позволит проверить корректность написанного playbook'а без внесения изменений на целевые узлы.

```
1 $ ansible-playbook -i hosts playbook-param-nginx.yml --check
```

Результат проверки можно увидеть на рисунке 9.

После удачной проверки, выполним playbook, исключив из команды ключ **-check**, а затем запросим базовую страницу при помощи утилиты curl. См. рисунок 10.

4 Роли Ansible

Попробуем создать свою роль для установки ранее разобранного playbook'а nginx. Сперва вернёмся в рабочую папку со всеми файлами для Ansible и создадим там директорию **roles**. После чего перейдём в эту директорию и иницилируем роль стандартной структуры при помощи данной команды:

```
1 $ ansible-galaxy init nginx
```

При помощи команды **tree** можно опять же посмотреть структуру директории.

Перейдём к созданию роли. Заполним соответствующие файлы данными из секций `playbook`'а и соответствующие директории ранее созданными файлами.

Теперь напишем `playbook`, запускающий созданную роль. Для вызова составленной роли в `playbook`'е используется секция **roles**. Создадим файл **nginx-role.yml** в рабочей директории `ansible` и заполним его следующим образом.

/home/denilai/Desktop/nginx-role.yml

```
1 ---
2 - name: Install and config Nginx via Role
3   hosts: webserver
4   become: yes
5
6   roles:
7     - nginx
```

После этого выполним данный `playbook` на первом сервере. См. рисунок 11.

5 Индивидуальное задание

В качестве индивидуального задания для **варианта 6** предлагается написать роль для сервера **nginx**, написать `playbook` для применения роли, провести тестовый запуск `playbook`'а, в случае успешного прохождения теста, применить `playbook` к серверам.

Необходимо добавить переменную, содержащую ФИО, номер группы и номер варианта.

Данная переменная должна выводиться в шаблонный файл `nginx`. Установка пакета выполняется при помощи модуля **APT**, используемого для установки **nginx** в базовой роли.

/home/denilai/Desktop/nginx-role-sql.yml

```
1 ---
2 - name: Install and config Nginx via Role
3   hosts: webserver
4   become: yes
```

```
5
roles:
7   - mysql
```

Добавим еще одну секцию в файл `/tasks/main.yml`, чтобы установить пакет клиента базы данных MySQL.

```
2   - name: Install Mysql client
apt:
name=default-mysql-client
4 state=present
update_cache=yes
```

Изменим содержание файла **hello.html.j2** — добавим ФИО студента и номер варианта.

`/home/denilai/Desktop/hello.html.j2`

```
1 "Server {{ ansible_hostname }} ( ip {{ ansible_default_ipv4.address }} )
   greets you: {{ greeting | default("Hello") }}!
author: "Denisov Kirill IVBO-02-19 6 VAR"
```

Протестируем корректность составления роли, выполнив следующую команду:

```
$ ansible-playbook -i ./hosts nginx-role-sql.yml --check
```

См. рисунок 12.

Применим playbook к серверам. См. рисунок См. рисунок 13. Далее проверим содержание созданной при помощи шаблона страницы **hello** с помощью данной команды:

```
1 $ curl -L http://192.168.43.101/hello
```

См. рисунок 14.

6 Вывод

В результате выполнения данной практической работы мы научились использовать систему конфигурационного управления Ansible. Написали playbook для применения роли для сервера nginx, протестировали его работу.

ПРИЛОЖЕНИЕ А

```
root@first:/etc/network# cat interfaces
# This file describes the network interfaces available on your system
# and how to activate them. For more information, see interfaces(5).

source /etc/network/interfaces.d/*

# The loopback network interface
auto lo
iface lo inet loopback

# The primary network interface
allow-hotplug enp0s3
iface enp0s3 inet static
address 192.168.43.101
mask 255.255.255.0
gateway 192.168.43.1
root@first:/etc/network#
```

Рис. 1: Файл /etc/networks/interfaces 1-ой машины

```
root@second:~# cat /etc/network/interfaces
# This file describes the network interfaces available on your system
# and how to activate them. For more information, see interfaces(5).

source /etc/network/interfaces.d/*

# The loopback network interface
auto lo
iface lo inet loopback

# The primary network interface
allow-hotplug enp0s3
iface enp0s3 inet static
address 192.168.43.201
mask 255.255.255.0
gateway 192.168.43.1
root@second:~#
```

Рис. 2: Файл /etc/networks/interfaces 2-ой машины

```

root@lirik-pc:~/ansible# systemctl status ssh
● ssh.service - OpenBSD Secure Shell server
   Loaded: loaded (/lib/systemd/system/ssh.service; enabled; vendor preset: enabled)
   Active: active (running) since Sat 2021-05-22 20:31:12 MSK; 51min ago
     Docs: man:sshd(8)
           man:sshd_config(5)
   Process: 724 ExecStartPre=/usr/sbin/sshd -t (code=exited, status=0/SUCCESS)
  Main PID: 767 (sshd)
    Tasks: 1 (limit: 9272)
   Memory: 2.7M
   CGroup: /system.slice/ssh.service
           └─767 sshd: /usr/sbin/sshd -D [listener] 0 of 10-100 startups

мая 22 20:31:12 lirik-pc systemd[1]: Starting OpenBSD Secure Shell server...
мая 22 20:31:12 lirik-pc sshd[767]: Server listening on 0.0.0.0 port 22.
мая 22 20:31:12 lirik-pc sshd[767]: Server listening on :: port 22.
мая 22 20:31:12 lirik-pc systemd[1]: Started OpenBSD Secure Shell server.
root@lirik-pc:~/ansible#

```

Рис. 3: Версия сервера ssh

```

root@lirik-pc:~/ansible# ssh root@192.168.43.101
Linux first 4.19.0-16-amd64 #1 SMP Debian 4.19.181-1 (2021-03-19) x86_64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Sat May 22 20:55:36 2021 from 192.168.43.24
root@first:~# exit
выход
Connection to 192.168.43.101 closed.
root@lirik-pc:~/ansible# ssh root@192.168.43.201
Linux second 4.19.0-16-amd64 #1 SMP Debian 4.19.181-1 (2021-03-19) x86_64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Sat May 22 20:51:27 2021 from 192.168.43.24
root@second:~#

```

Рис. 4: Подключение к машинам

```

root@lirik-pc:~/ansible# ansible --version
ansible 2.9.6
  config file = /etc/ansible/ansible.cfg
  configured module search path = ['/root/.ansible/plugins/modules', '/usr/share/ansible/plugins/modules']
  ansible python module location = /usr/lib/python3/dist-packages/ansible
  executable location = /usr/bin/ansible
  python version = 3.8.5 (default, Jul 28 2020, 12:59:40) [GCC 9.3.0]
root@lirik-pc:~/ansible#

```

Рис. 5: Версия ansible

```

root@lirik-pc:~/ansible# ansible -i ./hosts -m ping all
server-2 | SUCCESS => {
  "changed": false,
  "ping": "pong"
}
server-1 | SUCCESS => {
  "changed": false,
  "ping": "pong"
}
root@lirik-pc:~/ansible# ansible -i ./hosts -m command -a free all
server-2 | CHANGED | rc=0 >>


|       | total   | used  | free    | shared | buff/cache | available |
|-------|---------|-------|---------|--------|------------|-----------|
| Mem:  | 2043248 | 96284 | 1519844 | 8156   | 427120     | 1789504   |
| Swap: | 998396  | 0     | 998396  |        |            |           |


server-1 | CHANGED | rc=0 >>


|       | total   | used  | free    | shared | buff/cache | available |
|-------|---------|-------|---------|--------|------------|-----------|
| Mem:  | 2043248 | 96220 | 1519708 | 8156   | 427320     | 1789516   |
| Swap: | 998396  | 0     | 998396  |        |            |           |


root@lirik-pc:~/ansible#

```

Рис. 6: Простые команды

```

root@lirik-pc:~/ansible# ansible-playbook -i ./hosts playbook-nginx.yml
PLAY [Install Nginx to Webservers] *****
TASK [Gathering Facts] *****
ok: [server-1]
ok: [server-2]
TASK [Install Nginx] *****
changed: [server-1]
changed: [server-2]
RUNNING HANDLER [Nginx Systemd] *****
ok: [server-1]
ok: [server-2]
PLAY RECAP *****
server-1 : ok=3    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
server-2 : ok=3    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
root@lirik-pc:~/ansible#

```

Рис. 7: Простой playbook

```

root@first:~# systemctl status nginx
● nginx.service - A high performance web server and a reverse proxy server
   Loaded: loaded (/lib/systemd/system/nginx.service; enabled; vendor preset: enabled)
   Active: active (running) since Sat 2021-05-22 17:52:51 MSK; 3h 54min ago
     Docs: man:nginx(8)
   Process: 13435 ExecStartPre=/usr/sbin/nginx -t -q -g daemon on; master_process on; (code=exited, status=0/SUCCESS)
   Process: 13436 ExecStart=/usr/sbin/nginx -g daemon on; master_process on; (code=exited, status=0/SUCCESS)
   Process: 17466 ExecReload=/usr/sbin/nginx -g daemon on; master_process on; -s reload (code=exited, status=0/SUCCESS)
  Main PID: 13437 (nginx)
    Tasks: 3 (limit: 2359)
   Memory: 2.1M
   CGroup: /system.slice/nginx.service
           └─13437 nginx: master process /usr/sbin/nginx -g daemon on; master_process on;
             └─17467 nginx: worker process
               └─17468 nginx: worker process

may 22 17:52:51 first systemd[1]: Starting A high performance web server and a reverse proxy server...
may 22 17:52:51 first systemd[1]: nginx.service: Failed to parse PID from file /run/nginx.pid: Invalid argument
may 22 17:52:51 first systemd[1]: Started A high performance web server and a reverse proxy server.
may 22 19:01:05 first systemd[1]: Reloading A high performance web server and a reverse proxy server.
may 22 19:01:05 first systemd[1]: Reloading A high performance web server and a reverse proxy server.
may 22 19:05:17 first systemd[1]: Reloading A high performance web server and a reverse proxy server.
may 22 19:05:17 first systemd[1]: Reloading A high performance web server and a reverse proxy server.
may 22 19:05:17 first systemd[1]: Reloading A high performance web server and a reverse proxy server.
may 22 19:51:08 first systemd[1]: Reloading A high performance web server and a reverse proxy server.
may 22 19:51:08 first systemd[1]: Reloading A high performance web server and a reverse proxy server.
root@first:~#

```

Рис. 8: Статус nginx

```

root@lirik-pc:~/ansible# ansible-playbook -i ./hosts playbook-param-nginx.yml --check

PLAY [Install and config Nginx] *****

TASK [Gathering Facts] *****
ok: [server-2]
ok: [server-1]

TASK [Install Nginx] *****
ok: [server-2]
ok: [server-1]

TASK [Delete default HTML files] *****
skipping: [server-1]
skipping: [server-2]

TASK [Replace config file] *****
ok: [server-1]
ok: [server-2]

TASK [Copy index file] *****
changed: [server-1]
changed: [server-2]

TASK [Generate dynamic HTML from template] *****
changed: [server-2]
changed: [server-1]

RUNNING HANDLER [Reload Nginx] *****
changed: [server-1]
changed: [server-2]

PLAY RECAP *****
server-1      : ok=6    changed=3    unreachable=0    failed=0    skipped=1    rescued=0    ignored=0
server-2      : ok=6    changed=3    unreachable=0    failed=0    skipped=1    rescued=0    ignored=0

root@lirik-pc:~/ansible#

```

Рис. 9: Проверка сложного playbook-а

```

root@lirik-pc:~/ansible# ansible-playbook -i ./hosts playbook-param-nginx.yml

PLAY [Install and config Nginx] *****
TASK [Gathering Facts] *****
ok: [server-1]
ok: [server-2]

TASK [Install Nginx] *****
ok: [server-2]
ok: [server-1]

TASK [Delete default HTML files] *****
changed: [server-2]
changed: [server-1]

TASK [Replace config file] *****
ok: [server-2]
ok: [server-1]

TASK [Copy index file] *****
changed: [server-2]
changed: [server-1]

TASK [Generate dynamic HTML from template] *****
changed: [server-1]
changed: [server-2]

RUNNING HANDLER [Reload Nginx] *****
changed: [server-1]
changed: [server-2]

PLAY RECAP *****
server-1      : ok=7    changed=4    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
server-2      : ok=7    changed=4    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0

root@lirik-pc:~/ansible# curl -L 192.168.43.101
Web server is working!
root@lirik-pc:~/ansible#

```

Рис. 10: Запуск сложного playbook-а

```

root@lirik-pc:~/ansible# ansible-playbook -i ./hosts nginx-role.yml

PLAY [Install and config Nginx via Role] *****
TASK [Gathering Facts] *****
ok: [server-2]
ok: [server-1]

TASK [nginx : Install Nginx] *****
ok: [server-1]
ok: [server-2]

TASK [nginx : Delete default HTML files] *****
changed: [server-1]
changed: [server-2]

TASK [nginx : Replace config file] *****
ok: [server-1]
ok: [server-2]

TASK [nginx : Copy index file] *****
changed: [server-2]
changed: [server-1]

TASK [nginx : Generate dynamic HTML from template] *****
changed: [server-1]
changed: [server-2]

RUNNING HANDLER [nginx : Reload Nginx] *****
changed: [server-1]
changed: [server-2]

PLAY RECAP *****
server-1      : ok=7    changed=4    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
server-2      : ok=7    changed=4    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0

root@lirik-pc:~/ansible#

```

Рис. 11: Запуск playbook-а с ролью

```

root@lirik-pc:~/ansible# ansible-playbook -i ./hosts nginx-role-sql.yml --check
PLAY [Install and config Nginx via Role] *****
TASK [Gathering Facts] *****
ok: [server-1]
ok: [server-2]

TASK [mysql : Install Nginx] *****
ok: [server-1]
ok: [server-2]

TASK [mysql : Install Mysql client] *****
ok: [server-2]
ok: [server-1]

TASK [mysql : Delete default HTML files] *****
skipping: [server-2]
skipping: [server-1]

TASK [mysql : Replace config file] *****
ok: [server-1]
ok: [server-2]

TASK [mysql : Copy index file] *****
changed: [server-1]
changed: [server-2]

TASK [mysql : Generate dynamic HTML from template] *****
changed: [server-2]
changed: [server-1]

RUNNING HANDLER [mysql : Reload Nginx] *****
changed: [server-1]
changed: [server-2]

PLAY RECAP *****
server-1 : ok=7 changed=3 unreachable=0 failed=0 skipped=1 rescued=0 ignored=0
server-2 : ok=7 changed=3 unreachable=0 failed=0 skipped=1 rescued=0 ignored=0
root@lirik-pc:~/ansible#

```

Рис. 12: Проверка индивидуального playbook-а с ролью

```

root@lirik-pc:~/ansible# ansible-playbook -i ./hosts nginx-role-sql.yml
PLAY [Install and config Nginx via Role] *****
TASK [Gathering Facts] *****
ok: [server-2]
ok: [server-1]

TASK [mysql : Install Nginx] *****
ok: [server-1]
ok: [server-2]

TASK [mysql : Install Mysql client] *****
ok: [server-2]
ok: [server-1]

TASK [mysql : Delete default HTML files] *****
changed: [server-1]
changed: [server-2]

TASK [mysql : Replace config file] *****
ok: [server-2]
ok: [server-1]

TASK [mysql : Copy index file] *****
changed: [server-1]
changed: [server-2]

TASK [mysql : Generate dynamic HTML from template] *****
changed: [server-2]
changed: [server-1]

RUNNING HANDLER [mysql : Reload Nginx] *****
changed: [server-1]
changed: [server-2]

PLAY RECAP *****
server-1 : ok=8 changed=4 unreachable=0 failed=0 skipped=0 rescued=0 ignored=0
server-2 : ok=8 changed=4 unreachable=0 failed=0 skipped=0 rescued=0 ignored=0
root@lirik-pc:~/ansible#

```

Рис. 13: Запуск индивидуального playbook-а с ролью

```

root@lirik-pc:~/ansible/roles/mysql/templates# curl -L 192.168.43.101
Web server is working!
author: "Denisov Kirill IVB0-02-19 6 VAR"
root@lirik-pc:~/ansible/roles/mysql/templates# curl -L 192.168.43.101/hello
'Server first ( ip 192.168.43.101 ) greets you: Hello Everybody!!'
author: "Denisov Kirill IVB0-02-19 6 VAR"
root@lirik-pc:~/ansible/roles/mysql/templates#

```

Рис. 14: Страница hello