

# Д303. Ansible

## Дедлайны

- **Мягкий deadline:** 10.12, 23:59
- **Жёсткий deadline:** 17.12, 23:59

## Подробное описание

Необходимо с помощью Ansible реализовать следующую автоматизацию на виртуальной машине на одном из дистрибутивов Linux:

1. Установить пакеты `nginx`, `cron`, `jq`
2. Изменить конфигурацию `nginx` таким образом, чтобы по запросу **GET** `/service_data` отдавалось содержимое файла `/opt/service_state.json`
3. Создать файл `/opt/service_state.json` по следующему шаблону:

```
{
    "title": "Seems working",
    "uptime": 0
}
```

4. Обеспечить запуск `nginx`
5. Добавить в `cron` выполнение раз в минуту команды (одной строкой):

```
echo "$(jq ".uptime = (($($ps -o etimes= -p $(cat /var/run/nginx.pid)) / 60))" /opt/service_state.json)" > /opt/service_state.json
```

**Замечание:** для автоматизации работы с `cron` можно пользоваться `python-crontab`.

6. Выполнить проверку того, что значение поля `uptime` в файле `/opt/service_state.json` начало изменяться.

**Внимание** - конфигурация `ansible` должна быть идемпотентной, т.е. соответствовать следующим требованиям:

- Повторный запуск `ansible` с той же конфигурацией **не должен сбрасывать** значение `uptime` в файле `/opt/service_state.json` и **не должен рестартовать** `nginx`.
- После изменения поля `title` **в шаблоне** в файле `/opt/service_state.json` (например, на "Seems work ok") должно происходить обновление файла `/opt/service_state.json` и рестарт сервиса `nginx`.

# Нюансы для потоков

## Базовый поток:

- ваше решение должны быть выполнено в виде **playbook**-а с именем `playbook.yml` в корне репозитория
- ваше решение должно работать для **Ubuntu 22.04**

## Продвинутый поток:

- ваше решение должно быть выполнено в виде `ansible`-роли и файла `playbook.yml` в корне репозитория, в котором импортируется созданная вами роль
- ваше решение должно работать для **Ubuntu 22.04, Centos 7 и Arch Linux**

**Совет** - для тестирования вашего решения вами использовать Vagrant и provisioning силами Ansible, чтобы проверять ваше решение на чистых виртуалках. Подойдут следующие box-ы:

- `ubuntu/jammy64`
- `centos/7`
- `archlinux/archlinux`

## Формат проверки

Запускается чистая виртуалка с нужной ОС и для нее будем запускать ваш `playbook.yml`.

### Проверки:

1. Запускаем `playbook.yml`. Запрос `GET /service_data` с помощью `curl` на порт 80, должен отдать:

```
{
  "title": "Seems working",
  "uptime": 0
}
```
2. Делаем паузу на X минут. Снова запрос `GET /service_data` с помощью `curl` на порт 80, должен отдать:

```
{
  "title": "Seems working",
  "uptime": X
}
```

3. Еще раз запускаем `playbook.yml`. Снова запрос `GET /service_data` с помощью `curl` на порт 80, должен отдать:

```
{
    "title": "Seems working",
    "uptime": X
}
```
4. Изменяем первую строку в шаблоне и еще раз запускаем `playbook.yml`. Снова запрос `GET /service_data` с помощью `curl` на порт 80, должен отдать:

```
{
    "title": "[Измененная первая строка]",
    "uptime": 0
}
```
5. Делаем пауза `Y` минут. Снова запрос `GET /service_data` с помощью `curl` на порт 80, должен отдать:

```
{
    "title": "[Измененная первая строка]",
    "uptime": Y
}
```

## Штрафы

### Базовый поток:

- [-20%] - за невыполнение каждого из пунктов

### Продвинутый поток

- [-20%] - за невыполнение каждого из пунктов хотя бы на одной из требуемых ОС

## Формат сдачи

Аналогично Д31 и Д32.

Репозитории будут доступны 27го ноября.