**Использование отечественных облачных инфраструктур для развертывания контейнеров Docker.**

**Симбирцев Е.А.**  
студент  
Российский государственный университет нефти и газа имени И.М. Губкина  
(г. Москва, Россия)  
  
**Галяутдинов Д.А.**  
студент  
Российский государственный университет нефти и газа имени И.М. Губкина   
(г. Москва, Россия)

**Аннотация.** В статье анализируется использование российских облачных платформ (Cloud.ru, VK Cloud) для Docker-контейнеров. Эксперимент с Cloud.ru демонстрирует развертывание веб-сервера nginx на ALT Linux. Подчеркивается удобство и эффективность отечественных решений для контейнеризации.

**Abstract.** This article analyzes the use of Russian cloud platforms (Cloud.ru, VK Cloud) for Docker containers. An experiment with Cloud.ru demonstrates the deployment of an nginx web server on ALT Linux. The convenience and efficiency of domestic solutions for containerization are highlighted.

**Ключевые слова.** Docker, контейнеризация, облачные инфраструктуры, Cloud.ru Container Apps, VK Cloud Containers, российские облачные сервисы, ALT Linux, веб-сервер nginx, развертывание контейнеров, информационная безопасность, эксперимент.

**Keywords.** Docker, containerization, cloud infrastructures, Cloud.ru Container Apps, VK Cloud Containers, Russian cloud services, ALT Linux, nginx web server, container deployment, information security, experiment.

# Введение

Контейнеризация приложений быстро становится стандартом де-факто в разработке и развертывании программного обеспечения. Ее преимущества – повышенная переносимость, изоляция, эффективность использования ресурсов – привели к широкому распространению решений, наиболее известным из которых является Docker. Так же в небольших компаниях растёт необходимость в облачных сервисах для развертывания контейнеров для минимизации использования своих ресурсов. Эту проблему решают облачные инфраструктуры. Однако, в свете геополитических событий и растущей озабоченности вопросами информационной безопасности, возросла актуальность поиска и оценки отечественных решений. Выбор подходящей платформы для контейнеризации напрямую влияет на производительность и безопасность приложения, поэтому объективное сравнение различных решений становится критически важным.

В настоящей статье будут рассмотрены отечественные инфраструктуры для разворачивания контейнеров. А также проведена экспериментальная часть с одним из таких облачных сервисов с целью проверки удобства и работоспособности таких решений.

# Обзор существующих решений

Для начала необходимо разобрать, что такое Docker. Это программа для создания и запуска приложений в рамках контейнера. Он используется для оптимизации работы над разработкой или поддержкой приложений. Представим ситуацию, что на работу в IT-компанию пришел стажер. Старший разработчик дал ему задание найти и исправить небольшой баг. Но для того, чтобы это сделать, стажеру будет необходимо настроить web-сервер, скачать и развернуть все фреймворки, которые использует данный сайт, прежде чем запустить и начать исправление ошибки. В итоге он потратит намного больше времени на настройку среды для корректной работы сайта, чем на исправление бага. Эту проблему и решает docker, благодаря ему можно создать образ машины с уже настроенным web-сервером, а также скачанными и подключенными фреймворками.

В данной статье будут рассмотренны облачные инфраструктуры хранения образов и разворачивания контейнеров отечественного производства. Данные сервисы необходимы компаниям с ограниченными техническими ресурсами, чтобы самим поддерживать все контейнеры. Облачные сервисы выступают в роли тонких клиентов и используют только свои ресурсы для поддержания контейнеров. Ниже рассмотрены отечественные сервисы:

1. Cloud.ru Container Apps: сервис для запуска контейнерных приложений в облаке, без знаний Kubernetes и создания виртуальных машин. Контейнеры запускаются на базе Docker-образов, загруженных в репозиторий сервиса Artifact Registry. Контейнеры могут использовать любую среду выполнения и язык программирования. Развертывание контейнера каждый раз при загрузке новой версии Docker-образа в Artifact Registry. Сервис автоматически создает и удаляет экземпляры контейнера в соответствии с нагрузкой. Поддержка масштабирования до нуля экземпляров контейнеров.

2. VK Cloud Containers: это сервис по управлению кластерами Kubernetes от VK Cloud, в котором можно создавать приватные репозитории для хранения Docker-образов. Он управляет жизненным циклом контейнеров и автоматически масштабируется под изменение нагрузки. Дополнительный сервис (аддон) Docker Registry работает в отказоустойчивой конфигурации.

В таблице ниже приведена сравнительная характеристика двух сервисов, рассмотрены только свойства, характеристики которых в сервисах различается.

***Таблица 1.***

**Сравнительная характеристика Cloud.ru и VK Cloud.**

|  |  |  |
| --- | --- | --- |
| **Характеристика** | **Cloud.ru Container Apps** | **VK Cloud Containers** |
| **Управление доступом** | Гибкое, возможна детальная настройка | Стандартное |
| **Защита от уязвимостей** | Регулярные проверки, потенциально менее быстрое устранение проблем. | Регулярные проверки, оперативное устранение проблем. |
| **Контроль целостности** | Более развитые средства проверки цифровых подписей и хэшей | Стандартные механизмы |
| **Бесплатный контент** | Есть бесплатные возможности | Только платное |

В экспериментальной части будет рассмотрен сервис Cloud.ru Container Apps, так как он предоставляет возможность использовать некоторые свои функции бесплатно.

**Экспериментальная часть**

**Методика исследования:** необходимо создать образ docker с запущенным веб сервером nginx на ОС ALT Linux, проверить корректность работы образа, изучить что происходит с ОС при создании образа. Затем подключить локальный docker к сервису Cloud.ru Container Apps, отправить готовый образ в Artifact Registry и на основе этого образа создать и запустить контейнер в сервисе. В конечном итоге результат работы на локальном контейнере должен совпасть с работой облачного контейнера. Регистрация на платформе пропускается.

1) Первым делом необходимо скачать docker на локальную ОС ALT Linux.

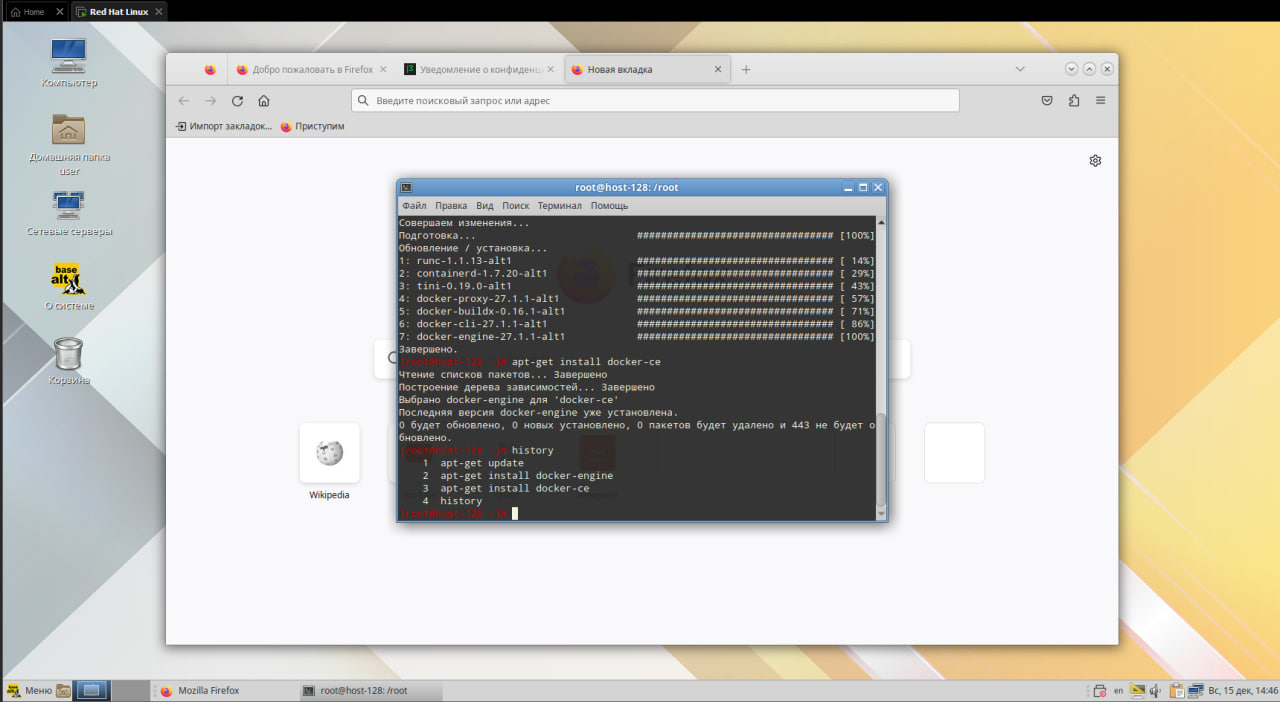
Для этого необходимо обновить дерево пакетов с помощью команды:

#apt-get update

Установить необходимые пакеты с docker:

#apt-get install docker-engine

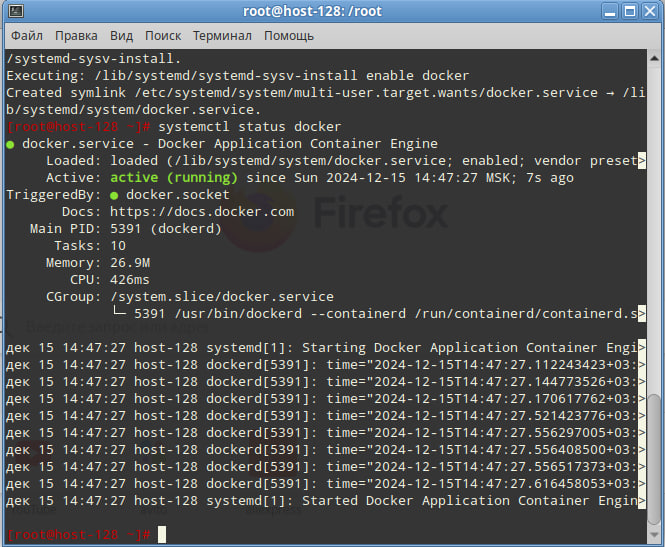
#apt-get install docker-ce



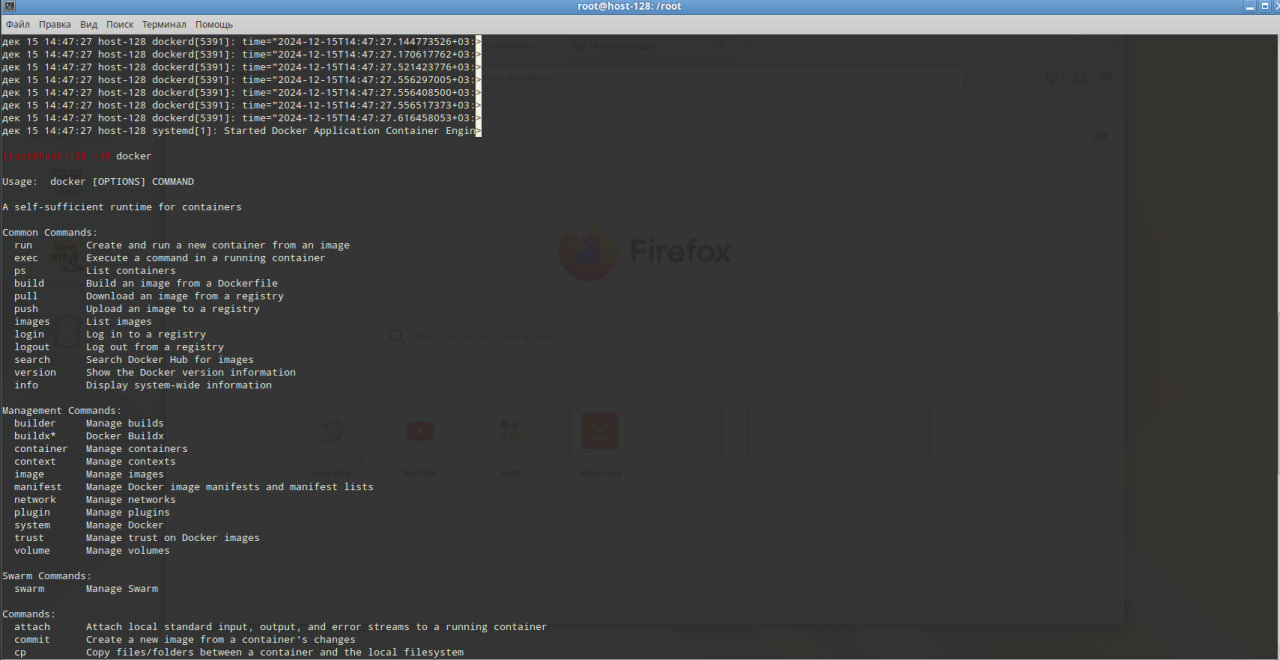
2) После установки необходимо запустить docker и проверить статус его работы:

# systemctl enable --now docker

# systemctl status docker

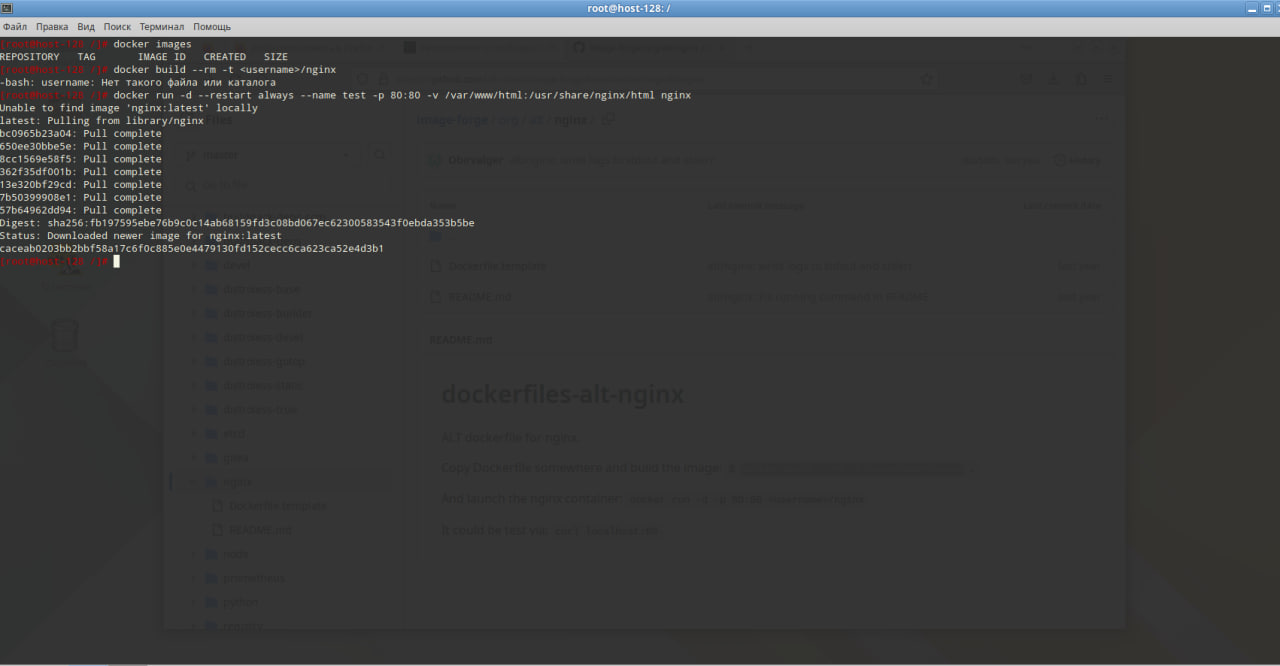


3) С помощью команды докер можно ознакомиться с функционалом данной программы:



4) Далее переходим к созданию контейнера с названием test с веб-сервером nginx, который будет слушать 80-ый порт.

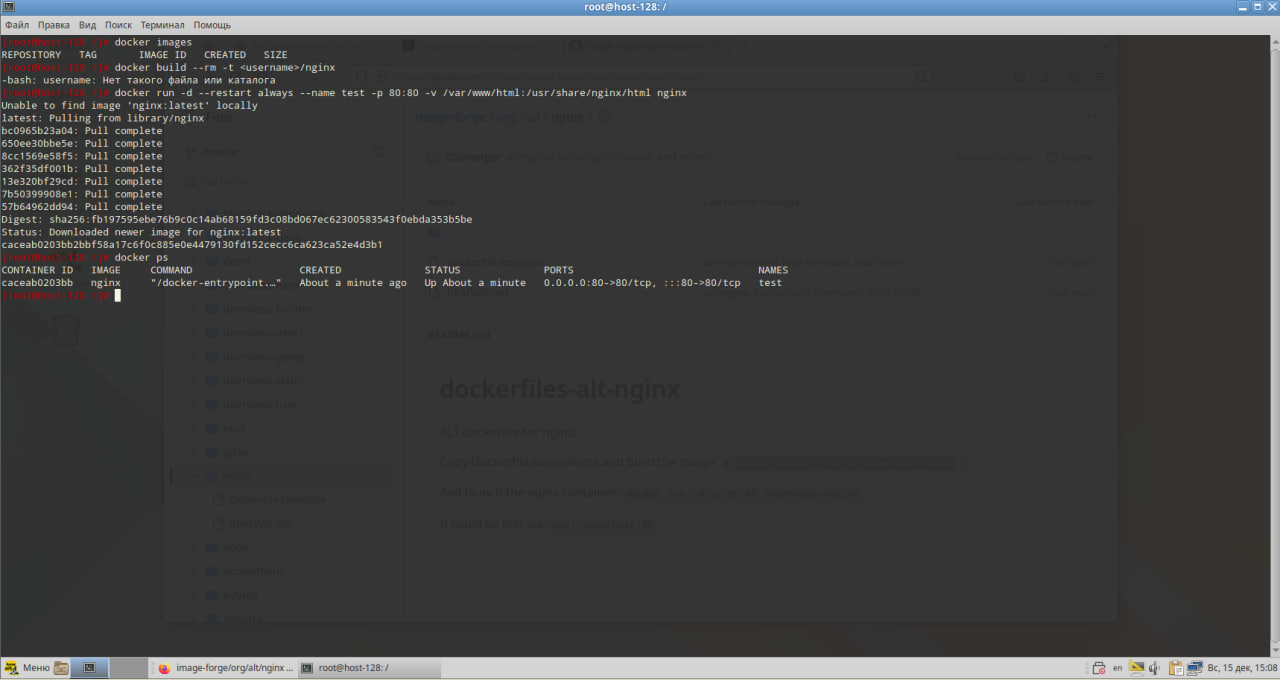
#docker run -d –restart always –name test -p 80:80 -v /var/www/html:/usr/share/nginx/html nginx



После этой команды nginx попытается найти изображение с nginx локально, не найдя его, начнет загрузку с серверов.

5) Далее проверим, что контейнер был создан с помощью команды:

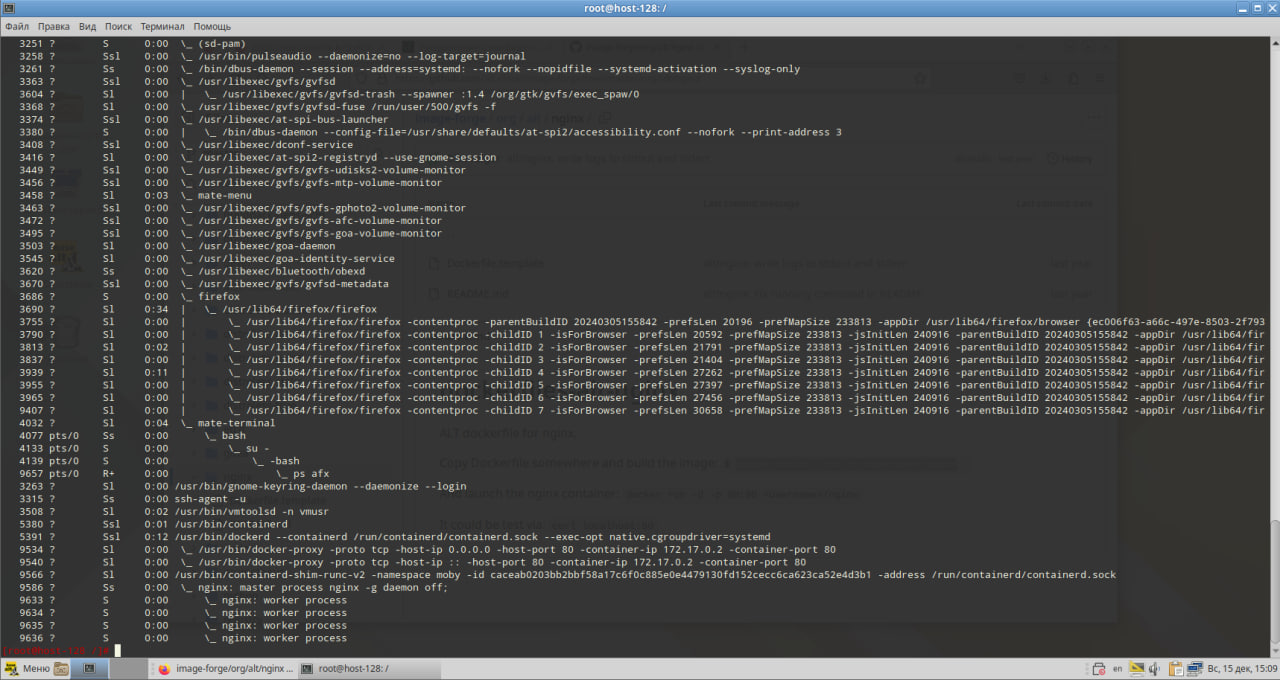
#docker ps



Исходя из результата можно сделать вывод о том, что контейнер создался и находится в статусе up

6) Далее можно посмотреть какие процессы запустил docker в системе:

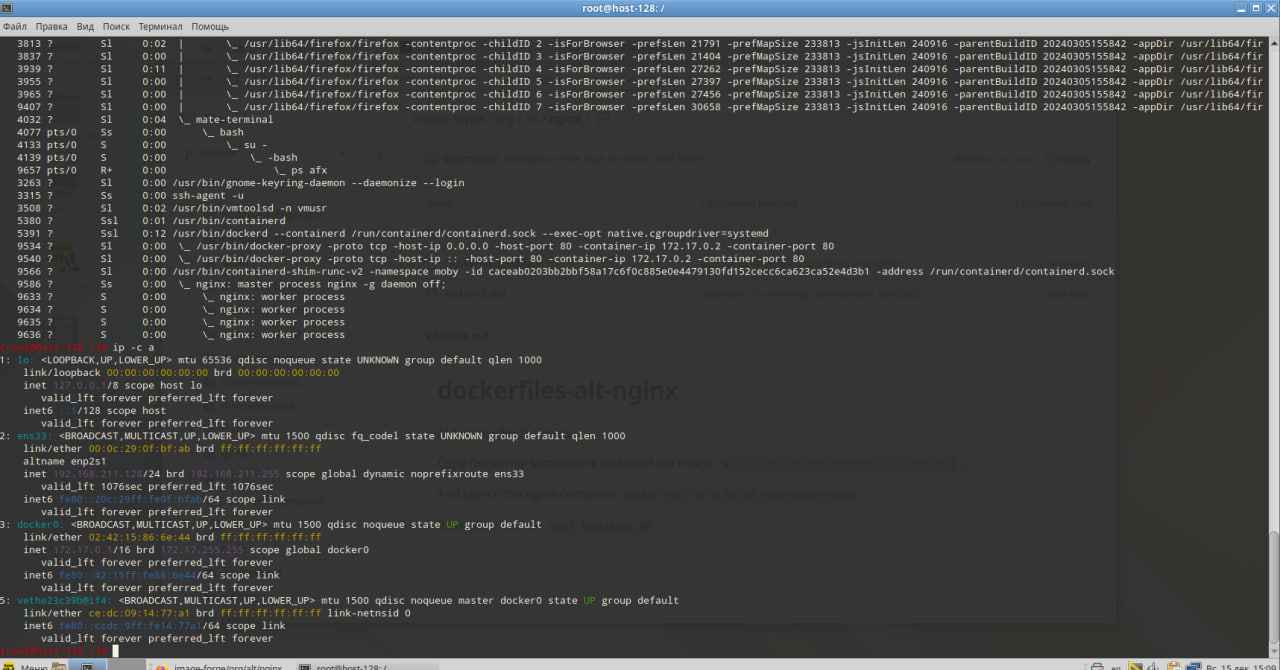
#ps fx



Как видно из снимка docker начал перехват всех запросов с порта 80 на контейнер

7) Далее можно вывести интерфейсы и подключенные к ним адреса:

#ip -c a

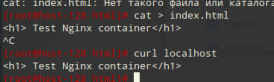


Можно заметить, что для docker создался отдельный интерфейс docker0

8) Далее создадим в директории /var/www/html файлик index.html, который nginx сможет прочесть и проверим что вернется после запроса к localhost:

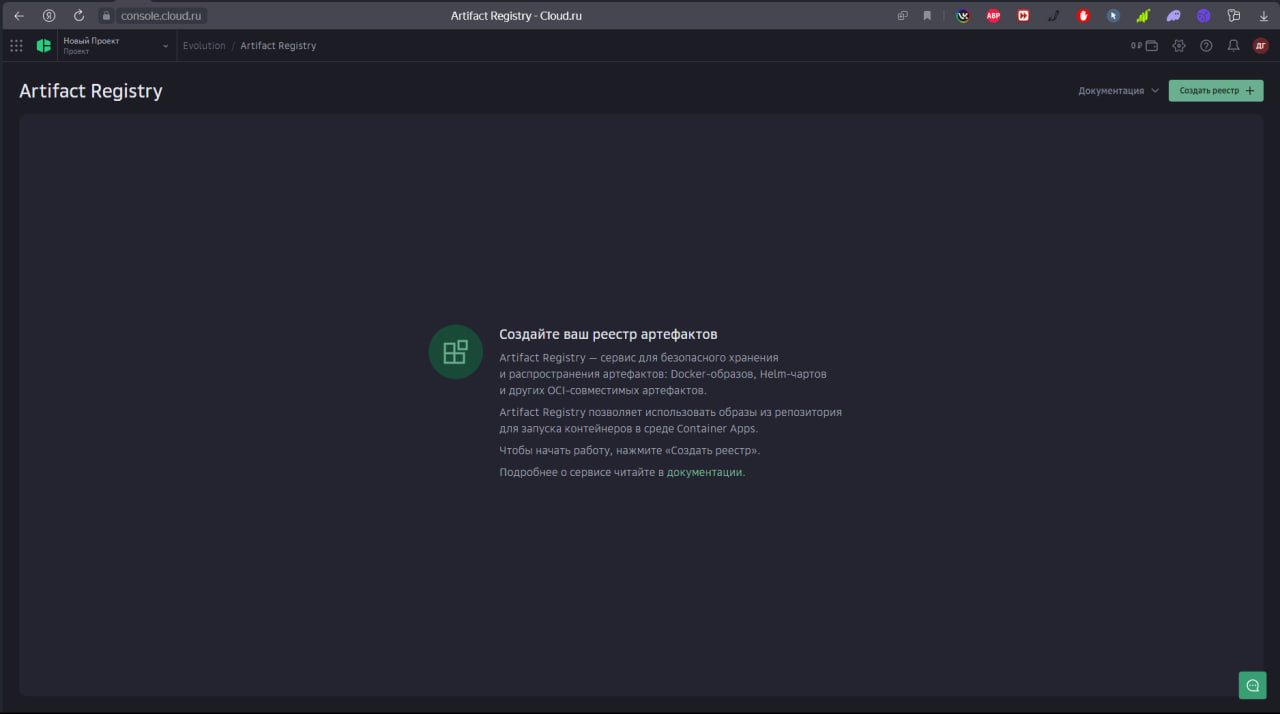
#cat > index.html

#curl localhost

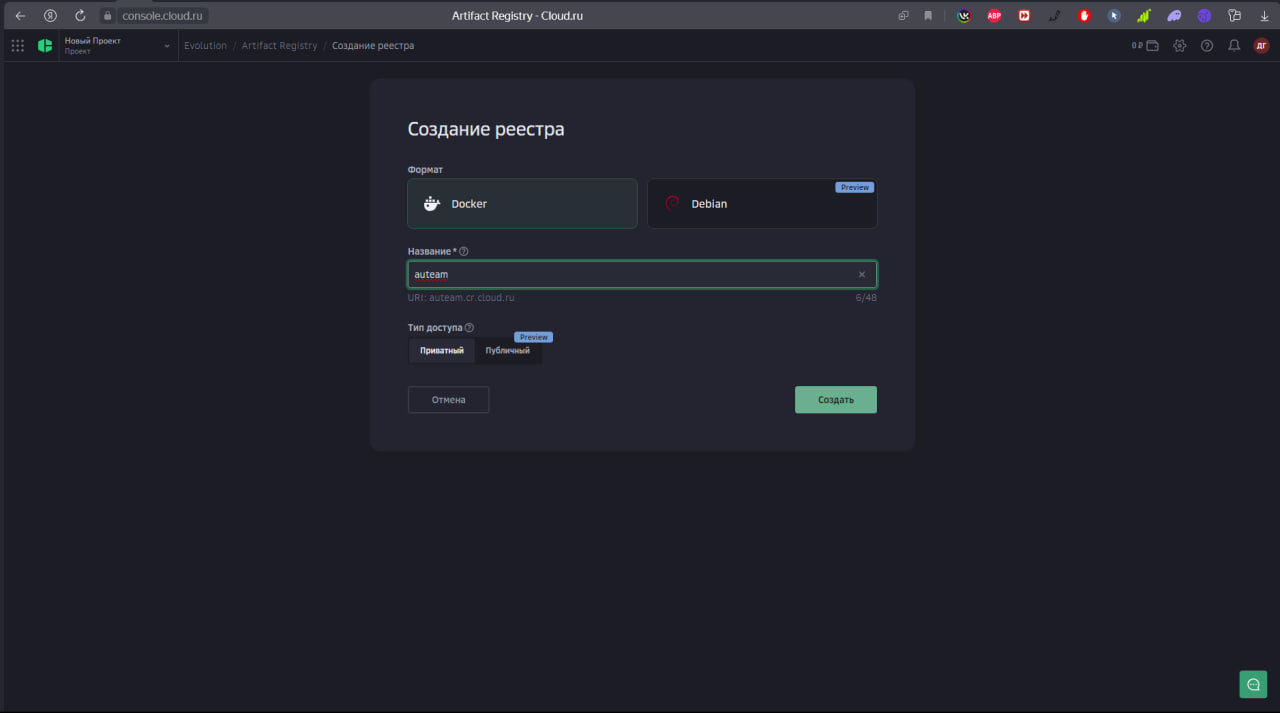


Теперь можно сделать вывод, что контейнер настроен корректно

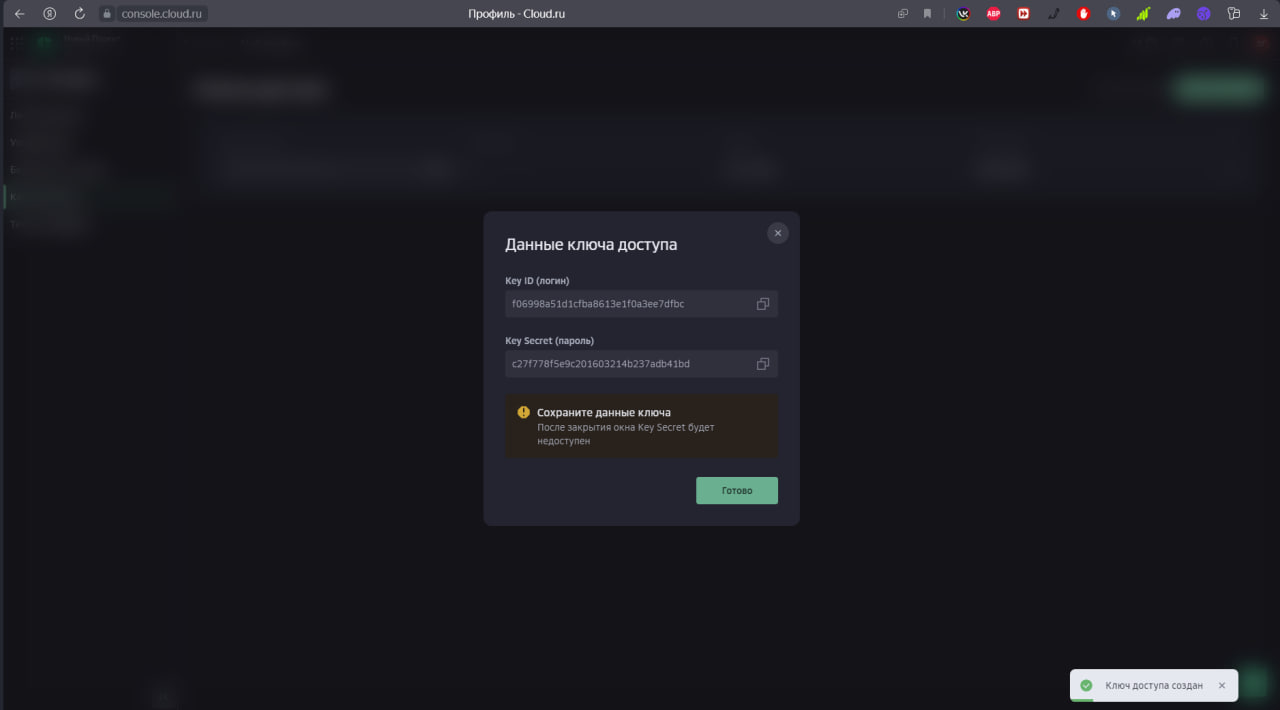
9) После настройки контейнера, необходимо перейти в сервис и выбрать Artifact Registry



10) В этой вкладке необходимо создать реестр куда будут заливаться образы контейнеров

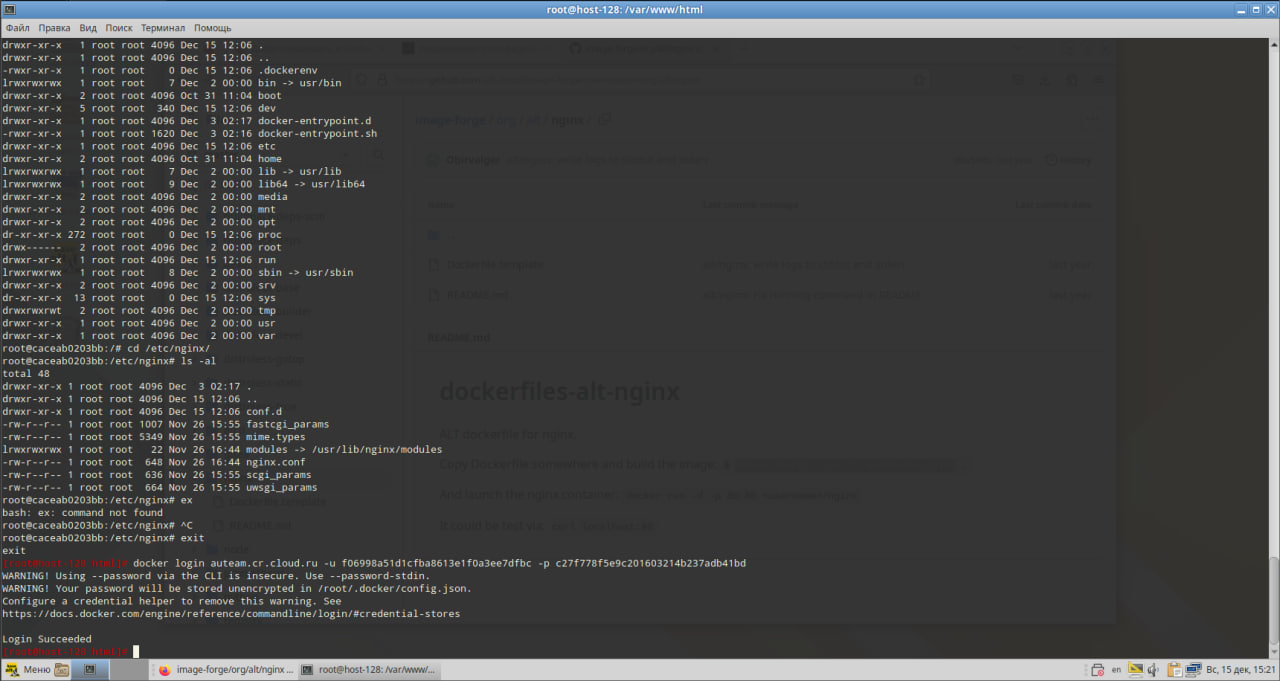


11) После создания реестра необходимо получить ключи доступа в профиле



12) Затем необходимо вернуться в командную строку и через docker залогиниться в реестре:

#docker login auteam.cr.cloud.ru -u <ключ логин> -p <ключ пароль>

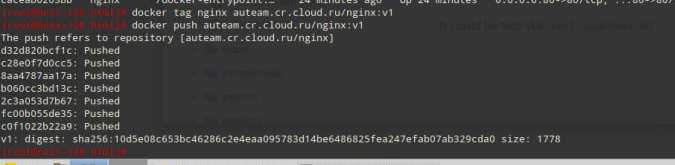


По выполнению этой команды можно увидеть, что авторизация прошла успешно

13) Теперь нужно выдать нашему образу тэг и залить его в реестр:

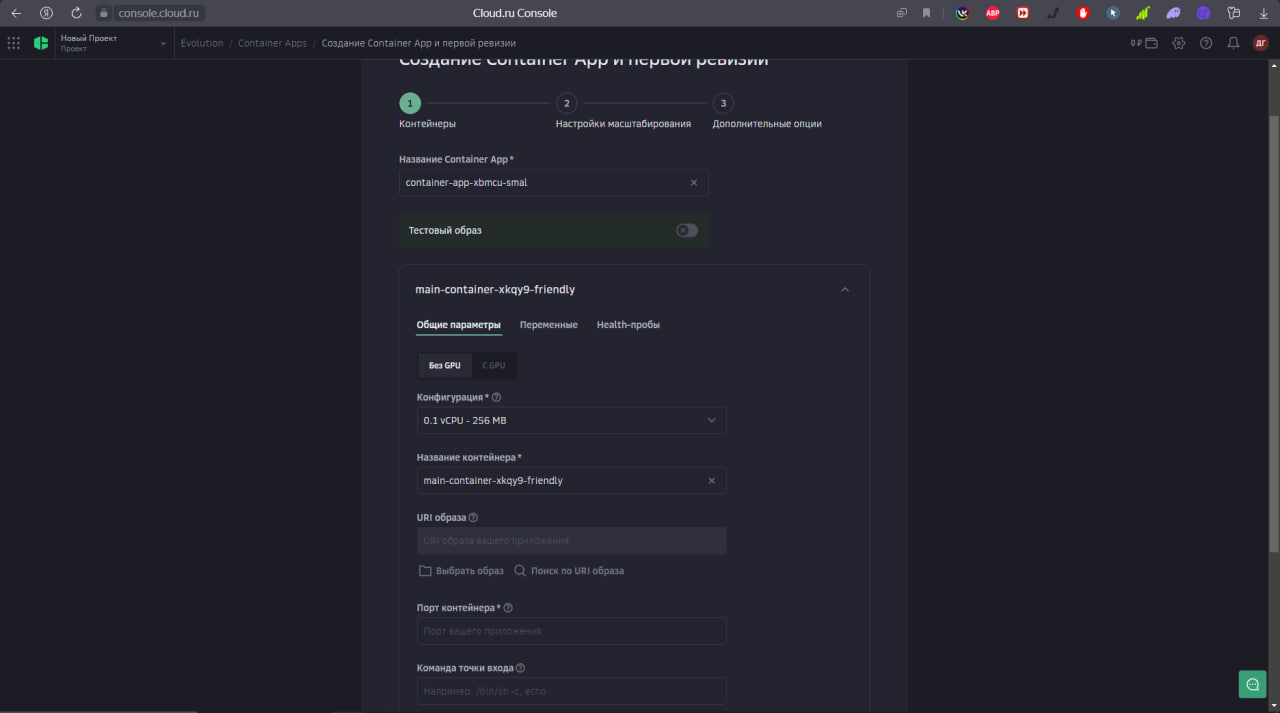
#docker tag nginx auteam.cr.cloud.ru/nginx:v1

#docker push auteam.cr.cloud.ru/nginx:v1

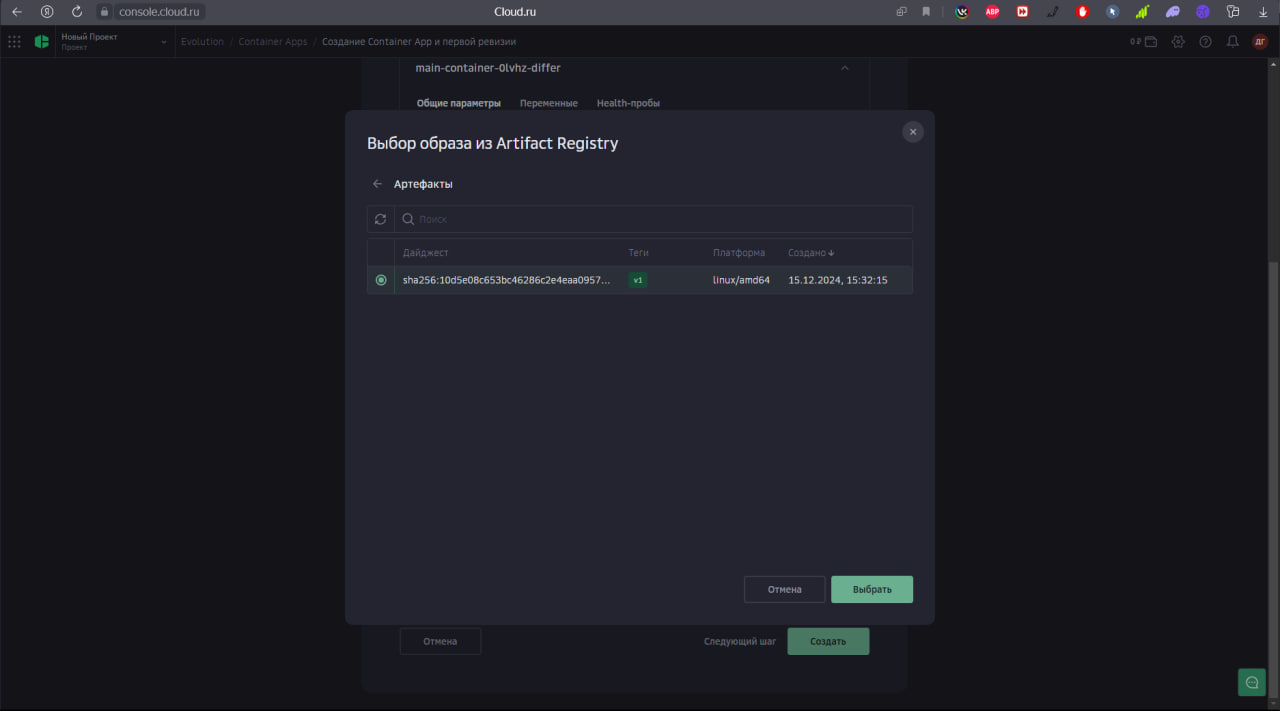


Как видно на снимке образ был успешно залит на сервер

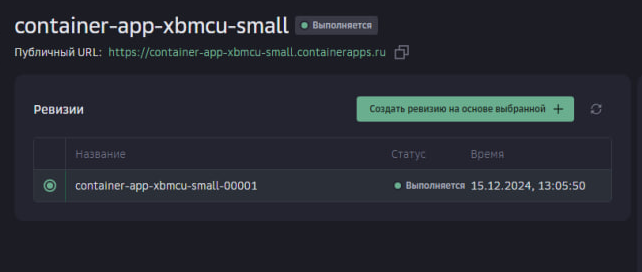
14) Переход к созданию контейнера



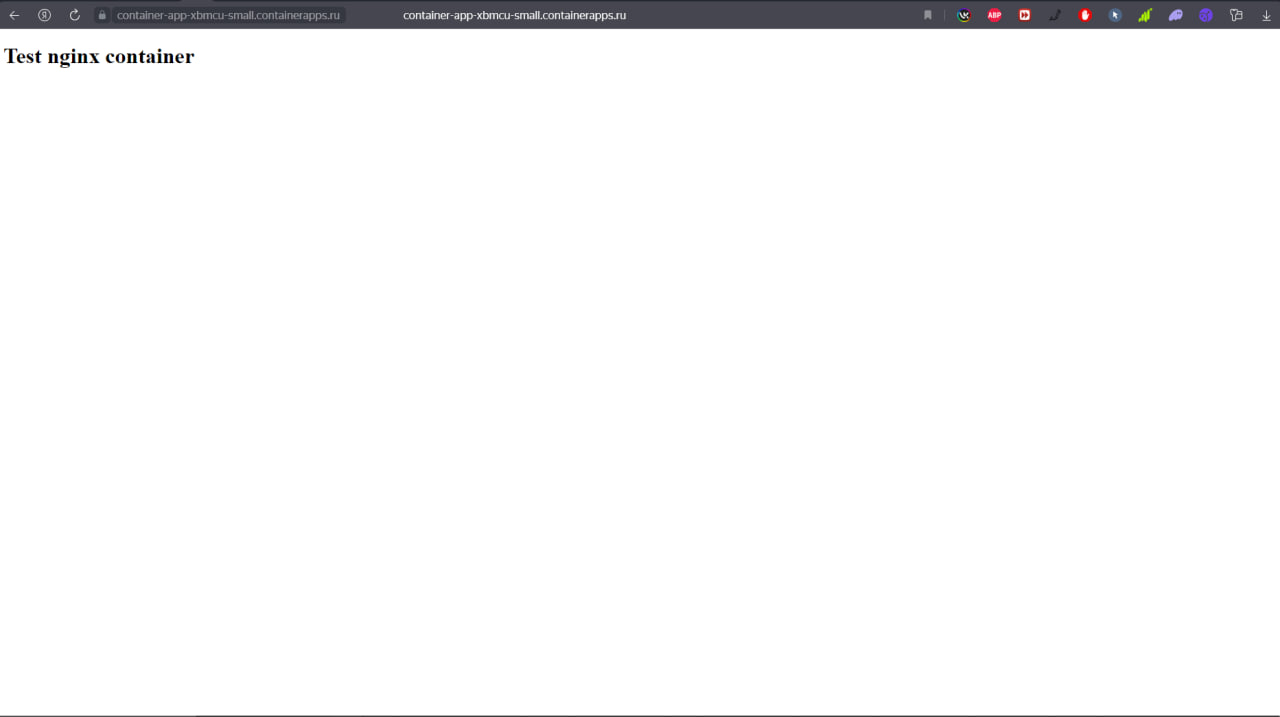
15) В качестве образа необходимо выбрать наш образ



16) После этого нужно дождаться пока контейнер перейдет в статус Выполняется и можно будет перейти по ссылке:



17) После перехода по ссылке видим ту же запись, которую видели на локальной машине:



По итогу эксперимента можно сделать вывод, что интерфейс данного сервиса довольно прост и понятен. Ниже представлена сравнительная характеристика локального контейнера и удаленного.

***Таблица 2.***

**Сравнительна характеристика локального контейнера и облачного.**

|  |  |  |
| --- | --- | --- |
| **Характеристика** | **Локальный контейнер** | **Облачный контейнер** |
| Загрузка | 1 секунда | 5 секунд |
| Нагрузка на процессор | 10% | Нагрузка идет на удаленный сервер |
| Использование памяти | 100Мбайт | Используется память сервера |
| Логирование | Локальное хранение логов | Централизованное логирование в Cloud.ru |
| Контроль целостности | Ручной контроль с помощью проверки хэшей образов | Автоматическая проверка целостности образов при загрузке из registry Cloud.ru |
| Защита от уязвимостей | Ручное обновление Docker, образов и хост-системы, ручное сканирование | Сканирование образов на уязвимости с помощью сервисов Cloud.ru |

**Заключение.**

В заключении можно сказать, что российские облачные сервисы для хранения и развертывания контейнерных приложений активно развиваются. И уже являются серьёзными конкурентами для зарубежных разработок. Выбор сервиса должен зависеть от требований проекта и выгодных предложениях.

**Список использованных источников**

1. Docker на ALT Linux [сайт]. URL: <https://www.altlinux.org/Docker>

2. Аналоги Docker Hub в России [сайт]. URL: <https://trends.rbc.ru/trends/industry/6659ad199a7947d621b9e7c9>

3. Документация по docker [сайт]. URL: <https://docs.docker.com/>

4. Евстратов, В. В. Контейнеризация как современный способ виртуализации / В. В. Евстратов. — Текст : непосредственный // Молодой ученый. — 2020. — № 49 (339). — С. 7-9. — URL: <https://moluch.ru/archive/339/76072/>

5. Контейнеризация приложений: что это такое и когда стоит использовать [сайт]. URL: <https://cloud.vk.com/blog/chto-takoe-kontejnerizacija-prilozhenij/>

6. Погружение в контейнеризацию [сайт]. URL: <https://habr.com/ru/companies/otus/articles/767884/>

7. ПРЕИМУЩЕСТВА КОНТЕЙНЕРИЗАЦИИ И СЕРВИСНО-ОРИЕНТИРОВАННОЙ АР-ХИТЕКТУРЫ ПРИ РАЗРАБОТКЕ И РАЗВЕРТЫВАНИИ ВЕБ-ПРИЛОЖЕНИЯ [сайт]. URL: <https://cyberleninka.ru/article/n/preimuschestva-konteynerizatsii-i-servisno-orientirovannoy-ar-hitektury-pri-razrabotke-i-razvertyvanii-veb-prilozheniya>

8. Уймин, А. Г. Обзор систем моделирования: анализ эффективности на примере чемпионата AtomSkills-2023 / А. Г. Уймин, В. С. Греков // Автоматизация и информатизация ТЭК. – 2023. – № 11(604). – С. 25-34. – DOI 10.33285/2782-604X-2023-11(604)-25-34. – EDN QYQRCO.