Проблемы, которые решает Ansible. Область применимости

Предпосылки

- Простои ИТ-систем
- Время восстановления после сбоя
- Сложность
- Командная работа
- Antifragile

Экономика СМ

- Компании, где требуется интеграция большого числа компонентов
- Команда инженеров по эксплуатации > 10 человек
- Частые изменения = Прибыль компании
- Нет времени на документацию

CM vs Scripting

- Нет, ну а зачем этот ваш ансибл, если я все на баше уже запилил

- Ну ок, иногда работает

- повторное использование кода (да, да!)
- версионирование
- совместная работа
- base-app-service модель
- повторяемость (изменение состояния)
- общий фреймворк

- Ansible
- Chef
- Puppet
- Salt

Итак, Ansible

- версия 2.1.1.0
- готовые модули (более 600)
- готовые роли на Ansible Galaxy (~7500 ролей)
- быстрый старт
- отсутствие агента и минимальные требования к хостам
- идемпотентность (и не только)

а еще

- декларативный язык (YAML)
- поддержка Windows
- поддержка облаков

Эко-система

- Простая инсталляция
- Тестирование
- Поддержка RedHat
- Активное сообщество

Язык YAML

- Проще читать и редактировать
- Отступы для уровней вложенности
- Массивы и словари (hash, dictionary)

Пример

```
---

- hosts: ec2
  user: ubuntu
  sudo: yes
  gather_facts: no

  tasks:
    - name: Update apt cache
      apt: update_cache=yes cache_valid_time=10000

    - name: install packages
      apt:
        name: "{{ item }}"
      with_items:
        - build-essential
        - erlang
        - netdiag
```

Пример

```
---
- hosts: ec2
  user: ubuntu
  sudo: yes
  gather_facts: no

  tasks:
    - name: Update apt cache
      apt: update_cache=yes cache_valid_time=10000

    - name: install packages
      apt:
        name: "{{ item }}"
      with_items:
        - build-essential
        - erlang
        - netdiag
```

YAML начинается с “ --- ”

Каждый уровень вложенности отделяется двумя пробелами

Пример

```
---

- hosts: ec2
  user: ubuntu
  sudo: yes
  gather_facts: no

tasks:
  - name: update apt cache
    apt: update_cache=yes cache_valid_time=10000

  - name: install packages
    apt:
      name: "{{ item }}"
    with_items:
      - build-essential
      - erlang
      - netdiag
```

tasks – массив (список) элементов

Каждый элемент массива определяется “ - ”

```yaml
---
- ansible:
    templates:
      - nginx.conf
      - zabbix-agent.conf
    files:
      - ivanov.pub
      - petrov.pub
  variables:
    host: example.com
    user: admin
...
```

```yaml
pre_tasks:
  - name: "Install packages"
    apt: name={{item}} update_cache=yes
    with_items:
      - git-core
      - curl
      - libpq-dev
```

```yaml
roles:
  - role: rvm_io.rvm1-ruby
    rvm1_install_flags: '--user-install --auto-dotfiles'
    rvm1_user: '{{app_user}}'
```

http://docs.ansible.com/ansible/YAMLSyntax.html

```yaml
tasks:
  - name: Variant one
    debug: msg="test"

  - name: Variant two
    debug:
      msg: "test"

  - { name: "Variant three", debug: msg=test }
```

```yaml
templates:
- nginx.conf
- zabbix-agent.conf

templates: ["nginx.conf", "zabbix-agent.conf"]
```

Идемпотентность

```
➜  practice-1 ansible-playbook site.yml

PLAY [Install default packages] ************************************************

TASK [setup] *******************************************************************
ok: [common]

TASK [default_packages : Install GnuPG v.2] ************************************
changed: [common]

PLAY RECAP *********************************************************************
common                     : ok=2    changed=1    unreachable=0    failed=0

➜  practice-1 ansible-playbook site.yml

PLAY [Install default packages] ************************************************

TASK [setup] *******************************************************************
ok: [common]

TASK [default_packages : Install GnuPG v.2] ************************************
ok: [common]

PLAY RECAP *********************************************************************
common                     : ok=2    changed=0    unreachable=0    failed=0

➜  practice-1
```

1 запуск

2 запуск

Сущности Ansible

- модули
- плейбуки
- роли
- переменные
- хосты и группы (inventory)
- шаблоны
- теги и остальное

Модули

- Минимальные выполняемые блоки
- Идемпотентны

```
➜  practice-1 ansible all -m ping
common | SUCCESS => {
    "changed": false,
    "ping": "pong"
}
➜  practice-1
```

```
- name: restart webserver
  service: name=httpd state=restarted
```

```
- name: restart webserver
  service:
    name: httpd
    state: restarted
```

ansible all -m setup


```
"ansible_facts": {
    "ansible_all_ipv4_addresses": [
        "10.211.55.163"
    ],
    "ansible_all_ipv6_addresses": [
        "fe80::21c:42ff:fe64:9e06"
    ],
    "ansible_architecture": "x86_64",
    "ansible_bios_date": "07/15/2016",
    "ansible_bios_version": "11.2.1 (32626)",
    "ansible_cmdline": {
        "BOOT_IMAGE": "/vmlinuz-4.2.0-38-generic",
        "quiet": true,
        "ro": true,
        "root": "/dev/mapper/system-root"
    }
```

ansible all -m command -a “ls -la”

```
➜  practice-1 ansible all -m command -a "ls -la"
common | SUCCESS | rc=0 >>
total 40
drwxr-xr-x 4 vagrant vagrant 4096 Aug 10 17:04 .
drwxr-xr-x 3 root    root    4096 Jun 10 15:20 ..
-rw------- 1 vagrant vagrant   98 Aug 10 17:04 .bash_history
-rw-r--r-- 1 vagrant vagrant  220 Jun 10 15:20 .bash_logout
-rw-r--r-- 1 vagrant vagrant 3637 Jun 10 15:20 .bashrc
drwx------ 2 vagrant vagrant 4096 Jun 10 15:20 .cache
-rw-r--r-- 1 vagrant vagrant    6 Jun 10 15:20 .prlctl_version
-rw-r--r-- 1 vagrant vagrant  675 Jun 10 15:20 .profile
drwx------ 2 vagrant root    4096 Aug 10 17:04 .ssh
-rw------- 1 vagrant vagrant  669 Aug 10 17:04 .viminfo
```

Популярные модули

```
- name: PostgreSQL | Make sure the postgres data directory exists
  file:
    path: "{{postgresql_data_directory}}"
    owner: "{{ postgresql_service_user }}"
    state: directory
    mode: 0700
```

```
- name: PostgreSQL | Ensure the locale is generated | RedHat
  command: >
    localedef -c -i {{ postgresql_locale_parts[0] }} -f
{{ postgresql_locale_parts[1] }}
    {{ postgresql_locale }}

- name: PostgreSQL | Update configuration - pt. 1 (pg_hba.conf)
  template:
    src: pg_hba.conf.j2
    dest: "{{postgresql_conf_directory}}/pg_hba.conf"
    mode: 0640
```

```yaml
- name: PostgreSQL | Restart PostgreSQL
  service:
    name: "{{ postgresql_service_name }}"
    state: restarted

- name: PostgreSQL | Make sure the CA certificates are available
  apt:
    pkg: ca-certificates
    state: present

- name: PostgreSQL | Add PostgreSQL repository apt-key
  apt_key:
    id: "{{ postgresql_apt_key_id }}"
    url: "{{ postgresql_apt_key_url }}"

- name: PostgreSQL | Add PostgreSQL repository
  apt_repository:
    repo: "{{ postgresql_apt_repository }}"

- name: Create app users
  user: name={{app_user}} shell=/bin/bash
```

```yaml
- name: Set up authorized_keys for the admin user
  authorized_key: user={{app_user}} key="{{ lookup('file', item) }}"
  with_fileglob:
    - public_keys/*


- name: download tsung
  get_url:
    url: http://tsung.erlang-projects.org/dist/tsung-1.6.0.tar.gz
    dest: /home/ubuntu/tsung.tar.gz

- name: unarchive
  unarchive:
    src: /home/ubuntu/tsung.tar.gz
    dest: /home/ubuntu/
    creates: /home/ubuntu/tsung-1.6.0

- name: copy xml files
  copy:
    src: "files/{{item}}.xml"
    dest: "/home/ubuntu/{{item}}.xml"
  with_items:
    - tsung
    - tsung-app
    - tsung-stat
```

Плейбуки

- Язык плейбуков — это описание конфигурации
- Плейбук содержит “пьесы” (plays)

```yaml
- hosts: webservers
  remote_user: root
  vars:
    http_port: 80
    max_clients: 200
  tasks:
  - name: ensure apache is at the latest version
    apt:
      name: httpd
      state: latest
  - name: write the apache config file
    template:
      src: /srv/httpd.j2
      dest: /etc/httpd.conf
    notify:
    - restart apache
- name: ensure apache is running
  service:
    name: httpd
    state: started
handlers:
  - name: restart apache
    service:
      name: httpd
      state: restarted
```

Фильтр хостов, пользователь

```yaml
- hosts: webservers
  remote_user: root
  vars:
    http_port: 80
    max_clients: 200
  tasks:
  - name: ensure apache is at the latest version
    apt:
      name: httpd
      state: latest
  - name: write the apache config file
    template:
      src: /srv/httpd.j2
      dest: /etc/httpd.conf
    notify:
    - restart apache
  - name: ensure apache is running
    service:
      name: httpd
      state: started
  handlers:
    - name: restart apache
      service:
        name: httpd
        state: restarted
```

```yaml
- hosts: webservers
  remote_user: root
  vars:
    http_port: 80
    max_clients: 200
  tasks:
  - name: ensure apache is at the latest version
    apt:
      name: httpd
      state: latest
  - name: write the apache config file
    template:
      src: /srv/httpd.j2
      dest: /etc/httpd.conf
    notify:
    - restart apache
- name: ensure apache is running
  service:
    name: httpd
    state: started
handlers:
  - name: restart apache
    service:
      name: httpd
      state: restarted
```

Переменные

Задачи

```
- hosts: webservers
  remote_user: root
  vars:
    http_port: 80
    max_clients: 200
  tasks:
  - name: ensure apache is at the latest version
    apt:
      name: httpd
      state: latest
  - name: write the apache config file
    template:
      src: /srv/httpd.j2
      dest: /etc/httpd.conf
    notify:
    - restart apache
  - name: ensure apache is running
    service:
      name: httpd
      state: started
  handlers:
    - name: restart apache
      service:
        name: httpd
        state: restarted
```

```yaml
- hosts: webservers
  remote_user: root
  vars:
    http_port: 80
    max_clients: 200
  tasks:
  - name: ensure apache is at the latest version
    apt:
      name: httpd
      state: latest
  - name: write the apache config file
    template:
      src: /srv/httpd.j2
      dest: /etc/httpd.conf
    notify:
    - restart apache
  - name: ensure apache is running
    service:
      name: httpd
      state: started
handlers:
  - name: restart apache
    service:
      name: httpd
      state: restarted
```

Обработчики

```
➜  ansible git:(rails4) ✗ ansible-playbook web.yml
[DEPRECATION WARNING]: Instead of sudo/sudo_user, use become/become_user and make sure becom
e_method is 'sudo' (default).
This
feature will be removed in a future release. Deprecation warnings can be disabled by setting
 deprecation_warnings=False in
ansible.cfg.

PLAY [all] *********************************************************************

TASK [setup] *******************************************************************
ok: [52.58.58.109]

TASK [Install packages] ********************************************************
ok: [52.58.58.109] => (item=[u'git-core', u'curl', u'libpq-dev', u'ruby2.0-dev', u'libgmp-de
v', u'lsb', u'unzip', u'vim', u'postgresql-client', u'nodejs', u'imagemagick', u's3cmd'])

TASK [rvm_io.rvm1-ruby : Detect rvm binary] ************************************
ok: [52.58.58.109]

TASK [rvm_io.rvm1-ruby : Detect rvm installer] *********************************
ok: [52.58.58.109]

TASK [rvm_io.rvm1-ruby : Detect current rvm version] ***************************
ok: [52.58.58.109]

TASK [rvm_io.rvm1-ruby : Install rvm installer] ********************************
skipping: [52.58.58.109]

TASK [rvm_io.rvm1-ruby : Configure rvm installer] ******************************
skipping: [52.58.58.109]

TASK [rvm_io.rvm1-ruby : Import GPG keys] **************************************
ok: [52.58.58.109]

TASK [rvm_io.rvm1-ruby : Install rvm] ******************************************
skipping: [52.58.58.109]
```

Общие блоки

- name - название пьесы
- hosts - фильтр для хостов
- gather_facts - запускать ли модуль setup
- remote_user - пользователь для логина

```yaml
---
- name: Ideal playbook
  hosts: all
  remote_user: vagrant
  gather_facts: false
```

Общие блоки

```yaml
vars_prompt:
  - name: prompt_var
    prompt: Enter secure prompt_var
    private: yes

vars:
  just_var: "Hello, world"

vars_files:
  - "{{inventory_dir}}/play_vars/vars.yml"
```

Переменные

Блоки задач

- pre_tasks
- roles
- tasks
- post_tasks
- handlers

```yaml
pre_tasks:
  - name: debug prompt_var
    debug:
      msg: "{{prompt_var}} is prompt_var"

tasks:
  - name: echo just_var
    shell: "echo {{just_var}}"
    notify:
      - debug handler

  - name: debug just_var
    debug:
      msg: "{{just_var}} is just_var"

post_tasks:
  - name: debug file_var
    debug:
      msg: "{{file_var}} is file_var"

roles:
  - role: vars

handlers:
  - name: debug handler
    debug:
      msg: "This is a handler"
```

Все вместе

```yaml
---
- name: Ideal playbook
  hosts: all
  remote_user: vagrant
  gather_facts: false

  vars_prompt:
    - name: prompt_var
      prompt: Enter secure prompt_var
      private: yes

  vars:
    just_var: "Hello, world"

  vars_files:
    - "{{inventory_dir}}/play_vars/vars.yml"

  pre_tasks:
    - name: debug prompt_var
      debug:
        msg: "{{prompt_var}} is prompt_var"

  tasks:
    - name: echo just_var
      shell: "echo {{just_var}}"
      notify:
        - debug handler

    - name: debug just_var
      debug:
        msg: "{{just_var}} is just_var"

  post_tasks:
    - name: debug file_var
      debug:
        msg: "{{file_var}} is file_var"

  roles:
    - role: vars

  handlers:
    - name: debug handler
      debug:
        msg: "This is a handler"
```

```
➜  practice-1 ansible-playbook site.yml
Enter secure prompt_var:

PLAY [Ideal playbook] ***********************************************************

TASK [debug prompt_var] *********************************************************
ok: [common] => {
    "msg": "assdf is prompt_var"
}

TASK [vars : debug inside role] *************************************************
ok: [common] => {
    "msg": "inside the role"
}

TASK [echo just_var] ************************************************************
changed: [common]

TASK [debug just_var] ***********************************************************
ok: [common] => {
    "msg": "Hello, world is just_var"
}

RUNNING HANDLER [debug handler] *************************************************
ok: [common] => {
    "msg": "This is a handler"
}

TASK [debug file_var] ***********************************************************
ok: [common] => {
    "msg": "This variable is from file is file_var"
}

PLAY RECAP **********************************************************************
common                     : ok=6    changed=1    unreachable=0    failed=0
```

Роли

- Вначале были include

```yaml
---
- name: Include and roles
  hosts: all
  remote_user: vagrant
  gather_facts: false

  tasks:
    - include: include.yml
      vars:
        include_var: include-test


# include.yml
- name: debug inside include
  debug:
    msg: "Inside include: include_var is {{include_var}}"
```

```
➜ practice ansible-playbook roles.yml

PLAY [Include and roles] *******************************************************

TASK [debug inside include] ****************************************************
ok: [common] => {
    "msg": "Inside include: include_var is include-test"
}

PLAY RECAP *********************************************************************
common                     : ok=1    changed=0    unreachable=0    failed=0
```

Сборник пьес

```
---
- include: playbooks/common.yml
- include: playbooks/zabbix.yml
```

Роли

- Организация и переиспользование заданий
- Инкапсуляция
- Организация зависимостей

Структура роли

```
example-role
├── README.md
├── defaults
│   └── main.yml
├── files
├── handlers
│   └── main.yml
├── meta
│   └── main.yml
├── tasks
│   └── main.yml
├── templates
├── tests
│   ├── inventory
│   └── test.yml
└── vars
    └── main.yml
```

defaults/main.yml

```yaml
# Basic settings
postgresql_version: 9.3
postgresql_encoding: 'UTF-8'
postgresql_locale_parts:
  - 'en_US' # Locale
  - 'UTF-8' # Encoding
postgresql_locale: "{{ postgresql_locale_parts | join('.') }}"

postgresql_admin_user: "postgres"
postgresql_default_auth_method: "trust"

# The user/group that will run postgresql process or service
postgresql_service_user: "{{ postgresql_admin_user }}"
postgresql_service_group: "{{ postgresql_admin_user }}"

postgresql_cluster_name: "main"
```

meta/main.yml

```yaml
dependencies:
  - common

  - role: foo_app_instance
    dir: '/opt/a'
    app_port: 5000

- role: foo_app_instance
  dir: '/opt/b'
  app_port: 5001
```

Типичный плейбук

- Описывает один сервис
- Вызывает несколько ролей
- Один плейбук на сервер (кроме базового)
- Роли брать из galaxy или писать самим

```yaml
- hosts: db-slave
  user: admin
  sudo: yes

  vars_files:
    - group_vars/deploy-key.yml
    - group_vars/openvpn.yml

  roles:
    - role: debops.lvm
      lvm_volume_groups:
      - vg: 'shared'
        pvs: '/dev/vdb'
      lvm_logical_volumes:
      - lv: 'postgresql'
        vg: 'shared'
        size: '100%VG'
        mount: '/var/lib/postgresql'

- role: ANXS.postgresql
- role: Stouts.wale
  wale_aws_access_key_id: "{{aws_access_key}}"
  wale_aws_secret_access_key: "{{aws_secret_key}}"
  wale_aws_s3_prefix: "s3://project-db-backups/"

- role: mjallday.pgbouncer
  pgbouncer:
    user: postgres
    group: postgres
    config:
      listen_addr: '0.0.0.0'
      listen_port: 6432
```

galaxy

- https://galaxy.ansible.com

- ansible-galaxy - командная утилита

- 7500 ролей

GALAXY

BROWSE ROLES

Keyword oracle SORT Relevance

Keyword: oracle ✖ CLEAR ALL

POPULAR TAGS

system	3146
development	1760
web	1528
monitoring	631
database	590
networking	573
packaging	510
cloud	461
nosql	196
sql	187

oracle_java7 162

oracle-java7

Author	briancoca
Platforms	Debian, Ubuntu
Tags	system, web
Created	3/27/14 2:06 AM
Last Imported	NA

Watch 0 Star 0

oracle_jdk 6

Install Oracle JDK

Author	wybczu
Platforms	Debian, Ubuntu
Tags	development, system
Created	9/21/14 8:58 PM
Last Imported	NA

Watch 0 Star 0

oracle-java 147

Install oracle java

Author	novafloss
Platforms	Debian
Tags	system
Created	5/3/16 1:20 PM
Last Imported	5/4/16 9:05 AM

Watch 3 Star 0

oracle 23

Oracle installation/configuration role

Author	ellotheth
Platforms	EL
Tags	database, sql
Created	1/27/14 7:53 AM
Last Imported	NA

Watch 0 Star 0

Переменные

- Буквы, цифры и подчеркивание
- Начинаются с буквы
- Могут быть вложенные
- Обращаться можно как к словарю и через точку

```yaml
vars:
  hash_var:
    one: "Hello, world"
    two: "Goodby, world"

tasks:
  - name: debug hash_var
    debug:
      var: hash_var.one

  - name: debug hash_var
    debug:
      var: hash_var['two']
```

```
➜  practice ansible-playbook vars.yml

PLAY [Ideal playbook] **********************************************************

TASK [debug hash_var] **********************************************************
ok: [common] => {
    "hash_var.one": "Hello, world"
}

TASK [debug hash_var] **********************************************************
ok: [common] => {
    "hash_var['two']": "Goodby, world"
}

PLAY RECAP *********************************************************************
common                     : ok=2    changed=0    unreachable=0    failed=0
```

Где задаются?

- Файл инвертаризации
- Плейбук
- Роли и инклюды
- Факты
- Зарегистрированные переменные

Зарегистрированные переменные

```yaml
tasks:
  - name: registered vars
    command: echo "Riders on the Storm"
    register: output

  - name: debug output
    debug:
      var: output
```

```
➜ practice ansible-playbook vars.yml

PLAY [Ideal playbook] **********************

TASK [registered vars] *********************
changed: [common]

TASK [debug output] ************************
ok: [common] => {
    "output": {
        "changed": true,
        "cmd": [
            "echo",
            "Riders on the Storm"
        ],
        "delta": "0:00:00.008518",
        "end": "2016-08-12 17:37:02.721961",
        "rc": 0,
        "start": "2016-08-12 17:37:02.713443",
        "stderr": "",
        "stdout": "Riders on the Storm",
        "stdout_lines": [
            "Riders on the Storm"
        ],
        "warnings": []
    }
}

PLAY RECAP *********************************************************************
common                     : ok=2    changed=1    unreachable=0    failed=0
```

Магические переменные

- Service Discovery для бедных
- Но неплохо работает для небольшого количества хостов
- groups, hostvars, group_names

```
➜  practice ansible-playbook vars.yml

PLAY [Ideal playbook] *********************************************************

TASK [debug] ******************************************************************
ok: [common] => {
    "groups": {
        "all": [
            "common"
        ],
        "ungrouped": [
            "localhost"
        ],
        "webserver": [
            "common"
        ]
    }
}

TASK [debug] ******************************************************************
ok: [common] => {
    "hostvars": {
        "common": {
            "agent_listeninterface": "eth0",
            "ansible_check_mode": false,
            "ansible_ssh_host": "10.211.55.163",
            "ansible_version": {
                "full": "2.1.1.0",
                "major": 2,
                "minor": 1,
                "revision": 1,
                "string": "2.1.1.0"
            },
            "group_names": [
                "webserver"
            ],
            "groups": {
```

```
tasks:
  - name: debug
    debug:
      var: groups

  - name: debug
    debug:
      var: hostvars
```

Приоритеты

- role defaults
- inventory vars
- inventory group_vars
- inventory host_vars
- playbook group_vars
- playbook host_vars
- host facts
- play vars
- play vars_prompt
- play vars_files
- registered vars
- set_facts
- role and include vars
- block vars (only for tasks in block)
- task vars (only for the task)
- extra vars (always win precedence)

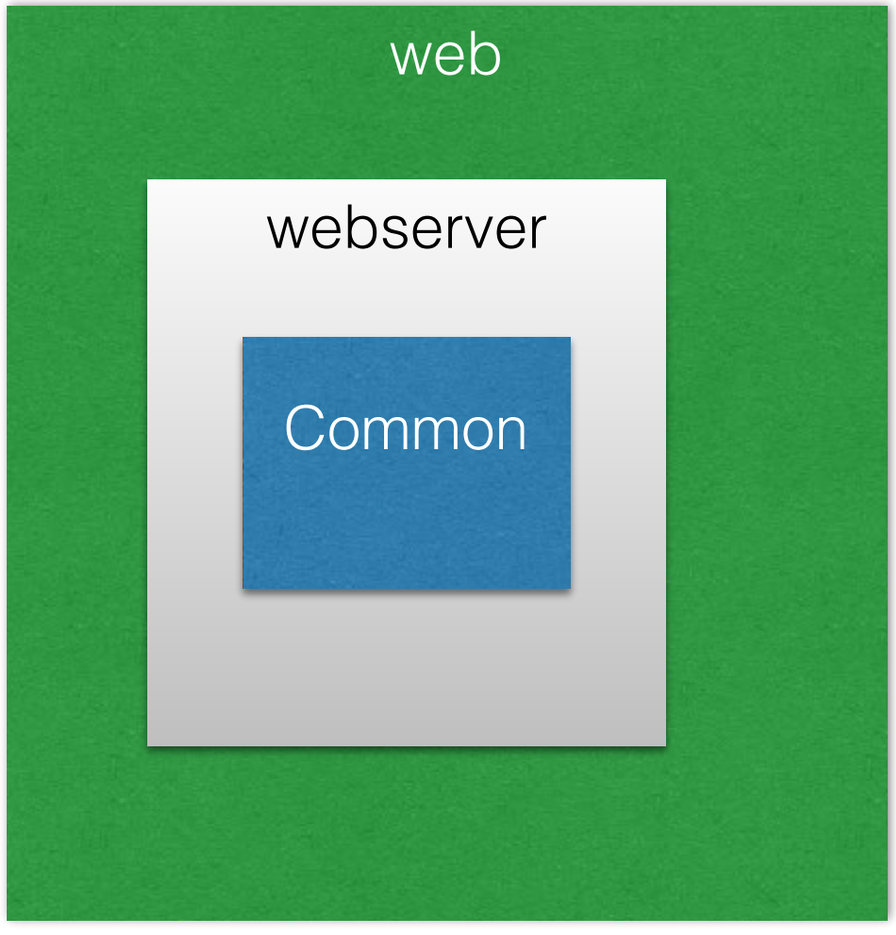
Хосты и группы

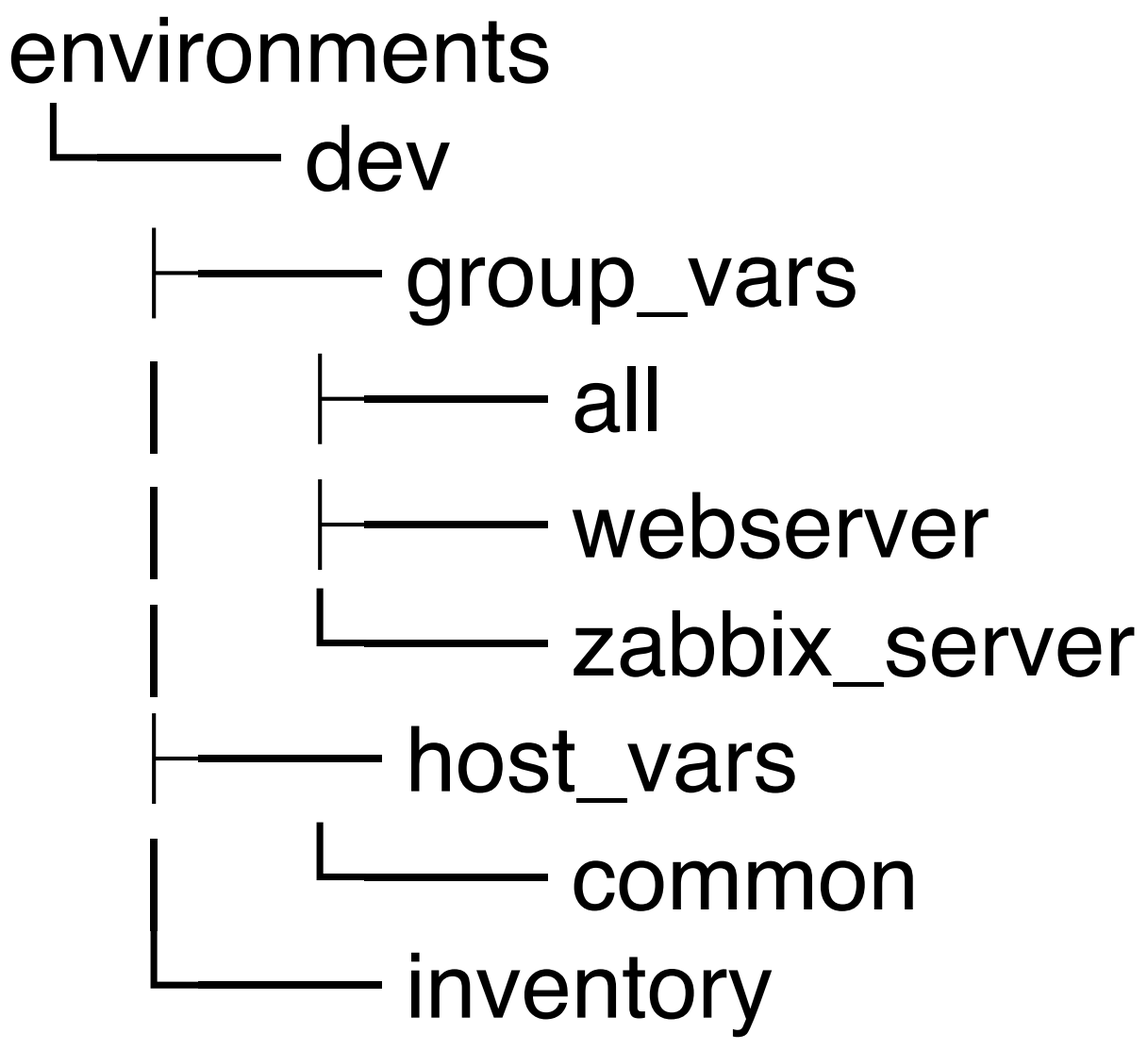
- Хост может входить в несколько групп
- Группы могут быть вложенными
- Все это описывается в файле инвентаризации
- Он может быть динамическим

```
# environments/dev/inventory
[webserver]
common ansible_ssh_host=10.211.55.163 database_port=5432

[webserver:vars]
workers_count=10

[web:children]
webserver
```




environments
dev
group_vars
all
webserver
zabbix_server
host_vars
common
inventory

Окружения

- Должны отличаться только файлы в environments
- Группы должны быть везде одинаковые
- ansible-playbook -i environments/**env_name/inventory**

Шаблоны

- Язык шаблонов jinja2
- http://jinja.pocoo.org/docs/dev/templates/
- Специальные слова в ansible
  - фильтры
  - условия
  - циклы

Фильтры

```
listen_addresses = '{{postgresql_listen_addresses|join(',')}}'
export KEY_OU="{{ openvpn_key_ou|default('MyUnit') }}"
```

```
port={{ port_list | min }}

{{ test_list | ipv4 }}
# ['192.24.2.1', '192.168.32.0/24']

{{ some_string_value | bool }}
```

Условия и циклы

```
# templates/test.j2

{% for i in list %}
   {% if i != item %}
      {{ i }}
   {% endif %}
{% endfor %}
```

```yaml
vars:
  list:
    - one
    - two
    - three

tasks:
  - name: template test
    template:
      src: test.j2
      dest: /home/{{ansible_ssh_user}}/test_{{item}}
    with_items: "{{list}}"
    register: template_changed

- name: Test conditions
  debug:
    var: template_changed
  when: template_changed.changed
```

Первый запуск

```
➜  practice ansible-playbook vars.yml

PLAY [Ideal playbook] **********************************************************

TASK [template test] ***********************************************************
changed: [common] => (item=one)
changed: [common] => (item=two)
changed: [common] => (item=three)

TASK [Test conditions] *********************************************************
ok: [common] => {
    "template_changed": {
        "changed": true,
        "msg": "All items completed",
        "results": [
            {
                "_ansible_item_result": true,
                "_ansible_no_log": false,
                "changed": true,
                "checksum": "adb38fc824bc042f1f9a872d319c76dd50f14ce0",
                "dest": "/home/vagrant/test_one",
                "gid": 1000,
                "group": "vagrant",
```

Второй запуск

```
➜  practice ansible-playbook vars.yml

PLAY [Ideal playbook] **********************************************************

TASK [template test] ***********************************************************
ok: [common] => (item=one)
ok: [common] => (item=two)
ok: [common] => (item=three)

TASK [Test conditions] *********************************************************
skipping: [common]

PLAY RECAP *********************************************************************
common                     : ok=1    changed=0    unreachable=0    failed=0
```

with_fileglob

```
- name: Create user
  user: name={{app_user}} shell=/bin/bash

- name: Set up authorized_keys for the user
  authorized_key: user={{app_user}} key="{{ lookup('file', item) }}"
  with_fileglob:
    - public_keys/*
```

with_dict

```
vars:
  mongodb_shell:
   dbname:
    - db.setProfilingLevel(1, 50)
```

```
tasks:
- name: Run mongoshell commands
  command: mongo {{ item.key }} --eval "{{ item.value|join('\n') }}"
  with_dict: mongodb_shell
```

http://docs.ansible.com/ansible/playbooks_loops.html

Остальное

- теги
- блоки

Блоки

- Способ организовывать группы заданий
- Обработка ошибок
- Принимают when, with_items и т.д.

Пример блока

```
tasks:
  - block:
      - debug: msg='i execute normally'
      - command: /bin/false
      - debug: msg='i never execute, cause ERROR!'
  rescue:
    - debug: msg='I caught an error'
    - command: /bin/false
    - debug: msg='I also never execute :-('
  always:
    - debug: msg="this always executes"
```

```
➜  practice ansible-playbook vars.yml

PLAY [Ideal playbook] *********************************************************

TASK [debug] ******************************************************************
ok: [common] => {
    "msg": "i execute normally"
}

TASK [command] ****************************************************************
fatal: [common]: FAILED! => {"changed": true, "cmd": ["/bin/false"], "delta": "0:
00:00.001401", "end": "2016-08-15 14:42:42.303641", "failed": true, "rc": 1, "sta
rt": "2016-08-15 14:42:42.302240", "stderr": "", "stdout": "", "stdout_lines": []
, "warnings": []}

TASK [debug] ******************************************************************
ok: [common] => {
    "msg": "I caught an error"
}

TASK [command] ****************************************************************
fatal: [common]: FAILED! => {"changed": true, "cmd": ["/bin/false"], "delta": "0:
00:00.001332", "end": "2016-08-15 14:42:42.574437", "failed": true, "rc": 1, "sta
rt": "2016-08-15 14:42:42.573105", "stderr": "", "stdout": "", "stdout_lines": []
, "warnings": []}

TASK [debug] ******************************************************************
ok: [common] => {
    "msg": "this always executes"
}

NO MORE HOSTS LEFT ************************************************************
        to retry, use: --limit @.ansible-retry/vars.retry

PLAY RECAP ********************************************************************
common                     : ok=3    changed=0    unreachable=0    failed=1
```

Теги

- Способ помечать задачи
- Незаменимы при разработке, т.к. позволяют запускать только то, что интересует в данный момент

```yaml
tasks:
  - name: Tags example part 1
    debug:
      msg: "Part 1"

  - name: Tags example part 2
    debug:
      msg: "Part 2"
    tags:
      - two
```

```
➜  practice ansible-playbook vars.yml -t two

PLAY [Ideal playbook] **********************************************************

TASK [Tags example part 2] *****************************************************
ok: [common] => {
    "msg": "Part 2"
}

PLAY RECAP *********************************************************************
common                     : ok=1    changed=0    unreachable=0    failed=0

➜  practice
```