

# **VisionLabs LUNA PLATFORM 5**

**Примечания к выпуску**

## Содержание

LUNA PLATFORM v.5.67.0	5
LUNA PLATFORM v.5.64.0	7
LUNA PLATFORM v.5.62.3	11
LUNA PLATFORM v.5.59.0	16
LUNA PLATFORM v.5.58.0	19
LUNA PLATFORM v.5.57.0	20
LUNA PLATFORM v.5.56.0	21
LUNA PLATFORM v.5.54.0	26
LUNA PLATFORM v.5.53.0	27
LUNA PLATFORM v.5.51.6	29
LUNA PLATFORM v.5.51.4	30
LUNA PLATFORM v.5.51.0	31
LUNA PLATFORM v.5.49.1	35
LUNA PLATFORM v.5.47.4	37
LUNA PLATFORM v.5.47.1	38
LUNA PLATFORM v.5.46.1	39
LUNA PLATFORM v.5.45.4	42
LUNA PLATFORM v.5.45.3	43
LUNA PLATFORM v.5.45.1	44
LUNA PLATFORM v.5.42.0	47
LUNA PLATFORM v.5.40.0	49
LUNA PLATFORM v.5.38.3	52
LUNA PLATFORM v.5.38.1	53

<b>LUNA PLATFORM v.5.36.5</b>	<b>55</b>
<b>LUNA PLATFORM v.5.35.0</b>	<b>58</b>
<b>LUNA PLATFORM v.5.34.0</b>	<b>59</b>
<b>LUNA PLATFORM v.5.33.0</b>	<b>63</b>
<b>LUNA PLATFORM v.5.32.0</b>	<b>67</b>
<b>LUNA PLATFORM v.5.31.0</b>	<b>69</b>
<b>LUNA PLATFORM v.5.30.0</b>	<b>70</b>
<b>LUNA PLATFORM v.5.28.0</b>	<b>75</b>
<b>LUNA PLATFORM v.5.27.0</b>	<b>75</b>
<b>LUNA PLATFORM v.5.26.0</b>	<b>77</b>
<b>LUNA PLATFORM v.5.25.0</b>	<b>79</b>
<b>LUNA PLATFORM v.5.24.2</b>	<b>81</b>
<b>LUNA PLATFORM v.5.24.1</b>	<b>81</b>
<b>LUNA PLATFORM v.5.24.0</b>	<b>82</b>
<b>LUNA PLATFORM v.5.23.1</b>	<b>83</b>
<b>LUNA PLATFORM v.5.23.0</b>	<b>84</b>
<b>LUNA PLATFORM v.5.22.0</b>	<b>86</b>
<b>LUNA PLATFORM v.5.21.0</b>	<b>90</b>
<b>LUNA PLATFORM v.5.20.0</b>	<b>91</b>
<b>LUNA PLATFORM v.5.19.0</b>	<b>92</b>
<b>LUNA PLATFORM v.5.18.0</b>	<b>93</b>
<b>LUNA PLATFORM v.5.17.0</b>	<b>95</b>
<b>LUNA PLATFORM v.5.16.0</b>	<b>98</b>
<b>LUNA PLATFORM v.5.15.0</b>	<b>101</b>

<b>LUNA PLATFORM v.5.14.0</b>	<b>102</b>
<b>LUNA PLATFORM v.5.13.0</b>	<b>104</b>
<b>LUNA PLATFORM v.5.12.2</b>	<b>104</b>
<b>LUNA PLATFORM v.5.12.1</b>	<b>105</b>
<b>LUNA PLATFORM v.5.12.0</b>	<b>106</b>
<b>LUNA PLATFORM v.5.11.0</b>	<b>107</b>
<b>LUNA PLATFORM v.5.10.0</b>	<b>109</b>
<b>LUNA PLATFORM v.5.9.0</b>	<b>110</b>
<b>LUNA PLATFORM v.5.8.0</b>	<b>113</b>
<b>LUNA PLATFORM v.5.7.0</b>	<b>116</b>
<b>LUNA PLATFORM v.5.6.0</b>	<b>118</b>
<b>LUNA PLATFORM v.5.5.0</b>	<b>119</b>
<b>LUNA PLATFORM v.5.4.0</b>	<b>121</b>
<b>LUNA PLATFORM v.5.3.0</b>	<b>122</b>
<b>LUNA PLATFORM v.5.2.1</b>	<b>124</b>
<b>LUNA PLATFORM v.5.2.0</b>	<b>124</b>
<b>LUNA PLATFORM v.5.1.3</b>	<b>127</b>
<b>LUNA PLATFORM v.5.1.2</b>	<b>128</b>
<b>LUNA PLATFORM v.5.1.1</b>	<b>129</b>

## LUNA PLATFORM v.5.67.0

### Изменения LP

- Добавлена возможность выполнения видеоаналитики на GPU.

GPU включается с помощью указания значения «gpu» в настройке «decoder\_device\_class» секции «LUNA\_REMOTE\_SDK\_VIDEO\_SETTINGS».

- SDK обновлен до версии 5.21.0.

В данной версии LUNA PLATFORM:

- обновлен HumanDetector до версии v6
- обновлен DeepFakeDetector до версии v3

**Важно!** Также обновился дефолтный порог для редетекции детектора FaceDetV3 с «0.3» до «0.5». При обновлении LUNA PLATFORM порог для редетекции в настройке «redetect\_score\_threshold» группы параметров «LUNA\_REMOTE\_SDK\_FACE\_DETECTOR\_SETTINGS» **не будет обновлен автоматически**. Необходимо вручную обновить порог до «0.5» для использования стандартной логики редетекции. Если ранее было установлено пользовательское значение, то необходимо самостоятельно обновить значение порога.

При установке LUNA PLATFORM с нуля значение порога «redetect\_score\_threshold» по умолчанию также будет равно «0.3». Необходимо использовать дамп-файл «platform\_settings.json» из комплекта поставки, либо обновить значение порога после установки. В следующем релизе при установке с нуля будет автоматически устанавливаться значение «0.5».

- Добавлен механизм указания доверенных детекций, позволяющий явно указать для каких детекций не нужно выполнять редетекцию.

Предполагается, что доверенные детекции были получены с помощью алгоритмов VisionLabs. Если отметить стороннюю детекцию как доверенную, то это может повлиять на результаты оценок.

Указать является ли детекция доверенной можно с помощью схемы «application/json» (параметр «trusted\_detections») или схемы «multipart/form-data» (заголовок «X-Luna-Trusted-Detections») в следующих запросах, где детекции явно указываются вместо изображений:

- «detect faces»
- «sdk»
- «iso»
- «generate events»
- «perform verification»
- «generate stream events (beta)»

- В запрос «generate stream events (beta)» с указанием исходного изображения (sources > source\_type > raw\_image) добавлено опциональное поле origin\_bounding\_box, поз-

воляющее указать координаты bbox с лицом или телом в системе координат исходного изображения, которые будут сохранены в генерируемое событие.

Таким образом в запросе можно указать как координаты bbox в системе координат Crop-изображения (параметр `bounding_box`), так и координаты bbox в системе координат исходного изображения (параметр `origin_bounding_box`).

Также параметр `sources > source > face/body > bounding_box`, доступный при указании детекций в запросе (`sources > source_type > detections`), переименован в `origin_bounding_box`.

- Добавлена поддержка задания области интереса DROI относительно исходного кадра.

DROI, в отличие от ROI, не является параметром оценки, а работает как фильтр после выполненной в соответствии с полем «targets» оценки. Иначе говоря, ROI предназначен для эстимации и оптимизации, а DROI для реализации бизнес-логики. Можно использовать DROI как вместе с ROI, так и отдельно. Например, если после выполнения оценки с учетом ROI количество людей на кадре было равно N, то после выполнения дополнительной фильтрации по области DROI количество людей на кадре может быть уменьшено до M.

Область интереса DROI на исходном кадре задается следующими параметрами:

- «area» — геометрия области интереса. Параметр представлен в виде массива полигонов. Каждый полигон представлен массивом объектов, где каждый объект содержит координаты вершины полигона (x и y). Так, например, можно задать треугольную область интереса.
  - «mode» – режим указания «x», «y». Доступно два режима:
    - \* «abs» — параметры «x», «y» задаются в пикселях;
    - \* «percent» — параметры «x», «y» задаются в процентах от текущего размера
  - «form» — формат области интереса. В текущей реализации имеет только одно возможное значение «common». В будущих реализациях помимо полигонов будут поддержаны другие форматы.
- Во все задачи добавлена возможность задания времени жизни (TTL) результатов задач.

TTL задается в параметре `result_storage_policy>ttl` в запросах на создание задачи или расписания.

Добавить TTL результатов задач к уже созданным и выполненным задачам невозможно.

Добавить TTL результатов задач к уже созданному расписанию можно с помощью запроса [«replace tasks schedule»](#). Для созданных или запущенных на момент запроса задач TTL результатов задач применен не будет.
  - Добавлена автоматическая отправка заголовка `Set-Cookie: LUNA_AUTH_TOKEN=""`; `Path=/`; `Max-Age=0` клиентскому приложению при получении им ошибки 401 от сервиса API из-за протухшего токена авторизации, размещенного в Cookie.

## Исправленные ошибки LP

- Исправлена миграция настроек Configurator с ревизией c51cdfac.

Данная миграция завершалась с ошибкой, если в настройке S3 сервиса Image Store не было поля bucket.

- Удалена ненужная авторизация для запросов «get version», «get health», «set login cookie» и «clear login cookie».

## LUNA PLATFORM v.5.64.0

### Изменения LP

- Добавлена возможность задать время жизни (TTL) для объектов в бакетах (как локальных, так и S3).

TTL для объектов рассчитывается относительно формата времени GMT.

TTL для объектов задается в днях следующими способами:

- во время создания бакета для всех объектов сразу (основная политика TTL бакета)
- после создания бакета для конкретных объектов с помощью запросов к соответствующим ресурсам

Количество дней выбирается из списка в соответствующих запросах (см. ниже).

Кроме количества дней параметр может принимать значение «-1», означающее, что нужно хранить объекты вечно.

### Настройка основной политики TTL бакета

Основную политику TTL бакета можно настроить следующими способами:

- с помощью аргумента `--bucket-ttl` для скрипта `lis_bucket_create.py`. Например, «python3 ./base\_scripts/lis\_bucket\_create.py -ii --bucket-ttl=2».
- с помощью запроса к сервису Image Store. Например, «curl -X POST http://127.0.0.1:5020/1/buckets?buckets=samples?ttl=2».

### Настройка TTL для конкретных объектов

TTL для конкретных объектов можно настроить с помощью параметра «ttl» в следующих местах:

- в политиках «storage\_policy» > «face\_sample\_policy», «body\_sample\_policy» и «image\_origin\_policy» обработчика
- в запросах «create object», «create images» и «save sample»

Если параметр «ttl» не задан, то будет применена основная политика бакета, в котором находится объект (см. выше).

### **Поддерживаемые облачные провайдеры**

Поддерживаются облачные провайдеры Amazon S3, облачное хранилище Яндекса и MinIO.

### **Добавление TTL к существующим объектам**

Добавить TTL для существующего конкретного объекта можно с помощью PUT-запросов к ресурсам `/objects`, `/images`, `/samples/{sample_type}` сервиса Image Store.

Добавить TTL к основной политике бакета (ко всем существующим объектам в бакете) можно с помощью PUT-запроса к ресурсу `/buckets` сервиса Image Store.

Для бакета, расположенного в S3, необходимо выполнить миграцию с помощью скрипта «`base_scripts/migrate_ttl_settings`» сервиса Image Store. Это связано с тем, что для TTL объектов в S3 применяется через [фильтры, связанные с тегами](#). Команда выполнения миграции бакетов в S3 приведена в руководстве по установке. См. подробную информацию о миграции бакетов S3 в разделе [«Миграция для добавления TTL к объектам в S3»](#) в руководстве администратора.

### **Окончание TTL**

Когда TTL объекта подходит к концу, то он помечается для удаления. Для локальных бакетов выполняется задача очистки 1 раз в день (в 01:00 утра). Для бакетов S3 используются внутренние правила настройки TTL.

Обратите внимание, что между датой истечения срока действия и датой фактического удаления объекта может возникнуть задержка. Как для локального хранилища, так и для S3.

### **Поиск истекающих объектов**

Чтобы узнать, когда срок действия объекта истекает, можно использовать запросы с методами HEAD к ресурсам `/objects` и `/images`. Эти запросы возвращают заголовки ответов `X-Luna-Expiry-Date`, в которых указана дата, когда объект больше не подлежит сохранению.

См. подробную информацию в разделе [«TTL для объектов»](#) в руководстве администратора.

- Теперь сервис Handlers будет отправлять только одно уведомление, если у пользователя есть два callback-а с одинаковыми настройками.

Если пользователь создал два callback-а, которые отличаются только фильтрами, система отправит уведомление только один раз, даже если событие удовлетворяет обоим фильтрам.

Например, если настроено два callback-а в одну стороннюю систему с различием только в том, что у одного callback'а установлены фильтры по прохождению Liveness, а у другого по достижению определенного уровня «similarity», то при генерации события, удовлетворяющего обоим фильтрам, во внешнюю систему придет только одно уведомление.



Это позволяет легче реализовывать логику отправки уведомлений при соблюдении определенных фильтров.

- Добавлена возможность отправки уведомлений о состоянии задач и подзадач в Telegram.

Отправка уведомлений в Telegram реализована за счет добавления нового типа callback'ов telegram в политику «notification\_policy» каждой задачи.

Для отправки уведомлений необходимо указать параметры chat\_id и token.

Отправку уведомлений в Telegram также можно настроить при создании расписания для задач.

- Во контейнерах всех сервисов обновлена версия Python до 3.12.

Поддержка более старых версий Python прекращена.

- В параметры запроса «[video analytics \(beta\)](#)» добавлен параметр «video» > «orientation» > «angle», отвечающий за возможность поворота видео на угол 90, 180 или 270 градусов перед его обработкой.
- Добавлена возможность использовать более безопасный алгоритм для генерации JWT токенов.

По умолчанию сервис Accounts использует алгоритм HS256 для подписи JWT-токенов. При необходимости можно использовать асимметричное криптографическое шифрование с помощью алгоритм ES256.

Для этого необходимо:

- сгенерировать закрытый ключ ECDSA
- закодировать закрытый ключ в формат Base64
- установить переменные окружения «ACCOUNTS\_JWT\_ECDSA\_KEY» и «ACCOUNTS\_JWT\_ECDSA\_KEY\_PASSWORD»

**Важно!** Переключение алгоритма подписи с HS256 на ES256 (или наоборот) оказывает значительное влияние на проверку токенов. **Все существующие токены, подписанные предыдущим алгоритмом, станут недействительными** после внесения изменений. Это происходит потому, что механизм проверки подписи токена ожидает, что структура и криптографическая база токена будут соответствовать новоуказанному алгоритму.

См. раздел «[Алгоритмы JWT-токенов](#)» в руководстве администратора.

- Добавлена возможность шифрования паролей и токенов, передаваемых в задаче Estimator и в политике «notification\_policy»

Для этого необходимо при старте контейнера сервиса Tasks передать пользовательские значения переменным окружения FERNET\_PASSPHRASE и SALT.

- FERNET\_PASSPHRASE — пароль или ключ, используемый для шифрования данных с помощью алгоритма Fernet.
- SALT — случайная строка, добавляемая к паролю перед его хешированием.

**Важно!** Когда контейнер запускается с указанием вышеописанных переменных окружений, то старые пароли и токены перестают работать. Необходимо в команде миграции БД Tasks указать переменные окружения, после чего необходимо запустить новый контейнер Tasks с указанием переменных окружения для возможности использования шифрования при создании новых объектов.

См. подробную информацию в разделе [«Дополнительная защита паролей и токенов»](#) в руководстве администратора.

- Добавлена группа параметров «FACE\_QUALITY\_DEFAULT\_THRESHOLDS» сервиса Remote SDK, содержащая стандартные пороги для группы проверок «face\_quality», политики «detect\_policy», ресурсов [«/handlers»](#) и [«/verifiers»](#).

В данной группе параметров есть два типа стандартных порогов:

- пороги из поля «threshold», определяющие при каком значении LUNA PLATFORM будет считать, что проверка пройдена. Данные пороги аналогичны порогам в группе проверок «face\_quality» в соответствующих запросах. Задание порогов в запросах переопределяет стандартные пороги, задаваемые в группе параметров «FACE\_QUALITY\_DEFAULT\_THRESHOLDS».
- пороги из поля «config», определяющие как интерпретировать ответ эстиматоров Natural Light, Fish Eye, Color Type и Red Eyes. Ранее данные пороги были недоступны.

Пример настройки порогов для эстиматора Red Eyes:

```
"red_eye": {
  "threshold": 0,
  "config": {
    "estimator": {
      "threshold": {
        "min": 0.5
      }
    }
  }
}
```

Здесь:

- поле «threshold» определяет значение «0» или «1», при котором LUNA PLATFORM будет считать, что проверка пройдена. Аналогично параметру «threshold» в запросах на выполнение «face\_quality»
- секция «config» > «estimator» > «threshold» определяет ответ эстиматора Red Eyes от «0» до «1».

- Добавлена проверка доступности пользовательского Docker registry и базовых образов Lambda во время запуска сервиса.
- Теперь не требуются никакие дополнительные права для выполнения запросов к ресурсу `/lambdas/{lambda_id}/proxy` сервиса Lambda.

### Исправленные ошибки LP

- Исправлена некорректная работа Cached Matcher с 64 версией нейронной сети для извлечения биометрических шаблонов лиц.

Cached Matcher не кэшировал биометрические шаблоны 64 версии и выполнял сравнение используя данные из БД.

- Исправлен некорректный формат поля «url» для лиц и атрибутов, генерируемый запросом `«generate events»`.

Например, адрес расположения лица мог возвращаться как `'url': 'http://127.0.0.1:5030/faces/359c8c12-b67d-4645-96dc-d8f1627f360f'`.

Правильный адрес: `'url': 'http://127.0.0.1:5030/3/faces/426542d6-5509-4e5b-8a01-e2abd5c0a8c6'`

- Запрос `«stream events (beta)»` теперь генерирует значение поля `image_origin` на основе детекции, если в теле запроса не указан параметр `image_origin`, но указан биометрический образец лица или тела.

## LUNA PLATFORM v.5.62.3

### Изменения LP

- SDK обновлен до версии 5.19.1.

В данной версии LUNA PLATFORM:

- поддержана 64ая модель нейронной сети для извлечения биометрических шаблонов лиц;
- дефолтная модель была изменена с 59ой на 62ую (настройка «DEFAULT\_FACE\_DESCRIPTOR\_VERSION» сервиса Remote SDK).

При выполнении миграции настроек сервиса Configurator последней версии, настройка «DEFAULT\_FACE\_DESCRIPTOR\_VERSION» **не будет автоматически обновлена**. При необходимости использования 62ой версии нейронной сети необходимо вручную изменить значение настройки «DEFAULT\_FACE\_DESCRIPTOR\_VERSION», предварительно выполнив повторное извлечение существующих биометрических шаблонов с помощью 62 модели, используя задачу `»Additional extraction`.

- В комплект поставки добавлены Helm чарты для разворачивания LUNA PLATFORM в кластере Kubernetes.

Администратор должен иметь развернутый и настроенный кластер Kubernetes для использования Helm чартов. Предполагается, что в пользовательском кластере Kubernetes:

- запущены СУБД PostgreSQL/Oracle и базы данных InfluxDB и Redis
- имеется доступ к объектному хранилищу S3-подобного типа для хранения бакетов

См. пример команд для разворачивания LUNA PLATFORM в кластере Kubernetes с использованием Helm чартов в новом руководстве (документ «LP\_Kubernetes\_Helm\_Deployment\_Example» в комплекте поставки) или на сайте [с онлайн-документацией](#).

- Для LUNA PLATFORM добавлен пользовательский интерфейс, реализованный в качестве веб-интерфейса для сервиса API.

Пользовательский интерфейс доступен по адресу `<host:5000/ui>`.

См. описание пользовательского интерфейса в новом руководстве (документ «LP\_User\_Interface\_Manual» в комплекте поставки) или на сайте [с онлайн-документацией](#).

- В запрос «[videosdk](#)» добавлена возможность отслеживания людей на видео и оценки их атрибутов как контексте всего видео, так и на отдельных кадрах.

Для видеоаналитики отслеживания людей ключевым понятием является трек. Трек — объект, содержащий информацию о положении одного человека на последовательности кадров. Трек создается тогда, когда на последнем кадре из указанного количества подряд идущих кадров (параметр «probe\_count») появляется детекция лица, тела или тела с лицом (зависит от типа детектора в параметре «detector\_type»). Треков может быть столько, сколько людей было обнаружено на кадре. Вместе с треком начинается событие видеоаналитики.

**Примечание.** В настоящий момент для одного трека может быть только одно событие. В будущих релизах для трека можно будет получить несколько событий, которые можно будет отправлять не дожидаясь окончания трека.

Например, если «probe\_count» = «3» и на двух подряд идущих кадрах есть детекции, а на третьем нет, то трек и событие не начнутся. Если в какой-то момент отслеживания человека на двух подряд идущих кадрах были детекции, а на третьем кадре детекция отсутствует, то трек и событие закончатся.

Во время отслеживания человека доступна возможность выполнить оценку атрибутов лиц, тел, изображения (перечень оценок задается в параметре «targets»), а также получить изображение, в котором была лучшая детекция. Оценка может быть выполнена по одному лучшему снимку из трека или по нескольким (параметр «face\_samples»/«body\_samples» > «count»). При необходимости можно фильтровать лучшие снимки по параметрам «score», «head\_pitch», «head\_roll», «head\_yaw». Все оценки, задаваемые в поле «targets» зависят от типа используемого детектора.

При необходимости можно включить повторную идентификацию тела (ReID). Функция ReID используется для повышения точности отслеживания путем объединения различных треков одного человека.

Для такой видеоаналитики также доступна возможность настроить прямоугольную область интереса ROI (параметр «roi») на кадре (координаты «x», «y», ширина, высота, единицы измерения — проценты или координаты). Принцип работы области интереса ROI аналогичен FaceStream.

В каждом событии видеоаналитики подсчета количества людей содержится следующая информация:

- «event\_id» — идентификатор события
- «track\_id» — идентификатор трека
- «video\_segment» — информация о сегменте видео (время начала события, время конца события, номер первого кадра, номер последнего кадра)
- «frames\_estimations» — массив с информацией об оценках на каждом кадре, содержащий следующую информацию:
  - \* номер кадра
  - \* время, соответствующее кадру
  - \* координаты bbox и степень достоверности («score»)
- «aggregated\_estimations» — результат оценки, задаваемой в параметре «targets» в теле запроса
- «track\_count» — количество треков

См. подробную информацию в [спецификации OpenAPI](#).

- В видеоаналитику подсчета количества людей добавлена возможность получения координат людей и изображений с максимальным количеством людей.

Для этого в тело запроса был добавлен новый параметр «targets», принимающий следующие значения:

- «coordinates» — вычисление координат людей
- «overview» — получение изображения с максимальным количеством людей

Можно задать как оба значения, так и ни одного. Если значения не заданы, то будет выполнен базовый анализ, содержащий только информацию о видеозаписи, количестве людей и кадрах, связанных с ними.

Координаты людей возвращаются в теле ответа в поле «events» > «aggregated\_estimations» > «overview» > «people\_coordinates».

- Переработана структура запроса [«generate stream events \(beta\)»](#).

Поля «sources» > «source» > «face\_bounding\_boxes» и «body\_bounding\_boxes» (доступны при

«source\_type» = «raw\_image») переименованы в «face\_detection\_data» и «body\_detection\_data» соответственно и переработаны следующим образом:

- Параметры «height», «width», «x», «y» перемещены в поля «face\_detection\_data»/«body\_detection\_data» > «bounding\_box».
- Поля «face\_detection\_data»/«body\_detection\_data» теперь содержат новые опциональные поля «score», «angles» (содержит параметры «pitch», «yaw» и «roll») и обязательное поле «bounding\_box» (см. выше).
- В поля «face\_detection\_data»/«body\_detection\_data» добавлена возможность указания пользовательских ключей со значениями.

Поле «sources» > «source» > «face» (доступно при «source\_type» = «detections») теперь содержит новые опциональные поля «score» и «angles». Также в поле «face» добавлена возможность указания пользовательских ключей со значениями.

Также в поле «sources» > «source» добавлена возможность указания пользовательских ключей со значениями.

- Расширено описание поля «cron» в запросах [«create tasks schedule»](#), [«replace tasks schedule»](#), [«get tasks schedules»](#) и [«get tasks schedule»](#).

- Статистика запросов в формате Prometheus дополнена временем выполнения запроса.

Данные представлены в формате гистограммы, которая позволяет анализировать распределение времени выполнения запросов и выявлять потенциальные проблемы производительности. Каждый интервал времени (бакет) в гистограмме отображает количество запросов, время выполнения которых попадает в соответствующий диапазон.

См. подробную информацию в разделе [«Экспорт метрик в формате Prometheus»](#).

- Добавлена возможность установки меток для запуска lambda с использованием конкретных узлов Kubernetes с помощью параметра «deploy\_parameters» > «selector».

Это позволяет более детально контролировать развертывание lambda, позволяя администраторам указывать узлы, на которых должны быть размещены lambda в зависимости от их потребностей в ресурсах. Применяя метки к узлам Kubernetes и используя параметр «selector», администраторы могут управлять выделением ресурсов не только для отдельных lambda, но и для групп lambda с аналогичными требованиями к ресурсам.

См. запросы [«create lambda»](#), [«put lambda»](#), [«get lambda»](#), [«get lambdas»](#).

- Уведомление, отправляемое во внешнюю систему с помощью механизма callback'ов, расширено параметром «result\_url», содержащим внешний адрес к сервису API с результатом задачи или подзадачи.
- Версия SDK для Handlers-lambda обновлена с версии 5.17.0 до версии 5.19.0.
- Добавлена возможность указания количества подов для lambda.

Количество подов задается в параметре «pod\_count» секции «deploy\_parameters» запроса [«create lambda»](#). Можно задать от 1 до 32 подов.

Для каждого пода lambda также можно установить ограничения на процессор и память с помощью новых параметров «cpu\_limit», «ram\_limit», «cpu\_request», «ram\_request» в секции «deploy\_parameters» > «resources».

Кроме того, в запросах [«get lambda status»](#) и [«get lambda status»](#) обновился выходной формат. Теперь логи возвращаются как словарь, что позволяет отобразить информацию о нескольких подах lambda.

Задать количество подов и ресурсы для существующей lambda можно с помощью запроса [«put lambda»](#).

- В сервис Lambda добавлена поддержка использования пространств имен (namespace) Kubernetes, обеспечивающих изоляцию групп ресурсов внутри единого кластера.

Namespace задается в параметре «namespace» секции «deploy\_parameters» запроса [«create lambda»](#). См. рекомендации к названиям namespace в описании параметра соответствующего запроса.

Задать namespace для существующей lambda можно с помощью запроса [«put lambda»](#).

- Неиспользуемый параметр «landmarks17\_threshold» удален из группы параметров «LUNA\_REMOTE\_SDK\_BODY\_DETECTOR\_SETTINGS».
- Новый ресурс /metrics добавлен в создаваемые lambda, позволяющий собирать и сохранять метрики в формате Prometheus в виде данных временных рядов, которые можно использовать для отслеживания поведения lambda.

См. подробную информацию в разделе [«Экспорт метрик в формате Prometheus»](#).

- Переработана структура документации в комплекте поставки.

Системные требования LUNA PLATFORM вынесены в отдельный документ в комплекте поставки.

Руководства по установке перенесены в директорию «InstallationManuals».

### Исправленные ошибки LP

- Исправлена ошибка, из-за которой некоторые параметры не отображались, когда фильтр «show\_all» был включен в поле «Service name» на странице «Settings» пользовательского интерфейса сервиса Configurator.

Кроме того, фильтр «show\_all» теперь установлен как значение по умолчанию.

- Исправлена ошибка, при которой в некоторых случаях обработка расписания задач не перезапускалась должным образом после временной недоступности среды (например, базы данных).

- Исправлена ошибка в сервисе Backport 3, из-за которой не происходила очистка списка закрытых веб-сокет соединений в случае, когда веб-сокеты были недоступны или не инициализированы.

## LUNA PLATFORM v.5.59.0

### Изменения LP

- **Примечание.** В следующем релизе будет обновлено дефолтное значение нейронной сети для извлечения биометрических шаблонов лиц с 59 версии на 62 версию. Версия 59 будет удалена из контейнера Remote SDK.
- Разработана утилита Storages, позволяющая проверить и/или подготовить окружение для сервисов LUNA PLATFORM v.5.46.1 и выше перед их непосредственным запуском.

В качестве подготовки окружения понимается следующее:

- Подготовка бакетов в InfluxDB для работы мониторинга
- Подготовка бакетов для сервиса Image Store, позволяющих хранить пользовательские данные (изображение, метаданные, архивы и пр.)
- Подготовка БД Influx для сбора агрегированной статистики сервисом Admin (см. раздел «Подсчет статистики выполненных запросов и оценок» в руководстве администратора)
- Подготовка баз данных, добавление функций VLMatch, создание сценариев миграции для баз данных LUNA PLATFORM (PostgreSQL или Oracle) и управление ими
- Выполнение миграции/загрузка настроек в БД Configurator
- Загрузка дамп-файлов в БД сервиса Configurator

Основное преимущество Storages перед ручной подготовкой окружения в том, что утилите известны ревизии настроек сервиса Configurator и сценарии миграции баз данных сервисов LP, что значительно упрощает процесс обновления и даунгрейда LUNA PLATFORM. Кроме того, с помощью утилиты можно оценить текущее состояние окружения LUNA PLATFORM и определить что именно будет нужно обновить.

Утилита поставляется в Docker-контейнере и может быть использована в качестве инструмента подготовки окружения LUNA PLATFORM, разворачиваемой в Docker-контейнерах, скриптом Docker Compose, системой оркестрации Kubernetes и др. Основная задача администратора - указать утилите Storages адреса баз данных, бакетов и пр.

В комплект поставки и на сайт с онлайн-документацией добавлен новый документ [«Руководство по утилите Storages»](#), описывающий принципы использования утилиты и примеры использования. Также добавлено два новых руководства по разворачиванию LUNA PLATFORM -



[«Руководство по установке с помощью утилиты Storages»](#) и [«Руководство по обновлению с помощью утилиты Storages»](#). Документы аналогичны обычным руководствам по установке и обновлению, за исключением того, что вся подготовка окружения выполняется с помощью утилиты Storages.

**Важно!** Утилита Storages находится в состоянии бета-тестирования. См. подробную информацию об утилите в [руководстве по утилите Storages](#).

- SDK обновлен до версии 5.18.0.
- В политику «notification\_policy» каждой задачи добавлена возможность включить отправку уведомлений о состоянии подзадач (группа параметров «subtask\_callbacks»).

Состояние подзадачи содержит следующую информацию:

- идентификатор подзадачи
- статус подзадачи («in\_progress», «failed», «cancelled», «done»)
- количество выполненных подзадач

В ответах на запросы [«get task notification policy»](#) и [«replace task notification policy»](#) также содержится вышеописанная информация.

Уведомления о состоянии подзадач также доступны при использовании политики «notification\_policy» в расписании.

- Во все сервисы LUNA PLATFORM добавлен новый ресурс `/metrics`, позволяющий собирать и сохранять метрики в формате Prometheus в виде данных временных рядов, которые можно использовать для отслеживания поведения сервиса.

По умолчанию сбор метрик отключен. Сбор метрик включается в параметре «enabled» в группе «LUNA\_SERVICE\_METRICS». Если сбор метрик отключен, то запрос к ресурсу `/metrics` вернет ошибку с кодом «12049» и сообщением «Forbidden, Resource is disabled».

Доступно два типа метрик:

- `request_count_total` — общее количество запросов
- `errors_count_total` — общее количество ошибок

Каждый из них имеет как минимум два лейбла для сортировки:

- `status_code` (или `error_code` для метрик ошибок)
- `path` — путь, состоящий из метода запроса и маршрута конечной точки

При необходимости можно добавить пользовательские типы меток, указав пару `label_name=label_value` в параметре «extra\_labels» в группе «LUNA\_SERVICE\_METRICS».

См. подробную информацию в разделе [«Экспорт метрик в формате Prometheus»](#).

- Добавлена возможность получения X и Y координат людей при выполнении оценки количества людей.

Возможность получения координат добавлена в запросы [«sdk»](#) и политику `«detect_policy»` запроса `«create handler»`. Для этого необходимо включить параметры `«people_count_coordinates»` и `«people_coordinates»` соответственно.

Структура генерируемого события также обновлена.

- В спецификацию OpenAPI добавлены примеры для запроса [«generate stream events \(beta\)»](#).
- Добавлена возможность запускать lambda GPU.

Для использования GPU необходимо включить параметр `«deploy_parameters» > «enable_gpu»` в запросе [создания lambda](#).

Есть некоторые требования и ограничения при использовании GPU в Kubernetes. См. подробную информацию в разделе [«Создание lambda с поддержкой GPU»](#) руководства администратора.

### Исправленные ошибки LP

- Исправлено отсутствие поддержки метода `OPTIONS` для ресурса `/plugins`.
- Исправлено появление ошибки `«Internal server error»`, возникавшей при выполнении задачи Garbage collection с включенным фильтром `events` и параметром `remove_image_origins`, когда в событии отсутствовало исходное изображение (значение `None`).
- Исправлена ошибка `value too long for type character varying(512)`, возникавшая при выполнении даунгрейда настроек Configurator с ревизии `d718e5954caf`.

Изменения, внесенные в ревизии `d718e5954caf`, привели к увеличению длины строки значения настроек токенов с 512 символов до бесконечности. При выполнении даунгрейда с ревизий, где в соответствующей настройке значение длины уже было больше 512 символов, возвращалась ошибка `value too long for type character varying(512)`.

Теперь при выполнении даунгрейда с ревизии `d718e5954caf`, значение настройки, превышающее 512 символов, будет удалено. Необходимо заново задать значение токена для соответствующей настройки.

- Исправлена ошибка `Key (name)=(<setting>)is not present in table "limitation"`, возникавшая при выполнении даунгрейда настроек Configurator с ревизии `dd2641137b42`.

Изменения, внесенные в ревизии `dd2641137b42`, привели к обязательной привязке каждой настройки к определенному шаблону. Для этого была добавлена новая таблица `limitation`, содержащая ограничения. Ошибка возникала при выполнении даунгрейда из ревизий, где в базе данных Configurator уже присутствовали данные об ограничениях в таблице `setting` и невозможно было установить связь между удаляемой таблицей `limitation` и существующей таблицей `setting`.

Теперь при выполнении даунгрейда с ревизии d718e5954caf, все записи из таблицы `setting` будут удалены при удалении таблицы `limitation`.

## LUNA PLATFORM v.5.58.0

### Изменения LP

- **Примечание.** Через один релиз будет обновлено дефолтное значение нейронной сети для извлечения биометрических шаблонов лиц с 59 версии на 62 версию. Также будет прекращена поддержка 52 версии нейронной сети.
- Добавлена поддержка использования токена (Bearer-авторизации) через Cookies для упрощенного процесса аутентификации в веб-браузерах.

Для сохранения Cookies нужно выполнить запрос [«set login cookie»](#), указав токен в качестве авторизации. Затем Cookies отправляется обратно в браузер пользователя. Последующие запросы, отправленные браузером пользователя, автоматически включают эти Cookies, позволяя серверу распознавать авторизацию пользователя, без необходимости явно отправлять токен в каждом запросе.

При необходимости можно очистить Cookies с помощью запроса [«clear login cookie»](#).

- Во все сервисы добавлена новая переменная окружения `«LUNA_SKIP_CHECK_CONNECTION»`.

Переменная позволяет отключить проверку соединения (значение «1») до всех основных сервисов (Image Store, Faces и т.д.), выполняющуюся по умолчанию при старте сервиса.

Проверка соединения выполняется для исключения ошибок, связанных с неправильной конфигурацией LUNA PLATFORM, но при этом может замедлить процесс поднятия контейнера. Кроме того, в некоторых случаях проверка соединения может вызывать проблемы при поднятии контейнера, особенно при нестабильном соединении.

Переменную окружения можно передать с помощью аргумента `--env` при старте контейнера.

- Добавлена поддержка запуска сервисов с использованием протокола HTTPS.

Для использования этой возможности необходимо передать следующие аргументы командной строки соответствующего сервиса:

- `tls_cert` — путь к SSL-сертификату
- `tls_key` — путь к SSL-закрытому ключу
- `tls_key_pass` — пароль для SSL-закрытого ключа (необязательно)

Пример команды: `python3 /srv/luna_<service>/run.py --tls_cert /srv/my_certificate.crt --tls_key /srv/my_private_key.key --tls_key_pass my_password`

Обратите внимание, что сертификат и ключ должны быть примонтированы к Docker-контейнеру в указанные директории.

Список всех доступных аргументов можно получить с помощью следующей команды:

```
python3 /srv/luna_<service>/run.py -h
```

- Во все сервисы LUNA PLATFORM добавлена возможность указания переменной окружения `--EXTEND_CMD`, позволяющей передать аргументы, для которых не предусмотрена переменная окружения, в команду запуска сервиса.

Например, можно явно задать тегированные настройки при запуске сервисов: `--env=EXTEND_CMD="--INFLUX_MONITORING=TAG_1 --LUNA_EVENTS_DB=TAG_2"`

См. подробную информацию о переменных окружения и аргументах в разделе «Аргументы сервисов» в руководствах по установке.

- В тела запросов `«create lambda»` и `«put lambda»` добавлен новый параметр `«workers»`, позволяющий выделить количество «рабочих процессов» на указанный экземпляр Lambda.
- Версия SDK для Handlers-lambda обновлена с версии 5.16.0 до версии 5.17.0.

### Исправленные ошибки LP

- В описаниях запросов `«create handler»`, `«validate handler policies»` и `«generate events»` исправлено отсутствие информации о допустимой длине символов в параметрах политики `«callbacks»`.
- Исправлена ошибка, приводившая к тому, что новое значение настройки, передаваемой в переменной окружения, не обновлялось, если эта настройка отсутствовала в сервисе Configurator или конфигурационном файле.
- Исправлено появление ошибки `«Internal server error»`, возникавшей при попытке обращения к ресурсам `/1/buckets/{bucket}/objects/{object_id}` и `/1/buckets/{bucket}/objects` сервиса Image Store не поддерживаемыми методами.

Теперь возвращаются корректные ошибки.

- Исправлена ошибка, из-за которой при создании задачи Estimator с несуществующим обработчиком, в базе данных Handlers все равно создавалась соответствующая запись.

## LUNA PLATFORM v.5.57.0

### Изменения LP

- Во все задачи добавлена политика `«notification_policy»`, которая позволяет отправлять уведомления об изменении статуса задач с помощью механизма callback'ов.

Callback'и позволяют отправлять данные в стороннюю систему по указанному URL.

Параметры из новой политики аналогичны параметрам, задаваемым в соответствующей политике обработчика («authorization», «content\_type», «timeout» и пр.).

Также можно настроить отправку уведомлений для задач в настройках расписания.

При необходимости можно получить информацию о текущем состоянии расписания или обновить его с помощью ресурсов [get task notification policy](#) и [replace task notification policy](#).

- В запросе [«create token»](#) добавлена возможность указать разрешение «account» > «view» для возможности получить данные аккаунта, к которому выдан сам токен.
- Для Tasks-lambda добавлен клиент «luna-handlers», позволяющий отправлять пользовательские события в сервис Handlers.

См. подробную информацию про клиенты в разделе [«Luna services clients»](#) руководства разработчика.

- В сервис Lambda добавлена возможность создания пользовательских точек мониторинга для экземпляров Lambda-tasks.

См. подробную информацию про мониторинг в разделе [«Lambda monitoring»](#) руководства разработчика.

- Добавлена офлайн версия сайта с онлайн-документацией в комплект поставки LUNA PLATFORM вместо предыдущих документов в формате HTML.

### Исправленные ошибки LP

- Теперь при выполнении запросов [«generate events»](#) и [«generate stream events \(beta\)»](#) с биометрическими образцами, в сервис Remote SDK не будут отправляться ограничивающие прямоугольники, т.к. они не используются.
- Из группы параметров LUNA\_REMOTE\_SDK\_LIMITS удалены неиспользуемые параметры raw\_event\_detections\_limit, raw\_event\_arrays\_limit и result\_candidate\_limit. Остался только параметр received\_images\_limit.
- Исправлена ошибка, при которой сервис Python Matcher Proxy использовал 100% CPU при включенном плагине «indexed\_matcher» (при использовании LIM).
- Исправлена ошибка, из-за которой при незаполненных query-параметрах геопозиции в запросах [«generate events»](#), [«save event»](#), [«generate stream events \(beta\)»](#), в базу данных Events все равно записывались значения «null».

Теперь если ни один параметр геопозиции не задан, то в таблице «location» не будет создано никакой записи.

## LUNA PLATFORM v.5.56.0

### Изменения

- Добавлена возможность выполнения видеоаналитики с помощью нового ресурса [«videosdk»](#).

**Важно!** На данный момент ресурс находится в статусе бета-тестирования. Входные и выходные схемы могут быть изменены в будущих релизах без поддержки обратной совместимости.

Видеоаналитика — это набор функций, которые обрабатывают кадр за кадром и оценивают полезные данные.

Для выполнения видеоаналитики необходимо указать внешнюю ссылку на видеофайл, не превышающий размер 1 Гб.

В процессе выполнения видеоаналитики генерируется определенное количество событий по некоторым правилам, где каждое событие имеет начало и конец. События записываются в тело ответа и содержат специфическую информацию конкретной видеоаналитики. В теле ответа также содержится базовая метаданная о видео (количество кадров, частота кадров, длительность видео). Полученная информация доступна только в теле ответа и нигде не сохраняется.

**Важно!** Событие в видеоаналитике никак не связано с событиями, генерируемыми обработчиком.

В группе настроек «LUNA\_REMOTE\_SDK\_VIDEO\_SETTINGS» сервиса Remote SDK можно задать настройки обработки видео, такие как количество рабочих процессов декодера, временная директория для хранения видео пр.

**Примечание.** В текущей версии видео декодируется только на CPU. В будущих релизах будет добавлено декодирование на GPU.

В настоящий момент поддерживается только видеоаналитика подсчета количества людей.

### **Видеоаналитика подсчета количества людей**

Событие видеоаналитики подсчета количества людей начинается тогда, когда на последнем кадре из указанного количества подряд идущих кадров (параметр «probe\_count») появляется указанное число людей (параметр «people\_count\_threshold»). Например, если минимальное количество людей равно 10, а количество подряд идущих кадров равно 3, то в случае наличия на 3 кадрах 10 человек начнется событие с 3го кадра. Если на 2 кадрах 10 человек, а на 3 кадре 9 человек, то событие не начнется.

По умолчанию обрабатываются каждые 10 кадров. При необходимости можно настроить частоту обработки кадров с помощью параметра «rate» (например, обрабатывать каждые 3 секунды или каждые 3 кадра).

Для такой видеоаналитики также доступна возможность настроить прямоугольную область интереса ROI (параметр «roi») на кадре (координаты «x», «y», ширина, высота, единицы из-

мерения — проценты или координаты). Принцип работы области интереса ROI аналогичен FaceStream.

В каждом событии видеоаналитики подсчета количества людей содержится следующая информация:

- «event\_id» — идентификатор события
- «max\_people\_count» — максимальное количество людей, обнаруженное на обработанном сегменте видео
- «video\_segment» — информация о сегменте видео (время начала события, время конца события, номер первого кадра, номер последнего кадра)
- «frames\_estimations» — массив с информацией об оценках на каждом кадре, содержащий следующую информацию:
  - \* номер кадра
  - \* время, соответствующее кадру
  - \* количество людей

См. подробную информацию о новом ресурсе в [спецификации OpenAPI](#).

- SDK обновлен до версии 5.17.0.

В данной версии SDK обновлен [эстиматор LivenessOneShotRGB](#).

- В сервис API добавлен встроенный плагин, позволяющий выполнять часть запросов к сервису LUNA Streams через сервис API.

Перед использованием плагин необходимо включить и настроить (опционально).

Плагин «luna-streams.py» расположен в контейнере сервиса API по пути /srv/luna\_api/plugins и включается с помощью настройки «LUNA\_API\_ACTIVE\_PLUGINS».

Плагин можно настроить в настройке «LUNA\_API\_PLUGINS\_SETTINGS». В настройке можно задать адрес и версию API запущенного сервиса LUNA Streams, а также таймауты соединения.

Пример содержимого настройки «LUNA\_API\_PLUGINS\_SETTINGS» для плагина «luna-streams.py»:

```
{
  "luna-streams": {
    "luna-streams-address": {
      "origin": "http://127.0.0.1:5160/1",
      "api_version": 1
    },
    "luna-streams-timeouts": {
      "request": 60,
      "connect": 20,
    }
  }
}
```

```

        "sock_connect": 10,
        "sock_read": 60
    }
}
}

```

Для разграничений прав доступа для запросов к LUNA Streams добавлена возможность указывать пользовательский ключ для токена. Для этого нужно создать новый или обновить существующий токен, передав пользовательский ключ «streams» с разрешениями для токена.

Для проксирования запросов из сервиса LUNA API к сервису LUNA Streams нужно указывать разрешение с именем «streams». Другие пользовательские названия ключей не подходят. Использование пользовательского ключа в иных случаях не приведет ни к какому результату.

Можно задать следующие разрешения:

- «creation» (POST-запросы)
- «view» (GET-запросы)
- «modification» (PUT-запросы и PATCH-запросы)
- «deletion» (DELETE-запросы)

См. раздел [«Plugins»](#) руководства разработчика для получения перечня возможных запросов.

Информация также будет добавлена в руководство администратора FaceStream в ближайший релиз.

- Во все сервисы LUNA PLATFORM добавлен новый запрос «get list of plugins», позволяющий получить список импортированных плагинов и их статус.
- В сервис Lambda добавлен новый тип Tasks-lambda, предназначенный для реализации дополнительных пользовательских длинных типов задач.

Примеры возможного функционала длинных задач:

- открепить лица от списка, если лица не похожи на указанное лицо
- удалить дубликаты лиц из списка
- рекурсивно найти события, похожие на указанное изображение

Lambda-tasks создается аналогично остальным типам с запроса [«create lambda»](#).

После создания lambda необходимо выполнить запрос [«lambda task»](#) для создания задачи Lambda. Результаты задачи/подзадачи можно получить с помощью стандартных запросов сервиса Tasks.

Задача Lambda создается согласно общему процессу создания задач, **за исключением того**, что вместо «рабочего процесса» Tasks используется сервис Lambda.



См. дополнительную базовую информацию в разделе «Tasks-lambda» руководства администратора.

См. подробную информацию и примеры кода в [руководстве разработчика сервиса Tasks](#).

Также для задач Lambda доступна возможность создания расписания.

- Улучшено описание ошибок для некоторых случаев валидации тела запроса [«generate events»](#) при использовании схемы «multipart/form-data».

### Исправленные ошибки

- Исправлена генерация некорректного времени истечения токена при его создании/обновлении в случаях, когда установлено локальное время сервера («STORAGE\_TIME» = «LOCAL»).
- Исправлена ошибка, из-за которой при создании задачи Linker с большим количеством лиц происходило избыточное открытие соединений с базой данных, что приводило к зависанию задачи.
- Исправлена ошибка, из-за которой невозможно было выполнить запрос «create new events» к сервису Events с ненулевыми полями «location».

Попытка выполнить подобный запрос заканчивалась ошибкой с кодом «12047» и описанием «Failed to validate input msgpack. Path: „[0].location.<location\_filed>“, message: „[0].location.<location\_filed> must be string“».

- Исправлена ошибка ['a coroutine was expected, got None](#), возникавшая при использовании плагина отправки событий с большим количеством аргументов.

См. подробную информацию и пример плагина отправки событий в разделе [«Plugins»](#) руководства разработчика сервиса Handlers.

- Исправлена ошибка отсчета текущего времени («now-time») для фильтров «create\_time\_\_lt» и «create\_time\_\_gte», указываемых в политике «match\_policy» > «candidates» при создании обработчика.

Раньше текущее время отсчитывалось от времени создания обработчика. Теперь же текущее время будет отсчитываться от времени генерации события.

- Исправлена проверка лицензии на функцию ISO.

Ранее, если лицензируемая функция ISO была отключена, то запросы на выполнение любых оценок завершались ошибкой с кодом «11055» и описанием «License problem: ISO feature is disabled».

- Исправлена обработка детекций в запросе [generate stream events \(beta\)](#).

Теперь если биометрические образцы передаются с типом «sources» > «source\_type» > «detections», то будет выполнена детекция и экстракция этих биометрических образцов.

- Исправлена ошибка отсутствия поля «unknown» в описании запроса [liveness](#) в спецификации OpenAPI сервиса Backport3.

## LUNA PLATFORM v.5.54.0

### Изменения

- В запросы «[create token](#)» и «[replace token](#)» добавлен query-параметр «grant\_all\_permissions», позволяющий создать токен со всеми разрешениями или добавить все разрешения уже к существующему токenu.
- Добавлен новый запрос «[get accounts](#)».

В запросе можно указать query-параметр «targets» для фильтрации аккаунтов по полям «account\_id», «login», «account\_type», «description», «create\_time», «last\_update\_time».

Фильтрация по вышеописанным полям также добавлена в запрос «[get account](#)».

- В сервисах Admin и Tasks при выполнении задачи Additional extraction теперь поле «options» не является обязательным, если параметр «extraction\_target» установлен в «basic\_attributes».
- В сервис Events добавлен новый запрос «[get list of plugins](#)», позволяющий получить список импортированных плагинов и их статус.

В следующем релизе данный запрос будет добавлен во все остальные сервисы.

- В руководство администратора добавлен раздел «[Use tagged settings](#)», описывающий запуск сервисов с настройками, отличными от стандартных.

Также в руководство по установке добавлен раздел «[Service arguments](#)», описывающий основные аргументы сервисов и способы их передачи.

### Исправленные ошибки

- Исправлено отображение пути (строка «Path») при валидации некоторых тел запросов с некорректной структурой JSON.
- Исправлена ошибка, при которой запросы к ресурсу «[/features](#)» завершались ошибкой «Internal server error», когда лицензия не содержала какую-либо лицензируемую функцию.
- Исправлена некорректная обработка биометрического образца или бинарного изображения, передаваемых в качестве исходного изображения, в запросе [generate stream events \(beta\)](#).
- Исправлена ошибка, связанная с отсутствием проверки наличия функции VLMATCH сервисом Python Matcher при подключении к базам данных Faces и Events.

## LUNA PLATFORM v.5.53.0

### Изменения LP

- Обновлен образ VisionLabs для PostgreSQL с 12 версии на 16 версию.

Если ранее использовался данный образ, то для перехода на новую версию необходимо самостоятельно выполнить миграцию согласно [официальной документации](#). При необходимости можно продолжить использовать PostgreSQL 12, указав образ «postgis-vmatch:12» в команде запуска контейнера.

Монтирование данных PostgreSQL 12 из директории «/var/lib/luna/postgres» в контейнер для PostgreSQL 16 приведет к ошибке.

В документацию по обновлению добавлен раздел «Миграция с PostgreSQL 12 на PostgreSQL 16», содержащий напоминание о необходимости миграции.

- Добавлена возможность обнаруживать подмену лиц с помощью технологии DeepFake на фотоизображениях.

Параметр «estimate\_deepfake» добавлен в запросы [«create handler»](#), [«create verifier»](#) и [«sdk»](#).

В структуру события добавлено поле «deepfake». Данное поле может быть использовано в качестве значения для поля «target» или в качестве фильтра для получения события с помощью GET-запросов.

В результате оценки Deepfake могут вернуться следующие результаты:

- «prediction» = «fake» — человек не является реальным;
- «prediction» = «real» — человек является реальным;
- «score» = [0...1] — степень достоверности выполнения оценки.

В запросах [«create handler»](#) и [«create verifier»](#) доступна возможность задать порог «real\_threshold» и режим работы «mode». В запросе [«sdk»](#) будут использованы значения данных параметров по умолчанию без возможности явного указания.

С помощью порога «real\_threshold» можно задать значение в диапазоне [0...1], ниже которого система будет считать, что человек не является реальным.

Доступно два режима работы:

- «mode» = «1» — упрощенный режим работы;
- «mode» = «2» (по умолчанию) — режим работы с использованием дополнительной модели нейронной сети для предварительной оценки. Если результат предварительной проверки определил, что лицо является поддельным, то в теле ответа будет возвращен результат «score» = «0» и «prediction» = «fake».

Для выполнения оценки также необходимо выполнить следующие требования к изображению:

- положение головы: «pitch» = [-20...20]
- положение головы: «yaw» = [-30...30]
- ширина лица: «face\_width» > 150

Фильтр «deepfake» также можно использовать в запросах на сравнение.

В обработчик и верификатор также добавлен параметр «deepfake\_states», позволяющий отфильтровать события по предполагаемому результату оценки Deepfake.

Поле «deepfake» также добавлено в фильтры для задач Clustering, Exporter, Cross-matching и Linker. Для задач Exporter и Reporter поле поддержано в качестве колонки.

Поле «deepfake» также поддержано в качестве фильтра в запросе [«ws handshake»](#).

- В сервис API добавлен новый запрос [«get list of plugins»](#), позволяющий получить список импортированных плагинов и их статус.
- В политики обработчика и верификатора «storage\_policy» добавлена новая политика «callbacks», с помощью которой можно отправлять сгенерированные события (уведомления) во внешнюю систему по указанному URL.

Фактически, callback'и представляют собой аналог отправки уведомлений через веб-сокеты, но с ключевым отличием в том, что они используют принципы HTTP-вебхуков, что предоставляет более гибкий и настраиваемый механизм для отправки уведомлений в сторонние системы. Можно настроить тип протокола, адрес внешней системы, параметры запроса и данные авторизации.

События, отправленные с помощью поля «callbacks» имеют формат, соответствующий формату запроса [«generate events»](#).

См. подробную информацию в запросах [«create handler»](#) и [«create verifier»](#).

- Обновлена структура входящих данных Handlers-lambda.

Это означает, что все существующие lambda должны быть переработаны и пересозданы в соответствии с новой структурой.

Все примеры также были обновлены.

Пример старой структуры входящих данных: {"body": getImage("empty.jpeg"), "filename": "empty.jpeg", "source\_type": 0}

Пример новой структуры входящих данных: {"source": {"body": getImage("empty.jpeg")}, "filename": "empty.jpeg", "source\_type": "raw\_image"}

- Теперь образ Kaniko executor (образ для построения lambda) должен находиться в registry, указанном в настройке «LAMBDA\_REGISTRY».

В руководство по установке добавлена инструкция по переносу образа Kaniko executor из registry VisionLabs в пользовательский registry.

- Добавлены новые настройки соединения с Redis и Redis Sentinel.

В группы настроек «REDIS\_DB\_ADDRESS», «TASKS\_REDIS\_DB\_ADDRESS», «LUNA\_ATTRIBUTES\_DB», «BACKPORT3\_EVENTS\_DB\_ADDRESS» добавлен параметр «user».

В секцию «sentinel» вышеперечисленных настроек добавлены параметры «user» и «password».

- Обновлен Guardant с версии 3.15 до 3.21.

## LUNA PLATFORM v.5.51.6

### Изменения

- Добавлена поддержка формата логирования [ECS](#).

Для использования нового формата необходимо задать значение «ecs» в настройке «format» секции «LUNA\_service\_LOGGER».

При использовании значения «ecs» в логах будут возвращаться следующие поля:

- «http.response.status\_code» — содержит код состояния ответа HTTP (200, 404, 500 и т.д.);
- «http.response.execution\_time» — содержит информацию о времени, затраченном на выполнение запроса и получение ответа;
- «http.request.method» — содержит метод HTTP-запроса (GET, POST, PUT и т.д.);
- «url.path» — содержит путь в URL-адресе запроса;
- «error.code» — содержит код ошибки, если запрос завершается с ошибкой.

- Добавлена возможность создавать [расписание](#) для задачи Reporter.
- В запросы «[create lambda](#)» и «[update lambda](#)» добавлен новый параметр «base\_image», позволяющий явно указать имя базового образа для построения Docker-контейнера.

Ранее можно было использовать только два заранее перенесенных на пользовательский registry образа — «lpa-lambda-base» (базовая функциональность) и «lpa-lambda-base-fsdk» (базовая функциональность и функциональность для использования FSDK).

Пользовательский образ lambda предназначен для сложных lambda, которым не хватает возможностей, реализованных в базовых образах «lpa-lambda-base» и «lpa-lambda-base-fsdk» (например, многоэтапное создание Docker-контейнера или включение больших библиотек или данных в Docker-контейнер).

Для пользовательского образа lambda должны соблюдаться следующие условия:

- образ должен быть основан на образах «lpa-lambda-base» или lpa-lambda-base-fsdk»
- образ не должен удалять или изменять какие-либо установленные зависимости (python3, gcc и т. д.)

- образ должен находиться в том же registry, что и другие базовые лямбда-образы.
- В сервис Lambda добавлена возможность создания пользовательских точек мониторинга для экземпляров lambda.  
Например, можно отправить данные о том, сколько времени потребовалось для загрузки или обработки изображений.  
См. подробную информацию в разделе «Monitoring» руководства разработчика сервиса Lambda.
- Обновлена логика обработки ограничивающих прямоугольников в запросе [generate stream events \(beta\)](#).  
Теперь исходное изображение тела (параметр «body») не требуется для сохранения ограничивающих прямоугольников лица/тела (поле «events» > «detections» > «samples» > «face»/«body» > «detection» > «rect»).
- В руководство по активации лицензии LUNA PLATFORM добавлен раздел «Смена вендора» с инструкциями по переходу с вендора HASP на Guardant и наоборот.

#### Исправленные ошибки

- Удален неиспользуемый параметр «FETCH\_EXTERNAL\_IMAGE\_TIMEOUTS» из настроек сервиса API.
- Исправлено появление ошибки «Internal server error» при выполнении запроса [«get system info»](#), когда в базе данных Influx отсутствовали требуемые измерения.

## LUNA PLATFORM v.5.51.4

#### Изменения

- Добавлена возможность создавать [расписание](#) для задач Estimator, Clustering, Exporter, Cross-matching, Roc-curve calculating и Additional extraction.  
При создании расписания задачи Estimator невозможно указать ZIP-архив в качестве источника изображений.
- В мониторинг сервиса Python Matcher добавлена новая серия «Matching-Process», содержащая информацию по выполнению сравнения — время сравнения, время получения кандидата из БД и пр.  
См. список всех полей и тегов новой серии в разделе «Monitoring» руководства разработчика сервиса Python Matcher.
- Добавлен мониторинг запросов и ошибок для сервиса Lambda.

См. подробную информацию в разделе «Monitoring» руководства разработчика сервиса Lambda.

### Исправленные ошибки

- Исправлена ошибка, при которой запросы к ресурсу [«/features»](#) завершались ошибкой «Internal server error» при использовании бессрочной лицензии.
- Исправлена ошибка, которая в некоторых случаях приводила к разрыву соединения через веб-сокеты.
- Исправлена ошибка в спецификации OpenAPI, при которой в телах запросов [«matching faces»](#), [«human body matching»](#) отображались лишние типы кандидатов.
- Исправлена ошибка, приводившая к ошибке «Internal server error», если в теле запроса [«generate stream events»](#) (запрос в состоянии бета-тестирования) передавался некорректный msgpack.
- Исправлена ошибка, при которой в ответе на запрос [«human body matching»](#), содержащий различные типы кандидатов и эталонов в фильтрах, возвращалась ошибка с кодом «12022» с описанием «message: 'filters.origin must be one of [„faces“, „events“, „attributes“]».

Теперь ошибка содержит не список всех типов, а конкретный тип, который должен быть указан в фильтре.

## LUNA PLATFORM v.5.51.0

### Изменения

- SDK обновлен до версии 5.16.0.

В LUNA PLATFORM добавлена поддержка новой 62-ой модели нейронной сети для извлечения биометрических шаблонов лиц.

- Добавлен новый сервис Lambda, предназначенный для работы с пользовательскими модулями, имитирующими функционал отдельного сервиса. В настоящий момент функционал находится в бета-тестировании.

**Полноценная работа с сервисом Lambda возможна только при разворачивании сервисов LUNA PLATFORM в Kubernetes.** Для использования необходимо самостоятельно развернуть сервисы LUNA PLATFORM в Kubernetes или обратиться за консультацией к специалистам VisionLabs. При необходимости можно использовать Minikube для локальной разработки и тестирования, обеспечивая таким образом Kubernetes-подобную среду без необходимости управления полноценным продакшн Kubernetes-кластером.

Сервис позволяет написать и использовать собственный обработчик или написать внешний сервис, который будет тесно взаимодействовать с LUNA PLATFORM и сразу иметь

несколько функций, типичных для сервисов LP (таких как логирование, автоматическая перезагрузка конфигураций и т.д.). Пользователю достаточно лишь написать код и отправить в сервис Lambda, после чего можно будет использовать свой модуль без разворачивания дополнительных контейнеров и пр.

Сервис создает образ Docker на основе ZIP-архива с кодом разработчика, а затем запускает его в кластере Kubernetes. Пользовательский модуль, запущенный в кластере Kubernetes, называется **lambda**.

Для работы с сервисом Lambda нужна **отдельная лицензируемая функция**. В запросы `«get platform features»`, `«get system info»` и `«get license»` добавлены поля `«lambdas»`, отображающие статус лицензируемой функции.

В токен добавлено новое разрешение `«lambda»`, регулирующее права `«creation»`, `«view»`, `«modification»`, `«deletion»` для `lambda`.

Для запуска и работы с сервисом требуется:

- наличие запущенных сервисов Licenses и Configurator\*;
- наличие бакета S3 для хранения архивов с кодом разработчика;
- наличие Docker registry для хранения Docker-образов;
- наличие кластера Kubernetes.

\* во время своей работы, `lambda` будет дополнительно взаимодействовать с некоторыми сервисами LUNA PLATFORM. Перечень сервисов зависит от типа `lambda`.

Кроме вышеперечисленных требований к окружению, необходимо соблюдать требования к написанию кода и архива, а также правильно настроить сервис. См. дополнительную информацию в документации.

`Lambda` создается с помощью запроса `«create lambda»`, где указывается архив, имя и её тип (см. ниже).

`Lambda` может быть двух типов:

### **Handlers-lambda**

Такой тип предназначен для замены функционала классического обработчика. `Lambda`-обработчик может использоваться в двух случаях:

- как пользовательский обработчик, который имеет свою собственную схему ответа, которая может отличаться от ответа классических обработчиков и не может быть должным образом использована в других сервисах LUNA PLATFORM (например, `{ "similarity": 0.123 }`);
- как пользовательский обработчик, который имитирует ответ классического обработчика (например, `{ "images": [ { ... } ], "events": [ { ... } ] ... }`). Такой обработчик должен соответствовать схеме ответа запроса на генерацию события и правильно обрабатывать данные, чтобы другие сервисы могли его использовать.



В связи с добавлением Handlers-lambda была переработана схема тела ответа запроса [«create handler»](#). В тело запроса добавлен новый параметр `«handler_type»`, принимающий три значения — «0» (статический обработчик), «1» (динамический обработчик), «2» (обработчик lambda). Для использования значения «2» требуется также указать идентификатор `«lambda_id»` (см. логику работы ниже). Параметр `«is_dynamic»` считается устаревшим (Deprecated) и будет проигнорирован при использовании параметра `«handler_type»`.

Доступно два способа взаимодействия с Handlers-lambda:

1. С помощью запросов [«generate events»](#) или [«estimator task»](#). Для использования данных запросов необходимо предварительно [создать обработчик](#), указав `«handler_type» = «2»` и `«lambda_id»`, полученный на этапе создания lambda. В ответе на генерацию события будет выдан пользовательский результат, формат которого либо совпадает с форматом классического события и позволяет использовать его сервисами LP в дальнейшем, либо не совпадает.
2. С помощью запроса [«proxy post request to lambda»](#), когда не предполагается имитировать ответ классического обработчика, поскольку сервисы LP предполагают строгий формат ответа на запрос генерации события.

Например, с помощью Handlers-lambda можно реализовать логику отправки, извлечения и сравнения двух биометрических шаблонов в одном запросе. В ответе может быть выдана только схожесть кандидатов без лишней информации. В классическом сценарии использования LUNA PLATFORM пользователь не может выполнить данный сценарий и вынужден писать логику на стороне внешней системы. См. примеры кода в руководстве разработчика.

### **Standalone-lambda**

Такой тип предназначен для реализации самостоятельного функционала для выполнения тесной интеграции с LUNA PLATFORM. Для работы с таким типом используется запрос [«proxy post request to lambda»](#) с собственной схемой запроса и ответа. Данный тип предназначен для узкой целевой аудитории и достаточно сложен в реализации.

С помощью Standalone-lambda можно написать сервис, реализующий запись видеопотока в файл и сохраняющий его в сервис Image Store для последующей обработки приложением FaceStream.

Также в сервис LUNA Dashboards добавлено создание дашборда для сервиса Lambda.

См. диаграммы последовательности создания и обработки lambda в разделе [«Lambda diagrams»](#).

См. дополнительную информацию для базового ознакомления с функционалом сервиса Lambda в разделе [«Lambda service»](#) руководства администратора.

См. дополнительную информацию для более глубокого погружения в [документации разработчика сервиса Lambda](#). В руководстве приведен набор шагов для быстрого начала работы

с сервисом.

- Ответ на запрос «[get system info](#)» к сервису Admin расширен дополнительной информацией по статистике использования эстиматоров и изображений.

Добавлено поле «`estimators_performance_stats`», отображающее месяц, имя эстиматора, среднее время выполнения эстимации и средний размер батча для каждого эстиматора.

Пример:

```
"estimators_performance_stats": [  
  {  
    "month": "2021-09",  
    "name": "body_descriptor",  
    "execution_time": 0.029135203011156546,  
    "batch_size": 1.0952380952380951  
  }  
]
```

Добавлено поле «`image_processing_stats`», отображающее месяц, среднее время декодирования изображения, количество изображений по их размеру и количество изображений по высоте лица. Пример:

```
"image_processing_stats": [  
  {  
    "month": "2021-09",  
    "image_load_time": 0.006479793069339647,  
    "image_size": {  
      "w_1000_h_1050": 4,  
      "w_1000_h_800": 212,  
    },  
    "face_detection_size": {  
      "h_100": 2,  
      "h_180": 197  
    }  
  }  
]
```

Здесь:

- «`w_1000_h_800`»: 212 – 212 изображений с шириной 1000 пикселей и высотой 800 пикселей
- «`h_180`»: 197 – 197 изображений с высотой лица 180 пикселей

Для изображений применяется округление до ближайших 50 пикселей. Например, если у

изображения ширина 105 пикселей, то она будет округлена до 150 пикселей. Все изображения с одинаковыми округленными значениями будут находиться в одном и том же бакете.

**Примечание.** Для использования новой статистики необходимо выполнить команду `python influx2_cli.py create_usage_task --luna-config http://127.0.0.1:5070/1` после запуска сервиса Admin (см. руководство по установке)

- В пользовательский интерфейс сервиса Admin добавлена вкладка Schedules, позволяющая управлять расписанием задач и создавать расписание для задачи Garbage Collection.

Во вкладке отображаются все созданные расписания задач и вся соответствующая информация (статус, идентификатор, Cron-строка и т.д.). Во вкладке также можно создавать, удалять и редактировать расписания, а также управлять отложенным запуском (ставить на паузу и запускать остановленные расписания).

См. подробную информацию в разделе «Пользовательский интерфейс сервиса Admin».

### Исправленные ошибки

- Исправлено отсутствие следующих параметров сортировки аккаунтов и токенов:
  - фильтров «create\_time» и «create\_time\_\_gte» в запросе `«get tokens»`;
  - полей «create\_time» и «last\_update\_time» в телах ответов на запросы `«get account»`, `«get tokens»`, `«get token»`.
- Исправлена ошибка, при которой во время запуска сервиса Admin не выполнялась проверка соединения и healthcheck к сервису Remote SDK.
- Исправлена ошибка, при которой во время миграции базы данных сервиса Handlers на версию v.3.0.0 некорректно обновлялось поле «detect\_policy» > «estimate\_people\_count» в существующих обработчиках.

В результате после обновления при попытке использования обработчиков с таким полем могла возникать ошибка с кодом «12022» и описанием «Failed to validate input json. Path: „policies.detect\_policy.estimate\_people\_count“, message: „value is not a valid dict“».

## LUNA PLATFORM v.5.49.1

### Изменения

- Добавлена возможность задать расписание выполнения задач Garbage collection и Linker.

С помощью расписания можно гибко управлять временем начала выполнения задач. Например, можно настроить регулярное расписание для очистки событий старше одного месяца каждую пятницу в ночное время.

Расписание создается с помощью запроса [«create tasks schedule»](#) к сервису API, в котором указывается содержимое создаваемой задачи и временной интервал её запуска. Для указания временного интервала используются [Cron-выражения](#).

При необходимости можно создать отложенное расписание, а затем активировать его с помощью параметра «action» = «start» запроса [«patch tasks schedule»](#). Аналогично можно остановить работу задачи по расписанию с помощью «action» = «stop». Для удаления расписания можно воспользоваться запросом [«delete tasks schedule»](#).

Права на работу с расписаниями задаются в токене разрешением «task». Это означает, что если у пользователя есть разрешение на работу с задачами, то он также сможет воспользоваться расписанием.

В руководство по установке и по запуску с помощью Docker Compose добавлен пример CURL-запроса для запуска ежедневного расписания для задачи Garbage collection для событий старше 30 дней с удалением БО и исходных изображений.

В базу данных Tasks добавлена новая таблица «schedule», содержащая всю информацию о расписании выполнения задач. Получить информацию из БД о созданном расписании можно с помощью запросов [«get tasks schedule»](#) и [«get tasks schedules»](#).

См. подробную информацию в разделе «Запуск задач по расписанию» руководства администратора.

- Теперь если формат запроса равен «application/msgpack» и входные данные не соответствуют формату MessagePack, то будет возвращена ошибка с кодом «12047» и описанием «Failed to validate input msgpack. Path: „{}“, message: „{}“».

Ранее возвращался код ошибки «12022» с описанием «Failed to validate input json. Path: „{}“, message: „{}“».

- В запрос [«generate events»](#) добавлена поддержка указания пользовательских метаданных для изображения, передаваемого с помощью схемы «multipart/form-data» в поле «image».

Ранее указание пользовательских метаданных было доступно только для исходных изображений (поля «image\_origin»). Также исправлена ошибка, из-за которой в описании запроса [«generate events»](#) не отображалось описание про указание пользовательских метаданных для исходных изображений.

Если метаданные будут предоставлены одновременно в обоих полях, то будет использовано значение из поля «image\_origin».

Метаданные передаются с помощью заголовков вида «X-Luna-Meta-<user\_defined\_key>:<user\_defined\_value>», которые отправляются в сервис Image Store при сохранении изображения во время генерации события.

- Обновлен сервис LUNA Dashboards до версии 0.0.8.

В этой версии добавлен отсутствующий дашборд для сервиса Remote SDK.

Также в комплект поставки добавлен архив с плагином «grafana-piechart-panel», предназначенным для ручной установки дашбордов.

### Исправленные ошибки

- Исправлен ряд следующих ошибок при запросах к сервису Events:
  - Исправлено появление ошибки с кодом состояния 500 при запросе к ресурсу «/events/stats» с параметром «filters» > «value», имеющим тип «bool».
  - Исправлено появление ошибки с кодом состояния 500 при запросе к ресурсу «/events/stats» и использовании недопустимых операторов в фильтре, который был равен «null».
  - Исправлено появление ошибки с кодом состояния 500 при запросе к ресурсу «/events», когда в JSON-запросе содержалось число, превышающее предел «int64».
  - Исправлена ошибка с некорректной фильтрацией статистики событий по «null» с использованием операторов «neq» и «nin».
- Актуализирована схема тела запроса для задачи Estimator для типа источника «zip» в спецификации OpenAPI сервиса API.
- Исправлена ошибка, при которой загрузка значений настроек сервиса API из переменных окружения не работала для:
  - вложенных ключей настройки, например, «VL\_SETTINGS.LUNA\_SERVICE\_NAME\_DB.DB\_SETTINGS.CONNECTION\_POOL\_SIZE=10»
  - значений типа массива, например, «VL\_SETTINGS.OTHER.LUNA\_SERVICE\_NAME\_ACTIVE\_PLUGINS=[logger]»
- Исправлена ошибка в запросе «[save event](#)» спецификации OpenAPI сервиса API, при которой параметр «detections» > «samples» > «face» > «detection» > «face\_quality» > «checks» > «check\_face\_properties\_request» > «max» отображался как обязательный.

## LUNA PLATFORM v.5.47.4

### Изменения

- Настройки баз данных <service\_name>\_DB всех сервисов расширены новым опциональным параметром «dsn», в котором задается строка DSN, которая может содержать различные настройки для управления подключением к базе данных, такие как множественные хосты, аутентификационные данные, порт и другие (настройки зависят от типа БД).

За счет внедрения новой настройки, классические настройки для подключения к БД («db\_host», «db\_port», «db\_name», «db\_user» и «db\_password») во всех настройках всех сервисов стали опциональными.

При необходимости можно комбинировать строку DSN и классические настройки, однако строка DSN является более приоритетной. Можно частично заполнить строку DSN (например, «postgres01,postgres02/luna\_handlers»), и тогда недостающие параметры будут заполнены из значений параметров «db\_host», «db\_port», «db\_name», «db\_user» и «db\_password».

После выполнения обновления на новую версию LUNA PLATFORM, параметр «dsn» не появится во вкладке «Settings» в Configurator. Для использования DSN, необходимо вручную указать соответствующий параметр. Ниже приведен пример указания параметра «dsn» в группе параметров «LUNA\_FACES\_DB»:

```
{
  "dsn": "luna:luna@postgres01:5432,postgres02:5432/luna_faces?some_option=some_value"
  "db_settings": {
    "connection_pool_size": 5
  }
}
```

См. подробную информацию в разделе «Подключение к БД использованием DSN» руководства администратора.

- Теперь в ответе на запросы [«get system info»](#) и [«get license»](#) в поле «expiration\_time» будет отображаться значение «perpetual» для бессрочной лицензии.
- Теперь для плагина сравнения «Thin faces» можно настроить максимальный размер списка для выполнения сравнения. Если список превышает установленное количество, то для сравнения будет использован Python Matcher или LIM Indexed Matcher.

Максимальный размер списка устанавливается с помощью указания переменной окружения «VL\_SETTINGS.THIN\_FACE.MAX\_LIST\_LENGTH» при запуске сервиса Python Matcher.

Пример указания переменной при запуске сервиса Python Matcher через Docker: `docker run \ -e VL_SETTINGS.THIN_FACE.MAX_LIST_LENGTH=100.`

### Исправленные ошибки

- Исправлено появление ошибки 31006 «Unexpected behavior of the {matcher\_type} matcher: {plugin\_error\_description}» при выполнении сравнения с использованием атрибута-эталона и плагина сравнения.

## LUNA PLATFORM v.5.47.1

### Изменения

- Обновлен внутренний механизм взаимодействия сервиса Tasks с «рабочими процессами».

Теперь вместо отправки HTTP-запросов «рабочим процессам», сервис Tasks будет взаимодействовать с ними с помощью Redis.

В настройке сервиса Tasks добавлена новая группа настроек «TASKS\_REDIS\_DB\_ADDRESS», где:

- «host» — IP-адрес Redis
- «port» — порт Redis
- «password» — пароль для авторизации в Redis
- «number» — номер базы данных Redis (от 0 до 15). Каждый номер соответствует отдельной базе данных, что позволяет разделить данные.

При обновлении на текущую версию LUNA PLATFORM, значения вышеописанных настроек будут заполнены в соответствии с группой настроек «LUNA\_ATTRIBUTES\_DB» сервиса Faces. Это позволит использовать для сервиса Tasks тот экземпляр Redis, который используется для сервиса Faces. При необходимости разделения данных сервисов Faces и Tasks в Redis, можно указать пользовательские настройки в группе настроек «TASKS\_REDIS\_DB\_ADDRESS» после запуска сервиса (например, указать номер базы данных текущего экземпляра Redis или указать адрес другого экземпляра Redis).

Также «рабочие процессы» сервиса Tasks теперь не имеют доступа в БД Tasks.

См. обновленную диаграмму последовательности работы сервиса Tasks в разделе [«Tasks diagrams»](#).

- В контейнерах сервисов Events и Licenses обновлена версия Python до 3.11.

Поддержка более старых версий Python прекращена.

### Исправленные ошибки

- Исправлена ошибка в сервисах Python Matcher, Handlers, Sender и Backport 3, из-за которой проверка подключения к Redis Sentinel не проходила с первого раза.
- Исправлена ошибка, из-за которой сервис Sender не восстанавливал соединение с Redis после перезагрузки.

## LUNA PLATFORM v.5.46.1

### Изменения

- Функционал для работы с нейронными сетями (детекция, эстимация и извлечение) перенесен из сервиса Handlers в новый сервис Remote SDK. Это позволило сделать сервис Handlers опциональным и отключать его в настройке «ADDITIONAL\_SERVICE\_USAGE», когда нет необходимости работы с обработчиками.

Теперь при работе с обработчиками, сервис Handlers перенаправляет запросы на детекцию, эстимацию и извлечение к сервису Remote SDK, а затем обрабатывает полученный результат.

Если сервис Handlers отключен, то:

- запуск сервиса API приведет к отсутствию возможности использования следующих запросов: «[detect faces](#)», «[extract attributes](#)», «[estimator task](#)», все запросы на ресурс «[/handlers](#)», все запросы на ресурс «[/verifiers](#)»;
- запуск сервиса Tasks приведет к отсутствию возможности выполнения задач «Additional extraction» и «Estimator»;
- запуск сервиса Admin приведет к отсутствию возможности выполнения задачи «Additional extraction»;

Все нейронные сети и настройки, связанные с детекцией, эстимацией и извлечением, также перенесены в сервис Remote SDK. Теперь для выбора таких настроек необходимо в пользовательском интерфейсе Configurator вводить в поле «Service name» значение «luna-remote-sdk» вместо «luna-handlers».

Запросы к ресурсам «[/iso](#)», «[/sdk](#)», «[/liveness](#)» теперь выполняются напрямую к сервису Remote SDK без участия Handlers.

Отключение неиспользуемых нейронных сетей теперь выполняется с помощью передачи соответствующей переменной окружения (например, `--env=EXTEND_CMD="--enable-all-estimators-by-default=0 --enable-face-detector=0`) в команде запуска контейнера Remote SDK, а не контейнера Handlers.

Теперь вместо запуска Handlers на GPU необходимо запускать Remote SDK на GPU (флаг `--gpus device=0`).

- SDK обновлен до версии 5.15.0. Основные изменения SDK, затрагивающие LUNA PLATFORM:
  - поддержаны 109ая и 110ая модели нейронной сети для извлечения биометрических шаблонов тел;
  - 105ая, 106ая, 107ая модели нейронной сети для извлечения биометрических шаблонов тел считаются устаревшими;
  - встроен [эстиматор CrowdEstimatorV2](#).

В данной версии LUNA PLATFORM:

- 105ая модель **была удалена** из контейнера Remote SDK;
- дефолтная модель была изменена со 107ой на 110ую (настройка «DEFAULT\_HUMAN\_DESCRIPTOR\_VERSION» сервиса Remote SDK).

**Обновление с версии, где использовалась 107ая модель (модель по умолчанию с версий 5.34.0 и выше)**



Если выполняется обновление с версии, где использовалась 107ая модель, то рекомендуется при обновлении указать 110ую модель нейронной сети в настройке «DEFAULT\_HUMAN\_DESCRIPTOR\_VERSION» и выполнить задачу «Additional extraction» после запуска сервиса Admin (см. раздел [«Launch Additional extraction task»](#) в руководстве администратора) для продолжения сравнения по старым биометрическим шаблонам тел.

### **Обновление с версии, где использовалась 105ая модель**

Если выполняется обновление с версии, где использовалась 105ая модель, то **запуск сервис Remote SDK завершится ошибкой** если не выполнить одно из следующих действий перед запуском контейнера Remote SDK:

- вручную изменить значение настройки «DEFAULT\_HUMAN\_DESCRIPTOR\_VERSION» с «105» на «110». После изменения версии нейронной сети извлечения биометрических шаблонов тел необходимо выполнить задачу «Additional extraction» после запуска сервиса Admin (см. раздел [«Launch Additional extraction task»](#) в руководстве администратора). В противном случае, поиск и сравнение по старым биометрическим шаблонам будут недоступны;
- выключить использование нейронной сети для извлечения биометрического шаблона тела с помощью передачи аргумента `--enable-body-descriptor-estimator=0` при старте контейнера Remote SDK;
- запросить у VisionLabs 105ую модель нейронной сети и перенести её в контейнер Remote SDK по инструкции, описанной в разделе [«Use non-delivery neural network model»](#) руководства администратора.

Вся вышеописанная информация добавлена в руководство по обновлению в раздел [«Change the neural network model for extracting descriptors»](#).

При запуске LUNA PLATFORM с нуля никаких дополнительных действий не требуется.

- В ресурсы [«sdk»](#) и [«/handlers/{handler\\_id}/events»](#) добавлен новый параметр «estimate\_people\_count», позволяющий выполнить оценку количества людей на изображении.

Данный функционал лицензируется отдельно.

Необходимо понимать, что такая оценка не может сравниться по точности с отдельными детекторами лиц или тел. Следует использовать её для приблизительной оценки количества людей. См. [документацию SDK](#) для более подробной информации.

В теле ответа результат возвращается в отдельном поле «image\_estimations», поскольку данная оценка не относится к лицам или телам события.

- В задаче [Estimator](#) для ZIP архива добавлены следующие параметры:
  - «prefix» — префикс ключа файла. Может использоваться для загрузки изображений из определенной директории;

- «postfix» — постфикс ключа файла. Может использоваться для загрузки изображений с определенным расширением;
- «recursive» — рекурсивное получение изображений из вложенных директорий.
- Добавлена возможность указать относительное время (формат now-time) в параметрах «create\_time\_\_gte», «create\_time\_\_lt», «end\_time\_\_gte», «end\_time\_\_lt», «insert\_time\_\_gte», «insert\_time\_\_lt» в следующих задачах:
  - [Clustering](#)
  - [Exporter](#)
  - [Linker](#)
  - [Garbage collection](#)
  - [Additional extraction](#)
  - [Cross-matching task](#)

Это может быть полезно для фильтрации данных за определенный интервал времени относительно текущего времени. Например, можно выполнить задачу «Garbage collection» за несколько последних дней.

- В контейнерах сервисов API и Admin обновлена версия Python до 3.11.  
Поддержка более старых версий Python прекращена.
- В скрипте Docker Compose «start\_platform.sh» закомментированы строки, касающиеся запуска Backport 3, Backport 4, User Interface 3 и User Interface 4.  
Теперь эти сервисы не будут запускаться при выполнении скрипта.
- В руководство по активации лицензии добавлена инструкция по активации лицензии Guardant без графического интерфейса.

Для этого необходимо дополнительно установить пакет, предназначенный для запуска интерфейсных приложений без физического вывода на экран.

### **Исправленные ошибки**

- Исправлена ошибка, из-за которой проверка подключения к Redis Sentinel не проходила с первого раза.

## **LUNA PLATFORM v.5.45.4**

### **Изменения**

- Уменьшена нагрузка на БД Faces при выполнении запросов лицензии на проверку максимально количества лиц с привязанными БШ или базовыми атрибутами.
- В сервис Image Store добавлены новые ресурсы:

- HEAD [«/1/buckets/{bucket}»](#), позволяющий проверить существование бакета
  - GET [«/1/buckets/{bucket}»](#), позволяющий получить дату создания указанного бакета
  - Из группы настроек «S3» сервиса Image Store удалена неиспользуемая настройка «bucket».
  - В контейнерах сервисов Tasks, Accounts, Configurator, Sender, Backport 3 и Backport 4 обновлена версия Python до 3.11.
- Поддержка более старых версий Python прекращена.

### Исправленные ошибки

- Добавлена отсутствующая миграция стандартного значения настройки «score\_threshold» в группе настроек «LUNA\_HANDLERS\_FACE\_DETECTOR\_SETTINGS» сервиса Handlers.
- При выполнении обновления на текущую версию, предыдущее стандартное значение «0.42» автоматически обновится до актуального стандартного значения «0.5».
- Если значение настройки «score\_threshold» отличается от стандартного, то миграция выполнена не будет.
- Некорректная структура поля «face\_detections» > «detection» события в сервисе Events приведена к правильному виду, соответствующему спецификации OpenAPI.

## LUNA PLATFORM v.5.45.3

### Изменения

- В ресурсы [«/matcher/faces»](#), [«/matcher/bodies»](#) и [«/matcher/raw»](#) добавлена поддержка заголовка Ассерта, определяющего MIME-тип ответа, который следует ожидать от клиента.
- Для заголовка доступно два значения: `application/json` (по умолчанию) и `application/msgpack`.
- В контейнерах сервисов Faces, Image Store, Handlers, Python Matcher и Python Matcher Proxy обновлена версия Python до 3.11.
- Поддержка более старых версий Python прекращена.
- Переход на новую версию Python позволил увеличить скорость выполнения сравнения на 10-20% в некоторых случаях.
- В задачу [Estimator](#) добавлен параметр «verify\_ssl», позволяющий отключить проверку SSL-сертификата для S3-подобного хранилища.
- Это позволяет использовать самоподписанный SSL-сертификат.
- Ускорено выполнение оценки параметров лиц в некоторых случаях, когда в запросе не указаны параметры, отвечающие за фильтрацию («yaw\_threshold», «fd\_score\_threshold», «liveness\_threshold» и др.).

- Ускорено выполнение запросов на сравнение в случае, когда для кандидатов-событий используются поля «target» из нижеописанного списка:
  - «match\_result»
  - «body\_detections»
  - «face\_detections»
  - «attach\_result»
  - «tags»
  - «location»

### Исправленные ошибки

- Исправлена ошибка, при которой запрос на [создание бакета S3](#) к сервису Image Store успешно выполнялся, но бакет не создавался, если в настройках сервиса не была указана настройка «region».

## LUNA PLATFORM v.5.45.1

### Изменения

- SDK обновлен до версии 5.14.0. Основные изменения SDK, затрагивающие LUNA PLATFORM:
  - обновлен [детектор лиц FaceDetV3](#);
  - обновлен [эстиматор Eyebrows](#).

Значение по умолчанию настройки `score_threshold` в секции «LUNA\_HANDLERS\_FACE\_DETECTOR\_SETTINGS» настроек сервиса Handlers изменено с **0.42** до **0.5**. Миграция настроек автоматически обновит это значение (см. раздел «Миграция базы данных Configurator» в руководстве по обновлению). Проверьте логику распознавания лиц, если вы используете значение `score_threshold`, отличное от значения по умолчанию.

**Примечание.** В ближайших релизах будет изменена версия нейронной сети для извлечения биометрических шаблонов тел по умолчанию. Необходимо будет вручную сменить версию в настройках сервиса Handlers, иначе его запуск завершится ошибкой.

- Раздел по активации лицензии LUNA PLATFORM вынесен в отдельное руководство по активации лицензии «LP\_License\_Activation\_Manual.pdf/html».
- Добавлена возможность лицензирования LUNA PLATFORM с помощью ключей Guardant.

Такой способ лицензирования требует наличия графического интерфейса системы и доступа к сети Интернет. Если сервер, где планируется использовать LUNA PLATFORM, не отвечает данным требованиям, то можно выполнить часть действий на вспомогательном сервере на ОС Windows или ОС Linux.

Старый способ лицензирования с помощью ключей HASP остается доступным.

См. подробную информацию в разделе «Активация лицензии с помощью Guardant-ключа» в новом руководстве по активации лицензии.

- Добавлена возможность фильтрации по значениям `null` (значение означает, что эстимация по атрибуту не выполнялась) для событий-кандидатов в следующих запросах:
  - «[matching faces](#)»;
  - «[human body matching](#)»;
  - «[create handler](#)»;
  - «[generate events](#)»;
  - «[save event](#)»;

Также фильтрация по значениям `null` добавлена для фильтров по событиям для следующих задач:

- «[Clustering](#)»;
- «[Exporter](#)»;
- «[Cross-matching](#)»;
- «[Linker](#)»;
- «[Estimator](#)».

Фильтрация добавлена для следующих атрибутов:

- meta
- source
- emotion
- mask
- ethnic\_group
- liveness
- gender
- apparent\_gender
- headwear\_state
- sleeve\_length
- upper\_clothing\_colors
- lower\_garment\_type
- lower\_garment\_colors
- shoes\_apparent\_color
- backpack\_state
- city
- district
- street
- house\_number
- area
- geo\_position

- track\_id

Фильтрация по null позволяет фильтровать события, сгенерированные по разным обработчикам с разными политиками, где в первом выполнялась эстимация определенного атрибута (например, состояния маски равно occluded), а во втором не выполнялась эстимация (например, состояние маски равно null), но нужно получить оба события.

- Выполнение запросов «get events» и «save event» было ускорено.
- В секцию «S3» настроек сервиса Image Store добавлен новый параметр «verify\_ssl», позволяющий отключить проверку SSL-сертификата для S3-подобного хранилища.

Это позволяет использовать самоподписанный SSL-сертификат.

- Обновлен механизм проверки соединения с сервисом Image Store.

Ранее сервис Admin выполнял проверку с помощью получения списка всех бакетов, что могло приводить к ошибке из-за отсутствия у пользователя доступа к бакетам. Теперь проверка соединения выполняется без получения списка всех бакетов.

- В группу настроек DESCRIPTORS\_CACHE сервиса Python Matcher добавлена новая подгруппа CACHED\_DATA, позволяющая задавать данные для кеширования.

В поле face\_lists можно настроить, какие именно списки будут кэшироваться, а какие будут игнорироваться. Для данного поля доступно два значения:

- include — будут кэшироваться только списки, заданные в данном разделе (для отключения нужно задать null);
- exclude — списки из этого раздела будут проигнорированы.

- В сервис Python Matcher добавлен новый пример встроенного плагина «Thin face».

Плагин «Thin face» приводится в качестве примера для быстрого сравнения лиц (объектов) с упрощенными лицами (объектами). Упрощенные лица хранятся в отдельной таблице базы данных «luna\_faces» с тремя обязательными столбцами («face\_id», «descriptor», «descriptor\_version»). При необходимости можно настроить ряд дополнительных столбцов: «account\_id», «lists», «create\_time», «external\_id», «user\_data», «event\_id», «avatar».

См. подробное описание плагина «Thin face» и инструкцию по написанию пользовательских плагинов в документе «PythonMatcherDevelopmentManual» в комплекте поставки.

## Исправленные ошибки

- В нижеперечисленных запросах спецификации OpenAPI сервиса API исправлены значения по умолчанию для некоторых параметров:

- «create handler»;
- «extract attributes»;
- «iso»;

- «sdk»;
- «create attributes»;
- «get attributes».

Также исправлено значение по умолчанию для параметра «extract\_descriptor» запроса «create descriptors» в спецификации OpenAPI сервиса Backport 3 и значение по умолчанию для параметра «policies» > «create\_face\_policy» > «set\_sample\_as\_avatar» запроса «create handler» в спецификации OpenAPI сервиса Backport 4.

- Исправлена ошибка в задаче «Estimator», при которой возвращался код состояния 500 при попытке подключения к несуществующей конечной точке (эндпоинту) S3-подобного сервера.

Теперь выдается код состояния 400 и код ошибки 12031 с содержимым «Specified bucket not available».

- Исправлена ошибка в задаче «Estimator», при которой возвращался код состояния 500 при попытке подключения к серверу Samba без авторизации.

Теперь выдается код состояния 400 и код ошибки 12031 с содержимым ошибки Samba.

- Исправлено поведение, когда в некоторых случаях ошибки прав доступа, обнаруженные при инициализации лицензии, могли не отображаться в логах сервиса Licenses.

Теперь при подобных ошибках в логах сервиса Licenses всегда будут выдаваться сообщения вида `Failed to init licensing`.

- Исправлено появление ошибки `object of type 'Image' has no len()` с кодом состояния 500 при выполнении эстимации некоторых повернутых изображений с включенным параметром извлечения EXIF-данных `use_exif_info`.
- Исправлена ошибка в сервисе Python Matcher, приводившая к перезагрузке кеша биометрических шаблонов при изменении настроек логирования.
- Исправлена ошибка, при которой невозможно было указать более 36 символов для фильтра `source` для кандидатов в запросах на сравнение.

Теперь можно указать не более 128 символов.

- Исправлена ошибка, возникавшая при миграции аккаунтов и токенов сервиса Backport 3 при обновлении с версий 5.2.0...5.28.0 на версии 5.30.0 и выше.

## LUNA PLATFORM v.5.42.0

### Изменения

- Теперь сервис Image Store может хранить любые файлы в качестве объектов (например, видеофайл).

Ранее были доступны только следующие типы объектов: json, text, zip, pdf.

Загрузить файлы можно с помощью запроса «create objects». Байты файла должны быть указаны в теле запроса, а заголовок Content-Type должен содержать MIME-тип файла (например, video/mp4). Ответ на запрос «get object» содержит заголовок Content-Disposition. Этот заголовок содержит имя файла объекта вложения (например, video\_1.mp4). Имя файла генерируется на основе object\_id и MIME-типа.

Если MIME-тип файла не получилось определить, то расширение файла будет установлено как .bin.

Теперь заголовок Accept игнорируется для запроса «get object». Ранее сервис возвращал ошибку 12023, Content type is unacceptable со статус-кодом 406, если тип содержимого объекта не совпадал с MIME-типом файла.

- Теперь при выполнении любого запроса с корректной авторизацией, в логах сервиса API выводится информация о соответствующем аккаунте.

Данный функционал позволяет определить кто именно выполнил тот или иной запрос. Это может потребоваться для информационной безопасности и администраторов систем.

Если запрос был выполнен с авторизацией типа BasicAuth или LunaAccountIdAuth, то в логах будет выдано следующее сообщение:

```
Request invoked by user (account_id: '270531af-e52e-4538-9181-628d9900a0db')
```

Если запрос был выполнен с авторизацией типа BearerAuth, то в логах будет выдано следующее сообщение:

```
Request invoked by user (account_id: '270531af-e52e-4538-9181-628d9900a0db' token_id: 'd57e16f5-e243-47d2-aa85-8b200c12d86f')
```

Если запрос был выполнен без авторизации, то в логах будет выдано следующее сообщение:

```
Request invoked by user (account_id: null)
```

В логах сервиса Accounts дополнительно выводится информация о создании токенов конкретными пользователями:

```
User with account_id: '270531af-e52e-4538-9181-628d9900a0db' create token: 'd57e16f5-e243-47d2-aa85-8b200c12d86f'
```



Логирование информации о создании токенов позволяет отследить откуда появился токен и какому пользователю он принадлежал, даже после его удаления.

- В группы проверок `face_quality` и `iso` добавлена новая проверка `shoulders_position`, определяющая наиболее вероятное состояние положения плеч из следующих состояний:
  - `non-parallel` (плечи не параллельны)
  - `parallel` (плечи параллельны)
  - `hidden` (плечи скрыты)
- Ускорено выполнение запроса `«get faces»` со всеми полями `target`.
- Уменьшено использование памяти сервисом `Tasks` в некоторых случаях использования задачи `«Cross-matching»`.

### Исправленные ошибки

- Исправлена ошибка в сервисе `Configurator`, при которой не работал скрипт `db_create.py` при нестандартных значениях конфигураций сервиса `Configurator`.
- Исправлена ошибка, при которой прогревался кэш при старте сервиса `Python Matcher` при выключенной настройке `«DESCRIPTORS_CACHE»`.
- Исправлена ошибка, при которой сервис `Python Matcher` запускался и продолжал работать в случае, когда некоторые «рабочие процессы» останавливались с ошибкой.  
«Рабочий процесс» — процесс сервиса `Python Matcher`, обрабатывающий входящие (HTTP) запросы.
- Исправлено поведение, которое могло приводить к удалению одной или всех новых подписок на события, если создание новой подписки и удаление старой происходили одновременно.
- Исправлена ошибка, при которой использование оценки EXIF данных (параметр `use_exif_info`) могло приводить к ошибке вида `Internal server error`.

## LUNA PLATFORM v.5.40.0

### Изменения

- Начиная с текущего релиза, в командах запуска контейнеров PostgreSQL, InfluxDB и Image Store прописываются пути директорий для монтирования, расположенные в корневом каталоге `/var/lib/luna/<db_or_bucket_folder>`, в отличие от предыдущих версий, где пути прописывались для определенной версии LUNA PLATFORM `/var/lib/luna/current/example-docker/<db_or_bucket_folder>`.

Это означает, что данные PostgreSQL, InfluxDB и Image Store теперь будут храниться в корневом каталоге и больше не потребуется переносить их в директорию с новой версией LUNA PLATFORM при обновлении.

Скрипт Docker Compose также обновлен в соответствии с вышеописанной информацией.

**Примечание.** При обновлении на текущую версию необходимо перенести старые данные PostgreSQL, InfluxDB и Image Store в корневой каталог, а затем удалить и заново создать контейнеры, прописав новые пути директорий для монтирования. См. раздел «Перенос данных» в руководстве по обновлению.

- Добавлена поддержка работы сервисов LUNA PLATFORM без сервиса Image Store.

Сервис можно отключить в настройке «ADDITIONAL\_SERVICES\_USAGE» сервиса Configurator.

При использовании ресурсов, требующих наличия отключенного сервиса Image Store, будет возвращена ошибка 11070, Luna Image Store service is disabled.

При отключении сервиса Image Store, существуют некоторые особенности, перечисленные ниже:

- Объекты типа `images`, `objects`, `samples` и политики сохранения биометрических образцов в обработчиках/верификаторах будут недоступны.
- Пропадет возможность использования всех задач, кроме задач Garbage Collection, Linker и Estimator, однако для этих задач есть ряд ограничений:
  - \* Garbage Collection, Estimator, Linker: результаты задач/подзадач не будут сохраняться
  - \* Garbage Collection, Estimator, Linker: после завершения подзадачи, статус задачи будет обновлен на Done и идентификатор результата задачи будет равен None
  - \* Garbage Collection: удаление биометрических образцов станет недоступным

Если сервис Image Store отключается после того, как были сгенерированы события с включенной политикой «`image_origin_policy`», то при использовании задачи Garbage Collection и параметра `remove_image_origins`, сервис Tasks будет по-прежнему пытаться удалять исходные изображения, имеющие внешний URL-адрес.

При отключении сервиса Image Store, для сервиса Backport 3 БО и портреты станут недоступны, как и ресурсы «`get portrait`» и «`get portrait thumbnail`».

В сервис Backport 3 добавлена новая настройка «`BACKPORT3_ENABLE_PORTRAITS`», позволяющая отключить возможность использования портретов, но оставить возможность использования остального функционала сервиса Image Store. Если использование сервиса Image Store отключено в настройке «`ADDITIONAL_SERVICES_USAGE`», то вышеописанная настройка также должна быть отключена.

См. подробную информацию в разделе «Отключаемые сервисы» руководства администратора.

- Библиотека HASP обновлена до версии 1.7.3.
- Добавлен новый способ активации лицензии с помощью ключа Guardant.

Для активации данным способом требуется наличие сети Интернет и доступ к графическому интерфейсу системы. Если сервер, где запускается LUNA PLATFORM, не подходит под данные требования, то необходимо использовать вспомогательный сервер. В качестве вспомогательного сервера может выступать как сервер с ОС Linux, так и сервер с ОС Windows.

Старый способ активации с помощью ключа HASP также остается доступным.

См. подробную информацию в разделе «Активация лицензии с помощью Guardant-ключа» во всех руководствах по установке и обновлению.

- В запрос [«generate events»](#) добавлена поддержка указания пользовательских метаданных для исходного изображения.

Метаданные передаются с помощью заголовков вида «X-Luna-Meta-<user\_defined\_key>:<user\_defined\_value>», которые отправляются в сервис Image Store при сохранении исходного изображения во время генерации события.

Заголовки необходимо указывать в секции `image_origin` при использовании типа содержимого запроса `multipart/form-data`.

См. подробную информацию об использовании пользовательских метаданных при сохранении изображений в сервис Image Store в разделе «Сохранение исходных изображений».

### Исправленные ошибки

- Исправлена ошибка, при которой при указании в теле запроса [«generate events»](#) исходного изображения не в виде URL, оно сохранялось в сервис Image Store независимо от состояния политики `image_origin_policy`.
- Исправлено описание некорректных заголовков ответа «Content-Type» для запросов [«detect faces»](#) и [«get list count»](#).
- Практически во все запросы сервиса Admin добавлен новый заголовок `Exclude-Header`.

Благодаря этому исправлена ошибка в пользовательском интерфейсе Admin, из-за которой после перезагрузки страницы происходил выход из учетной записи.

- Исправлена ошибка лицензии, возникавшая при использовании пробной лицензии на указанное количество дней.

## LUNA PLATFORM v.5.38.3

### Изменения

- SDK обновлен до версии 5.13.0. Основные изменения SDK, затрагивающие LUNA PLATFORM:
  - обновлен детектор тел
  - обновлен эстиматор FishEye

Значение по умолчанию настройки `score_threshold` в секции «LUNA\_HANDLERS\_BODY\_DETECTOR\_SETTINGS» настроек сервиса Handlers изменено с **0.3** до **0.5**. Миграция настроек автоматически обновит это значение (см. раздел «Миграция базы данных Configurator» в руководстве по обновлению). Проверьте логику распознавания тел, если вы используете значение `score_threshold`, отличное от значения по умолчанию.

Значение по умолчанию настройки `redetect_face_target_size` в секции «LUNA\_HANDLERS\_FACE\_DETECTOR\_SETTINGS» настроек сервиса Handlers изменено с **45** до **64**. Миграция настроек автоматически обновит это значение (см. раздел «Миграция базы данных Configurator» в руководстве по обновлению). Проверьте логику распознавания лиц, если вы используете значение `redetect_face_target_size`, отличное от значения по умолчанию.

- Теперь при первоначальной проверке лицензии проверяется только время окончания действия лицензии, вместо всех проверок лицензии. Остальные проверки теперь выполняются при получении запросов сервисом Licenses.

Это изменение ускоряет загрузку сервиса Licenses.

### Исправленные ошибки

- В спецификацию OpenAPI добавлено отсутствующее описание для формата `msgpack` в запросы на сравнение «[matching faces](#)» и «[matching bodies](#)».
- Исправлена ошибка, при которой в мониторинге считывались не все форматы дат и времени. Если был передан непредусмотренный формат, то возникала ошибка чтения.

Теперь добавлена возможность чтения всех форматов дат из БД Influx.

- Исправлена ошибка сервиса Accounts, при которой некоторые ошибки, связанные с обработкой запросов к базе данных Accounts, не обрабатывались должным образом и не выводились в логи сервиса.
- Исправлена ошибка, из-за которой могли отсутствовать некоторые данные в результате выполнения задач, когда для параметра `tasks_to_faces_requests_concurrency` сервиса Tasks было установлено значение 1.
- Исправлена маршрутизация запросов к плагину сравнения.

Если запрос на сравнение содержит значение поля `target`, которое плагин не поддерживает, то сервис добавляет к запросу ключевое значение для поля `target` (`face_id` для сравнения по лицам и `event_id` для сравнения по событиям).

Ранее плагин не поддерживал поле `face_id` или `event_id` в качестве значения поля `target` и запрос мог завершиться ошибкой или вернуть пустой список кандидатов. Теперь сервис Python Matcher Proxy не будет перенаправлять такие запросы к плагину сравнения.

См. подробную информацию в разделе «Поля `target`, выступающие в качестве сравнения» руководства администратора.

## LUNA PLATFORM v.5.38.1

### Изменения

- Добавлена возможность передачи исходных изображений в формате URL или Base64 в запросе [«generate events»](#).

Исходные изображения можно передать с помощью указания типа содержимого запроса `Content-Type = application/json` или `Content-Type = multipart/form-data`.

- В комплект поставки LUNA PLATFORM добавлен скрипт Docker Compose `start_logging.sh`, запускающий сервис LUNA Dashboards, Grafana Loki и Promtail, позволяющие гибко работать с логами LUNA PLATFORM в Grafana.

Скрипт расположен в директории `example-docker`.

В руководство по установке LUNA PLATFORM с помощью Docker Compose добавлена соответствующая информация в раздел запуска основного скрипта Docker Compose.

См. подробную информацию о визуализации логирования в разделе «Grafana Loki» руководства администратора.

- Ускорена отправка данных в InfluxDB.
- Добавлена возможность фильтрации по значениям `null` (значение означает, что эстимация по атрибуту не выполнялась) для перечисленных ниже атрибутов в запросах [«get events»](#) и [«get statistics on events»](#):

- `meta`
- `source`
- `emotion`
- `mask`
- `ethnic_group`
- `liveness`
- `gender`

- `apparent_gender`
- `headwear_state`
- `sleeve_length`
- `upper_clothing_colors`
- `lower_garment_type`
- `lower_garment_colors`
- `shoes_apparent_color`
- `backpack_state`
- `city`
- `district`
- `street`
- `house_number`
- `area`
- `geo_position` (поля `origin_longitude` и `origin_latitude` для запроса «[get events](#)»)
- `track_id`

Это позволяет фильтровать события, сгенерированные по разным обработчикам с разными политиками, где в первом выполнялась эстимация определенного атрибута (например, состояния маски равно `occluded`), а во втором не выполнялась эстимация (например, состояние маски равно `null`), но нужно получить оба события.

- В настройки каждого сервиса добавлена группа настроек вида `<service_name>_HTTP_SETTINGS`, содержащая настройки, отвечающие за обработку HTTP-подключений.

Доступны следующие настройки:

- `request_timeout` — продолжительность времени между моментом, когда новое открытое TCP-соединение передается на сервер. Значение (в секундах) — целое число, по умолчанию 60.
- `response_timeout` — продолжительность времени между моментом, когда сервер передает HTTP-запрос приложению, и моментом, когда HTTP-ответ отправляется клиенту. Значение (в секундах) — целое число, по умолчанию 600.
- `request_max_size` — насколько большим может быть запрос. Значение (в байтах) — целое число, по умолчанию 1 Гб для всех сервисов кроме Image Store и ~130 Гб для сервиса Image Store.
- `keep_alive_timeout` — тайм-аут поддержания активности HTTP. Значение (в секундах) — целое число, по умолчанию 15.

См. подробную информацию по следующей ссылке: <https://sanic.dev/en/guide/deployment/configuration.html#builtin-values>

- Улучшена производительность запроса «[get faces](#)» с фильтрами по `list_id` и `face_id__gte/lt`.

- Добавлена возможность переопределить настройки сервисов при их старте с помощью переменных окружения.

Для переопределения настроек используется префикс `VL_SETTINGS`. Примеры:

- `VL_SETTINGS.INFLUX_MONITORING.SEND_DATA_FOR_MONITORING=0`. Использование переменной окружения из данного примера установит значение настройки `SEND_DATA_FOR_MONITORING` для секции `INFLUX_MONITORING` равным 0.
- `VL_SETTINGS.OTHER.STORAGE_TIME=LOCAL`. Для несоставных настроек (настроек, которые расположены в секции `OTHER` в конфигурационном файле) необходимо указать префикс `OTHER`. Использование переменной окружения из данного примера установит значение настройки `STORAGE_TIME` (если сервис использует данную настройку) на значение `LOCAL`.

Переменная окружения может быть указана с помощью аргумента `ENV` при запуске сервисов в Docker-контейнерах.

- В запрос [«get license»](#) сервиса `Licenses` добавлена возможность получить значение определенной функции лицензирования с помощью нового поля `targets`.
- Добавлена поддержка дашбордов для `LIM`.

### Исправленные ошибки

- В примерах ответов запросов [«get events»](#), [«get event»](#), [«get task result»](#) и [«ws handshake»](#) в спецификации OpenAPI исправлено описание полей с `body_basic_attributes` на `basic_attributes`, `cupper_body_attributes` на `upper_body`, `clower_body_attributes` на `lower_body` и с `body_accessories` на `accessories`.
- Исправлена ошибка, из-за которой не освобождалась память после выполнения задачи [«Exporter»](#).
- Исправлена ошибка, при которой задача [«Exporter»](#) продолжала выполняться с неполными данными даже если при её выполнении возникала ошибка.

Теперь при возникновении ошибки при выполнении задачи `Exporter` будут возвращаться ошибки с кодами 28038 (не удалось загрузить данные) и 28039 (не удалось загрузить данные, но были повторные попытки переподключения к сервисам `Faces` или `Events`).

- Исправлена ошибка, из-за которой прерывалась загрузка большого результата задачи.

## LUNA PLATFORM v.5.36.5

### Изменения

- Теперь адрес сервера лицензирования задается в настройках сервиса `Licenses`, а не в файле `hasp_30147.ini`.

Таким образом, больше нет необходимости монтировать данный файл при запуске контейнера Licenses.

В руководства по установке и обновлению добавлены инструкции по указанию адреса сервера лицензирования с помощью пользовательского интерфейса Configurator перед запуском сервиса Licenses.

В руководство по установке также добавлена инструкция по указанию адреса сервера лицензирования с помощью файла «platform\_settings.json» перед началом процесса установки.

Примечание. При обновлении с предыдущих версий нужно обязательно задать адрес лицензирования в новой настройке, иначе сервис Licenses не запустится.

- Обновлен внешний вид пользовательского интерфейса сервиса Admin.

Логика работы с пользовательским интерфейсом осталась прежней.

- Добавлена возможность записывать в генерируемое событие координаты ограничивающих прямоугольников лица или тела при использовании биометрического образца в качестве источника изображения. Координаты можно передать либо с помощью внешнего приложения, либо задать вручную в запросе на генерацию события (например, с помощью схемы «multipart/form-data»).

Указанные координаты сохраняются в поле `face_detections/body_detections > detection > rect` события.

Задание координат ограничивающих прямоугольников будет игнорироваться в других запросах, где можно указать биометрический образец в качестве источника изображения.

Ранее в событие можно было сохранять только ограничивающие прямоугольники лица или тела исходных изображений.

- В политику «match\_policy» запросов «create handler» и «validate handler policies» добавлена поддержка фильтрации кандидатов по полю meta события.
- В задачи «Clustering», «Exporter», «Cross-matching» и «Linker» добавлены фильтры по полю meta события.
- Улучшена валидация фильтра по полю meta для сравнения. Данный фильтр используется в следующих запросах:
  - «matching faces»,
  - «matching bodies»,
  - «cross-matching task».
- В запрос «get statistics on events» добавлена поддержка фильтрации и агрегирования по полю meta события.



- В сервис API добавлен новый запрос [«get platform features»](#), в ответе на который можно получить информацию активна и не истек ли срок действия лицензии, а также информацию о включенных функциях лицензии («face\_quality», «body\_attributes» и «liveness») и доступности функционала работы с сервисами Events, Tasks и Sender. Использование данных сервисов включается в настройке «ADDITIONAL\_SERVICES\_USAGE» сервиса Configurator.
- В задачу [«Garbage collection»](#) добавлена возможность выбрать лица в качестве объекта для очистки (поле target).

Также доступны фильтры create\_time\_\_lt, create\_time\_\_gte, user\_data, list\_id, а также параметры сохранения результатов (поле store\_results) и удаления биометрических образцов лиц (поле remove\_samples).

Выбор лиц в качестве объекта для очистки также доступен в пользовательском интерфейсе сервиса Admin.

- Обновлена логика работы эстиматора проверки динамического диапазона (dynamic\_range) в группе проверок «face\_quality».
- Ресурс /attributes/batches сервиса Faces перенесен в /descriptors/batches.
- Во всех сервисах LUNA PLATFORM отключена запись логов в файл по умолчанию (настройка log\_to\_file каждого сервиса).
- Обновлена документация в комплекте поставки.

Переработан раздел «Общие положения» руководства администратора.

Переработаны разделы «Подготовка к установке/обновлению» всех руководств по установке/обновлению/миграции. Часть информации описательного характера перенесена в раздел «Дополнительная информация», расположенный в конце каждого руководства.

## Исправленные ошибки

- В спецификации OpenAPI сервиса Admin было исправлено перечисление в поле content > filter > object\_type с «face»/«event» на «faces»/«events» запроса [«create additional extract task»](#).
- Исправлена ошибка, выводящая в логи лишнюю информацию при использовании команды включения подсчета статистики create\_usage\_task.
- Исправлена ошибка в спецификации OpenAPI сервиса Python Matcher, из-за которой фильтр meta не имел тип nullable в запросах [«face matching»](#), [«human body matching»](#), [«cross matching faces»](#) и [«cross matching bodies»](#).
- Исправлена ошибка, из-за которой в запросе [«face matching»](#) сервиса Python Matcher объект candidates > filters не был обязательным.

## LUNA PLATFORM v.5.35.0

### Изменения

- Прекращена поддержка старых сервисов Индексирования и поиска по индексу.

Упоминание данных сервисов удалено из документации и настроек сервисов.

Теперь для построения индекса и поиска по индексу используется только модуль LUNA Index Module.

- Добавлена функциональность для сохранения и получения пользовательских метаданных изображения через настраиваемые заголовки.

Сохранить изображение с пользовательскими метаданными можно в ресурсе [«create images»](#), задав в заголовок «X-Luna-Meta-`<user_defined_key>`» с значением «`<user_defined_value>`». В бакете исходных изображений в хранилище Image Store, метаданные сохраняются в отдельный файл «`<image_id>.meta.json`», который расположен рядом с исходным изображением.

В ответе на запрос на получение изображения (запрос [«get image»](#)) нужно указать заголовок `with_meta=1` для получения метаданных изображения в заголовке ответа.

Чтобы сохранять значения метаданных для нескольких ключей, необходимо задать несколько заголовков.

- Версия LUNA Dashboards обновлена до 0.0.5. Версия Grafana в контейнере LUNA Dashboards обновлена до 8.5.20.

Теперь не нужно использовать отдельную команду для запуска дашбордов, они создаются автоматически при запуске контейнера LUNA Dashboards.

- Добавлены контейнеры Grafana Loki (система агрегации логов) и Promtail (агент, доставляющий логи LUNA PLATFORM в Grafana Loki).

Для запуска контейнеров требуется наличие запущенной Grafana и настроенного источника данных «Loki». В контейнере LUNA Dashboards уже создан источник данных «Loki».

В раздел [«Grafana Loki»](#) руководства по установке добавлена инструкция по запуску Grafana Loki и Promtail.

См. дополнительную информацию в разделе [«Grafana Loki»](#) руководства администратора.

- В задачу Estimator ([песчпс «/tasks/estimator»](#)) добавлена поддержка сетевой файловой системы Samba в качестве источника изображений (параметр «`source_type`»).

Для такого типа источника в теле запроса доступно задание следующих параметров для соединения с Samba:

- Параметр «`host`» — IP-адрес (обязательное поле);

- Параметр «port» — порт;
- Параметры «user» и «password» — параметры авторизации. Если нет данных авторизации, подключение к Samba будет осуществляться как гостевое.

Как и в задачах Estimator с использованием FTP, S3-подобного хранилища или сетевого диска в качестве источников изображений, доступна возможность задания пути до директории с изображениями, рекурсивного получения изображений из вложенных директорий, выбора типа передаваемых изображений, а также указания префикса и постфикса.

Для получения корректных результатов обработки изображений с помощью задачи Estimator все обрабатываемые изображения должны быть либо в исходном формате, либо в формате биометрических образцов.

См. соответствующие примеры и дополнительную информацию в [спецификации OpenAPI](#).

- В ресурс [«sdk»](#) добавлен параметр «extract\_exif», позволяющий извлекать EXIF данные из изображения.
- В ресурс [«ws handshake»](#) добавлены фильтры «geo\_position» (параметры «origin\_longitude», «origin\_latitude», «longitude\_delta» и «latitude\_delta») и «user\_data».
- В ресурсы [«matching faces»](#), [«matching bodies»](#) и политику «match\_policy» запроса [«create handler»](#) добавлена поддержка фильтрации кандидатов по полю «meta» события.
- В раздел «Информация по обновлению» руководства по обновлению добавлена таблица с ключевыми изменениями предыдущих версий LUNA PLATFORM, которые влияют на процесс установки и работы LUNA PLATFORM. Данная таблица может быть полезна при попытке обновления LUNA PLATFORM не с предыдущей версии.

## LUNA PLATFORM v.5.34.0

### Изменения

- SDK обновлен до версии 5.12.0. Основные изменения SDK, затрагивающие LUNA PLATFORM:
  - обновлены следующие эстиматоры: LivenessOneShotRGB, FishEye, Orientation, HeadWear;
  - удалены 54-ая, 56-ая и 57-ая модели нейронных сетей извлечения биометрических шаблонов лиц и 104-ая и 106-ая модели нейронных сетей извлечения биометрических шаблонов тел. Поддержка данных моделей остается доступной. При необходимости можно запросить их у специалистов VisionLabs.

Теперь модель нейронной сети извлечения биометрических шаблонов тел по умолчанию — 107 (настройка «DEFAULT\_HUMAN\_DESCRIPTOR\_VERSION» сервиса Handlers).

### Важная информация по обновлению с предыдущих версий

Необходимо выполнить одно из следующих действий перед запуском контейнера Handlers:

- вручную изменить значение настройки «DEFAULT\_HUMAN\_DESCRIPTOR\_VERSION» с «104» или «106» на «107». После изменения версии нейронной сети извлечения биометрических шаблонов тел необходимо выполнить задачу «Additional extraction» после запуска сервиса Admin (см. раздел [«Launch re-extraction task»](#) в руководстве администратора). В противном случае, поиск и сравнение по старым биометрическим шаблонам будут недоступны.
- выключить использование нейронной сети для извлечения биометрического шаблона тела с помощью передачи аргумента `--enable-body-descriptor-estimator=0` при старте контейнера Handlers
- запросить у VisionLabs 104-ую модель нейронной сети и перенести её в контейнер Handlers по инструкции, описанной в разделе [«Use non-delivery neural network model»](#) руководства администратора

Если не выполнить одно из вышеописанных действий, то **запуск сервис Handlers завершится ошибкой**.

Вся вышеописанная информация добавлена в руководство по обновлению в раздел [«Changing the model of the neural network of bodies»](#).

При установке LUNA PLATFORM с нуля никаких дополнительных действий выполнять не требуется.

- В структуру события добавлено новое поле «meta», предназначенное для хранения произвольных пользовательских данных в JSON-формате (не более 2 Мб).

Предполагается, что с помощью данного функционала пользователь создаст свою модель данных (структуру события) и будет использовать её для хранения этих данных. Обратите внимание, что если планируется хранить несколько структур, то необходимо явно их разделять, чтобы избежать пересечения полей. Например, следующим образом:

```
{
  "struct1": {
    ...
  },
  "struct2": {
    ...
  }
}
```

Пользовательские данные можно передать тремя способами:

- в теле запроса [«generate events»](#) с типом содержимого `application/json` или `multipart/form-data`;

- в теле запроса [«save event»](#);
- с помощью пользовательского плагина или клиентского приложения.

В теле запроса [«generate events»](#) доступна возможность задать поле «meta» как для конкретных изображений, так и для всех изображений сразу (взаимная метаданная). Для запросов с включенной агрегацией для агрегированного события будет использоваться только взаимная метаданная, а метаданная для конкретных изображений будет игнорироваться. См. подробную информацию в теле запроса [«generate events»](#) в спецификации OpenAPI.

Поле «meta» может быть использовано как фильтр в запросе [«get events»](#) или в качестве значения для параметра «target» в запросе [«get event»](#).

Колонка «meta» добавлена в задачи [Reporter](#) и [Exporter](#).

Поддержка метаданных события также добавлена в ресурс [«ws handshake»](#).

При необходимости можно построить индекс для улучшения поиска.

См. подробное описание и особенности работы в разделе [«Events meta-information»](#) руководства администратора.

- Прекращена поддержка сервиса Liveness V1.

Сервис Liveness V1 удален из документации и настроек сервисов. Liveness V2 переименован в Liveness.

- В аргументы запуска контейнера Handlers добавлен новый аргумент `enable-all-estimators-by-default`, включающий/отключающий инициализацию всех эстиматоров и детекторов по умолчанию.

Ранее для использования определенных эстиматоров или детекторов приходилось указывать статус каждого эстиматора из всех существующих. Теперь достаточно отключить инициализацию всех эстиматоров `enable-all-estimators-by-default=0` по умолчанию, а затем указать только те эстиматоры или детекторы, которые необходимо включить.

Пример команды запуска сервиса Handlers с использованием только детектора лиц и эстиматоров биометрического образца лица и эмоций.

```
docker run \
...
--env=EXTEND_CMD="--enable-all-estimators-by-default=0 --enable-face-
    detector=1 --enable-face-warp-estimator=1 --enable-emotions-
    estimator=1" \
...
```

См. подробную информацию в разделе [«Enable/disable several estimators and detectors»](#) руководства администратора.

- Теперь фильтр по «account\_id» не является обязательным для задач [Clustering](#) (фильтр в теле запроса), [Cross-matching](#) (фильтра в теле запроса для кандидатов или эталона) и [Estimator](#) (фильтры сравнения в политиках обработчика). Это позволяет выполнять сравнение по объектам разных аккаунтов.
- Для нижеперечисленных ресурсов в тело ответа добавлен новый параметр «external\_url», указывающий абсолютный адрес до объекта:
  - «create account»,
  - «create token»,
  - «replace token»,
  - «create images»,
  - «create objects»,
  - «extract attributes»,
  - «create face»,
  - «create list»,
  - «save event»,
  - «create verifier»,
  - «create handler».

В качестве абсолютного адреса используется адрес сервиса API, указанный в настройке «EXTERNAL\_LUNA\_API\_ADDRESS» сервиса API. Значение настройки по умолчанию — <http://127.0.0.1:5000/6/>.

Данное изменение позволяет использовать ссылки из ответов сервиса API в своих целях, не зная точного адреса сервиса. Также изменение позволяет передавать ссылки в удобном формате, по которому можно сразу получить их содержимое.

Пример относительной ссылки (параметр «url»): «/6/objects/4a870804-0cd6-4c13-9c78-98ad167dc4ec»

Пример абсолютной ссылки (параметр «external\_url»): «<http://127.0.0.1:5000/6/objects/4a870804-0cd6-4c13-9c78-98ad167dc4ec>»

- В ресурс пакетной выгрузки биометрических шаблонов «get face descriptor batches» сервиса Faces добавлена поддержка заголовка «Accept», принимающего два значения — `application/x-flutbuf` (по умолчанию) и `application/x-msgpack`.
- Из настроек сервиса Handlers удален неиспользуемый параметр «max\_face\_size» из секции «LUNA\_HANDLERS\_FACE\_DETECTOR\_SETTINGS».

Параметр «max\_face\_size» рассчитывается как `min_face_size * 32`.

## Исправленные ошибки

- В спецификации OpenAPI сервиса API исправлены и дополнены примеры ответов с кодами 403, 408, 413, 500, 503 и 504. См. примеры ответов в [спецификации OpenAPI](#).

- В запрос [«perform verification»](#) добавлен отсутствующий ранее параметр указания типа обрабатываемого изображения «image\_type». Для данного параметра доступно два значения — 0 (исходное изображение) и 1 (биометрический образец лица).
  - Исправлены некорректные пороги для параметров «mouth\_occluded», «mouth\_smiling» и «mouth\_open» группы параметров «face\_quality» в запросах [«create handler»](#), [«get handlers»](#), [«get handler»](#) и [«replace handler»](#) в спецификациях OpenAPI сервисов API и Handlers.
- В спецификации было указано, что допустимые пороги для вышеперечисленных параметров равны [-1..1], хотя на самом деле допустимые пороги равны [0..1].

## LUNA PLATFORM v.5.33.0

### Изменения

- Начиная со следующего релиза прекращается поддержка Liveness V1. Из документации будут удалены команды запуска контейнера и описание.
- В ресурс [«sdk»](#) и политику «detect\_policy» > «body\_attributes» ресурса [«/handlers»](#) добавлен новый параметр «estimate\_lower\_body», позволяющий выполнить оценку следующих атрибутов тел:
  - «lower\_garment» — тип нижней одежды (брюки, шорты, юбка, неизвестно), цвет нижней одежды (бежевый, чёрный, синий, коричневый, зеленый, серый, хаки, разноцветный, оранжевый, розовый, фиолетовый, красный, белый, желтый, неизвестно);
  - «shoes» — цвет обуви (черный, белый, прочий, неизвестный).

Данная оценка по умолчанию выключена в обоих ресурсах.

Также теперь в параметре «estimate\_upper\_body» определяется цвет головного убора (параметр «headwear\_apparent\_color») (цвета аналогичны цветам нижней одежды).

Новые атрибуты тел добавлены в структуру событий (см. «detections» > «samples» > «body» > «detection» > «attributes»).

В политику «match\_policy» ресурса [«/handlers»](#) и в ресурс [«human body matching»](#) добавлены фильтры атрибутов тел «lower\_garment\_types» (тип нижней одежды), «lower\_garment\_colors» (цвет нижней одежды), «shoes\_apparent\_colors» (цвет обуви) и «headwear\_apparent\_color» (цвет головного убора), указываемые при использовании событий в качестве кандидатов. Также добавлена возможность задать данные атрибуты тела в качестве значений для поля «targets».

Фильтры данных атрибутов тел также могут быть использованы при отправке событий через веб-сокеты (см. ресурс [«ws handshake»](#)).

Соответствующие фильтры для событий добавлены в задачи Cross-matching, Clustering, Reporter, Exporter и Linker. В задачи Reporter и Exporter добавлены соответствующие колонки.

Поддержана возможность указания данных атрибутов тел при создании нового обработчика в задаче Estimator.

Структура события в запросах «generate events»/«save events» также расширена данными атрибутами тел. Кроме того, в данных запросах и запросе «sdk» доступна агрегация по данным атрибутам тел. Агрегированные значения атрибутов отображаются в полях «aggregate\_estimations» соответствующих ресурсов.

- Обновлено настройки сервиса Handlers.

Теперь можно выставлять как индивидуальные настройки среды выполнения оценок для каждого эстиматора/детектора, так и глобальные настройки для всех эстиматоров/детекторов.

Глобальные настройки задаются в секции «LUNA\_HANDLERS\_RUNTIME\_SETTINGS» и содержат в себе три настройки:

- global\_device\_class — тип устройства («cpu» или «gpu») для всех эстиматоров/детекторов, у которых значение параметра «device\_class» = «global» (см. ниже);
- num\_threads — количество потоков «рабочего процесса» для всех эстиматоров/детекторов;
- num\_compute\_streams — количество потоков для всех эстиматоров/детекторов.

Индивидуальные настройки задаются в секции вида "LUNA\_HANDLERS\_<estimator-or-detector-name>\_SETTINGS.runtime\_settings" и содержат в себе три настройки:

- device\_class — тип устройства для выполнения оценки («cpu», «gpu» или «global»);
- optimal\_batch\_size — размер батча для выполнения оценки;
- worker\_count — количество «рабочих процессов» для выполнения оценки.

- В сервисе Handlers добавлена возможность включения/выключения использования отдельных эстиматоров и детекторов. По умолчанию все эстиматоры включены.

Ранее можно было включать/выключать только все эстиматоры сразу.

Возможность включения отдельных эстиматоров позволяет экономить оперативную память или память GPU, поскольку при старте сервиса Handlers выполняется проверка возможности выполнения указанных оценок и загрузка данных в память. При отключении эстиматора или детектора можно также удалить его нейронную сеть из контейнера Handlers.

Отключение эстиматоров или детекторов возможно с помощью передачи аргументов с названиями эстиматоров или детекторов в команду запуска сервиса Handlers. Аргументы пе-



редаются в контейнер с помощью переменной «EXTEND\_CMD». Перечень всех аргументов и пример запуска контейнера добавлены в руководство администратора LUNA PLATFORM 5.

- Добавлена поддержка изображений со схемой CMYK.
- В сервис Backport 3 добавлен ресурс [«ws handshake»](#) для возможности получения событий через веб-сокеты.

Данный функционал аналогичен функционалу сервиса «Event & Statistic Service» LUNA PLATFORM 3.

- Обновлено настройки сервиса API.

Секция «LUNA\_HANDLERS\_LIVENESS\_SETTINGS» удалена из настроек сервиса API, т.к. она не использовалась.

Параметр «LIVENESS\_THRESHOLD» переименован в «LIVENESSV1\_THRESHOLD».

- В политику «detect\_policy» ресурсов [«create handler»](#) и [«create verifier»](#) добавлен параметр «estimate\_glasses», позволяющий выполнить оценку типа очков (очки, солнечные очки, нет очков).
- Параметр «estimate\_attributes» в запросах [«create descriptors»](#) и [«search»](#) сервиса Backport 3 расширен возможностью оценки типа очков (очки, солнечные очки, нет очков). Теперь при выполнении оценки атрибутов будут определяться пол, возраст и очки.

В ресурсе [«ws handshake»](#) доступны фильтры «glasses\_\_gt»(нижний порог) и «glasses\_\_lt»(верхний порог).

- Фильтр по «account\_id» добавлен в параметры запросов большинства ресурсов с методами GET, которые возвращают в ответе несколько объектов.

Использование данного фильтра позволяет получать данные (токены, лица, атрибуты и др.), принадлежащие конкретному аккаунту.

Исключениями являются ресурс «get accounts», сервисные ресурсы («get health», «get configs» и т. д.) и ресурс «ws handshake».

Также параметр запроса «account\_id» был добавлен в ресурсы [«face matching»](#) и [«human body matching»](#).

Данные параметры запроса будут работать только в том случае, если для параметра «visibility\_area» токена установлено значение «all». В противном случае будет возвращена ошибка.

- Добавлена возможность использования пользовательской SQL-функции для сравнения.

Данная функция может быть полезна если нет возможности использовать C-расширения (C-extensions) базы данных PostgreSQL. Функция может понадобиться, например, для разворачивания LUNA PLATFORM 5 в AWS и использования Amazon Aurora PostgreSQL.

См. подробное описание в главе «Alternative matching options» в файле комплекта поставки «luna\_v.5.33.0/extras/VLMatch/postgres/readme.md».

- В контейнерах сервиса API и Licenses обновлена версия Python до 3.10.
- Обновлена база данных Redis до версии 7.0.5-alpine3.16.

### Исправленные ошибки

- Исправлена ошибка, из-за которой в запросе на получение статистики по событиям (ресурс «[/events/statistic](#)») всегда передавался фильтр с текущим «account\_id», из-за чего статистика выдавалась только по данным аккаунта пользователя, который выполнял запрос.

Это приводило к тому, что пользователи с типом аккаунта «admin» и «advanced\_used» не могли получать статистику по другим пользователям.

- Исправлена ошибка, из-за которой в запросе «[get handlers](#)» невозможно было получить одновременно и динамический и статический обработчики с помощью фильтра «is\_dynamic». Теперь если не указывать данный фильтр, то будут возвращены обработчики обоих типов.
- Исправлена ошибка в запросе «[detect face](#)» спецификации API, где в теле ответа на запрос отображалось несуществующее поле «liveness».
- Исправлена ошибка, из-за которой при выполнении запроса к ресурсу «[/3/attributes/{attribute\\_id}/samples](#)» сервиса Faces не учитывался фильтр по «account\_id».
- Исправлена ошибка, при которой в ответе на запрос «clear authorization» сервиса Admin возвращалась ошибка «Internal server error» при отсутствии куки.

Теперь возвращается ошибка «Cookies for current session not found».

- Исправлено появление ошибки «Internal server error», возникавшее при загрузке изображения с цветовой схемой, которую не поддерживает SDK.

Теперь если будет совершена попытка загрузить изображение с неподдерживаемой цветовой схемой, то будет возвращена корректная ошибка под номером [18003](#).

- Исправлена ошибка в сервисе Python Matcher с включенным кешем, при которой могли возникать непредвиденные ошибки в логах после изменения настроек в сервисе Configurator.

Теперь при остановке сервиса обеспечивается закрытие открытых соединений и не появляются ошибки в логах.

- Исправлена ошибка фильтра по «account\_id» в сервисе Python Matcher.

Если в запросах «face matching» и «human body matching» с указанным фильтром «account\_id» значение поля «account\_id» события-эталона отличалось от значения поля «account\_id» кандидатов, то выдавалось некорректное сообщение, что отфильтрован только кандидат. Теперь выдается корректное сообщение, что указанный эталон не найден.

- Исправлена ошибка в сервисе Python Matcher, которая возникала в запросах «face matching» и «human body matching» с Basic-авторизацией и пользователем типа «user» при указании идентификатора аккаунта в поле «candidates» > «filter» > «account\_id», отличающегося от идентификатора по которому выполняется запрос.

Теперь при попытке использовать чужой «account\_id» в фильтрах при типе аккаунта «user», будет выдано корректное сообщение.

- Исправлена ошибка в запросе «upgrade attribute» сервиса Handlers, когда при ошибке обработки изображения (например, если вместо БО лица указать БО тела) возвращалось пустое поле, либо ошибка «Internal server error». Теперь при ошибках обработки изображения будет возвращаться корректная ошибка [11043](#).

## LUNA PLATFORM v.5.32.0

### Изменения

- Добавлена возможность выполнять сравнение по биометрическим шаблонам тел при генерации события.

Политика «match\_policy» обработчика расширена новым фильтром «descriptor» > «descriptor\_type», позволяющим явно указать по какому типу биометрического шаблона будет происходить сравнение — по БШ лица (значение «face») или по БШ тела (значение «body»).

Если выполняется сравнение по лицам, но тип биометрического шаблона указан как «body», то будет выдана ошибка.

БШ лиц привязываются к объекту «face», а БШ тел привязываются к объекту «event». Если пользователем не был написан специальный плагин сравнения, то сравнение по событиям может занимать длительное время.

Обратите внимание, что сравнение по телам не такое точное, как сравнение по лицам. При большом количестве событий-кандидатов, вероятность ложных определений лучших совпадений выше, чем при большом количестве лиц-кандидатов.

- Контейнеры сервисов теперь именуются одинаково при ручном запуске и при запуске с помощью Docker Compose.

Например, раньше контейнер для сервиса Configurator создавался под именем example-docker-configurator\_1, а теперь будет создаваться под именем luna-configurator.

- В контейнерах сервисов Faces, Image Store, Accounts, Tasks, Events, Configurator, Sender, Handlers, Backport3 и Backport4 обновлена версия Python до 3.10.

В документации обновлены все команды, связанные с использованием Python внутри контейнеров, а именно, команды вида python3.9 заменены на python3.

- В ресурс [«healthcheck»](#) сервиса Faces добавлен параметр `«include_luna_services»`, с помощью которого можно включать и отключать проверку healthcheck для сервисов LUNA PLATFORM, от которых зависит данный сервис. Если опция включена, то отправляются дополнительные запросы на ресурсы `«/healthcheck»` этих сервисов.
- Добавлена поддержка нейросети 60 в сервисы Построения индекса и поиска по индексу.

### Исправленные ошибки

- В тела ответов ресурсов [«get tasks»](#), [«get task»](#) и [«get task result»](#) спецификации OpenAPI добавлена отсутствующая информация о задаче `«Additional extract»`.

Как и раньше, задача `«Additional extract»` должна выполняться с помощью сервиса Admin.

- Параметр `«tasks» > «content» > «filters» > «account_id»` в телах ответа ресурсов [«get tasks»](#), [«get task»](#) для задачи `«Garbage collection»` теперь не является обязательным, поскольку `«account_id»` может отсутствовать если задача была запущена из сервиса Admin.
- Параметр `«tasks» > «content» > «target»` в телах ответа ресурсов [«get tasks»](#), [«get task»](#) для задачи `«Garbage collection»` расширен параметрами `«event_descriptors»` и `«face_descriptors»` для случаев удаления биометрических шаблонов лиц и тел по версии и типу.
- Из тел ответа ресурсов [«get tasks»](#), [«get task»](#) удален параметр `«tasks» > «content» > «options» > descriptor_version»` для типа задачи `«Additional extract»` для случая, когда в качестве извлечения использовались базовые атрибуты (`«extraction_target» = «basic_attributes»`).
- В ресурсы [«create account»](#), [«patch account»](#), [«create token»](#), [«replace token»](#) и [«verify credentials»](#) спецификации OpenAPI добавлены отсутствующие заголовки `«Content-Type»`.
- Исправлена ошибка, при которой сервис Image Store не запускался и не выдавал никаких логов если не указывалась схема URL-адреса хранилища S3. Теперь при подобной ошибке будет выданы соответствующие логи.

### LUNA Index Module

Начиная с этой сборки для LUNA PLATFORM 5 доступен новый модуль LUNA Index Module (LIM), значительно ускоряющий сравнение большого количества биометрических шаблонов лиц. Модуль предварительно строит индексы по набору списков лиц и выполняет сравнение по ним. Пользователь либо сам указывает списки для обработки, либо выставляет автоматическую обработку всех существующих в LP списков.

LIM повторяет основной функционал ранее существовавших Сервисов формирования индекса и поиска по индексу, при этом его производительность выше. В отличие от старых сервисов, в LIM не требуется подключение через SSH за счет переработки механизма доставки индексов. На данный момент в LIM отсутствует функционал построения индексов, доступный в существовавших ранее Сервисах формирования индекса и поиска по индексу, т.е. если лицо прикрепилось к списку уже после того, как индекс начал строиться, то оно не появится в результатах сравнения. Данный

функционал находится в работе и будет доступен в ближайших релизах. Если данный функционал является критичным, то рекомендуется дождаться его выхода в ближайших релизах.

LIM поставляется в качестве отдельного дистрибутива, содержащего руководство администратора с информацией о новых сервисах и их работе, инструкции по установке (ручной и с помощью Docker Compose), спецификации OpenAPI и скрипты запуска. Более подробную информацию о комплекте поставки можно найти в документе «LIM\_Quick\_Start\_Guide.pdf» из комплекта поставки LIM.

Для использования LIM требуется отдельный параметр в лицензионном ключе LUNA PLATFORM 5. Необходимо связаться с VisionLabs возможности добавления нового параметра для работы с LIM в существующий лицензионный ключ.

LIM работает со всеми версиями нейронных сетей извлечения биометрических шаблонов лиц.

Предыдущие сервисы Построения индекса и поиска по индексу остаются доступными до 2023 года, после чего будет прекращена поддержка и сервисы будут убраны из поставки.

Для дополнительной информации см. руководство администратора LIM в соответствующем комплекте поставки.

## LUNA PLATFORM v.5.31.0

### Изменения

- SDK обновлен до версии 5.10.0.
- Добавлена поддержка 60 нейронной сети извлечения биометрических шаблонов лиц и 105, 106, 107 нейронных сетей извлечения биометрических шаблонов тел. По умолчанию используются версии 59 и 104.

Нейросеть 60 не поддерживается при использовании сервисов для построения индекса и поиска по индексу.

В целях уменьшения размера контейнера Handlers в ближайших сборках LP 5 из поставки будут удалены нейросети извлечения биометрических шаблонов лиц 54, 56, 57 и нейросети извлечения биометрических шаблонов тел 102, 103 и 104. Для их использования потребуется отдельно скачать необходимую нейросеть и поместить в контейнер сервиса Handlers. Данная процедура описана в разделе «Использование модели нейросети не из поставки» руководства администратора LP 5.

- Добавлена поддержка детектора HumanFace для одновременной детекции лица и тела в кадре.

Детектор используется автоматически при выполнении запросов POST на ресурсы «/handlers/{handler\_id}/events» и «/tasks/estimator», если в обработчике указаны обе опции

«detect\_face» и «detect\_body». При этом значительно повышается скорость обработки изображений по сравнению с использованием отдельных детекторов.

В связи с этим в сервисе Handlers обновлены названия полей в БД Influx, используемых при мониторинге:

- detection\_width -> face\_detection\_width (серия face\_detection);
  - detection\_height -> face\_detection\_height (серия face\_detection);
  - detection\_width -> body\_detection\_width (серия body\_detection);
  - detection\_height -> body\_detection\_height (серия body\_detection).
- Сервис Accounts добавлен в инструмент визуализации данных мониторинга LUNA Dashboards.

### Исправленные ошибки

- Исправлена ошибка, из-за которой сервис Handlers в некоторых случаях не возвращал соединение в пул соединений БД Postgres и мог выводить ошибку с кодом 10017 (Database connection timeout error) пока сервис не был перезапущен.
- Из скрипта миграции БО «migrate\_4\_to\_5.py» удалена зависимость «logbook», отсутствие которой приводило к ошибке «ModuleNotFoundError: No module named 'logbook'».
- Исправлено отсутствие описания поля «remove\_image\_origins» в запросе «create gc task» в OpenAPI документации сервиса Admin.
- Исправлена ошибка, при которой в запросе [на создание аккаунта](#) к сервису Backport4 в теле ответа возвращалась неправильная версия API.

## LUNA PLATFORM v.5.30.0

### Изменения

- Добавлен новый механизм для реализации ролевой модели в LP 5. Он предоставляет возможность создать аккаунты с определённой зоной видимости данных и выпускать для них токены с разграничением прав на доступ к ресурсам LP в рамках аккаунта, для которого создан токен.

Теперь для большинства запросов требуется обязательное наличие аккаунта, кроме запросов, не предполагающих авторизации.

Все аккаунты, созданные с помощью сервиса Admin будут автоматически мигрированы. Для сохранения возможности работы с ранее созданными с помощью заголовка «Luna-Account-Id» данными нужно создать новый аккаунт, указав при создании использованный ранее «account\_id». Для сервиса Backport 3 необходимо отдельно запустить скрипт миграции.

См. подробное описание о новой системе авторизации и о миграции аккаунтов ниже или в разделе «Аккаунты, токены и способы авторизации» руководства администратора.

## Аккаунты

Аккаунт необходим для разграничения областей видимости объектов для конкретного пользователя. Каждый созданный аккаунт имеет свой собственный уникальный идентификатор «account\_id». Все данные аккаунтов сохраняются в БД сервиса Accounts под этим идентификатором.

Аккаунт можно создать с помощью POST запроса «[create account](#)» к сервису API, либо с помощью запроса «[register account](#)» к сервису Admin, либо с помощью пользовательского интерфейса Admin. При создании аккаунта необходимо указать следующие данные: login (email), password и account type (тип аккаунта).

Тип аккаунта определяет, какие данные доступны пользователю. Существует три типа аккаунтов:

- user — тип аккаунта, с помощью которого можно создавать объекты и использовать только данные своего аккаунта.
- advanced\_user — тип аккаунта, для которого доступны права, аналогичные «user», а также есть доступ к данным всех аккаунтов. Доступ к данным других аккаунтов означает возможность получать данные (запросы GET), проверять их наличие (запросы HEAD) и выполнять запросы на сравнение по данным других аккаунтов.
- admin — тип аккаунта, для которого доступны права, аналогичные «advanced\_user», а также есть доступ к сервису Admin (см. «Сервис Admin» ниже)

С помощью заголовка «Luna-Account-Id» в запросе «[create account](#)» можно задать желаемый идентификатор аккаунта. Также его необходимо использовать в случае, если необходимо сохранить возможность работы с данными, созданными в версиях LP до 5.30.0 (см. «Миграция аккаунтов с предыдущих сборок LP 5» ниже).

## Токены

Токен привязывается к существующему аккаунту любого типа и позволяет наложить расширенные ограничения на выполняемые запросы. Например, при создании токена можно дать пользователю разрешение только на создание и изменение всех списков и лиц, или можно запретить использование определенных обработчиков, указав их идентификатор.

Токен и все его разрешения сохраняются в БД и привязываются к аккаунту по параметру «account\_id».

При создании токена доступно задание следующих параметров:

- expiration\_time – время окончания действия токена в формате RFC 3339. Можно указать бесконечное время действия токена с помощью значения «null»
- permissions – разрешения, которые доступны пользователю

- `visibility_area` – видимость токеном данных других аккаунтов (см. раздел «Просмотр данных других аккаунтов» в руководстве администратора)

См. список всех разрешений и подробную информацию по токенам в разделе «Разрешения, задаваемые в токене» руководства администратора.

### Типы авторизации для доступа к ресурсам

В LUNA PLATFORM доступно три типа авторизации:

- **BasicAuth**. Авторизация по логину и паролю (задаются во время создания аккаунта).
- **BearerAuth**. Авторизация по JWT токenu (выдается после создания токена).
- **LunaAccountIdAuth**. Авторизация по заголовку «Luna-Account-Id», в котором указывается сгенерированный после создания аккаунта «`account_id`» (данный способ был принят за основной до версии 5.30.0).

Авторизация **LunaAccountIdAuth** имеет наименьший приоритет по сравнению с другими способами и может быть отключена с помощью настройки «`ALLOW_LUNA_ACCOUNT_AUTH_HEADER`» в секции «OTHER» настроек сервиса API в Configurator (по умолчанию включена). В спецификации [OpenAPI](#) заголовок «Luna-Account-Id» помечен словом **Deprecated**.

### Сервис Admin

Аккаунтом по-прежнему можно управлять с помощью сервиса Admin. Пользовательский интерфейс сервиса был обновлен. Теперь диалог создания аккаунта содержит расширенный набор полей.

Старый аккаунт администратора преобразован в аккаунт с типом «admin». Логин по умолчанию теперь **root@visionlabs.ai** вместо root.

При установке с нуля для сервиса Admin больше не создается БД. Теперь данные аккаунтов хранятся в БД Accounts (см. ниже «Миграция БД Admin»).

### Миграция аккаунтов с предыдущих сборок LP 5

Ранее аккаунт создавался с помощью ресурса «/accounts» сервиса Admin или пользовательского интерфейса сервиса Admin.

Все аккаунты, созданные до текущей версии будут автоматически мигрированы при обновлении на актуальную версию. Аккаунту администратора будет присвоен тип «admin», а аккаунтам, создаваемым при запросе к ресурсу «/accounts» будет присвоен тип «advanced\_user». В качестве логина и пароля будет использован почтовый адрес. Имя организации будет записано в поле «description».

Также ранее можно было выполнять запросы к сервисам LP 5 с помощью указания идентификатора «`account_id`» в заголовке «Luna-Account-Id». Для того, чтобы сохранить возможность работы с данными, созданными ранее с помощью указания идентификатора «`account_id`» в



заголовке «Luna-Account-Id», необходимо в запросе создания аккаунта [«create account»](#) указать «login», «password», «account\_type» и старый идентификатор «account\_id» в заголовке «Luna-Account-Id» запроса. Таким образом, старый «account\_id» привяжется к создаваемому аккаунту.

### **Миграция БД Admin**

Ранее данные об аккаунтах хранились в БД Admin.

Если LP разворачивается впервые, то будет создана БД Accounts с которой взаимодействует сервис Accounts. Если LP обновляется с версии 5.28.0 и меньшей, то БД Admin будет преобразована для работы с аккаунтами, а сервис Admin больше не будет взаимодействовать с базой данных Admin.

В руководство по обновлению LP 5 с предыдущей версии добавлен раздел «Преобразование БД Admin», а также раздел «Создание аккаунта с помощью сервиса API», где приведен пример CURL-запроса для создания нового аккаунта и миграции аккаунта, если его «account\_id» использовался в заголовке «Luna-Account-Id» без создания аккаунта в сервисе Admin.

### **Миграция аккаунтов с предыдущих версий Backport 3**

При обновлении с предыдущей версии Backport 3 для миграции аккаунтов нужно дополнительно запустить скрипт миграции после миграции базы данных Backport 3 (см. раздел «Миграция аккаунтов и токенов» в руководстве по обновлению LP 5 с предыдущей версии).

### **Миграция аккаунтов с LP 3.3.8**

При обновлении с LUNA PLATFORM 3.3.8 миграция аккаунтов будет выполнена при запуске скрипта «start\_migration.py» (см. раздел «Описание скрипта миграции» в руководстве по миграции с LUNA PLATFORM 3.3.8).

### **Миграция аккаунтов с LP 4.5.4**

При обновлении с LUNA PLATFORM 4.5.4 миграция аккаунтов выполняется аналогично описанию из пунктов «Миграция аккаунтов с предыдущих сборок LP 5» и «Миграция БД Admin» выше. В руководство по миграции с LUNA PLATFORM 4.5.4 добавлен раздел «Преобразование БД Admin», а также раздел «Создание аккаунта с помощью сервиса API», где приведен пример CURL-запроса для создания нового аккаунта и миграции аккаунта, если его «account\_id» использовался в заголовке «Luna-Account-Id» без создания аккаунта в сервисе Admin.

### **Сервис Backport 4**

В [API Backport 4](#) добавлена возможность авторизации по логину и паролю (BasicAuth) и по идентификатору аккаунта (LunaAccountIdAuth). Авторизация по токену (BearerAuth) невозможна.

Для сервиса User Interface 4 теперь используется авторизация по логину и паролю (параметр «BASIC\_AUTH» в .env-файле) вместо «account\_id» (параметр «LUNA\_ACCOUNT\_ID» в .env-

файле). Логин и пароль нужно указывать в формате login:password в Base64.

- Количество получаемых и удаляемых лиц, задач, обработчиков, событий и настроек LP, задаваемых в параметрах «page\_size» различных ресурсов и сервисов, увеличено с 100 до 1000.
- Обновлена обработка входящих URL-адресов в запросе на создание задачи [Estimator](#) с ZIP-архивом в качестве источника изображений. Если в указанном URL-адресе содержится адрес сервиса API, заданный в новой настройке «EXTERNAL\_LUNA\_API\_ADDRESS» в сервисе Configurator, объекты будут загружаться с помощью сервиса Image Store напрямую, а не отправлять запрос в сервис API с последующим перенаправлением в сервис Image Store.
- В настройки сервисов Handlers и Tasks добавлена новая секция «EXTERNAL\_LUNA\_API\_ADDRESS», предназначенная для корректной обработки ссылок на объекты, созданные с помощью ресурсов «/images» и «/objects» в сервисе API. В данной секции задается адрес и версия API сервиса API.

Если в качестве входных данных для ресурсов «/detector», «handlers/{handler\_id}/events», «/iso», «/sdk» и «verifiers/{verifier\_id}/verification» указывается URL-адрес и версия сервиса API объекта типа «images», совпадающие с адресом и версией API из секции «EXTERNAL\_LUNA\_API\_ADDRESS» сервиса Handlers, то данные объекты будут загружаться с помощью сервиса Image Store напрямую, а не отправлять запрос в сервис API с последующим перенаправлением в сервис Image Store.

Если параметре «content» > «source» > «reference» ресурса «/tasks/estimator» указывается URL-адрес и версия сервиса API до объекта типа «object» (ZIP-архив), совпадающие с адресом и версией API из секции «EXTERNAL\_LUNA\_API\_ADDRESS» сервиса Tasks, то данный объект будет загружаться с помощью сервиса Image Store напрямую, а не отправлять запрос в сервис API с последующим перенаправлением в сервис Image Store.

Пример формата: `http://10.15.3.144:5000/6/images/141d2706-8baf-433b-82eb-8c7fada847da`, где значение `http://10.15.3.144:5000` должно совпадать со значением из настройки «origin», а значение 6 должно совпадать со значением настройки `api_version` секции «EXTERNAL\_LUNA\_API\_ADDRESS».

Для избежания ошибок необходимо настроить данную секцию в настройках Handlers или Tasks перед использованием URL-адресов до объектов типа «objects» или «images» в качестве источника входных данных.

- Обновлены библиотеки вендора для HASP ключа и HASP утилиты.

Теперь вместо библиотек `haspvlib_x86_64_111186.so` и `haspvlib_111186.so` используются библиотеки `haspvlib_x86_64_30147.so` и `haspvlib_30147.so`, а вместо HASP утилиты `aksusbd-8.23-1.x86_64.rpm` используется утилита `aksusbd-8.43-1.x86_64.rpm`.

При обновлении на новую сборку LP требуется удалить старые библиотеки и утилиту HASP и установить новые. Сервис Licenses не запустится, если использована утилита и библиотеки

предыдущих версий.

Требуется обратиться в техническую поддержку для перевыпуска ключа LUNA PLATFORM 5.

### Исправленные ошибки

- Исправлена ошибка, при которой на запрос к ресурсу [«4/healthcheck»](#) возвращалась ошибка «Internal server error» при отключенном сервисе Tasks в секции ADDITIONAL\_SERVICES\_USAGE сервиса Configurator.
- Исправлена ошибка, при которой на запрос к ресурсу [«/luna\\_sys\\_info»](#) возвращалась ошибка «Internal server error», если в лицензионном файле отсутствовала функция выполнения проверок изображений на соответствие стандартам в ресурсах [«/iso»](#), [«/detector»](#), [«/handlers»](#) и [«/verifiers»](#).
- Исправлена ошибка в спецификации OpenAPI, при которой в схемах ответа на запросы [«get task result»](#) и [«generate events»](#) отображался тип Nullable для некоторых порогов [«face\\_quality»](#).

## LUNA PLATFORM v.5.28.0

### Изменения

- SDK обновлен до версии 5.9.0. Основные изменения SDK, затрагивающие LUNA PLATFORM 5:
  - Обновлен Liveness V2
  - Обновлен детектор лиц FaceDetV3

Значение по умолчанию настройки [«score\\_threshold»](#) в секции [«FACE\\_DETECTOR\\_V3»](#) сервиса Configurator изменено с **0.89** до **0.42**. Миграция настроек автоматически обновит это значение (см. раздел «Миграция базы данных Configurator» в руководстве по обновлению). Проверьте логику распознавания лиц, если вы используете значение [«score\\_threshold»](#), отличное от значения по умолчанию.

### Исправленные ошибки

- Исправлена ошибка, при которой игнорировался порог [«real\\_threshold»](#) при проверках Liveness в ресурсах [«/liveness»](#) и [«/sdk»](#). Ошибка могла приводить к неожиданному поведению системы при оценке Liveness.

## LUNA PLATFORM v.5.27.0

### Изменения

- В ресурс [«/detector»](#) добавлен новый параметр запроса «estimate\_face\_quality», позволяющий выполнить все проверки «face\_quality» из ресурсов [«/handlers»](#) и [«/verifiers»](#) с порогами по умолчанию.

Проверки в данном ресурсе можно выполнять с помощью соответствующей лицензируемой функции.

- В группу проверок «face\_quality» ресурсов [«/handlers»](#) и [«/verifiers»](#) и в ресурсы [«/iso»](#) и [«/detector»](#) добавлены новые проверки — «background\_lightness» и «background\_uniformity».

Проверка «background\_lightness» позволяет определить освещённость фона от 0 до 1, где:

- [0...0.1] — черный фон
- [0.1...0.3] — темный фон
- [0.3...0.97] — светлый фон
- [0.97...1] — белый фон

Проверка «background\_uniformity» позволяет определить степень однородности фона от 0 до 1, где 0 — фон неоднородный, 1 — фон однородный.

Новые проверки также добавлены в поле «face\_quality» структуры события в ресурсе [«ws handshake»](#).

Для использования данных проверок требуется соответствующая лицензируемая функция.

- Появилась возможность изменить порог «quality\_threshold» для Liveness V2, заданный в системе по умолчанию. Для этого используется соответствующая настройка в секции «LUNA\_HANDLERS\_LIVENESS\_SETTINGS» сервиса Configurator. Ранее этот порог невозможно было изменить для ресурсов [«/sdk»](#) и [«/liveness»](#).

Значение порога по умолчанию — 0.5.

Ранее данный порог находился в настройке «config.py» сервиса Handlers и имел название «LIVENESS\_V2\_QUALITY\_THRESHOLD».

При использовании ресурсов [«/handlers»](#) или [«/verifiers»](#) значение порога из Configurator будет переопределено соответствующей настройкой в параметре запроса.

- В тело запроса на получение статистики по событиям ([«/events/statistic»](#)) добавлена возможность указания атрибутов тел в поля «filters» и «targets».
- Расширен мониторинг сервиса Handlers. Теперь кроме отслеживания запросов (Requests series), запросов, выполненных с ошибкой (Errors series), количества выполненных оценок SDK (Usages\_statistic series) и лицензирования (Licensing series) фиксируются данные о каждой выполненной оценке SDK.

Например, для каждой оценки наличия маски на лице фиксируется время выполнения этой оценки в секундах.

См. подробную информацию и перечень оценок по которым собираются данные в разделе «Monitoring» руководства разработчика сервиса Handlers.

### Исправленные ошибки

- Исправлена ошибка, из-за которой URL-адреса БО лиц и атрибутов возвращались в неправильном формате при запросе к ресурсу [«perform verification»](#).
- В спецификации OpenAPI исправлена схема ответа поля «stats» в теле ответа на запрос [«get statistics on events»](#).

Ранее было указано, что в качестве значений для поля «stats» могут возвращаться массивы из типов «string», «number», «integer», «boolean», «Array of any» и «object». Теперь для поля «stats» указано, что может возвращаться непустой массив, содержащий массивы из типов «integer», «number» и «string».

- Исправлено появление ошибки «internal server error», возникавшей при обработке изображений, у которых в полях координат EXIF данных содержалось значение NaN.

## LUNA PLATFORM v.5.26.0

### Изменения

- SDK обновлен до версии 5.8.0.
- Обновлена оценка определения состояния масок. Помимо трёх основных статусов теперь определяются следующие дополнительные свойства:
  - «correct» — маска присутствует на лице, рот и нос закрыты маской
  - «mouth» — маска присутствует на лице и закрывает только рот
  - «clear» — на лице нет маски
  - «chin» — маска присутствует на лице и находится под подбородком, не перекрывая зону от глаз до рта
  - «partially» — лицо частично перекрыто, но не медицинской маской и не маской с полным перекрытием лица
  - «full» — маска присутствует на лице, при которой полностью закрыто лицо, например, балаклава/лыжная маска

Каждому основному статусу маски соответствует одно из двух свойств. Наиболее вероятное свойство возвращается в поле «predominant\_occlusion»:

- статусу «medical\_mask» соответствует свойство «correct» или «mouth»
- статусу «missing» соответствует свойство «clear» или «chin»
- статусу «occluded» соответствует свойство «partially» или «full»

Для каждого из свойств возвращается вероятностная оценка в диапазоне [0..1].

Дополнительные свойства маски не записываются в БД и по ним не выполняется фильтрация.

Обратите внимание, что наличие маски на подбородке с этой версии относится к состоянию «missing». В предыдущих версиях оно относилось к состоянию «medical\_mask».

Дополнительные свойства возвращаются при оценке состояний маски в ресурсах «[sdk](#)» и «[detect faces](#)», а также при создании событий (запросы «[save event](#)» и «[generate events](#)») и выполнении верификации (запрос «[perform verification](#)»). Дополнительные свойства маски также учитываются при отправке событий через веб-сокеты (запрос «[ws handshake](#)»)

Пример определения обновленного состояния маски в ресурсе «/sdk»:

```
"mask": {
  "predominant_mask": "occluded",
  "estimations": {},
  "face_occlusion": {
    "predominant_occlusion": "correct",
    "estimations": {
      "full": 0.019,
      "clear": 0.02,
      "correct": 0.6108324766,
      "partially": 0.31,
      "mouth": 0.0209,
      "chin": 0.019097
    }
  }
},
```

- В структуру отчета в формате CSV (см. задачи «[Reporter](#)» и «[Exporter](#)») добавлены колонки атрибутов тел «`apparent_gender`», «`apparent_age`», «`headwear_state`», «`sleeve_length`», «`upper_clothing_colors`» и «`backpack_state`».
- Для параметров «`num_threads`» из настроек «`LUNA_HANDLERS_HUMAN_EXTRACTOR_RUNTIME_SETTINGS`», «`LUNA_HANDLERS_WARP_ESTIMATOR_RUNTIME_SETTINGS`», «`LUNA_HANDLERS_EXTRACTOR_RUNTIME_SETTINGS`», «`LUNA_HANDLERS_DETECTOR_RUNTIME_SETTINGS`» и «`LUNA_HANDLERS_HUMAN_DETECTOR_RUNTIME_SETTINGS`» изменено значение по умолчанию с 4 до 6. Данное изменение повышает производительность по умолчанию для серверов с большим количеством ядер CPU.

### Исправленные ошибки

- Исправлена ошибка, при которой в задачах можно было задать пустые значения для фильтров `handler_ids`, `masks`, `face_ids` и др., что приводило к фильтрации по всем соответствующим объектам.

- Исправлено отсутствие ошибки при вводе неправильного логина/пароля в пользовательском интерфейсе сервиса Admin.
- Теперь при отсутствии лицензируемой функции, в логах сервиса Licenses явно прописывается название отсутствующей функции.

## LUNA PLATFORM v.5.25.0

### Изменения

- В ресурс [«sdk»](#) и политику [«detect\\_policy»](#) ресурса [«/handlers»](#) добавлена возможность определения атрибутов тел (см. дополнительную информацию об атрибутах тела в [документации SDK](#)).

Доступно определение следующих атрибутов:

- пол (мужской, женский, неизвестно) и возраст по изображению тела. Определение пола и возраста таким способом является менее точным, чем по лицу.
- наличие головного убора, тип рукавов (длинные рукава, короткие рукава, неизвестно), цвет верхней одежды (чёрный, синий, зеленый, серый, оранжевый, фиолетовый, красный, белый, желтый, неизвестно). В будущих обновлениях будут добавлены следующие цвета: коричневый, розовый, хаки, бежевый, разноцветный.
- наличие рюкзака.

Для использования нового функционала требуется отдельная лицензия. Получить информацию о статусе лицензии можно запросом к ресурсу [«/luna\\_sys\\_info»](#) (поле [«license\\_info»](#) > [«body\\_attributes»](#)).

Новые атрибуты тел добавлены в структуру событий (см. [«detections»](#) > [«samples»](#) > [«body»](#) > [«detection»](#) > [«attributes»](#)).

Пример определения атрибутов тел в сгенерированном событии, с включенными параметрами [«estimate\\_upper\\_body»](#), [«estimate\\_accessories»](#) и [«estimate\\_basic\\_attributes»](#) в обработчике:

```
"attributes": {
  "basic_attributes": {
    "apparent_age": 25,
    "apparent_gender": 0
  },
  "upper_body": {
    "headwear": {
      "state": 0
    },

```

```

        "sleeve": {
            "length": "short"
        },
        "upper_clothing": {
            "colors": [
                "white",
                "black"
            ]
        }
    },
    "accessories": {
        "backpack": {
            "state": 0
        }
    }
}

```

В политику «match\_policy» ресурса [«/handlers»](#) и в ресурс [«human body matching»](#) добавлены фильтры атрибутов тел «apparent\_gender», "apparent\_age\_\_gte«,»apparent\_age\_\_lt«,»headwear\_states«,»sleeve\_lengths«,»upper\_clothing\_colors"и «backpack\_states», указываемые при использовании событий в качестве кандидатов. Также добавлена возможность задать атрибуты тела в качестве значений для поля «targets».

Фильтры атрибутов тел также могут быть использованы при отправке событий через веб-сокеты (см. ресурс [«ws handshake»](#)).

Соответствующие фильтры для событий добавлены в задачи Cross-matching, Clustering, Reporter, Exporter и Linker.

Поддержана возможность указания атрибутов тел при создании нового обработчика в задаче Estimator.

Для атрибутов тел доступна возможность агрегации при [генерации/сохранении](#) событий и выполнении оценки в ресурсе [«sdk»](#). Агрегированные значения атрибутов отображаются в полях «aggregate\_estimations» соответствующих ресурсов.

- Теперь для проверки состояния запущенного контейнера сервиса при старте с помощью start\_platform.sh выполняется запрос на ресурс [«/healthcheck»](#).

### Исправленные ошибки

- Исправлена проверка событий, генерируемых запросом [«save event»](#). Если в событии были заданы проверки из группы параметров «face\_quality», то событие могло быть отправлено в сервис Sender в неправильном формате.
- Исправлена ошибка, при которой можно было задать непустые значения для фильтров emotions, masks и liveness при указании событий в качестве кандидатов в политике



«match\_policy» ресурса [«/handlers»](#).

- Исправлено падение скрипта start\_platform.sh при изменении значения переменной «DATA» в .env файле.

## LUNA PLATFORM v.5.24.2

### Изменения

#### Исправленные ошибки

- Настройка «reference\_limit» сервиса Python Matcher теперь учитывается при сравнении с кандидатами типа «event\_external\_id». Игнорирование данной настройки приводило к ошибке «Database connection timeout error», возникавшая при сравнении большого количества лиц с одинаковым «external\_id».
- Исправлено логирование в сервисе LUNA Handlers, связанное с инициализацией, эстимированием и другими событиями.
- Исправлено логирование модуля MatcherLib сервиса Python Matcher. Ранее логи не записывались на уровне INFO.

Также при включенном Cached Matcher, модуль MatcherLib сохранял логи в директорию «/tmp» и данное поведение не отключалось с помощью настройки «LUNA\_PYTHON\_MATCHER\_LOGGER.LOG\_TO\_FIE=false».

- Исправлена работа модуля MatcherLib сервиса Python Matcher с 52 версией нейронной сети.

## LUNA PLATFORM v.5.24.1

### Изменения

- В тела ответов запросов, выполненных с ошибкой, добавлено поле «link», содержащее ссылку на [онлайн-документацию](#) с подробным описанием полученной ошибки. Ссылки на описания ошибок также добавлены в примеры ответов в документации OpenAPI (поле «Response samples»).

Для перехода по ссылкам необходимо подключение к сети Интернет.

#### Исправленные ошибки

- Удалено сообщение `sanic.error: <ApiRequest: OPTIONS /6/objects> body not consumed` из логов сервиса LUNA API. Сообщение выводилось если в запросе OPTIONS /6/objects было непустое тело.

- Обработка пустых ZIP-архивов и директорий из S3-подобных хранилищ с помощью задачи «Estimator» больше не возвращает никаких ошибок. Ранее при таких обработках возвращалась ошибка «Unsupported media type».
- Исправлена ошибка при сохранении события со значением поля «detect\_ts=0». При попытке получить событие с таким значением, возвращалось значение «detect\_ts=null».
- Исправлена ошибка, при которой обновление ограничения или настройки с пустой строкой в поле «description» в сервисе Configurator приводило к выставлению значения «null» в поле «description». Ошибка возникала при использовании базы данных Oracle.
- Исправлена ошибка «Internal server error», возникавшая при сохранении события (запрос «save event») со значением «predominant\_mask»: «medical\_mask».
- Исправлена ошибка, при которой все эstimаторы (DETECTOR, EXTRACTOR, WARP, HUMAN\_EXTRACTOR, HUMAN\_DETECTOR) использовали значения параметра device\_class («gpu» или «cpu») из настройки «LUNA\_HANDLERS\_DETECTOR\_RUNTIME\_SETTINGS», игнорируя остальные соответствующие эstimаторам настройки, что приводило к запуску всех эstimаторов только на GPU или только на CPU.

Известная проблема: Настройки num\_threads и num\_compute\_streams по-прежнему определяются для всех эstimаторов в секции «LUNA\_HANDLERS\_DETECTOR\_RUNTIME\_SETTINGS». Соответствующие настройки в остальных секциях игнорируются.

## LUNA PLATFORM v.5.24.0

### Изменения

- В задачу Estimator ([печать «/tasks/estimator»](#)) добавлена поддержка FTP-сервера в качестве источника изображений (параметр «source\_type»).

Для такого типа источника в теле запроса доступно задание следующих параметров для соединения с FTP-сервером:

- Параметр «host» — IP-адрес или имя хоста FTP-сервера (обязательное поле);
- Параметр «port» — порт FTP-сервера;
- Параметр «max\_sessions» — максимальное количество разрешенных сеансов на FTP-сервере;
- Параметры «user» и «password» — параметры авторизации (обязательные поля).

Как и в задачах Estimator с использованием S3-подобного хранилища или сетевого диска в качестве источников изображений, доступна возможность задания пути до директории с изображениями, рекурсивного получения изображений из вложенных директорий, выбора типа передаваемых изображений, а также указания префикса и постфикса.

Для получения корректных результатов обработки изображений с помощью задачи Estimator все обрабатываемые изображения должны быть либо в исходном формате, либо в формате биометрических образцов.

См. соответствующие примеры и дополнительную информацию в [спецификации OpenAPI](#).

- Для событий добавлен новый параметр «detect\_ts», позволяющий хранить временную метку (timestamp) относительно чего-либо, например, относительно начала видеофайла (см. тело ответа запроса [generate events](#)).

Значение параметра «detect\_ts» также задать вручную с помощью запроса «[save event](#)».

- Значительно снижено потребление памяти для задач Clustering, ROC-curve calculating и Cross-matching, а также для сервисов Python Matcher и Image Store.

Также для увеличения производительности работы в ресурсах «[cross matching faces](#)» и «[cross matching bodies](#)» сервиса Python Matcher заменен Content-Type тела ответа с application/json на application/msgpack.

- В ресурсы «[cross matching faces](#)» и «[cross matching bodies](#)» сервиса Python Matcher добавлен новый параметр запроса «sorting», позволяющий отключить сортировку результатов сравнения в лексикографическом порядке.
- Прекращена поддержка БД Vertica для сервиса Events. Настройки смены БД на Vertica удалены из настроек сервиса Configurator, а также удалена соответствующая информация из документации.
- Добавлена проверка наличия инструкций avx2 на процессоре при старте контейнеров с помощью скрипта Docker Compose. Если процессор не имеет поддержки avx2, то при старте контейнеров будет выдана ошибка avx2 not supported.

### Исправленные ошибки

- Исправлена ошибка, при которой для некоторых параметров запроса с типом «number» обрабатывались только вещественные числа. При вводе целых чисел возникала ошибка Failed to validate input json.

## LUNA PLATFORM v.5.23.1

### Исправленные ошибки

- Исправлена ошибка, из-за которой при выполнении некоторых оценок с использованием 8-битных png изображений выдавались некорректные результаты.

Актуально для следующих ресурсов:

- [detect faces](#)

- [iso resource](#)
  - [generate events](#)
  - [perform verification](#)
  - [sdk](#)
- Исправлена ошибка в сервисе Python Matcher. Сервис не перезагружался после ввода некорректных параметров в настройки сервиса Configurator, если тип базы данных oracle был указан как db\_type в разделе «LUNA\_FACES\_DB».
  - Исправлена ошибка в примере плагина «Thin Event», приводившая к двойной инициализации плагина.
  - Теперь сервис Handlers не проверяет соединение с сервисом Events, если второй отключен в настройке «ADDITIONAL\_SERVICES\_USAGE» в сервисе Configurator.

## LUNA PLATFORM v.5.23.0

### Изменения

- В задачу Estimator ([песчпс «/tasks/estimator»](#)) добавлена поддержка сетевого диска в качестве источника изображений (параметр «source\_type»).

Для такого типа источника в теле запроса доступно задание следующих параметров:

- Параметр «path» — абсолютный путь к директории с изображениями в контейнере (обязательное поле);
- Параметр «follow\_links» — включение/выключение обработки символических ссылок (symlink);
- Параметр «prefix» — префикс ключа файла. Может использоваться для загрузки изображений из определенной директории;
- Параметр «postfix» — постфикс ключа файла. Может использоваться для загрузки изображений с определенным расширением.

См. пример использования префиксов и постфиксов в описании ресурса [«/tasks/estimator»](#).

При использовании такого типа источника и запуске сервисов Tasks и Tasks Worker через Docker-контейнеры необходимо смонтировать директорию с изображениями с сетевого диска в локальную директорию и синхронизировать её с указанной директорией в контейнере. Смонтировать директорию с сетевого диска можно любым удобным способом. После этого можно синхронизировать смонтированную директорию с директорией в контейнере с помощью следующей команды при запуске сервисов Tasks и Tasks Worker:

```
docker run \
...
```

```
-v /var/lib/luna/current/images:/srv/images
...
```

`/var/lib/luna/current/images` — путь к предварительно смонтированной директории с изображениями с сетевого диска.

`/srv/images` — путь к директории с изображениями в контейнере, куда они будут перенесены с сетевого диска. Этот путь должен быть указан в теле запроса задачи Estimator (параметр «path»).

Как и в задаче Estimator с использованием S3-подобного хранилища в качестве источника изображений, доступна возможность рекурсивного получения изображений из вложенных директорий бакета (параметр «recursive») и выбора типа передаваемых изображений (параметр «image\_type»). Для получения корректных результатов обработки нужно использовать однотипные изображения (исходное изображение, биометрический образец лица/тела).

См. соответствующие примеры и дополнительную информацию в [спецификации OpenAPI](#).

- В группу проверок «face\_quality» ресурсов «/handlers» и «/verifiers» сервисов API и Handlers добавлены две новые проверки изображения на соответствие [стандарту ICAO](#) — «illumination\_uniformity» и «dynamic\_range».

Проверка «illumination\_uniformity» позволяет проверить равномерность освещения лица на изображении. В соответствии со стандартом ICAO рекомендуется использовать цветные изображения. При использовании черно-белых изображений результаты могут быть неожиданными. Рекомендуется использовать эту проверку при необходимости получения результатов, соответствующих стандарту ICAO. В остальных случаях можно использовать алгоритм VisionLabs для проверки равномерности освещения лица на изображении (проверка «illumination\_quality»).

Проверка «dynamic\_range» позволяет проверить динамический диапазон тона кожи лица.

Для данных проверок невозможно использовать биометрический образец в качестве входного изображения.

См. подробное описание проверок в главе «Параметры лиц и изображений» в руководстве администратора.

- Обновлены пороги по умолчанию (рекомендуемые) для следующих проверок в группе проверок «face\_quality» и ресурсе «/iso»:

Наименование проверки	Старые значения порогов	Новые значения порогов
mouth_occluded	min=0, max=0.3	min=0, max=0.5
mouth_open	min=0, max=0.64	min=0, max=0.5

Рекомендуется выставить обновленные пороги для данных проверок в созданных ранее обработчиках.

- Ускорено выполнение запросов на сравнение тел, когда в качестве эталона указано событие.

### Исправленные ошибки

- Исправлена ошибка, при которой в теле ответа результата задачи Estimator в поле «filename» группы «detections» возвращалось значение «raw image» вместо имени файла.

## LUNA PLATFORM v.5.22.0

### Изменения

- SDK обновлен до версии 5.6.0.
- В политику «detect\_policy» ресурсов [«/handlers»](#) и [«/verifiers»](#) сервисов API и Handlers добавлена группа проверок «face\_quality», позволяющая настраивать проверку входящих изображений и лиц на этих изображениях в соответствии с предварительно заданными условиями. С помощью «face\_quality» возможно проверять изображения лиц на соответствие требованиям стандарта ISO/IEC 19794-5:2011 или схожих.

### Доступные проверки

Доступны следующие проверки лиц и изображений:

- **Положение головы:** «head\_yaw», «head\_pitch», «head\_roll»
- **Направление взгляда:** «gaze\_yaw», «gaze\_pitch»
- **Наличие улыбки:** «mouth\_smiling»
- **Наличие перекрытого рта:** «mouth\_occluded»
- **Наличие открытого рта:** «mouth\_open»
- **Тип улыбки, который определяется при наличии улыбки** (нет улыбки, улыбка с закрытым ртом, улыбка с открытым ртом): «smile\_properties»
- **Статус глаз** для каждого глаза (открыты, закрыты, перекрыты): «left\_eye», «right\_eye»
- **Наличие эффекта красных глаз:** «red\_eyes»
- **Расстояние между центрами глаз:** «eye\_distance»
- **Статус бровей** (нейтральный, подняты, опущены, нахмурены): «eyebrows\_state»
- **Естественность освещения:** «natural\_light»
- **Наличие бочкообразной дисторсии (эффект Fisheye):** «radial\_distortion»
- **Отсутствие бликов:** «specularity\_quality»
- **Размытость:** «blurriness\_quality»
- **Контраст и насыщенность** (недостаточная или слишком большая экспозиция): «dark\_quality», «light\_quality»

- **Тип цвета по лицу** (цветное, черно-белое, инфракрасное (ближний инфракрасный диапазон)): «face\_color\_type»
- **Положение центральной точки лица** на фотоизображении по горизонтали и вертикали: «head\_horizontal\_center», «head\_vertical\_center»
- **Вертикальный и горизонтальный размер головы относительно размера фотоизображения**: «head\_width», «head\_height»
- **Ширина и высота лица**: «face\_width», «face\_height»
- **Отступы лица от краёв фото**: «indent\_upper», «indent\_lower», «intent\_left», «intent\_right»
- **Размер изображения в байтах**: «image\_size»
- **Ширина и высота изображения**: «image\_width», «image\_height»
- **Соотношение сторон изображения**: «aspect\_ratio»
- **Формат изображения** (JPG, JPG2000, PNG): «image\_format»
- **Наличие и тип головного убора** (нет, кепка, шапка, фуражка, шарф, ушанка, шлем/каска, капюшон, шляпа, другое): «headwear\_type»

Описание каждой из проверок приведено в главе «Параметры лиц и изображений» в руководстве администратора.

#### **Включение проверок face\_quality**

Проверки выполняются для каждой детекции лица, найденного на фото. Результаты перечисленных выше проверок не агрегируются. Можно включать и отключать обработку изображений с несколькими лицами с помощью опции «multiface\_policy».

Для включения проверок необходимо указать значение «1» в поле «estimate» для «face\_quality». По умолчанию проверка изображений отключена.

Для включения фильтрации по результатам проверок необходимо указать значение «1» в поле «filter». При провале одной или более проверок для детекции лица дальнейшие политики для данной детекции выполнены не будут. При этом результаты, полученные в соответствии с политикой «detect\_policy» для данной детекции, будут возвращены в ответе.

Параметры группы проверок «face\_quality»:

```
"face_quality": {
  "estimate": 0,
  "filter": 0,
  "checks": {}
}
```

Выполняемые проверки и их параметры перечислены в группе «checks».

Каждую из проверок можно включать или отключать. Это позволяет задать набор проверок, необходимых для конкретного сценария.

Для отключения проверки требуется выставить «estimate»: 0 для конкретной проверки. По умолчанию все проверки включены и будут выполнены при включении «face\_quality».

В зависимости от типа проверки пользователь может указать минимальное и максимальное значения порога или допустимые значения для этой проверки. Для этого используется поле «threshold».

Пример задания параметров проверок:

```
"checks": {
  "image_format": {
    "estimate": 1,
    "threshold": [
      "JPEG",
      "JPEG2000",
      "PNG"
    ]
  },
  "illumination_quality": {
    "estimate": 1,
    "threshold": {
      "min": 0.3,
      "max": 0.9
    }
  }
}
```

### Возвращаемый ответ

Название проверки («name»), определённое значение («object\_value»), заданные значения порогов/допустимые значения («threshold\_value») и итоговый вердикт («result») выводятся в ответе на запрос [«/handlers/{handler\\_id}/events»](#) в «events» > «detection» > «samples» > «face» > «detection» > «face\_quality».

По результатам всех проверок выводится общий вердикт (поле «status»), который равен «1» при успешном прохождении всех проверок и равен «0» при провале хотя бы одной проверки.

Результаты проверок не сохраняются в БД, они возвращаются только в ответе.

Пример ответа системы для двух проверок:

```
"face_quality": {
  "status": 1,
  "checks": [
    {
      "name": "image_format",
      "object_value": "PNG",
```



```

        "threshold_value": [
            "JPEG",
            "JPEG2000",
            "PNG"
        ],
        "result": 1
    },
    {
        "name": "illumination_quality",
        "object_value": 0.5182177424430847,
        "threshold_value": {
            "min": 0.3,
            "max": 1.0
        },
        "result": 1
    }
]

```

### Лицензирование

Данный функционал лицензируется отдельно, в ключе должна быть прописана опция ISO estimation. Для группы параметров «face\_quality» и ресурса [«/iso»](#) используется одна и та же лицензия.

Если в лицензии отсутствует опция ISO estimation, то при «face\_quality» > «estimate»: 1 вернётся ошибка «License problem: „ISO license feature is disabled.“».

### Отличие проверок для «face\_quality» и ресурса «/iso»

Набор проверок для «face\_quality» и ресурса «/iso» отличается. См. раздел «Проверка изображений» в руководстве администратора. В главе приведена таблица сравнения доступных проверок.

- В ресурс [«/iso»](#) добавлены следующие новые проверки:
  - Статус бровей («eyebrows\_state»)
  - Наличие головного убора («headwear\_type»)
  - Тип улыбки, который определяется при наличии улыбки («smile\_properties»)
  - Проверка естественного освещения («natural\_light»)
  - Наличие бочкообразной дисторсии (эффект Fisheye) («radial\_distortion»)
  - Наличие эффекта красных глаз («red\_eyes»)
  - Тип цвета по лицу («face\_color\_type»)

Данные проверки проводятся в соответствии со значениями порогов по умолчанию, заданными в системе. Если требуется изменить порог или отключить проверку, необходимо воспользоваться группой проверок «face\_quality» в ресурсах [«/handlers»](#) и [«/verifiers»](#).

Описание каждой из проверок приведено в главе «Параметры лиц и изображений» в руководстве администратора.

- Обновился внутренний механизм эстимаций для сервиса Handlers, что значительно сократило потребление ресурсов (CPU, GPU, RAM) для выполнения вычислений. Теперь можно запускать больше экземпляров сервиса Handlers или увеличить количество «рабочих процессов» для одного экземпляра на тех же мощностях, что приводит к ускорению выполнения запросов.

В некоторых случаях данное изменение может приводить к снижению производительности при старой конфигурации Handlers. В таком случае требуется увеличить количество экземпляров или «рабочих процессов» Handlers для повышения производительности.

- Было ускорено сравнение по большому списку (более 100 000 лиц) при большом количестве одновременно выполняемых запросов. Скорость выполнения таких запросов в некоторых случаях увеличена до двух раз.
- В примерах спецификации OpenAPI, поле «exif» расширено информацией об ориентации для всех ресурсов, поддерживающих извлечение exif.

## LUNA PLATFORM v.5.21.0

### Изменения

- SDK обновлен до версии 5.5.1.
- В задачу пакетной обработки «/tasks/estimator» добавлена поддержка S3-подобного хранилища в качестве источника изображений (параметр «source\_type»).

Для такого типа источника доступно задание следующих параметров:

- Имя бакета/[Access Point ARN](#)/[Outpost ARN](#) (обязательное поле);
- Endpoint (только при указании имени бакета);
- Bucket region (только при указании имени бакета);
- [Префикс ключа файла](#). Также может использоваться для загрузки изображений из определенной директории, например, «2022/January».

Для настройки авторизации предназначены следующие параметры:

- Public access key (обязательное поле);
- Secret access key (обязательное поле);
- Signature version («s3v2»/«s3v4»).

Также доступна возможность рекурсивного получения изображений из вложенных директорий бакета (параметр «recursive») и сохранения исходных изображений (параметр «save\_origin»).

См. соответствующие примеры и дополнительную информацию в [спецификации OpenAPI](#).

- Улучшена производительность обработки лиц в задаче [прикрепления лица к списку](#).

Ранее сервис Tasks запрашивал у сервиса Faces полный набор целевых полей, что уменьшало производительность обработки лиц. Теперь сервис запрашивает только необходимое ему целевое поле «face\_id».

- Увеличено количество подключений к БД LUNA Admin по умолчанию с 1 до 5 (настройка «connection\_pool\_size», секция «LUNA\_ADMIN\_DB» в сервисе Configurator).
- Значительно уменьшено потребление памяти при использовании задачи [перекрестного сравнения](#).

### Исправленные ошибки

- Исправлена ошибка подключения к базам данных сервисов, при которой в некоторых сценариях работы подключения не возвращались в пул подключений, что приводило к возвращению кода состояния 500 и ошибке 10017 (превышен лимит времени подключения к базе данных).

## LUNA PLATFORM v.5.20.0

### Изменения

- В задачу сборки мусора (garbage collection task) добавлена возможность удаления исходных изображений, связанных с событием. Для этого используется параметр «remove\_image\_origins», который задаётся в теле запроса.

Опция добавлена в следующие ресурсы:

- «/tasks/gc» сервиса API.
- «/tasks/gc» сервиса Admin.
- «/tasks/gc» сервиса Tasks.

Возможность удаления исходных изображений также добавлена в пользовательский интерфейс сервиса Admin: «Tasks» > «Garbage Collecting tasks» > «start descriptors gc» > «Remove image origins».

Если изображения хранятся во внешнем источнике, то система попытается удалить их.

- В ответе на запрос «[delete events](#)» сервиса Events теперь может возвращаться поле «image\_origins», содержащее URL всех исходных изображений, связанных с удаляемым событием. Эти URL могут быть использованы для последующего удаления исходных изображений.

В ресурс добавлен параметр запроса «tagret». В нём можно указать следующие целевые поля:

- «face\_samples».
- «body\_samples».
- «image\_origins».

В зависимости от указанных целевых полей после удаления события в ответе вернётся информация о связанных с этим событием объектах: ID биометрических образцов лиц/тел, URL исходных изображений.

По умолчанию в теле ответа отображается информация обо всех связанных с событием объектах: «face\_samples», «body\_samples» и «image\_origins».

Если оставить поле «tagret» пустым, то вернётся только «event\_id» удаляемого события.

- Для ресурсов [«face matching»](#) и [«human body matching»](#) добавлены новые типы эталонов:
  - «event\_track\_id» позволяет выполнить сравнение с каждым событием, содержащим указанный «track\_id».
  - «external\_event\_id» позволяет выполнить сравнение с каждым событием, содержащим указанный «external\_id».

Использование данных типов позволяет выполнять поиск сразу по нескольким событиям, связанным с одним телом или лицом, чтобы снизить влияние условий съёмки (освещения, расположения относительно камеры и т. д.) на точность распознавания.

Для каждого события, которое используется в качестве эталона, в ответе возвращается «event\_id».

- В руководство администратора добавлен раздел «Продвинутая настройка PostgreSQL» с информацией о конфигурации PostgreSQL для работы с LUNA PLATFORM.

### Исправленные ошибки

- В сервисе Events исправлена ошибка, при которой в запросе [«create new events»](#) неправильно обрабатывались специальные символы, введенные в полях «label» («matches» > «label») и «external\_id» («matches» > «candidates» > «face»/«event» > «external\_id»).
- Исправлена ошибка агрегирования результатов оценки Liveness V2 для ресурса [«/liveness»](#). При отправке в ресурс одного фотоизображения низкого качества, для которого система возвращала «prediction» = 2, в агрегированных результатах для того же фотоизображения возвращалось иное значение «prediction».

## LUNA PLATFORM v.5.19.0

### Изменения

- Максимальный размер архива, передаваемого в задачу пакетной обработки [«/tasks/estimator»](#), был увеличен с 1 Гб до 100 Гб. Не рекомендуется передавать архивы большего размера, так как это может повлиять на производительность системы.
- В сервисе Faces ускорено выполнение запросов получения лиц (ресурс [«/faces»](#)) и получения количества лиц (ресурс [«/faces/count»](#)) при фильтрации по списку.

### Исправленные ошибки

- Исправлена ошибка, когда при отключённой функции Liveness в логах сервиса Handlers выводилось сообщение, что количество транзакций Liveness является безлимитным.

## LUNA PLATFORM v.5.18.0

- В сервисы API и Handlers добавлен новый ресурс [«/iso»](#). Ресурс выполняет оценку фотоизображения лица на соответствие требованиям стандарта ISO/IEC 19794-5.

В ответе ресурса возвращается вердикт по каждой проверке (1 — фотоизображение соответствует стандарту, 0 — полученное значение не соответствует стандарту) и общий вердикт на основании всех выполненных проверок. Общий вердикт равен 1, если фотоизображение успешно прошло все проверки.

Для каждой проверки возвращается полученное значение и порог/значение с которым выполняется сравнение. Пороги заданы в системе в соответствии с требованиями стандарта ISO/IEC 19794-5 и не изменяются пользователем.

Пример:

```
{
  "name": "head_roll",
  "object_value": 5.434040069580078,
  "threshold_value": {
    "min": -8,
    "max": 8
  },
  "result": 1
},
```

Сейчас доступны следующие проверки:

- Качество фотоизображения: контраст и насыщенность (недостаточная или слишком большая экспозиция), фокусировка, засветы и зеркальность, равномерность освещения.
- Статус рта (открыт, закрыт, перекрыт).

- Наличие очков (нет очков, обычные очки, солнцезащитные очки).
- Статус глаз (для каждого глаза: открыт, закрыт, перекрыт).
- Направление взгляда.
- Углы поворота головы (угол поворота, наклона и отклонения).
- Положение центральной точки лица на фотоизображении по горизонтали и вертикали.
- Вертикальный и горизонтальный размер головы относительно размера фотоизображения.
- Расстояние между центрами глаз.
- Формат изображения.

Требования можно найти на официальном сайте: <https://www.iso.org/obp/ui/#iso:std:iso-iec:19794:-5:en>

В запросе можно дополнительно включить извлечение EXIF данных фотоизображения.

По умолчанию проверки выполняются для фотоизображений, на которых присутствует одно лицо. Можно включить оценку для нескольких лиц на фотоизображении с помощью параметра «multiface\_policy». Для каждого из найденных лиц вернутся оценки и координаты найденного лица. Следует учитывать, что многие проверки по стандарту ISO предполагают наличие одного лица в кадре, поэтому не все проверки для нескольких лиц будут выполнены успешно.

Порядок возвращаемых ответов после обработки соответствует порядку передаваемых фотоизображений.

Если одно или несколько передаваемых в запросе изображений повреждено, то вернётся ошибка. Остальные изображения в запросе будут обработаны в обычном режиме.

Данный функционал лицензируется отдельно. Если в лицензии отсутствует опция iso, то при использовании ресурса «/iso» вернётся ошибка.

См. дополнительную информацию в разделе «Проверка изображений по стандарту ISO/IEC 19794-5» в руководстве администратора и в документации OpenAPI.

- Удалены опции `profile` и `limitations-file` для скрипта `db_create.py`. Опция `profile` использовалась для заполнения базы данных сервиса Configurator настройками на основе профиля по умолчанию. `Limitations-file` использовалась для загрузки шаблонов настроек из файла.

Сейчас для создания пустой базы рекомендуется использовать команду `db_create.py` без дополнительных параметров, затем загружать необходимые настройки из файла с помощью флага `dump-file` или выполнять миграция настроек командой `python3.9 -m configs.migrate --config /srv/luna_configurator/configs/config.conf --profile platform head`.

- В комплект поставки и в документации была добавлена библиотека вендора для HASP ключа haspvlib\_x86\_64\_30147.so.

## LUNA PLATFORM v.5.17.0

### Изменения

- Добавлен функционал для подсчёта статистики по выполненным запросам. Полученная статистика позволит проводить анализ использования LP для дальнейшего улучшения системы. Подсчитываются:
  - количество запросов на ресурсы LP, например, «POST: „handlers/handler\_id/events“».
  - суммарное количество оценок одного типа (например, «estimate\_liveness»).
  - количество успешных запросов и запросов, завершившихся с ошибкой.

Статистика ведётся для каждого кода ошибки по каждому ресурсу для каждого сервиса. Статистика сортируется по месяцам.

Сбор статистики работает только при включённом мониторинге и установленной InfluxDB версии 2.0.8 и более поздних версий.

Мониторинг теперь включён в сервисах по умолчанию. Отправка этих данных в мониторинг включена по умолчанию. Информация является обезличенной и содержит только количественные данные.

Для получения статистики следует отправить запрос на ресурс [«/luna\\_sys\\_info»](#) сервиса Admin или перейти в GUI сервиса Admin на вкладку «help» и нажать «Get LUNA PLATFORM system info». Необходимая информация содержится в разделе «stats».

Пример ответа:

```
{
  "stats": {
    "estimators_stats": [
      {
        "count": 1,
        "month": "2021-09",
        "name": "body_descriptor_extractor_usages"
      }
    ],
    "routes_stats": [
      {
        "service": "luna-api",
        "route": "GET:/version",
```

```

        "month": "2021-09",
        "errors": [
            {
                "count": 1,
                "error_code": "12012"
            }
        ],
        "request_stats": [
            {
                "count": 1,
                "status_code": "200"
            }
        ]
    }
    [...]
]
}

```

Статистика подсчитывается в InfluxDB на основе данных из бакета «luna\_monitoring» и хранится в бакете «luna\_monitoring\_aggregated». Бакеты создаются в InfluxDB. Не следует удалять данные из данного бакета, иначе будет невозможно получить статистику.

Задачи для подсчёта статистики можно найти в GUI InfluxDB на вкладке «Tasks».

Статистика подсчитывается раз в сутки, поэтому не отображается сразу после запуска LP. Можно вручную запустить выполнение задач на вкладке «Tasks» в GUI InfluxDB.

Для включения подсчёта статистики следует выполнить команду `python influx2_cli.py create_usage_task --luna-config http://127.0.0.1:5070/1` после запуска сервиса Admin. Соответствующая информация добавлена в инструкцию по установке. Команда автоматически создаёт необходимый бакет «luna\_monitoring\_aggregated». Если данная команда не выполнена, то в ответе «/luna\_sys\_info» не будет отображаться статистика.

При необходимости можно отключить сбор статистики, удалив или отключив соответствующие задачи на вкладке «Tasks» в GUI InfluxDB.

Подробную информацию «Подсчет статистики выполненных запросов и оценок» в руководстве администратора.

- Прекращена поддержка InfluxDB версии 1. Параметры для этой версии были удалены из конфигурационных файлов. Теперь для мониторинга работы LP используется InfluxDB 2, начиная с версии 2.0.8.

Если ранее использовался InfluxDB 1, то можно выполнить миграцию на новую версию. Описание миграции дано на официальном сайте InfluxDB:

<https://docs.influxdata.com/influxdb/v2.0/upgrade/v1-to-v2/docker/>



- В ресурс [«/sdk»](#) добавлена фильтрация по состояниям Liveness (spoof, real, unknown).
- Внесены изменения в лицензирование.

Для Liveness V2 добавлена возможность лицензировать количество выполненных транзакций. Теперь при лицензировании Liveness V2 можно выбрать между безлимитной лицензией и лицензией с ограниченным числом транзакций.

Каждое использование Liveness в запросах уменьшает счётчик транзакций. После исчерпания лимита транзакций будет невозможно использовать оценку Liveness в запросах. На запросы не использующие Liveness или с отключённой оценкой Liveness исчерпание лимита не влияет, они продолжают работать в обычном режиме.

Утилита HASP обновлена до версии 8.23. При обновлении на новую сборку LP требуется удалить старую утилиту HASP и установить новую. Сервис Licenses не запустится, если использована утилита предыдущих версий.

При использовании Liveness V2 с транзакциями потребуются обновить лицензию при переходе на новую сборку, т. к. ранее выданные лицензии не поддерживают данную возможность.

Информация о текущих параметрах лицензии отображается в ответе на запрос GET на ресурс [«/license»](#) сервиса Licenses, в ответе на запрос в ресурс [«/luna\\_sys\\_info»](#) сервиса Admin и в GUI сервиса Admin на вкладке «help».

Подробную информацию можно найти в разделе «Информация о лицензии» в руководстве администратора.

- Сервис Licenses больше не зависит от других сервисов LP, кроме Configurator (при получении настроек из него), и может запускаться независимо от них. Но сервисы LP, например Faces, теперь зависят от Licenses. В связи с этим был изменён порядок запуска сервисов в документации по установке.

Сервисы теперь сами пишут данные о лицензии в базу мониторинга:

- Сервис Faces пишет данные о созданных лицах с прикрепленными биометрическими шаблонами в базу мониторинга в поле `license_faces_limit_rate`.
- Сервис Handlers пишет данные о количестве доступных транзакций Liveness V2 в базу мониторинга в поле `liveness_balance`.

Предупреждение об исчерпании количества доступных транзакций отправляется в мониторинг и логи сервиса при достижении 2000 оставшихся транзакций Liveness V2 (данный порог задан в системе).

- Сервис Licenses пишет данные о сроке истечения лицензии в базу мониторинга в поле `license_period_rest`. Данное поведение использовалось и в предыдущих сборках.

Подробную информацию можно найти в разделах «Информация о лицензии» и «Сервис Licenses» в руководстве администратора.

### Исправленные ошибки

- Ускорен запрос на получение лиц с атрибутами в сервисе Faces. Ранее сервис Licenses мог некорректно обрабатывать значение параметра FacesLimit на больших базах данных.
- В сервисе Index Manager исправлена проблема, когда не перестраивался индекс. Индекс был удалён вручную или не был построен из-за падения сервиса, но был указан в базе данных как существующий, поэтому система не перестраивала его.

Теперь в этих случаях статус списков с отсутствующим индексом будет выставляться как failed. Далее задача на построение индекса будет добавлена в планировщик, и индекс будет построен.

## LUNA PLATFORM v.5.16.0

### Изменения

- В события добавлено поле «top\_similar\_external\_id», содержащее «external\_id» наиболее схожего кандидата (события или лица), с которым выполнялось сравнение лица.

В ответах на следующие запросы в объект «top\_match» событий было добавлено поле «external\_id», содержащее «external\_id» кандидатов для сравнения:

- «generate events».
- «get event».
- «ws handshake».

Данные поля позволяют сразу получать «external\_id» наиболее похожих лиц или событий после выполнения запросов. Например, ранее событие не содержало «external\_id» наиболее схожего лица ни в ответе на запрос, ни в базе данных событий. Требовалось выполнить дополнительный запрос для определения «external\_id» лица по его «face\_id».

В перечисленные далее запросы добавлены соответствующие фильтры для сортировки событий по «external\_id» наиболее похожего объекта:

- В политику «match\_policy» для событий добавлен фильтр «top\_similar\_external\_ids» (см. описание политики в запросе «create handler»).
- Добавлена фильтрация по значению параметра «top\_similar\_external\_ids» для получения событий в сервисах API и Events (см. запрос «get events»).
- В запрос получения статистики по событиям добавлены target и фильтр «top\_similar\_external\_id» в сервисах API и Events (см. запрос «get statistics on events»).

- Фильтр «top\_similar\_external\_ids» добавлен при указании событий в качестве кандидатов для запросов «match faces» и «match bodies».
- Фильтр «top\_similar\_external\_ids» добавлен в фильтры событий для задач:
  - \* [прикрепления к списку](#).
  - \* [кластеризации](#).
  - \* [перекрестного сравнения](#).
  - \* [экспорта данных](#).

Колонка «top\_match» в результатах [задачи создания отчёта](#) и [задачи экспорта данных](#) теперь содержит «external\_id» лучшего кандидата сравнения.

- Для кандидатов в запросах «[matching faces](#)» и «[human body matching](#)» добавлен параметр «order», определяющий порядок сортировки результатов. Доступна сортировка в порядке возрастания времени создания события (опция «create\_time\_asc»), в порядке убывания времени создания события (опция «create\_time\_desc») и по схожести кандидатов (опция «similarity»).

Использование новых параметров сортировки позволяет, например, получить события и лица в порядке, в котором они приходили в систему. Таким образом можно определить первое появление объекта.

Для корректной работы параметров сортировки по времени требуется задать порог по степени схожести объектов. При указании новых параметров сортировки используется матчинг по базе данных. Скорость обработки запроса будет ниже, чем при матчинге по списку с сортировкой по «similarity».

По умолчанию используется опция «similarity».

- В [задачу кластеризации](#) добавлен параметр «limit», позволяющий задать максимальное количество кандидатов, возвращаемых для каждого сравнения. Значение по умолчанию изменено с 5 на 20 000.

Параметр управляет количеством кандидатов, возвращаемых в ответе. Чем больше кандидатов возвращается, тем лучше. Но при большом размере кластера могут возникать проблемы с производительностью и точностью.

## Исправленные ошибки

- Исправлена ошибка, при которой в результатах запроса в Backport 4 на сравнение по индексированному списку не возвращались указанные параметры «targets», а в качестве «targets» возвращался набор полей по умолчанию. Проблема возникала в сервисе Python Matcher Proxy при выполнении сравнения одного эталона с несколькими кандидатами.
- Изменено поведение при загрузке изображений. Теперь каждое изображение загружается независимо. В случае возникновения ошибки при загрузке одного из изображений загрузка других изображений не прерывается.

- Поле «result» в ответе на запрос [«get task result»](#) для задачи пакетной обработки больше не является обязательным в сервисе Tasks.

Если при выполнении одной подзадачи возникает ошибка, то в теле ответа вернётся только поле «errors» с ошибкой. Ранее вместе с «errors» возвращалось пустое поле «result».

- В сервисе Tasks исправлены ограничения для разметки [задачи расчёта ROC-кривой](#). Теперь максимально можно задать 20 000 элементов.

Ранее при передаче более 20 000 элементов, в логах сервиса Tasks Worker возвращалась ошибка «Uncaught exception occurred». Теперь возвращается ошибка с уровнем «Error» и описание ошибки.

- Из настроек сервиса Configurator удалена неиспользуемая настройка «CLUSTERING\_MATCH\_LIMIT».
- Исправлена информация, возвращаемая в секции «LUNA\_CONFIGURATOR» в ответе на запрос [«get service configuration»](#). Теперь она содержит текущее состояние параметра.
- В документе «EventsReferenceManual.html» в описании для перечисленных полей в схеме запроса [«create new events»](#) теперь указано, что поля не являются «Nullable»:

- matches/candidates/face/external\_id
- matches/candidates/face/user\_data
- matches/candidates/event/external\_id
- matches/candidates/event/handler\_id
- matches/candidates/event/user\_data

- В документе «EventsReferenceManual.html» в описании для перечисленных полей в схемах запросов [«get event»](#) и [«get events»](#) теперь указано, что поля являются «Nullable»:

- top\_match/event\_id
- top\_match/face\_id
- match\_result/candidates/event/event\_id
- match\_result/candidates/event/create\_time

- В документе «EventsReferenceManual.html» в описании для перечисленных полей в схемах запросов [«face matching»](#) и [«human body matching»](#) теперь указано, что поля являются «Nullable»:

- matches/result/event/top\_match/event\_id
- matches/result/event/top\_match/face\_id

- В документе «SenderReferenceManual.html» в описании для перечисленных полей в схеме запроса [«ws handshake»](#) теперь указано, что поля являются «Nullable»:

- event/matches/candidates/event/match\_result/candidates/event/external\_id
- event/matches/candidates/event/match\_result/candidates/event/user\_data

- event/matches/candidates/event/match\_result/candidates/event/top\_match/face/face\_id
- event/matches/candidates/event/match\_result/candidates/event/top\_match/event/event\_id
- В документе «HandlersReferenceManual.html» в описании для указанного поля в схеме запроса [«generate events»](#) теперь указано, что поле является «Nullable»:
  - events/matches/candidates/event/match\_result/candidates/event/user\_data

## LUNA PLATFORM v.5.15.0

### Изменения

- Добавлена поддержка динамического обработчика для задачи пакетной обработки (см. запрос [«/tasks/estimator»](#)).

Ранее ресурс поддерживал работу только со статическими обработчиками. Теперь можно указать ID статического или динамического обработчика в поле `handler_id`.

Для динамического обработчика можно указать политики для обработки прямо в запросе. Один и тот же динамический обработчик можно указывать в разных задачах `estimator` и не создавать для каждой задачи новый обработчик.

Если указать ID статического обработчика и попытаться указать в запросе политики для него, то вернётся ошибка.

- Во все сервисы LP добавлен новый ресурс [«/configs»](#). Он позволяет получить используемые настройки соответствующего сервиса.

Если сервис запущен с опцией автоматического обновления настроек, то в случае их обновления ресурс вернёт актуальные значения, а не те настройки, с которыми сервис был запущен изначально.

Для получения списка настроек необходимо выставить заголовок Ассерта, принимающий значения `application/json` (настройки будут возвращены в формате JSON для Configurator) или `text/plain` (настройки будут возвращены в формате конфигурационного файла `config.conf`).

Пароли и токены, указанные в конфигурационных файлах, будут скрыты в полученном ответе.

- Добавлен новый ресурс [«/handlers/validator»](#). Ресурс позволяет проверить корректность указанных политик обработчика.

Ресурс может быть полезен для проверки корректности динамических обработчиков.

- Добавлен Docker-контейнер с Grafana, который дополнительно содержит все необходимые скрипты создания дашбордов LUNA PLATFORM. Описание запуска контейнера и создания дашбордов приведено в документации по ручному запуску контейнеров LUNA PLATFORM.
- В документацию добавлена инструкция для использования GPU при запуске LUNA PLATFORM с помощью Docker Compose.

### Исправленные ошибки

- Исправлена ошибка, когда при генерации событий возвращалась ошибка с кодом 500. Ошибка возвращалась в случае задания тега без указания фильтров в политике `conditional_tags_policy` (см. запрос [«/handlers/{handler\\_id}/events»](#)).
- Исправлено описание параметра `page_size` в запросе [«get events»](#) в документации OpenAPI. Сейчас указано, что количество объектов не может превышать 1000. Ранее было указано, что количество объектов на странице не может превышать 100.

## LUNA PLATFORM v.5.14.0

### Изменения

- SDK обновлён до версии 5.4.1.
  - Обновлён алгоритм Liveness V2. Пороги по умолчанию для `liveness_threshold` и `quality_threshold` теперь равняются «0.5». Рекомендуемый порог `liveness_threshold` теперь равен «0.5» вместо «0.88».
  - Исправлена проблема, когда SDK возвращал ошибки «Invalid detection» и «Invalid image size» в случае получения неправильного ограничивающего прямоугольника. Теперь в этих случаях возвращается ошибка «Invalid rectangle».

- В сервис Python Matcher Proxy добавлена поддержка плагинов сравнения.

Использование плагинов может значительно ускорить выполнение запросов на сравнение. Например, с помощью плагинов возможно организовать хранение необходимых для выполнения операций сравнения данных и дополнительных полей объектов в отдельном хранилище, что позволит ускорить доступ к данным по сравнению с использованием стандартной БД LUNA PLATFORM.

В документации сервиса Python Matcher Proxy ([«ServiceManuals/PythonMatcherDevelopmentManual»](#)) в разделе «Plugins» приводится общее описание основных шагов для создания собственных плагинов и приводится пример плагина «Thin event».

«Thin event» используется для быстрого сравнения дескрипторов лиц с дескрипторами упрощённых событий. Упрощённые события содержат меньше полей по сравнению с событиями из БД `luna_events`. Все данные для них хранятся в одной таблице.

Требования для запуска «Thin event» приведены в его документации. По умолчанию плагин не используется.

Важно, что плагины не предоставляются как готовое решение для матчинга. Требуется дополнительно реализовать логику, необходимую для решения конкретных бизнес-задач.

Для работы с плагинами должен быть установлен сервис Python Matcher Proxy. Он принимает решение об использовании стандартных механизмов матчинга LP или плагинов на основе сложности запроса (cost). Активацию плагинов следует выполнять в этом сервисе.

См. общее описание плагинов сравнения в руководстве администратора в разделе «Плагины сравнения» и подробную инструкцию с примерами реализации плагинов в документации Python Matcher Proxy в разделе «Plugins».

- Добавлена возможность указывать время окончания события с помощью заголовка «Luna-Event-End-Time» в запросе «[generates events](#)». Если данный заголовок установлен, то все события будут сгенерированы с полем окончания времени события «end\_time».

Поле может быть использовано для передачи времени окончания события из внешних систем.

Поле окончания времени события «end\_time» поддерживается для:

- задания «target» событий в политике сравнения обработчика (см. запросы «[create handler](#)» и «[generate events](#)»);
- сохранения события (см. запрос «[save event](#)»);
- задания «target» при получении событий и их статистики (см. запросы «[get events](#)» и «[get statistics on events](#)»);
- события, возвращаемого с помощью веб-сокетов (см. запрос «[ws handshake](#)»). Время создания и окончания события будет возвращены в полях «event-create-time» и «event-end-time» соответственно.
- колонок в документе CSV в задачах создания отчета и экспорта (см. запросы «[reporter task](#)» и «[exporter task](#)»).

Фильтры события «end\_time\_\_gte» и «end\_time\_\_lt» доступны для:

- политики сравнения, когда события заданы как кандидаты (см. запросы «[create handler](#)» и «[generates events](#)»);
  - получения событий (см. запрос «[get events](#)»);
  - обработки задач привязки, перекрестного сравнения и кластеризации (см. запросы «[linker task](#)», «[cross-matching task](#)» и «[clustering task](#)»).
- В сервисы Events, Faces, Tasks, Configurator, Admin, Handlers, Python Matcher и Backport3 добавлена настройка «CONNECTION\_POOL\_SIZE», позволяющая задавать размер пула соединений к БД.

## Исправленные ошибки

- В документации «APIReferenceManual» исправлено неправильное название поля для указания версии дескриптора в запросах, принимающих на вход дескрипторы. Ранее поле называлось «descriptor\_version», теперь поле называется «version».

## LUNA PLATFORM v.5.13.0

### Изменения

- В ресурсе [«/sdk»](#) добавлена агрегация полученных значений статуса маски, эмоций и Liveness. При выполнении запроса с включенным параметром «aggregate\_attributes» агрегированные значения данных оценок возвращаются в поле «aggregate\_estimations» в теле ответа.
- Добавлена поддержка типа содержимого application/msgpack для запроса на создание лица с атрибутами. См. запрос [«create face»](#).

MessagePack упаковывает данные эффективнее, чем JSON. Это бинарный формат, поэтому не требует декодирования из BASE64.

### Исправленные ошибки

- Из документа OpenAPI сервиса API удалено лишнее описание создания задач удаления биометрических шаблонов указанной версии для лиц и событий. Для задачи сборки мусора в сервисе API доступно только удаление событий. См. запрос [«garbage collection task»](#).

Задача удаления биометрических шаблонов необходимой версии для лиц и событий доступна в API сервиса Admin.

- Исправлена необработанная ошибка, возникавшая при некорректном значении поля «account\_id» в политике «match\_policy» при отправке запросов POST на ресурс [«/handlers»](#) и PUT на ресурс [«/handlers/{handler\\_id}»](#) сервиса Handlers. Ошибка возникала, если «account\_id» не был задан в формате UUID.

## LUNA PLATFORM v.5.12.2

### Изменения

- В сервисе Faces исправлено описание тела запроса на создание лиц с типом содержимого application/msgpack.



## LUNA PLATFORM v.5.12.1

### Изменения

- В сервисы LUNA PLATFORM добавлен новый ресурс «/healthcheck». Ресурс может использоваться для активной проверки состояния сервиса, а именно, может ли сервис выполнять свои функции в полном объёме или нет. Проверяется возможность подключения данного сервиса к сервисам LP и БД, от которых он зависит.
  - «/healthcheck API»
  - «/healthcheck Admin»
  - «/healthcheck Handlers»
  - «/healthcheck Events»
  - «/healthcheck Backport3»
  - «/healthcheck Backport4»
  - «/healthcheck Tasks»
  - «/healthcheck Image Store»
  - «/healthcheck Python Matcher»
  - «/healthcheck Licenses»
  - «/healthcheck Faces»
  - «/healthcheck Sender»
  - «/healthcheck Configurator»

Возможно настроить периодическую проверку ресурса с использованием HAProxy, NGINX или иной системы. Это позволит определить, что сервис недоступен, и принять решение о его отключении из контура или перезапуске.

С помощью опции «include\_luna\_services» можно включать и отключать проверку healthcheck для сервисов LUNA PLATFORM, от которых зависит данный сервис. Если опция включена, то отправляются дополнительные запросы на ресурсы «/healthcheck» этих сервисов.

Опция «include\_luna\_services» отключается, чтобы не выполнять рекурсивную проверку одних и тех же сервисов. Например, когда сразу несколько сервисов, от которых зависит данный сервис, будут отправлять запросы в сервис Faces и тем самым увеличивать нагрузку на него.

При успешном выполнении проверки healthcheck возвращается только время выполнения подключения в поле «execution\_time».

При недоступности одного или нескольких сервисов возвращается ошибка с кодом 502 «Unhealthy». В теле ответа перечисляются компоненты, статусы проверок и возникшие ошибки.

Код ошибки 500 в ответе не обязательно означает проблему в работе сервиса. Долгий за-

прос может завершиться с ошибкой из-за превышения таймаутов, повышенной нагрузки на сервер, проблем в сети или по иным причинам.

При выполнении запроса на ресурс «/healthcheck» рекомендуется выставить таймаут в несколько секунд. Если запрос не успевает отработать, это признак того, что при работе системы возникли проблемы.

- При получении изображений из сервиса Image Store для выполнения запросов «[detect face](#)», «[extract attributes](#)», «[generates events](#)» и «[perform verification](#)», отправляемых напрямую в сервис Handlers, теперь используется значение параметра «account\_id». Изображения получают из Image Store, если в запросе указан ID биометрического образца или URL хранящегося в Image Store изображения.
- Описания выполнения миграций с LUNA PLATFORM 3 и LUNA PLATFORM 4 исправлены в соответствии с последними изменениями в LUNA PLATFORM 5.

## LUNA PLATFORM v.5.12.0

### Изменения

- Теперь в обработчиках (handlers) фильтры «create\_time\_\_lt» и «create\_time\_\_gte» для кандидатов (лиц и событий) матчинга в политике «match\_policy» можно задавать относительно текущего времени.

Например, данный формат фильтра позволит выбирать для сравнения только события, созданные за последний час, что может быть использовано для случаев оплаты по лицу в транспорте для исключения повторной оплаты проезда.

В этом формате время задаётся по следующему шаблону '[now-\(\d+\)](#) [[smhdwMy](#)]', где [d+] — число, [smhdwMy] — необходимый период: m (минуты), h (часы), d (дни), w (недели), M (месяцы), y (годы).

Примеры:

- Запись «create\_time\_\_gte»: «now-3h» означает, что будут выбраны все объекты, созданные за последние три часа.
- Запись «create\_time\_\_lt»: «now-4w» означает, что будут выбраны все объекты, созданные раньше, чем четыре недели назад.
- В ресурс ручного сохранения события «[handlers/{handler\\_id}/events/raw](#)» сервисов API и Handlers добавлена новая опция «wait\_saving», позволяющая включать и отключать ожидание сохранения событий в БД Events перед отправкой ответа.

При отключённой опции ответ на запрос «[handlers/{handler\\_id}/events/raw](#)» возвращается быстрее, т. к. система не будет ждать, пока событие сохранится в базе данных. Но при этом

система не присылает никаких уведомлений в случае, если не удалось выполнить сохранение.

При включённой опции система дожидается сохранения событий перед отправкой ответа. Если сохранение успешно, то будет возвращен код состояния 201. Если же по каким-то причинам событие не сохранилось, вернется код ошибки 500.

По умолчанию опция включена.

- В запрос на получение файла с текущими настройками LUNA PLATFORM «[/dump](#)» из сервиса Configurator добавлен новый параметр «version», содержащий версию миграции этих настроек. Версия изменяется при обновлении на новые сборки LUNA PLATFORM.

Не следует вручную обновлять настройки системы из файла предыдущей сборки, т. к. миграция настроек системы происходит автоматически при обновлении на новую сборку. При попытке использовать dump-файл от предыдущей сборки LP на новой сборке возникнет ошибка, т. к. версия в этом файле будет отличаться.

Файлы с настройками от предыдущей сборки рекомендуется использовать только для проверки коррекции миграции настроек.

- В OpenAPI документации сервисов Sender и API удалено повторяющееся описание сообщения WebSocket из раздела «Callback».
- Добавлена возможность указывать значение «application/msgpack» в качестве заголовка «Content-Type» для следующих запросов сервиса Events:
  - GET на «[/events](#)»
  - PATCH на «[/events/event\\_id](#)»

MessagePack упаковывает данные эффективнее, чем JSON. Это бинарный формат, поэтому не требует декодирования из BASE64.

### Исправленные ошибки

- Исправлена ошибка, при которой лицензия становилась недоступной за день до истечения её срока.
- В сервисе LUNA Events исправлена ошибка с превышением максимального количества клиентов для БД Vertica.

## LUNA PLATFORM v.5.11.0

### Изменения

- Добавлена новая задача «estimator» («[/tasks/estimator](#)»). Она позволяет выполнять пакетную обработку изображений с использованием указанных политик.

Данную задачу можно создать с помощью API сервисов API или Tasks.

В качестве результата выполнения задачи возвращается JSON с данными по каждому из обработанных изображений и информацией о возникших ошибках.

В теле запроса можно указать ID уже сохранённого обработчика или задать политики обработки вручную.

Ресурс принимает для обработки ссылку на ZIP-архив с изображениями.

В качестве ссылки на архив может использоваться внешний URL или URL до архива, сохранённого в Image Store. Во втором случае архив должен быть предварительно сохранён в LP с помощью POST запроса на ресурс [«/objects»](#)

Архив может быть защищён паролем. Пароль можно передать в запросе с помощью параметра «authorization» -> «password».

Для получения корректных результатов обработки следует использовать изображения одного типа (исходное изображение, биометрический образец лица, биометрический образец тела). Тип передаваемых изображений задаётся в запросе в параметре «image\_type».

- В задачу удаления событий ([«/tasks/gc»](#)) добавлен параметр «remove\_samples», при включении которого вместе с событиями удаляются биометрические образцы лиц и тел.

Важно, что после удаления биометрических образцов события будет невозможно повторно извлечь базовые атрибуты и биометрический шаблон для этого события.

Данный параметр добавлен в ресурсы сервисов LUNA API, LUNA Admin и LUNA Tasks, а также вынесен в пользовательский интерфейс сервиса LUNA Admin.

- При отправке запросов другим сервисам сервис API теперь может отправлять заголовок accept-encoding с директивами «identity», «deflate», «gzip» (ранее можно было указывать только директивы «deflate» и «gzip»). Директива «identity» позволяет отключить автоматическое сжатие (на стороне других сервисов) и автоматическую распаковку (на стороне сервиса API) для тел запросов с изображениями или ZIP-архивами.
- Теперь сервис API повторно использует подключения к другим сервисам платформы. Это изменение уменьшает количество открытых соединений и ускоряет запросы к другим сервисам.
- В сервисе Events изменен ответ при удалении событий. Ранее ID удалённых биометрических образцов для лиц и тел возвращались в одном массиве — «samples». Теперь в ответе возвращаются два массива — «face\_samples» и «body\_samples». См. ресурс [«event deletion»](#).
- В скрипт миграции с LP 3 «start\_migration.py» добавлен параметр *skip\_missing\_descriptors*. Параметр позволяет игнорировать отсутствующие в БД LP 3 биометрические шаблоны при выполнении миграции. Информацию о скрипте можно найти в главе «Migration launch» руководства «LP\_Migration\_from\_LP3».

## Исправленные ошибки

- Внесены изменения в обработку параметра «external\_ids» для ресурса «[/verifiers/{verifier\\_id}/verifications](#)». Ранее значение данного параметра можно было указать только в формате «uuid», а теперь используется формат «string».
- Исправлена ошибка в сервисе Backport3, которая возникала при автоматическом обновлении настроек сервиса и останавливала его работу.

## LUNA PLATFORM v.5.10.0

### Изменения

- В набор политик «storage\_policy» ресурса «/handlers» была добавлена политика отправки уведомления о создании событий «notification\_policy». В ней можно включить или отключить отправку уведомлений о создании событий в сервис Sender, а также настроить фильтры для отправки уведомлений.

По умолчанию отправка уведомлений включена.

- Группа параметров «LUNA\_HANDLERS\_LIMITS» добавлена в настройки сервиса Handlers.

Параметры позволяют задать:

- Максимальное количество изображений, получаемых в запросе.
- Максимальное количество детекций в событии, создаваемом вручную.
- Максимальный размер массивов в событии, создаваемом вручную.
- Максимальное количество кандидатов в результатах сравнения.

Ранее данные значения можно было изменить только с помощью специального конфигурационного файла в комплекте поставки.

Следует учесть, что увеличение лимитов может привести к проблемам в работе LP.

- Теперь при включённом параметре «aggregate\_attributes» в запросе «[/handlers/{handler\\_id}/events](#)» выполняется агрегация всех полученных значений для параметров «mask» и «emotions». Агрегированные значения будут возвращаться в ответе в полях «mask» и «emotions» в группе параметров «aggregate\_estimations».

Значения данных параметров, определённые для каждой отдельной детекции, тоже возвращаются в ответе в группе «detection».

Агрегированные значения этих параметров можно указать в поле «aggregate\_estimations» при ручном создании события с помощью ресурса «[/handles/{handler\\_id}/events/raw](#)».

Агрегированные значения параметров возвращаются при получении уведомлений о созданных событиях через веб-сокеты с помощью сервиса Sender.

- SDK был обновлён до версии 5.3.0.

Значение параметра «redetect\_score\_threshold» было задано равным «0.3»

- В руководство администратора в раздел «Плагины» добавлено общее описание работы с плагинами LP и базовая информация по их добавлению.

### **Исправленные ошибки**

- Добавлена поддержка работы с «обрезанными» JPEG изображениями в сервисе Handlers. У таких изображений отсутствуют последние байты, обозначающие окончание изображения. Ранее при обработке таких изображений возвращалась внутренняя ошибка.
- В GUI сервиса Admin исправлена ошибка с отображением фильтров для задачи GC. При выборе объекта faces отображался лишний фильтр «descriptor type».
- Исправлена ошибка с запросом POST на ресурс «/faces/attributes/descriptors» в сервисе Faces. Сервис возвращал пустой список лиц, если в нём находилось лицо с биометрическим шаблоном, версия которого равнялась версии, указанной в поле «missing\_version».
- Исправлена ошибка с проверкой версии биометрических шаблонов при удалении биометрических шаблонов в сервисе Events. Теперь версия биометрических шаблонов не будет проверяться при удалении, что позволит удалять любые указанные версии биометрических шаблонов, в том числе те, которые уже не поддерживаются системой.
- Описание ошибки при получении объекта с неправильным Accept header было исправлено для сервиса Image Store.
- Исправлена схема ответа для ресурса «luna\_sys\_info» сервиса Admin.
- Исправлено падение при автоматическом обновлении настроек сервиса Handlers. Падение возникало при изменении настроек для данного сервиса в сервисе Configurator.
- В документацию сервиса Handlers добавлено отсутствовавшее описание некоторых полей мониторинга.
- Для сервиса Handlers исправлена проверка поля «mimetype» для запроса создания событий. Больше не возвращается «Internal server error», если указанный в запросе тип изображения отличается от типа передаваемого в запросе изображения.

## **LUNA PLATFORM v.5.9.0**

### **Изменения**

- В поставку добавлены дашборды с информацией о запросах LP и возникающих ошибках.  
Для отображения дашбордов используется Grafana. Доступ к GUI Grafana осуществляется по порту 3000.

Для создания дашбордов требуется InfluxDB версии 2. Дашборды не работают с версией 1. Теперь InfluxDB 2 версии используется по умолчанию при установке LP. InfluxDB 1 все ещё поддерживается, но рекомендуется использовать версию 2.

При установке для подключения к InfluxDB 2 используются token и password, заданные по умолчанию. Измените их при необходимости.

В инструкции по установке добавлены описания запуска контейнеров Grafana и InfluxDB 2, запуска скрипта создания дашбордов.

Дополнительную информацию о мониторинге и дашбордах можно найти в разделе «Мониторинг» в руководстве администратора.

- Добавлены ресурсы для сохранения и получения объектов из Image Store: «/objects» и «/objects/{object\_id}».

Доступны следующие типы содержимого запроса:

- «application/json»
- «application/pdf»
- «application/zip»
- «text/plain»

Ресурс «/objects» позволяет сохранить объект одного из перечисленных форматов в Image Store под уникальным ID.

Ресурс «/objects/{object\_id}» позволяет:

- получить объект.
- удалить объект.
- проверить, существует ли объект с указанным ID.

- Добавлена возможность выполнять задачу GC для удаления биометрических шаблонов лиц и тел для событий.

В качестве «target» теперь можно указать «event\_descriptors». Далее следует указать, какой тип биометрических шаблонов следует удалить для события (face, body) и их версию.

Важно, что было внесено изменение в тело запроса создания задачи GC «/tasks/gc». Теперь для удаления биометрических шаблонов лиц следует указывать «face\_descriptors» вместо «descriptors» в качестве «target».

В GUI Admin добавлена возможность запуска задачи удаления биометрических шаблонов событий.

В сервис Events добавлен ресурс «/events/descriptors», позволяющий удалять биометрические шаблоны, прикрепленные к событию.

- В настройки сервисов Tasks и Python Matcher добавлена группа параметров «PLATFORM\_LIMITS». С их помощью для запросов «/matcher/face», «/matcher/body», «/tasks/clustering» и «/tasks/cross\_match» можно настроить максимальное количество:
  - эталонов и кандидатов в запросе
  - передаваемых значений для некоторых фильтров
  - возвращаемых в ответе кандидатов

Ранее данные значения можно было изменить только с помощью специального конфигурационного файла в комплекте поставки.

- В сервисах Python Matcher, Admin и Index Manager теперь используется Python 3.9. Поддержка более старых версий Python для сервисов прекращена.
- Список зависимостей для выполнения скрипта миграции с LUNA PLATFORM 3 на LUNA PLATFORM 5 был вынесен в отдельный requirements.txt файл в контейнере Backport 3. Файл может использоваться при необходимости выполнять миграцию вне контейнера Backport 3.
- Добавлена возможность указывать атрибуты в качестве кандидатов для верификации в ресурсе «/verifiers/{verifier\_id}/verifications».

Атрибут имеет ограниченный срок существования и будет удалён по его завершении, поэтому верификация через некоторое время будет невозможна.

Например, это позволит выписать пропуск на 24 часа (задать соответствующий TTL атрибута) и пропускать человека только в течение этого периода.

- Добавлена следующая информация в OpenAPI документацию сервиса API:
  - в документацию в описание ресурса «/tasks/{task\_id}/result» возвращены примеры результатов выполнения задач сервиса Tasks.
  - в документацию в описание ответа ресурса «/tasks/{task\_id}/result» добавлен пример результата выполнения задачи «exporter». В описании ответа ресурса необходимо выбрать response schema «application/zip».

### Исправленные ошибки

- Исправлена ошибка, когда задача «Exporter» выполнялась без учёта разделителя, указанного в запросе в параметре «csv\_delimiter».
- ID событий без атрибутов теперь возвращаются в упорядоченном виде в ответе ресурса «/events/attributes/missing» сервиса Events.
- В сервисе Events исправлена ошибка, когда в ответе на запрос DELETE в ресурс «/events» возвращались sample ID со значением «Null».
- Исправлена ошибка при смене пароля в сервисе Admin при отправке запроса PATCH на ресурс «/login».



- Для сервиса Events исправлена ошибка с попаданием в лог сервиса следующего сообщения `WARNING: sanic.root: Message body set in response on /2/events. A 204 response may only have headers, no body..` Сервис больше не возвращает тело запроса при указании `gzip` или `deflate` в качестве `accept-encoding` при статусе 204 или методе `HEAD`.
- В документации обновлена схема базы данных для сервиса Events.
- Сервис Backport 3:
  - Статус код для успешного ответа на `POST «/handlers/verify/raw»` был обновлён.
  - Была добавлена схема ответа на запрос `PATCH «/storage/persons/{person_id}»` для статуса 400.
  - Обновлена схема ответа для запроса `GET` в ресурс `«/version»`.
  - Для полей персоны `«user_data»` и `«external_id»` значения по умолчанию были заменены на пустые строки.

## LUNA PLATFORM v.5.8.0

### Изменения

- В политику `«detect_policy»` для ресурсов `«/handlers»` и `«/verifiers»` добавлены параметры для работы с Liveness V2. При включении определения Liveness следует учитывать требования Liveness V2 к входящим изображениям. См. «Требования Liveness V2» для получения более подробной информации.

При отсутствии лицензии для Liveness V2 в ответе на запросы `«/handlers/{handler_id}/events»` и `«/verifiers/{verifier_id}/verifications»` вернётся ошибка `«License problem: 'Liveness v.2 feature disabled'»`. Поэтому при отсутствии лицензии значение `«estimate_liveness > estimate»` должно равняться 0.

Для обработчиков сервиса Backport 4 возможность определения Liveness недоступна.

#### Liveness в обработчиках (handlers и verifiers)

Группа параметров `«estimate_liveness»` позволяет включить определение Liveness и статуса входящего изображения:

- `«estimate»` — включить определение Liveness на входящих изображениях. По умолчанию значение равно `«0»`, и Liveness V2 не используется.
- `«liveness_threshold»` — задать порог Liveness. Лицо на входящем изображении будет считаться настоящим (статус `«real»`) только если значение Liveness равно или превышает заданный порог. В противном случае лицо будет считаться фейком (статус `«spoof»`). Значение по умолчанию: `«0.88»`.

- «quality\_threshold» — задать порог качества изображения для оценки Liveness. Если качество входящего ниже указанного порога, то событию будет присвоен статус «unknown». Значение по умолчанию: «0».

Параметр «liveness\_states» позволяет задать фильтр по статусам Liveness. Обработаны будут только те события, статусы которых соответствуют одному из указанных в данном параметре статусу.

Если событие отфильтровано по «liveness\_states», то для его детекции лица в ответе возвращаются все определённые свойства лица, но последующие политики «extract\_policy», «match\_policy», «storage\_policy», «conditional\_tags\_policy» не будут выполнены для данного события.

В фильтре «liveness\_states» можно указать одно или несколько значений из следующего списка:

- spoof — 0
- real — 1
- unknown — 2

По умолчанию значение параметра равняется «null», фильтрация по нему не выполняется. Если использование Liveness выключено в «estimate\_liveness > estimate», то поле игнорируется.

Фильтр «liveness» добавлен в следующие политики:

- «match\_policy».
- «storage\_policy» для всех создаваемых объектов.
- «conditional\_tags\_policy».

Поле «liveness» добавлено в политику «match\_policy» в качестве возможного значения поля target.

### **Агрегация результатов Liveness**

При включённой в запросе «/handlers/{handler\_id}/events» опции «aggregate\_attributes» для нескольких переданных изображений будет выполнена агрегация результатов проверки Liveness для получения более точных данных. Эти данные будут возвращены в поле «aggregate\_estimations».

Поле «aggregate\_estimations» является обязательным и всегда возвращается в ответе на данный запрос. При этом в ответе также возвращаются значения Liveness для каждой из детекций. При отключённой опции «aggregate\_attributes» значения Liveness для детекции лица и значения в поле «aggregate\_estimations» будут одинаковыми.

При сохранении события в БД Events будет записан агрегированный результат проверки Liveness из поля «aggregate\_estimations».

## Новое поле «liveness»

Поле «liveness» было добавлено в события. Это позволяет:

- Получать события, используя фильтр «liveness», и указывать «liveness» в качестве «target» в запросе GET [«/events»](#).
- Получать статистику по событиям и указывать «liveness» в качестве target или фильтра в запросе GET [«/events/statistics»](#).
- Использовать «liveness» в качестве фильтра для событий в запросе в ресурс [«/ws»](#) и получать значения «liveness» в ответе данного ресурса.

Поле «liveness» было поддержано в качестве:

- фильтра для событий для запросов сравнения [«/matcher/face»](#) и [«/matcher/body»](#).
- фильтра для событий в задачах перекрестного сравнения ([«/tasks/cross\\_match»](#)), кластеризации ([«/tasks/clustering»](#)), и привязки ([«/tasks/linker»](#)).
- столбца для отчёта в задаче создания отчета ([«/tasks/reporter»](#)).

Значения параметров «liveness» можно задать при ручном создании события с помощью ресурса [«/handles/{handler\\_id}/events/raw»](#). Возможно указать как отдельные значения Liveness для каждой детекции лица, так и агрегированное значение.

- Добавлена возможность повторно извлекать базовые атрибуты и биометрические шаблоны лиц и тел для событий в запросе [«/tasks/additional\\_extract»](#) в сервисе Admin. Теперь возможен переход на новую версию нейросети для событий.

В задачу [«/tasks/additional\\_extract»](#) добавлены:

- Возможность задавать события в качестве объектов для обновления биометрических шаблонов лиц и базовых атрибутов.
- Возможность повторно извлекать биометрические шаблоны тел для событий.

В запросе в задаче повторного извлечения биометрических шаблонов следует указать извлечение базовых атрибутов или биометрических шаблонов (лиц или тел). После этого для извлечения биометрических шаблонов в фильтрах следует указать, для каких объектов (лиц или событий) следует извлечь биометрические шаблоны.

В пользовательский интерфейс сервиса Admin добавлена возможность повторно извлекать атрибуты событий. Для этого следует выбрать «Events» в качестве «Object type» и указать «Face» или «Body» в качестве «Descriptor type».

В сервисе Events добавлен запрос PATCH в ресурс [«/events/{event\\_id}»](#) для обновления базовых атрибутов и биометрических шаблонов существующих событий.

В сервис Events добавлены ресурсы для получения событий с биометрическими образцами и отсутствующими атрибутами [«/events/attributes/missing»](#) и для подсчёта количества таких событий [«/events/attributes/missing/count»](#).

См. дополнительную информацию в «Запуск задачи повторного извлечения» в документе LP\_Administration\_Manual.

- В сервисах API, Faces, Image Store, Tasks, Events, Configurator, Sender, Handles, Backport 3, Backport 4, Licenses теперь используется Python 3.9. Поддержка более старых версий Python прекращена.

Для работы клиентской библиотеки «luna3», скриптов миграции с LP 3 и скрипта загрузки изображений «folder\_uploader.py» доступно использование Python 3.7 и более новых версий.

- В POST запрос в ресурс [«/handlers/{handler\\_id}/events»](#) для схем «application/json» и «multipart/form-data» добавлено поле «image\_origin».

В этом поле можно указать ссылку на исходное изображение для каждого из передаваемых в запросе изображений. Переданный URL будет добавлен в поле «image\_origin» созданного события.

Изображение, переданное в поле «image\_origin», не будет обработано в запросе «/handlers/{handler\_id}/events», оно используется только в качестве исходного изображения.

Если поле «image\_origin» в запросе не пустое, то указанная в нём ссылка будет использована в создаваемом событии независимо от настроек в политике «image\_origin\_policy».

См. дополнительную информацию о сохранении исходных изображений в событиях в разделе «Особенности сохранения исходных изображений» в LP\_Administration\_Manual.

- Добавлена поддержка символов UTF8 в тегах EXIF. Добавлено распознавание кириллицы.

Символы в ответе ограничены множеством ASCII. Для всех символов в тегах EXIF, которые не входят в это множество, используется экранирование.

Например, экранированные символы для тега «artist» будут выглядеть следующим образом:

[illegible]

## Исправленные ошибки

- В OpenAPI документации сервиса Image Store были добавлены отсутствовавшие схемы ответов..
- Исправлена ошибка, когда некоторые EXIF теги изображений приводили к «Internal server error».

## LUNA PLATFORM v.5.7.0

## Изменения

- Добавлен новый ресурс [«/tasks/exporter»](/tasks/exporter), с помощью можно выгружать данные событий и лиц из LP и сохранять их в CSV файл.

На вход принимается набор фильтров для определения объектов, которые необходимо выгрузить в файл. На выходе возвращается ZIP архив, в котором будет храниться CSV файл с данными об объектах и изображения для каждого объекта (опционально).

Возможность поддержана в сервисах API и Tasks.

См. раздел «Задача экспорта» в руководстве администратора.

- Добавлена опция для использования внешнего URL изображения при сохранении исходного изображения.

В политику сохранения исходного изображения добавлена настройка `use_external_references`. Она позволяет включить сохранение ссылки на внешнее изображение в поле `«image_origin»`, чтобы избежать дублирования одного и того же изображения в базе данных.

- Если в запросе передан внешний URL на изображение, то в поле `«image_origin»` будет указан этот URL.
- Если в запросе передан биометрический образец и он был сохранён в хранилище Image Store, то в поле `«image_origin»` будет указана ссылка на него.

Если длина URL превышает 256 символов, то вместо него будет сохранено само исходное изображение.

Возможность поддержана в сервисах API и Handlers.

- Добавлена возможность передавать время детекции вместе с изображениями.

В запрос создания событий в ресурс [«/handlers/{handler\\_id}/events»](/handlers/{handler_id}/events) добавлена возможность явного указания времени детекции лица для каждого из передаваемых изображений. Доступно для схем запроса `multipart/form-data` и `application/json`.

Данная возможность требуется для случаев, когда отправка изображений выполняется не сразу, а через некоторое время.

- Для сервисов API, Configurator, Backport 4 был изменён механизм перезагрузки настроек. Теперь перезагрузка выполняется путём перезапуска соответствующих процессов. Механизм обеспечивает более надёжное обновление настроек сервисов.

Следует учесть, что запросы, выполняемые в момент изменения настроек, могут закончиться с ошибкой. Сервис может быть недоступен некоторое время.

- При указании несуществующего `list_id` в задаче кластеризации теперь возвращается две ошибки, вместо одной: `«List with id {ID} not found»` и `«Objects for clustering not found (empty set)»`.

## Исправленные ошибки

- Теперь при провале проверки соединения сервисов LP с InfluxDB возвращается сообщение «Check connection to Influxdb: connection refused». Ранее вместо ошибки возвращался список вызовов до возникновения ошибки.
- Исправлены замедления работы в сервисе Image Store.
- В сервисе Handlers исправлена ошибка, когда не обрабатывалось значение параметра `face_bounding_boxes`, передаваемое в запросе в ресурс [«/verifiers/{verifier\\_id}/verifications»](#).
- В сервисе Handlers исправлен формат URL аватара лица, создаваемого с помощью обработчика (handler). Ранее он возвращался в формате `«/samples/{sample_id}»`. Теперь URL возвращается в формате `«/samples/faces/{sample_id}»`.
- В спецификацию OpenAPI сервиса Backport 4 добавлены поля `user_data` и `external_id`, которые возвращаются в ответе создания событий с помощью ресурса [«/handlers/{handler\\_id}/events»](#).

## LUNA PLATFORM v.5.6.0

### Изменения

- SDK был обновлён на версию 5.2.0.
- Добавлена поддержка 59 сети для извлечения биометрических шаблонов лиц.

Начиная со сборки 5.6.0, для новых установок LUNA PLATFORM по умолчанию используется 59 версия сети извлечения биометрических шаблонов.

Если LUNA PLATFORM уже установлена, то сеть не будет автоматически обновлена на 59. В Configurator останется та сеть, которую уже использует LP. В этом случае для перехода на новую версии нейросети следует выполнить повторное извлечение уже существующих биометрических шаблонов для лиц (faces). Описание данного процесса дано в разделе «Переключение версии нейросети» руководства администратора.

Важно, что повторное извлечение биометрических шаблонов для событий (events) в данный момент невозможно. При переходе на новую версию сети существующие события не могут быть использованы для сравнения.

Поставка с сервисами индексирования и поиску по индексу не поддерживает 59 версию нейросети, там по умолчанию используется 56 версия.

- Добавлена поддержка 102, 103 и 104 сетей для извлечения биометрических шаблонов тел. Теперь по умолчанию используется 104 сеть для извлечения биометрических шаблонов тела.

Сеть версии 101, которая была использована в предыдущих релизах, больше не поставляется и не поддерживается. При переходе на сборку LP 5.6.0 существующие биометрические шаблоны тел не могут быть повторно извлечены с использованием 104 версии сети и не смогут быть использованы для сравнения.

- Значения полей «user\_data» и «external\_id» в запросах теперь заданы равными «по умолчанию для всех сервисов и не могут иметь значение null».
- Фильтры «event\_id\_\_gte» и «event\_id\_\_lt» были добавлены в [«задачу перекрестного сравнения»](#), [«задачу кластеризации»](#) и [«задачу привязки»](#). Фильтры позволяют задать верхнюю и нижнюю границы значений «event\_id». Последующая обработка будет выполнена для событий, ID которых входят в заданный фильтром диапазон. Например, эти фильтры можно использовать, чтобы разделить обработку большого событий на несколько задач и выполнять эти задачи параллельно.
- Для сервисов Image Store, Tasks, Events, Admin и Backport 3 был изменён механизм перезагрузки настроек. Теперь перезагрузка выполняется путём перезапуска соответствующих процессов. Механизм обеспечивает более надёжное обновление настроек сервисов.

Следует учесть, что запросы, выполняемые в момент изменения настроек, могут закончиться с ошибкой. Сервис может быть недоступен некоторое время.

### Исправленные ошибки

- В сервисе API исправлена ошибка, когда запись об обработке запроса не добавлялась в лог в случае отключения клиента. Теперь при отключении клиента до получения ответа от него в лог будет записана ошибка со статус кодом 499.
- В ответы сервисов добавлены новые статус коды:
  - Статус код 408 возвращается, если за указанный период сервис не получил полный текст запроса (по умолчанию, 60 секунд).
  - Статус код 501 возвращается, если сервис не выполнил обработку запроса за указанный период (по умолчанию, 600 секунд).
  - Статус код 413 возвращается, если нагрузка на сервис превышает допустимую максимальную нагрузку.
- В спецификации OpenAPI для сервиса API исправлена ошибка с указанием типа SDK биометрического шаблона. Ранее SDK биометрический шаблон отображался как тип object с полями «descriptor» и «version». Теперь тип данного объекта указан как «string <byte>».

## LUNA PLATFORM v.5.5.0

### Изменения

- В сервисы API и Handlers новый ресурс [«/handles/{handler\\_id}/events/raw»](#) для ручного создания событий (объект event в LP). Поля события заполняются при отправке запроса.

Формат создаваемого события аналогичен формату, возвращаемым ресурсом [«/handlers/{handler\\_id}/events»](#). При создании события не указываются «event\_id» и «url», они возвращаются в ответе после создания события.

Для создаваемых вручную событий создаются оповещения с использованием веб-сокетов.

Ресурс позволяет задать свою логику заполнения полей событий, отличную от логики с использованием обработчиков (handlers). Например, когда требуется извлекать биометрические шаблоны только для части детекций, а не для всех детекций.

- В сервис API добавлен ресурс [«/ws»](#). Теперь настройка веб-сокетов и проксирование ответов Sender выполняется через сервис API. Это позволяет обеспечить единую точку входа в LP через сервис API и избежать отправки запросов напрямую в сервис Sender.

Настройка веб-сокетов напрямую через Sender остаётся доступной. Её можно использовать для снижения нагрузки на сервис API, в остальных случаях рекомендуется использовать сервис API.

- В сервисах API и Events в ресурсы [«/events»](#) для метода GET добавлены фильтры «event\_id\_\_gte» и «event\_id\_\_lt».

С помощью данных фильтров можно выполнять пагинацию, которая:

- быстрее пагинации по параметрам «page» и «page\_size». Она не замедляется при большом числе событий
- стабильнее пагинации по параметрам «page» и «page\_size». Изменение количества событий в процессе пагинации не приводят к потере событий или их дублированию в ответе.

- Для сервисов Faces и Python Matcher был изменён механизм перезагрузки настроек. Теперь перезагрузка выполняется путём перезапуска соответствующих процессов. Механизм обеспечивает более надёжное обновление настроек сервисов.

Следует учесть, что запросы, выполняемые в момент изменения настроек, могут закончиться с ошибкой. Сервис может быть недоступен некоторое время.

- Значения полей «user\_data» и «external\_id» в запросах теперь заданы равными «» по умолчанию в сервисе Events.
- Общий доступ к кэшу был добавлен для сервиса Python Matcher при запуске нескольких «рабочих процессов» (workers) одного сервиса. Теперь каждый из «рабочих процессов» использует один и тот же кэш биометрических шаблонов. Ранее при создании нескольких «рабочих процессов» каждый из них имел свой кэш биометрических шаблонов.

Данное изменение может как ускорить, так и замедлить работу сервиса. Если необходимо обеспечить сохранение кэша в каждом из процессов Python Matcher, следует запускать каждый из экземпляров сервиса по отдельности.

- В сервисы LUNA PLATFORM добавлена поддержка InfluxDB OSS версии 2.x для мониторинга. В LP\_Administrator\_Manual добавлен раздел «InfluxDB OSS 2», описывающий настройку LP для работы со второй версией БД. БД в комплекте поставки LP не была обновлена на новую версию.



## Исправленные ошибки

- Для сервиса Faces поддержана возможность обновления поля «event\_id» значением «null» при отправке запроса Patch в ресурс [«/faces/{face\\_id}»](#).
- В сервисе Tasks было исправлена проблема с экранированием символа разделителя в колонке отчёта. Если символ разделителя встречается в колонке отчёта, то теперь он экранируется.
- В сервисе Tasks добавлена обработка ошибок взаимодействия сервисов, возникших во время выполнения задач. Например, если сервис недоступен или возникли проблемы с соединением, то соответствующие ошибки корректно отображаются в списке ошибок выполнения задачи.

## LUNA PLATFORM v.5.4.0

### Изменения

- В события были добавлены поля «detect\_time» и «image\_origin»:
  - Поле «detect\_time» содержит время детекции на исходном кадре.
  - Поле «image\_origin» содержит ссылку на сохранённое исходное изображение, на котором была выполнена детекция.

Эти поля возвращаются в ответе на запрос POST в ресурс [«/handlers/{handler\\_id}/events»](#).

При сохранении события данные этих полей добавляются в базу данных Events в таблицы «face\_detect\_result» и «body\_detect\_result».

В связи с добавлением данных полей изменён формат входных (POST [«/events»](#)) и выходных (GET [«/events»](#), GET [«/events/{event\\_id}»](#)) данных событий в сервисе Events.

Эти новые поля возвращаются в ответе ресурса [«/ws»](#) при отправке событий сервисом Sender.

- В запрос POST [«/handlers»](#) в «storage\_policy» добавлена политика сохранения исходного изображения.

Для сохранения исходных изображений следует указать «store\_image» равным «1» для «image\_origin\_policy». По умолчанию сохранение исходных изображений отключено.

- Добавлена возможность указывать атрибуты в качестве кандидатов и эталонов для задачи перекрестного сравнения. См. [«/tasks/cross\\_match»](#).

В сервис Tasks добавлена возможность построения ROC-кривых с использованием атрибутов. См. [«/tasks/roc»](#).

Использование атрибутов позволяет проводить эксперименты (например, построить ROC-кривую по заданной выборке) и получать результаты, при этом не сохраняя лишние данные

в базу данных.

- В документ «LP\_Administration\_manual» добавлен раздел «Сохраняемые и оцениваемые данные», который описывает данные, которые извлекает и хранит LP.
- Добавлена возможность записи логов сервисов LP на сервер при использовании Compose. Ранее при установке с помощью Compose было нельзя сохранять логи на сервер.

Для сохранения логов на сервер следует предварительно создать директорию для хранения логов сервисов и назначить права для этих директорий (см. «Создание каталога журналов»). Если права доступа к директориям логов на сервере не назначены, то логи не смогут быть записаны и сервис упадёт. После запуска контейнеров следует обновить настройки логирования в Configurator (вручную или с использованием файла `logging.json`), чтобы включить логирование в файлы в директорию «`/srv/logs`» в контейнере (см. «Включение ведения журнала в каталог сервера»).

По умолчанию логирование в файлы на сервер выключено.

### Исправленные ошибки

- Исправлена ошибка с отображением полных URL для событий и лиц вместо относительных в ответе на запрос POST в ресурс «`/handlers/{handler_id}/events`».
- В запросе GET «`/events`» отключено автоматическое задание фильтра по времени, если указаны ID событий. Ранее значение по умолчанию для данного фильтра задавалось неявно, если оно не было явно задано в запросе.
- Исправлено создание Docker-контейнера для сервиса Handlers. Размер контейнера уменьшился на 5 гигабайт

## LUNA PLATFORM v.5.3.0

### Изменения

- Добавлена возможность сохранять исходные изображения. Сохраняемые изображения могут иметь следующие форматы: JPEG, PNG, BMP, TIFF, Portable pixmap.

Для сохранения изображений используется ресурс «`/images`».

Для получения существующих изображений и удаления изображений используется ресурс «`/images/{image_id}`».

Изображения сохраняются в бакет «`visionlabs-image-origin`» в хранилище Image Store.

URL сохранённого изображения можно указывать при выполнении запросов в ресурсы «`/detector`», «`/handlers/{handler_id}/events`», «`/verifiers/{verifier_id}/verifications`». Значение заголовка «Content-Type» такого запроса должно быть «`application/json`».

Пример URL изображения, сохранённого в LUNA PLATFORM:

http://<server\_ip>:5000/6/images/10bc2cb4-db84-410d-adc7-ff2ac17e4b2d

- Изменена логика запуска сервисов LUNA PLATFORM внутри контейнеров. Теперь приложения запускаются не от пользователя *root*, а от пользователя *luna*.

Данное изменение не коснулось контейнеров UI 3 и UI 4, а также сервисов построения индекса и сравнения по индексу.

- Прекращена поддержка детекторов FaceDetV1 и FaceDetV2.

Если ранее использовался другой детектор, то FaceDetV3 будет автоматически задан как используемый детектор в настройке «LUNA\_HANDLERS\_DETECTOR\_TYPE» после выполнения миграции настроек в сервисе Configurator. Миграция выполняется при обновлении базы данных Configurator.

- В политику сохранения событий («storage\_policy» > «event\_policy») ресурса «[/handlers](#)» добавлен параметр «wait\_saving». Он позволяет включать и отключать ожидание сохранения событий в БД Events перед отправкой ответа.
  - При отключённом параметре ответ на запрос «[/handlers/{handler\\_id}/events](#)» возвращается быстрее, но при этом система не присылает никаких уведомлений в случае, если не удалось выполнить сохранение. Данное поведение использовалось по умолчанию в предыдущих версиях LUNA PLATFORM.
  - При включённом параметре система дожидается сохранения событий перед отправкой ответа и возвращает код ошибки 500, если сохранение событий не удалось.

Параметр «wait\_saving» включен по умолчанию.

Параметр будет автоматически добавлен в «storage\_policy» уже созданных обработчиков при выполнении миграции базы данных сервиса Handlers. Для уже существующих обработчиков параметр будет выключен по умолчанию, чтобы не изменять поведение системы. Миграция выполняется автоматически в процессе установки при обновлении базы данных Handlers.

- Добавлена возможность указывать значение «application/msgpack» в качестве заголовка «Content-Type» для ресурсов:
  - «[/matcher/faces](#)»
  - «[/matcher/bodies](#)»

MessagePack упаковывает данные эффективнее, чем JSON. Это бинарный формат, поэтому не требует декодирования из BASE64.

- Версия CUDA в контейнере сервиса Handlers обновлена до версии 11.1.
- В документе LP\_Administration\_Manual дополнено описание для использования сетей извлечения биометрических шаблонов версий 46 и 52 (раздел «Переключение на нейросеть вер-

сии 46 или 52»).

- Увеличена скорость выполнения задач перекрестного сравнения.

### Исправленные ошибки

- Исправлена ошибка при выполнении сравнения с использованием биометрических шаблонов, полученных 58 версией нейросети.
- В сервисе Admin исправлена ошибка при авторизации с использованием cookies. При попытке авторизации возвращался 500 код ошибки.
- В сервисе Admin исправлена ошибка при открытии страницы с аккаунтами.
- Исправлено падение при обработке символа новой строки в запросе в сервис Events при использовании БД Vertica. При наличии в запросе символа «\n» теперь возвращается ошибка..

## LUNA PLATFORM v.5.2.1

### Изменения

- Сервис API:
  - Была исправлена валидация полей даты и времени для ресурсов, использующих фильтры `create_time__gte` и `create_time__lt`. При указании даты и времени в формате UTC (в поле использовался суффикс «Z») возникала ошибка. Пример ошибки: «2018-08-11T09:11:41.674Z».
  - Исправлена валидация для политики «match\_policy» при задании событий в качестве кандидатов для сравнения. При некорректном задании фильтров для событий возвращалось неправильное сообщение об ошибке: «unexpected value; permitted: 'faces'», если фильтры для событий были установлены неправильно.
- Сервис Admin:
  - Исправлена внутренняя ошибка с кодом ответа 500 для запросов без авторизационных данных.
  - При авторизации с использованием неправильной пары логин/пароль теперь возвращается ошибка «BadAdminAuth».

## LUNA PLATFORM v.5.2.0

### Изменения

- Появилась поддержка 57 и 58 сетей извлечения биометрических шаблонов. По умолчанию LUNA PLATFORM использует 56 сеть. Описание смены используемой версии нейросети можно найти в разделе «Переключение версии нейросети» документа LP\_Administration\_Manual.

В данной сборке 57 и 58 сети не поддерживаются в Сервисах построения индекса и поиска по индексу, т. к. для них не был обновлён SDK.

Нейросети версий 46 и 52, которые могут потребоваться для обратной совместимости, в дистрибутиве не предусмотрены. Они предоставляются VisionLabs по запросу. См. раздел «Переключение на нейросеть версии 46 или 52» в документе LP\_Administration\_Manual для получения информации о добавлении этих нейросетей в контейнер Handlers.

- Добавлен новый механизм Liveness V2. Теперь его можно использовать в ресурсах [«/liveness»](#) и [«/sdk»](#).

Liveness с использованием отдельного сервиса теперь называется Liveness V1.

Важно, что Liveness v2 не требует отдельного сервиса, он является частью сервиса Handlers. Поэтому опция «liveness» в «ADDITIONAL\_SERVICES\_USAGE» должна быть отключена при использовании Liveness v2.

В ресурсе [«/liveness»](#) можно использовать как новый Liveness v2, так и Liveness v1. Liveness V2 доступен для ресурса [«/liveness»](#) сервисов API, Backport 3 и Backport 4. Запрос (кроме раздела metadata) и возвращаемый результат в данном ресурсе будут иметь одинаковый формат для обоих Liveness. Раздел metadata не используется для Liveness V2.

Liveness V1 не используется в ресурсе [«/sdk»](#). Если отправить запрос на этот ресурс и указать «estimate\_liveness = 1» при включённом Liveness V1, то вернётся ошибка.

Нельзя одновременно использовать обе версии Liveness в LUNA PLATFORM. Переключение между используемыми версиями Liveness происходит на основе значения в файле лицензии LUNA PLATFORM.

- 0 — Liveness не используется
- 1 — Используется механизм Liveness V1
- 2 — Используется механизм Liveness V2

Не требуется обновлять лицензию LUNA PLATFORM, если она у вас уже есть и не требуется использовать Liveness V2. В противном случае необходимо обновить существующий ключ.

Описание работы механизмов Liveness дано в разделе «Описание Liveness» документа LP\_Administration\_Manual. Также там приведены таблицы условий работы версий Liveness для ресурсов [«/sdk»](#) и [«/liveness»](#).

См. требования для Liveness V2 в разделе «Требования к Liveness V2».

- Поворот изображений:
  - Добавлена возможность автоматического поворота изображения на основе EXIF данных (параметр «use\_exif\_info»). Данная возможность добавлена в ресурсы:
    - \* [«/detector»](#)
    - \* [«/verifiers/{verifier\\_id}/verifications»](#)

- \* «/handlers/{handler\_id}/events»
- \* «/sdk»

Параметр «use\_exif\_info» включен по умолчанию.

Если в запросе указано, что обработанное изображение является биометрическим образцом, опция игнорируется.

- В конфигурации сервиса Handlers добавлена опция «LUNA\_HANDLERS\_USE\_AUTO\_ROTATION». Она позволяет включить режим автоматического поворота изображения, если оно повернуто на 90, 180, 270 градусов.

Данная сеть потребляет значительное количество ресурсов сервера, поэтому по умолчанию она выключена.

Опция не применяется к биометрическим образцам. Если в запросе указано, что обрабатываемое изображение является биометрическим образцом, но при этом передаваемое изображение в действительности повернуто, то автоматический поворот такого изображения выполнен не будет. И LUNA PLATFORM будет выполнять обработку изображения, которое не является биометрическим образцом.

Две вышеперечисленные опции автоматического поворота и поворота на основе EXIF данных можно использовать совместно.

- Прекращена поддержка сервиса CORE Matcher. Все операции для сравнения по спискам теперь выполняются с помощью сервиса Python Matcher.

Для увеличения производительности настройка «cache\_enabled» для данного сервиса должна быть выставлена в True (выставлена по умолчанию). В этом случае биометрические шаблоны будут сохраняться в оперативную память и быстрее обрабатываться Python Matcher.

- При использовании внешней СУБД PostgreSQL (не контейнера PostgreSQL из комплекта поставки) следует пересоздать функцию сравнения в БД с добавлением директивы «PARALLEL SAFE». Эта директива ускоряет сравнение по базе данных.

Эта директива будет добавлена автоматически во время базовой установки контейнера PostgreSQL из дистрибутива.

Пример строки для удаления функции из базы данных: `DROP FUNCTION VLMatch;`

Пример строки для добавления функции:

```
CREATE FUNCTION VLMatch(bytea, bytea, int) RETURNS float8 AS '
    VLMatchSource.so', 'VLMatch' LANGUAGE C PARALLEL SAFE;
```

## LUNA PLATFORM v.5.1.3

### Изменения

- Добавлен заголовок Accept для ресурса «/sdk», который позволяет указать content-type запроса: JSON или MessagePack. MessagePack упаковывает данные эффективнее, чем JSON. Это бинарный формат, поэтому не требует декодирования из BASE64. См. [«sdk»](#) > [«sdk resource»](#).
- В ресурс «/sdk» в сервисе API добавлено определение наличия очков. См. [«sdk»](#) > [«sdk resource»](#).
- В API в ресурсы «/lists» и «/lists/count» для метода GET добавлены фильтры «list\_id\_\_gte» и «>list\_id\_\_lt». Полученные списки сортируются по list\_id.

С помощью данных фильтров можно выполнять пагинацию, которая:

- быстрее пагинации по параметрам «page» и «page\_size». Она не замедляется при большом числе списков
- стабильнее пагинации по параметрам «page» и «page\_size». Изменения списков в процессе пагинации не приводят к потере списков или их дублированию в ответе.

См. [«lists»](#) > [«get lists»](#)

См. [«lists»](#) > [«get lists count»](#)

- В сервис Faces добавлен ресурс «/lists/deletions», который позволяет отслеживать историю удаления списков. См. [«administration»](#) > [«get lists deletions»](#).

В сервис Faces добавлено ведение лога удалённых списков. См. [«administration»](#) > [«clear lists deletions log»](#).

- В сервисе Tasks в ресурсе «/additional\_extract» добавлена возможность извлекать базовые атрибуты для лиц, для которых доступны биометрические образцы (samples), но нет извлечённых базовых атрибутов. См. [«tasks processing»](#) > [«additional extract task»](#).
- В сервисе Events в ресурсе «/events» при использовании метода POST добавлен query параметр «wait\_events\_saving». См. [«events»](#) > [«create new events»](#).

При включённом параметре система будет ожидать сохранение события в базу, прежде чем возвращать ответ (поведение использовалось по умолчанию ранее).

При выключенном параметре система не ожидает сохранения событий в базу, а отправляет ответ сразу после проверки события и добавления его в буфер.

- В сервисе Python Matcher уменьшено потребление памяти при кэшировании большого числа маленьких списков.

## LUNA PLATFORM v.5.1.2

### Изменения

- Добавлена возможность задавать количество «рабочих процессов» (workers) для сервисов. Запросы автоматически балансируются между активными процессами. Для задания количества процессов в Docker-контейнерах используется настройка WORKER\_COUNT.

Для увеличения количества экземпляров сервисов LUNA на одном сервере рекомендуется увеличивать количество «рабочих процессов», а не запускать дополнительные контейнеры сервисов.

Не рекомендуется увеличивать количество «рабочих процессов» для сервиса Handlers при использовании GPU. Могут возникнуть проблемы при недостаточном количестве памяти GPU и «рабочие процессы» будут препятствовать друг другу.

См. раздел «Рабочие процессы» в LP\_Administration\_Manual.pdf.

- Добавлен функционал обновления настроек сервисов без перезагрузки самих сервисов. Настройки могут быть получены как из сервиса Configurator, так и из конфигурационных файлов.

Для включения данной возможности в контейнерах используется переменная RELOAD\_CONFIG.

Период проверки обновлений настроек задаётся переменной RELOAD\_CONFIG\_INTERVAL.

Обратите внимание, что во время изменения важных системных настроек (настроек баз данных, списка плагинов и др.) не следует отправлять в сервис запросы.

См. раздел «Автоматическая перезагрузка конфигураций» в LP\_Administration\_Manual.pdf и раздел «Configuration» в «ServiceManuals» каждого сервиса.

- В раздел конфигурации «ADDITIONAL\_SERVICES\_USAGE» добавлена настройка «Matcher» для включения и отключения использования CORE Matcher.
- В сервис Faces добавлен функционал missing basic attributes. Функциональность позволяет:
  - Получить лица, не имеющие базовых атрибутов «[administration](#)» > «[get faces without basic attributes](#)»;
  - Получить количество лиц без базовых атрибутов «[administration](#)» > «[get face count without basic attributes](#)»;
  - Получить количество базовых атрибутов, связанных с лицами «[administration](#)» > «[get basic attributes count info](#)».
- Добавлен параметр «DB\_CONNECT\_TIMEOUT» в конфигурацию сервиса Events.
- Удален скрипт dump\_emails.py из сервиса Admin.



## LUNA PLATFORM v.5.1.1

### Изменения

- Добавлена возможность указывать атрибуты в качестве кандидатов для сравнения в ресурсе [«/matcher/faces»](#) в APIReferenceManual.html. Это позволяет выполнять сравнение биометрических шаблонов без создания объекта face или event для кандидата.
- Доступен механизм миграции настроек в сервисе Configurator. Перенос настроек выполняется без изменения пользовательских значений. Для применения этого механизма к LP build 5.1.0 требуются дополнительные действия, которые описаны в руководстве «LP\_Docker\_Upgrade\_Manual.html».
- Для событий добавлено поле «track\_id». Оно позволяет отмечать события, принадлежащие одному и тому же человеку, для последующего анализа. Вы должны вручную указать «track\_id» в запросе [«/handlers/{handler\\_id}/events»](#). Данное поле можно использовать:
  - в качестве фильтра для кандидатов сравнения;
  - Для задач перекрестного сопоставления, кластеризации, компоновщика и репортера. В задаче кластеризации теперь можно указать, следует ли добавлять события с одинаковым «track\_id» в тот же кластер.
- Вы можете указать настройки из БД Configurator при выполнении миграции с помощью Alembic.

```
alembic -x luna-config=http://127.0.0.1:5070/1 upgrade head
```

- Добавлены примеры для использования ресурса «ws» в документацию «SenderReferenceManual.html».
- Теперь возвращается ошибка в случае, когда в политике storage\_policy указано сохранение событий, но сам сервис не используется. В этом случае возвращается ошибка 403 Forbidden 11040 «Luna Events service is disabled».
- Исправлена неправильная ссылка на нормализованное изображение, возвращаемая при генерации событий «events» > «generate events».

Ссылка в предыдущей сборке LP:

«url»: «http://image-store:5020/1/buckets/visionlabs-samples/images/2e8045f2-14cd-4c8b-af3a-7c959e85fe6f»

Исправленная ссылка:

«url»: «/6/samples/faces/939fca8a-753f-4ee4-8450-9c1bb3c1f76c»