Рафи Кази ар

НКАбд-03-24

1032238132@pfur.ru

Кулябов Дмитрий Сергеевич

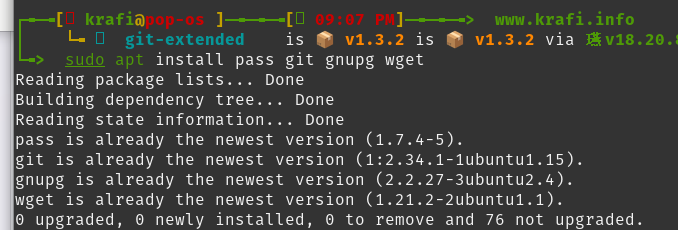
Moscow

### **Часть 1: Установка и настройка менеджера паролей Pass в Ubuntu**

#### **1.1. Установка**

**В системе Ubuntu:**

* **Обновление списка пакетов:** bash
* **Установка классического pass и необходимых зависимостей:** bash

sudo apt update & sudo apt install pass git gnupg wget

* + git - для синхронизации хранилища.
  + gnupg - для работы с GPG-ключами.
  + wget - для загрузки скриптов.
* **Установка gopass (расширенная версия на Go):** bash sudo apt install gopass *Далее в инструкции предполагается использование pass, но команды для gopass часто идентичны.*

#### **1.2. Настройка GPG-ключа**

Pass использует для шифрования GPG. Сначала нужен ключ.

* **Проверить существующие секретные ключи:** bash

gpg --list-secret-keys

* **Если ключей нет, создать новый (рекомендуется):** bash

gpg --full-generate-key

* 1. Выберите тип ключа RSA and RSA (по умолчанию).
  2. Размер ключа: 4096.
  3. Срок действия: 0 (ключ не истечёт).
  4. Введите своё имя и email. Эти данные будут идентифицировать ключ.
  5. **Внимание!** Придумайте и надёжно сохраните *парольную фразу (passphrase)* для этого ключа. Без него вы не расшифруете свои пароли.

#### **1.3. Инициализация хранилища паролей**

* Проинициализируйте хранилище, указав email или ID вашего GPG-ключа: bash pass init your-email@example.com Хранилище создастся в папке ~/.password-store/.

#### **1.4. Синхронизация с Git**

* **Настройка Git (если ещё не сделано):** bash

git config --global user.name "krafi"

git config --global user.email "krafi.info@gmail.com"

* **Инициализировать Git-репозиторий внутри хранилища:** bash

pass git init

* **Добавить удалённый репозиторий (предварительно создайте приватный репозиторий на GitHub/GitLab):** bash

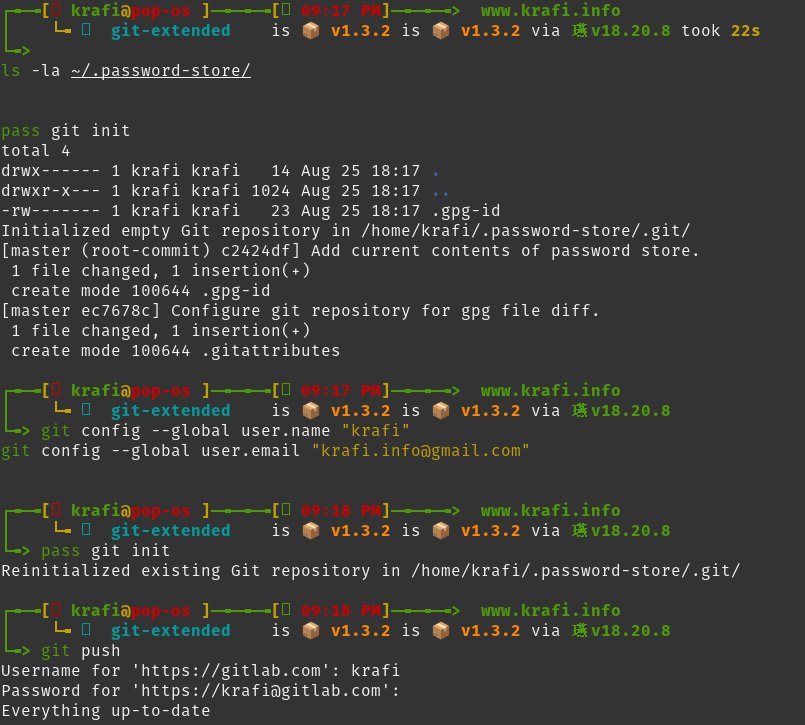
pass git remote add origin git@github.com:<ВАШ\_USERNAME>/<ВАШ\_REPO>.git

*Для использования HTTPS вместо SSH:* bash

pass git remote add origin https://github.com/<ВАШ\_USERNAME>/<ВАШ\_REPO>.git

* **Для синхронизации изменений:** bash

pass git pull # Получить изменения с удалённого репозитория

pass git push # Отправить локальные изменения на удалённый репозиторий

* **Важное замечание:** pass отслеживает только те изменения, которые сделаны через его команды. Если вы редактируете файлы напрямую в ~/.password-store/, вам нужно вручную сделать коммит и пуш: bash

cd ~/.password-store/

git add .

git commit -m 'Описание правок'

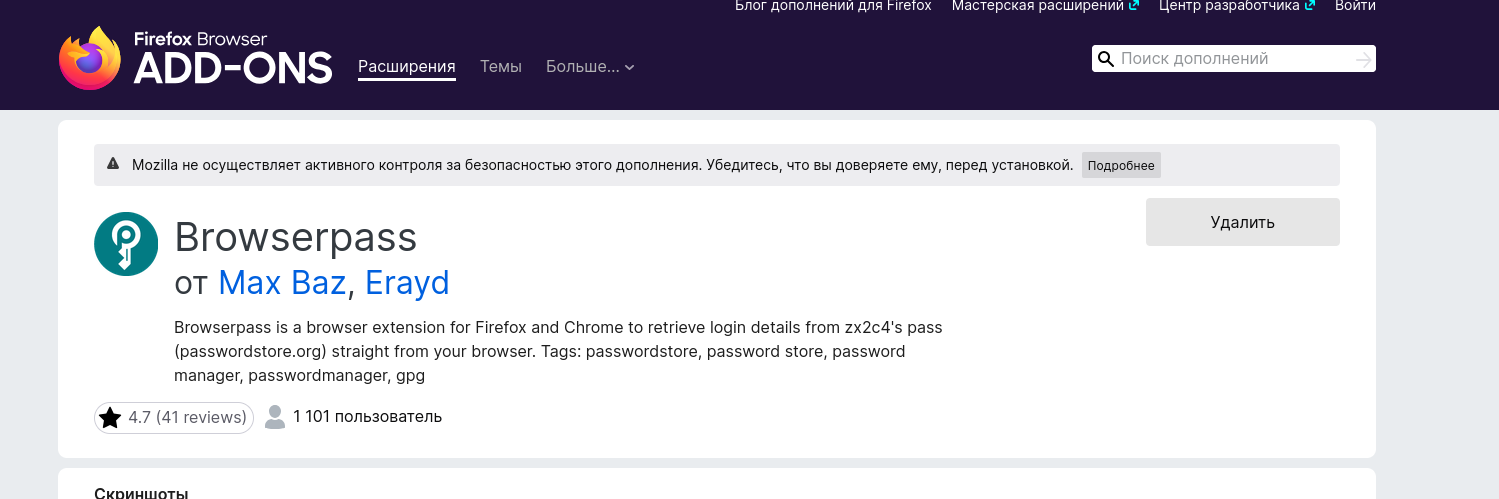
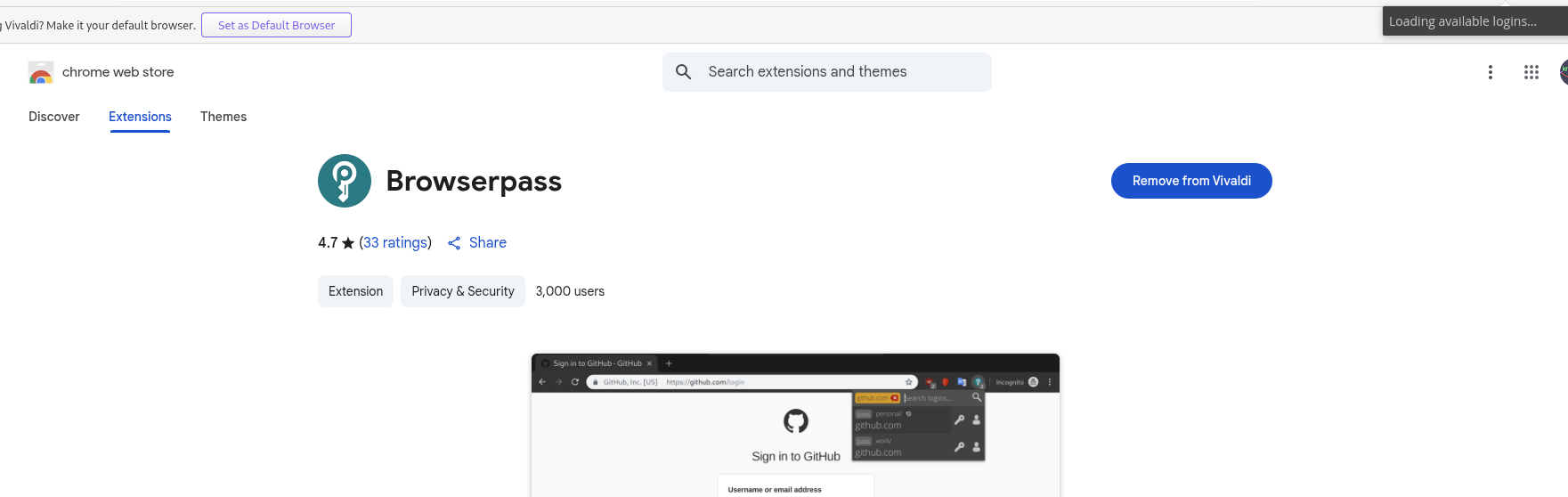
git push origin main

* **Проверить статус репозитория:** bash

pass git status

#### **1.5. Интеграция с браузером (Browserpass)**

Для удобного автозаполнения паролей в браузере.

* **Установите расширение для вашего браузера:**
  + **Firefox:** [Browserpass CE](https://addons.mozilla.org/ru/firefox/addon/browserpass-ce/)
  + **Chrome/Chromium:** [Browserpass CE](https://chrome.google.com/webstore/detail/browserpass-ce/naepdomgkenhinolocfifgehidddafch)

*После установки необходимо настроить браузер для работы с хостом. Часто это делается командой:* bash

sudo mkdir -p /etc/opt/chrome/native-messaging-hosts/ # Для Chrome sudo mkdir -p /etc/chromium/native-messaging-hosts/ # Для Chromium sudo mkdir -p /usr/lib/mozilla/native-messaging-hosts/ # Для Firefox sudo ./configure

# Запустить из каталога установки browserpass *Актуальные инструкции всегда смотрите на* [*GitHub Browserpass*](https://github.com/browserpass/browserpass-native)*.*

#### **1.6. Базовое использование Pass**

* **Добавить новый пароль (программа запросит его ввод):** bash

pass insert Соцсети/instagram.com

* **Показать пароль в терминале:** bash

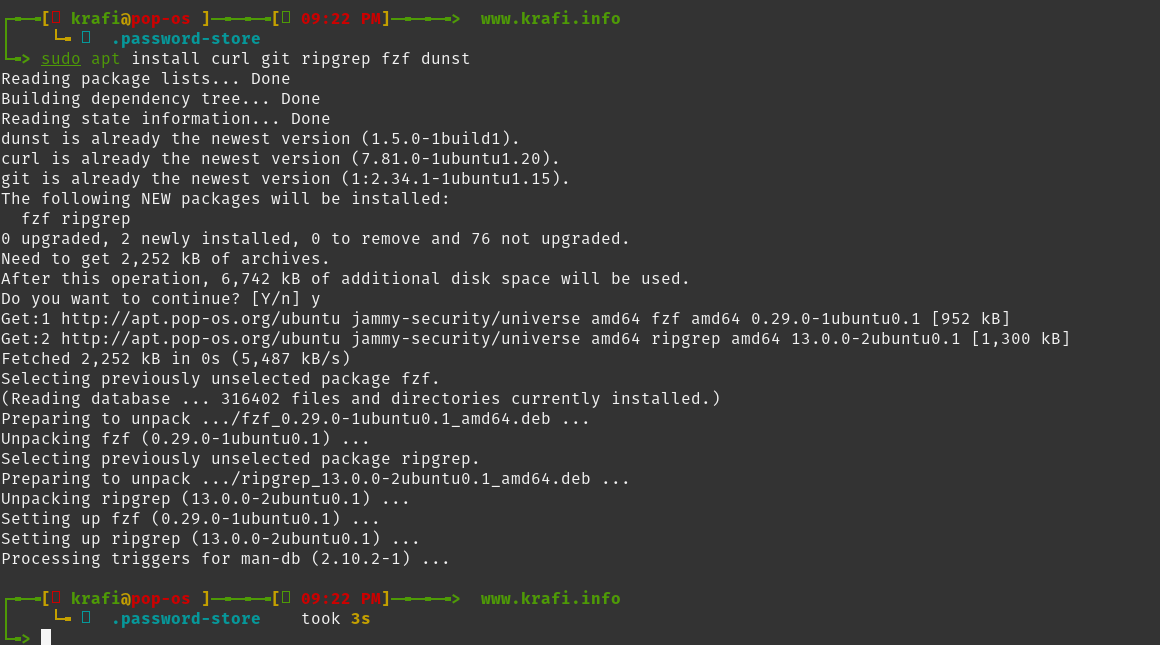
pass Соцсети/instagram.com

* **Скопировать пароль в буфер обмена (без отображения на экране):** bash pass -c Соцсети/instagram.com
* **Сгенерировать новый сложный пароль и сохранить его (заменит старый):** bash pass generate -n Соцсети/instagram.com 20 # 20 - длина пароля

### **Часть 2: Установка и настройка Chezmoi для управления конфигурацией в Ubuntu**

#### **2.1. Установка дополнительного ПО (опционально)**

Установка некоторых полезных утилит, часто встречающихся в конфигах:

sudo apt install curl git ripgrep fzf dunst *# Основные утилиты и dunst (уведомления)*

#### **2.2. Установка шрифсов (рекомендуется)**

Для правильного отображения значков и оформления в некоторых конфигах (например, для статус-баров). \* **Скачать и установить шрифты Nerd Fonts (например, FiraCode) вручную:** 1. Скачайте архив с [www.nerdfonts.com](https://www.nerdfonts.com/font-downloads)

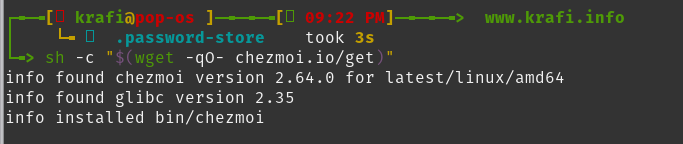
2. Распакуйте и скопируйте .ttf файлы в ~/.local/share/fonts/

3. Обновите кэш шрифтов:

fc-cache -fv

#### **2.3. Установка Chezmoi**

Установите последнюю версию с официального сайта (рекомендуемый способ):

sh -c "$(wget -qO- chezmoi.io/get)"

* *Альтернатива: установка из бинарного пакета (актуальную версию уточняйте на* [*GitHub*](https://github.com/twpayne/chezmoi/releases/latest)*):* bash # Пример для версии 2.47.1 и архитектуры AMD64 wget https://github.com/twpayne/chezmoi/releases/download/v2.47.1/chezmoi\_2.47.1\_linux\_amd64.deb sudo dpkg -i chezmoi\_2.47.1\_linux\_amd64.deb

#### **2.4. Создание своего репозитория dotfiles**

Используйте GitHub CLI (gh), чтобы создать **приватный** репозиторий. \* **Сначала установите gh:** bash sudo apt install gh \* **Авторизуйтесь:** bash

gh auth login \* **Создайте репозиторий:** bash

gh repo create dotfiles --private --clone *Если хотите использовать шаблон, добавьте флаг --template="yamadharma/dotfiles-template", но убедитесь, что шаблон совместим с Ubuntu.*

#### **2.5. Первоначальная настройка Chezmoi на первой машине**

1. **Инициализация:** Перейдите в склонированный репозиторий и инициализируйте Chezmoi. bash cd ~/dotfiles chezmoi init
2. **Просмотр планируемых изменений (ВСЕГДА делайте это сначала!):** bash chezmoi diff Эта команда покажет, какие файлы будут добавлены/изменены в вашем домашнем каталоге.
3. **Применение изменений:** Если вас всё устраивает, примените конфигурацию. bash chezmoi apply -v # Ключ -v для подробного вывода
4. **Добавление существующих конфигов:** Чтобы добавить текущий файл конфигурации (например, .bashrc) под управление Chezmoi: bash chezmoi add ~/.bashrc
5. **Сохранение изменений в ваш репозиторий:** bash chezmoi cd # Перейти в каталог хранилища Chezmoi git add . git commit -m "Добавил .bashrc" git push origin main

#### **2.6. Настройка второй машины (или чистой системы)**

1. **Инициализация и немедленное применение (рекомендуемый способ):** bash chezmoi init --apply https://github.com/<ВАШ\_USERNAME>/dotfiles.git
2. **Или стандартный способ:** bash chezmoi init https://github.com/<ВАШ\_USERNAME>/dotfiles.git chezmoi diff # Посмотреть, что поменяется chezmoi apply -v # Применить изменения

#### **2.7. Ежедневное использование Chezmoi**

* **Обновление и применение изменений из репозитория:** bash
* chezmoi update # Эквивалент:

git pull + chezmoi apply

* **Редактирование управляемого файла (правильный способ):** bash

chezmoi edit ~/.bashrc

# Откроет файл в $EDITOR После сохранения Chezmoi автоматически применит изменения локально.

* **Просмотр изменений перед применением:** bash

chezmoi diff

* **Если возник конфликт (файл изменён и локально, и в репозитории):** bash chezmoi merge ~/.bashrc # Запустит инструмент для разрешения конфликтов

#### **2.8. Автоматизация (опционально, с осторожностью)**

Чтобы Chezmoi автоматически коммитил и пушил изменения после команд add или edit, добавьте в ~/.config/chezmoi/chezmoi.toml:

**[git]**  
 autoCommit = true  
 autoPush = true