

Ansible роли, окружения и best practices

План

- Анатомия Ansible плейбука
- `ansible.cfg`
- Роли и Ansible Galaxy
- Работа с окружениями
- Ansible best practices



Анатомия плейбука

- **name:** Configure hosts & deploy application
hosts: all
vars:
 mongo_bind_ip: 0.0.0.0
tasks:
 - **name:** Change mongo config file
 become: true
 template:
 src: templates/mongod.conf.j2
 dest: /etc/mongod.conf
 mode: 0644
 tags: db-tag
 notify: restart mongod
 - **name:** Add unit file for Puma
 become: true
 copy:
 src: files/puma.service
 dest: /etc/systemd/system/puma.service
 tags: app-tag
 notify: reload puma



Сценарий (play)

- Позволяет группировать части кода в рамках одного плейбука
- Плейбук состоит из одного или нескольких сценариев
- Основные секции: name, hosts, tasks.

Несколько сценариев

- **name:** Configure MongoDB
hosts: db
tags: db-tag
become: true
vars:
 - mongo_bind_ip: 0.0.0.0**tasks:**
 - **name:** Change mongo config file
template:
 - src: templates/mongod.conf.j2
 - dest: /etc/mongod.conf
 - mode: 0644

- **name:** Configure App
hosts: app
tags: app-tag
become: true
tasks:
 - **name:** Add unit file for Puma
copy:
 - src: files/puma.service
 - dest: /etc/systemd/system/puma.service

Пример инвентори файла

./hosts

[app]

appserver ansible_ssh_host=35.195.112.228

[db]

dbserver ansible_ssh_host=104.199.79.61

ansible.cfg

- конфигурационный файл для Ansible.

[defaults]

```
hostfile = ./hosts  
remote_user = appuser  
private_key_file = ~/.ssh/appuser  
host_key_checking = False
```

Более широкий список опций можно посмотреть [здесь](#).

Как организовать код?

- Один плейбук и один сценарий -> используем теги и лимиты для запуска задач
- Один плейбук много сценариев -> используем теги для запуска сценариев
- Множество плейбуков
- ...

Один сценарий

- **name:** Configure hosts & deploy application
hosts: all
vars:
 mongo_bind_ip: 0.0.0.0
tasks:
 - **name:** Change mongo config file
 become: true
 template:
 src: templates/mongod.conf.j2
 dest: /etc/mongod.conf
 mode: 0644
 tags: db-tag
 notify: restart mongod
 - **name:** Add unit file for Puma
 become: true
 copy:
 src: files/puma.service
 dest: /etc/systemd/system/puma.service
 tags: app-tag
 notify: reload puma

Один сценарий

```
$ ansible-playbook reddit_app.yml --limit app --tags app-tag
```

Хосты, на котором
запустить задачи

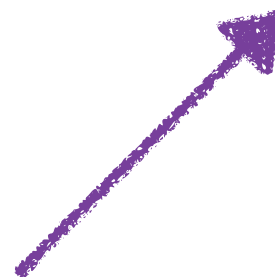
Запустить задачи,
имеющие тег

Несколько сценариев

```
---
- name: Configure MongoDB
  hosts: db
  tags: db-tag
  become: true
  vars:
    mongo_bind_ip: 0.0.0.0
  tasks:
    - name: Change mongo config file
      template:
        src: templates/mongod.conf.j2
        dest: /etc/mongod.conf
        mode: 0644

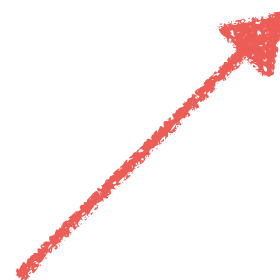
- name: Configure App
  hosts: app
  tags: app-tag
  become: true
  tasks:
    - name: Add unit file for Puma
      copy:
        src: files/puma.service
        dest: /etc/systemd/system/puma.service
```

Теги для сценария



Несколько сценариев

```
$ ansible-playbook reddit_app.yml --tags app-tag
```



Запустить указанный
сценарий

Множество плейбуков?

db.yml

app.yml

```
- name: Configure Mongo
  hosts: db
  vars:
    mongo_bind_ip: 0.0.0.0
  tasks:
    - name: Change mongo config file
      become: true
      template:
        src: templates/mongod.conf.j2
        dest: /etc/mongod.conf
        mode: 0644
      notify: restart mongod
```

Скопировать шаблон

```
- name: Configure App
  hosts: app
  become: true
  tasks:
    - name: Add unit file for Puma
      copy:
        src: files/puma.service
        dest: /etc/systemd/system/puma.se
```

Скопировать файл

Как организовать код?

- Один плейбук и один сценарий -> используем теги для запуска задач
- Один плейбук много сценариев -> используем теги для запуска сценариев
- Множество плейбуков
- **Роли**

Роли

- Механизм группировки и переиспользования кода
- Включают: tasks, handlers, variables, tests, templates, files
- Роль составляется под задачу. Например, конфигурация сервиса или настройка базового слоя.
- Community roles

Пример

```
$ tree db
```

```
db
```

```
├── README.md
├── defaults
│   └── main.yml
├── files
├── handlers
│   └── main.yml
├── meta
│   └── main.yml
├── tasks
│   └── main.yml
├── templates
│   └── mongod.conf.j2
├── tests
│   ├── inventory
│   └── test.yml
└── vars
    └── main.yml
```

Директория для
переменных по
умолчанию

Информация о роли и
зависимостях

Директория для
тестов

Директория переменных, с
большим приоритетом, чем
по умолчанию

```
8 directories, 9 files
```


Ansible Galaxy

- Централизованное место поиска комьюнити ролей
- Встроенная команда ansible-galaxy

Использование комьюнити роли

```
$ ansible-galaxy install --roles-path _ geerlingguy.apache
```

- downloading role 'apache', owned by geerlingguy
- downloading role from <https://github.com/geerlingguy/ansible-role-apache/archive/2.1.1.tar.gz>
- extracting geerlingguy.apache to ./geerlingguy.apache
- geerlingguy.apache (2.1.1) was installed successfully

apache.yml

- hosts: appserver
become: true

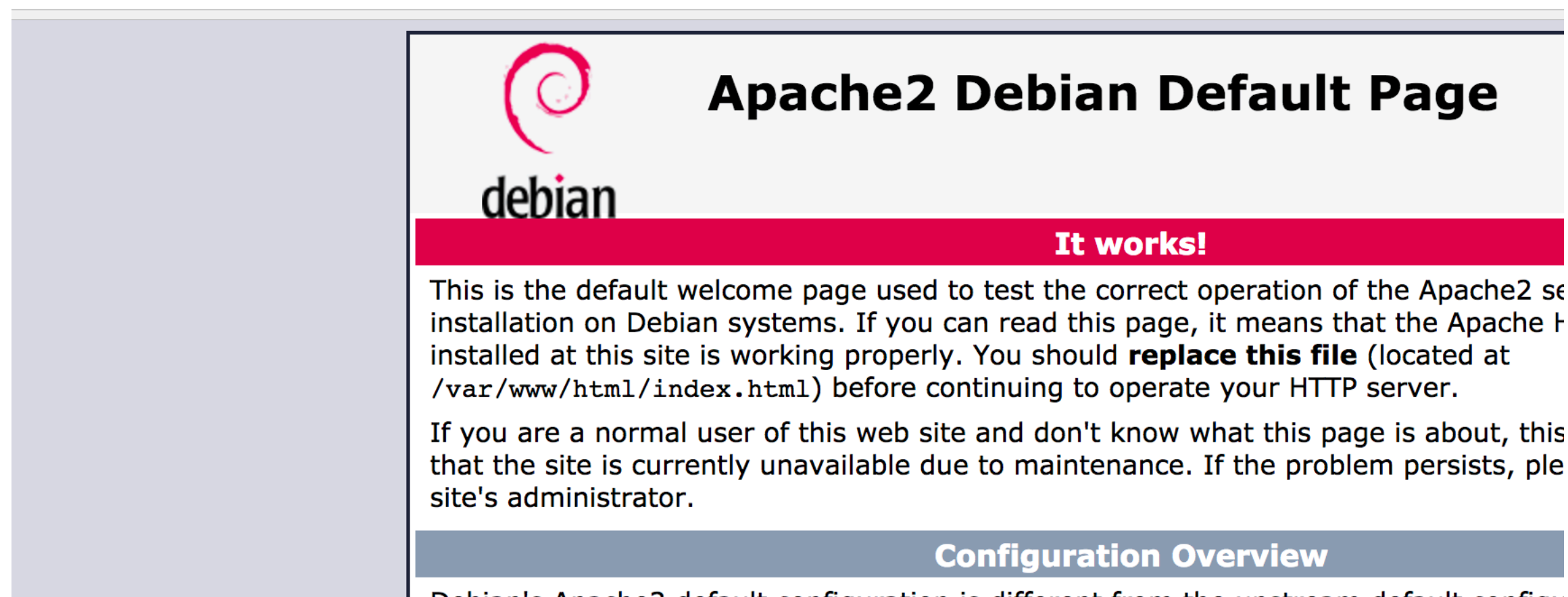
roles:

- geerlingguy.apache



```
$ ansible-playbook apache.yml
```

① 35.195.112.228



ansible-galaxy init

- Создает заготовку роли в соответствии с форматом Galaxy
- Не создает директорий templates и files

Структура директории

Пример структуры директории с учетом ролей

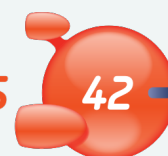
```
app.yml  
db.yml  
deploy.yml  
hosts  
ansible.cfg  
roles/  
    app/  
    db/
```

site.yml

- Основной плейбук
- Описывает настройку всей инфраструктуры
- Запускает почти все остальные плейбуки

Пример site.yml

- include: db.yml
- include: app.yml
- include: deploy.yml



Структура директории

```
app.yml  
db.yml  
deploy.yml  
hosts  
ansible.cfg  
site.yml  
roles/  
    app/  
    db/
```


Вопросы управления окружениями

- Инвентори файл
- Определение специфических настроек окружения



Инвентори на окружение

Пример организации директории:

```
environments/  
  production/  
    hosts  
  stage/  
    hosts
```

```
app.yml  
db.yml  
deploy.yml  
ansible.cfg  
roles/  
  app/  
  db/
```



group_vars и host_vars

- Позволяют определять переменные для групп хостов и отдельных хостов
- Используются для управления настройками окружения

Пример структуры директории с учетом окружений

```
environments/  
  production/  
    hosts  
    group_vars/  
      app  
      web  
    host_vars/  
      appserver2  
  stage/  
    hosts  
    group_vars/  
      app  
      db  
    host_vars/  
      appserver1  
site.yml  
app.yml  
db.yml  
deploy.yml  
ansible.cfg  
roles/  
  app/  
  db/
```

Пример настройки окружения

```
$ ansible-playbook -i environments/prod/hosts site.yml --check  
$ ansible-playbook -i environments/prod/hosts site.yml
```

```
$ ansible-playbook -i environments/stage/hosts site.yml --check  
$ ansible-playbook -i environments/stage/hosts site.yml
```

Ansible best practices

- Советы по организации структуры репозитория
- Управление окружениями
- Советы по написанию кода
- Ссылка
- Линтеры (ansible-lint)