

Linux: права доступа и docker

НАШИ ПРАВИЛА



Включенная камера



Вопросы по поднятой руке



Не перебиваем друг друга



Все вопросы, не связанные с тематикой курса (орг-вопросы и т. д.), должны быть направлены куратору



Подготовьте свое рабочее окружение для возможной демонстрации экрана (закройте лишние соцсети и прочие приложения)

ЦЕЛЬ

Настроить среду для работы с Linux (запуск виртуальной машины). Изучить основные команды взаимодействия с системой Linux для терминала

ПЛАН ЗАНЯТИЯ

Пользователи в linux и права доступа к файлам и директориям

Docker: основы и установка

Docker: использование

Docker compose

Практика

Пользователи в Linux

В Linux как и в windows можно создавать несколько пользователей

Создание нового пользователя:

```
sudo adduser username
```

Удаление пользователя:

```
sudo deluser --remove-home username
```

Группы в Linux

Группы это объединение пользователей

Создание новой группы: `sudo addgroup groupname`

Добавление пользователя в группу: `sudo usermod -aG groupname username`

Просмотр групп пользователя: `groups username`

Удаление пользователя из группы: `sudo deluser username groupname`

Права доступа в Linux

Каждый файл и директория в Linux имеют три набора прав доступа: для владельца, группы и всех остальных пользователей. Эти права определяют, кто может читать, писать и выполнять файлы (просматривать директорию в случае директорий).

Права доступа в Linux

Просмотр прав доступа: `ls -l`

d или -
показывает
директория
это или файл

d	rwxr-xr-x	1	username	groupname	480	Jul 19 03:35	src
-	rw-r--r--	1	username	groupname	510484	Jul 19 00:00	yarn.lock

Группа владеющая файлом

Владелец файла

Права доступа для всех остальных

Права доступа для группы

Права доступа для владельца

Изменение прав доступа в Linux

Команда `chmod` изменяет права доступа.

Изменение прав доступа: `chmod u+rw, g+rx, o+r filename`

- `u+rw`: добавляет права чтения, записи и выполнения для владельца (user)
- `g+rx`: добавляет права чтения и выполнения для группы (group)
- `o+r`: добавляет право чтения для остальных пользователей (others)

Установка прав доступа с помощью числовых кодов: `chmod 755 filename`

Числовые коды соответствуют правам доступа:

- 7: чтение, запись, выполнение (rwx) (user)
- 5: чтение, выполнение (rx) (group)
- 4: чтение (r) (others)

Одновременное изменение владельца и группы: `sudo chown newowner:newgroup filename`

sudo

`sudo` (superuser do) — это команда, которая позволяет обычным пользователям выполнять команды от имени суперпользователя (`root`). Это необходимо для выполнения административных задач, таких как установка программного обеспечения, изменение системных настроек и управление пользователями.

Например: `sudo apt update`

Чтобы ваш пользователь мог выполнить `sudo` команду он должен быть в группе `sudo`, для этого супер юзер должен выполнить для вашего юзера команду:

```
usermod -aG sudo username
```

Docker

Docker - это платформа для разработки, доставки и запуска приложений в контейнерах.

Контейнеры представляют собой легковесные, автономные пакеты программного обеспечения, которые включают в себя все необходимое для запуска приложения: код, библиотеки, зависимости, настройки и т. д.

Docker обеспечивает стандартизацию окружения приложения, упрощает развертывание и обеспечивает консистентность работы приложения в различных средах, таких как локальная среда разработки, тестирование и производство.

Docker: установка

На macos: [ТУТ](#)

На windows: [ТУТ](#)

На linux: [ТУТ](#)

Docker: основные понятия

Образ (Image): Образ Docker - это шаблон, который содержит все необходимые файлы и настройки для создания контейнера. Образы используются для создания контейнеров.

Контейнер (Container): Контейнер - это экземпляр образа Docker, который запущен с определенными настройками. Контейнеры изолированы друг от друга и от хост-системы.

Dockerfile: Dockerfile - это текстовый файл, который содержит инструкции для создания образа Docker. Он описывает шаги для установки и настройки окружения приложения.

Docker: основные команды

1. **docker build**: Создает образ Docker из Dockerfile.
2. **docker run**: Запускает контейнер из образа.
3. **docker pull**: Загружает образ Docker из репозитория Docker Hub или другого удаленного реестра.
4. **docker push**: Отправляет образ Docker в репозиторий Docker Hub или другой удаленный реестр.
5. **docker ps**: Показывает запущенные контейнеры.
6. **docker stop**: Останавливает запущенный контейнер.
7. **docker rm**: Удаляет контейнер.
8. **docker rmi**: Удаляет образ Docker.

Dockerfile и docker-compose

Dockerfile и docker-compose — это два разных инструмента, используемых для управления контейнерами в Docker, но они служат разным целям и работают по-разному.

Dockerfile

Dockerfile используется для создания образов Docker. Это скрипт с инструкциями, которые Docker использует для сборки образа контейнера. Он состоит из последовательности команд, таких как FROM, RUN, COPY, EXPOSE и других, которые определяют, как создать образ. Применяется при создании образов (например, установка зависимостей, копирование файлов, настройка конфигураций). Каждый образ создается из одной Dockerfile.

Docker-compose

Docker Compose используется для управления сложными или многоконтейнерными приложениями. Docker-compose файл (docker-compose.yml) использует YAML для описания конфигураций сервисов, сетей и томов.

С его помощью мы управляем сложными приложениями (например, веб-сервер + база данных). Он может определять и координировать работу нескольких контейнеров, их сетевых соединений и хранилищ.

`docker compose up` - команда - чтобы воспроизвести образ по yml



Ваша новая IT-профессия – Ваш новый уровень жизни

Программирование с нуля в
немецкой школе AIT TR GmbH