

Установка Loginom в систему контейнеризации Podman или Docker

Дистрибутив для установки Loginom на OS Linux поставляется в виде архива `Loginom-*.tar.gz`, в состав которого входят: бинарные файлы, файлы библиотек, конфигурационные файлы, бинарный файл BatchLauncher и параметризуемый Bash-скрипт по формированию docker образов и конфига docker-compose.

Наличие Bash-скрипта дает клиенту возможность самостоятельной кастомизации, т.е. клиент может сам дополнить Dockerfile нужными конкретно ему библиотеками или программами, добавить в будущий образ свой ODBC драйвер, включить в сборку свою версию Python.

Варианты установки

Loginom может быть установлен с использованием систем контейнеризации Podman или Docker. Рекомендуемым контейнерным движком является Podman.

В случае, если инсталлятор одновременно находит и podman и docker пользователю будет задан вопрос: Выберите систему контейнеризации. Вариантом, предлагаемым по умолчанию является Podman.

Основные отличия установок:

- Для **Podman** контейнеры будут работать от имени текущего пользователя. Рабочая директория по умолчанию `$HOME/loginom`. Автозапуск конфигурации после перезагрузки хоста обеспечивается через cron.
- Для **Docker** система контейнеризации и сами контейнеры будут работать от пользователя **root**. Рабочая директория по умолчанию `/var/opt/loginom-docker`.

Минимально необходимые версии ПО:

- для **Podman** — версия не ниже `3.4.0`, и **Podman-compose** — версия `1.0.3` ;
- для **Docker** — версия не ниже `20.10.1`, и **Docker-compose** — версия `1.21.0`.

Предустановка

Перед установкой Loginom необходимо предварительно установить выбранную систему контейнеризации (рекомендуется Podman), а также произвести некоторые настройки в зависимости от операционной системы.

Режим контейнеризации Docker:

- [Astra Linux 1.7](#)

Режим контейнеризации Podman:

- [Debian 11.5 и Ubuntu 22.04](#)
- [Fedora 36](#)
- [OpenSUSE 15.4](#)
- [РЕД ОС 7.3.2](#)

Внимание: если предварительные настройки не будут произведены, то далее не получится корректно установить Loginom.

Базовый образ

Наличие Bash-скрипта позволяет пользователю изменить базовый образ. Например, по требованию специалистов информационной безопасности, можно произвести сборку сервера Loginom с применением более свежего базового образа на основе `ubuntu` вместо используемого по умолчанию `debian`. При сборке сервера инсталлятор будет запрашивать на какой операционной системе собрать образ. Сейчас на выбор предлагаются:

- `debian:11.5-slim`
- `ubuntu:22.04`
- `ubuntu:23.04`.

Loginom на данных ОС протестирован и точно работает. Если необходимо, то пользователь вручную может задать нужную ему ОС отличную от представленных.

Установка

Скопировать любым способом дистрибутив Loginom в виде архива `**loginom-*.tar.gz**` на целевую систему, например, в домашнюю папку нашего пользователя `$HOME: /home/user`.

Распаковать архив и перейти в распакованный каталог командой:

```
mkdir loginom-dist && tar -xf loginom.tar.gz -C loginom-dist/ && cd loginom-dist
```

Дополнительные компоненты

Для включения в образ Loginom дополнительных компонентов, например, ODBC драйверов или модуля Python, нужно внести изменения в файл-сценарий сборки образа сервера `server/Dockerfile`. В данном файле необходимо раскомментировать строки с соответствующими командами.

```
1 # Установка Microsoft ODBC Driver 18 for SQL Server {#ustanovka-microsoft-
odbc-driver-18-for-sql-server}
2 # Файл msodbcsql18_18.0.1.1-1_amd64.deb должен лежать в каталоге server/libs
дистрибутива {#fayl-msodbcsql1818011-1amd64deb-dolzhen-lezhat-v-kataloge-
serverlibs-distributiva}
3 #RUN apt-get update && DEBIAN_FRONTEND=noninteractive ACCEPT_EULA=Y apt-get
install -y /app/msodbcsql18_18.0.1.1-1_amd64.deb
4
5 # Установка Python и некоторых его библиотек {#ustanovka-python-i-
nekotorykh-ego-bibliotek}
6 #RUN apt-get install -y python3 python3-pip; \
7 #pip3 install -U pandas numpy scikit-learn
```

Запуск установщика для формирования образов контейнеров и конфигурационных файлов осуществляется с помощью команды:

- Для **Podman**:

```
./setup.sh
```

- Для **Docker**:

```
sudo ./setup.sh
```

Настройки портов

При наличии firewall, открываем необходимые (в зависимости от выбранной конфигурации) порты. Например, для конфигурации по умолчанию в системах, использующих firewall-cmd:

- порт http:

```
sudo firewall-cmd --permanent --zone=public --add-port=80/tcp
```

- порт https:

```
sudo firewall-cmd --permanent --zone=public --add-port=443/tcp`
```

- порт Loginom для подключения websocket:

```
sudo firewall-cmd --permanent --zone=public --add-port=8080/tcp
```

- порт Loginom для подключения защищенного websocket:

```
sudo firewall-cmd --permanent --zone=public --add-port=8443/tcp
```

- порт для подключения к серверу Loginom для Batchlauncher или внешнего Интегратора:

```
sudo firewall-cmd --permanent --zone=public --add-port=4580/tcp
```

- перезагружаем правила firewall:

```
sudo firewall-cmd --reload
```

Запуск Loginom

Внимание: для коммерческого применения Loginom необходимо установить и активировать Лицензионный ключ и Сервер лицензий.

Перед запуском службы `loginomd` в строке `IP_NAME=` в файле, расположенном по умолчанию `/home/user/loginom/server/gnclient.ini`, указать ip-адрес машины.

Создать из образов Loginom соответствующие контейнеры и запустить их можно с помощью команды:

- Для **Podman**:

```
podman-compose -f ${Путь_к_каталогу_конфигураций}/docker-compose.yml up -d
```

При выборе значений по умолчанию команда будет следующей:

```
podman-compose -f ${HOME}/loginom/docker-compose.yml up -d
```

- Для **Docker**:

```
pushd /var/opt/loginom-docker; sudo docker-compose up -d; popd
```

Для **Podman** при выборе значений по умолчанию для текущего пользователя сформированные скриптом установки конфигурации формирования и запуска контейнеров будут находиться в каталоге: `${HOME}/loginom`.

Изменяемые файлы конфигураций контейнеров и пользовательские данные будут находиться соответственно в:

- `${HOME}/loginom/server` ;
- `${HOME}/loginom/integrator` ;
- `${HOME}/loginom/http` .

Также скрипт установки добавляет в пользовательский `crontab` запуск конфигурации при перезагрузке хоста.

Для **Docker** конфигурации и файлы будут находиться: `/var/opt/loginom-docker/` .

Установка Loginom с помощью готовых образов контейнеров

Если на целевой машине доступ в Интернет ограничен, его нет, или машина находится за прокси сервером, то для установки Loginom в этом случае необходимы готовые образы. Сформировать готовые образы можно с помощью ПК с доступом в интернет или получить их, запросив в технической поддержке Loginom.

- На ПК с доступом в интернет (который не за прокси) нужно запустить установку loginom скриптом `setup.sh` .
- Задавая нужные значения для параметров, сформировать образы loginom, после чего в конце установки появится пункт `Сохранить собранные образы отдельными файлами? [y/n]` . Ввести `y` , чтобы согласиться.
- Файлы сохранятся в домашнюю папку по следующему пути: `loginom/images/` .
- Полученные образы `loginom-http_date.tar` , `loginom-server_date.tar` , `loginom-integrator_date.tar` (где `_date` — дата и время формирования образа) на целевой машине нужно положить по следующему пути, при необходимости создав недостающие каталоги: `/home/user/loginom/images/` .
- На целевой машине запустить скрипт установки loginom `setup.sh` .
- Дойти до пункта `"Выберите источник для образов Loginom"` .
- Для использования готовых образов выбрать пункт `"[2] Использовать предварительно подготовленные образы"` . Скрипт установки должен самостоятельно найти образы в каталоге `/home/user/loginom/images/` или в папке `images` в самом дистрибутиве (рядом с `setup.sh`) последовательно для каждого приложения, пользователю нужно лишь подтвердить их расположение и название.

Сборка конкретных образов Loginom

Файл `/loginom-dist/setup.sh` собирает все образы Loginom без возможности выбрать конкретный образ для сборки.

При необходимости, конкретный образ Loginom можно пересобрать с помощью следующей команды:

```
podman build http -t loginom-http
```

Где `loginom-http` — название образа.

Команда должна выполняться из каталога `/loginom-dist/`, содержащего каталоги `/http/`, `/integrator/`, `/server/`.

Имена образов/контейнеров

- `loginom-http_date`;
- `loginom-server_date`;
- `loginom-integrator_date`.

Где `_date` — дата и время формирования образа.

Аргументы, принимаемые скриптом `setup.sh`

- `-h --help` — показать подсказку (несовместимо ни с какими другими аргументами);
- `-d --dir` — задание каталогов `_${CONF}` и `_${DATA}`;
- `--datadir` — если задан этот параметр, то он переопределяет задание каталога `_${DATA}`;
- `-s --save` — сохранение собранных образов в локальные файлы на диск (несовместимо с `--nosave`);
- `--nosave` — не сохранять образы дисков в локальные файлы (несовместимо с `--save`);
- `-c --change` — изменение установленных компонентов Loginom (несовместимо с `--update`);
- `** -u --update` — обновление образов Loginom без изменения конфигурации (несовместимо с `--change` и `--nobuild`);
- `-b --build` — перестроить образы, собрав их из исходников;
- `-l --load` — загрузить образы в систему из локальных файлов;
- `-n --nobuild` — не перестраивать существующие образы (несовместимо с `--update`);
- `--defaults` — где возможно, применять значения по умолчанию. Значения применяются на этапе ответов пользователя, вместо таковых. Аргумент имеет наименьший приоритет и может частично переопределяться остальными аргументами.

За исключением несовместимых опций, описанных выше, аргументы допустимо записывать в любом порядке, примеры: `-cbs` или `-b -c --save`, или `-sb --change`.

Полезные команды

Примеры для системы контейнеризации Podman. Для Docker-а используется `docker-compose` с аналогичными параметрами:

- `podman-compose ps` — список работающих контейнеров;
 - `podman-compose up -d` — собрать/compose и запустить контейнеры в "детач" (отсоединенном от консоли) режиме;
 - `podman-compose start` — старт контейнеров;
 - `podman-compose stop` — стоп контейнеров;
 - `podman-compose rm` — удалить контейнеры;
 - `podman-compose down` — остановить и удалить контейнеры;
 - `podman start ${CONTAINER_NAME}` — пример операции с отдельным контейнером (запуск);
 - `journalctl -n 10 -o verbose _COMM=loginomd` — логи программы loginomd, 10 последних записей в подробном формате;
 - `journalctl -n 10 -o verbose _COMM=Integrator` — логи программы Integrator, 10 последних записей в подробном формате;
 - `journalctl -b -f -n 30 CONTAINER_NAME=${CONTAINER_NAME}` — логи контейнера, 30 последних записей онлайн;
 - `podman exec -it ${CONTAINER_NAME} /bin/bash` — зайти внутрь контейнера (либо `/bin/sh` вместо `/bin/bash`).
-

Статьи в разделе:

- [Astra Linux 1.7](#)
- [Debian 11.5 и Ubuntu 22.04](#)
- [Fedora 36](#)
- [OpenSUSE 15.4](#)
- [РЕД ОС 7.3.2](#)