

*Общество с ограниченной ответственностью Управляющая компания «Нектарин»  
ОГРН 1087746039886, ИНН 7710703730  
430030, Россия, Республика Мордовия, г. Саранск, ул. Васенко, д. 9, этаж 4, помещение 1.*

**ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ  
«Admify 2.0»**

**Инструкция по установке программного обеспечения**

на 8 листах

2022 г.

## 1. ОБЩИЕ СВЕДЕНИЯ

Документ содержит описание порядка установки программного обеспечения «Admify 2.0» (далее — ПО, **Admify 2.0**) и включает информацию по установке и настройке данной программы для ЭВМ.

ПО представляет собой облачную платформу (интернет-сервис, распространяемый по модели SaaS) для автоматизации реферального маркетинга с применением искусственного интеллекта для создания и проведения рекламных кампаний.

## 2. ТРЕБОВАНИЯ К ОБОРУДОВАНИЮ

Для получения доступа к ПО не требуется специальное оборудование. ПО может быть скомпилировано и запущено на следующих ОС:

- Windows (7 или выше)
- Apple macOS
- Linux (любой дистрибутив на архитектуре i386, amd64, arm64, armv6l)
- FreeBSD (на архитектуре i384, amd64)

## 3. ПОРЯДОК УСТАНОВКИ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

Для работы с ПО эксперту необходимо развернуть систему.

Рекомендуемая ОС — Linux (debian).

**Ссылка на образ диска виртуальной машины с yandex-cloud:**

<https://mail.ali33.ru/demo-admify-ru-vm.iso>

**user:** admify

**password:** NectarinOwner2022\$

Данный образ является образом диска виртуальной машины, который может быть запущен на любой облачной платформе или на локальной виртуальной локальной машине на базе openvz или hyper-v

Доступ по ssh в виде удалённого доступа к инфраструктуре с развёрнутым экземпляром ПО (VPN, SSH, RDP и т.п.) с подробным описанием процессов, микроконтэйнеров, расположения файлов ПО, его компонентов и т.п.:

host: demo.admify.pro

user: admify

password: NectarinOwner2022\$

Доступ по ssh:

user: admify

password: NectarinOwner2022\$

Описание процессов:

1) admify-application — приложение Admify, которое, включает в себя http сервер сайтов домашней страницы (homepage), личного кабинета, REST API (api)

2) PostgreSQL — сервер базы данных

3) Asterisk — сервер SIP-телефонии

## Инструкция по сборке в ОС Linux.

Данная инструкция расположена на git проекта <https://git.ali33.ru/fcg-xvii/admify-release>.

Логин: **nectarin**, пароль: **NectarinOwner2022\$**

### Необходимое ПО:

- **Golang** (<https://go.dev/dl/>)
  - **Asterisk** (<https://www.asterisk.org/>)
  - **Python3** (<https://www.python.org/>)
  - **PostgreSQL** (<https://postgrespro.ru/docs/postgresql/14/index>)
  - **plpython3u** (<https://postgrespro.ru/docs/postgresql/14/plpython-python23>)
  - **git** (<https://git-scm.com/>)
  - **pyinstaller** (<https://pyinstaller.org/en/stable/>)
- 

### **Git**

Необходимо внести в файл **/home/[user]/.netrc** следующие изменения:

```
machine git.ali33.ru login nectarin password NectarinOwner2022$
```

Далее необходимо клонировать репозиторий приложения

```
# git clone https://git.ali33.ru/fcg-xvii/admify-release
```

---

### **Настройка asterisk**

Необходимо скопировать файлы конфигурации из директории приложения **[app\_dir]/backend/config/asterisk** в директорию конфигурации asterisk **/etc/asterisk**

```
# sudo cp -rf [app_dir]/backend/config/asterisk/* /etc/asterisk
```

Устанавливаем владельца для всех файлов конфигурации пользователя **asterisk**

```
# sudo chown -R asterisk:asterisk /etc/asterisk
```

### **Asterisk REST API (ARI)**

Необходимо указать имя пользователя и пароль в файле **/etc/asterisk/ari.conf**:

```
:[admify]
;type = user
;read_only = no
;password = *****
```

Имя пользователя указывается в квадратных скобках. Так же необходимо убрать символы **[:]** в начале каждой строки блока. Для примера укажем имя пользователя **admify-user**, пароль **mypass12345**, в результате блок должен соответствовать:

```
[admify-user]
type = user
read_only = no
password = mypass12345
```

### **Extensions (диалпланы)**

Диалпланы расположены в директории **/etc/asterisk/extensions.d**:

- admify.conf ..... выполняется приложением
- call-in.conf ..... для входящих звонков
- call-out.conf ..... для исходящих звонков
- ivr-main.conf ..... меню для входящих звонков (по умолчанию отключено)
- general.conf ..... глобальные настройки

Для работы приложения достаточно настроек по умолчанию.

### SIP (настройки SIP-абонентов и SIP-транка)

Настройки расположены в директории `/etc/asterisk/sip.d`

- general.conf ..... глобальные настройки
  - localnet [ ip-subnet/netmask ] ..... локальная сеть (например, 192.168.0.0/255.255.255.0)
  - externaddr [ host:port ] ..... внешний ip адрес (например, 1.1.1.1:5060)
  - register [ trunk-link ] ..... адрес транк-провайдера (например, user:password@host/user)
- external.conf ..... настройки внешнего транка
  - secret [ password ] ..... пароль
  - username [ username ] ..... имя пользователя
  - host [ host ] ..... имя хоста
  - fromuser [ username ] ..... имя пользователя, передаваемое внешнему транку (рекомендуется идентичное username)
  - fromdomain [ host ] ..... имя хоста, передаваемое внешнему транку (рекомендуется идентичное host)
- groups.conf ..... настройки групп
  - home ..... группа локальных абонентов (по умолчанию отключена)
- point777.conf ..... пример настройки локального абонента (в примере абонент имеет внешний номер 777, по умолчанию отключен. для включения необходимо предварительно включить группу home в настройках групп)

Для работы приложения необходимо указать в файле `/etc/asterisk/sip.d/general.conf` настройки локальной сети, внешний ip адрес и регистрацию внешнего транка. Так же необходимо указать настройки внешнего транка в файле `/etc/asterisk/sip.d/external.conf`.

### Asterisk Manager API (AMI)

Настройки расположены в директории `/etc/asterisk/manager.d`

- admin.conf ..... настройки пользователя admin, необходимо задать пароль (в поле secret) и раскомментировать блок

### Подготовка базы данных PostgreSQL.

Сначала необходимо создать пользователя, который будет управлять базой данных, а так же будет являться её владельцем. Ниже приводится пример создания пользователя. Для упрощения пользователю назначаются права superuser. Команды ниже вызываются из терминала linux от пользователя postgres:

```
postgres@# psql
postgres=# create user [user_name] with login superuser password '[user_password]';
postgres=# create database [database_name] with owner [user_name];
```

Добавим необходимые расширения и произведем заливку структуры базы данных из файла `[app_dir]/backend/db-dump/admify.sql`

```
# psql [database_name] [user_name]
[database_name]=# create extension plpython3u;
[database_name]=# create extension jsonb_plpython3u;
```

```
[database_name]=# exit
# psql [database_name] [user_name] < [app_dir]/backend/db-dump/admify.sql
```

Так же необходимо указать настройки доменных имен и почтового сервера в базе данных:

```
// обновление настроек почтового сервера
# psql [database_name] [user_name]
[database_name]=# update tool.conf set value = '{
  "host": "[host]",
  "port": [port],
  "user": "[username]",
  "password": "[password]"
}' where name = 'email_auth';
// обновление доменных имен (указываются только главные зеркала)
[database_name]=# update tool.conf set value = '{
  "cns1": "[domain_cns1]",
  "homepage": "[domain_homepage]"
}' where name = 'domain';
// обновление почтовых ящиков администраторов
[database_name]=# update tool.conf set value = '[
  "email1@mail.mail",
  "email2@mail.mail",
]' where name = 'emails_admin';
```

Далее необходимо скопировать python скрипты в одну из директорий видимости python модулей. Список таких директорий можно получить Копирование скриптов plpython3u. Необходимо скопировать содержимое директории **[app\_dir]/backend/plpython3u** в **/usr/lib/python3/dist-packages**

```
# sudo cp -rf [app_dir]/backend/plpython3u/* /usr/lib/python3/dist-packages
```

Так же необходимо установить модуль **numpy**

```
# sudo pip3 install numpy
```

---

## Сборка микросервисов

Исходники микросервисов находятся в директории **[app\_dir]/backend/mk-services**

### google-sheets

Данный микросервис реализует экспорт документов приложения в таблицы google sheets. Для работы микросервиса предварительно необходимо создать приложение google на платформе google cloud (<https://cloud.google.com>), создать ключ сервисного аккаунта (он понадобится для доступа микросервиса к работе с приложением google). Подробнее о создании приложения <https://cloud.google.com/resource-manager/docs/creating-managing-projects>. Предполагается, что приложение и сервисный аккаунт уже зарегистрированы. При регистрации сервисного ключа приложения Вы получите файл **credentials.json** с описанием параметров доступа. Необходимо перейти в директорию микросервиса и собрать исполняемый файл при помощи **pyinstaller**

```
# cd [app_dir]/backend/mk-services/google-sheets
# pyinstaller --onefile app.py
```

Далее необходимо скопировать исполняемый файл и шаблон файла конфигурации в рабочую директорию микросервисов **[app\_dir]/mk-service/google-sheets** (если директории не существует, её необходимо создать):

```
# cp [app_dir]/backend/mk-services/google-sheets/dist/app [app_dir]/mk-service/google-sheets/
```

Так же в директорию [app\_dir]/mk-service/google-sheets необходимо скопировать файл credentials.json, полученного при создании сервисного ключа приложения google.

### Подключение рекламных каналов (социальных сетей)

Для подключения социальных сетей необходимо зарегистрировать соответствующие приложения и указать параметры приложений в файле настроек [app\_dir]/frontend/cnsl/ad/config.script

### Подключение VK (ВКонтакте)

Необходимо зарегистрировать приложение на платформе VK <https://vk.com/apps?act=manage>:

- Тип приложения - "Сайт"
- Адрес сайта - [https://\[homepage\\_domain\]](https://[homepage_domain])
- Базовый домен - [homepage\_domain]

В настройках приложения следует указать:

- Пользов. соглашение - [https://\[homepage\\_domain\]/info/user\\_agreement.html](https://[homepage_domain]/info/user_agreement.html)
- Политика конфиденциальности - [https://\[homepage\\_domain\]/info/private\\_policy.html](https://[homepage_domain]/info/private_policy.html)
- Базовый домен - необходимо добавить домены [homepage\_domain], [cnsl\_domain]
- Доверенный URL - добавить два URL: [https://\[cnsl\\_domain\]/ad/vk/auth.script](https://[cnsl_domain]/ad/vk/auth.script),  
[https://\[cnsl\\_domain\]/ad/vk/login.script](https://[cnsl_domain]/ad/vk/login.script)

Далее необходимо внести изменения в файл настроек [app\_dir]/frontend/cnsl/ad/config.script, в объект поля **vk**

```
vk: {  
  appID: '0000000', // ID приложения  
  appSecret: '*****', // Защищённый ключ  
  appServiceKey: '*****', // Сервисный  
  ключ доступа  
  authRedirectURI: 'https://[cnsl_domain]/ad/vk/auth.script',  
  loginRedirectURI: 'https://[cnsl_domain]/ad/vk/login.script'  
}
```

### Сборка Admify Application

Для запуска процесса сборки необходимо запустить скрипт [app\_dir]/build.sh

```
# cd [app_dir]  
# ./build.sh
```

На завершающем этапе сборки программа спросит, необходима ли приложению возможность прослушивать 80 и 443 порты. Если это необходимо, возможно (в зависимости от дистрибутива), потребуется ввести пароль суперпользователя. Если этого не требуется, ответьте "n". После завершения сборки будет создан исполняемый файл [app\_dir]/admify\_application

---

### Настройка приложения и запуск

Перед запуском приложения необходимо настроить конфигурационный файл [app\_dir]/config.cfg

```
listen_address = 10.129.0.7 # ..... Внешний адрес, который будет слушать веб сервер  
listen_port_http = 80 # ..... Порт http  
listen_port_https = 443 # ..... Порт https
```

homepage\_domains = www.admify.pro, admify.pro # .... Домены сайта. Первый - главное зеркало  
cnsl\_domains = cnsl.admify.pro # ..... Домены консоли. Первый - главное зеркало

# Параметры соединения с базой данных

db\_connection = postgres://user:password@127.0.0.1/dbname?port=3345

ad\_stats\_script\_url = /ad/ad-stats.script # ... Ссылка на скрипт обработчика статистики внешних каналов, рекомендуется оставить текущее значение

reports\_script\_url = /ad/report.script #..... Ссылка на скрипт обработчика отчетов, рекомендуется оставить текущее значение

# Asterisk AMI

ami\_host = 127.0.0.1:5002 # ..... Адрес подключения к Asterisk AMI

ami\_login = admin # ..... логин

ami\_password = \*\*\*\*\* # ..... пароль

ami\_bot\_context = admify # ..... контекст диалплана для верификации роботом

ami\_bot\_exten = s # ..... расширение контекста верификации роботом

ami\_bot\_timeout\_msec = 20000 # ..... время ожидания ответа абонента в миллисекундах при звонке робота

ami\_manager\_context = admify # ..... контекст диалплана для верификации менеджером

ami\_manager\_exten = s # ..... расширение контекста верификации менеджером

ami\_manager\_timeout\_msec = 20000 # ..... время ожидания ответа абонента в миллисекундах при звонке менеджера

# Asterisk ARI

ari\_url = http://localhost:8088/ari # ..... URL сервиса Asterisk ARI

ari\_username = admify # ..... имя пользователя

ari\_password = \*\*\*\*\* # ..... пароль

ari\_websocket\_url = ws://localhost:8088/ari/events # ..... URL открытия вебсокета

ari\_bot\_application = admify # ..... наименование приложения верификации роботом в контексте диалплана

ari\_manager\_application = admify # ..... наименование приложения верификации менеджером в контексте диалплана

# yandex speech kit

yask\_folder\_id = b1\_01e2nlall # ..... yandex folder id

yask\_api\_key\_id = aj\_qf5lf # ..... yandex api key id

yask\_api\_key = \_wCKx7jOhcnPTFcM8V9TU9 # .... yandex api key

# hooks. Настройки телеграм бота. Бот пишет сообщения с описанием ошибок в системе

hook\_telegram\_bot\_token = \*\*\*\*\* # ..... токен бота телеграм

# for group id - [https://api.telegram.org/bot\[your\\_owersame\\_token\]/getUpdates](https://api.telegram.org/bot[your_owersame_token]/getUpdates)

hook\_telegram\_chat\_id = \*\*\*\*\* # ..... идентификатор группы для сообщений бота

# storage dir

storage\_path = storage

Далее рекомендуется осуществить пробный запуск приложения для проверки корректности подключения приложения к внешним сервисам:

```
# cd [app_dir]
```

```
# ./admify-application
```

```
2022/06/25 17:23:42 secret generated...
```

```
52fdcf072182654f163f5f0f9a621d729566c74d10037c4d7bbb0407d1e2c649
```

```
INFO[0000] Database connection test...
```

