<https://docs.ansible.com/ansible/latest/index.html>

<https://pocoz.gitbooks.io/ansible_for_dev_ops_russian/>

**Подключение по SSH сертификатам**

ssh-keygen *генерация ключей*

ssh-copy-id -i public\_key\_file user@machine –p (порт) *копирование открытого ключа на сервер (к кому подкл) (путь до файла ключа потом адрес компа имяполь@айпи) должен попасть в $HOME/.ssh/authorized\_keys на ансибл указать в конфиге указать путь до закрытого ключа*

Для передачи паролей через ssh нужна пр-ма sshpass

Запускается файлом сл. содержания:

[servers]

Nextcloud ansible\_host=192.168.0.80 ansible\_ssh\_port=2222 ansible\_user=and ansible\_pass=B0ltanka

FTP ansible\_host=192.168.0.118 ansible\_ssh\_port=2222 ansible\_user=root ansible\_ssh\_private\_key\_file=/home/and/ssh\_key/ftp2.pem

Команда запуска:

Ansible –i hosts.txt all –m ping

Первый раз добавить в конец --ask-pass

Или скопировать ключ на сервер ансибл (будут хранится в /root/.ssh/autorized\_keys):

ssh-copy-id -i location of id\_rsa.pub ip-address of host

**конфигурирование ansible /etc/ansible/ansible.cfg**

host\_key\_checking = false отключить проверку отпечатка (чтоб не спрашивал при первом соед-ии)

inventory = ./hosts путь до файла с адресами серверов

теперь можно запускать ansible all –m ping (первый раз –ask-pass)

**конфигурирование hosts**

[servers\_DB]

ftp ansible\_host=192.168.0.118 ansible\_ssh\_port=2222

[servers\_AP]

test ansible\_host=192.168.0.140 ansible\_ssh\_port=22

[servers\_ALL:children] *тут объединяются в группу servers\_ALL сервера*

servers\_DB

servers\_AP

[servers\_ALL:vars] *тут настраиваются параметры общие для группы серверов*

ansible\_user = and

ansible\_ssh\_password = B0ltanka

ansible-inventory –list *команда просмотра всех групп ансибл*

**Запуск AD-Hock комманд**

-vvv *дебагинг, до 5v можно, чем больше тем больше информации*

ansible all –m setup *полная информация о сервере или группе*

ansible all –m shell –a “uptime” *(-m модуль, shell – запуск команд, -a аргумент (сама команда)) запустить команду времени на всех серверах*

ansible all -m copy -a "src=hello.txt dest=/home/and mode=777" *скопировать файл hello.txt на все сервера и дать права 777*

ansible all -m file -a "path=/home/and/hello.txt state=absent” *удалить файл с серверов (модуль file состояние state=absent)*

ansible all -m get\_url -a "url=https://ya.ru/ 961.jpg dest=/home/and" *скачать файл ин инета в папку /home/and модуль get\_url*

ansible servers\_AP -m yum -a "name=net-tools state=latest" -b *установка приложения net-tools (-b это права sudo(надо в конфиге прописать ansible\_sudo\_pass=)*

ansible servers\_AP -m yum -a "name=net-tools state=absent" –b *удалить приложение*

ansible all -m uri -a "url=https://h7n9.pro return\_content=yes" *подключиться к сайту (return\_content прочитать содержимое)*

ansible servers\_ALL -m yum -a "name=apache2 state=latest" –b *установить апач*

ansible servers\_ALL -m service -a "name=apache2 state=started enabled=yes" -b *запустить сервис апач с автозагрузкой*

**ПЕРЕНОС ПЕРЕМЕННЫХ В ПАПКУ group\_vars**

Создаем в папке /etc/ansible папку group\_vars в ней создаем файлы по названиям групп серверов и родительских групп, в эти файлы пихаем настройки типа:

---

ansible\_user : and

ansible\_ssh\_password : B0ltanka

ansible\_sudo\_pass : B0ltanka

в файле hosts остаются только сервера, все настройки выносятся в отдельные файлы

**СОЗДАНИЕ РОЛЕЙ – ROLES**

Создать папку roles в каталоге ansible

<https://docs.ansible.com/ansible/latest/user_guide/playbooks_reuse_roles.html>

ansible-galaxy init deploy\_apache\_web создание папок для ролей

потом распихиваем файлы и части плейбука по разным файлам в папках

запуск горызком плейбук файла:

---

- name: Install Apache and upload web page blyat

hosts: servers\_LO

become: yes

roles:

- { role: deploy\_apache\_web, when ansible\_system == 'Linux' } #запустить роль deploy\_apache\_web если ОС Линукс, можно просто: - deploy\_apache\_web

**ВНЕШНИЕ ПЕРЕМЕННЫЕ extra-vars**

Переменные в extra-vars имеют высший приоритет перед остальными

В плейбуке создана переменная хостов

hosts: "{{ MYHOSTS }}"

ansible-playbook play\_roles.yml --extra-var "MYHOSTS=servers\_AP"

переменная в index.j2

ansible-playbook play\_roles.yml --extra-var "MYHOSTS=servers\_AP owner=Vasya"

**ИСПОЛЬЗОВАНИЕ Import, Include**

В плейбук добавляется - include: create\_folder.yml файл с командами yml

**ПЕРЕНАПРАВЛЕНИЕ ВЫПОЛНЕНИЯ TASKS ИЗ ПЛЕЙБУК НА ОПРЕДЕЛЕННЫЙ СЕРВЕР DELEGATE\_TO**

В нужном таскс добавляется команда delegate\_to: название сервера из хостс выпольнится только для этого сервера

Выполняемая задача будет делегирована на указанный сервер

**ПЕРЕХВАТ ОШИБОК**

В tasks ставится: ignore\_errors: yes и в случае ошибки это задание проигнорируется и продолжится выполнение следующих

failed\_when: будет ошибка таск если верный аргумент этой команды

- name: второе задание

shell: echo Hello World

register: results # сохранить результат

failed\_when: results.rc == 0 # return cod из вывода, если равен нулю – ошибка

any\_errors\_fatal : true в начале файла yml после --- при любой ошибки останавливается выполение плейбук

**ХРАНЕНИЕ СЕКРЕТОВ VAULT**

ansible-vault create назв файла - создать зашифрованный файл

ansible-vault view - просмотр

ansible-vault edit - редактирование

ansible-vault rekey - изменение пароля

ansible-vault decrypt - расшифровать файл

ansible-vault encrypt назв плейбук - зашифровать готовый плейбук

ansible-playbook play.yml –ask-vault-pass - запустить зашифрованный плейбук спросив пароль

ansible-playbook ping.yml --vault-password-file pass.txt - запустить с паролем из файла pass.txt

ansible-vault encrypt\_string слово - зашифровать слово, потом его можно вставить в плейбук