



★ 4.29

Оценка

163.48

Рейтинг

## Southbridge

Обеспечиваем стабильную работу highload-проектов



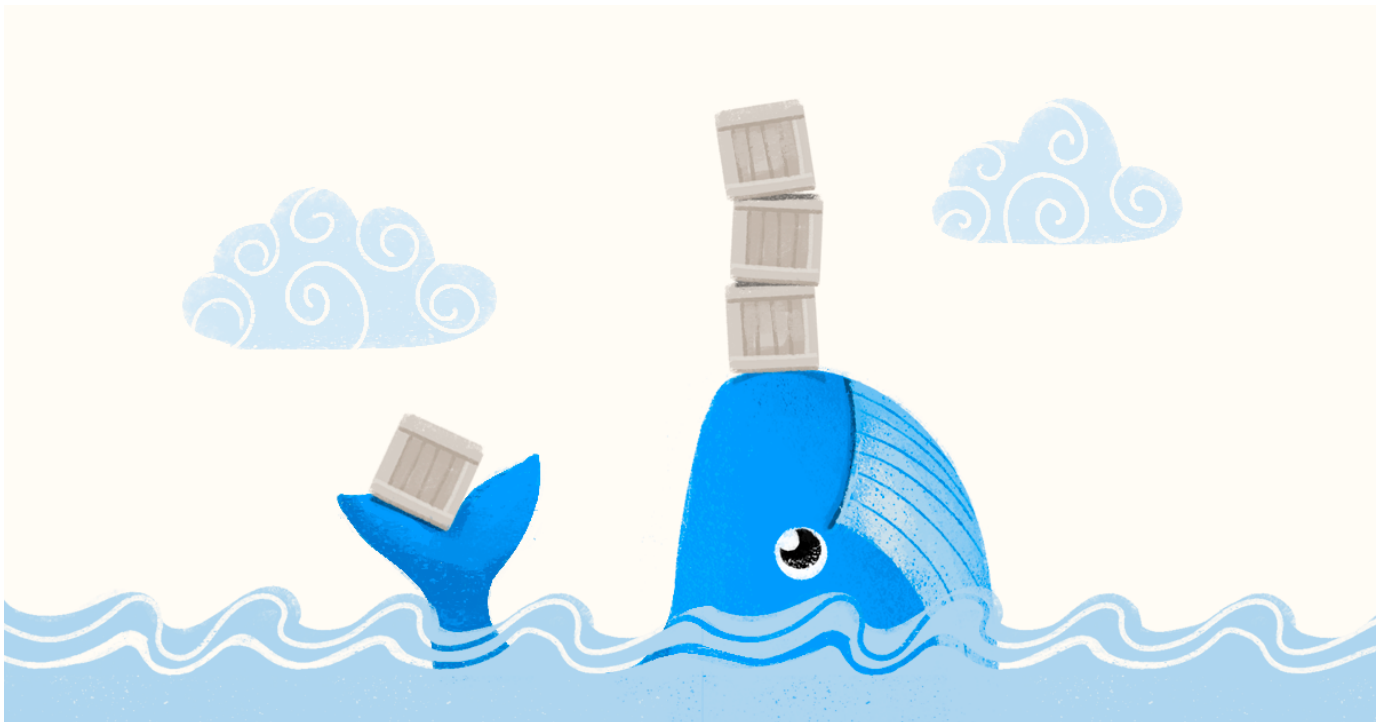
Rekken

23 янв 2021 в 14:14

## Хранение данных в Docker

🕒 7 мин 👁 69K

Блог компании Southbridge, Системное администрирование\*, Виртуализация\*, Серверное администриров



Важная характеристика Docker-контейнеров – эфемерность. В любой момент контейнер может рестартовать: завершиться и вновь запуститься из образа. При этом все накопленные в нём данные будут потеряны. Но как в таком случае запускать в Docker приложения, которые должны сохранять информацию о своём состоянии? Для этого есть несколько инструментов.

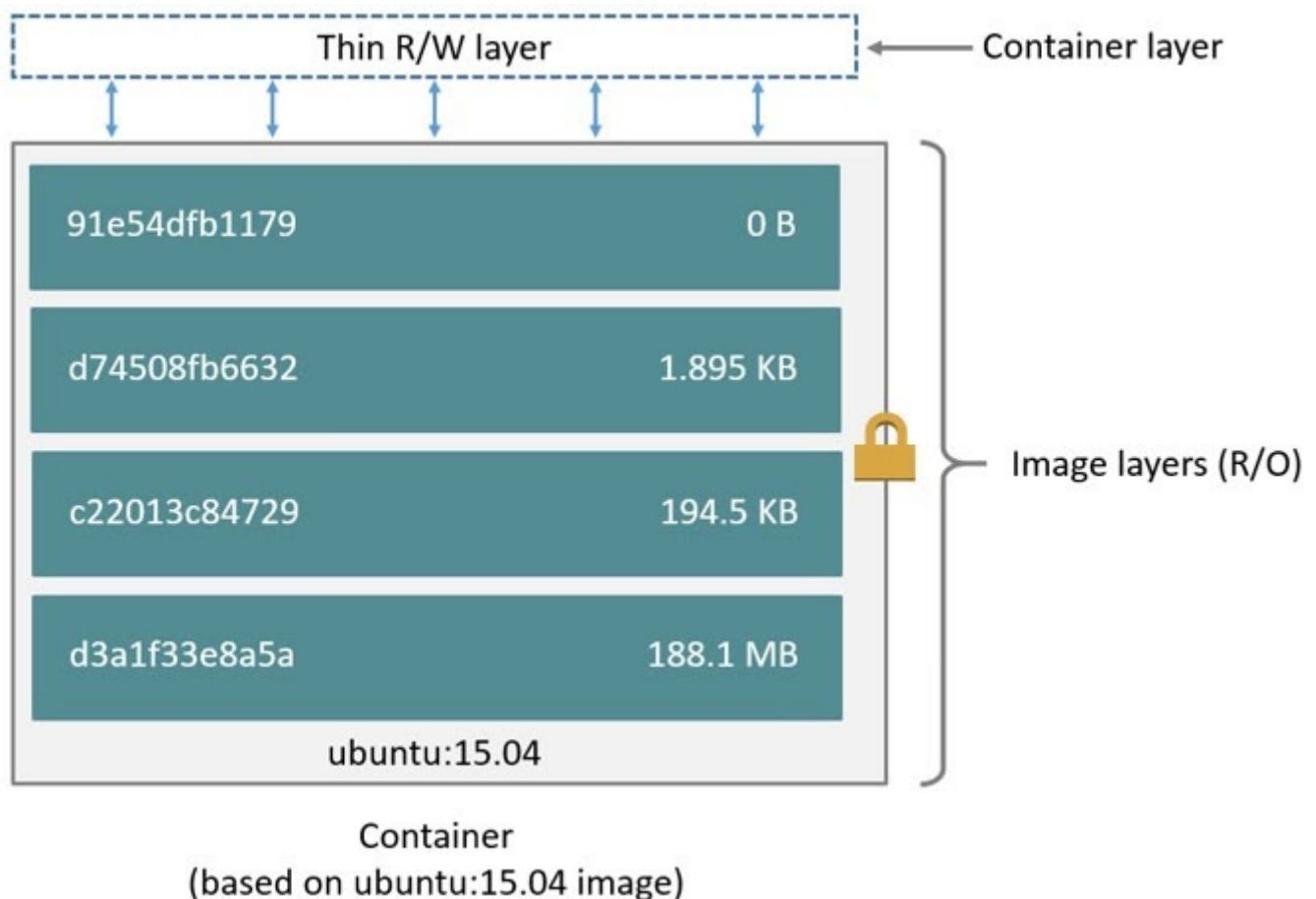
В этой статье рассмотрим docker volumes, bind mount и tmpfs, дадим советы по их использованию, проведём небольшую практику.



Прежде чем перейти к способам хранения данных, вспомним устройство контейнеров. Это поможет лучше понять основную тему.

Контейнер создаётся из образа, в котором есть всё для начала его работы. Но там не хранится и тем более не изменяется ничего важного. В любой момент приложение в контейнере может быть завершено, а контейнер уничтожен, и это нормально. Контейнер отработал – выкидываем его и собираем новый. Если пользователь загрузил в приложение картинку, то при замене контейнера она удалится.

На схеме показано устройство контейнера, запущенного из образа Ubuntu 15.04. Контейнер состоит из пяти слоёв: четыре из них принадлежат образу, и лишь один – самому контейнеру. Слои образа доступны только для чтения, слой контейнера – для чтения и для записи. Если при работе приложения какие-то данные будут изменяться, они попадут в слой контейнера. Но при уничтожении контейнера слой будет безвозвратно потерян, и все данные вместе с ним.



В идеальном мире Docker используют только для запуска stateless-приложений, которые не читают и не сохраняют данные о своём состоянии и готовы в любой момент завершиться. Однако в реальности большинство программ относятся к категории stateful, то есть требуют сохранения данных между перезапусками.

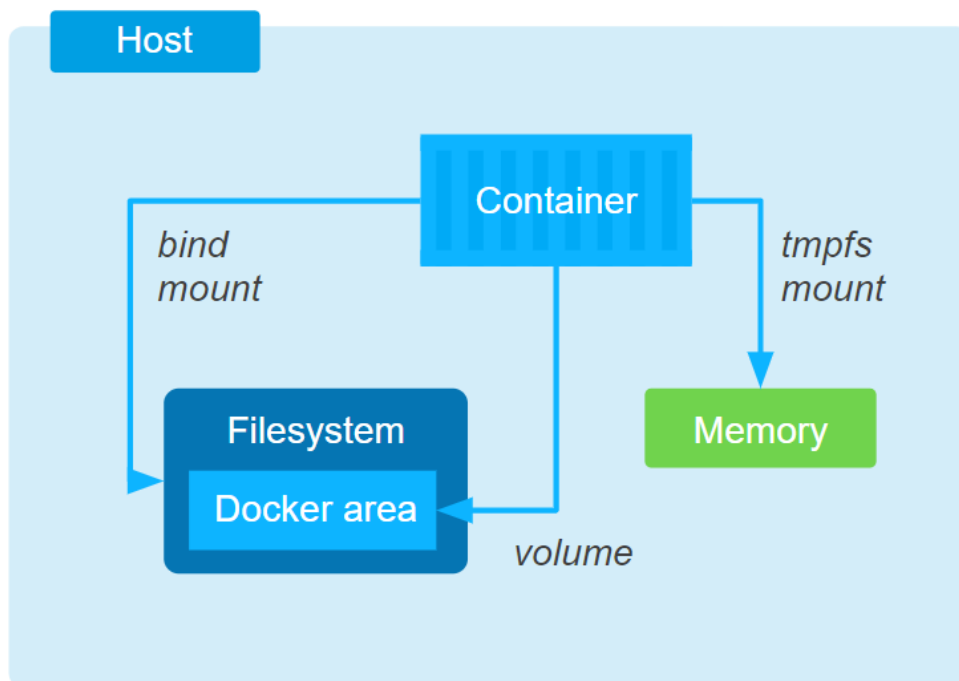
Поэтому нужны способы сделать так, чтобы важные изменяемые данные не зависели от эфемерности контейнеров и, как бонус, были доступными сразу из нескольких мест.

В Docker есть несколько способов хранения данных. Наиболее распространенные:

- тома хранения данных (`docker volumes`),
- монтирование каталогов с хоста (`bind mount`).

Особые типы хранения:

- именованные каналы (`named pipes`, только в Windows),
- монтирование `tmpfs` (только в Linux).



На схеме показаны самые популярные типы хранения данных для Linux: в памяти (`tmpfs`), в файловой системе хоста (`bind mount`), в томе Docker (`docker volumes`). Разберём каждый вариант.

## Тома (`docker volumes`)

Тома – рекомендуемый разработчиками Docker способ хранения данных. В Linux тома находятся по умолчанию в `/var/lib/docker/volumes/`. Другие программы не должны получать к ним доступ напрямую, только через контейнер.

Тома создаются и управляются средствами Docker: командой `docker volume create`, через указание тома при создании контейнера в `Dockerfile` или `docker-compose.yml`.

В контейнере том видно как обычный каталог, который мы определяем в `Dockerfile`. Тома могут быть с именами или без – безымянным томам Docker сам присвоит имя.

Один том может быть примонтирован одновременно в несколько контейнеров. Когда никто не использует том, он не удаляется, а продолжает существовать. Команда для удаления томов: `docker volume prune`.

Можно выбрать специальный драйвер для тома и хранить данные не на хосте, а на удалённом сервере или в облаке.

### Для чего стоит использовать тома в Docker:

- шаринг данных между несколькими запущенными контейнерами,
- решение проблемы привязки к ОС хоста,
- удалённое хранение данных,
- бэкап или миграция данных на другой хост с Docker (для этого надо остановить все контейнеры и скопировать содержимое из каталога тома в нужное место).

### Монтирование каталога с хоста (bind mount)

Это более простая концепция: файл или каталог с хоста просто монтируется в контейнер.

Используется, когда нужно пробросить в контейнер конфигурационные файлы с хоста. Например, именно так в контейнерах реализуется DNS: с хоста монтируется файл `/etc/resolv.conf`.

Другое очевидное применение – в разработке. Код находится на хосте (вашем ноутбуке), но выполняется в контейнере. Вы меняете код и сразу видите результат. Это возможно, так как процессы хоста и контейнера одновременно имеют доступ к одним и тем же данным.

### Особенности bind mount:

1. Запись в примонтированный каталог могут вести программы как в контейнере, так и на хосте. Это значит, есть риск случайно затереть данные, не понимая, что с ними

работает контейнер.

2. Лучше не использовать в продакшене. Для продакшена убедитесь, что код копируется в контейнер, а не монтируется с хоста.
3. Для успешного монтирования указывайте полный путь к файлу или каталогу на хосте.
4. Если приложение в контейнере запущено от `root`, а совместно используется каталог с ограниченными правами, то в какой-то момент может возникнуть проблема с правами на файлы и невозможность что-то удалить без использования `sudo`.

### Когда использовать тома, а когда монтирование с хоста

<u>Volume</u>	<u>Bind mount</u>
Просто расшарить данные между контейнерами.	Пробросить конфигурацию с хоста в контейнер.
У хоста нет нужной структуры каталогов.	Расшарить исходники и/или уже собранные приложения.
Данные лучше хранить не локально (а в облаке, например).	Есть стабильная структура каталогов и файлов, которую нужно расшарить между контейнерами.

### Монтирование tmpfs

Tmpfs – временное файловое хранилище. Это некая специально отведённая область в оперативной памяти компьютера. Из определения выходит, что tmpfs – не лучшее хранилище для важных данных. Так оно и есть: при остановке или перезапуске контейнера сохранённые в tmpfs данные будут навсегда потеряны.

На самом деле tmpfs нужно не для сохранения данных, а для безопасности, полученные в ходе работы приложения *чувствительные* данные безвозвратно исчезнут после завершения работы контейнера. Бонусом использования будет высокая скорость доступа к информации.

Например, приложение в контейнере тормозит из-за того, что в ходе работы активно идут операции чтения-записи, а диски на хосте не очень быстрые. Если вы не уверены, в какой каталог идёт эта нагрузка, можно применить к запущенному контейнеру команду `docker diff`. И вот этот каталог смонтировать как tmpfs, таким образом перенести ввод-вывод с диска в оперативную память.

Такое хранилище может одновременно работать только с одним контейнером и доступно только в Linux.

## Общие советы по использованию томов

### Монтирование в непустые директории

Если вы монтируете *пустой* том в каталог контейнера, где уже есть файлы, то эти файлы не удалятся, а будут скопированы в том. Этим можно пользоваться, когда нужно скопировать данные из одного контейнера в другой.

Если вы монтируете *непустой* том или каталог с хоста в контейнер, где уже есть файлы, то эти файлы тоже не удалятся, а просто будут скрыты. Видно будет только то, что есть в томе или каталоге на хосте. Похоже на простое монтирование в Linux.

### Монтирование служебных файлов

С хоста можно монтировать любые файлы, в том числе служебные. Например, сокет docker. В результате получится `docker-in-docker`: один контейнер запустится внутри другого. UPD: (\*это не совсем так. @mwizard в комментариях пояснил, что в таком случае родительский docker запустит sibling-контейнер). Выглядит как бред, но в некоторых случаях бывает оправдано. Например, при настройке CI/CD.

### Монтирование `/var/lib/docker`

Разработчики Docker говорят, что не стоит монтировать с хоста каталог `/var/lib/docker`, так как могут возникнуть проблемы. Однако есть некоторые программы, для запуска которых это необходимо.

#### ► Практика: создадим тестовый том

Если вам интересно узнать подробнее о работе с данными в Docker и других возможностях технологии, приглашаем на двухдневный онлайн-интенсив в феврале. Будет много практики.

*Автор статьи:* Александр Швалов, практикующий инженер Southbridge, Certified Kubernetes Administrator, автор и разработчик курсов Слёрм.


**Теги:** `docker`, `docker volumes`, `volume`, `bind mount`, `tmpfs`, контейнеризация, хранение данных

**Хабы:** Блог компании Southbridge, Системное администрирование, Виртуализация, Серверное администрирование, DevOps

Редакторский дайджест

Присылаем лучшие статьи раз в месяц

Электронпочта



Southbridge

Обеспечиваем стабильную работу highload-проектов

Сайт Сайт



25

0

Карма

Рейтинг

**Александр Швалов** @Rekken  
Администратор linux-серверов

 Комментарии 14

Публикации

ЛУЧШИЕ ЗА СУТКИ    ПОХОЖИЕ

22 часа назад

**Герои напильника и паяльника: итоги сезона DIY**

 8 мин

 8.4K


 Сезон DIY

 Спецпроект

 +44

 42


 6





spiritus\_sancti

23 часа назад

**RGB-усилители. Особенности, проблемы, выбор**

 Простой

 6 мин

 5.5K

[Тutorial](#) +40 24 11

ru\_vds

19 часов назад

## Профилирование Python — почему и где тормозит ваш код



Средний



10 мин



3.5K

[Tutorial](#)[Перевод](#) +31 92 4

ViktorSergeev

16 часов назад

## WebOne: даём жизнь старым браузерам



6 мин



3.3K

 +30 29 11

it\_union

2 часа назад

## ЗАО Гейм Инсайт Труп



Простой



4 мин



3.2K

 +22 7 7

zatiм

1 час назад

## Видеокарта VGA для микроконтроллера



Средний



17 мин



805

[Tutorial](#) +21 14 5

AnnaVIMorozova

19 часов назад

## Как повысить эффективность коммуникаций в команде: учимся решать конфликты



 7 мин  2.2K

 +19

 43

 2



Yu-Leo

вчера в 15:22

## Обзор электронной книги Meebook P10 Pro



Простой



9 мин



6.3K

Обзор

 +18

 18

 8



mr-pickles

22 часа назад

## Архетипы программных архитекторов. Часть 2



Простой



9 мин



2.5K

Перевод

 +15

 32

 0



Sagidullin

6 часов назад

## Всего два месяца — и новый релиз ядра Linux. Что появилось в ядре 6.5, что изменилось и что удалили. Новые возможности



5 мин



5.2K

 +14

 5

 5

Показать еще

### ИНФОРМАЦИЯ

Сайт

southbridge.io

Дата регистрации

15 ноября 2012

Дата основания

22 февраля 2008

Численность

51–100 человек

Местоположение

Россия

Представитель

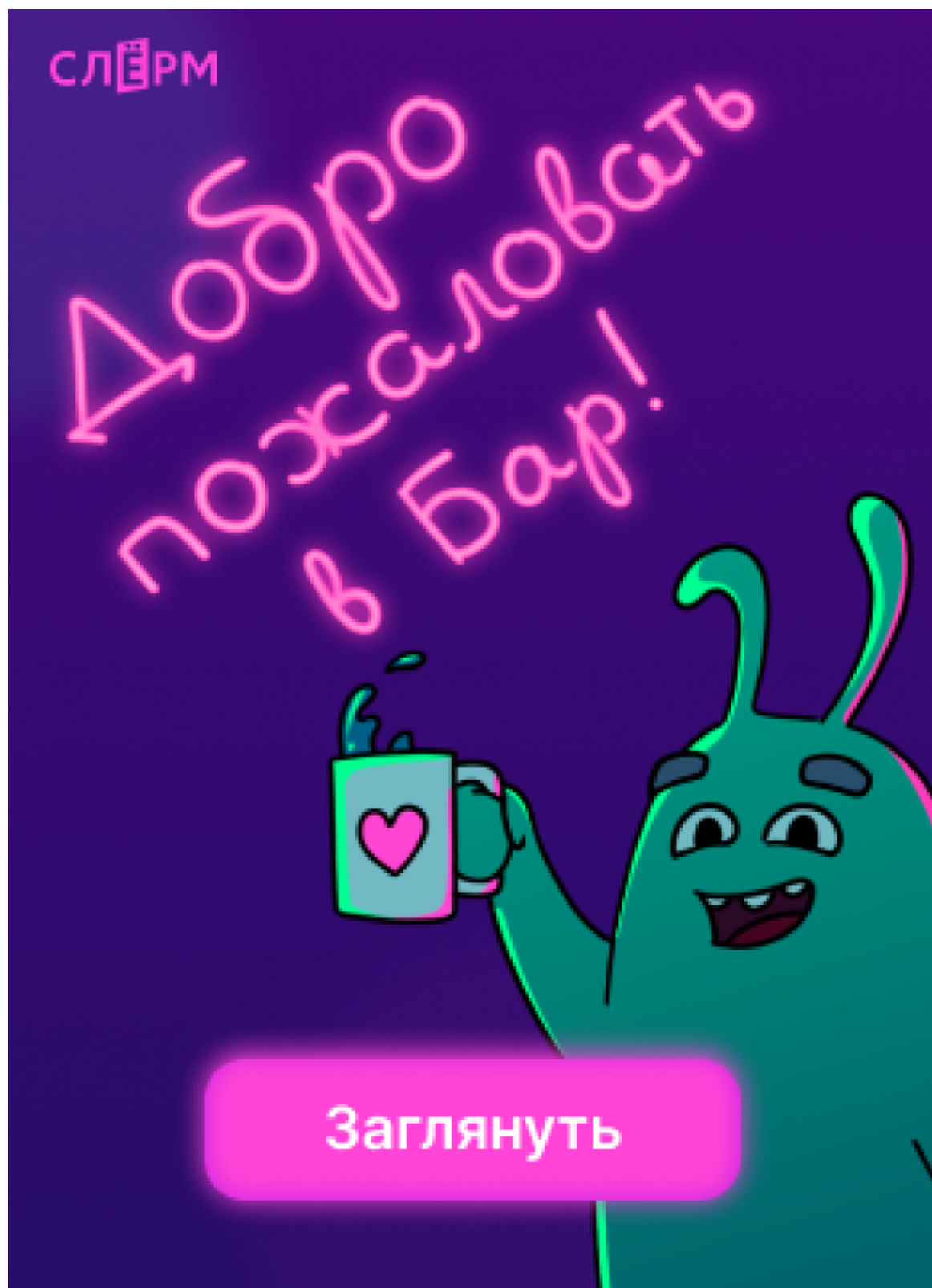
Антон Скобин

ССЫЛКИ

---

Подкаст «Так дела не делаются»  
slurm.club  
IT-курсы Слёрма  
slurm.club

ВИДЖЕТ



ВИДЖЕТ



## БЛОГ НА ХАБРЕ

24 авг в 21:00

Мы заглянули под капот Kafka и решили проблему потерянных сообщений

👁 4.2K 💬 3

24 авг в 16:58

Вам в хранилище или к озеру? Чем занимаются специалисты по работе с данными и как стать Data-инженером

3.3K 0

22 авг в 22:00

Преобразование табличных данных в Python

2.4K 1

18 авг в 18:00

Учимся использовать разные типы переменных в Ansible

2.3K 1

14 авг в 19:16

Apache Spark 3.4 для Databricks Runtime 13.0

699 0

Ваш аккаунт

Войти

Регистрация

Разделы

Статьи

Новости

Хабы

Компании

Авторы

Песочница

Информация

Устройство сайта

Для авторов

Для компаний

Документы

Соглашение

Конфиденциальность

Услуги

Корпоративный блог

Медийная реклама

Нативные проекты

Образовательные

программы

Стартапам

Спецпроекты



Настройка языка

[Техническая поддержка](#)

[Вернуться на старую версию](#)

© 2006–2023, Habr