

A2P Platform

Руководство по установке

Содержание

| | |
|---|----------|
| Общие сведения | 3 |
| Описание процедуры установки Платформы | 3 |
| <i>1. Помощь в установке</i> | 3 |
| <i>2. Установка открытого ПО</i> | 3 |
| <i>3. Установка Платформы</i> | 4 |

Общие сведения

Платформа поставляется в виде исходного кода. Это позволяет скомпилировать и установить платформу на большее количество операционных систем, чем если бы она была скомпилирована на какой-то конкретной операционной системе.

Компиляция и установка на целевую ОС автоматизирована. От оператора требуется только выполнить команду. Подготовка целевой ОС не автоматизирована, так как в каждой ОС есть особенности.

Описание процедуры установки Платформы

1. Помощь в установке

Служба поддержки ООО «ТДМ ТЕХ»: телефон +79776290907, email: a2p@tdm-tech.ru

2. Установка открытого ПО

На целевые сервера необходимо установить следующее ПО, необходимое для функционирования платформы:

- Сервер реляционных баз данных Pangolin
- Сервер колоночных баз данных ClickHouse
- Сервер простых структур данных Redis
- Распределенное файловое хранилище HDFS
- Файловый сервер WebDAV
- Распределенную шину данных Kafka с модулем REST
- Сервер аутентификации KeyCloak

На целевые сервера, на которых будут работать непосредственно компоненты платформы, необходимо установить следующие пакеты (их названия могут незначительно отличаться в разных операционных системах):

- file
- geos
- ImageMagick
- openssl
- libyaml
- libffi
- nginx
- zlib
- libxslt
- libpq
- init-exporter

Также на целевых серверах необходимо создать пользователей:

- `deployer` – От его имени на сервере будут разворачиваться релизы компонент платформы. У него должны быть права на выполнение следующих команд:
 - `sudo init-exporter` – для создания юнитов `systemd` для запуска компонент
 - `sudo systemctl` – для остановки и запуска созданных юнитов
 - `sudo service nginx` – для перезагрузки конфигурации веб-сервера
 - `sudo -u service` – для выполнения некоторых операций с правами работающих процессов (см. далее)
- `service` – От его имени компоненты будут запущены. У него должны быть ограниченные привилегии для безопасности.

Также необходимо подготовить образ целевой операционной системы для сборки компонентов платформы при помощи системы контейнеризации (`Docker`). В этом образе должны быть также установлены вышеперечисленные пакеты, включая их `development` версии. Важно, чтобы образ содержал такие же или максимально близкие версии всех пакетов, как на целевых серверах, потому что на целевых серверах будут запущены исполняемые файлы, скомпилированные в образе. Они будут динамически загружать библиотеки, и если версии будут сильно отличаться, то какие-то вызовы функций могут не работать.

Также в образе должны быть установлены языки программирования, на которых написаны компоненты платформы: `Erlang`, `Elixir`, `Ruby`, `Java`, `Python`, `NodeJS`. Если целевая операционная система не позволяет установить последние версии некоторых языков, можно установить их при помощи `ASDF` (<https://asdf-vm.com>). Сначала необходимо установить `ASDF`, затем при помощи `ASDF` установить нужные версии языков. Актуальные версии языков можно найти в каждом компоненте платформы в файле `.tool-versions` в формате `ASDF`.

В поставку включен файл `Dockerfile`, из которого можно собрать базовый образ на основе `Red OS`. Файл можно отредактировать под требуемую целевую ОС и собрать нужный образ.

На компьютере оператора, на котором будет выполняться компиляция компонентов платформы, необходимо установить следующее ПО:

- `Bash`
- `dd`
- `OpenSSH`
- `Docker`
- `yq` (<https://github.com/mikefarah/yq>)

3. Установка Платформы

Необходимо создать базы данных в `Pangolin`:

- `a2p_admin`
- `antifraud`
- `billing`
- `clients_notifier`
- `gray_traffic`

- stop_spam
- holograph
- dmp_airflow
- dmp_harvest
- dmp_profiler
- dmp_segmenter
- dmp_targets_stat
- target_admin
- target_billing

Для каждой базы данных достаточно одного пользователя с правами на изменение структуры, запись и чтение. Компоненты платформы будут подключаться от имени этого пользователя. Рекомендуется также создать пользователей с правами только на чтение для технической поддержки.

Таблицы в базах данных создаются автоматически во время первого развертывания каждого компонента платформы.

Далее необходимо создать образ для компиляции компонент. Можно использовать вложенный в дистрибутив Dockerfile как шаблон для создания своего Dockerfile для целевой ОС. Имя образа необходимо экспортировать в переменную окружения IMAGE. Например:

```
docker build -t a2p_build .
export IMAGE=a2p_build
```

Затем необходимо последовательно скомпилировать и выполнить деплой каждой компоненты:

Перейти в каталог компоненты. Например:

```
cd a2p/a2p-sms-proxy
```

Отредактировать файлы конфигурации в каталогах config и stages. Указать актуальные для вашей инфраструктуры адреса серверов, логины и пароли для подключения к базам данных, и другую информацию.

Выполнить команду для создания и деплоя релиза на целевые сервера:

```
/path/to/platform_a2p/bin/perform deploy production
```

Команда выполнит несколько этапов:

- Сформирует сценарий компиляции этого конкретного компонента.
- Выполнит сценарий в контейнере на основе собранного ранее образа из переменной IMAGE. Результатом будет являться архив с релизом.
- Загрузит архив на целевые сервера.
- На каждом сервере распакует архив, создаст недостающие каталоги для логов и временных файлов.
- Создаст юниты systemd и запустит их.

После выполнения этого сценария рекомендуется зайти на каждый сервер и проверить логи всех процессов на наличие ошибок конфигурации. Их можно найти в каталоге `/var/log/projects`.

Вышеописанный сценарий можно выполнить поэтапно. Программа `perform` имеет отдельные команды для каждого этапа сборки. Их все можно узнать, выполнив команду `help`. Например:

```
/path/to/platform_a2p/bin/perform help
```

Если компиляция завершается с ошибкой, существует возможность отладить ее прямо в контейнере. Для этого служит команда `debug-build`. Например:

```
/path/to/platform_a2p/bin/perform debug-build production
```

Вы окажетесь в контейнере для компиляции. Сценарий компиляции будет находиться в файле `build.sh`. Можно выполнять команды из него по одной, видеть лог выполнения, и понять что идет не так. Возможно, в образе не хватает каких-то пакетов. В этом случае их нужно добавить в `Dockerfile` и пересобрать образ. Затем перезапустить установку.

После развертывания всех компонент платформа будет готова к работе. Можно входить в интерфейс администратора и заводить учетные записи для клиентов.