

# СЛЁРМ: SRE





# Знакомство с системой

- Наше приложение: веб-агрегатор билетов в кино;
- Единый веб фронтенд;
- Распределенная система кинотеатров-бэкендов
  - Каждая команда отвечает за свой “кинотеатр” (provider backend).

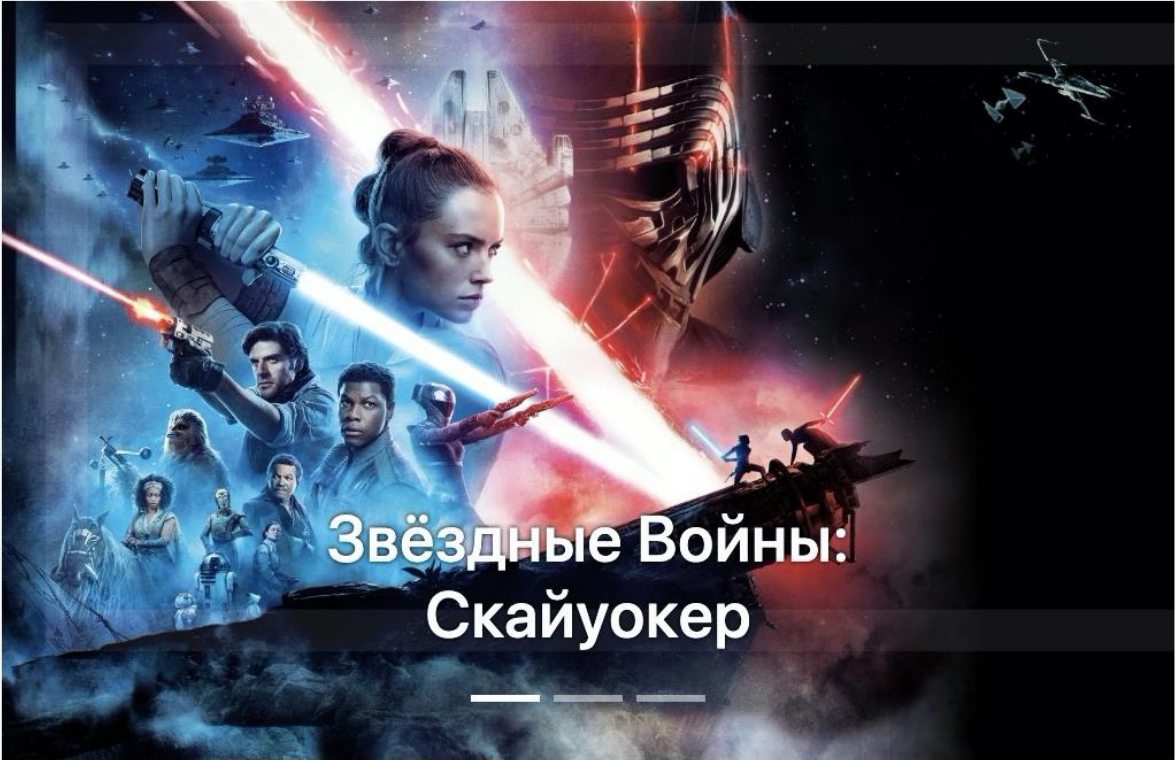


# Юзер-флоу

<https://tickets.sre.slurm.io/>

СЛЁРМ

Вход Регистрация



Звёздные Войны:  
Скайуокер

Фильмы

Moscow ▾



# Юзер-флоу

Мы наконец-то идём в кино: <https://tickets.sre.slurm.io/>

Здесь можно посмотреть, как работают наши сервисы со стороны конечных пользователей.

Есть меню выбора города в правой части под большими постерами. Сначала выберите город, потом фильм (у нас самый широкий выбор фильмов!), далее дату и время. Попробуйте выкупить места в зрительном зале.

При нагрузке на систему забронировать места вручную не получится — это нормально, кто-то их уже выкупил до вас.

Лучше всего, конечно же, общаться с API провайдера напрямую (через curl например).

Далее на картинках показана схема работы компонентов системы.

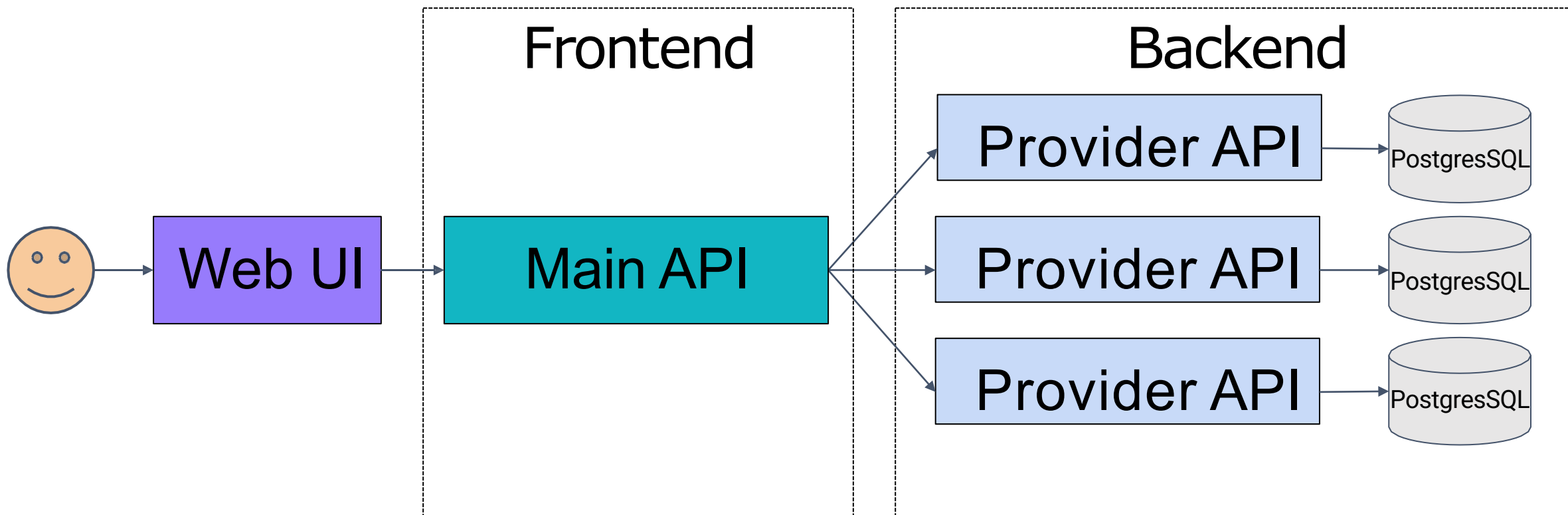


- СЛЁРМ**

[illegible]



# Структура системы





# Система одного кинотеатра

Компоненты:

- ReST API сервер, приложение на Python
  - Описание API: в каталоге docs/ в репозитории
- СУБД: PostgreSQL





# Система одного кинотеатра

Каждая команда отвечает за работоспособность своего сервиса:

- включая запуск, обновления, и т. д.
- у каждой команды свой репозиторий с кодом.



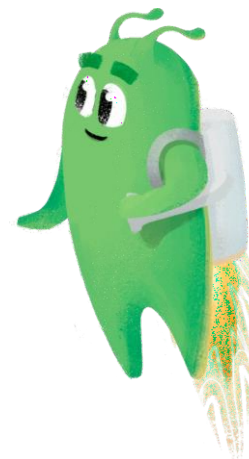


# Окружение



# Рабочая станция

- \$ ssh sbbox.slurm.io
- Login/пароль: <https://edu.slurm.io/>
- На этом хосте уже есть рабочий минимум:
  - kubectl
  - git
  - vim





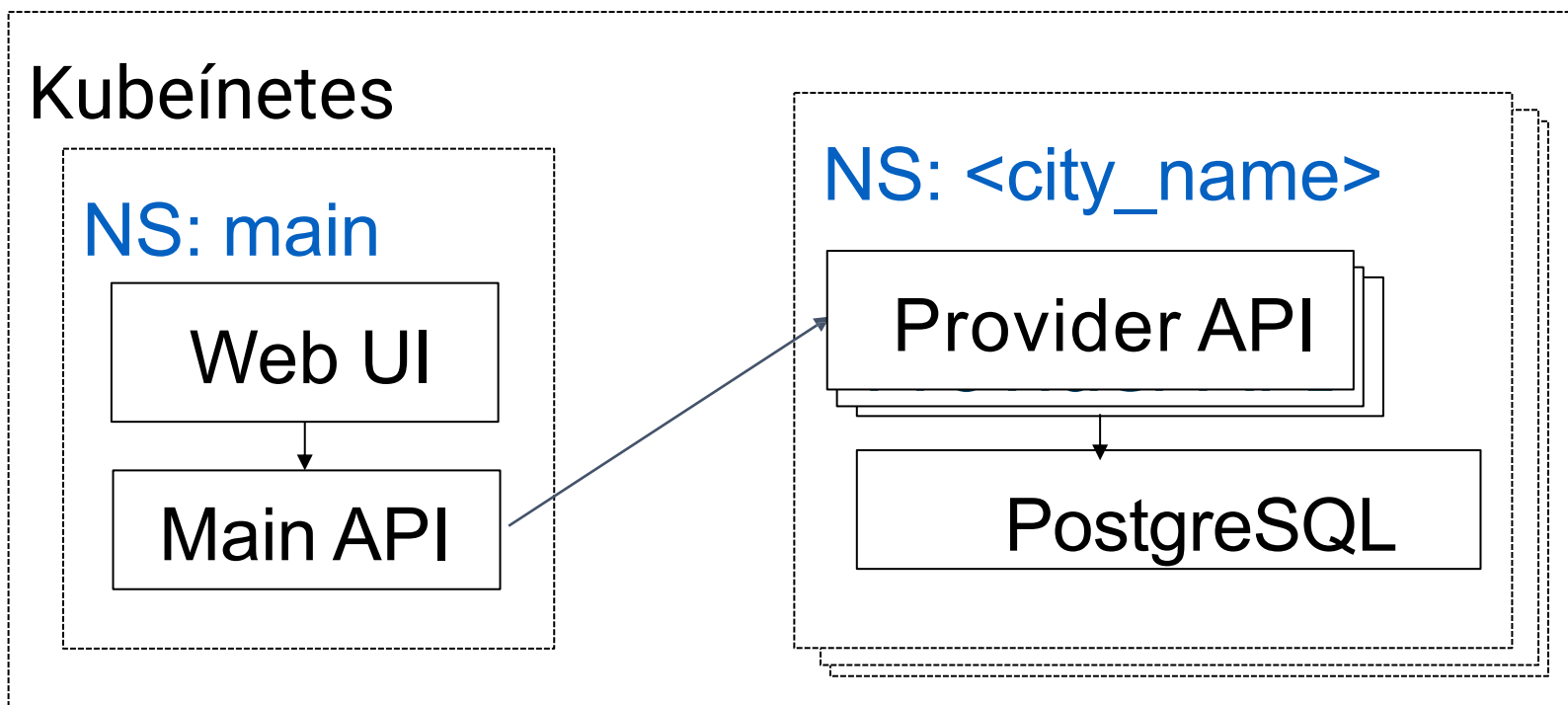
# Кластер

- Общий Kubernetes кластер;
- У каждой команды свой Kubernetes namespace;
- Компоненты в кластере общаются по сети;
- Чаще всего это HTTP ReST API.

Пожалуйста, не надо ломать наш кластер, это вредно для кармы. К тому же курс не про пентестинг, а про SRE.



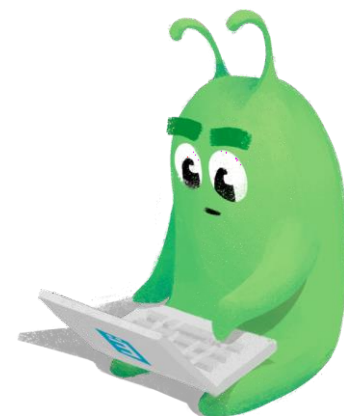
# Кластер





# Сервисы в кластере

- `$ kubectl config set-context \`  
`--current --namespace=<city_name>`
- `$ kubectl get pods`  
`<city_name> x5`  
`<city_name>-postgresql-primary x1`  
`<city_name>-postgresql-read x3`
- `$ kubectl get ingress`  
`provider-backend.<city_name>.sre.slurm.io`





# Исходники и CI/CD



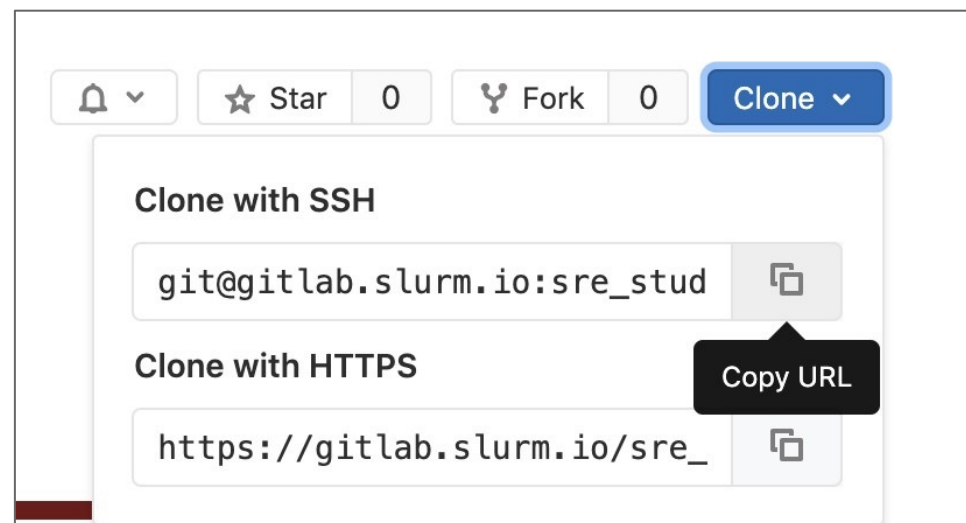
# Исходники

- [https://gitlab.slurm.io/sre\\_students](https://gitlab.slurm.io/sre_students)
- Убедитесь, что к ним есть доступ:

\$ git clone

[git@gitlab.slurm.io](https://gitlab.slurm.io):sre\_students/<city\_name>/provider\_backend.git

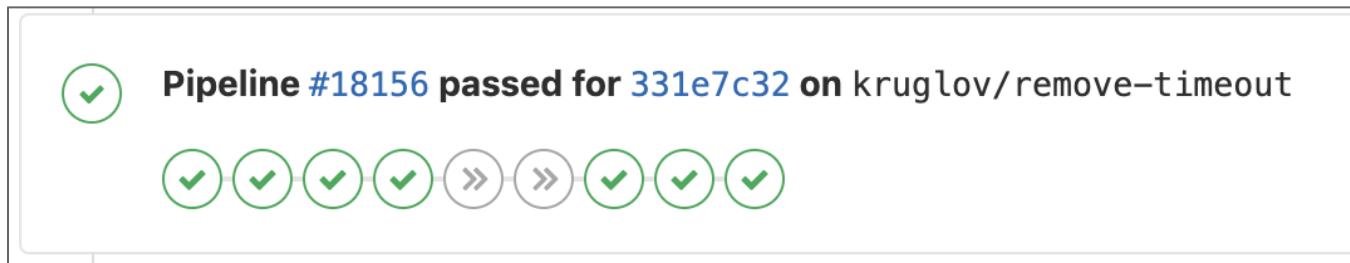
- Настройка SSH ключей в GitLab: <https://gitlab.slurm.io/profile/keys>





# Внесение изменений

- `git checkout -b <feature branch>`
- `<hack-hack-hack>`
- `git add && git commit`
- `git push`
- Create Merge Request
  - Это создаст ветку в GitLab и запустит тесты
- Если тесты прошли успешно → мержим и изменения выкатываются в кластер автоматически.







# Исходники, внесение изменений, CI/CD

Исходный код проекта для ознакомления заранее находится вот тут: [https://gitlab.slurm.io/sre-4/provider\\_backend](https://gitlab.slurm.io/sre-4/provider_backend)

Текстовые материалы курса находятся здесь: <https://gitlab.slurm.io/sre-4/sre>

На sbx уже настроены ключи для Gitlab, туда можно автоматически клонировать код и работать с ним там.

Никто не запрещает добавить в вашу учетку Gitlab ваш собственный публичный ключ тут: <https://gitlab.slurm.io/profile/keys> и клонировать репозиторий себе, чтобы работать в любимом IDE.

Можно также подключить себе sbx по SSH, например к VS Code.

Описание пайплайна – файл [.gitlab-ci.yml](#) в корне репозитория.



## Шпаргалка по API

- Подробнее см. каталог docs/
- GET /movies
- GET /movies/:id/seances
- POST /movies {..}
- POST /movies/:id/seances {..}
- POST /movies/:id/seances/:id/bookings {..}