

Отчёт: установка и настройка Tuleap, Jenkins, Kalithea.

Выполнил: Дороничев-Тедерсон Даниил Юрьевич (ИБТ-32).

1. В работе использованы:
 - Ubuntu Linux 20.04
 - Docker v19.03.13
 - Docker-compose v1.27.4
2. Файл docker-compose.yml.

```
version: '3.2'
services:
  jenkins:
    container_name: jenkins
    build:
      context: jenkins
      dockerfile: Dockerfile-alpine
      args:
        JENKINS_VERSION: 2.235
    ports:
      - 8080:8080
      - 50000:50000
    volumes:
      - /tmp:/var/jenkins_home/secrets/
  tuleap:
    container_name: tuleap
    build:
      context: tuleap
  kallithea:
    container_name: kallithea
    build:
      context: kallithea
    ports:
      - 3080:80
```

Рис. 1. Структура файла docker-compose.yml

На Рис. 1 можно видеть структуру файла docker-compose. На каждый из трёх компонентов приходится свой компонент services. Контекстом каждого компонента по сути является отдельная директория. Для Gitea и Openproject используются Docker-файлы представленные в master ветках репозитория. Для Jenkins в свою очередь используется модифицированный dockerfile на основе Alpine (модификация заключалась в изменении версии на более новую, а также удаление хэша проверки целостности SHA512, так как присутствующий хэш относился к старой версии Jenkins). Необходимость наличия директивы volumes в описании Jenkins обусловлена тем, что нам необходимо получить секретный одноразовый ключ необходимый при первом запуске всей инсталляции. Все порты проброшены согласно стандартной документации на каждый продукт, так имеем:

- Jenkins работает на порту 8080 (web) и 50000 (jenkins agent);
- Kallithea работает на порту 3080 (web);

– Tulear работает на порту 80 (web), но Tulear работает на специфичном домене tulear.local.

3. Скрипт резервного копирования.

```
#!/usr/bin/env bash

echo "Runing backup..."
for i in \
$(docker ps --filter "name=kallithea|jenkins|tulipe" | tail -n+2 | awk '{ print $1
→ }'); do docker export -o ${i}-ci-$(date +"%m-%d-%Y").tar ${i}; \
done
```

Рис. 2. Скрипт резервного копирования

Скрипт представлен на Рис.2. По сути, данный скрипт производит фильтрацию запущенных контейнеров по имени, и экспортирует запущенные экземпляры в tar архивы с датой и временем. Далее архивы можно загрузить, вновь используя команду `docker load -i <archive_name.tar>`.

4. Установка.

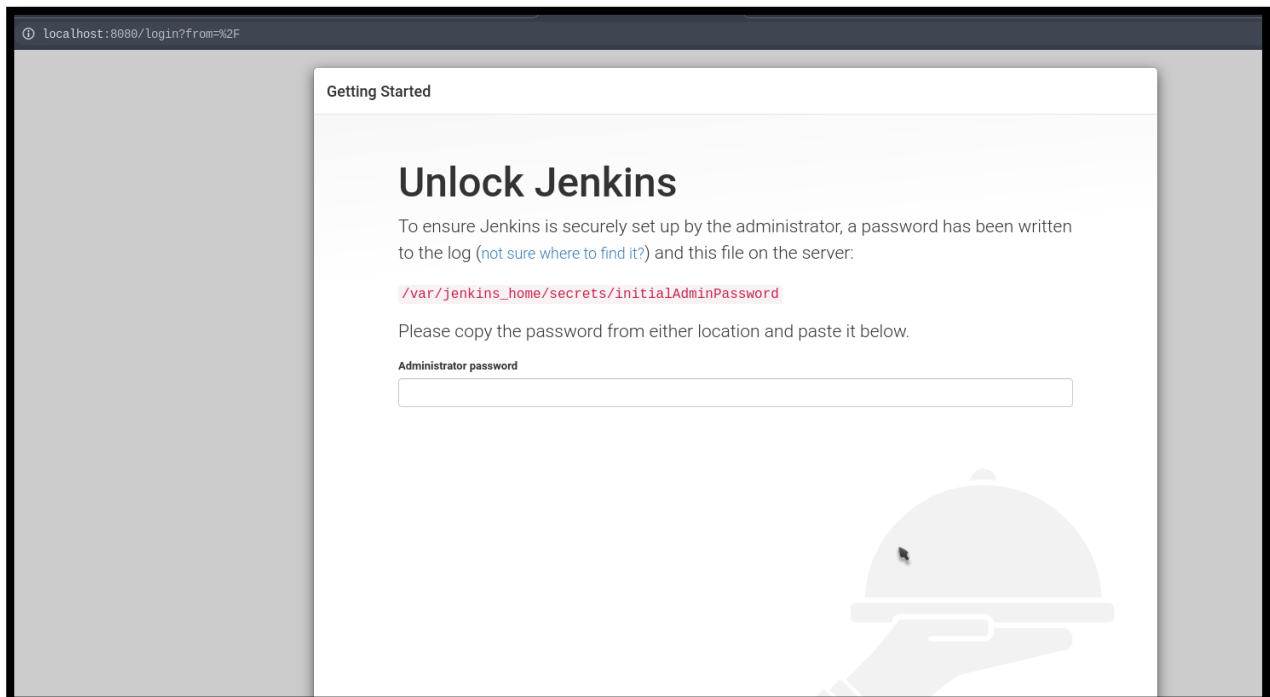


Рис. 3. Начальный экран Jenkins, с запросом уникального ключа.

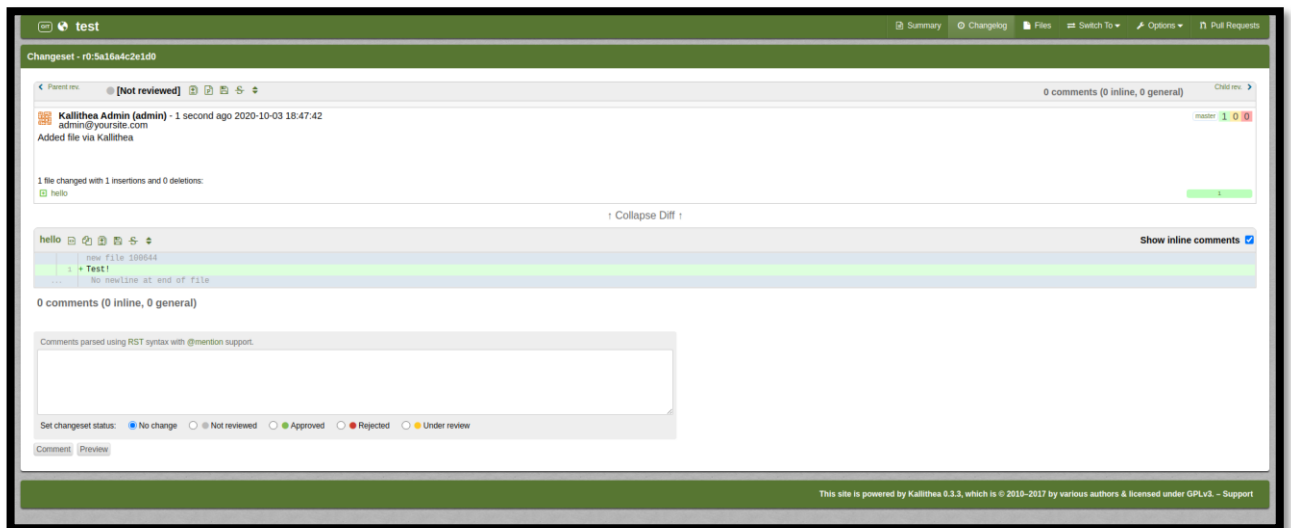


Рис. 4. Пример интерфейса Kallithea с коммитом.

Стоит сделать пояснение для уточнения. Для интеграции необходимо указать полный http-адрес до репозитория. Формально инсталляция Kallithea расположена по адресу localhost:8090. Однако данный адрес не является внутренним адресом подсети докера, для того-чтобы подключения прошло успешно необходим именно внутренний адрес. В данном случае он был выяснен путём запуска команды `docker inspect ${CONTAINER_ID} | grep IPAddr`, где `${CONTAINER_ID}` — идентификатор контейнера, в котором запущена инсталляция Kallithea. В качестве данных для авторизации используются данные пользователя `admin`. Тоже самое необходимо сделать для контейнера с инсталляцией Tulear, это необходимо для ассоциации домена `tulear.local` с реальным ip адресом контейнера путём редактирования файла `/etc/hosts` непосредственно на хост-машине.

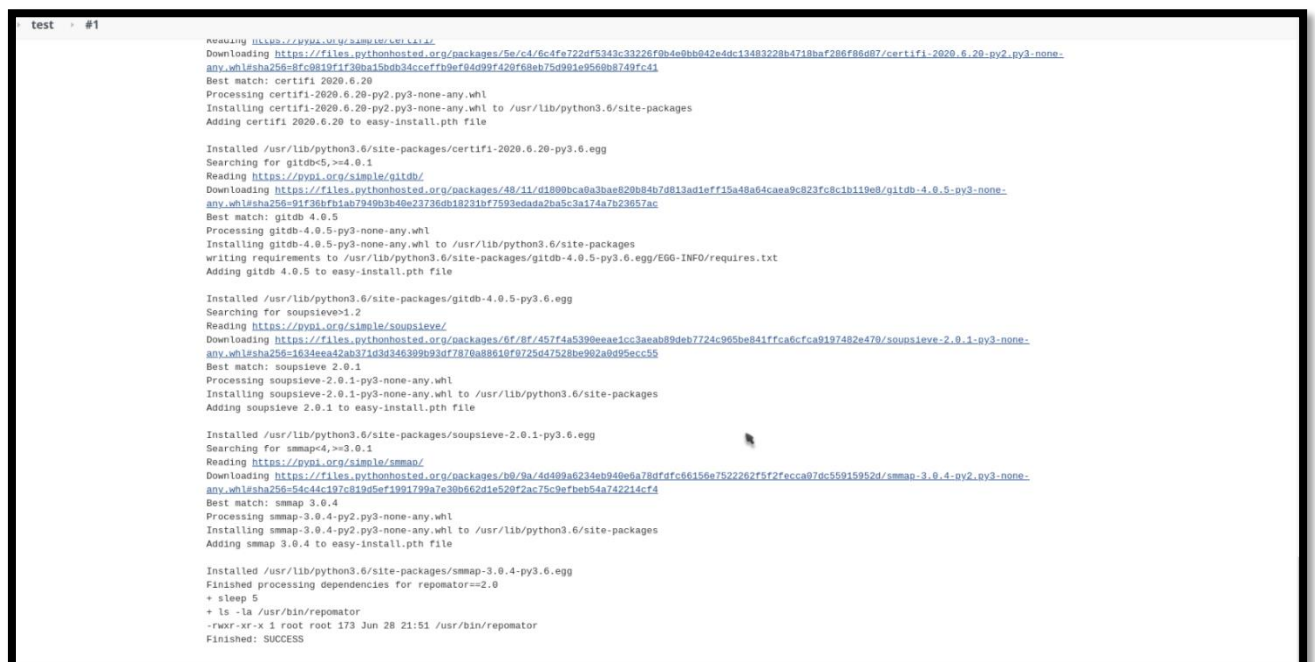


Рис. 5. Пример успешной работы проекта с помощью интеграции.

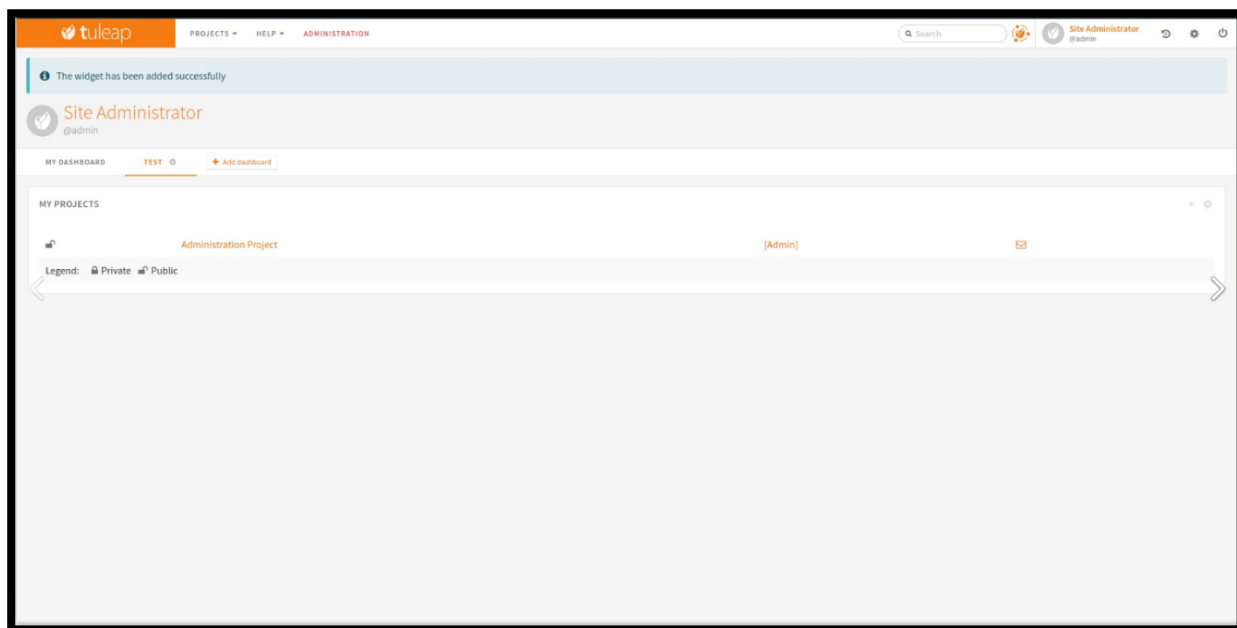


Рис. 6. Пример создания тикета в Tuleap.

К сожалению, согласно документации, (а также форумам поддержки всех трёх проектов), не существует интеграции Kallithea с Tuleap или Tuleap с Jenkins.