Как получить на Google Cloud Platform бесплатный сервер Nodejs на год и разместить на нём MERN-приложение.

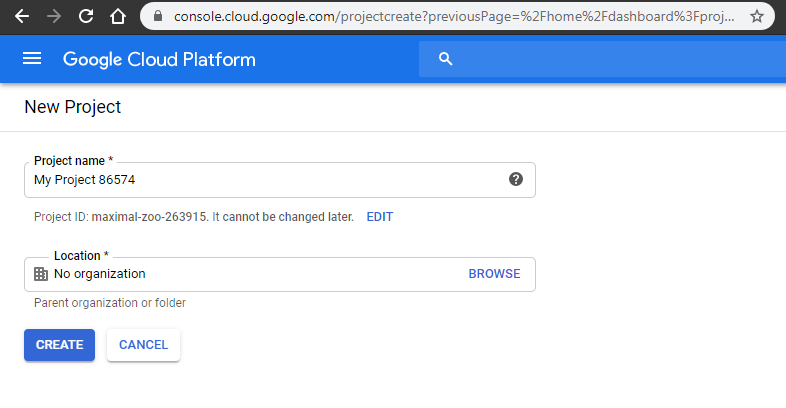
**Tags: nodejs, react.js, google app engine, google cloud platform**

В этой статье хочу описать мой опыт взаимодействия с Google Cloud Platform и Google App Engine.

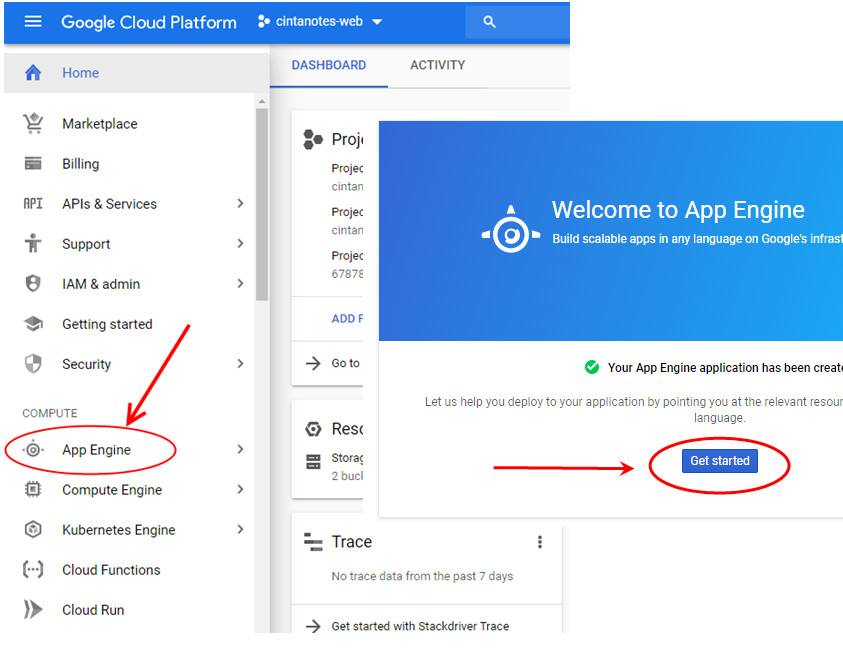
Plan такой

* Создание проекта, заливка файлов
* Подготовка к билду, настройка varEnv
* Собственно deploy

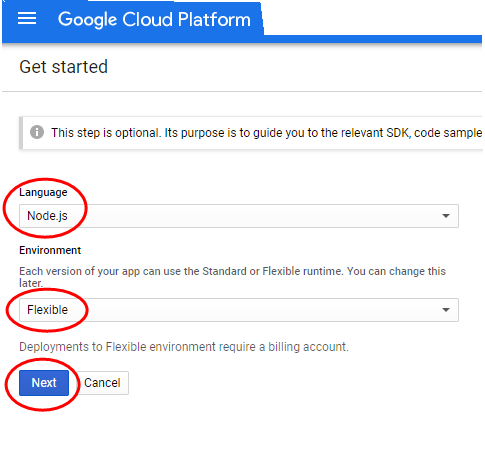
**Создаём новый проект. Заливаем файлы**



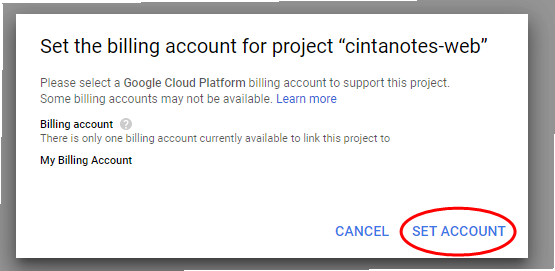
Заходим в App Engine, жмем Get started



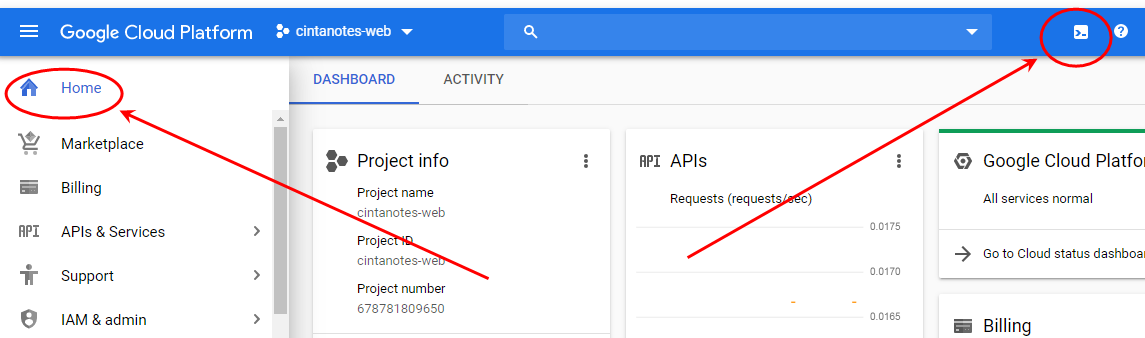
Выбираем ЯП сервера, тип формирования для environment и жамкаем next



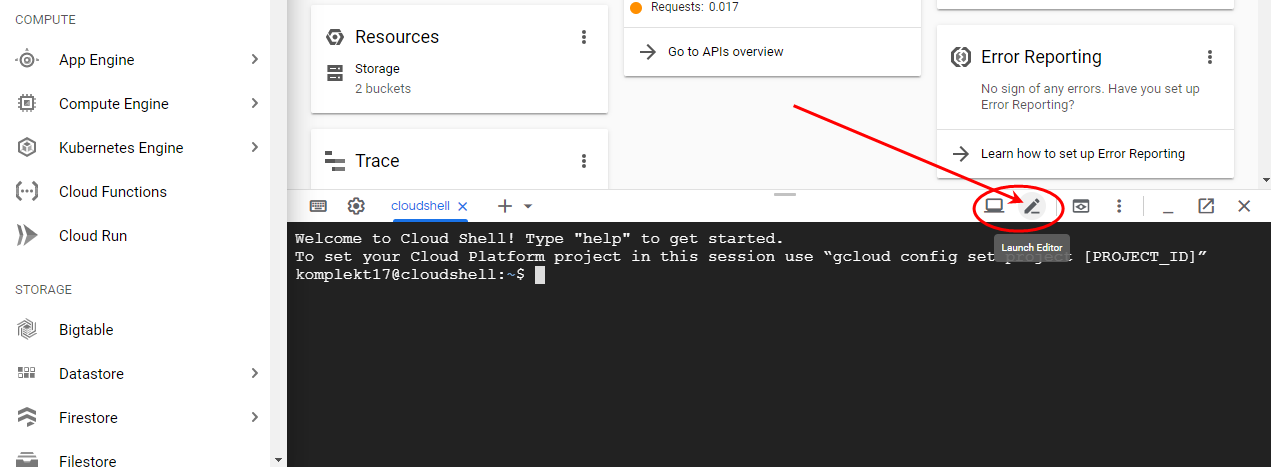
жамкаем Set Account и готово



Переходим на страницу Home, активируем Cloud Shell и переходим в редактор



Переходим в редактор

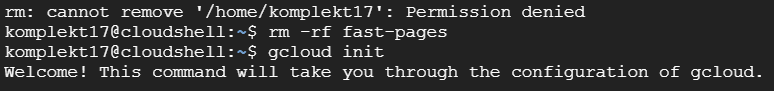


Очищаем редактор от файлов предыдущего проекта, если он был (у меня был).

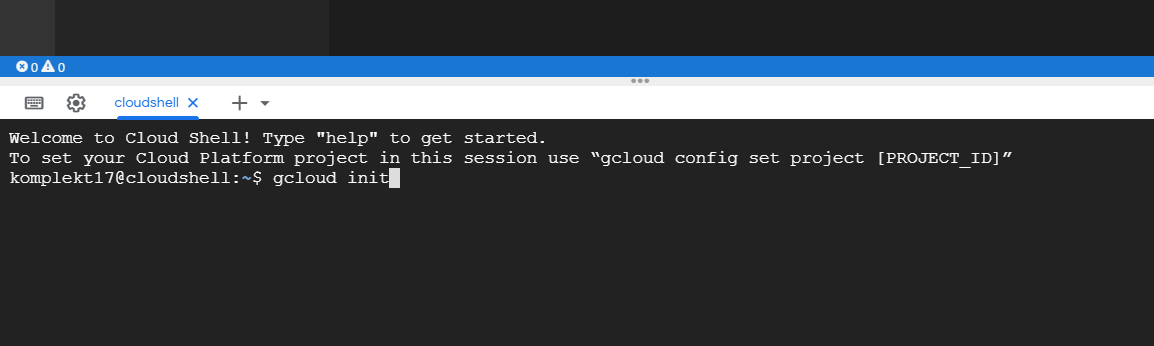
( для справки , как сделать тут. Как сбросить постоянный диск пользователя Google Cloud Shell?)

<https://stackoverflow.com/questions/36067968/how-to-reset-google-cloud-shell-user-persistent-disk>

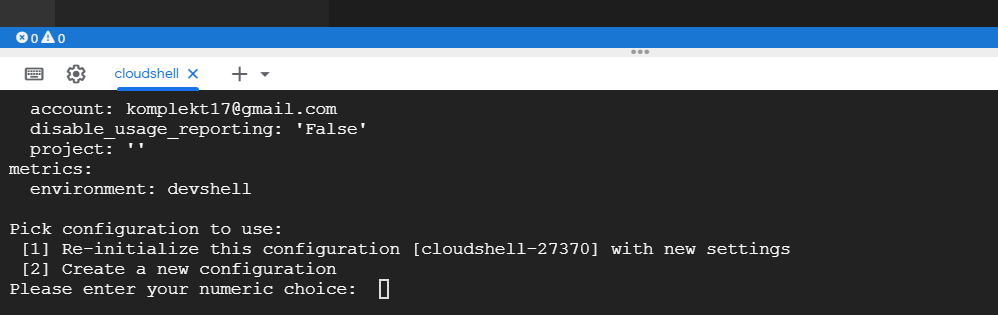
Доки по теме: <https://cloud.google.com/shell/docs/resetting-cloud-shell>



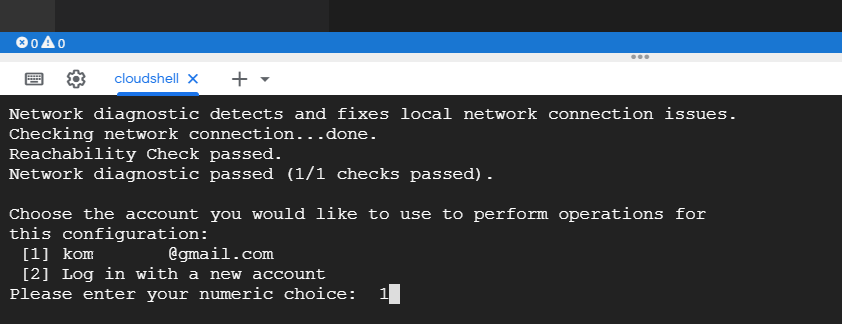
Если очищать не нужно, инициализируем новый проект gcloud init

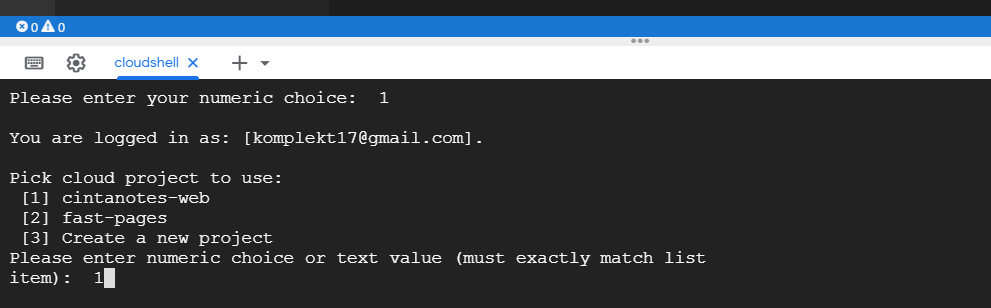


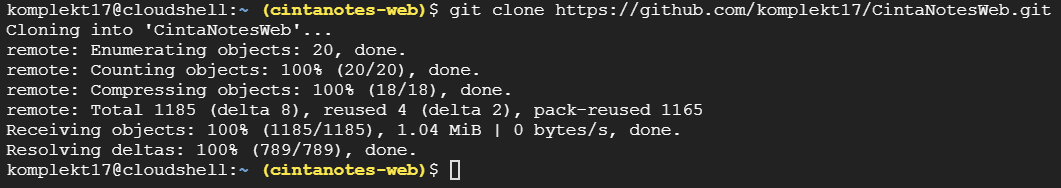
Выбираем [1] – Re-initialize this configuration



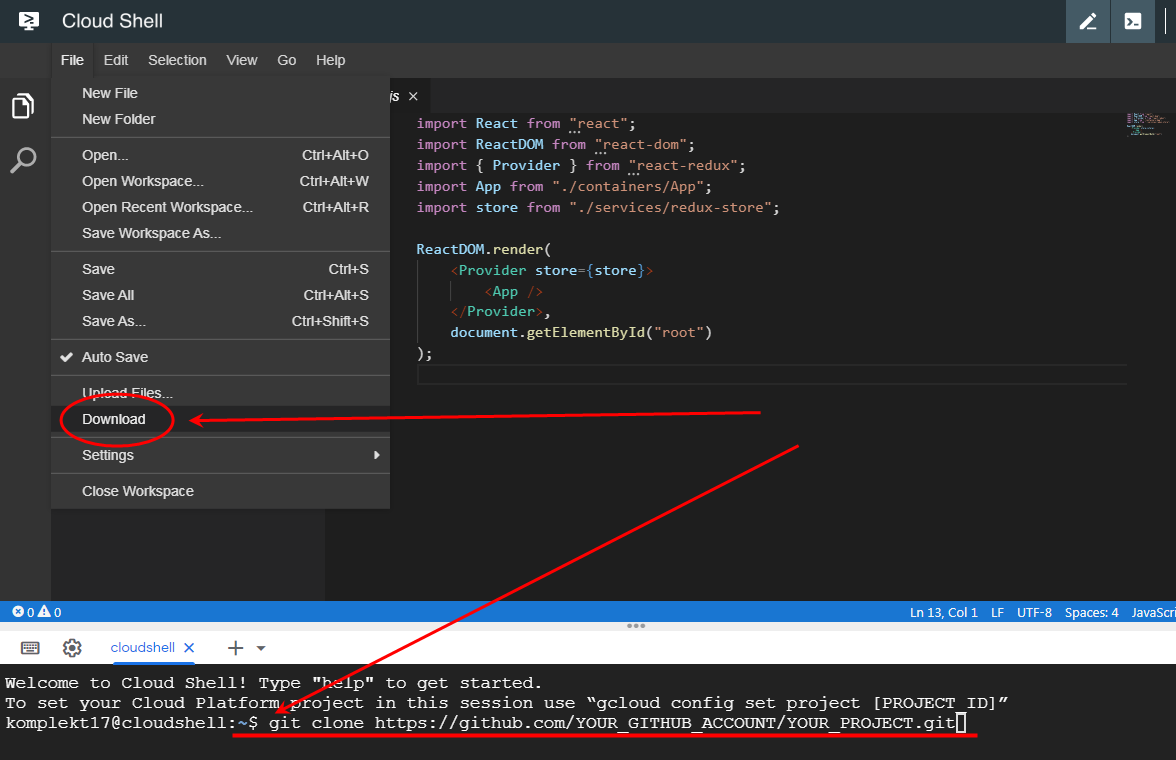
Выбираем [1] свой текущий аккаунт или другой



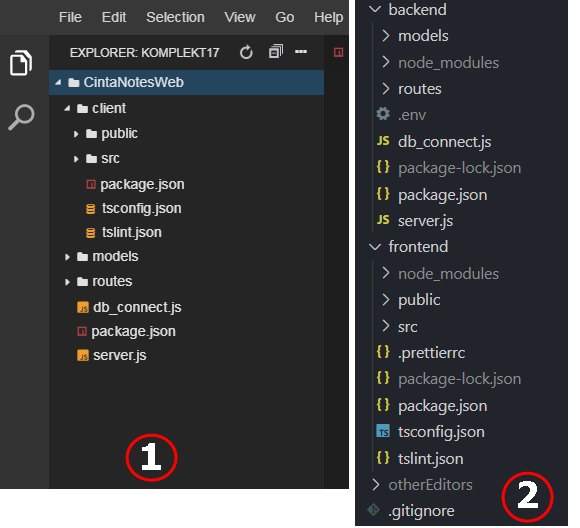
Выбираем существующий или создаём новый проект

Если все получилось, в скобках появится название вашего проекта желтым цветом. 

Теперь заливаем файлы проекта либо через меню File -> Upload Files… , либо клонируем через github

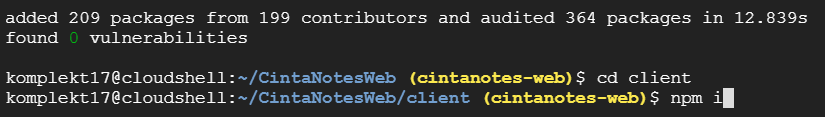


Настраиваем файлы. Обычно при разработке я использую расположение файлов проекта, как показано на рис 2, но для продакшена удобнее оказалось расположение, как на рис. 1



Устанавливаем пакеты для серверной части и client`а соответственно

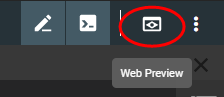




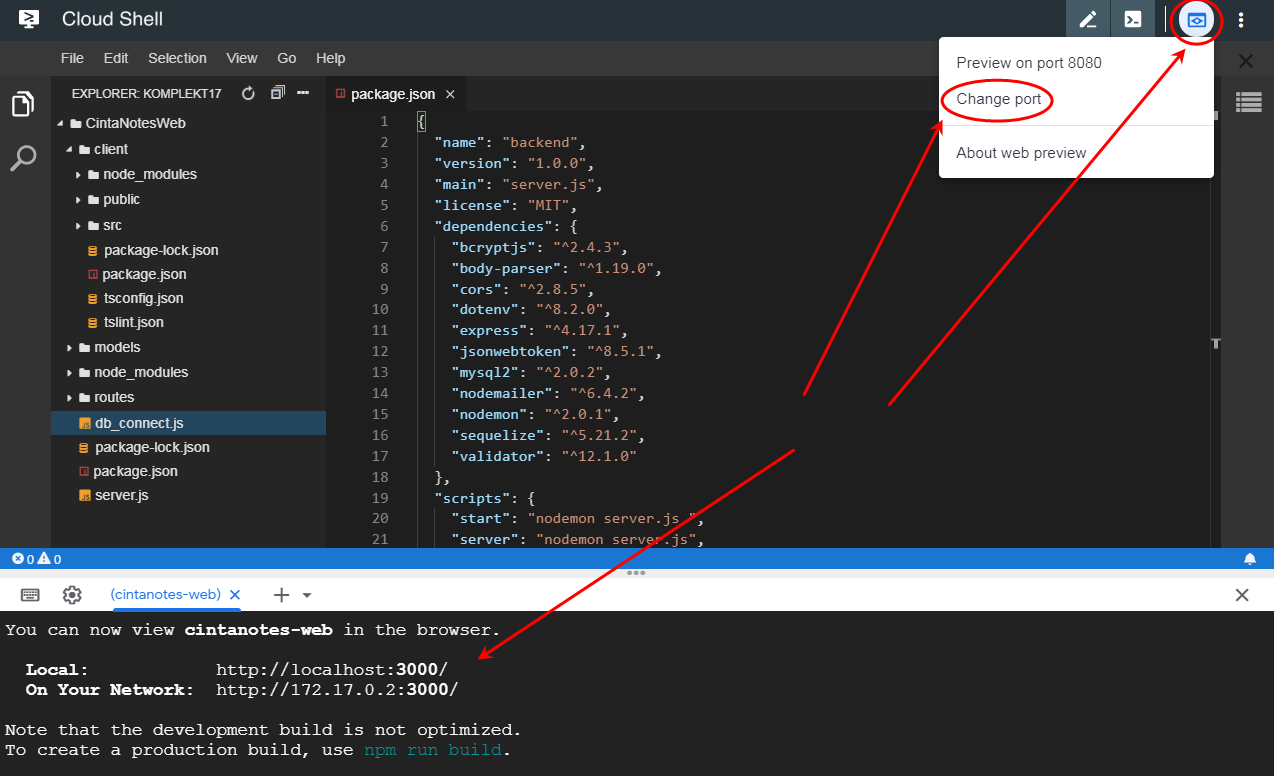
Теперь, если нужно для проверки корректности работы, можно запустить проект в режиме разработки.



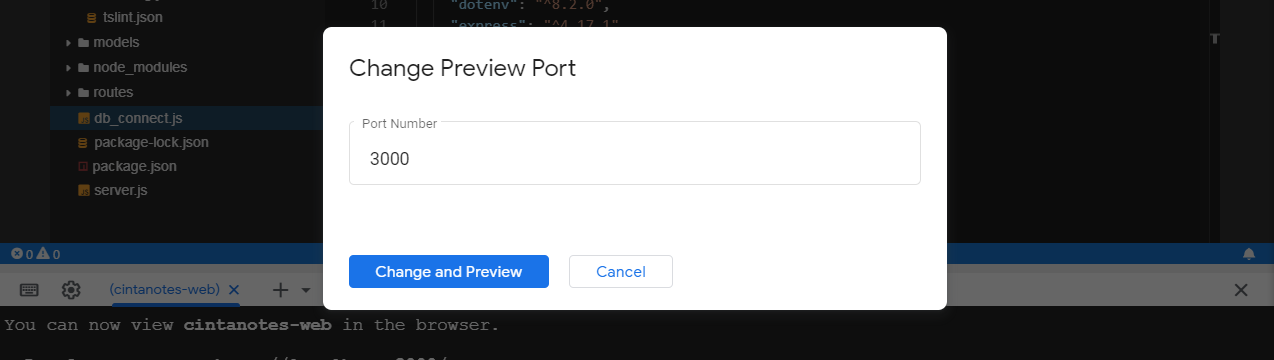
Для запуска предпросмотра жмём сюда



Обычно я использую шаблон разработки Create React App. Он запускает приложение на порте 3000, а редактор Cloud Shell по умолчанию запускает приложение на порте 8080.



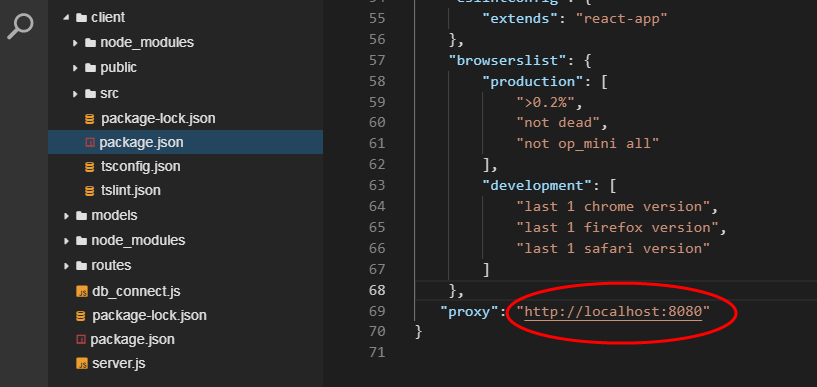
Поэтому меняем на 3000 и жамкаем Change and Preview



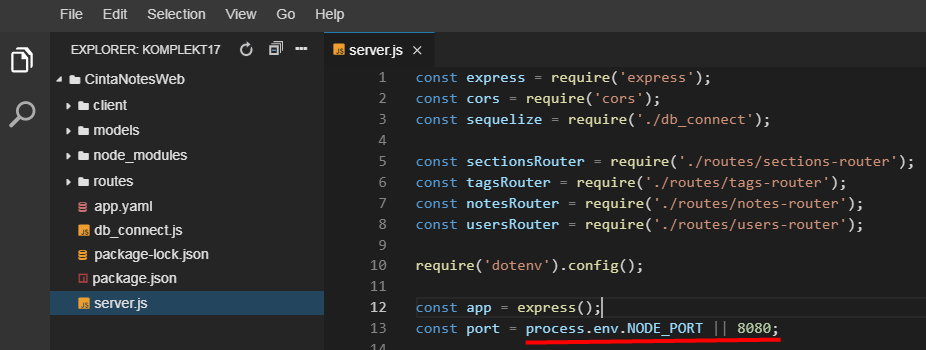
Теперь, если все запустилось и работает как надо, можно подготовить файлы для build`а проекта.

**Подготовка к билду и деплою, настройка varEnv.**

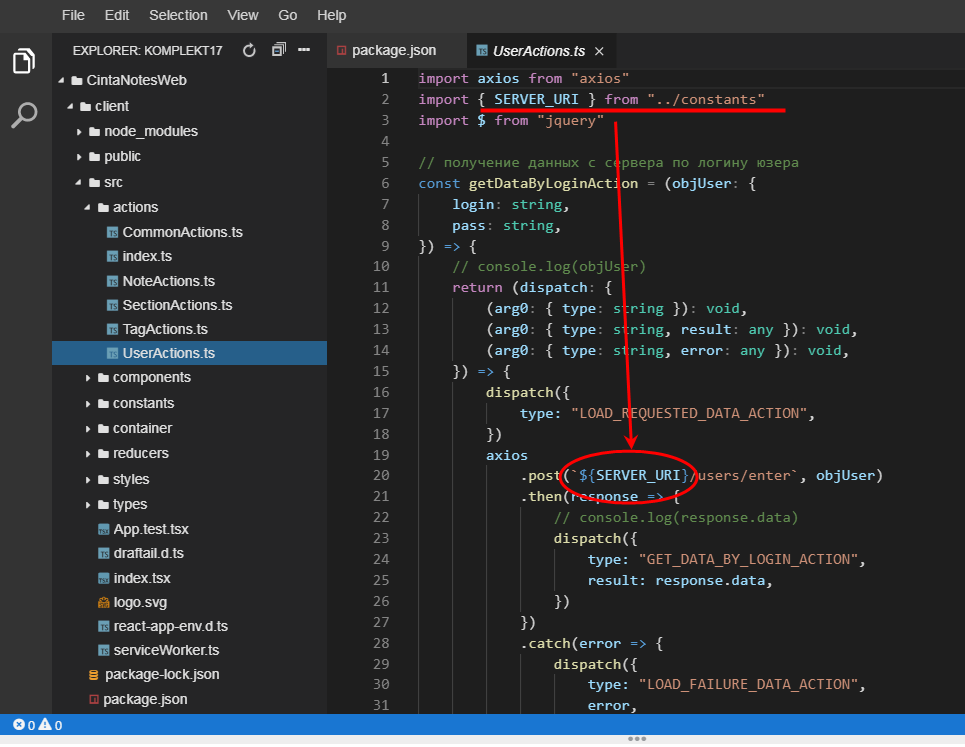
Настраиваем package.json client`а. Прописываем proxy на порт 8080.



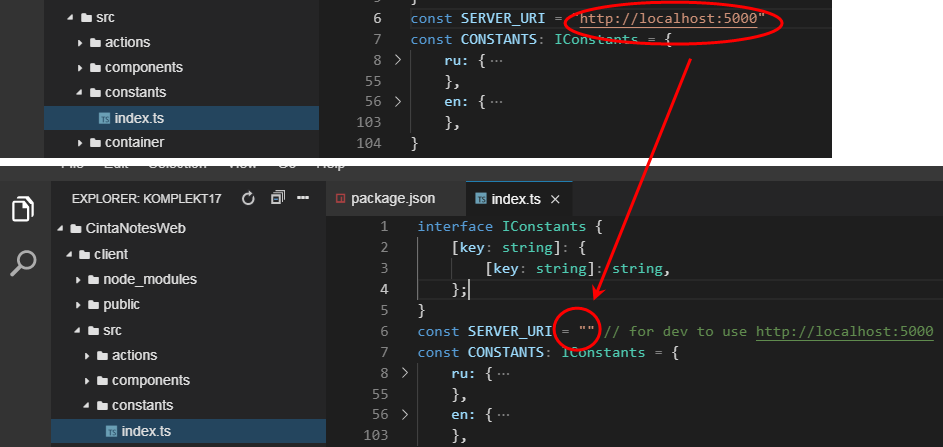
Для сервера тоже прописываем порт 8080



Настраиваем взаимодействие с сервером. Для асинхронных запросов на сервер в actions я использую константы из директории constants

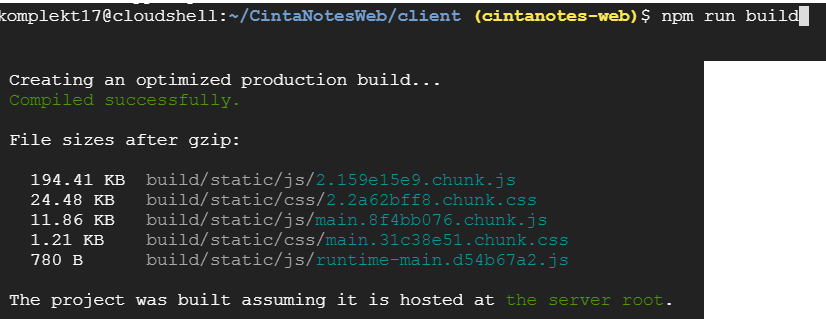


Необходимо оставить константу SERVER\_URI пустой



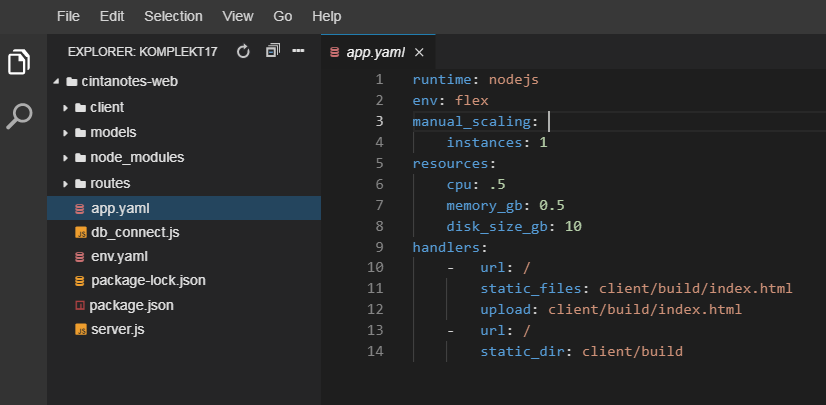
Доки по теме тут <https://cloud.google.com/nodejs/getting-started/background-processing>

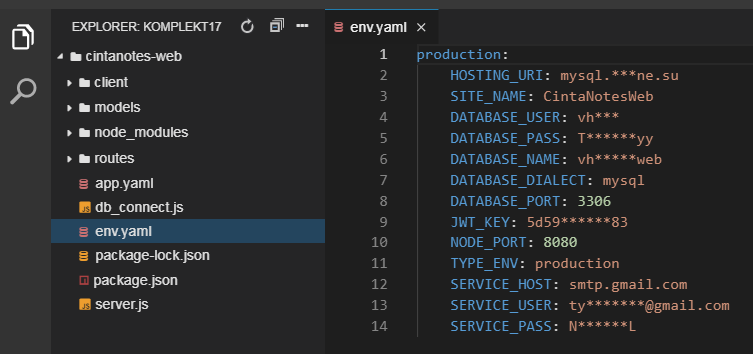
Собираем build client`а



**Настройка файла app.yaml и vars environtment**

Создаём файл app.yaml в корневом каталоге проекта, который позволит App Engine знать, какую конфигурацию вы хотите.

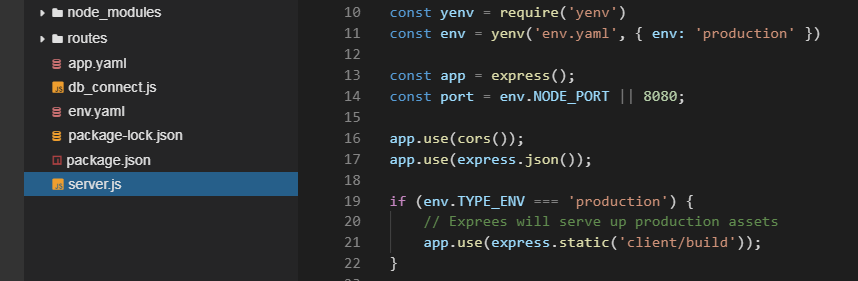


Далее создаём файл env.yaml в корневом каталоге проекта с переменными окружения

Далее устанавливаем этот пакет <https://www.npmjs.com/package/yenv> для обработки env-файла.

$ npm install yenv --save

Инструкция тут <https://github.com/jeffijoe/yenv>

Прменять так 

Теперь деплоим приложение

$ gcloud app deploy -–stop-previous-versions

Ждёмс, Деплой примерно минут 5-7. Если ошибок не случилось, запускаем такую команду:

$ **gcloud app browse**

Появится ссылка на ваш проект https://<your\_project>.appspot.com. Всё, конец фильма!

PS: Полезные ссылки

Deploying a MERN app to Google Cloud Platform’s Google App Engine

https://medium.com/@paulrohan/deploying-a-react-node-mongodb-app-to-google-cloud-platforms-google-app-engine-1ba680447d59

https://github.com/rohan-paul/material-ui-table-with-node-mongodb