СЕТЕВАЯ
АКАДЕМИЯ
ЛАНИТ

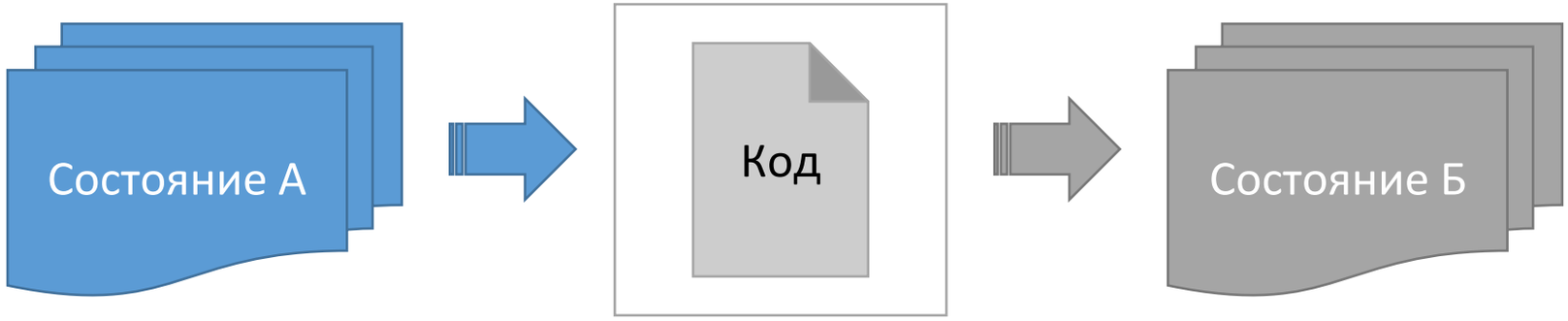
Управление конфигурациями

Ansible

ПЛАН

- Что такое и какие бывают системы управления конфигурациями
- Ansible
- Работа с секретами в Ansible Vault

Конфигурация?



- System Configuration Management, SCM

Для чего это нужно?

- Любая системная автоматизация
- Управление изменениями
  - Изменения в промышленной среде
- Provisioning
  - Настройка серверов с нуля, облака
- Оркестрация
  - Фреймворк для масштабного развертывания и автоматизации

История автоматизации SCM

1. Bash / batch-скрипты
2. Python/PERL/Ruby
3. PowerShell
4. Salt Stack
5. Puppet
6. Chef
7. Ansible
8. Terraform

- Легковесный и расширяемый
- исполнение из коробки
- Но нельзя писать на Ruby

Ansible

- Нет агентов
  - Через которые надо ходить на целевые машины
  - SSH, WinRM, API
- Нет баз данных
  - YAML, INI, TXT
- Нет сложной настройки
  - просто библиотека Python
- Нет лишнего ПО
  - Послал пакет
  - Выполнил
  - Вернул результат

YAML

- Без программирования (декларативный DSL)
- Структурированный
- Простой для восприятия человеком

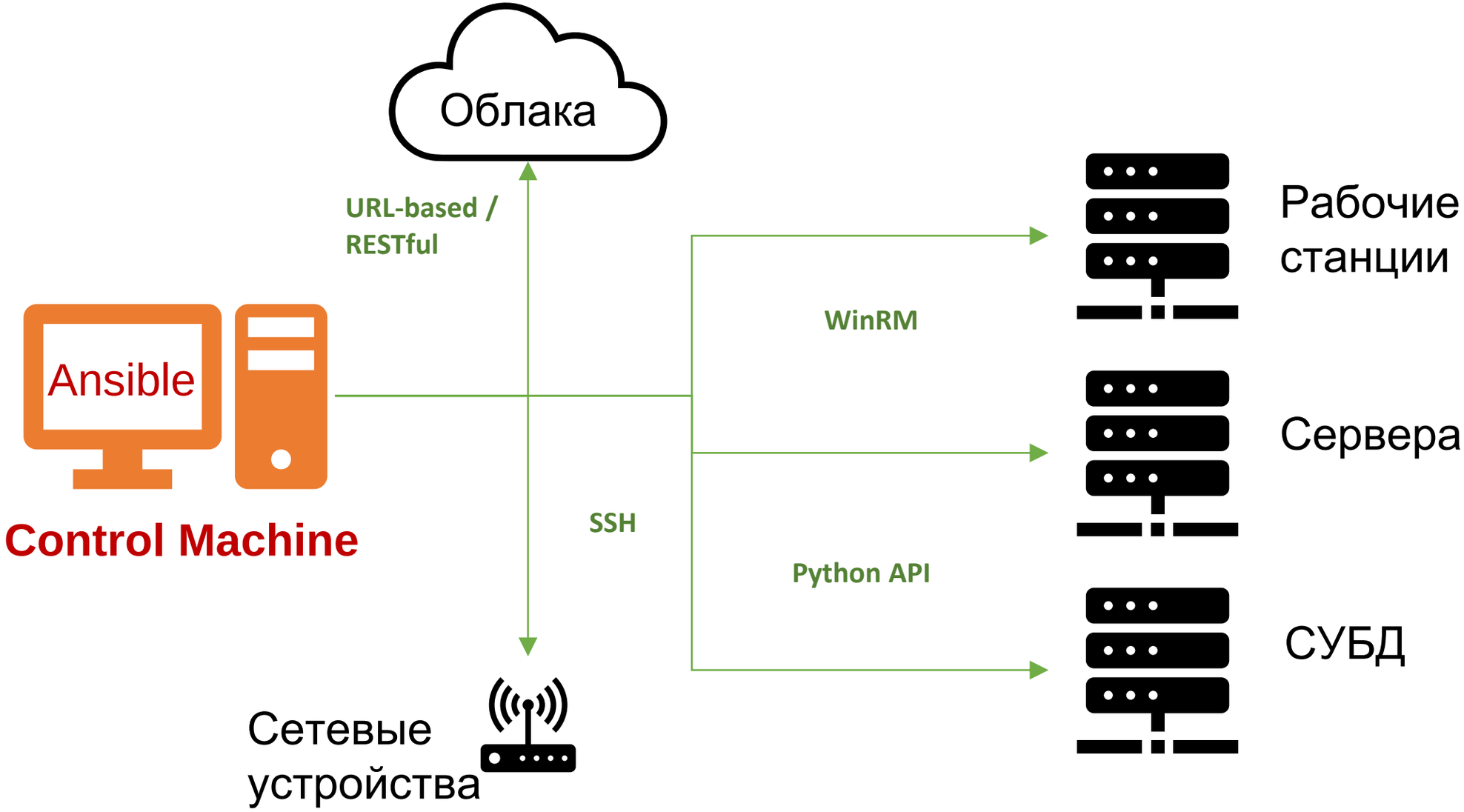
- Отступы!

```yaml
%YAML 1.2
---
YAML: YAML Ain't Markup Language™

What It Is:
  YAML is a human-friendly data serialization
  language for all programming languages.

YAML Resources:
  YAML Specifications:
  - YAML 1.2:
    - Revision 1.2.2        # Oct 1, 2021 *New*
    - Revision 1.2.1        # Oct 1, 2009
    - Revision 1.2.0        # Jul 21, 2009
  - YAML 1.1
  - YAML 1.0

  YAML on GitHub:           # github.com/yaml/
    YAML Specs:         yaml-spec/
    YAML 1.2 Grammar: yaml-grammar/
    YAML Test Suite:  yaml-test-suite/
    YAML Issues:        issues/

  YAML Test Matrix:    matrix.yaml.io
...
```

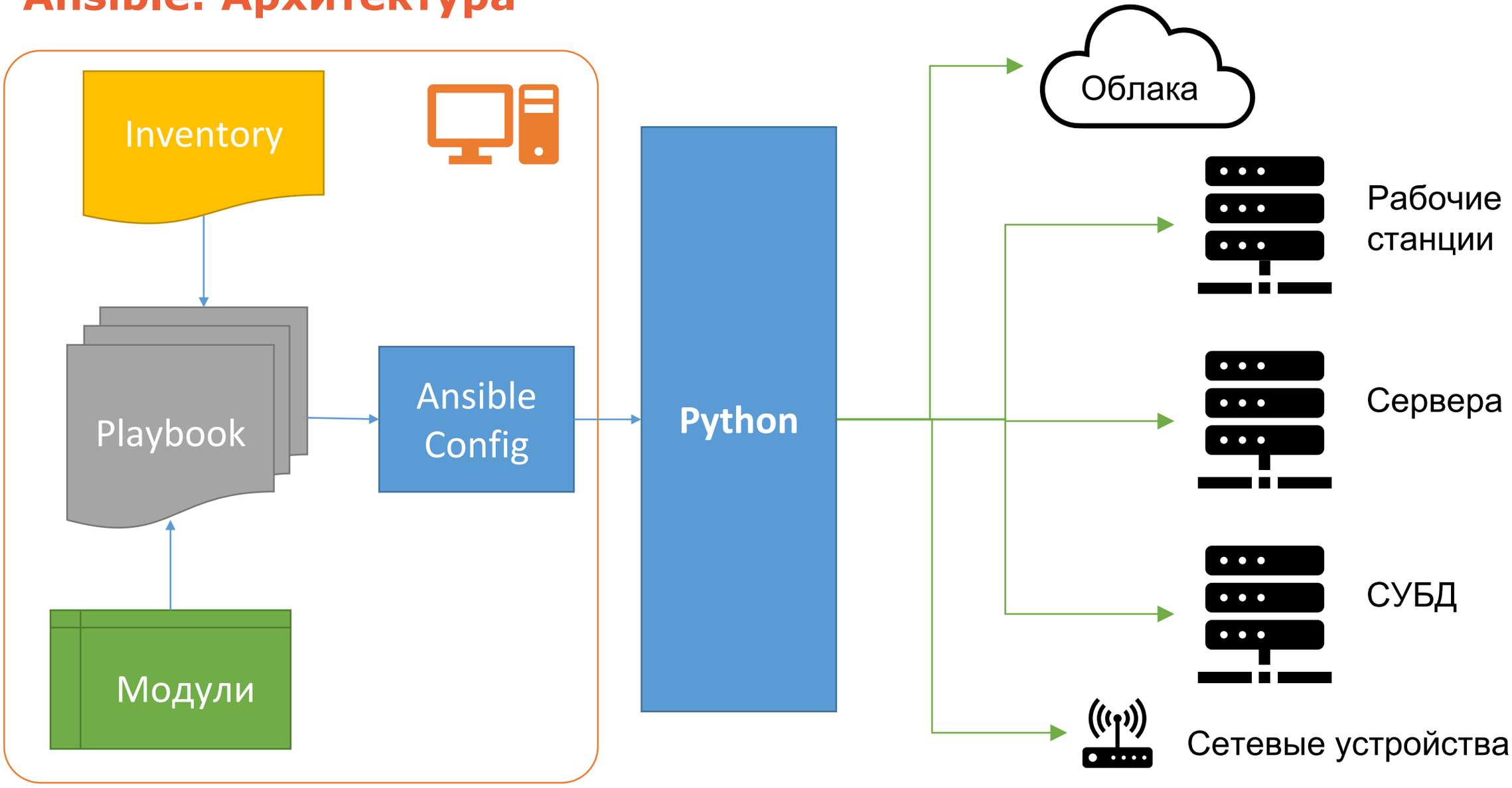
API

- URL-based / RESTful
- Утилиты командной строки
- Скриптинг на Python

Ansible: Соединения



Ansible: Архитектура



Красота в простоте

Идемпотентность

Идемпотентность – свойство операции при повторном применении к объекту давать тот же результат, что был при первом

Википедия

Запуск приводит систему в желаемое состояние, повторный запуск делает то же самое только при необходимости

Идемпотентность в Ansible

Установка Ansible

Распространяется через *Personal Package Archive (PPA)*

RedHat / CentOS

```
$ sudo yum install ansible
```

Fedora

```
$ sudo dnf install ansible
```

Debian / Ubuntu

```
$ sudo apt-get install ansible
```

PIP

```
$ sudo pip install ansible
```

Дополнительные опции:

- *Собрать из исходников*
- *Сделать свой RPM*

Playbook - пьеса, схема сборки или книга рецептов

YAML

```
- hosts: websrvgrp
  tasks:
    - name: Install Apache web server
      yum:
        name: httpd
        state: latest
    - name: Deploy Config
      copy:
        src: file/httpd.conf
        dest: /etc/httpd.conf
- hosts: databasesrvgrp
  tasks:
    - name: Install PostgreSQL
      yum:
        name: postgresql
        state: latest
```

Ansible: Конфигурация

Порядок поиска значений

1. `ANSIBLE_CONFIG` (и переменные окружения, если установлены)
2. `ansible.cfg` (в текущей папке)
3. `~/.ansible.cfg` (в домашней папке)
4. `/etc/ansible/ansible.cfg`

Ansible: Переменные

Playbook

```
- hosts: websrvgrp
  vars:
    http_port: 80
    sqluser: admin
```

Inventory

```
inventory
group_vars/all
group_vars/groupname
host_vars/hostname
```

Роли

Подстановка переменных
из файлов роли
в playboook

Факты и модуль *setup*

`ansible_os_family`	Семейство ОС: RedHat, Debian, …
`ansible_processor_cores`	Количество ядер ЦПУ
`ansible_kernel`	Версия ядра
`ansible_devices`	Информация о подключенных устройствах
`ansible_default_ipv4`	IP-адрес, MAC, шлюз, …
`ansible_architecture`	64/32 бит

Сохранить вывод: модуль *register*

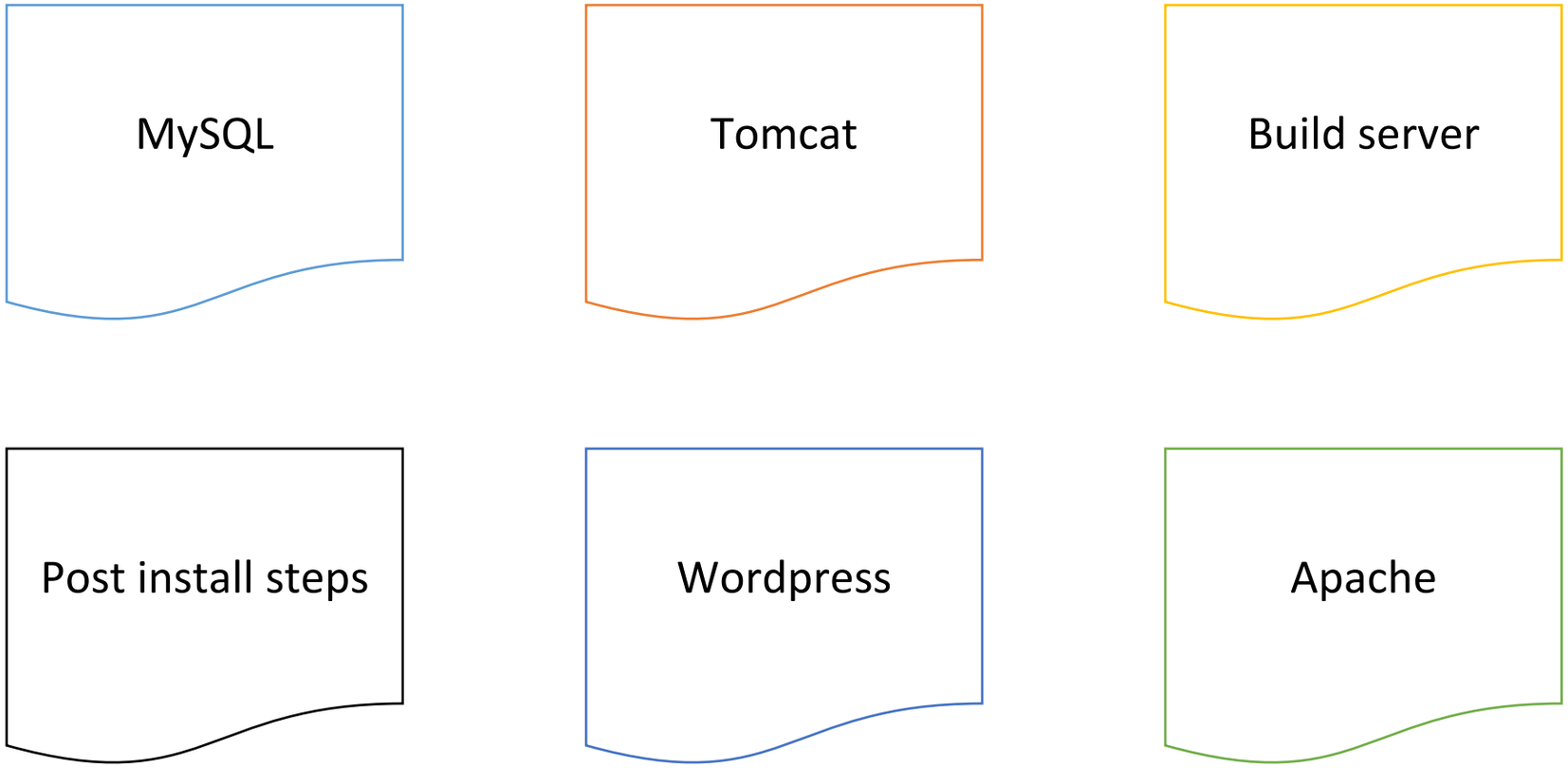
Playbook
tasks

Формат
JSON

В переменной

В tasks
(debug)

Роли


MySQL
Tomcat
Build server
Post install steps
Wordpress
Apache

Роли: структура папок

Содержимое Playbook

1. Глобальные определения
2. Переменные
3. Задачи
4. Файлы
5. Шаблоны
6. Обработчики

```
roles
  |_some_role
        |_README.md
        |_files
        |    |_...
        |_templates
        |    |_...
        |_tasks
        |    |_ main.yml
        |    |_ ...
        |_handlers
        |    |_ main.yml
        |    |_ ...
        |_vars
        |    |_ main.yml
        |    |_ ...
        |_meta
            |_main.yml
            |_ ...
```

Ansible и shell

Best Practices — Ansible Documentation

- Ansible != shell
  - Казалось бы, обычный SSH,
  - Но Ansible делает для нас гораздо больше.
  - Код собирается в модули на каждый случай жизни.
    Не надо изобретать велосипед.
- Использование ignore_errors
  - Сценарии с кодом возврата <>0
    - Модуль register + failed_when:
  - Состояние гонки
  - Избыточные действия
- shell = changed
  - creates: и removes:
  - changed_when: false
  - Обработка вывода и кода возврата

```
- name: First attempt
  shell: exit 0
  register: first_attempt
  failed_when: false
  changed_when: first_attempt.rc == 0

- name: Second attempt
  shell: exit 0
  when: not first_attempt.changed
```

```
- name: First attempt
  shell: exit 124
  register: first_attempt
  failed_when: false
  changed_when: first_attempt.rc == 0

- name: Second attempt
  shell: exit 0
  when: not first_attempt.changed
```

Секрет

информация, которая хранится в зашифрованных файлах
и расшифровывается в нужный момент.

Эволюция прикладной криптографии

1. Симметричные ключи шифрования
2. Асимметричное шифрование с парой ключей — публичным и приватным
3. HSM (Модули аппаратной безопасности)
4. Облачный KMS (сервис управления ключами)

Ansible Vault

- Криптографическая защита для любого файла данных
  - `group_vars/`
  - `host_vars/`
  - `inventory`
  - `include_vars/vars_files`
- для одиночных переменных
  - Тег `!vault`
- для бинарников и др.

Утилита `ansible-vault`

Создать зашифрованный файл	`$ ansible-vault create users.yml`
Зашифровать существующий файл	`$ ansible-vault encrypt data.csv`
Расшифровать файл	`$ ansible-vault decrypt data.csv`
Сменить пароль шифрования	`$ ansible-vault rekey data.csv`
Отредактировать зашифрованный файл	`$ ansible-vault edit users.yml`
Посмотреть зашифрованный файл	`$ ansible-vault view users.yml`

Использование в ansible

Передать пароль в Playbook

```
$ ansible-playbook --ask-vault-pass <encrypted-playbook>.yml
```

```
$ ansible-playbook <encrypted-playbook>.yml --vault-id @prompt
```

Шифрование переменных

```
$ nano vars/vault.yml
vault_db_pass: MyStrongPassword

$ ansible-vault encrypt vars/vault.yml

$ nano vars/main.yml
db_user: admin
db_port: 3306
db_pass: "{{ vault_db_pass }}"
```

```
$ echo -n 'MyStrongPassword' \
 | ansible-vault encrypt_string \
  --vault-id dev@~/.vault_pass \
  --stdin-name  'vault_db_pass'

$ nano playbook.yml
vault_db_pass: !vault |
  $ANSIBLE_VAULT;1.2;AES256;dev
  386535643063323665613630626436326968…
```

Файл паролей

Создать файл пароля

```
$ echo 'MyStrongVaulPassword' > .ansible_vault_pass
```

Добавить файл пароля в .gitignore

```
$ echo '.ansible_vault_pass' >> .gitignore
```

Использовать файл пароля при запуске ansible или ansible-playbook

```
$ ansible --vault-password-file=.ansible_vault_pass ...
$ ansible-playbook --vault-password-file=.ansible_vault_pass ...
```

Настроить файл пароля по-умолчанию для среды

```
$ export ANSIBLE_VAULT_PASSWORD_FILE=./.ansible_vault_pass
```

А также в файле конфигурации ansible.cfg

В заключение

- Привыкайте к декларативной модели
- Используйте модули
- Обрабатывайте ошибки
- Будьте внимательны к окружению

- Чистота кода

44 совета по Ansible: рекомендации и Best Practices / Хабр (habr.com)


СЕТЕВАЯ
АКАДЕМИЯ
ЛАНИТ
Больше, чем обучение!
www.academy.ru