

Отчёт по лабораторной работе №5

Настройка рабочей среды

Седохин Даниил Алексеевич

Содержание

1	Цель работы	4
2	Задание	5
3	Теоретическое введение	6
4	Выполнение лабораторной работы	17
5	Выводы	38

Список иллюстраций

4.1	Установка pass	18
4.2	Установка gorpass	19
4.3	Просмотр списка ключей	20
4.4	Инициализация хранилища	20
4.5	Создание структуры git	21
4.6	Задаем адрес репозитория	21
4.7	Синхронизация	22
4.8	Коммит	22
4.9	Проверка статуса синхронизации	23
4.10	Настройка интерфейса	24
4.11	Плагин browserpass	25
4.12	Добавление нового пароля	26
4.13	Замена существующего пароля	26
4.14	Установка дополнительного ПО	27
4.15	Установка шрифтов	28
4.16	Установка шрифтов	29
4.17	Установка бинарного файла	30
4.18	Создание репозитория для конфигурационных файлов	30
4.19	Создание релиза на github	31
4.20	Проверка изменений	32
4.21	Запуск chezmoi	33
4.22	Инициализация chezmoi на второй машине	34
4.23	Запуск chezmoi	34
4.24	Применение изменений	35
4.25	Установка dotfiles	35
4.26	Извлечение изменений	36
4.27	Извлечение изменений	36
4.28	Автоматическая фиксация и отправление изменений в репозиторий	37

1 Цель работы

Получение навыков настройки рабочей среды.

2 Задание

Настроить рабочую среду.

3 Теоретическое введение

Менеджер паролей pass

Менеджер паролей pass – программа, сделанная в рамках идеологии Unix.

Также носит название стандартного менеджера паролей для Unix (The standard Unix password manager).

Основные свойства

Данные хранятся в файловой системе в виде каталогов и файлов.

Файлы шифруются с помощью GPG-ключа.

Структура базы паролей

Структура базы может быть произвольной, если Вы собираетесь использовать её напрямую.

Если же необходимо использовать дополнительное программное обеспечение, необходимо описать структуру базы.

Семантическая структура базы паролей

Рассмотрим пользователя `user` в домене `example.com`, порт `22`.

Отсутствие имени пользователя или порта в имени файла означает, что любое имя пользователя и порт будут использоваться.

`example.com.pgp`

Соответствующее имя пользователя может быть именем файла внутри каталога, имя которого `example.com`.

`example.com/user.pgp`

Имя пользователя также может быть записано в виде префикса, отделенного от хоста знаком

`user@example.com.pgp`

Соответствующий порт может быть указан после хоста, отделённый двоеточием (:):

`example.com:22.pgp`

`example.com:22/user.pgp`

`user@example.com:22.pgp`

Эти все записи могут быть расположены в произвольных каталогах, задающих Вашу собственную

Реализации

Утилиты командной строки

На данный момент существует 2 основных реализации:

`pass` – классическая реализация в виде shell-скриптов (<https://www.passwordstore.org/>)

`gopass` – реализация на go с дополнительными интегрированными функциями (<https://www.gopass.in/>)

Дальше в тексте будет использоваться программа `pass`, но всё то же самое можно сделать

Графические интерфейсы

`qtpass`

`qtpass` – может работать как графический интерфейс к `pass`, так и как самостоятельная

`gopass-ui`

`gopass-ui` – интерфейс к `gopass`.

`webpass`

Репозиторий: <https://github.com/emersion/webpass>

Веб-интерфейс к pass.

Написано на golang.

Приложения для Android

Password Store

URL: <https://play.google.com/store/apps/details?id=dev.msffjarvis.aps>

Репозиторий с кодом: <https://github.com/android-password-store/Android-Password-Store>

Документация: <https://android-password-store.github.io/docs/>

Для синхронизации с git необходимо импортировать ssh-ключи.

Поддерживает разблокировку по биометрическим данным.

Для работы требует наличия OpenKeychain: Easy PGP.

OpenKeychain: Easy PGP

URL: <https://play.google.com/store/apps/details?id=org.sufficientlysecure.keychain>

Операции с ключами pgp.

Необходимо будет импортировать pgp-ключи.

Не поддерживает разблокировку по биометрическим данным. Необходимо набирать пароль.

Пакеты для Emacs

pass

Основной режим для управления хранилищем и редактирования записей.

Emacs. Пакет pass

Репозиторий: <https://github.com/NicolasPetton/pass>

Позволяет редактировать базу данных паролей.

Запуск:

M-x pass

helm-pass

Интерфейс helm для pass.

Репозиторий: <https://github.com/emacs-helm/helm-pass>

Запуск:

M-x helm-pass

Выдаёт в минибуфере список записей из базы паролей. При нажатии Enter копирует пар

ivy-pass

Интерфейс ivy для pass.

Репозиторий: <https://github.com/ecraven/ivy-pass>

Управление файлами конфигурации

Использование chezmoi для управления файлами конфигурации домашнего каталога пользо

Общая информация

Сайт: <https://www.chezmoi.io/>

Репозиторий: <https://github.com/twpayne/chezmoi>

Конфигурация chezmoi

Рабочие файлы

Состояние файлов конфигурации сохраняется в каталоге

~/.local/share/chezmoi

Он является клоном вашего репозитория dotfiles.

Файл конфигурации ~/.config/chezmoi/chezmoi.toml (можно использовать также JSON или

Файлы, содержимое которых одинаково на всех ваших машинах, дословно копируются из ис

Файлы, которые варьируются от машины к машине, выполняются как шаблоны, обычно с испо

При запуске

```
chezmoi apply
```

вычисляется желаемое содержимое и разрешения для каждого файла, а затем вносит необходимые изменения, чтобы ваши файлы соответствовали этому состоянию.

По умолчанию chezmoi изменяет файлы только в рабочей копии.

Автоматически создавать файл конфигурации на новой машине

При выполнении chezmoi init также может автоматически создать файл конфигурации, есл

Если ваш репозиторий содержит файл с именем .chezmoi.\$FORMAT.tpl, где \$FORMAT есть с

Например, пусть ~/.local/share/chezmoi/.chezmoi.toml.tpl выглядит так:

```
{{- $email := promptStringOnce . "email" "Email address" -}}
```

```
[data]
```

```
email = {{ $email | quote }}
```

При выполнении chezmoi init будет создан конфигурационный файл ~/.config/chezmoi/

promptStringOnce – это специальная функция, которая запрашивает у пользователя зн

Чтобы протестировать этот шаблон, используйте chezmoi execute-template с флагами --init и --promptString, например:

```
chezmoi execute-template --init --promptString email=meghome.org < ~/.local/share/c
```

Пересоздание файл конфигурации

Если вы измените шаблон файла конфигурации, `chezmoi` предупредит вас, если ваш текущий

Вы можете повторно сгенерировать файл конфигурации, запустив:

```
chezmoi init
```

Шаблоны

Общая информация

Шаблоны используются для изменения содержимого файла в зависимости от среды.

Используется синтаксис шаблонов Go.

Файл интерпретируется как шаблон, если выполняется одно из следующих условий:

- имя файла имеет суффикс `.tmpl`;
- файл находится в каталоге `.chezmoitemplates`.

Данные шаблона

Полный список переменных шаблона:

```
chezmoi data
```

Источники переменных:

- файлы `.chezmoi`, например, `.chezmoi.os`;
- файлы конфигурации `.chezmoidata.$FORMAT`. Форматы (`json`, `jsonc`, `toml`, `yaml`) читают раздел `data` конфигурационного файла.

Способы создания файла шаблона

При первом добавлении файла передайте аргумент `--template`:

```
chezmoi add --template ~/.zshrc
```

Если файл уже контролируется `chezmoi`, но не является шаблоном, можно сделать его шаблоном:

```
chezmoi chattr +template ~/.zshrc
```

Можно создать шаблон вручную в исходном каталоге, присвоив ему расширение `.tmpl`:

```
chezmoi cd  
$EDITOR dot_zshrc.tmpl
```

Шаблоны в каталоге `.chezmoitemplates` должны создаваться вручную:

```
chezmoi cd  
mkdir -p .chezmoitemplates  
cd .chezmoitemplates  
$EDITOR mytemplate
```

Редактирование файла шаблона

Используйте `chezmoi edit`:

```
chezmoi edit ~/.zshrc
```

Чтобы сделанные вами изменения сразу же применялись после выхода из редактора, используйте `--apply`:

```
chezmoi edit --apply ~/.zshrc
```

Тестирование шаблонов

Тестирование с помощью команды `chezmoi execute-template`.

Тестирование небольших фрагментов шаблонов:

```
chezmoi execute-template '{{ .chezmoi.hostname }}'
```

Тестирование целых файлов:

```
chezmoi cd
chezmoi execute-template < dot_zshrc.tmpl
```

Синтаксис шаблона

Действия шаблона записываются внутри двойных фигурных скобок, `{{ }}`.

Действия могут быть переменными, конвейерами или операторами управления.

Текст вне действий копируется буквально.

Переменные записываются буквально:

```
{{ .chezmoi.hostname }}
```

Условные выражения могут быть записаны с использованием `if`, `else if`, `else`, `end`:

```
{{ if eq .chezmoi.os "darwin" }}
darwin
```

```
{{ else if eq .chezmoi.os "linux" }}
linux
```

```
{{ else }}
```

```
other operating system
```

```
{{ end }}
```

Удаление пробелов

Для удаления пробелов в шаблоне разместите знак минус и пробела рядом со скобками:

```
HOSTNAME={{- .chezmoi.hostname }}
```

В результате получим:

```
HOSTNAME=myhostname
```

Отладка шаблона

Используется подкоманда `execute-template`:

```
chezmoi execute-template '{{ .chezmoi.os }}/{{ .chezmoi.arch }}
```

Интерпретируются любые данные, поступающие со стандартного ввода или в конце команды.

Можно передать содержимое файла этой команде:

```
cat foo.txt | chezmoi execute-template
```

Логические операции

Возможно выполнение логических операций.

Если имя хоста машины равно work-laptop, текст между if и end будет включён в резул

```
# common config
export EDITOR=vi

# machine-specific configuration
{{- if eq .chezmoi.hostname "work-laptop" }}
# this will only be included in ~/.bashrc on work-laptop
{{- end }}
```

Логические функции

eq: возвращает true, если первый аргумент равен любому из остальных аргументов,
not: возвращает логическое отрицание своего единственного аргумента;
and: возвращает логическое И своих аргументов, может принимать несколько аргумен
or: возвращает логическое ИЛИ своих аргументов, может принимать несколько аргумен

Целочисленные функции

len: возвращает целочисленную длину своего аргумента;
eq: возвращает логическую истину arg1 == arg2;
ne: возвращает логическое значение arg1 != arg2;
lt: возвращает логическую истину arg1 < arg2;
le: возвращает логическую истину arg1 <= arg2;
gt: возвращает логическую истину arg1 > arg2;
ge: возвращает логическую истину arg1 >= arg2.

Переменные шаблона

Чтобы просмотреть переменные, доступные в вашей системе, выполните:

chezmoi data

Чтобы получить доступ к переменной `chezmoi.kernel.osrelease` в шаблоне, используйте:

```
{{ .chezmoi.kernel.osrelease }}
```


4 Выполнение лабораторной работы

1) Установка

Установим pass и gorpass (рис. 4.1 4.2).

```
[sdaniil@sdaniil ~]$ sudo dnf install pass pass-otp
[sudo] пароль для sdaniil:
Copr repo for gitflow owned by elegos      3.3 kB/s | 1.5 kB    00
Fedora 39 - x86_64                        54 kB/s | 22 kB     00
Fedora 39 - x86_64 - Updates               28 kB/s | 18 kB     00
Fedora 39 - x86_64 - Updates              3.9 MB/s | 7.9 MB    00
Зависимости разрешены.
=====
Пакет          Архитектура  Версия          Репозиторий
=====
Установка:
pass           noarch       1.7.4-8.fc39    fedora
pass-otp       noarch       1.2.0-12.fc39   fedora
Установка зависимостей:
liboath        x86_64       2.6.9-2.fc39    fedora
oathtool       x86_64       2.6.9-2.fc39    fedora
qrencode       x86_64       4.1.1-5.fc39    fedora

Результат транзакции
=====
Установка 5 Пакетов

Объем загрузки: 205 k
Объем изменений: 414 k
Продолжить? [д/Н]: д
Загрузка пакетов:
(1/5): liboath-2.6.9-2.fc39.x86_64.rpm 832 kB/s | 48 kB    00
(2/5): oathtool-2.6.9-2.fc39.x86_64.rpm 755 kB/s | 45 kB    00
(3/5): pass-1.7.4-8.fc39.noarch.rpm    946 kB/s | 59 kB    00
(4/5): qrencode-4.1.1-5.fc39.x86_64.rpm 1.5 MB/s | 25 kB    00
(5/5): pass-otp-1.2.0-12.fc39.noarch.rp 1.2 MB/s | 28 kB    00
-----
Общий размер          700 kB/s | 205 kB    00
Проверка транзакции
Проверка транзакции успешно завершена.
```

Рис. 4.1: Установка pass

```

[sdaniil@sdaniil ~]$ sudo dnf install gopass
Последняя проверка окончания срока действия метаданных: 0:04:28
6 16 мар 2024 16:58:17.
Зависимости разрешены.
=====
Пакет                Архитектура Версия                Репозиторий
=====
Установка:
gopass                x86_64        1.15.2-2.fc38          fedora
Установка зависимостей:
fish                  x86_64        3.7.0-1.fc39           updates
pcre2-utf32           x86_64        10.42-1.fc39.2         fedora

Результат транзакции
=====
Установка 3 Пакета

Объем загрузки: 11 М
Объем изменений: 33 М
Продолжить? [д/Н]: у
Загрузка пакетов:
(1/3): pcre2-utf32-10.42-1.fc39.2.x86_6 247 kB/s | 200 kB    00
(2/3): fish-3.7.0-1.fc39.x86_64.rpm    2.6 MB/s | 3.4 MB    00
(3/3): gopass-1.15.2-2.fc38.x86_64.rpm  2.3 MB/s | 7.1 MB    00
-----
Общий размер                2.4 MB/s | 11 MB    00
Проверка транзакции
Проверка транзакции успешно завершена.
Идет проверка транзакции
Тест транзакции проведен успешно.
Выполнение транзакции
Подготовка      :
Установка       : pcre2-utf32-10.42-1.fc39.2.x86_64
Установка       : fish-3.7.0-1.fc39.x86_64
Запуск скриптлета: fish-3.7.0-1.fc39.x86_64
Установка       : gopass-1.15.2-2.fc38.x86_64
Запуск скриптлета: gopass-1.15.2-2.fc38.x86_64
Проверка        : gopass-1.15.2-2.fc38.x86_64
Проверка        : pcre2-utf32-10.42-1.fc39.2.x86_64
Проверка        : fish-3.7.0-1.fc39.x86_64

Установлен:
fish-3.7.0-1.fc39.x86_64          gopass-1.15.2-2.fc38.x86_64
pcre2-utf32-10.42-1.fc39.2.x86_64

```

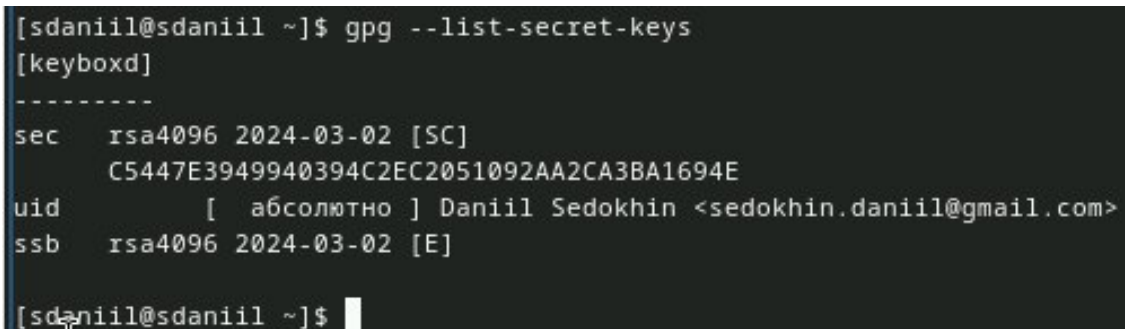
Рис. 4.2: Установка gopass

2) Настройка

Ключи GPG

Просмотр списка ключей:

`gpg --list-secret-keys` (рис. [-@fig:003]).



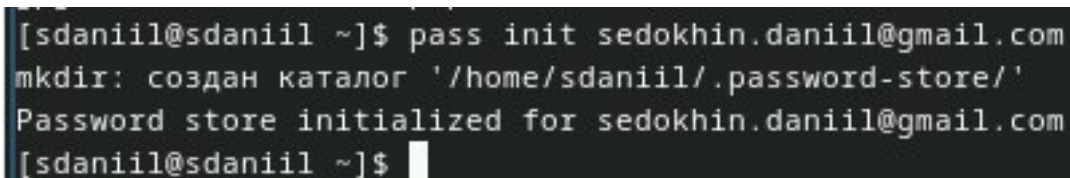
```
[sdaniil@sdaniil ~]$ gpg --list-secret-keys
[keyboxd]
-----
sec   rsa4096 2024-03-02 [SC]
      C5447E3949940394C2EC2051092AA2CA3BA1694E
uid   [ абсолютно ] Daniil Sedokhin <sedokhin.daniil@gmail.com>
ssb   rsa4096 2024-03-02 [E]
[sdaniil@sdaniil ~]$
```

Рис. 4.3: Просмотр списка ключей

3) Инициализация хранилища

Инициализируем хранилище:

`pass init` (рис. 4.4).



```
[sdaniil@sdaniil ~]$ pass init sedokhin.daniil@gmail.com
mkdir: создан каталог '/home/sdaniil/.password-store/'
Password store initialized for sedokhin.daniil@gmail.com
[sdaniil@sdaniil ~]$
```

Рис. 4.4: Инициализация хранилища

4) Синхронизация с git

Создадим структуру git:

pass git init (рис. 4.5)

```
[sdaniil@sdaniil ~]$ pass git init
подсказка: Using 'master' as the name for the initial branch. This default branch name
подсказка: is subject to change. To configure the initial branch name to use in all
подсказка: of your new repositories, which will suppress this warning, call:
подсказка:
подсказка:     git config --global init.defaultBranch <name>
подсказка:
подсказка: Names commonly chosen instead of 'master' are 'main', 'trunk' and
подсказка: 'development'. The just-created branch can be renamed via this command:
подсказка:
подсказка:     git branch -m <name>
Инициализирован пустой репозиторий Git в /home/sdaniil/.password-store/.git/
[master (корневой коммит) 773762f] Add current contents of password store.
1 file changed, 1 insertion(+)
create mode 100644 .gpg-id
[master a1a1e6d] Configure git repository for gpg file diff.
1 file changed, 1 insertion(+)
create mode 100644 .gitattributes
[sdaniil@sdaniil ~]$
```

Рис. 4.5: Создание структуры git

- 5) Также можно задать адрес репозитория на хостинге (репозиторий необходимо предварительно создать):

pass git remote add origin git@github.com:./git (рис. 4.6).

```
[sdaniil@sdaniil ~]$ pass git remote add origin git@github.com:Danil2234/pass.git
```

Рис. 4.6: Задаем адрес репозитория

- 6) Для синхронизации выполняется следующая команда:

pass git pull

pass git push (рис. 4.7).

```

[sdaniil@sdaniil ~]$ pass git push --set-upstream origin master
Перечисление объектов: 6, готово.
Подсчет объектов: 100% (6/6), готово.
При сжатии изменений используется до 6 потоков
Сжатие объектов: 100% (3/3), готово.
Запись объектов: 100% (6/6), 1.82 КиБ | 1.82 МиБ/с, готово.
Total 6 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To github.com:Danii12234/pass.git
 * [new branch]      master -> master
branch 'master' set up to track 'origin/master'.
[sdaniil@sdaniil ~]$

```

Рис. 4.7: Синхронизация

7) Прямые изменения

Следует заметить, что отслеживаются только изменения, сделанные через сам gorpass (или pass).

Если изменения сделаны непосредственно на файловой системе, необходимо вручную закоммитить и выложить изменения:

```

cd ~/.password-store/
git add .
git commit -am 'edit manually'
git push (рис. 4.8).

```

```

[sdaniil@sdaniil ~]$ cd ~/.password-store/
[sdaniil@sdaniil .password-store]$ git add .
[sdaniil@sdaniil .password-store]$ git commit -am 'edit manually'
Текущая ветка: master
Эта ветка соответствует «origin/master».

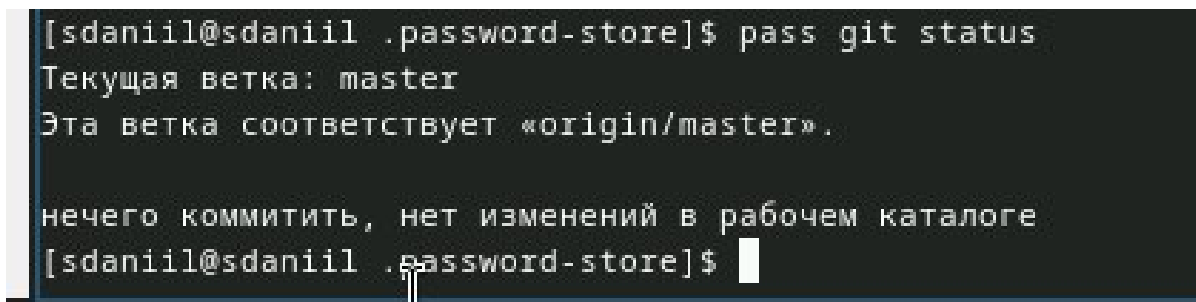
ничего коммитить, нет изменений в рабочем каталоге
[sdaniil@sdaniil .password-store]$ git push
Everything up-to-date
[sdaniil@sdaniil .password-store]$

```

Рис. 4.8: Коммит

8) Проверить статус синхронизации можно командой

`pass git status` (рис. 4.9).



```
[sdaniil@sdaniil .password-store]$ pass git status
Текущая ветка: master
Эта ветка соответствует «origin/master».

ничего коммитить, нет изменений в рабочем каталоге
[sdaniil@sdaniil .password-store]$
```

Рис. 4.9: Проверка статуса синхронизации

9) Настройка интерфейса с браузером

Для взаимодействия с браузером используется интерфейс `native messaging`.

Поэтому кроме плагина к браузеру устанавливается программа, обеспечивающая интерфейс `native messaging`.

Плагин `browserpass`

Репозиторий: <https://github.com/browserpass/browserpass-extension>

Плагин для браузера Плагин для Firefox: [https://addons.mozilla.org/en-](https://addons.mozilla.org/en-US/firefox/addon/browserpass-ce/)

US/firefox/addon/browserpass-ce/. Плагин для Chrome/Chromium: [https://chrome.google.co](https://chrome.google.com/webstore/detail/browser-password-manager/gaebfhgkneihfbiohmkngcmhphgmmgocm)
ce/naepdomgkenhinolocfifgehiddafch.

Интерфейс для взаимодействия с браузером (`native messaging`)

Репозиторий: [https://github.com/browserpass/browserpass-](https://github.com/browserpass/browserpass-native)
`native`

Gentoo:

`emerge www-plugins/browserpass`

Fedora

```
dnf copr enable maximbaz/browserpass
```

```
dnf install browserpass
```

(рис. 4.10 4.11).

```
[sdaniil@sdaniil .password-store]$ dnf copr enable maximbaz/browserpass
Ошибка: Эта команда должна быть выполнена от имени пользователя root.
[sdaniil@sdaniil .password-store]$ sudo dnf copr enable maximbaz/browserpass
[sudo] пароль для sdaniil:
Включение репозитория Copr. Обратите внимание, что этот репозиторий
не является частью основного дистрибутива, и качество может отличаться.

Проект Fedora не имеет какого-либо влияния на содержимое этого
репозитория за рамками правил, описанных в Вопросах и Ответах Copr в
<https://docs.pagure.org/copr.copr/user\_documentation.html#what-i-can-build-in-copr>,
а качество и безопасность пакетов не поддерживаются на каком-либо уровне.

Не отправляйте сообщения об ошибках этих пакетов в Fedora
Bugzilla. В случае возникновения проблем обращайтесь к владельцу этого репозито
рия.

Do you really want to enable copr.fedorainfracloud.org/maximbaz/browserpass? [y
/N]: y
Репозиторий успешно подключен.
[sdaniil@sdaniil .password-store]$ █
```

Рис. 4.10: Настройка интерфейса


```

Результат транзакции
=====
Установка 3 Пакета

Объем загрузки: 3.3 М
Объем изменений: 5.4 М
Продолжить? [д/н]: у
Загрузка пакетов:
(1/3): browserpass-chromium-3.8.0-2.fc39.noarc 1.9 MB/s | 1.1 MB    00:00
(2/3): browserpass-3.1.0-1.fc39.x86_64.rpm    1.9 MB/s | 1.1 MB    00:00
(3/3): browserpass-firefox-3.8.0-1.fc39.noarch 1.8 MB/s | 1.1 MB    00:00
-----
Общий размер                    5.3 MB/s | 3.3 MB    00:00
Copr repo for browserpass owned by maximbaz    15 kB/s | 1.0 kB    00:00
Импорт GPG-ключа 0xAE948483:
Идентификатор пользователя: "maximbaz_Browserpass (None) <maximbaz#Browserpass
@copr.fedorahosted.org>"
Отпечаток: 3686 65C9 3F16 72C7 AF55 1166 EF7C FF85 AE94 8483
Источник: https://download.copr.fedorainfracloud.org/results/maximbaz/browserp
ass/pubkey.gpg
Продолжить? [д/н]: у
Импорт ключа успешно завершен
Проверка транзакции
Проверка транзакции успешно завершена.
Идет проверка транзакции
Тест транзакции проведен успешно.
Выполнение транзакции
Подготовка      :                               1/1
Установка       : browserpass-chromium-3.8.0-2.fc39.noarch 1/3
Установка       : browserpass-firefox-3.8.0-1.fc39.noarch 2/3
Установка       : browserpass-3.1.0-1.fc39.x86_64        3/3
Запуск скриптлета: browserpass-3.1.0-1.fc39.x86_64        3/3
Проверка        : browserpass-3.1.0-1.fc39.x86_64        1/3
Проверка        : browserpass-chromium-3.8.0-2.fc39.noarch 2/3
Проверка        : browserpass-firefox-3.8.0-1.fc39.noarch 3/3

Установлен:
browserpass-3.1.0-1.fc39.x86_64
browserpass-chromium-3.8.0-2.fc39.noarch
browserpass-firefox-3.8.0-1.fc39.noarch

Выполнено!
[sdaniil@sdaniil .password-store]$ █

```

Рис. 4.11: Плагин browserpass

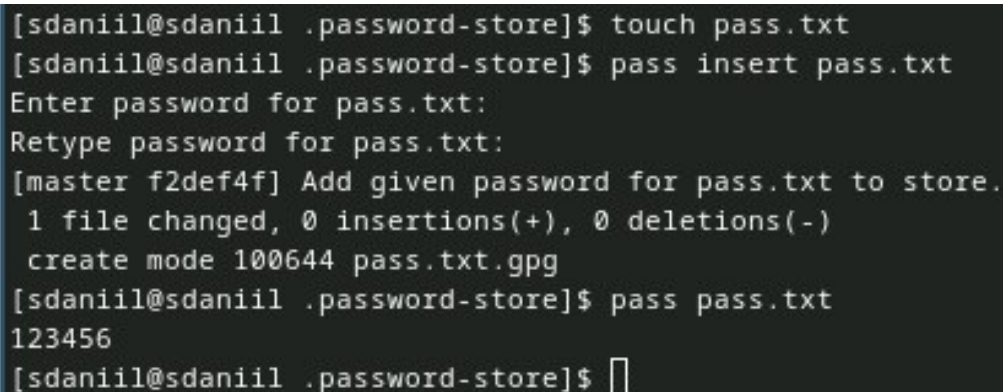
10) Сохранение пароля

Добавить новый пароль

Выполним:

```
pass insert [OPTIONAL DIR]/[FILENAME]
```

(рис. 4.12).



```
[sdaniil@sdaniil .password-store]$ touch pass.txt
[sdaniil@sdaniil .password-store]$ pass insert pass.txt
Enter password for pass.txt:
Retype password for pass.txt:
[master f2def4f] Add given password for pass.txt to store.
  1 file changed, 0 insertions(+), 0 deletions(-)
  create mode 100644 pass.txt.gpg
[sdaniil@sdaniil .password-store]$ pass pass.txt
123456
[sdaniil@sdaniil .password-store]$
```

Рис. 4.12: Добавление нового пароля

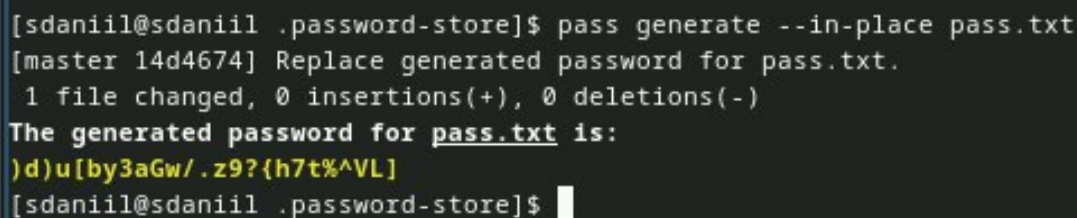
- 11) OPTIONAL DIR: необязательное имя каталога, определяющее файловую структуру для вашего хранилища паролей; FILENAME: имя файла, который будет использоваться для хранения пароля.

Отобразите пароль для указанного имени файла:

```
pass [OPTIONAL DIR]/[FILENAME]
```

Замените существующий пароль:

```
pass generate --in-place FILENAME (рис. 4.13).
```



```
[sdaniil@sdaniil .password-store]$ pass generate --in-place pass.txt
[master 14d4674] Replace generated password for pass.txt.
  1 file changed, 0 insertions(+), 0 deletions(-)
The generated password for pass.txt is:
)d)u[by3aGw/.z9?{h7t%^VL]
[sdaniil@sdaniil .password-store]$
```

Рис. 4.13: Замена существующего пароля

12) Установите дополнительное программное обеспечение:

`sudo dnf -y install dunst fontawesome-fonts powerline-fonts light fuzzel swaylock kitty waybar swaybg wl-clipboard mpv grim slurp` (рис. 4.14).

```
[sdaniil@sdaniil .password-store]$ sudo dnf -y install \
dunst \
fontawesome-fonts \
powerline-fonts \
light \
fuzzel \
swaylock \
kitty \
waybar swaybg \
wl-clipboard \
mpv \
grim \
slurp
[sudo] пароль для sdaniil:
Последняя проверка окончания срока действия метаданных: 0:11:27 назад, Сб 16 мар 2024 17:41:28.
Пакет dunst-1.9.2-2.fc39.x86_64 уже установлен.
Пакет light-1.2.2-10.fc39.x86_64 уже установлен.
Пакет swaylock-1.7.2-2.fc39.x86_64 уже установлен.
Пакет waybar-0.9.24-1.fc39.x86_64 уже установлен.
Пакет swaybg-1.2.0-3.fc39.x86_64 уже установлен.
Пакет wl-clipboard-2.2.1-1.fc39.x86_64 уже установлен.
Пакет mpv-0.36.0-3.fc39.x86_64 уже установлен.
Пакет grim-1.4.1-1.fc39.x86_64 уже установлен.
Пакет slurp-1.5.0-1.fc39.x86_64 уже установлен.
Зависимости разрешены.
=====
Пакет                Архитектура          Версия                Репозиторий  Размер
=====
Установка:
fontawesome4-fonts    noarch               1:4.7.0-19.fc39      fedora       204 k
fuzzel                x86_64              1.9.2-2.fc39         fedora       120 k
```

Рис. 4.14: Установка дополнительного ПО

13) Установим шрифты:

`sudo dnf copr enable peterwu/iosevka`
`sudo dnf search iosevka`

`sudo dnf install iosevka-fonts iosevka-aile-fonts iosevka-curly-fonts iosevka-slab-fonts iosevka-etoile-fonts iosevka-term-fonts` (рис. 4.15 4.16).

```
[sdaniil@sdaniil .password-store]$ sudo dnf copr enable peterwu/iosevka
Включение репозитория Copr. Обратите внимание, что этот репозиторий
не является частью основного дистрибутива, и качество может отличаться.

Проект Fedora не имеет какого-либо влияния на содержимое этого
репозитория за рамками правил, описанных в Вопросах и Ответах Copr в
<https://docs.pagure.org/copr.copr/user\_documentation.html#what-i-can-build-in-copr>,
а качество и безопасность пакетов не поддерживаются на каком-либо уровне.

Не отправляйте сообщения об ошибках этих пакетов в Fedora
Bugzilla. В случае возникновения проблем обращайтесь к владельцу этого репозито-
рия.

Do you really want to enable copr.fedorainfracloud.org/peterwu/iosevka? [y/N]:
y
Репозиторий успешно подключен.
[sdaniil@sdaniil .password-store]$ sudo dnf search iosevka
Copr repo for iosevka owned by peterwu          88 kB/s | 53 kB      00:00
===== Имя точное соответствие: iosevka =====
iosevka.src : Slender typeface for code, from code.
===== Имя совпадение: iosevka =====
iosevka-aile.src : Slender typeface for code, from code.
iosevka-aile-fonts.noarch : Quasi-proportional, Sans-serif
iosevka-curly.src : Slender typeface for code, from code.
iosevka-curly-fonts.noarch : Monospace, Curly Style
iosevka-curly-slab.src : Slender typeface for code, from code.
iosevka-curly-slab-fonts.noarch : Monospace, Curly Style, Slab-serif
iosevka-etoile.src : Slender typeface for code, from code.
iosevka-etoile-fonts.noarch : Quasi-proportional, Slab-serif
```

Рис. 4.15: Установка шрифтов


```

(4/6): iosevka-fonts-29.0.1-1.fc39.noarch.rpm 1.1 MB/s | 89 MB    01:18
(5/6): iosevka-slab-fonts-29.0.1-1.fc39.noarch 1.2 MB/s | 92 MB    01:15
(6/6): iosevka-term-fonts-29.0.1-1.fc39.noarch 1.2 MB/s | 91 MB    01:13
-----
Общий размер                               3.9 MB/s | 462 MB    01:57
Copr hero for iosevka owned by peterwu     9.3 kB/s | 1.0 kB    00:00
Импорт GPG-ключа 0xD02788C0:
Идентификатор пользователя: "peterwu_iosevka (None) <peterwu#iosevka@copr.fedorahosted.org>"
Отпечаток: CD9A 5D3D B3D7 AF70 C622 0342 B421 9F7F D027 88C0
Источник: https://download.copr.fedorainfracloud.org/results/peterwu/iosevka/pubkey.gpg
Продолжить? [д/н]: y
Импорт ключа успешно завершен
Проверка транзакции
Проверка транзакции успешно завершена.
Идет проверка транзакции
Тест транзакции проведен успешно.
Выполнение транзакции
  Подготовка      :                               1/1
  Установка       : iosevka-term-fonts-29.0.1-1.fc39.noarch 1/6
  Установка       : iosevka-slab-fonts-29.0.1-1.fc39.noarch 2/6
  Установка       : iosevka-fonts-29.0.1-1.fc39.noarch      3/6
  Установка       : iosevka-etoile-fonts-29.0.1-1.fc39.noarch 4/6
  Установка       : iosevka-curly-fonts-29.0.1-1.fc39.noarch 5/6
  Установка       : iosevka-aile-fonts-29.0.1-1.fc39.noarch 6/6
  Запуск скриптела: iosevka-aile-fonts-29.0.1-1.fc39.noarch 6/6
  Проверка        : iosevka-aile-fonts-29.0.1-1.fc39.noarch 1/6
  Проверка        : iosevka-curly-fonts-29.0.1-1.fc39.noarch 2/6
  Проверка        : iosevka-etoile-fonts-29.0.1-1.fc39.noarch 3/6
  Проверка        : iosevka-fonts-29.0.1-1.fc39.noarch      4/6
  Проверка        : iosevka-slab-fonts-29.0.1-1.fc39.noarch 5/6
  Проверка        : iosevka-term-fonts-29.0.1-1.fc39.noarch 6/6

Установлен:
iosevka-aile-fonts-29.0.1-1.fc39.noarch
iosevka-curly-fonts-29.0.1-1.fc39.noarch
iosevka-etoile-fonts-29.0.1-1.fc39.noarch
iosevka-fonts-29.0.1-1.fc39.noarch
iosevka-slab-fonts-29.0.1-1.fc39.noarch
iosevka-term-fonts-29.0.1-1.fc39.noarch

Выполнено!
[sdaniil@sdaniil .password-store]$

```

Рис. 4.16: Установка шрифтов

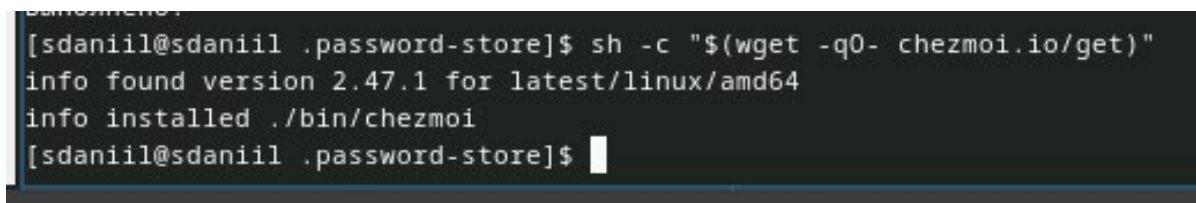
15) Установка бинарного файла. Скрипт определяет архитектуру процессора и

операционную систему и скачивает необходимый файл:

с помощью wget:

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

(рис. 4.17).



```
[sdaniil@sdaniil .password-store]$ sh -c "$(wget -qO- chezmoi.io/get)"
info found version 2.47.1 for latest/linux/amd64
info installed ./bin/chezmoi
[sdaniil@sdaniil .password-store]$
```

Рис. 4.17: Установка бинарного файла

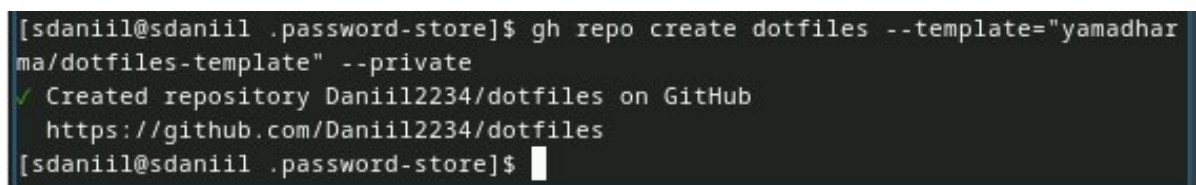
16) Создание собственного репозитория с помощью утилит

Будем использовать утилиты командной строки для работы с github.

Создадим свой репозиторий для конфигурационных файлов на основе шаблона:

```
gh repo create dotfiles --template="yamadharma/dotfiles-template" --private
```

4.18).



```
[sdaniil@sdaniil .password-store]$ gh repo create dotfiles --template="yamadhar
ma/dotfiles-template" --private
✓ Created repository Daniil2234/dotfiles on GitHub
https://github.com/Daniil2234/dotfiles
[sdaniil@sdaniil .password-store]$
```

Рис. 4.18: Создание репозитория для конфигурационных файлов

17) Подключение репозитория к своей системе

Инициализируйте chezmoi с вашим репозиторием dotfiles:

```
chezmoi init git@github.com:/dotfiles.git (рис. 4.19).
```

```
[sdaniil@sdaniil .password-store]$ chezmoi init git@github.com:Danii12234/dotfiles.git
Клонирование в «/home/sdaniil/.local/share/chezmoi»...
remote: Enumerating objects: 100, done.
remote: Counting objects: 100% (100/100), done.
remote: Compressing objects: 100% (87/87), done.
remote: Total 100 (delta 0), reused 99 (delta 0), pack-reused 0
Получение объектов: 100% (100/100), 77.86 КиБ | 857.00 КиБ/с, готово.
[sdaniil@sdaniil .password-store]$
```

Рис. 4.19: Создание релиза на github

- 18) Проверим, какие изменения внесёт chezmoi в домашний каталог, запустив:
(рис. 4.20).

```

+ a. For the avoidance of doubt, this Public License does not, and
+ shall not be interpreted to, reduce, limit, restrict, or impose
+ conditions on any use of the Licensed Material that could lawfully
+ be made without permission under this Public License.
+
+ b. To the extent possible, if any provision of this Public License is
+ deemed unenforceable, it shall be automatically reformed to the
+ minimum extent necessary to make it enforceable. If the provision
+ cannot be reformed, it shall be severed from this Public License
+ without affecting the enforceability of the remaining terms and
+ conditions.
+
+ c. No term or condition of this Public License will be waived and no
+ failure to comply consented to unless expressly agreed to by the
+ Licensor.
+
+ d. Nothing in this Public License constitutes or may be interpreted
+ as a limitation upon, or waiver of, any privileges and immunities
+ that apply to the Licensor or You, including from the legal
+ processes of any jurisdiction or authority.
+
+=====
+
+Creative Commons is not a party to its public
+licenses. Notwithstanding, Creative Commons may elect to apply one of
+its public licenses to material it publishes and in those instances
+will be considered the "Licensor." The text of the Creative Commons
+public licenses is dedicated to the public domain under the CC0 Public
+Domain Dedication. Except for the limited purpose of indicating that
+material is shared under a Creative Commons public license or as
+otherwise permitted by the Creative Commons policies published at
+creativecommons.org/policies, Creative Commons does not authorize the
+use of the trademark "Creative Commons" or any other trademark or logo
+of Creative Commons without its prior written consent including,
+without limitation, in connection with any unauthorized modifications
+to any of its public licenses or any other arrangements,
+understandings, or agreements concerning use of licensed material. For
+the avoidance of doubt, this paragraph does not form part of the
+public licenses.
+
+Creative Commons may be contacted at creativecommons.org.
+
+sdanil@sdanil - password store$

```

Рис. 4.20: Проверка изменений

19) Запуск:

chezmoi apply -v (рис. 4.21).

```
+Domain Dedication. Except for the limited purpose of indicating that
+material is shared under a Creative Commons public license or as
+otherwise permitted by the Creative Commons policies published at
+creativecommons.org/policies, Creative Commons does not authorize the
+use of the trademark "Creative Commons" or any other trademark or logo
+of Creative Commons without its prior written consent including,
+without limitation, in connection with any unauthorized modifications
+to any of its public licenses or any other arrangements,
+understandings, or agreements concerning use of licensed material. For
+the avoidance of doubt, this paragraph does not form part of the
+public licenses.
+
+Creative Commons may be contacted at creativecommons.org.
+
[sdaniil@sdaniil .password-store]$
```

Рис. 4.21: Запуск chezmoi

20) Использование chezmoi на нескольких машинах

На второй машине инициализируйте chezmoi с вашим репозиторием dotfiles:

```
chezmoi init https://github.com/<username>/dotfiles.git
```

Или через ssh:

```
chezmoi init git@github.com:<username>/dotfiles.git
```

(рис. 4.22).

```
dasedokhin@dasedokhin-VirtualBox:~$ sh -c "$(wget -qO- chezmoi.io/get)"
info found version 2.47.1 for latest/linux/amd64
info installed ./bin/chezmoi
dasedokhin@dasedokhin-VirtualBox:~$ sudo cp ./bin/chezmoi usr/local/bin
[sudo] пароль для dasedokhin:
cp: невозможно создать обычный файл 'usr/local/bin': Нет такого файла или каталога
dasedokhin@dasedokhin-VirtualBox:~$ sudo cp ./bin/chezmoi /usr/local/bin
dasedokhin@dasedokhin-VirtualBox:~$ chezmoi init git@github.com:Danil2234/dotfiles.git
Клонирование в «/home/dasedokhin/.local/share/chezmoi»...
remote: Enumerating objects: 100, done.
remote: Counting objects: 100% (100/100), done.
remote: Compressing objects: 100% (87/87), done.
remote: Total 100 (delta 0), reused 99 (delta 0), pack-reused 0
Получение объектов: 100% (100/100), 77.86 КиБ | 830.00 КиБ/с, готово.
dasedokhin@dasedokhin-VirtualBox:~$
```

Рис. 4.22: Инициализация chezmoi на второй машине

21) Проверим, какие изменения внесёт chezmoi в домашний каталог, запустив:

chezmoi diff (рис. ??).

![Запуск chezmoi diff](image/23.jpg){#fig:0023 width=100%}

22) Запустим

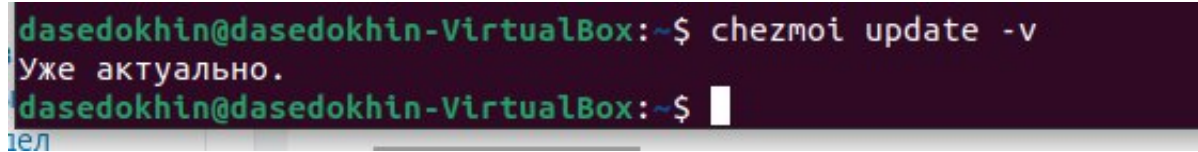
chezmoi apply -v (рис. 4.23).

```
+material is shared under a Creative Commons public license or as
+otherwise permitted by the Creative Commons policies published at
+creativecommons.org/policies, Creative Commons does not authorize the
+use of the trademark "Creative Commons" or any other trademark or logo
+of Creative Commons without its prior written consent including,
+without limitation, in connection with any unauthorized modifications
+to any of its public licenses or any other arrangements,
+understandings, or agreements concerning use of licensed material. For
+the avoidance of doubt, this paragraph does not form part of the
+public licenses.
+
+Creative Commons may be contacted at creativecommons.org.
dasedokhin@dasedokhin-VirtualBox:~$
```

Рис. 4.23: Запуск chezmoi

23) При существующем каталоге chezmoi можно получить и применить последние изменения из вашего репозитория:

`chezmoi update -v` (рис. 4.24).



```
dasedokhin@dasedokhin-VirtualBox:~$ chezmoi update -v
Уже актуально.
dasedokhin@dasedokhin-VirtualBox:~$
```

Рис. 4.24: Применение изменений

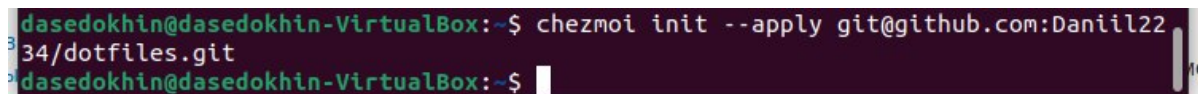
24) Настройка новой машины с помощью одной команды

Можно установить свои dotfiles на новый компьютер с помощью одной команды:

`chezmoi init --apply https://github.com//dotfiles.git`

Через ssh:

`chezmoi init --apply git@github.com:/dotfiles.git` (рис. 4.25).



```
dasedokhin@dasedokhin-VirtualBox:~$ chezmoi init --apply git@github.com:Danil2234/dotfiles.git
dasedokhin@dasedokhin-VirtualBox:~$
```

Рис. 4.25: Установка dotfiles

25) Ежедневные операции с chezmoi

Извлеките последние изменения из репозитория и примените их

Можно извлечь изменения из репозитория и применить их одной командой:

`chezmoi update` (рис. [-@fig:0027]).

```
[sdaniil@sdaniil .password-store]$ chezmoi update
Уже актуально.
[sdaniil@sdaniil .password-store]$
```

Рис. 4.26: Извлечение изменений

Это запускается `git pull --autostash --rebase` в вашем исходном каталоге, а затем `chezmoi apply`.

- 26) Извлеките последние изменения из своего репозитория и посмотрите, что изменится, фактически не применяя изменения

Выполним:

```
chezmoi git pull --autostash --rebase && chezmoi diff
```

Это запускается `git pull --autostash --rebase` в вашем исходном каталоге, а `chezmoi diff` затем показывает разницу между целевым состоянием, вычисленным из вашего исходного каталога, и фактическим состоянием.

Если вы довольны изменениями, вы можете применить их:

`chezmoi apply` (рис. 4.27).

```
[sdaniil@sdaniil .password-store]$ chezmoi git pull -- --autostash --rebase &&
chezmoi diff
Уже актуально.
[sdaniil@sdaniil .password-store]$ chezmoi apply
[sdaniil@sdaniil .password-store]$
```

Рис. 4.27: Извлечение изменений

- 27) Автоматически фиксируйте и отправляйте изменения в репозиторий

Можно автоматически фиксировать и отправлять изменения в исходный каталог в репозиторий.

Эта функция отключена по умолчанию.

Чтобы включить её, добавьте в файл конфигурации ~/.config/chezmoi/chezmoi.toml следующее:

```
[git] autoCommit = true autoPush = true
```

Всякий раз, когда в исходный каталог вносятся изменения, chezmoi фиксирует изменения с помощью автоматически сгенерированного сообщения фиксации и отправляет их в ваш репозиторий.

Будьте осторожны при использовании autoPush. Если ваш репозиторий dotfiles является общедоступным, и вы случайно добавили секрет в виде обычного текста, этот секрет будет отправлен в ваш общедоступный репозиторий. (рис. 4.28).



```
GNU nano 7.2 /home/sdaniil/.config/chezmoi/chezmoi.toml
[data]
email = "sedokhin.daniil@gmail.com"

[git]
  autoCommit = true
  autoPush = true
```

Рис. 4.28: Автоматическая фиксация и отправление изменений в репозиторий

5 Выводы

Я Получил навыки настройки рабочей среды.