

Jenkins установка и настройка. Используем Docker-compose

Linux:

Docker

Если вы начинающий DevOps, эта статья именно для вас. Jenkins позволяет автоматизировать процессы модели CI/CD для крупных проектов или небольшого офиса. Знание данного инструмента, наряду с аналогичными продуктами, делает из вас специалиста DevOps близкого к профессиональной среде.



Jenkins



docker

Jenkins - это программная система с открытым исходным кодом на Java, предназначенная для обеспечения процесса непрерывной интеграции программного обеспечения по модели CI/CD.

За первоначальные условия примем, что у нас стоит: Nginx + PHP-FPM + MariaDB

Создадим директорию под проект и начнем заполнять docker-compose файл:

1. `mkdir` docker
2. `cd` docker
3. `nano` docker-compose.yml

Файл внутри содержит следующий код:

```
1. version: '3.6'
2. services:
3.   jenkins:
4.     image: "jenkins/jenkins:lts"
5.     container_name: "jenkins_lts"
6.     volumes:
7.       - ./jenkins_home:/var/jenkins_home
8.       - /var/run/docker.sock:/var/run/docker.sock:rw
9.       - /usr/bin/docker:/usr/bin/docker
10.    network_mode: host
11.    ports:
12.      - 8080:8080
13.    restart: always
```

Сохраняем. Создаем директорию для проброса "jenkins_home" из контейнера Docker:

```
1. mkdir jenkins_home
```

Получим такую структуру:

```
1. tree
2. .
3. |— docker
4. |   |— docker-compose.yml
5. |   |— jenkins_home
```

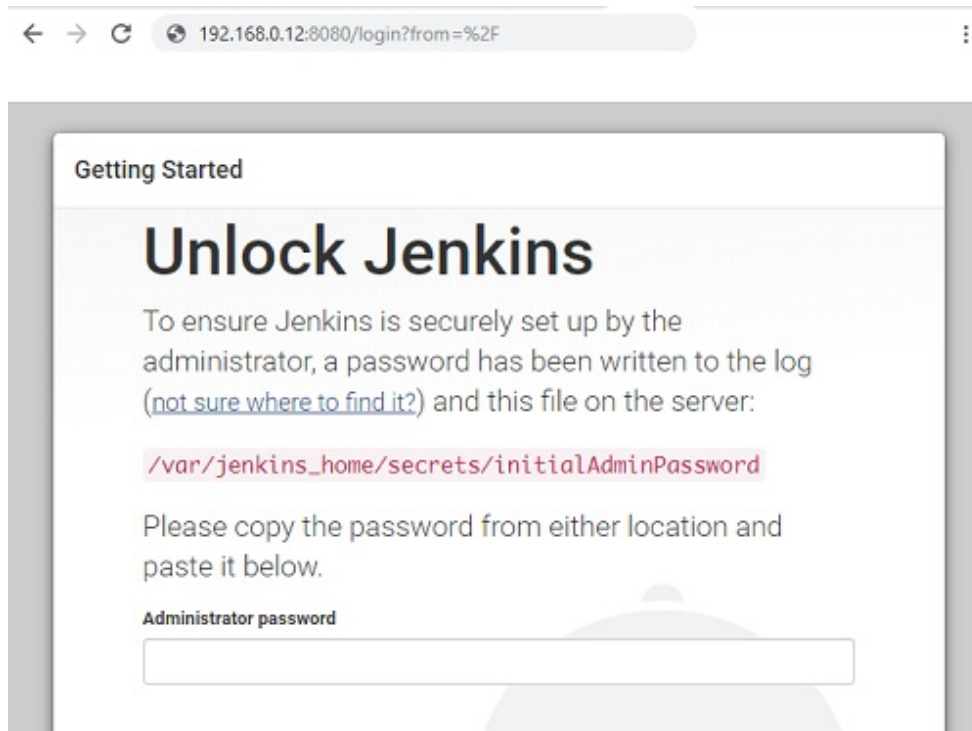
Из каталога docker запускаем наш файл:

```
1. docker-compose up -d
2. Pulling jenkins (jenkins/jenkins:lts)...
3. lts: Pulling from jenkins/jenkins
4. 092586df9206: Pulling fs layer
5. ef599477fae0: Pulling fs layer
6. 4530c6472b5d: Pulling fs layer
7. d34d61487075: Waiting
8. 272f46008219: Waiting
9. ...
10. ce3c0ae47fcb: Pull complete
11. Digest: sha256:bf741663dff2c5c163bd16706d4d1b9e6f86fa7b92e365257e170
12. Status: Downloaded newer image for jenkins/jenkins:lts
13. Creating jenkins_lts ... done
```

Проверяем:

```
1. docker images
2. REPOSITORY          TAG          IMAGE ID          CREATED
3. jenkins/jenkins     lts          5721a6cac43c     3 weeks
4. docker ps
5. CONTAINER ID        IMAGE          COMMAND           C
6. 9e169611c39e        jenkins/jenkins:lts  "/sbin/tini -- /usr/..."  7
```

Заходим на IP нашего сервера и указываем порт 8080:



Все работает. Jenkins просит указать пароль, расположенный по пути:

```
1. /var/jenkins_home/secrets/initialAdminPassword
```

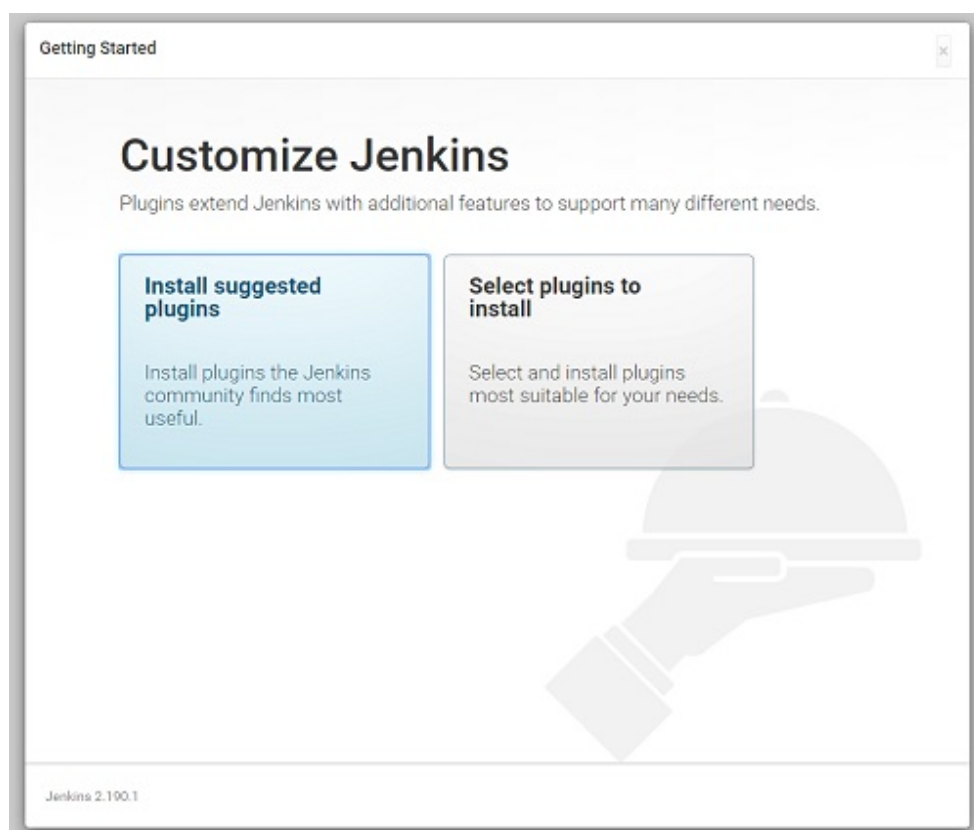
Так как это Docker, чтобы попасть по этому пути, надо зайти в движущийся контейнер:

```
1. sudo docker exec -it 9e169611c39e /bin/bash
2. jenkins@debi:/$
```

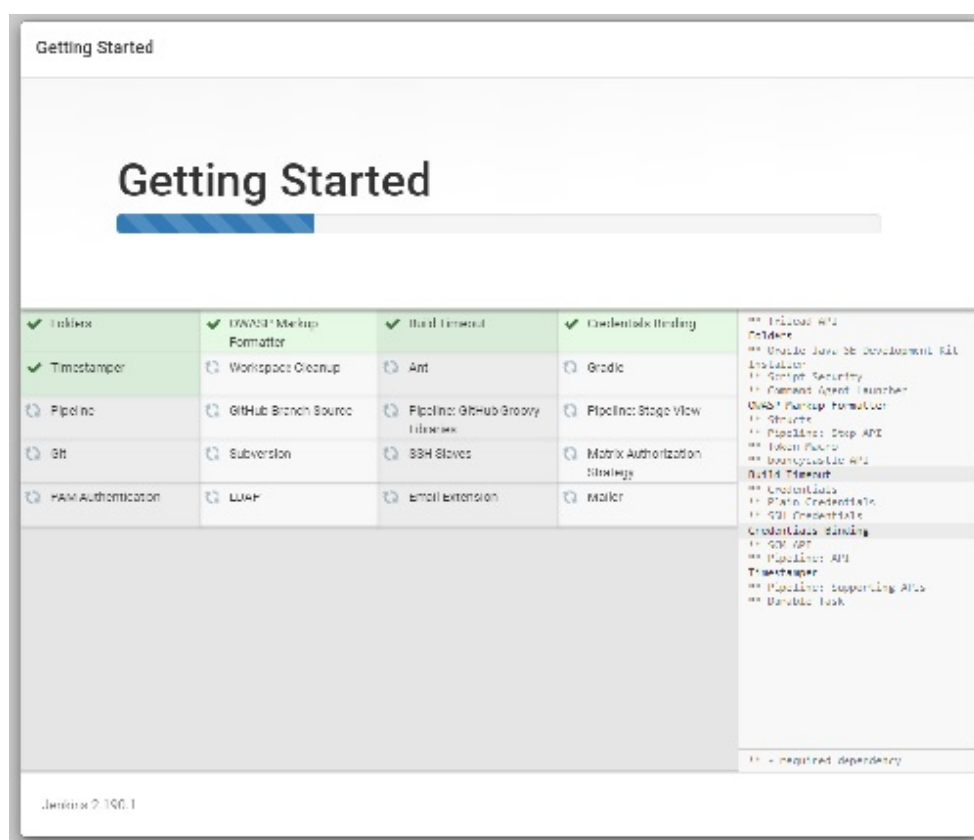
После входа в контейнер переходим по пути выше. Контейнер внутри ограничен, он собирается только для Jenkins: это значит, что там нет таких привычных приложений как nano и vim. Поэтому будем использовать команду вывода:

```
1. jenkins@debi:/$ cat < /var/jenkins_home/secrets/initialAdminPassword
2. 12e76a43aac14f76b7561df17f73fff6
3. # Выходим из контейнера
4. exit
```

После ввода пароля на странице с Jenkins, получаем:



Появляется выбор: поставить плагины по умолчанию, выбрать плагины самому. Мы выбрали по умолчанию:



После установки плагинов по умолчанию, Jenkins попросит указать данные пользователя:

Getting Started

Create First Admin User

Имя пользователя:

Пароль:

Повторите пароль:

Ф.И.О.:

Адрес электронной почты:

Jenkins 2.190.1 Continue as admin Save and Continue

На последнем этапе, попросит указать корневого URL-адрес. По умолчанию подхватит текущий. Если вы хотите его сменить, меняйте. Мы оставляем по умолчанию.

Getting Started

Instance Configuration

Jenkins URL:

The Jenkins URL is used to provide the root URL for absolute links to various Jenkins resources. That means this value is required for proper operation of many Jenkins features including email notifications, PR status updates, and the BUILD_URL environment variable provided to build steps.

The proposed default value shown is not saved yet and is generated from the current request, if possible. The best practice is to set this value to the URL that users are expected to use. This will avoid confusion when sharing or viewing links.

Jenkins is ready! Your Jenkins setup is complete.: Нажимаем "Start using Jenkins"

Поздравляем вас, вы подняли Jenkins из Docker!

Jenkins DevOps Linux SQL | [выход](#)

[Включить уведомление о статусе](#) [добавить описание](#)

- Создать Item
- Пользователи
- История сборки
- Настроить Jenkins
- My Views
- Lockable Resources
- Credentials
- New View

Добро пожаловать в Jenkins!

Пожалуйста, [создайте новую задачу](#) для того чтобы начать работу.

Очередь сборки ➡

Очередь сборки пуста

[Состояние сборки](#) ➡

- 1 В ожидании
- 2 В ожидании

Страница сгенерирована: 22.10.2019 15:19:00 UTC [REST API](#) [Jenkins ver. 2.190.1](#)

Проверяем нашу папку, которую мы пробросили из контейнера:

```
1. cd jenkins_home/
2. ls -la
3. итого 120
4. drwxr-xr-x 15 debiuser debiuser 4096 окт 22 20:18 .
5. drwxr-xr-x 3 debiuser debiuser 4096 окт 22 17:31 ..
6. -rw-r--r-- 1 debiuser debiuser 129 окт 22 18:08 atomic12517696541
7. -rw----- 1 debiuser debiuser 80 окт 22 18:08 .bash_history
8. drwxr-xr-x 3 debiuser debiuser 4096 окт 22 17:38 .cache
9. -rw-r--r-- 1 debiuser debiuser 1647 окт 22 20:18 config.xml
10. -rw-r--r-- 1 debiuser debiuser 100 окт 22 19:40 copy_reference_fi
11. drwxr-xr-x 3 debiuser debiuser 4096 окт 22 18:06 .groovy
12. -rw-r--r-- 1 debiuser debiuser 156 окт 22 19:41 hudson.model.Upda
13. -rw-r--r-- 1 debiuser debiuser 370 окт 22 18:06 hudson.plugins.gi
14. -rw----- 1 debiuser debiuser 1712 окт 22 17:38 identity.key.enc
15. drwxr-xr-x 3 debiuser debiuser 4096 окт 22 17:38 .java
16. -rw-r--r-- 1 debiuser debiuser 7 окт 22 20:18 jenkins.install.I
17. -rw-r--r-- 1 debiuser debiuser 7 окт 22 20:18 jenkins.install.U
18. -rw-r--r-- 1 debiuser debiuser 182 окт 22 20:17 jenkins.model.Jen
19. -rw-r--r-- 1 debiuser debiuser 171 окт 22 17:38 jenkins.telemetry
20. drwxr-xr-x 2 debiuser debiuser 4096 окт 22 17:38 jobs
21. -rw-r--r-- 1 debiuser debiuser 0 окт 22 19:41 .lastStarted
22. drwxr-xr-x 3 debiuser debiuser 4096 окт 22 17:38 logs
23. -rw-r--r-- 1 debiuser debiuser 907 окт 22 19:41 nodeMonitors.xml
24. drwxr-xr-x 2 debiuser debiuser 4096 окт 22 17:38 nodes
25. drwxr-xr-x 76 debiuser debiuser 12288 окт 22 18:06 plugins
26. -rw-r--r-- 1 debiuser debiuser 64 окт 22 17:38 secret.key
27. -rw-r--r-- 1 debiuser debiuser 0 окт 22 17:38 secret.key.not-so
28. drwx----- 4 debiuser debiuser 4096 окт 22 20:14 secrets
29. drwxr-xr-x 2 debiuser debiuser 4096 окт 22 18:06 updates
30. drwxr-xr-x 2 debiuser debiuser 4096 окт 22 17:38 userContent
31. drwxr-xr-x 3 debiuser debiuser 4096 окт 22 20:14 users
32. drwxr-xr-x 11 debiuser debiuser 4096 окт 22 17:38 war
33. drwxr-xr-x 2 debiuser debiuser 4096 окт 22 18:06 workflow-libs
```

Для чего нужно было делать проброс? Если контейнер упадет, или перезапустится, по умолчанию он возвращается в первоначальное состояние. Но так как папки с настройками мы храним не внутри контейнера, а на своей рабочей машине, данные подтянутся обратно. Всем спасибо кто читал! Дальше будет обзор Jenkins и DevOps работа на его базе.