

01 Oncall

план семинара [↗](#)

1. Что такое Oncall
2. Установка Oncall
 1. Установка MySQL
 2. Установка Oncall
3. Базовая конфигурация Oncall
 1. Создание пользователей
 2. Создание команд
 3. Создание календаря для команд
4. API oncall
 1. Получим команды как пример

Что такое Oncall

На первом семинаре мы будем разворачивать приложение Oncall и последующие практики SRE будут имплементироваться на нем.

Oncall - это opensource решение для построения календаря дежурств от компании LinkedIn.

Например в компании Тинькофф используется Oncall для построения календаря дежурств.

Примеры использования:

- Календарь дежурств (ведет SRE команда)
- Автоматизация для эскалации дежурного (Page)
- Ручная эскалация дежурного (Support)

Установка Oncall

Oncall хранит данные в базе данных MySQL. Ее мы сначала и установим.

Установка Python

```
apt-get update
```

```
apt-get install -y libsasl2-dev python3-dev libldap2-dev libssl-dev python3-pip python-setuptools git
```

Установка MySQL

```
apt-get install -y mysql-server mysql-client
```

Установка Oncall

Скачиваем(клонировем) исходный код oncall

```
cd /opt
git clone https://github.com/linkedin/oncall.git
cd oncall
```

Устанавливаем Oncall

```
python3 setup.py develop
pip3 install -e '.[dev]'
```

Запускаем MySQL и накатываем миграции на базу данных

```
service mysql start

mysql -u root < ./db/schema.v0.sql
echo "ALTER USER 'root'@'localhost' IDENTIFIED WITH mysql_native_password BY '1234'; FLUSH PRIVILEGES;" | mysql -u root
```

Запускаем oncall как сервис linux

```
cat << EOF > /lib/systemd/system/oncall.service

[Unit]
Description=Oncall service

[Service]
ExecStart=/usr/local/bin/oncall-dev /opt/oncall/configs/config.yaml
```

```
[Install]
WantedBy=multi-user.target
```

EOF

```
systemctl daemon-reload
systemctl enable oncall.service
systemctl start oncall.service
```

Если открыть в браузере :8080 мы должны увидеть веб-интерфейс приложения Oncall

Базовая конфигурация Oncall

Создание пользователей

Создание пользователей в Oncall доступно через отдельную интеграцию, которой у нас нет, либо через API, мы воспользуемся способом через API

Создадим пользователя Admin, Sergey Petrov и Vasilyi Ivanov

```
curl -O -v -X POST http://localhost:8080/api/v0/users \
-H "Expect:" \
-H 'Content-Type: application/json; charset=utf-8' \
--data-binary @- << EOF
{
  "name": "s.petrov"
}
EOF
```

Обновим данные по нему, для этого сделаем Rest запрос

```
curl -O -v -X PUT http://localhost:8080/api/v0/users/s.petrov \
-H "Expect:" \
-H 'Content-Type: application/json; charset=utf-8' \
--data-binary @- << EOF
{
  "contacts": {
    "call": "+7 111-111-1111",
    "email": "s.petrov@student.com",
    "slack": "s.petrov",
    "sms": "+7 111-111-1111"
  },
  "full_name": "Sergey Petrov",
  "photo_url": null,
  "time_zone": "Europe/Moscow",
  "active": 1
}
```

```
}  
EOF
```

Авторизация

Чтобы пользоваться веб-интерфейсом, необходимо в правом в верхнем углу авторизоваться. Логин <Имя пользователя> Пароль <Имя пользователя>

Создание команд

1. Создадим команду Team 1 через веб-интерфейс. Нажимаем Create Team (в сплывающем окне достаточно вписать название команды)
2. Добавим список "On Duty" в команду Team 1.
 1. Заходим во вкладку Team Info
 2. Нажимаем Add Roster - это группа наших дежурных, назовем например "On Duty@"
3. Добавим дежурных в список "On Duty"
 1. В группу "On Duty" добавим наших дежурных по имени пользователя

Создание календаря для команд

Дежурных обычно 2 (Primary, Secondary):

- Primary - основной дежурный
- Secondary - второй дежурный, приходит на помощь первому или заменяет в его отсутствие

На практике обычно Secondary стает Primary для передачи полного контекста предыдущего дня(какие есть проблемы, какие есть и были сбои)

Распишем календарь дежурств на текущий месяц для команды Team 1, дежурные будут дежурить неделя через неделю чередуя друг друга

API oncall

Oncall имеет API, через которое можно выполнять все действия, которые можно выполнять через веб-интерфейс

Получим список команд

```
curl -X GET http://localhost:8080/api/v0/teams
```

Запрос в API должен вернуть список команд, которые мы создавали до этого

Домашние задание

Задание

1. Установить Oncall
2. Создать тестовых пользователей, больше 3
3. Завести команды 2-3 команды, больше 1
4. Заполнить расписание дежурств на 2 месяца вперед, есть 2 варианта заполнения
 1. Через веб-интерфейс
 2. Через API
5. В 1 день должно быть 2 дежурных (Primary, Secondary). Secondary текущего дня, становится Primary на следующий день

Варианты

1. Пересменка каждые 2 дня в 8 часов утра
2. Пересменка каждые 2 дня в 10 часов утра
3. Пересменка каждые 3 дня в 8 часов утра
4. Пересменка каждые 3 дня в 10 часов утра
5. Пересменка каждые 4 дня в 8 часов утра
6. Пересменка каждые 4 дня в 10 часов утра
7. Пересменка каждые 5 дней в 8 часов утра
8. Пересменка каждые 5 дней в 12 часов дня

Критерии оценки

1. Установлен Oncall - 5 баллов
2. Заполнение календаря через веб-интерфейс - 8 баллов
3. Заполнение календаря через api - 10 баллов

Список литературы

- <https://github.com/linkedin/oncall>
- https://oncall.tools/docs/user_guide.html#
- <https://dev.mysql.com/doc/mysql-apt-repo-quick-guide/en/>
- <https://ubuntu.com/server/docs>
- <https://linux-packages.com/ubuntu-focal-fossa/package/systemctl>

