
Руководство администратора системы «Алмаз»

Выпуск 1.16.0

ООО «Инлексис»

июн. 07, 2018

1	Системные требования	1
1.1	Аппаратное обеспечение	1
1.2	Программное обеспечение	1
2	Установка	3
2.1	Зависимости	3
2.2	Алмаз	3
3	Конфигурация	5
3.1	almaz-core	5
3.2	almaz-spark-master	6
3.3	almaz-spark-streaming-master	7
3.4	almaz-notifier	7
4	Уведомления	9
4.1	API	9
4.2	Almaz Notifier	10

1.1 Аппаратное обеспечение

1.1.1 Минимальная конфигурация сервера

- 4 vCPU (например, один 2-х ядерный процессор)
- 16 Gb RAM
- 100 Gb HDD

1.1.2 Оптимальная конфигурация

- 32 vCPU (например, два 8-ми ядерных процессора или один 16-ти ядерный)
- 64 Gb RAM
- 1000 Gb HDD

Можно запустить и на 1 сервере, но оптимально от 4

1.2 Программное обеспечение

- OS – Linux CentOS последняя стабильная версия
- Docker Community Edition – последняя стабильная версия

Рекомендуемый способ развертывания системы Алмаз – это развертывание с помощью Docker контейнеров. Именно такой способ будет описан в этом документе.

Перед установкой, нужно убедиться, что у вас есть доступ к docker образу с системой Алмаз.

Совет: Например, образ может быть предоставлен вам в виде архива, скажем, `almaz.tar.gz`. Его можно импортировать с помощью команды:

```
$ docker import almaz.tar.gz almaz
```

2.1 Зависимости

В качестве базы данных Алмаз использует [MongoDB](#) (проверенная версия 3.2.20). Для обработки данных нужно развернуть standalone cluster [Apache Spark](#) (совместимая версия 2.0.2). Если Алмаз будет использоваться для работы с потоковыми данными, нужно развернуть [Kafka](#) из [Confluent Platform](#) (проверенная версия 3.1.1).

2.2 Алмаз

После того как зависимости были установлены, можно приступать к настройке и запуску системы Алмаз. Алмаз состоит из четырех сервисов.

2.2.1 almaz-core

Основной сервис, у которого есть web-интерфейс и в котором происходит расчёт моделей.

Пример команды для запуска:

```
$ docker run -d --namealmaz-core -p 8081:8081 -p 18081:18081almazalmaz-core
```

2.2.2almaz-spark-master

Модуль для управления задачами анализа данных в Spark для батчевых источников.

Пример команды для запуска:

```
$ docker run -d --namealmaz-spark-master -p 8061:8061 -p 18061:18061 -p 4040:4040almazalmaz-  
↪spark-master
```

2.2.3almaz-spark-streaming-master

Модуль для управления задачами анализа данных в Spark для потоковых источников. Можно не разворачивать, если не требуется мониторинг потоковых данных.

Пример команды для запуска:

```
$ docker run -d --namealmaz-spark-streaming-master -p 8062:8062 -p 18062:18062 -p 4041:4041almaz↪  
↪almaz-spark-streaming-master
```

2.2.4almaz-notifier

Модуль для рассылки нотификаций. Необходимо развернуть на машине, где есть доступ в Интернет.

Пример команды для запуска:

```
$ docker run -d --namealmaz-notifieralmazalmaz-notifier
```


Конфигурация модулей системы происходит с помощью переменных окружения.

Совет: При запуске docker контейнера из командной строки переменные окружения можно указать следующим образом:

```
$ docker run -e SERVER_PORT=8080 -d --name almaz-core -p 8080:8080 -p 18081:18081 almaz almaz-core
```

3.1 almaz-core

- **JAVA_OPTS** – параметры JVM, в зависимости от планируемой нагрузки имеет смысл увеличить количество используемой памяти (по-умолчанию, «-Xms512m -Xmx512m»)
- **SERVER_PORT** – порт, на котором запущен веб-сервер (по-умолчанию, 8081)
- **SPRING_DATA_MONGODB_HOST** – имя сервера с MongoDB (по-умолчанию, localhost)
- **SPRING_PROFILES_ACTIVE** – нужно перечислить через запятую профили запуска: prod – продуктивная среда (включена авторизация пользователя), kafka – требуется мониторинг потоков (например, «prod,kafka», по-умолчанию, «default», подходит для тестовой среды без потоков)
- **ALMAZ_DIAG_ALMAZ_SPARK_MASTER_URL** – адрес веб-сервиса диагностики модуля almaz-spark-master (по-умолчанию, «<http://localhost:18061>»), этот адрес нужно указать, чтобы корректно работала страница со статусом системы
- **ALMAZ_DIAG_ALMAZ_SPARK_STREAMING_MASTER_URL** – адрес веб-сервиса диагностики модуля almaz-spark-streaming-master (по-умолчанию, «<http://localhost:18062>»), этот адрес нужно указать, чтобы корректно работала страница со статусом системы
- **ALMAZ_SPARKSTREAMINGMASTER_KAFKA_BOOTSTRAPSERVERS** – параметры подключения к Kafka, нужно указывать если включен профиль kafka, то есть требуется мониторинг потоковых данных (по-умолчанию, пустая строка)

- `ALMAZ_SPARKMASTER_HOST` – имя сервера, на котором развернут `almaz-spark-master` (по-умолчанию, `localhost`)
- `ALMAZ_SPARKSTREAMINGMASTER_HOST` – имя сервера, на котором развернут `almaz-spark-streaming-master` (по-умолчанию, `localhost`)
- `ALMAZ_SPARKSTREAMINGMASTER_KAFKA_TOPIC` – название топика в Kafka, через который `almaz-core` получает результаты от `almaz-spark-streaming-master`, нужно указать если развернут `almaz-spark-streaming-master` (по-умолчанию, «`almaz_results`»)
- `ALMAZ_ASYNC_STREAMRESULTS_MAXPOOLSIZE` – максимальное количество потоков, которые обрабатывают результаты вычислений из `almaz-spark-streaming-master`, по-умолчанию, 10, но можно увеличить или уменьшить в зависимости от нагрузки (количества потоковых мониторингов) и мощности сервера
- `ALMAZ_NOTIFICATIONS_EMAIL_ENABLED` – включить или выключить рассылку уведомлений через email API в `almaz-notifier` (см. *Email*). По-умолчанию, `false`.
- `ALMAZ_NOTIFICATIONS_EMAIL_SUBJECT` – тема письма для рассылки уведомлений через email API в `almaz-notifier` (см. *Email*). Нужно указать такой же, как указан в настройках `almaz-notifier`
- `ALMAZ_NOTIFICATIONS_EMAIL_TO` – адрес email, на который нужно отправлять уведомления в API `almaz-notifier` (см. *Email*).
- `ALMAZ_EMAIL_FROM` – адрес email, с которого нужно отправлять письма
- `ALMAZ_EMAIL_LOGIN` – логин для доступа к почтовому SMTP серверу
- `ALMAZ_EMAIL_PASSWORD` – пароль для доступа к почтовому SMTP серверу
- `ALMAZ_EMAIL_SMTP_HOST` – имя почтового SMTP сервера
- `ALMAZ_EMAIL_SMTP_PORT` – порт почтового SMTP сервера
- `ALMAZ_EMAIL_SSL` – использовать ли SSL для доступа к SMTP серверу (по-умолчанию, `false`)
- `ALMAZ_EMAIL_TLS` – использовать ли TLS для доступа к SMTP серверу (по-умолчанию, `false`)
- `ALMAZ_LICENCE_TO` – на кого выдана лицензия Алмаза (как правило, название организации)
- `ALMAZ_LICENCE_KEY` – лицензионный ключ (уникальный для каждой организации)

3.2 `almaz-spark-master`

- `JAVA_OPTS` – параметры JVM, в зависимости от планируемой нагрузки имеет смысл увеличить количество используемой памяти (по-умолчанию, «`-Xms512m -Xmx512m`»)
- `SERVER_PORT` – порт, на котором запущен веб-сервер (по-умолчанию, 8061)
- `ALMAZ_SPARK_MASTER` – параметры подключения к `spark-master` (по-умолчанию, `spark://127.0.0.1:7077`)
- `ALMAZ_SPARK_CORES_MAX` – максимальное количество ядер Spark кластера, которое может занять `almaz-spark-master`. Должно быть равно общему количеству ядер в кластере, если `almaz-spark-streaming-master` не развернут, либо половине всех ядер, если `almaz-spark-streaming-master` развернут
- `ALMAZ_SPARK_EXECUTOR_CORES` – количество ядер Spark кластера, которое может занять `almaz-spark-master` на одной машине. Должно быть равно количеству ядер на одной машине, если `almaz-spark-streaming-master` не развернут, либо половине ядер на одной машине, если `almaz-spark-streaming-master` развернут

- `ALMAZ_SPARK_EXECUTOR_MEMORY` – объем памяти, который может занять `almaz-spark-master` на одной машине. Должно быть равно количеству доступной памяти на одной машине, если `almaz-spark-streaming-master` не развернут, либо половине доступной памяти, если `almaz-spark-streaming-master` развернут. Всю память занимать не нужно, нужно оставить некоторое количество операционной системе
- `ALMAZ_RDD_PARTITIONS` – оптимально должен быть от `ALMAZ_SPARK_CORES_MAX * 2` до `ALMAZ_SPARK_CORES_MAX * 4` (по-умолчанию, 16)
- `ALMAZ_SPARK_DRIVER_MAXRESULTSIZE` – максимальный размер результата вычислений в `almaz-spark-master`. По-умолчанию, 1g (1 гигабайт), но иногда требуется увеличить, если есть мониторинги с большим количеством полей и группировок
- `ALMAZ_IGNITE_ADDRESSES` – параметры подключения к Ignite. Ignite кластер поднимается автоматически, на всех машинах где развернуты `spark-slave`. Нужно сформировать строчку вида `spark-slave-host1:47500,spark-slave-host2:47500` и т.д. По-умолчанию, `127.0.0.1:47500`

3.3 `almaz-spark-streaming-master`

- `JAVA_OPTS` – Параметры JVM, в зависимости от планируемой нагрузки имеет смысл увеличить количество используемой памяти (по-умолчанию, «-Xms512m -Xmx512m»)
- `SERVER_PORT` – Порт, на котором запущен веб-сервер (по-умолчанию, 8062)
- `ALMAZ_SPARK_MASTER` – параметры подключения к `spark-master` (по-умолчанию, `spark://127.0.0.1:7077`)
- `ALMAZ_KAFKA_BOOTSTRAPSERVERS` – параметры подключения к Kafka (по-умолчанию, `localhost:9092`)
- `ALMAZ_KAFKA_TOPIC` – название топика в Kafka, через который `almaz-core` получает результаты от `almaz-spark-streaming-master` (по-умолчанию, «.almaz_results»)
- `ALMAZ_SPARK_CORES_MAX` – максимальное количество ядер Spark кластера, которое может занять `almaz-spark-streaming-master`. Должно быть равно половине всех ядер кластера
- `ALMAZ_SPARK_EXECUTOR_CORES` – количество ядер Spark кластера, которое может занять `almaz-spark-streaming-master` на одной машине. Должно быть равно половине ядер на одной машине
- `ALMAZ_SPARK_EXECUTOR_MEMORY` – объем памяти, который может занять `almaz-spark-streaming-master` на одной машине. Должно быть равно половине доступной на машине памяти. Всю память занимать не нужно, нужно оставить некоторое количество операционной системе
- `ALMAZ_RDD_PARTITIONS` – оптимально должен быть от `ALMAZ_SPARK_CORES_MAX * 2` до `ALMAZ_SPARK_CORES_MAX * 4` (по-умолчанию, 16)

3.4 `almaz-notifier`

- `JAVA_OPTS` – Параметры JVM, в зависимости от планируемой нагрузки имеет смысл увеличить количество используемой памяти (по-умолчанию, «-Xms128m -Xmx128m»)

Остальное см. :[ref:notifications-almaz-notifier](#)

4.1 API

Для рассылки уведомлений об ошибках или предупреждениях предусмотрен API в almaz-core, который предусматривает два интерфейса взаимодействия:

1. HTTP POST запрос (webhook)
2. Отправка email

В результате проверки мониторинга формируется сообщение в формате JSON с информацией для уведомлений. Сообщение имеет следующую структуру:

```
{
  "timestamp": 0.000000000, // Время проверки (Unix time)
  "checkedDate": 0.001000000, // Период мониторинга (Unix time)
  "monitoringObjectName": "TEST_ОБЪЕКТ", // Имя мониторинга
  "monitoringObjectId": "TEST_ОБЪЕКТ_ID", // ID мониторинга в базе данных
  "problems": [ // Список ошибок и предупреждений в данном периоде
    {
      "checkOutputId": "CHECK_OUTPUT_ID", // уникальный идентификатор проблемы
      "description": "В показателе null для группировки null найдены пустые значения", //  
↳ Краткое описание проблемы
      "level": "ERROR" // Уровень проблемы (WARNING, ERROR)
    }
  ],
  "emails": { // Подписки на уведомления по email
    "user@example.com": ["ERROR", "WARNING"]
  },
  "sms": { // Подписки на уведомления по sms
    "123": ["ERROR"]
  }
}
```

Далее это сообщение может быть отправлено потребителям с помощью webhook или сообщением на email.

4.1.1 Webhook

Может быть настроено отправление сообщения через HTTP POST в произвольное количество веб-сервисов. Для этого нужно настроить `almaz-core` следующим образом. В переменную окружения `ALMAZ_NOTIFICATIONS_WEBHOOK_ENABLED` записываем значение `true`. В переменную окружения `ALMAZ_NOTIFICATIONS_WEBHOOK_URLS` записываем через запятую адреса веб-сервисов, в которые мы хотим отправить сообщение с информацией для уведомлений.

Таким образом можно настроить интеграцию с различными системами для рассылки уведомлений, управления инцидентами и т.д.

4.1.2 Email

Для того чтобы настроить отправку уведомлений через Email, нужно в переменную окружения `ALMAZ_NOTIFICATIONS_EMAIL_ENABLED` записать значение `true`, а также установить тему письма `ALMAZ_NOTIFICATIONS_EMAIL_SUBJECT` и email получателя `ALMAZ_NOTIFICATIONS_EMAIL_TO`.

4.2 Almaz Notifier

Из коробки доступен модуль `almaz-notifier`, который получает уведомления от `almaz-core` с помощью *API*. Для работы некоторых функций `almaz-notifier`'а нужен доступ в Интернет. `almaz-notifier` получает уведомления от `almaz-core` в виде JSON документов такого вида:

```
{
  "timestamp": 0.000000000,
  "checkedDate": 0.001000000,
  "monitoringObjectName": "TEST_OBJECT",
  "monitoringObjectId": "TEST_OBJECT_ID",
  "problems": [
    {
      "checkOutputId": "CHECK_OUTPUT_ID",
      "description": "В показателе null для группировки null найдены пустые значения",
      "level": "ERROR"
    }
  ],
  "emails": {
    "user@example.com": ["ERROR"]
  },
  "sms": {
    "123": ["ERROR"]
  }
}
```

4.2.1 Входные каналы

Для получения уведомлений предусмотрено два канала:

1. HTTP POST веб-сервис
2. Email

HTTP POST веб-сервис

После запуска `almaz-notifier` начинает слушать POST запросы по адресу: `http://example.com:8011/post` (порт можно поменять на другой с помощью переменной окружения `SERVER.PORT`). POST запрос должен содержать в своем теле корректный JSON.

Данный способ получения уведомлений следует использовать, если `almaz-core` и `almaz-notifier` находятся в одной локальной сети.

Email

Часто корпоративные правила безопасности запрещают использовать Интернет на серверах внутри защищенного контура. При этом служба для отправления email, как правило, доступна. Чтобы иметь возможность рассылать уведомления даже в таких условиях, в `almaz-notifier` предусмотрено получение уведомлений через email. В этом случае `almaz-notifier` разворачивается где-то за пределами защищенного контура и получает уведомления от `almaz-core` через email. Уведомление при этом находится в теле письма в виде JSON.

`almaz-notifier` забирает новые письма каждые `ALMAZ_INPUT_MAIL_CHECKINTERVAL` миллисекунд, после обработки письма удаляются с почтового сервера. Для того чтобы отсеять нежелательные письма, `almaz-notifier` обрабатывает только письма с темой `ALMAZ_INPUT_MAIL_SUBJECT`.

Для того, чтобы включить получение уведомлений через email на стороне `almaz-notifier` нужно добавить в список в переменной окружения `SPRING_PROFILES_ACTIVE` значение `input_mail`.

По умолчанию включен только входной канал HTTP POST и логирующий выходной канал (выводит в стандартный вывод содержание входящего сообщения). Чтобы включить другие входные и выходные каналы их нужно добавить в переменную окружения `SPRING_PROFILES_ACTIVE`, например, чтобы получать уведомления через email и отправлять их наружу через email и sms, переменная окружения может содержать следующие значения: `SPRING_PROFILES_ACTIVE=input_mail,output_mail,output_sms`

Настройки

Переменная окружения	Описание	Значение по умолчанию	Обязательность
<code>ALMAZ_INPUT_MAIL_CHECKINTERVAL</code>	Время проверки почтового ящика в миллисекундах	60000	Да
<code>ALMAZ_INPUT_MAIL_USERNAME</code>	Имя пользователя на почтовом сервере		Да
<code>ALMAZ_INPUT_MAIL_PASSWORD</code>	Пароль пользователя на почтовом сервере		Да
<code>ALMAZ_INPUT_MAIL_HOST</code>	Имя хоста почтового сервера		Да
<code>ALMAZ_INPUT_MAIL_PORT</code>	Порт почтового сервера		Да
<code>ALMAZ_INPUT_MAIL_SUBJECT</code>	Тема письма, содержащего уведомление		Да

4.2.2 Выходные каналы

Для отправления уведомлений предусмотрено три канала:

1. Email
2. Sms

3. Trello

Email

Рассылка производится согласно подпискам пользователей. Для включения нужно добавить `output_mail` в `SPRING_PROFILES_ACTIVE`.

Настройки

Переменная окружения	Описание	Значение по умолчанию	Обязательность
<code>ALMAZ_OUTPUT_MAIL_HOST</code>	Имя хоста почтового сервера		Да
<code>ALMAZ_OUTPUT_MAIL_PORT</code>	Порт почтового сервера		Да
<code>ALMAZ_OUTPUT_MAIL_SSL</code>	Использовать SSL	true	Да
<code>ALMAZ_OUTPUT_MAIL_AUTH</code>	Требуется ли аутентификация пользователя	false	Да
<code>ALMAZ_OUTPUT_MAIL_USERNAME</code>	Имя пользователя на почтовом сервере		Нет
<code>ALMAZ_OUTPUT_MAIL_PASSWORD</code>	Пароль пользователя на почтовом сервере		Нет
<code>ALMAZ_OUTPUT_MAIL_FROM_EMAIL</code>	Адрес отправителя		Да
<code>ALMAZ_OUTPUT_MAIL_FROM_NAME</code>	Имя отправителя	Almaz	Да
<code>ALMAZ_OUTPUT_MAIL_FROM_PROBLEMS_LIMIT</code>	Количество проблем, которые следует вывести в одном письме	20	Да

Sms

Рассылка производится согласно подпискам пользователей. Для включения нужно добавить `output_sms` в `SPRING_PROFILES_ACTIVE`. Используется сервис <http://websms.ru>

Настройки

Переменная окружения	Описание	Значение по умолчанию	Обязательность
<code>ALMAZ_OUTPUT_SMS_WEBSMS_LOGIN</code>	Логин пользователя в системе http://websms.ru		Да
<code>ALMAZ_OUTPUT_SMS_WEBSMS_PASSWORD</code>	Пароль пользователя в системе http://websms.ru		Да
<code>ALMAZ_OUTPUT_SMS_WEBSMS_PROTOCOL</code>	Протокол взаимодействия	https	Да
<code>ALMAZ_OUTPUT_SMS_WEBSMS_HOST</code>	Имя хоста с веб-сервисом		Да
<code>ALMAZ_OUTPUT_SMS_WEBSMS_METHOD</code>	Метод веб-сервиса (например, <code>http_in6.asp</code>)		Да
<code>ALMAZ_OUTPUT_SMS_WEBSMS_FROM</code>	Адрес отправителя		Да
<code>ALMAZ_OUTPUT_SMS_WEBSMS_TEST</code>	Проверить работу (запрос происходит, но sms не отправляется)	1	Да

Trello

Этот канал уведомлений подразумевает наличие технического аккаунта trello. Каждое полученное уведомление преобразуется в карточку (card) trello согласно его описанию и добавляется на доску (board) с названием мониторинга. В зависимости от уровня проблемы (ERROR, WARNING) на карточку добавляется метка с красным или желтым цветом соответственно.

Настройки

Переменная окружения	Описание	Значение по умолчанию	Обязательность
ALMAZ_OUTPUT_TRELLO_KEY	Ключ для доступа к API Trello		Да
ALMAZ_OUTPUT_TRELLO_TOKEN	Токен для доступа к API Trello		Да
ALMAZ_OUTPUT_TRELLO_ORGANIZATION_NAME	Имя организации, используемое в качестве id	almaz	Нет
ALMAZ_OUTPUT_TRELLO_ORGANIZATION_DISPLAY_NAME	Имя для дисплея организации, которое будет отображаться на дашборде	Almaz Monitoring	Нет
ALMAZ_OUTPUT_TRELLO_LIST_ERRORS	Название списка с ошибками и предупреждениями	Errors & Warnings	Нет
ALMAZ_OUTPUT_TRELLO_LIST_PROGRESS	Название списка для управления инцидентами (в процессе)	In Progress	Нет
ALMAZ_OUTPUT_TRELLO_LIST_RESOLVED	Название списка для управления инцидентами (решено)	Resolved	Нет
ALMAZ_OUTPUT_TRELLO_LIST_CLOSED	Название списка для управления инцидентами (закрыто)	Closed	Нет