



НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
МОСКОВСКИЙ
ГОСУДАРСТВЕННЫЙ
СТРОИТЕЛЬНЫЙ
УНИВЕРСИТЕТ

Контейнеризация и переносимость программного обеспечения

Докладчик: Гагиев Георгий Русланович, студент бакалавриата 3 курса 4 группы ИЦТМС
Преподаватель: Иванов Николай Александрович, доцент кафедры ИСТАС, к.т.н., доцент

Что такое контейнеризация?



Контейнеризация — технология упаковки приложения с его зависимостями в изолированную среду (контейнер).

- Контейнеры используют ядро хост-системы, но работают изолированно.
- Позволяет запускать приложение "как есть" на любой платформе, где установлен контейнерный движок.

Почему контейнеризация актуальна?

- Упрощает развёртывание приложений
- Повышает переносимость и гибкость
- Идеально подходит для DevOps и CI/CD
- Позволяет запускать устаревшее ПО на новых системах

Популярные инструменты контейнеризации



Инструмент	Особенности	Преимущества	Недостатки
Docker	Самый популярный, CLI-интерфейс	Простота, экосистема, скорость	Требует root, GUI вручную
Podman	Без root, совместим с Docker CLI	Безопасность, rootless	Меньше документации
LXC	Контейнеры ОС уровня	Гибкость, системные службы	Сложнее в использовании
LXD	Надстройка над LXC с удобным API	Упрощённый интерфейс, образы	Более тяжёлый чем Docker

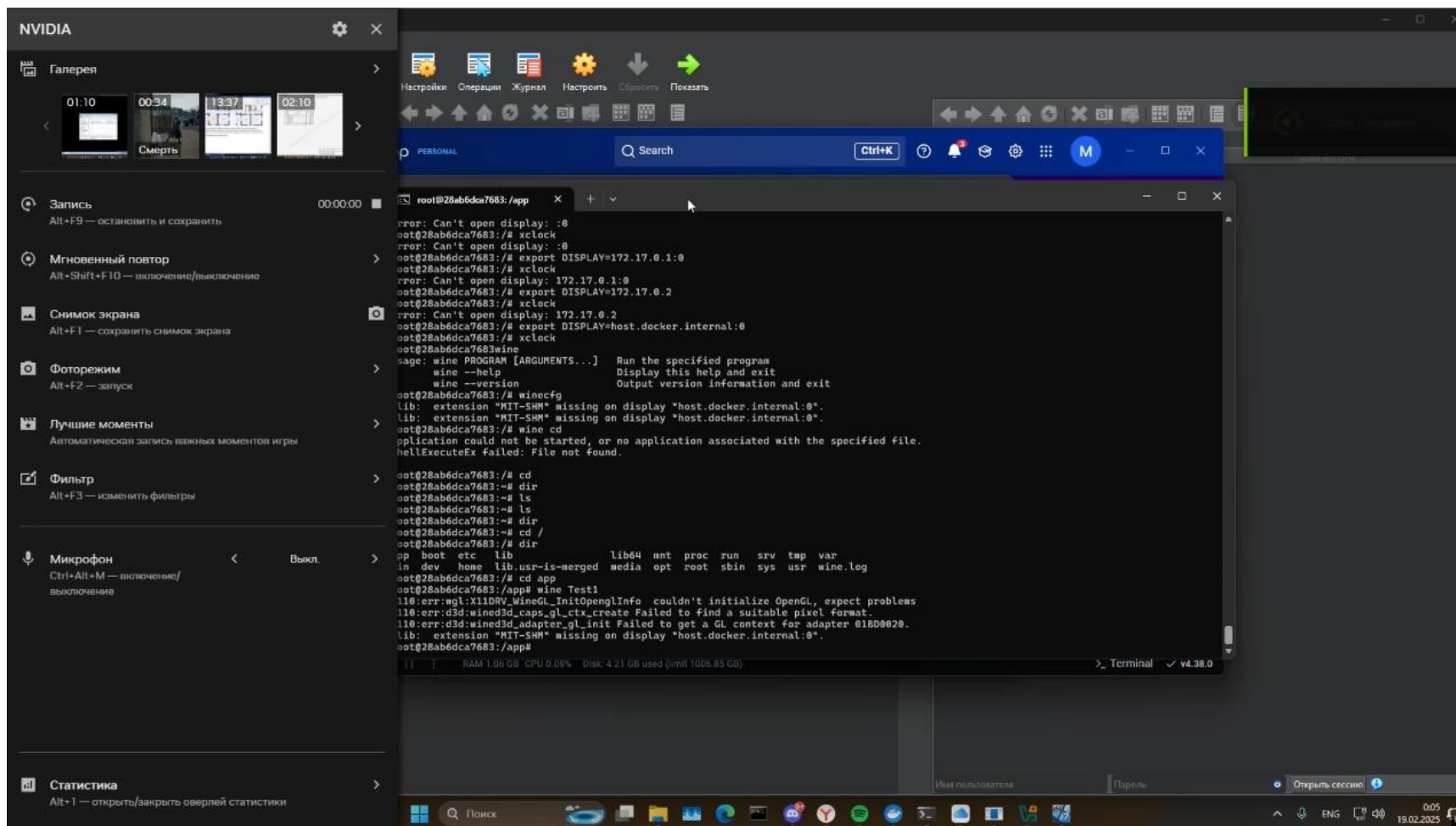
Контейнеризация vs Виртуализация

Параметр	Контейнер	Виртуальная машина
Размер	< 1 ГБ	> 5 ГБ
Скорость запуска	Миллисекунды	Минуты
Изоляция	На уровне ОС	Полная (с ядром)
Ресурсы	Меньше	Больше

Контейнеризация как средство переносимости ПО

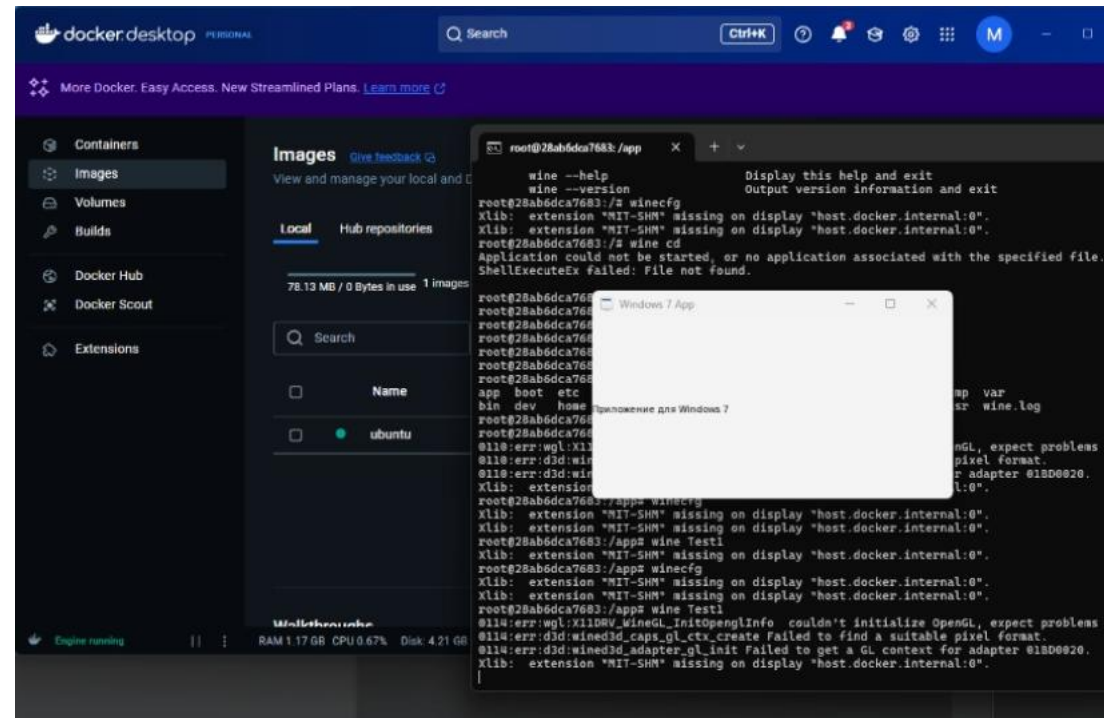
- Запуск приложений в разных ОС без переписывания кода
- Упаковка устаревшего ПО с зависимостями в контейнер
- Пример: запуск Windows 7-приложения через WINE внутри Docker-контейнера под Ubuntu

Пример



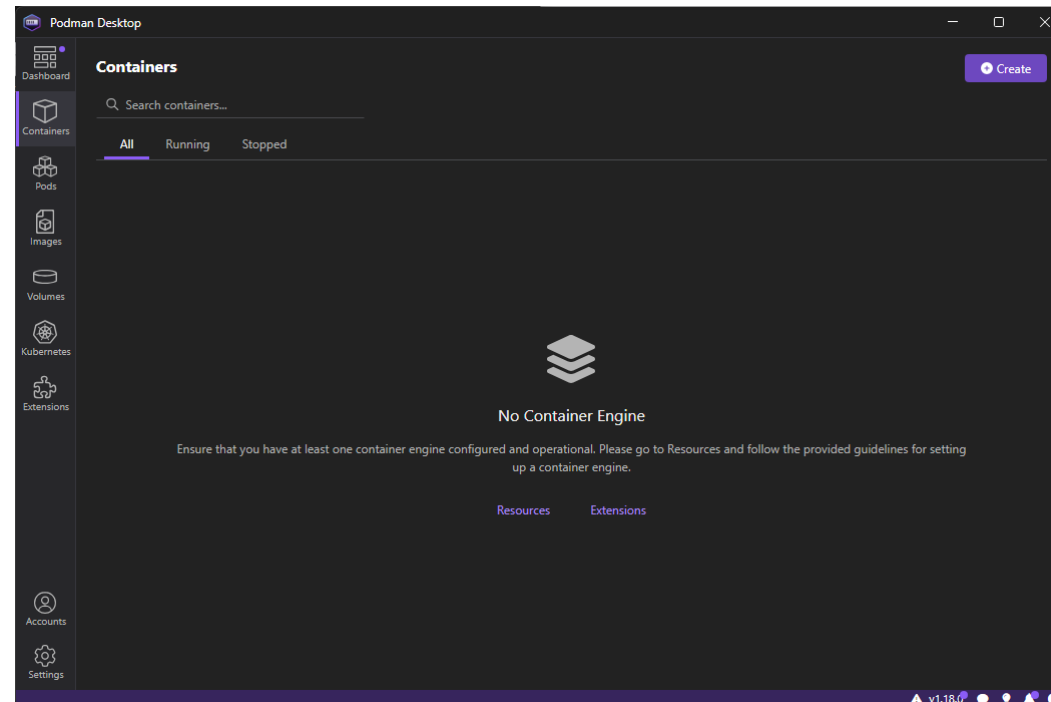
Пример: Docker + WINE под Ubuntu

- Запуск приложений в разных ОС без переписывания кода
- Упаковка устаревшего ПО с зависимостями в контейнер
- Пример: запуск Windows 7-приложения через WINE внутри Docker-контейнера под Ubuntu



Пример: Podman без root-доступа

- Идеален для ограниченных окружений (например, в вузе или корпорации)
- Запуск: `podman run --rm -it alpine`
- Поддержка GUI через X11-проброс



Пример: LXC/LXD

- Почти полноценная система внутри контейнера
- Подходит для работы с системными службами, init и сетевыми сервисами
- Пример команды: `lxc launch images:ubuntu/22.04 mycontainer`

```
└─| lxc list
+-----+-----+-----+-----+-----+
| NAME | STATE | IPV4 | IPV6 | TYPE | SNAPSHOTS |
+-----+-----+-----+-----+-----+
| ubuntu | RUNNING | 10.155.2.72 (eth0) | | CONTAINER | 0 |
+-----+-----+-----+-----+-----+
```

Рекомендации по выбору инструмента

Цель	Рекомендуемый инструмент
Простая упаковка и доставка	Docker
Безопасный запуск без root	Podman
Запуск системных служб	LXC/LXD
Контейнеризация GUI-приложений	Docker с WINE или LXD

Заключение



Контейнеризация упрощает переносимость, ускоряет разработку и снижает затраты. Использование WINE и Docker позволяет запускать устаревшее ПО в новых средах. LXD и Podman расширяют возможности контейнеризации в разных сценариях.



НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
МОСКОВСКИЙ
ГОСУДАРСТВЕННЫЙ
СТРОИТЕЛЬНЫЙ
УНИВЕРСИТЕТ

СПАСИБО ЗА ВНИМАНИЕ!