МИНИСТЕРСТВО НАУКИ И ОБРАЗОВАНИЯ РФ  
Федеральное государственное бюджетное образовательное учреждение  
высшего образования

«Тихоокеанский государственный университет»

Кафедра «Вычислительная техника»

Практическая работа №4  
«НЕПРЕРЫВНАЯ ИНТЕГРАЦИЯ/НЕПРЕРЫВНАЯ ПОСТАВКА»

Выполнил:  
Студент группы КСТ(б) - 01  
Яковец Александр Евгеньевич

Проверил:  
Преподаватель  
Саенко Александр Александрович

Хабаровск 2022

**Цель работы**: Осуществить настройку CI/CD.

**Задания:**

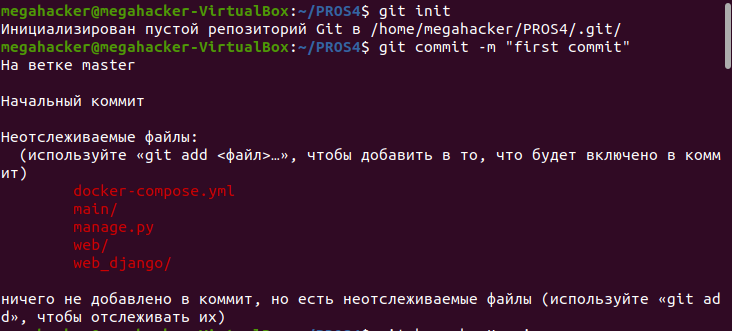
1. Написание своего приложения на любом языке программирования.

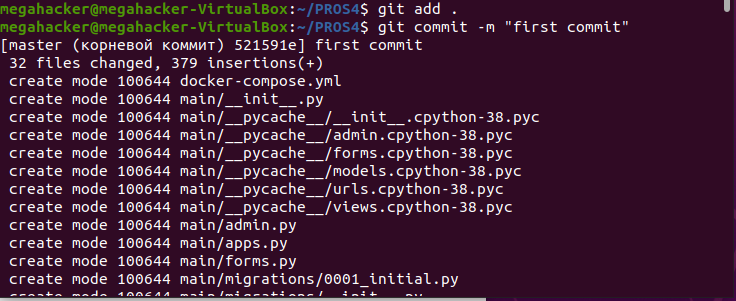
2. Отправка всех изменений в систему контроля версий.

3. Используя сервер непрерывной интеграции (например: Jenkins) и если необходимо систему управления конфигурациями (например: Ansible) либо любые аналогичные инструменты. Настроить CI/CD с помощью Docker и Kubernetes.

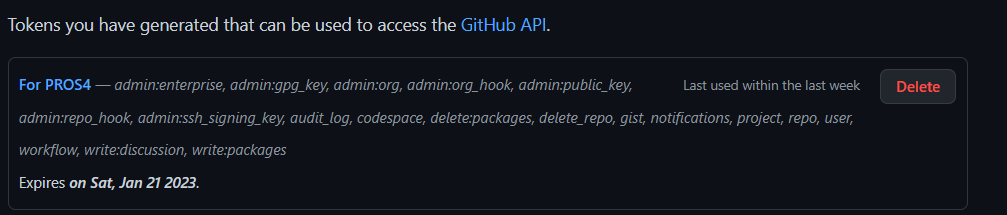
**Ход работы:**

Первое с чего мы начнем, настроим гит. Инициализируем папку, добавим файлы, добавим в локальную ветку, коммитним, изменим ветку master на мейн, свяжем нашу директорию с директорией гита и загрузим директорию (перед этим создав специальный токен для входа).

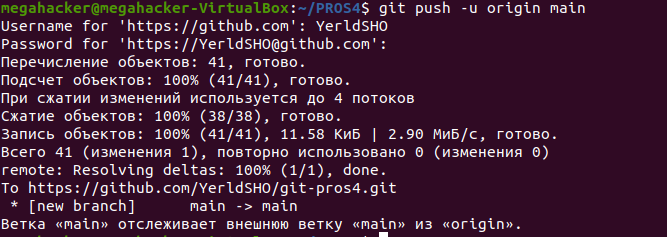


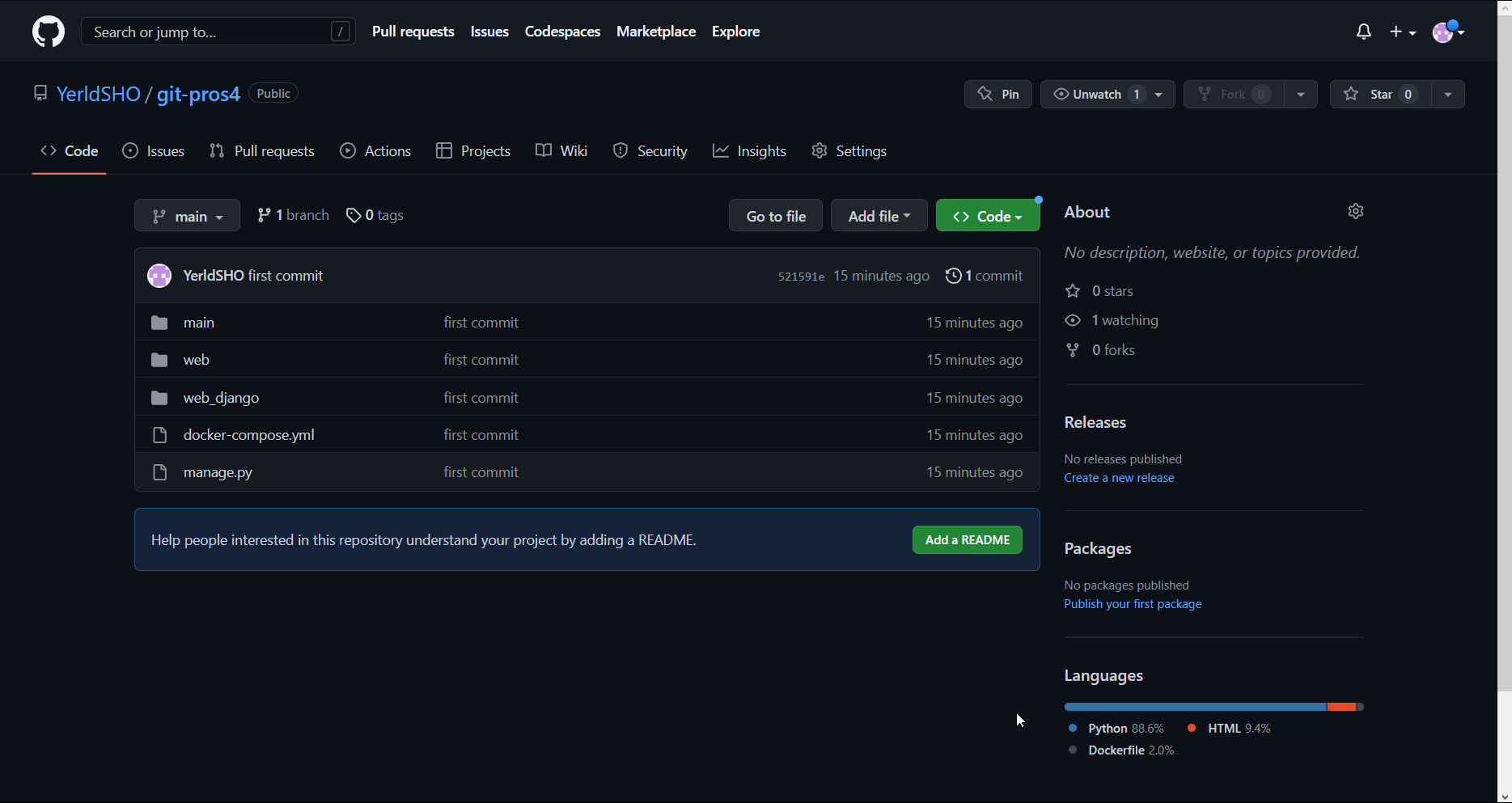






**“Здесь должен быть ваш токен GitHUB”**



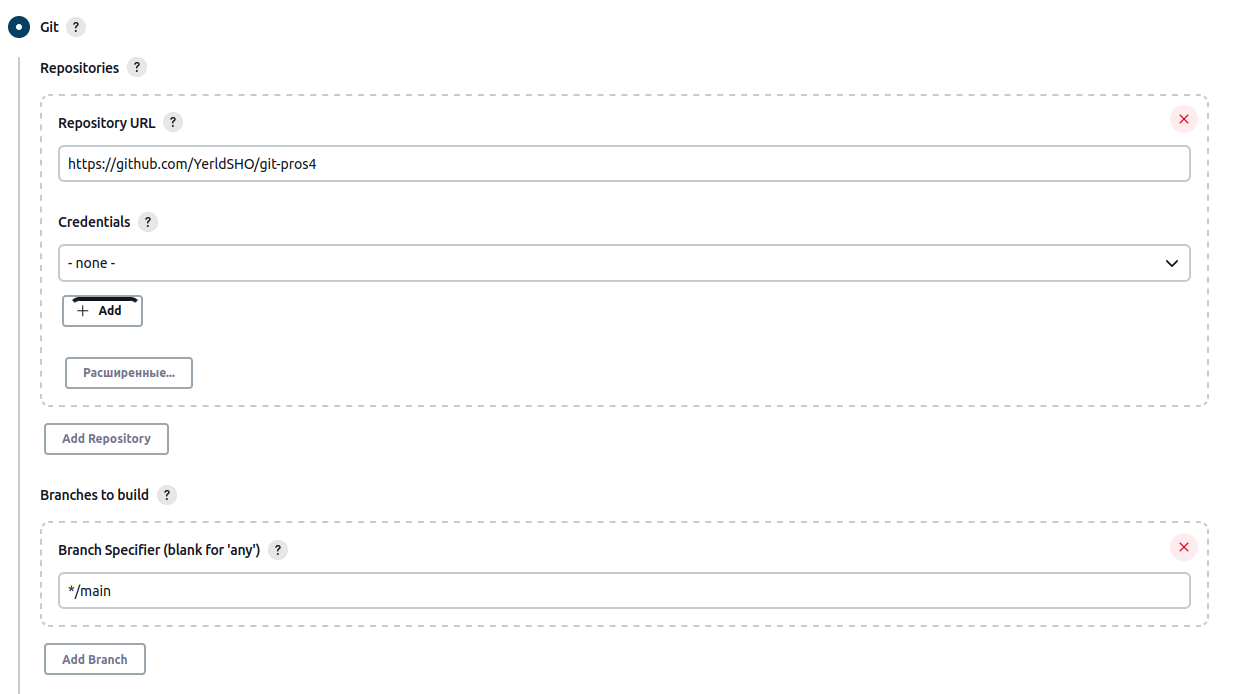


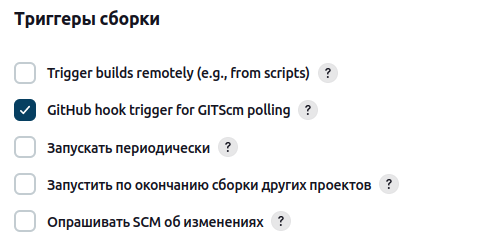
2 задание мы осуществили, перейдем к полной настройке Jenkins и Kubernesis.

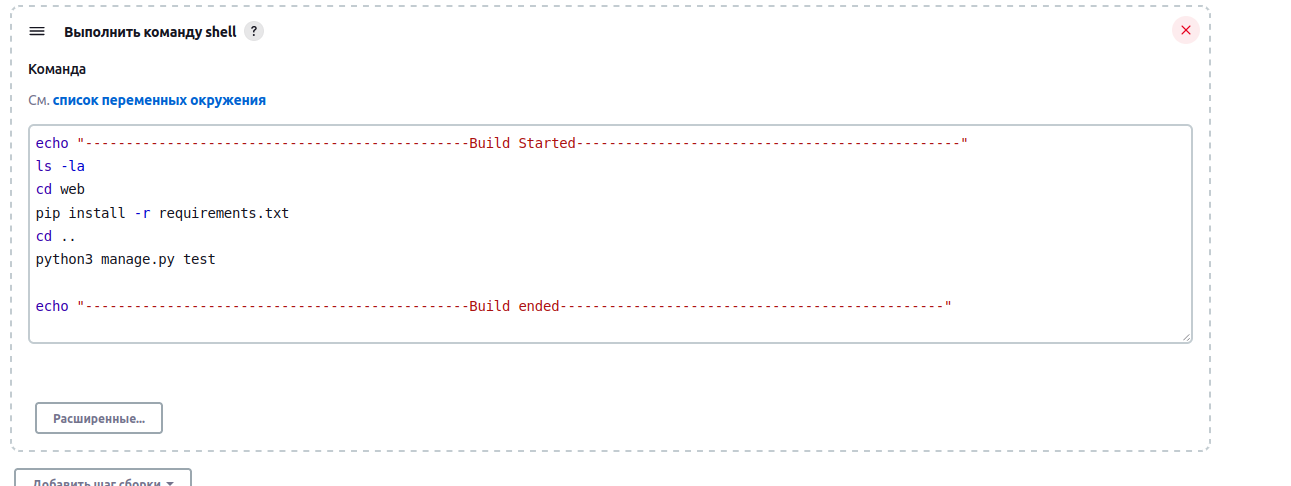
Установим Jenkins, создадим item Test, подключим к нему гит, а также авто проверку тестов после коммита и загрузки репозитория на гитхаб, также реализован автоматический заход (но он не работает), все тесты лежат в файле test.py. Для подключения используем admin и сгенерированный токен Jenkins

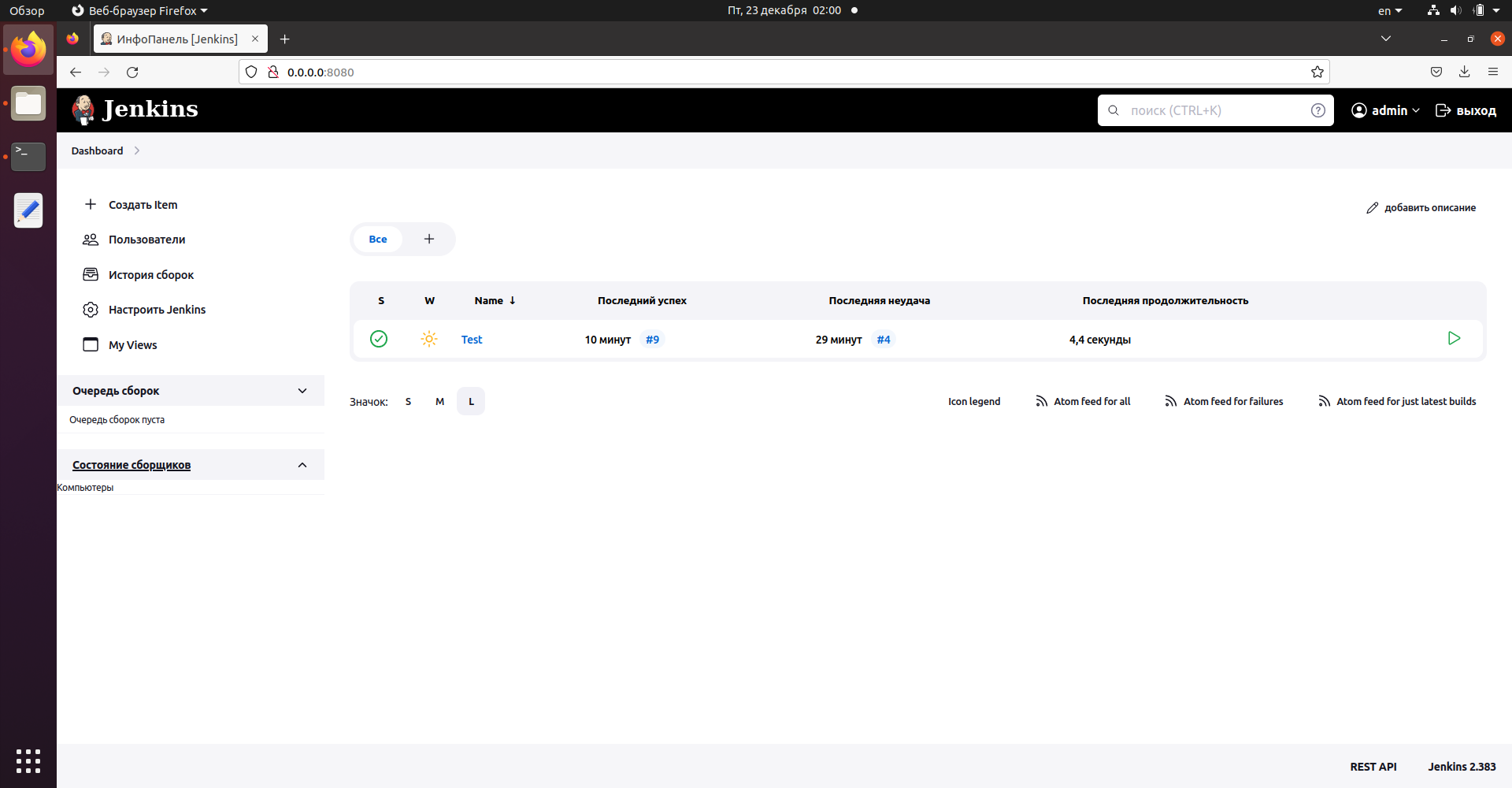
**“Здесь Должен быть ваш токен дженкинс”**

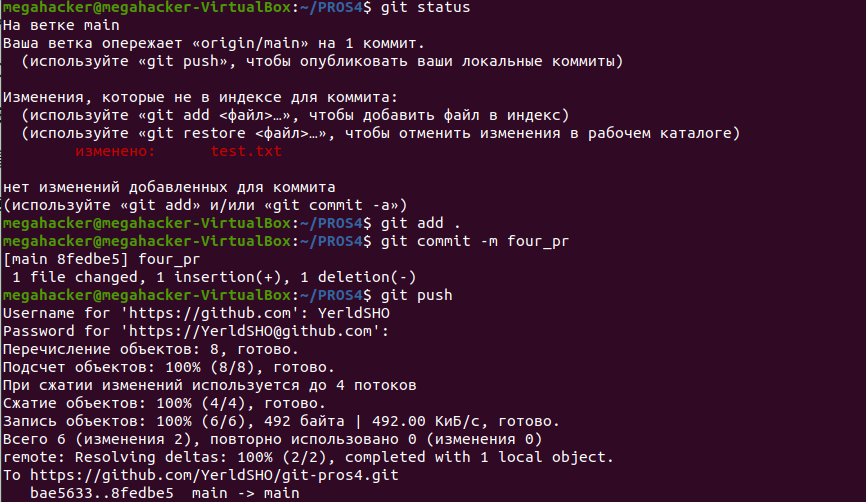


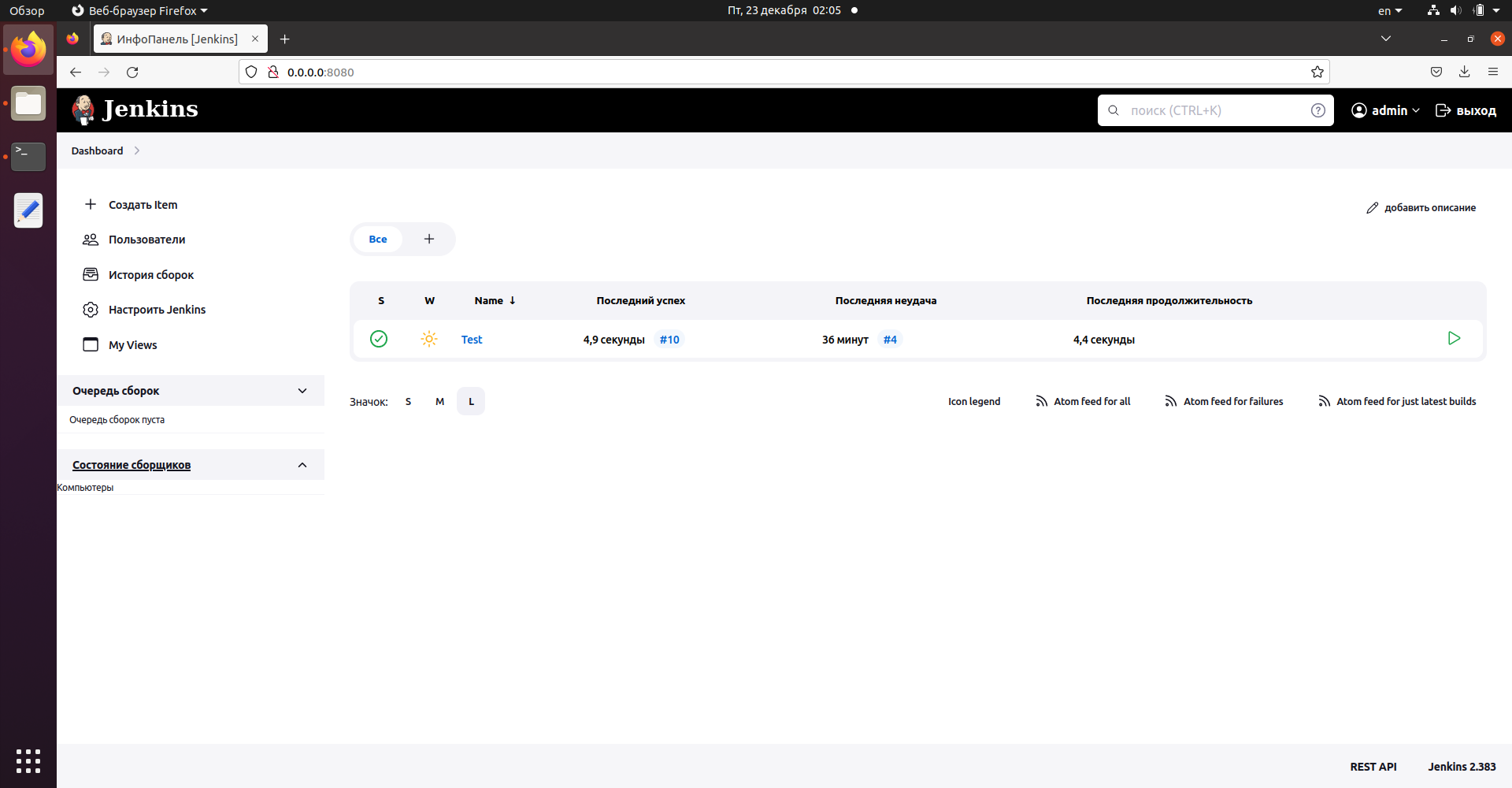


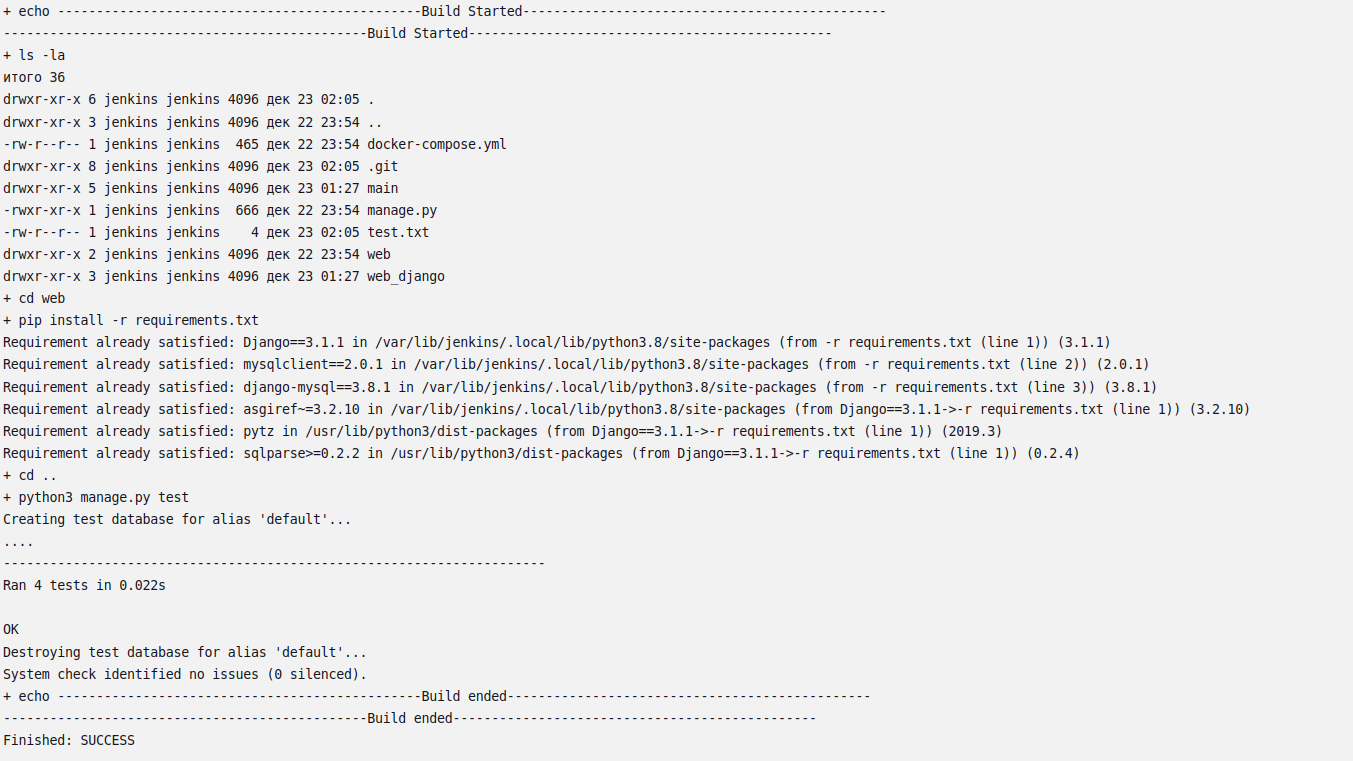




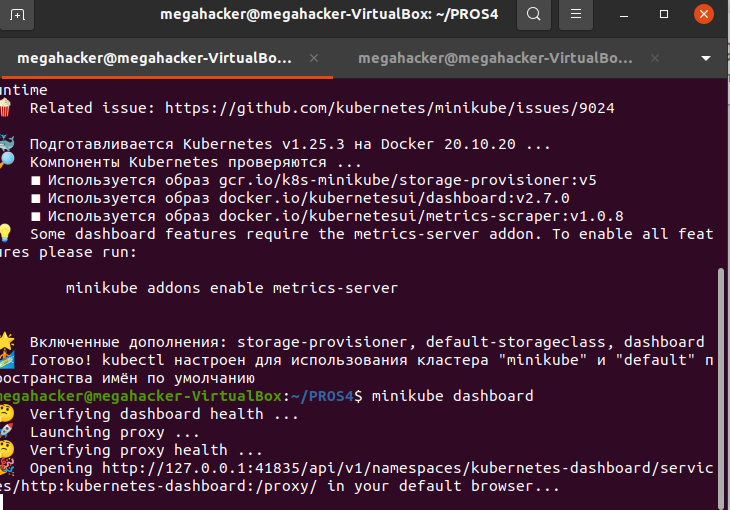


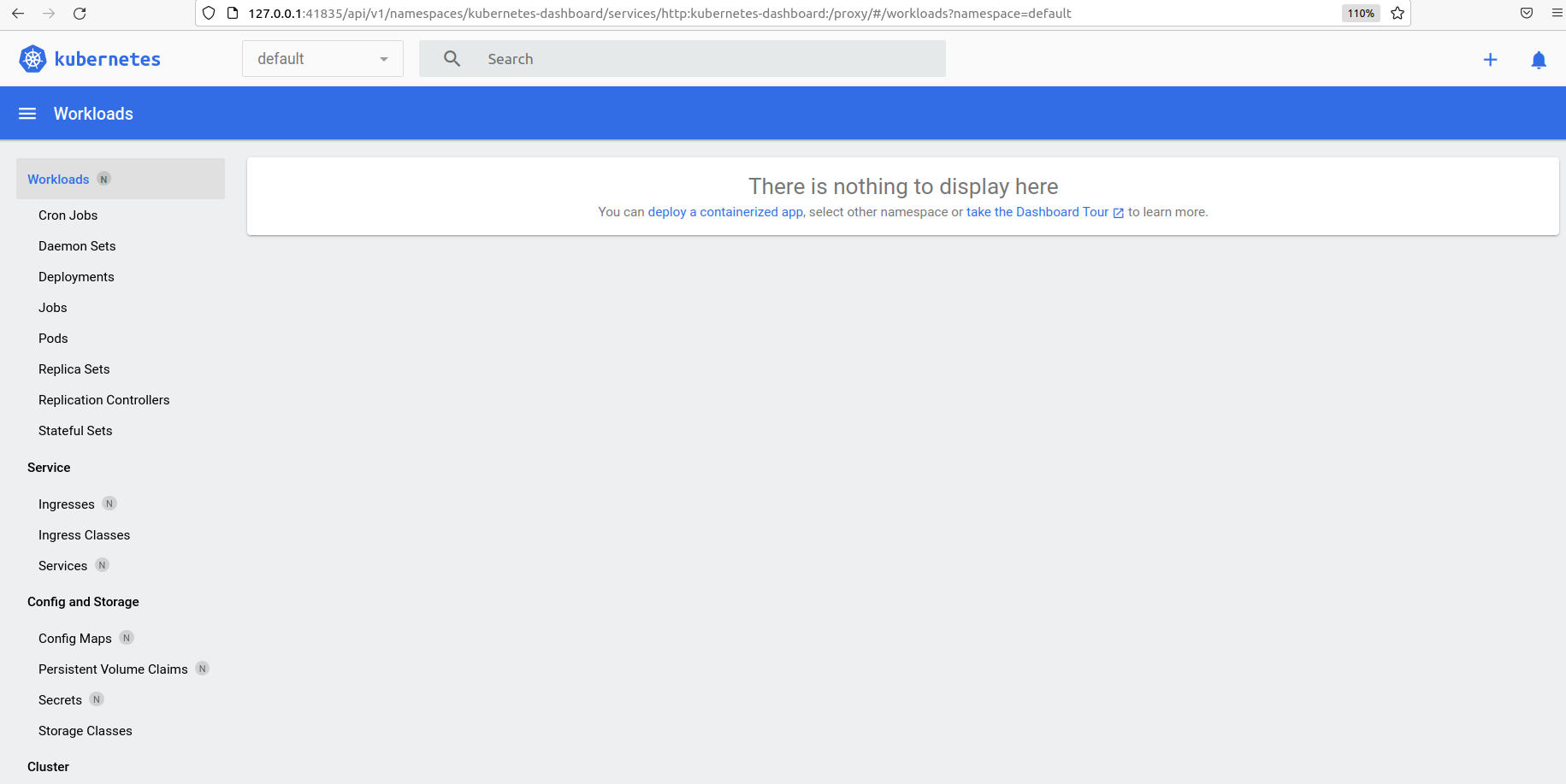




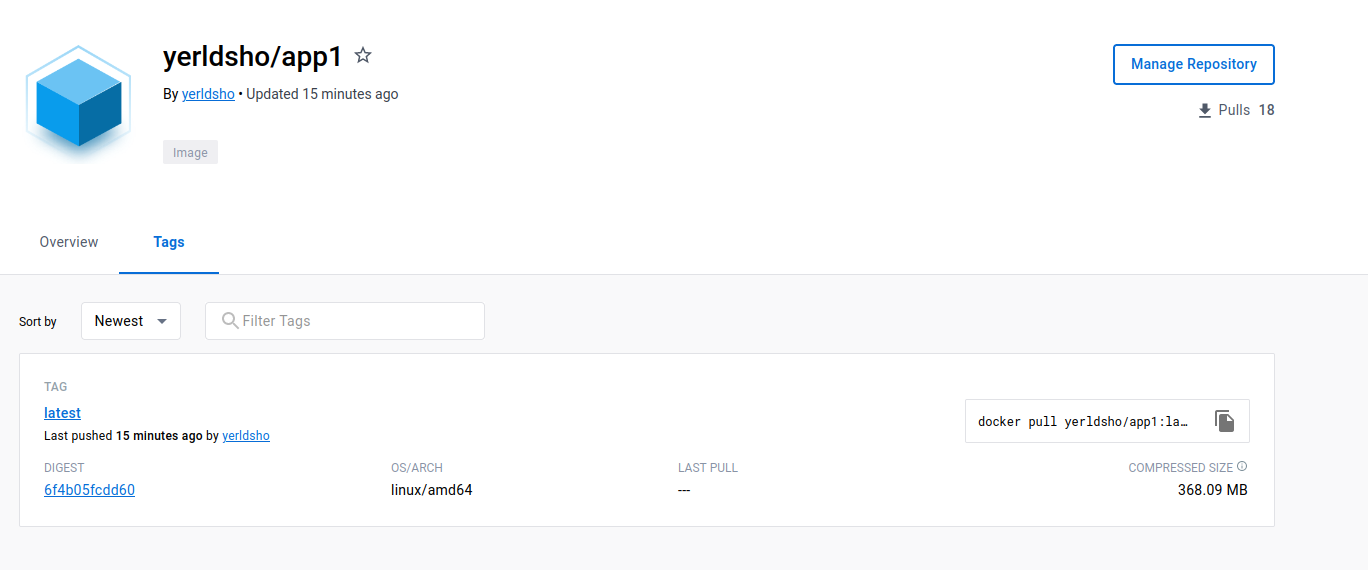


После всех установок minikube(kubernetes), мы плавно переходим к настройке куба. Запускаем куб и dashboard (minikube start и minikube dashboard).

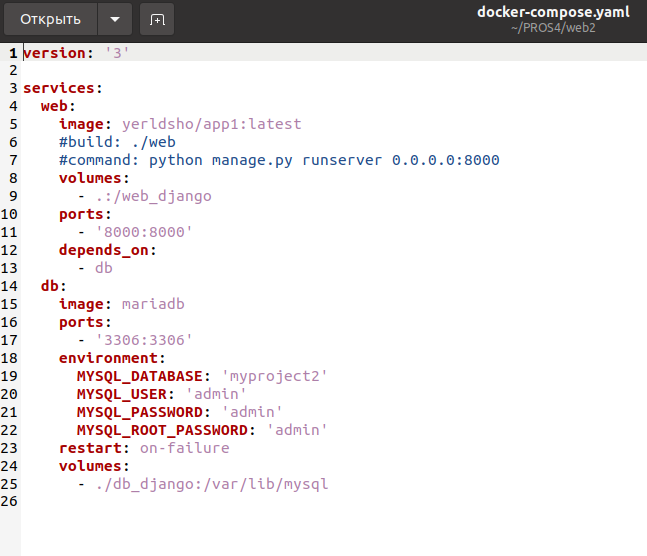




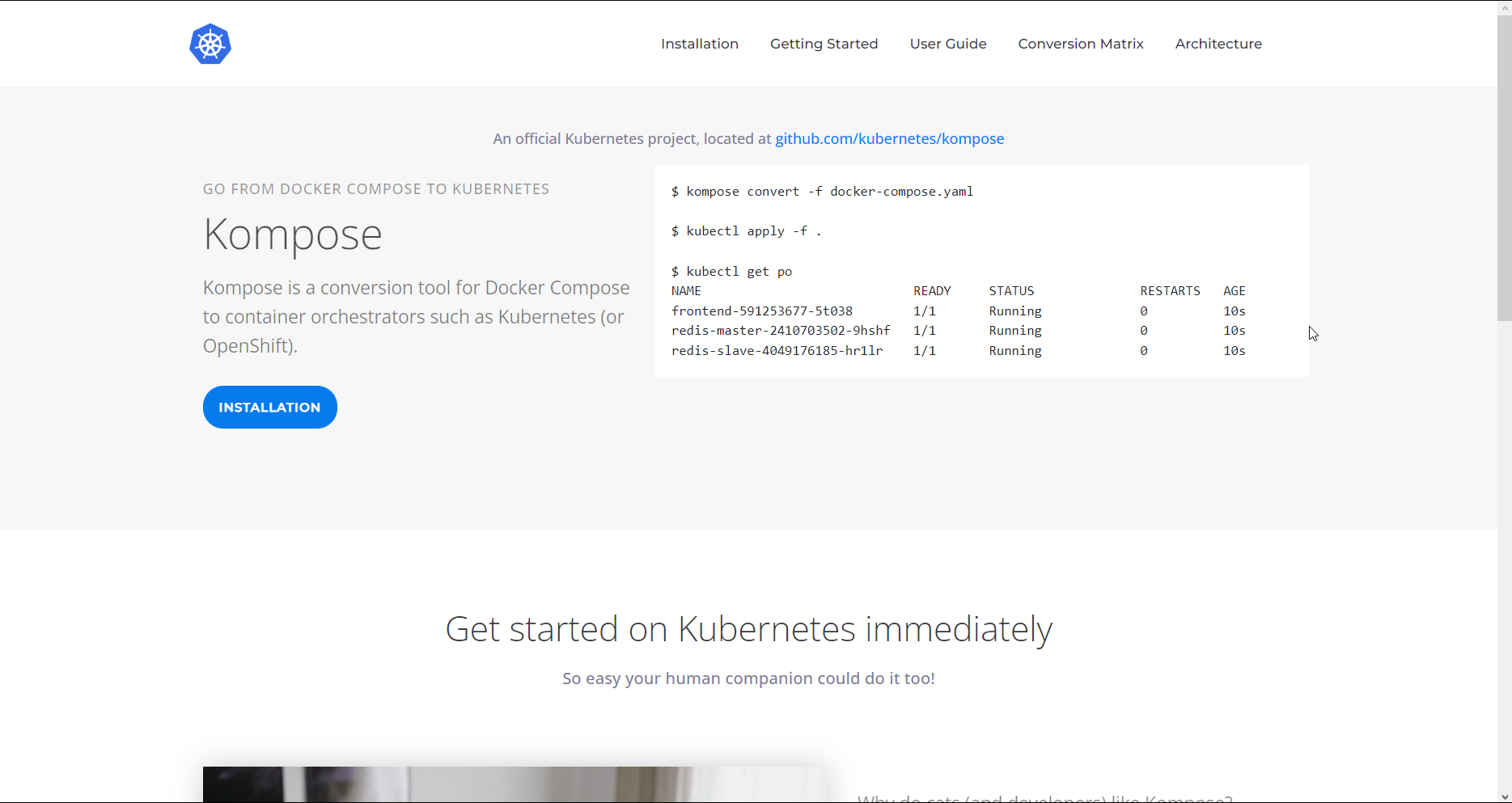
После этого создадим свои первые деплои и сервисы для kubernetes из dockerfile и kubernetes. Для этого нам понадобится докер хаб. Регистрируемся. Далее собираем dokerfile (“docker build –t никDockerHub/Название .”). После прописываем команды “docker login -u ник -p пароль” и “docker push никDockerHub/Название:тэг(типо latest)”.



Следующим действием будет создание deployment, services и pods. Изменим наш docker-compose.yaml добавив наш dockerHub image:



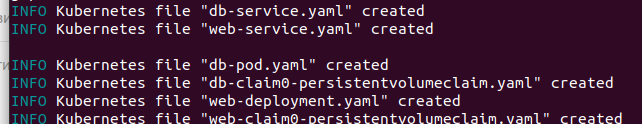
Далее качаем compose, это доп. Инструмент, который облегчает нам создание deployments, services и pods.

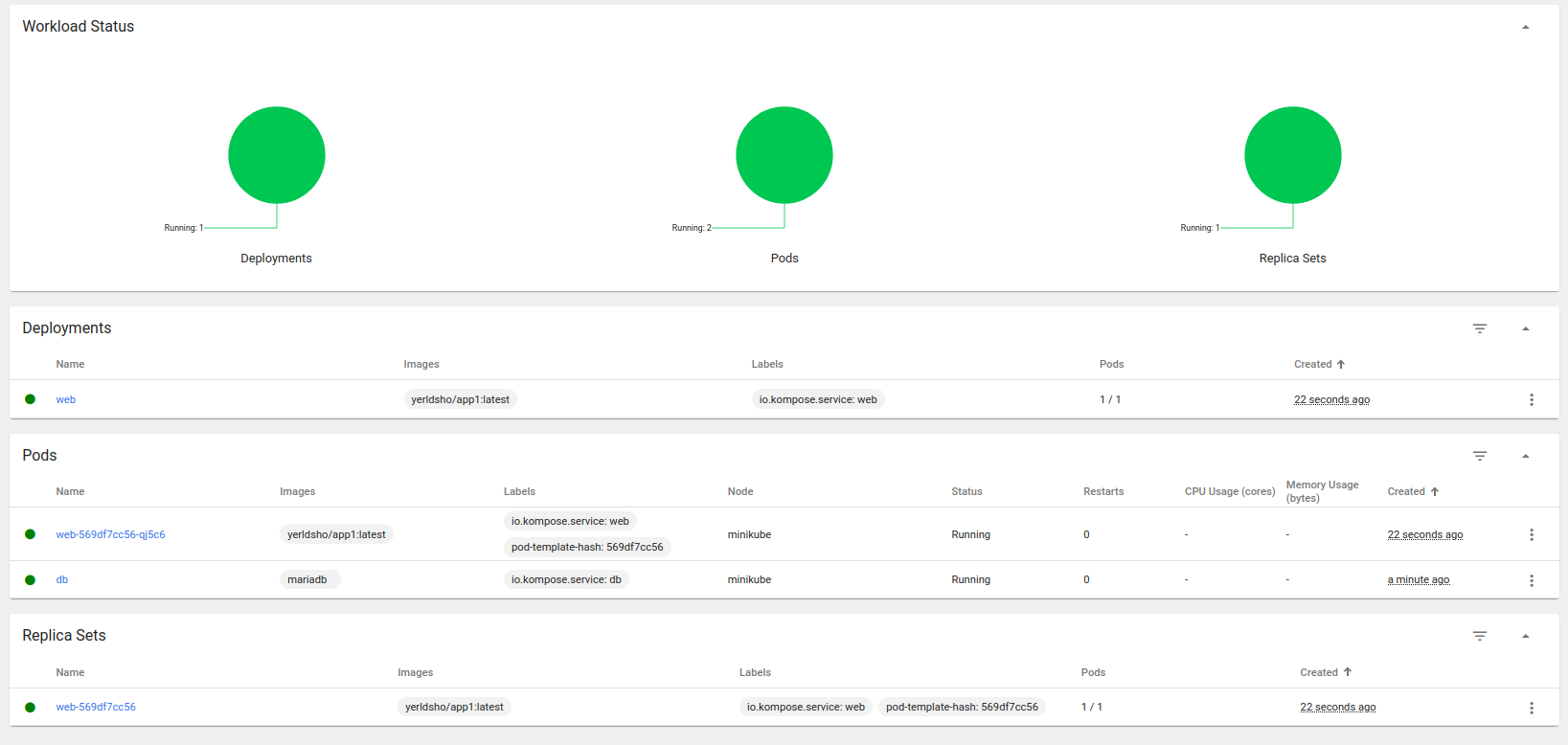


Далее прописываем это:

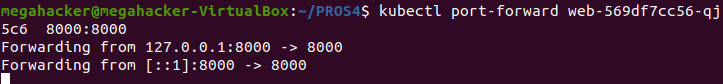
kompose convert -f docker-compose.yaml

kubectl apply -f .





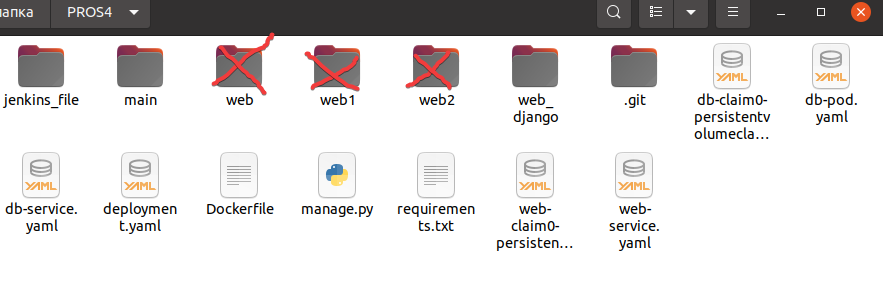
Далее выбираем имя пода web-569df7cc56-qj5c6 и прописываем:



Заходим на 127.0.0.1:8000 и видим, что все работает и в том числе бд. Удаляем и добавляем вместо web-deployment наш файл для следующего удобства:

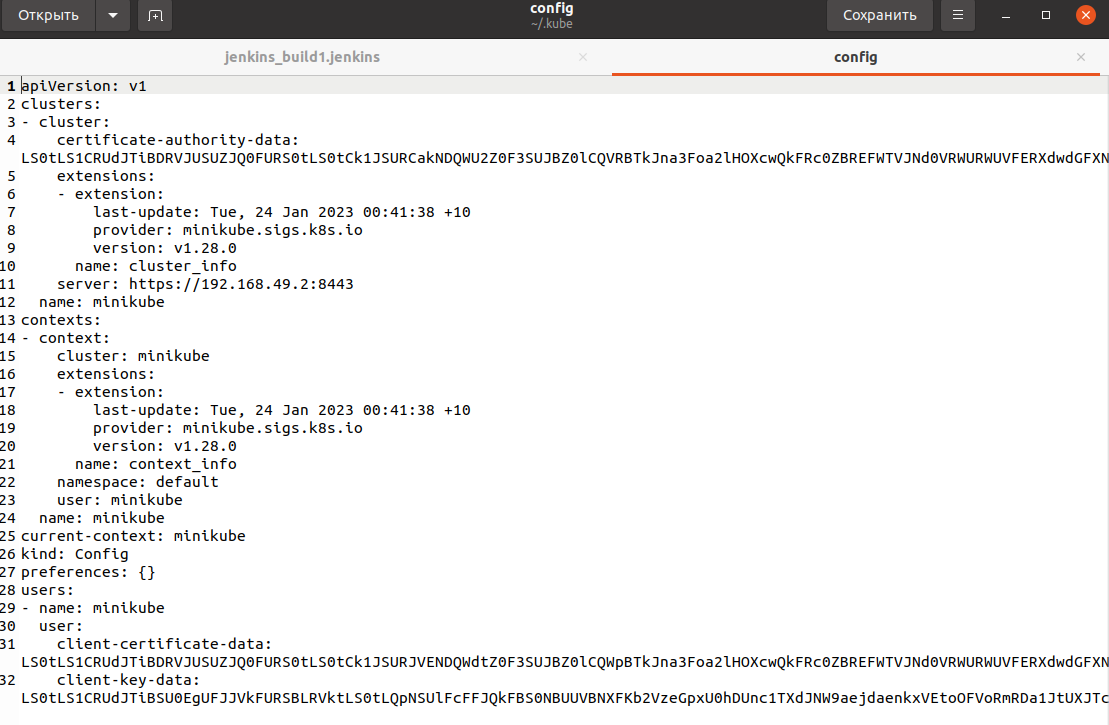


И так подготовку мы совершили, осталось оптимизировать все это до 4 команд. Составная проекта выглядит так (зачеркнуты не имеющие смысл папки):

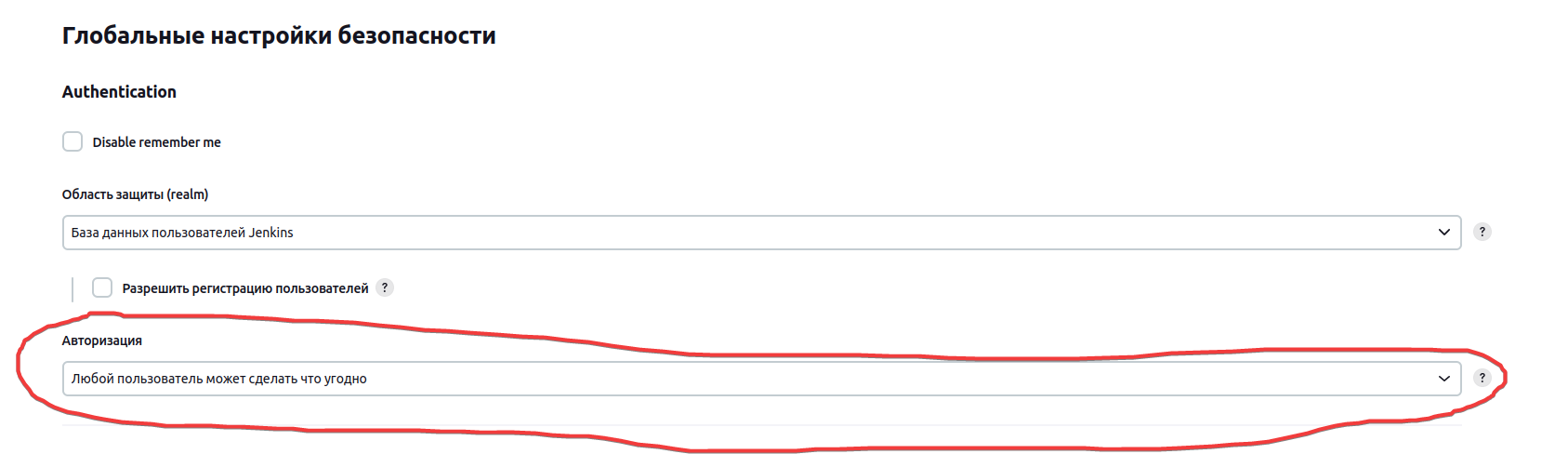


Прописываем это (перезапускаем кубик) и видим что наш конфиг в .kube/config меняется на лучший вид:

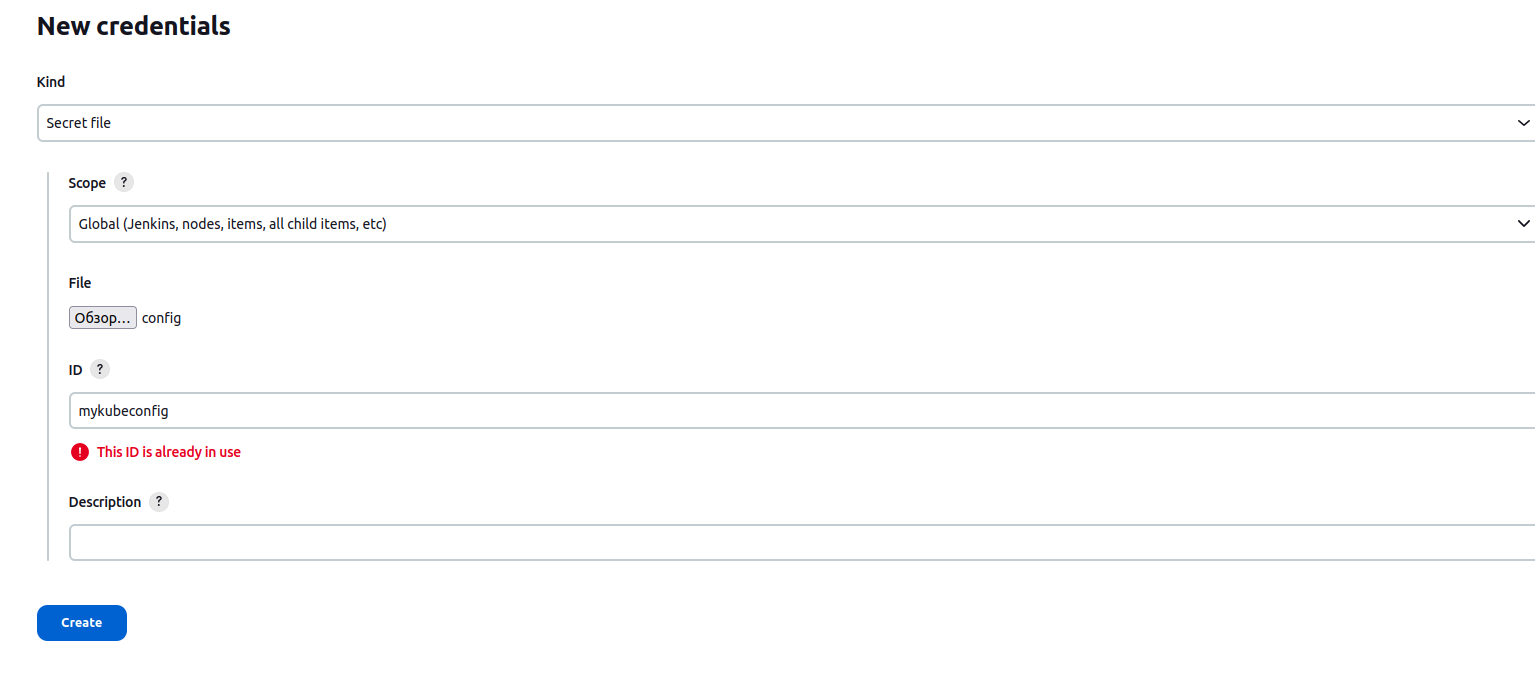
**minikube start --embed-cert**



Вернемся в Jenkins, сейчас нам нужно будет создать пайп лайн, но перед этим мы зададим все нужные настройки, на которые у меня ушло порядка недели :) повторяйте по скринам.

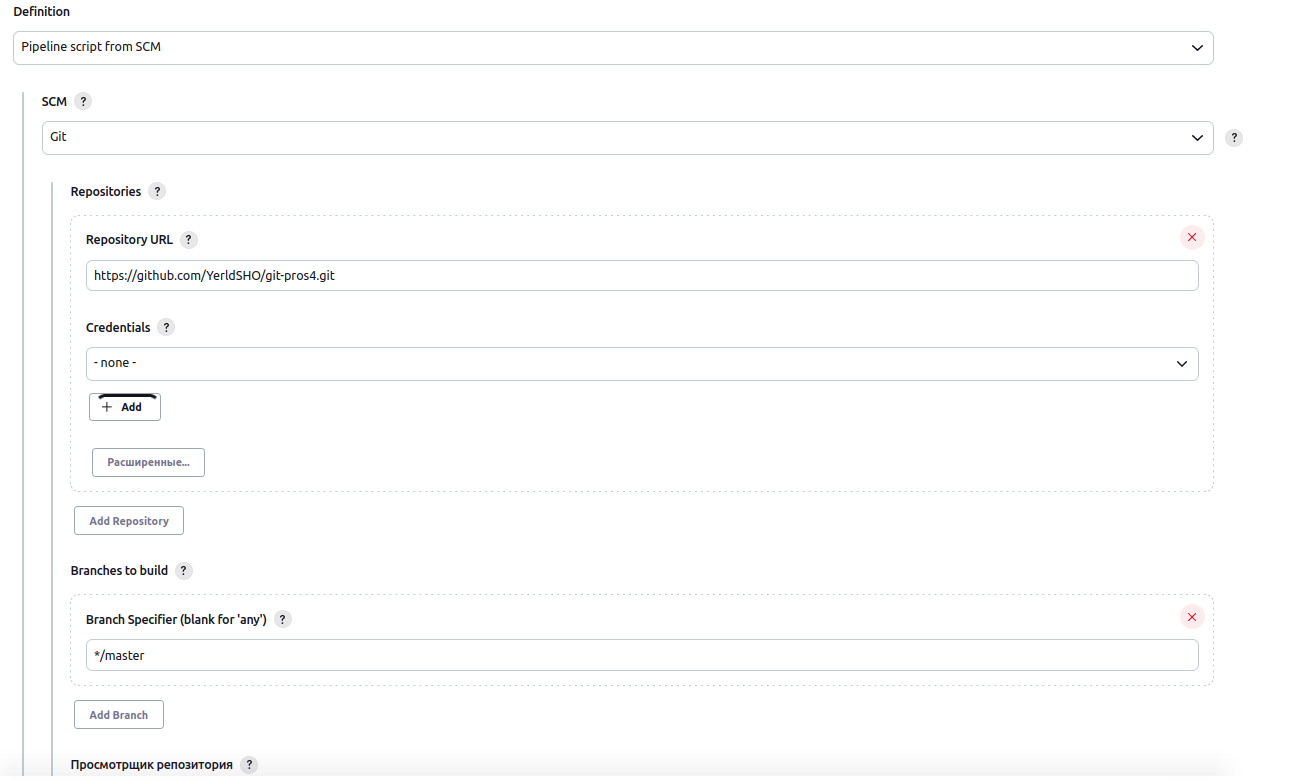


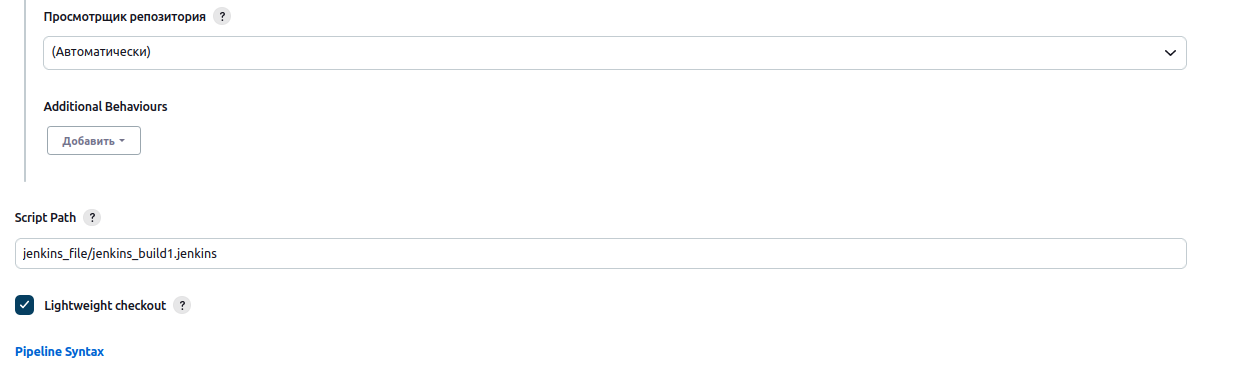
**Manage Credentials**



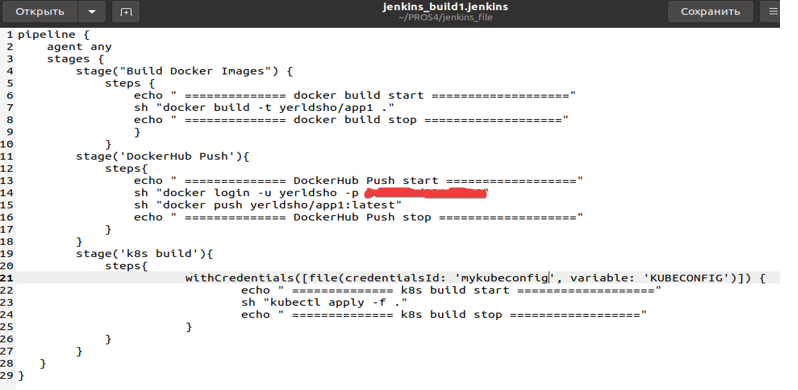
Все подготовления закончены, создадим пайп-лайн и Jenkins-file:

Пайп-лайн

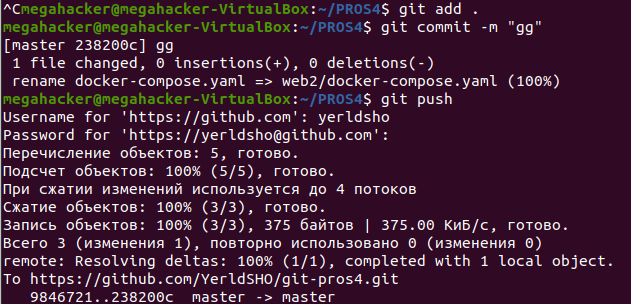


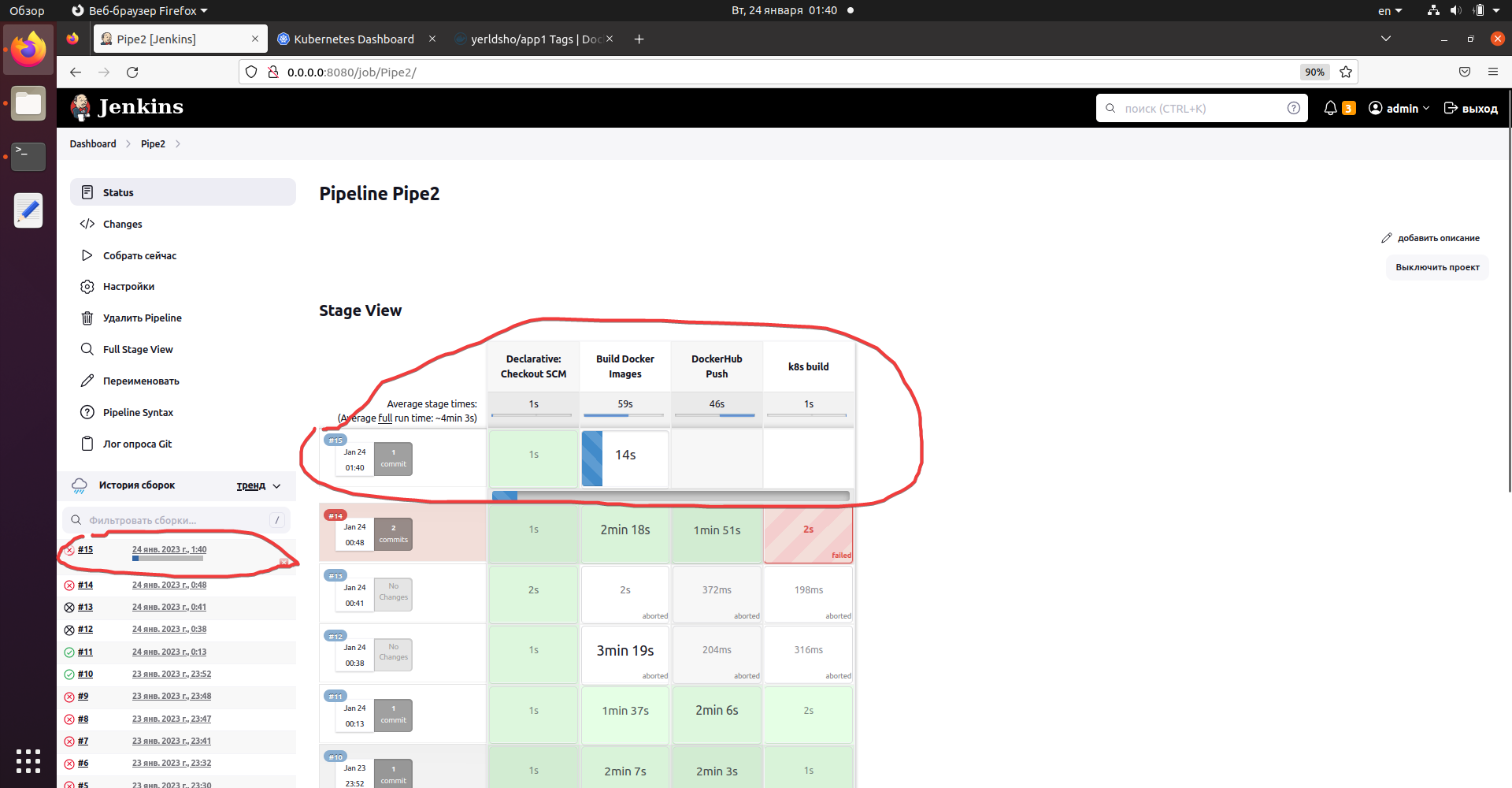


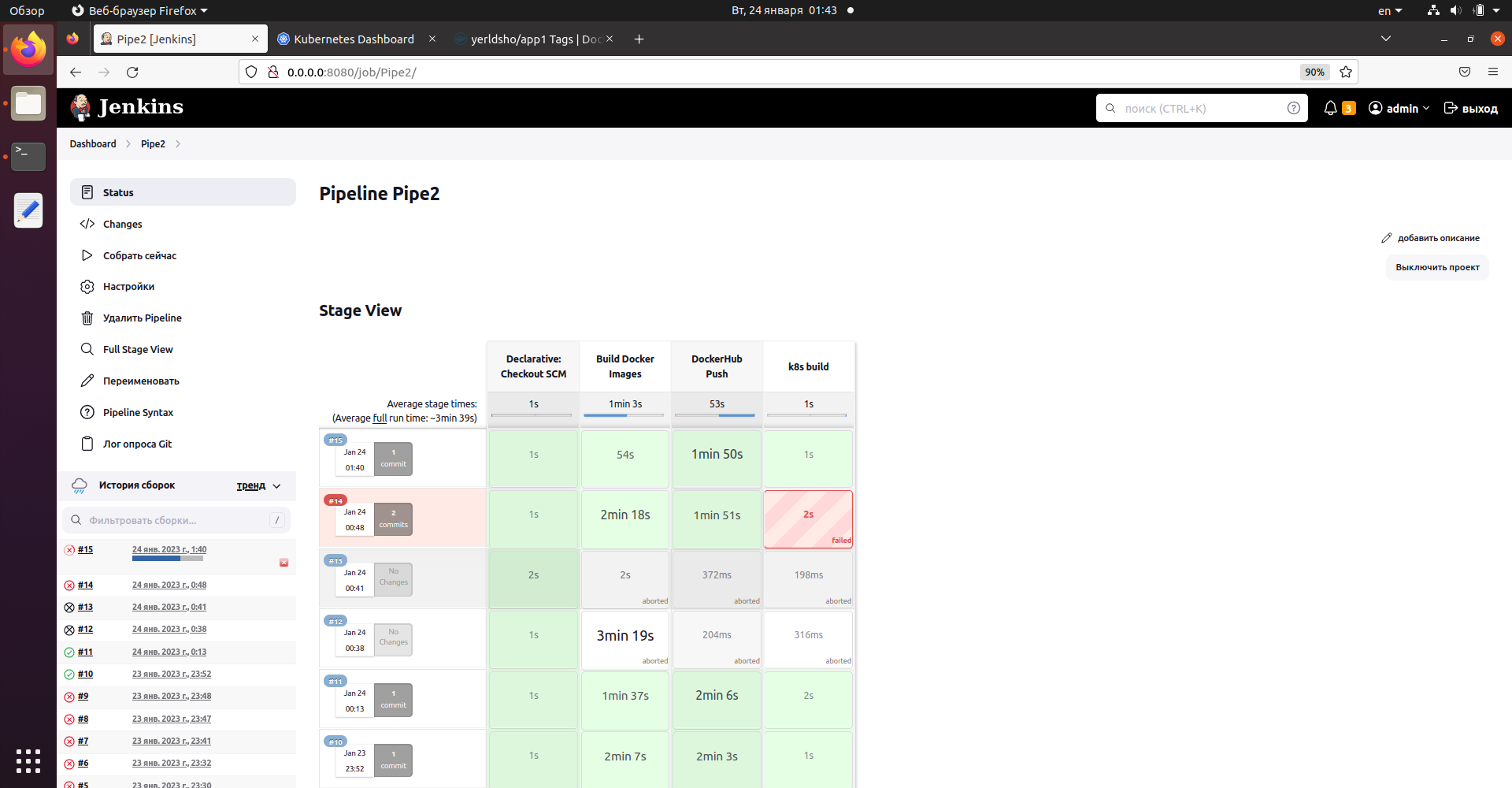
Путь указан, как на GitHub. Создадим же Jenkins-File:

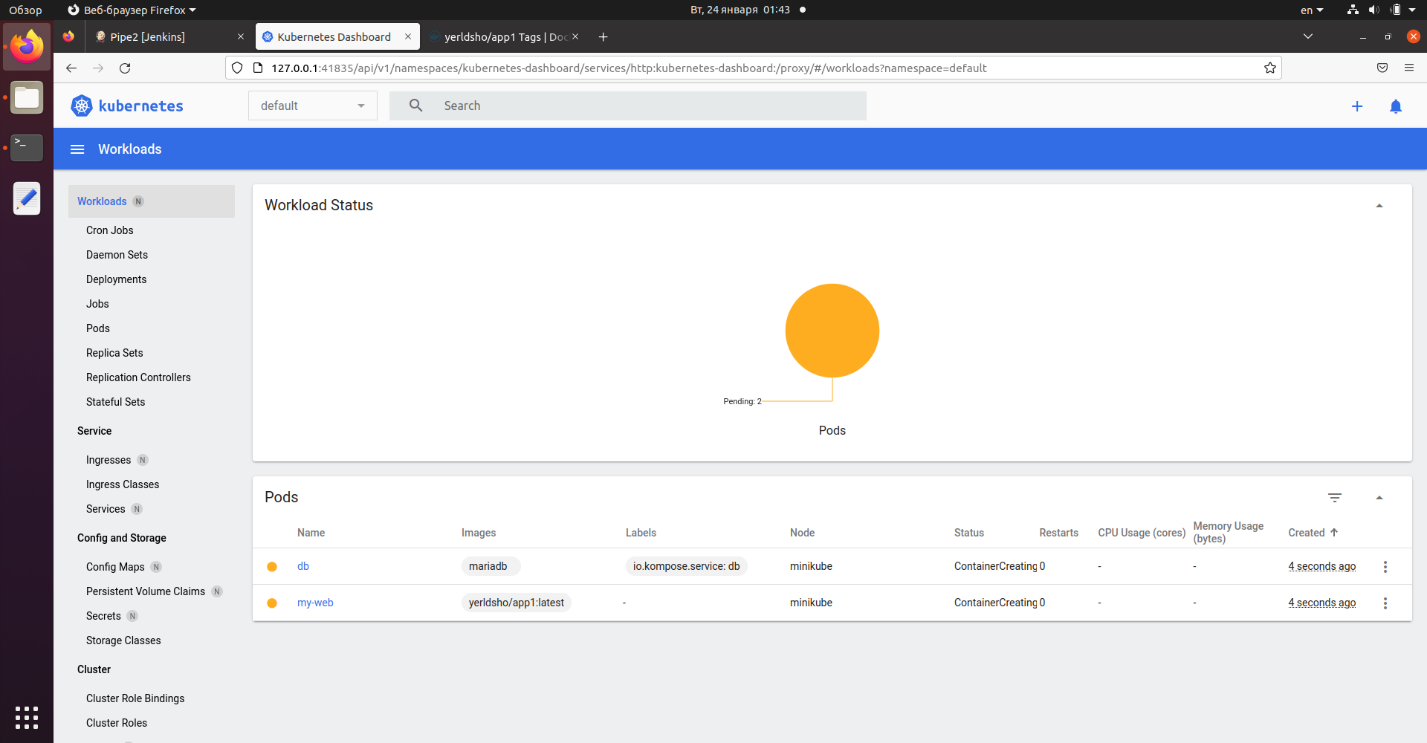


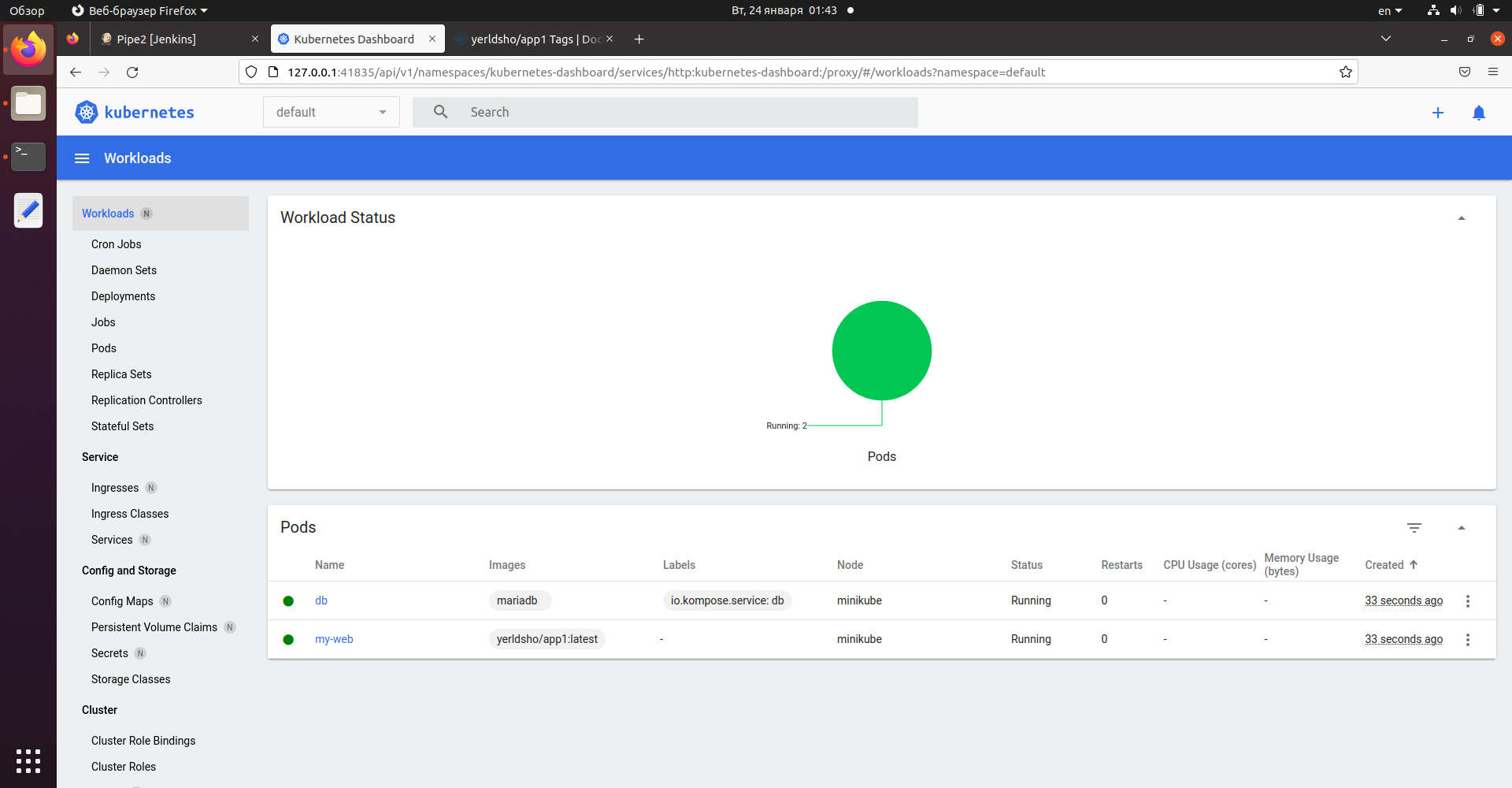
Итак, все готово, теперь мы прописываем 3 команды в начале и 1 в конце (обязательно, если мы осуществляем сборку по новой, нужно удалить все в kubernetes):

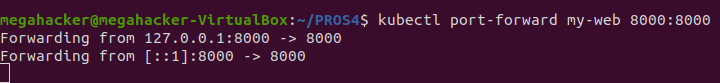


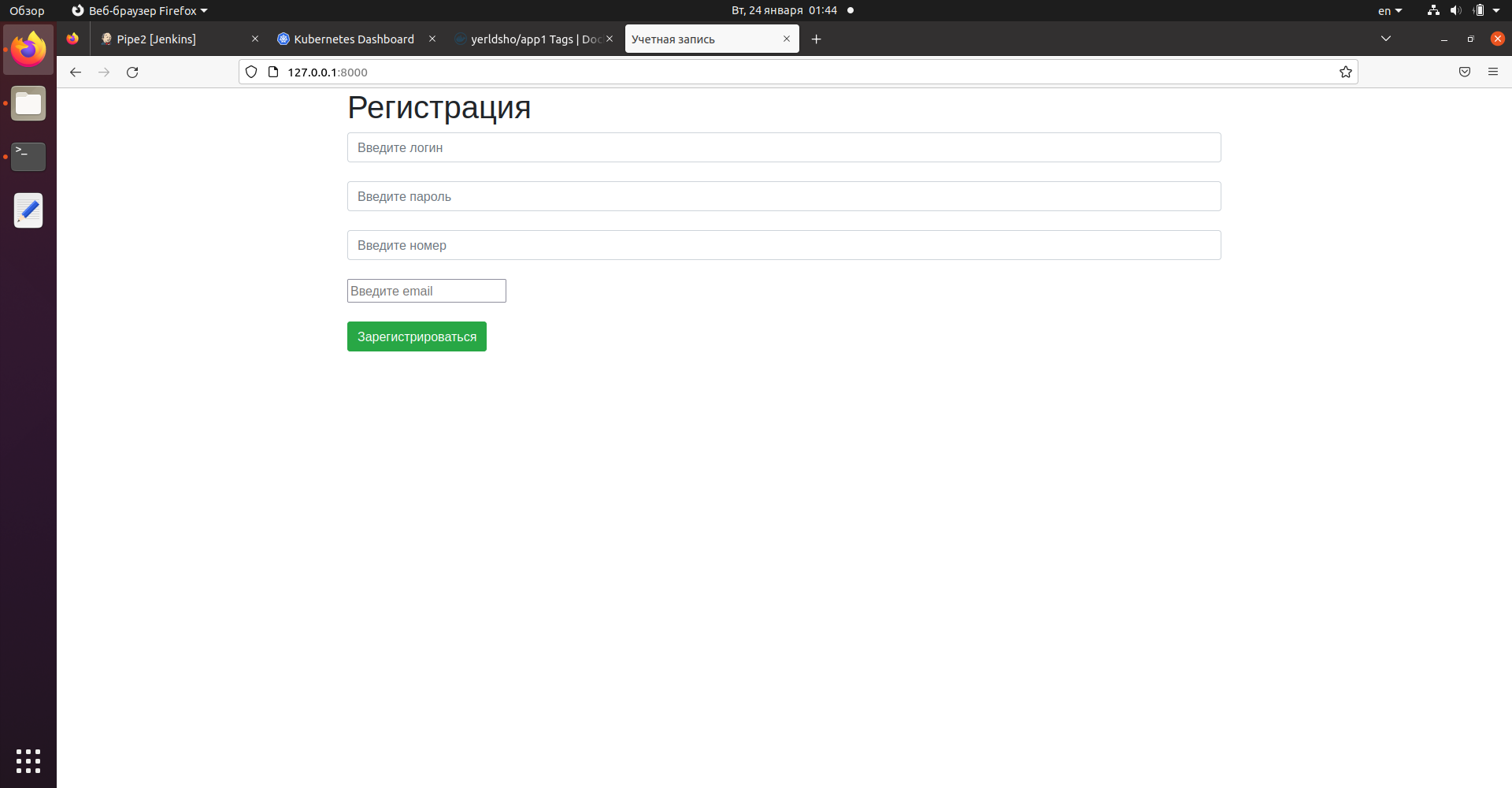












Итак, все работает и вместо 150 команд мы упростили тесты и автосборку до 4. Далее думаю можно реализовать скрипт для первых трех, добавить загрузочный бар, чтобы не заходить в Jenkins, настроить удаление подов перед сборкой и автоподачу проброса портов. Потраченных сил жалко, интересно, но того не стоит.

Оставлю ссылки на все это снизу.

**Вывод:** осуществили настройку CI/CD.

Источники:

<https://www.youtube.com/watch?v=XE_mAhxZpwU&t=228s>

<https://www.youtube.com/watch?v=YnZQJAMK6JI&t=968s>

<https://www.youtube.com/watch?v=q_nj340pkQo&list=PLg5SS_4L6LYvN1RqaVesof8KAf-02fJSi&index=1>

<https://www.youtube.com/watch?v=mnUvnbpbVOM>

<https://github.com/YerldSHO/git-pros4>

<https://habr.com/ru/post/519912/>

<https://kompose.io>

<https://stackoverflow.com/questions/47854463/docker-got-permission-denied-while-trying-to-connect-to-the-docker-daemon-socke>

<https://stackoverflow.com/questions/62232137/unable-to-read-client-cert-and-client-key-when-trying-to-run-kubectl-get-pods-wi/70451125#70451125>

https://www.youtube.com/watch?v=cyb10iplv7U&list=PLg5SS\_4L6LYvQbMrSuOjTL1HOiDhUE\_5a