

PYTHON ДЛЯ СЕТЕВЫХ ИНЖЕНЕРОВ



МОДУЛИ ДЛЯ РАБОТЫ С СЕТЕВЫМ ОБОРУДОВАНИЕМ



МОДУЛЬ IOS_FACTS



МОДУЛЬ IOS_FACTS

Модуль `ios_facts` - собирает информацию с устройств под управлением IOS.

Информация берется из таких команд:

- `dir`
- `show version`
- `show memory statistics`
- `show interfaces`
- `show ipv6 interface`
- `show lldp`
- `show lldp neighbors detail`
- `show running-config`

МОДУЛЬ IOS_FACTS

В модуле можно указывать какие параметры собирать - можно собирать всю информацию, а можно только подмножество. По умолчанию, модуль собирает всю информацию, кроме конфигурационного файла.

Какую информацию собирать, указывается в параметре `gather_subset`.

МОДУЛЬ IOS_FACTS

Поддерживаются такие варианты (указаны также команды, которые будут выполняться на устройстве):

- **all**
- **hardware**
 - dir
 - show version
 - show memory statistics
- **config**
 - show version
 - show running-config

МОДУЛЬ IOS_FACTS

- **interfaces**
 - dir
 - show version
 - show interfaces
 - show ipv6 interface
 - show lldp
 - show lldp neighbors detail

МОДУЛЬ IOS_FACTS

Собрать все факты:

```
- ios_facts:  
  gather_subset: all
```


МОДУЛЬ IOS_FACTS

Собрать только подмножество interfaces:

```
- ios_facts:  
  gather_subset:  
    - interfaces
```

МОДУЛЬ IOS_FACTS

Собрать всё, кроме hardware:

```
- ios_facts:  
  gather_subset:  
    - "!hardware"
```

МОДУЛЬ IOS_FACTS

Ansible собирает такие факты:

- `ansible_net_all_ipv4_addresses` - список IPv4 адресов на устройстве
- `ansible_net_all_ipv6_addresses` - список IPv6 адресов на устройстве
- `ansible_net_config` - конфигурация (для Cisco `sh run`)
- `ansible_net_filesystems` - файловая система устройства
- `ansible_net_gather_subset` - какая информация собирается (hardware, default, interfaces, config)
- `ansible_net_hostname` - имя устройства

МОДУЛЬ IOS_FACTS

- `ansible_net_image` - имя и путь ОС
- `ansible_net_interfaces` - словарь со всеми интерфейсами устройства. Имена интерфейсов - ключи, а данные - параметры каждого интерфейса
- `ansible_net_memfree_mb` - сколько свободной памяти на устройстве
- `ansible_net_memtotal_mb` - сколько памяти на устройстве
- `ansible_net_model` - модель устройства
- `ansible_net_serialnum` - серийный номер
- `ansible_net_version` - версия IOS

ИСПОЛЬЗОВАНИЕ МОДУЛЯ

Пример playbook 1_ios_facts.yml с использованием модуля ios_facts (собираются все факты):

```
- name: Collect IOS facts
  hosts: cisco-routers

  tasks:

    - name: Facts
      ios_facts:
        gather_subset: all
```

ИСПОЛЬЗОВАНИЕ МОДУЛЯ

```
$ ansible-playbook 1_ios_facts.yml
```

```
SSH password:

PLAY [Collect IOS facts] *****

TASK [Facts] *****
ok: [192.168.100.1]
ok: [192.168.100.2]
ok: [192.168.100.3]

PLAY RECAP *****
192.168.100.1      : ok=1    changed=0    unreachable=0    failed=0
192.168.100.2      : ok=1    changed=0    unreachable=0    failed=0
192.168.100.3      : ok=1    changed=0    unreachable=0    failed=0
```

ИСПОЛЬЗОВАНИЕ МОДУЛЯ

Для того, чтобы посмотреть, какие именно факты собираются с устройства, можно добавить флаг `-v` (информация сокращена):

```
$ ansible-playbook 1_ios_facts.yml -v
Using /home/nata/pyneng_course/chapter15/ansible.cfg as config file
```

```
SSH password:

PLAY [Collect IOS facts] *****

TASK [Facts] *****
ok: [192.168.100.1] => {"ansible_facts": {"ansible_net_all_ipv4_addresses": ["192.168.200.1", "192.168.100.1", "10.1.1.1"], "ansible_net_all_ipv6_addresses": [], "ansible_net_config": "Building configuration...\n\nCurrent configuration : 6716 bytes\n\n! Last configuration change at 09:09:04 UTC Sun Dec 18 2016\nversion 15.2\nno service times
```

ИСПОЛЬЗОВАНИЕ МОДУЛЯ

После того, как Ansible собрал факты с устройства, все факты доступны как переменные в playbook, шаблонах и т.д.

ИСПОЛЬЗОВАНИЕ МОДУЛЯ

Например, можно отобразить содержимое факта с помощью debug (playbook 2_ios_facts_debug.yml):

```
- name: Collect IOS facts
  hosts: 192.168.100.1

  tasks:

    - name: Facts
      ios_facts:
        gather_subset: all

    - name: Show ansible_net_all_ipv4_addresses fact
      debug: var=ansible_net_all_ipv4_addresses

    - name: Show ansible_net_interfaces fact
      debug: var=ansible_net_interfaces['Ethernet0/0']
```

ИСПОЛЬЗОВАНИЕ МОДУЛЯ

Результат выполнения playbook:

```
$ ansible-playbook 2_ios_facts_debug.yml
```

```
SSH password:

PLAY [Collect IOS facts] *****

TASK [Facts] *****
ok: [192.168.100.1]

TASK [Show ansible_net_all_ipv4_addresses fact] *****
ok: [192.168.100.1] => {
  "ansible_net_all_ipv4_addresses": [
    "192.168.200.1",
    "192.168.100.1"
  ]
}

TASK [Show fact] *****
ok: [192.168.100.1] => {
  "ansible_net_interfaces['Ethernet0/0']": {
    "bandwidth": 10000,
    "description": null,
    "duplex": null,
    "ipv4": {
      "address": "192.168.100.1",
      "masklen": 24
    },
    "lineprotocol": "up ",
    "macaddress": "aabb.cc00.6500",
    "mediatype": null,
    "mtu": 1500,
    "operstatus": "up",
    "type": "AmdP2"
  }
}

PLAY RECAP *****
192.168.100.1      : ok=3    changed=0    unreachable=0    failed=0
```

СОХРАНЕНИЕ ФАКТОВ

В том виде, в котором информация отображается в режиме verbose, довольно сложно понять какая информация собирается об устройствах. Для того, чтобы лучше понять какая информация собирается об устройствах, в каком формате, скопируем полученную информацию в файл.

Для этого будет использоваться модуль сору.

СОХРАНЕНИЕ ФАКТОВ

Playbook 3_ios_facts.yml:

```
- name: Collect IOS facts
  hosts: cisco-routers

  tasks:

    - name: Facts
      ios_facts:
        gather_subset: all
        register: ios_facts_result

    #- name: Create all_facts dir
    #  file:
    #    path: ./all_facts/
    #    state: directory
    #    mode: 0755

    - name: Copy facts to files
      copy:
        content: "{{ ios_facts_result | to_nice_json }}"
        dest: "all_facts/{{ inventory_hostname }}_facts.json"
```

СОХРАНЕНИЕ ФАКТОВ

Модуль `copy` позволяет копировать файлы с управляющего хоста (на котором установлен Ansible) на удаленный хост. Но, так как в этом случае, указан параметр `connection: local`, файлы будут скопированы на локальный хост.

Чаще всего, модуль `copy` используется таким образом:

```
- copy:
  src: /srv/myfiles/foo.conf
  dest: /etc/foo.conf
```

СОХРАНЕНИЕ ФАКТОВ

Но, в данном случае, нет исходного файла, содержимое которого нужно скопировать. Вместо этого, есть содержимое переменной `ios_facts_result`, которое нужно перенести в файл `all_facts/{{inventory_hostname}}_facts.json`.

Для того чтобы перенести содержимое переменной в файл, в модуле `copy`, вместо `src`, используется параметр `content`.

СОХРАНЕНИЕ ФАКТОВ

```
content: "{{ ios_facts_result | to_nice_json }}"
```

Параметр `to_nice_json` - это фильтр Jinja2, который преобразует информацию переменной в формат, в котором удобней читать информацию. Переменная в формате Jinja2 должна быть заключена в двойные фигурные скобки, а также указана в двойных кавычках.

Так как в пути `dest` используются имена устройств, будут сгенерированы уникальные файлы для каждого устройства.

Результат выполнения playbook:

```
$ ansible-playbook 3_ios_facts.yml
```

```
SSH password:

PLAY [Collect IOS facts] *****

TASK [Facts] *****
ok: [192.168.100.1]
ok: [192.168.100.2]
ok: [192.168.100.3]

TASK [Copy facts to files] *****
changed: [192.168.100.3]
changed: [192.168.100.1]
changed: [192.168.100.2]

PLAY RECAP *****
192.168.100.1      : ok=2    changed=1    unreachable=0    failed=0
192.168.100.2      : ok=2    changed=1    unreachable=0    failed=0
192.168.100.3      : ok=2    changed=1    unreachable=0    failed=0
```

СОХРАНЕНИЕ ФАКТОВ

После этого, в каталоге all_facts находятся такие файлы:

```
192.168.100.1_facts.json  
192.168.100.2_facts.json  
192.168.100.3_facts.json
```

СОХРАНЕНИЕ ФАКТОВ

Содержимое файла all_facts/192.168.100.1_facts.json:

```
{
  "ansible_facts": {
    "ansible_net_all_ipv4_addresses": [
      "192.168.200.1",
      "192.168.100.1",
      "10.1.1.1"
    ],
    "ansible_net_all_ipv6_addresses": [],
    "ansible_net_config": "Building configuration...\n\nCurrent configuration :
  ...
```

СОХРАНЕНИЕ ФАКТОВ

Сохранение информации об устройствах, не только поможет разобраться, какая информация собирается, но и может быть полезным для дальнейшего использования информации. Например, можно использовать факты об устройстве в шаблоне.

При повторном выполнении playbook, Ansible не будет изменять информацию в файлах, если факты об устройстве не изменились

СОХРАНЕНИЕ ФАКТОВ

Если информация изменилась, для соответствующего устройства, будет выставлен статус `changed`. Таким образом, по выполнению `playbook` всегда понятно, когда какая-то информация изменилась.

Повторный запуск playbook (без изменений):

```
$ ansible-playbook 3_ios_facts.yml
```

SSH password:

PLAY [Collect IOS facts] *****

TASK [Facts] *****

ok: [192.168.100.1]

ok: [192.168.100.3]

ok: [192.168.100.2]

TASK [Copy facts to files] *****

ok: [192.168.100.2]

ok: [192.168.100.1]

ok: [192.168.100.3]

PLAY RECAP *****

192.168.100.1	: ok=2	changed=0	unreachable=0	failed=0
---------------	--------	-----------	---------------	----------

192.168.100.2	: ok=2	changed=0	unreachable=0	failed=0
---------------	--------	-----------	---------------	----------

192.168.100.3	: ok=2	changed=0	unreachable=0	failed=0
---------------	--------	-----------	---------------	----------