**Документация по Ansible** [**https://docs.ansible.com/ansible/latest/index.html**](https://docs.ansible.com/ansible/latest/index.html)

**Установка Ansible**

**Установка Ansible на Debian:**

1. Заходим в файл /etc/apt/sources.list и добавляем строку

deb http://ppa.launchpad.net/ansible/ansible/ubuntu trusty main

1. Пишем команды sudo apt-key adv --keyserver keyserver.ubuntu.com --recv-keys 93C4A3FD7BB9C367
2. Обновляем базу откуда скачивать sudo apt update
3. Устанавливаем Ansible: sudo apt install ansible -y
4. Проверяем версию ansible --version
5. Создаем тестовую директорию project и заходим в нее
6. Создаем файл inventory и в нем пишем Myubuntu ansible\_host=192.168.100.32 ansible\_ssh\_pass=1234
7. Проверяем ansible Myubuntu -m ping -i inventory



1. Добавляем ключ ssh контроллера Ansible: для этого создаем пару ключ с помощью команды ssh-keygen и потом закидываем на клиенты с помощью команды ssh-copy-id username@remote\_host.

**Файл Inventory**

**Файл Inventory по умолчанию находится по пути /etc/ansible/hosts**

**ansible-inventory --list -**показывает какие группы и переменные есть в inventory

**Команды в файле Inventory:**

Пример 1) Linux и 2)Windows:

1. Serv1 ansible\_host=Serv1.4dgp.local ansible\_connection=ssh ansible\_user=sadmin ansible\_ssh\_pass=1234
2. Serv2 ansible\_host=Serv2.4dgp.local ansible\_connection=winrm ansible\_user=sadmin ansible\_password=1234

Где Serv\* - короткое имя обращения (alias), ansible\_host – полное имя FQDN, ansible\_connection - через что подключаемся (для Linux = ssh, для Windows = winrm), ansible\_user = имя юзера через кого подключаемся, ansible\_ssh\_pass и ansible\_password пароль юзера для Linux и Windows соответственно.

Также еще есть ansible\_port=22/5896 – через какой порт подключаться

**Объединение групп в общую группу:**

[all:children]

Serv1

Serv2

**Playbooks and Moduls**

**host** – показывает к кому будет применяться playbook (host: serv1), главное чтобы имя было в файле inventory.

**command** – для выполнения определенной команды

**name** - для названия

**service** – для работы с службами пример ( service: name=httpd state=started) или

service:

name: httpd

state: started

**script** – для запуска скриптов с контроллера ansible пример (script: /etc/script.sh)

**lineinfile -** для добавление строки в файл (идемпотентен = один раз делает, если она есть не делает) пример lineinfile: path: /etc…. line: ‘Привет’

**notify**: Имя названия handlers Restart apache2

**handlers**:

* name: Restart apache2

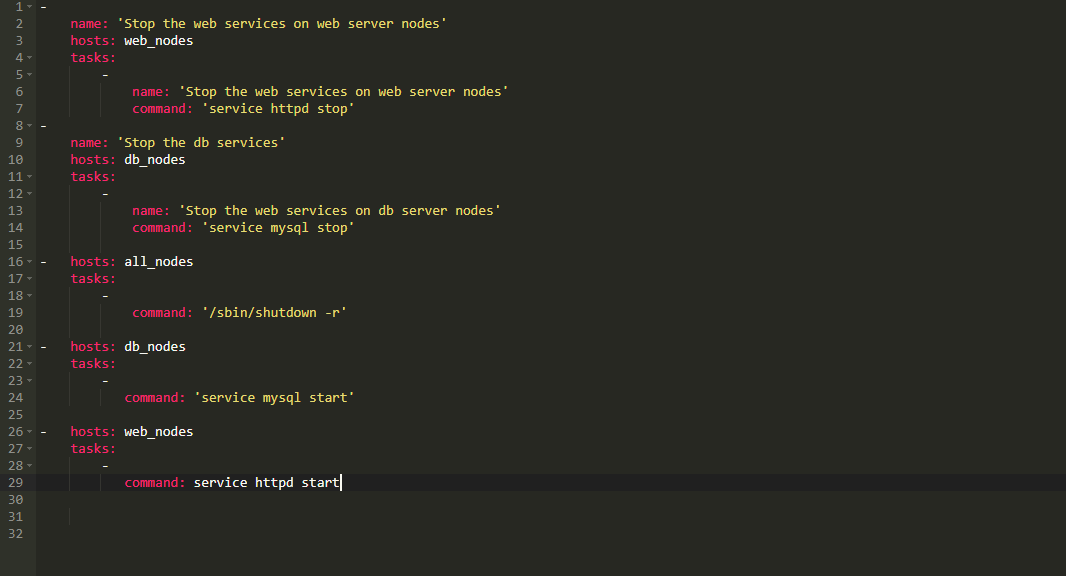
service: name=apache2 state=restarted

**debug:** имеет две переменные **var** (используем только имя переменной) и **msg** (используем {{ имя переменной }} )

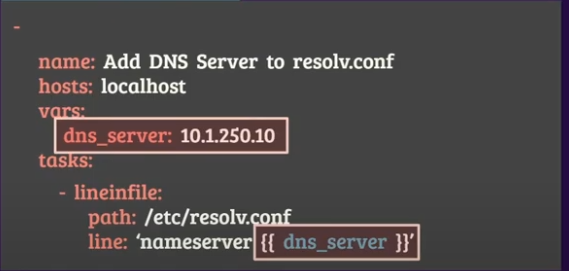
**set\_fact**: primer="{{ message }} from {{ owner }}" – для объединение нескольких переменных ( для вывода нужно использовать debug: var: primer)

**register:** сохраняет вывод предыдущей команды в переменную которую мы указываем пример: register: name

**Для запуска playbook:** ansible-playbook <имя playbook>.yaml -i inventory



**Variables**



Для создания переменных можно в inventory создать группу [Test:vars] или более профессионально создать директорию где файл inventory под названием group\_vars и там файлик с названием группы(пример: Test) в файлике начинаем с --- и переносим все переменные. Для проверки используем команду **ansible-inventory –list , Пример:**

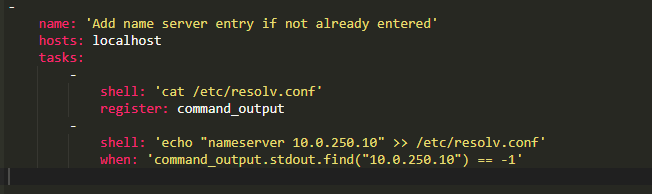
ansible\_become\_user : root

ansible\_become\_pass : 1234

**Conditionals (When)**



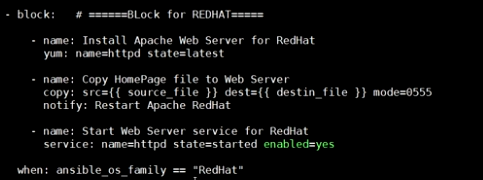
Директива **register**, которая сохраняет вывод первой команды в переменную command\_output (проверяет состояния служб и т.д.).



When: **ansible\_os\_family** == “Debian” – проверяет какая операционная система стоит если семейство Debian, то выполнит task.

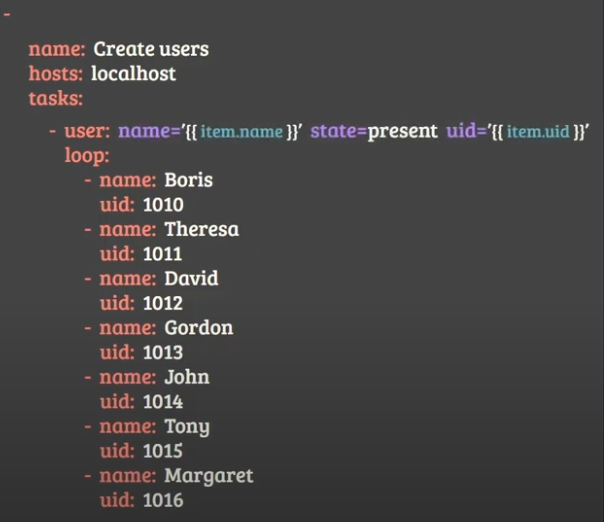
When: **ansible\_distribution\_version** == “16.04” – проверяет версию ОС.

**Использование Блоков(что часто не писать when)**:

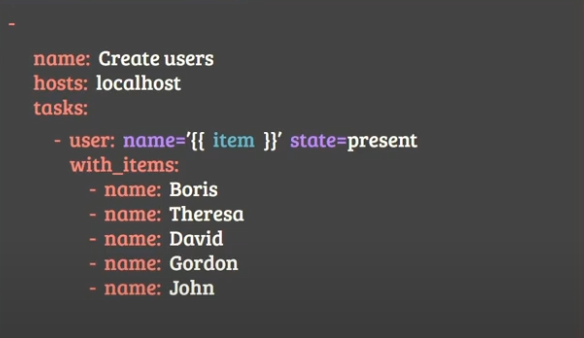


**Loops (циклы)**

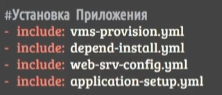
Циклы имеют переменную **item**.



Также есть **with\_items** (но это уже устаревшая форма).



**Roles**



**Include** позволяет собрать несколько playbooks в один.

**ansible-galaxy init** mysql-role – позволяет инициализировать и создать структуру каталогов.

**/etc/ansible/roles** – путь по умолчанию для ролей он находится в /etc/ansible/ansible.cfg

**ansible-galaxy install** – установить роль.

**ansible-galaxy list** - показывает установленные роли.

**ansible-config dump | grep ROLE** – показывает путь установки ролей.

**ansible-galaxy install** mysql-role **-p** ./roles - для установки роли в определенную директорию

**Ad-Hoc команды**

**ansible Linux -m setup -i inventory –** показывает информацию о серверах

**ansible Linux -m shell -a “date” -i inventory -** отправлять команды на клиенты

**ansible all -m copy -a “src=text.txt dest=/home mode=0555” -bK** - скопировать файл text.txt на клиенты в директорию home (ключ -bK – для прав sudo)

**ansible all -m file -a "path=/home/text.txt state=absent"** – удалить файлы с клиентов

**ansible all -m get\_url -a “url=https://images.app.goo.gl/i14rTewBT74keL3j8 dest=/home” -bK -** скачивание файлов на серваки

**ansible all -m apt -a "name=stress state=latest" -bK –** установка приложений на семейство Debian

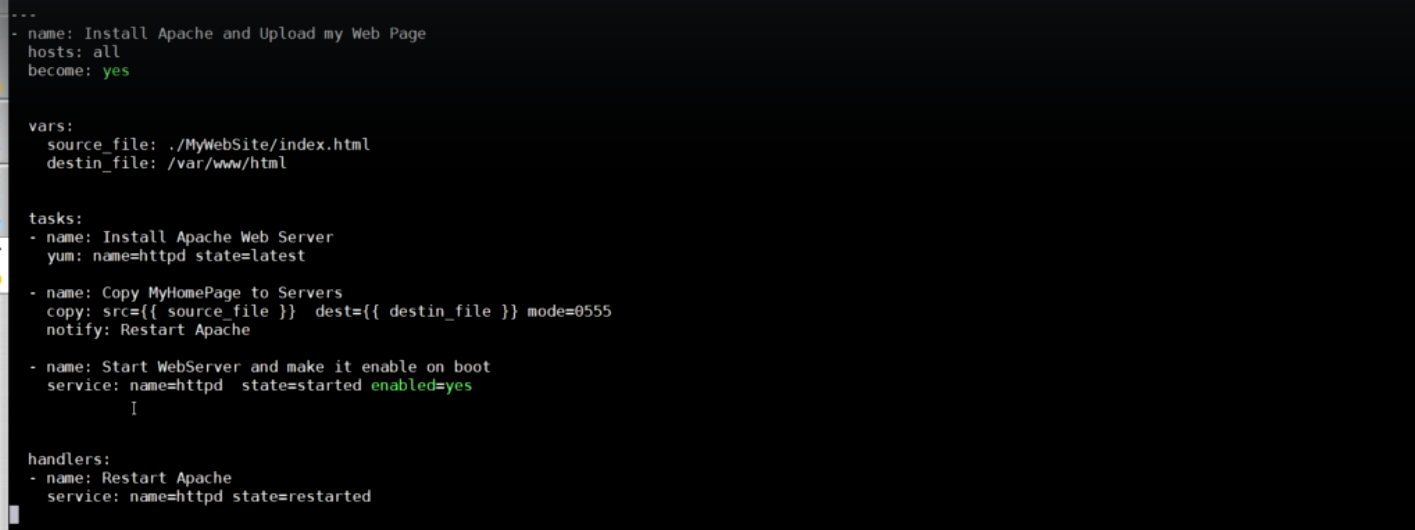
**ansible all -m apt -a "name=stress state=absent" -bK –**удаление приложений

**ansible all -m apt -a "name=apache2 state=latest" -bK -**установка службы на семейство Debian

**ansible all -m service -a "name=apache2 state=started enabled=yes" -bK –**установка чтобы после перезагрузки служба запускалась

**Ansible (Astahov)**

**/etc/ansible/ansible.cfg -** конфигурационный файл (обычно держат в директории проекта): чтобы не спрашивал finger print нужно расскомитеть **host\_key\_checking**



**Roles**

Создаем директорию **roles**, заходим и прописываем **ansible-galaxy init “название роли”**

**delegate\_to: “имя сервера”** – для запуска таска на определенной машине

**ignore\_errors: yes –** игнорирует провалившиеся таски и продолжает выполнять другие таски. (по умолчанию при провале, плейбук отключается)

**any\_errors\_fatal: true** – при любой хоть одной ошибки, плейбук мгновенное закончится.

**ansible-vault create** “имя файла” – для создания зашифрованного файла.

**ansible-vault view** “имя файла” – для просмотра зашифрованного файла.

**ansible-vault edit** “имя файла” – для редактирования зашифрованного файла.

**ansible-vault rekey** “имя файла” – для изменения пароля файла.

**ansible-vault encrypt** “имя файла” – для защифрования файла.

**ansible-vault decrypt** “имя файла” – для дешифрования файла.

**ansible-playbook** “имя файла” --ask-vault-pass – для запуска зашифрованного плейбука.