

PL6: Платформа Tantor 6

мониторинг и администрирование
СУБД PostgreSQL и кластеров Patroni



Учебный курс
Теоретическая часть



tantor

Автор: Олег Иванов

Оглавление

Глава	PL6: Платформа Tantor 6	страница
1	Обзор курса	7
	Предварительная подготовка	
	Материалы курса	
	Разделы курса	
	О курсе	
	О компании Тантор Лабс	
	О курсе	
	Платформа Tantor	
	СУБД Tantor Postgres	
	Tantor xData	
Доработка расширений PostgreSQL		
Конференции PGBootCamp		
1 Возможности Платформы Tantor	18	
Возможности Платформы Tantor		
Работа с экземплярами PostgreSQL: Обзор		
Расширение pg_stat_statements		
Работа с экземплярами: Графики		
Работа с экземплярами: Конфигурация		
Работа с экземплярами: Обслуживание		
Работа с экземплярами: Браузер БД		
Браузер БД -> Аудит		
Браузер БД -> SQL-редактор и Схема		
Работа с экземплярами: Профилировщик запросов		
Профилировщик запросов -> Планы		
Работа с экземплярами: Текущая активность		
Репликация и Табличные пространства		
Работа с экземплярами: Настройки мониторинга		
Работа с экземплярами: Расширенная аналитика		
Работа с экземплярами: Задачи		
Модули Платформы Tantor		
Оповещения		
Практика 1		
2 Архитектура Платформы Tantor	38	
Виды установки Платформы Tantor		
Агенты Платформы Tantor		
Архитектура Платформы Tantor		
Типичная Архитектура		
Масштабируемая архитектура		
Агент		
Файлы Агента		
Список контейнеров		
2-2 Интеграция - Swagger UI	47	
Swagger UI		
Страница Swagger UI		
Токены		
Ввод токена в Swagger UI		
Практика 2		
3 Начало работы с Платформой	53	
Первый вход в Платформу		
Лицензия на использование Платформы		
Начальная страница Платформы - список тенантов		
Тенант: Расширенная информация		
Настройки тенанта: Пользователи		
Страница пользователя Платформы		
Включение пользователя в группы		
Настройки тенанта: Группы		
Тенант: Рабочие пространства		
Тенант: Рабочие пространства		

	Практика 3	
4	Разграничение доступа	65
	Виды пользователей Платформы Tantor	
	Группы пользователей	
	Внешняя аутентификация	
	Страница тенанта	
	Страница тенанта	
	Страница рабочих пространств	
	Страница экземпляра	
	Практика 4	
5	Мониторинг	74
	Панель управления Платформой	
	Экземпляры PostgreSQL	
	Страница тенанта со списком экземпляров	
	Рабочее пространство: Агенты	
5-2	Patroni	79
	Рабочее пространство: Кластеры Patroni	
	Рабочее пространство: КЛАСТЕРЫ	
	Страница кластера Patroni: ОБЗОР И УПРАВЛЕНИЕ	
	Страница кластера Patroni: МОНИТОРИНГ	
	Страница кластера Patroni: Конфигурация	
	Страница кластера Patroni: Применение изменений	
	Практика 5	
6	Обзор экземпляра	87
	Страница Обзор	
	Страница Обзор	
	Страница Обзор -> УВЕДОМЛЕНИЯ -> Подсказки системы	
	Оповещения (сообщения, уведомления)	
	Экземпляр: Сессии	
	Экземпляр: Транзакции	
	Экземпляр: Ввод/Вывод на диск	
	Экземпляр: Ввод/Вывод блоков	
	Экземпляр: Нагрузка CPU	
	Экземпляр: Доступно ОЗУ	
	Экземпляр: Размер данных	
	Экземпляр: Размер диагностических логов	
	Экземпляр: Среднее время запроса	
	Экземпляр: 5 самых длительных запросов	
	Экземпляр: % КОНТР. ТОЧЕК	
	Экземпляр: Временные файлы	
	Экземпляр: Временные файлы (события)	
	Экземпляр: Сеть	
	Практика 6	
7	Оповещения	107
	Оповещения	
	Страница со списком оповещений	
	Страница оповещения: ОБЗОР	
	Страница оповещения: СВЯЗАННЫЕ ОПОВЕЩЕНИЯ	
	Страница оповещения: ЖУРНАЛ СОБЫТИЙ	
7-2	Уведомления через e-mail	113
	Интеграция со службами сообщений	
	Настройки тенанта: Почтовый сервер (SMTP)	
	Интеграция со службами сообщений	
	Отправка тестового письма	
	Шаблоны писем	
	Редактор шаблона писем	
7-3	Оповещения через Telegram и Mattermost	120
	Посылка сообщений в канал Telegram	
	Создание канала Telegram	
	Создание бота для канала Telegram	
	Добавление бота в канал Telegram	

	Создание канала уведомлений в Платформе	
	Добавление правил для канала Telegram в Платформе	
	Пример сообщений в канале Telegram	
	Mattermost	
	Практика 7	
8	Экземпляр: Профилировщик запросов	130
	Экземпляр: Настройки мониторинга	
	Экземпляр: Настройки мониторинга -> ТРИГГЕРЫ	
	Экземпляр: Профилировщик запросов	
	Экземпляр: Профилировщик запросов: СТАТИСТИКА	
	Экземпляр: Профилировщик запросов: ПЛАНЫ	
	Экземпляр: Профилировщик запросов: простой план	
	Экземпляр: Профилировщик запросов: долгий запрос	
	Экземпляр: Профилировщик запросов: tilemap	
	Экземпляр: Профилировщик запросов: piechart	
	Экземпляр: Профилировщик запросов: диаграмма	
	Экземпляр: Профилировщик запросов: рекомендации	
	Экземпляр: Профилировщик запросов: статистика	
8-2	Расширенная аналитика	143
	Экземпляр: Расширенная аналитика	
	Экземпляр: Проверка настройки аналитики	
	Экземпляр: страница pg-monitor	
	Экземпляр: Расширенная аналитика	
	Расширенная аналитика: ошибки в логе PostgreSQL	
	Расширенная аналитика: статистика по таблицам	
	Расширенная аналитика: взаимоблокировки	
	Расширенная аналитика: Проблемные запросы	
	Расширенная аналитика: PSA (pg_stat_activity)	
	Расширенная аналитика: СНЕЧКПОИНТ	
	Расширенная аналитика: Логи	
	Расширенная аналитика: Мегазапросы	
	Расширенная аналитика: PG config	
	Практика 8	
9	Конфигурирование и обслуживание PostgreSQL	158
	Текущая активность	
	Текущая активность: Завершить процесс	
	Текущая активность: ОЖИДАНИЕ и ЗАБЛОКИРОВАНО	
	Текущая активность: дерево блокировок	
9-2	Конфигурация экземпляра PostgreSQL	163
	Экземпляр: Конфигурация	
	Экземпляр: изменение параметров конфигурации	
	Параметры конфигурации: настройка и фильтр	
	Описание параметров	
	Группы параметров конфигурации и модификаторы	
	Модуль: Группы параметров	
	Сравнение и создание групп параметров	
	Модификаторы групп параметров	
	Редактирование модификаторов	
	Синхронизация на странице Конфигурация	
	Синхронизация на странице модуля: Группы параметров	
9-3	Параметры конфигурации Patroni	175
	Список кластеров Patroni и их экземпляров	
	Кластер Patroni: МОНИТОРИНГ	
	Страница Конфигурация у экземпляра в кластере Patroni	
	Страница настройки кластера	
	Практика 9	
10	Браузер БД	181
	Браузер БД	
	Аудит объектов Баз Данных	
	Аудит объектов Баз Данных: графики	
	Аудит объектов Баз Данных: Рекомендации	

	Статистика по таблице	
	Браузер БД: Схема данных	
	Браузер БД: SQL-редактор	
10-2	Обслуживание баз данных	189
	Обслуживание баз данных	
	Запуск задачи обслуживания	
	Статус задач обслуживания	
	Обслуживание индексов и счетчика транзакций	
10-3	Планировщик Платформы Тантор	194
	Планировщик Платформы	
	Страница со списком задач	
	Добавление задачи	
	Добавление действий в задачу	
	Редактирование действий задачи	
	Задачи в Журнале событий	
	Фильтры в Журнале событий	
	Экземпляр: страница Репликация у мастера	
	Экземпляр: пункт меню Репликация на странице реплики	
	Настройка тенанта: Резервное копирование	
	Мониторинг бэкапов	
	Практика 10	
11	Анонимайзер (маскировка данных)	207
	Обзор анонимайзера	
	Словари анонимайзера	
	Мета-словари	
	Шаблон сенситивного словаря	
	Источники данных	
	Сканирование	
	Результат сканирования	
	Создание предпросмотра	
	Предпросмотр	
	Дамп	
	Загрузка дампа	
	Практика 11	
12	Установка Платформы Тантор	220
	Способы установки Платформы Tantor	
	Утилита диагностики Платформы Tantor	
	Запуск инсталлятора Платформы Tantor	
	Обновление Платформы Tantor	
	Очистка места после обновления	
12-2	Обзор docker и docker-compose	226
	docker и docker-compose	
	Статусы контейнеров docker	
	Сетевая изоляция контейнеров docker	
	Сети docker типа bridge	
	Содержимое файла docker-compose.yml	
	Файл docker-compose.yml	
	Файл docker-compose.yml: networks	
	Файл docker-compose.yml	
	Файл docker-compose.yml: volumes	
	Файл /etc/hosts в контейнере docker	
	Установка и обновление docker и docker-compose	
12-3	Установка Платформы, Prometheus, Grafana	238
	Установка Prometheus и Grafana	
	Создание в Grafana соединения с Prometheus	
	Импорт Dashboard в Grafana	
	Интеграция с Traefik	
	Деинсталляция Платформы	
	Установочные файлы	
	Установка Платформы	
	Установка Платформы	

	Установка Платформы вне докера	
	Установка Платформы вне докера	
12-4	Установка Агента Платформы Tantor	249
	Установка Агента Платформы	
	Установка Агента в веб-интерфейсе	
	Установка Агента в веб-интерфейсе	
	Регистрация Агента в Платформе через веб-интерфейс	
	Установка и регистрация Агента в командной строке	
	Перезапуск Агента после регистрации в Платформе	
	Регистрация экземпляра PostgreSQL в командной строке	
	Практика 12	

Авторские права

Учебное пособие, практические задания, презентации (далее документы) предназначены для учебных целей.

Документы защищены авторским правом и законодательством об интеллектуальной собственности.

Вы можете копировать и распечатывать документы для личного использования в целях самообучения, а также при обучении в авторизованных ООО «Tantor Labs» учебных центрах и образовательных учреждениях. Авторизованные ООО «Tantor Labs» учебные центры и образовательные учреждения могут создавать учебные курсы на основе документов и использовать документы в учебных программах с письменного разрешения ООО «Tantor Labs».

Вы не имеете права использовать документы для обучения сотрудников или других лиц без разрешения ООО «Tantor Labs». Вы не имеете права лицензировать, коммерчески использовать документы полностью или частично без разрешения ООО «Tantor Labs».

При некоммерческом использовании (презентации, доклады, статьи, книги) информации из документов (текст, изображения, команды) сохраняйте ссылку на документы.

Текст документов не может быть изменен каким-либо образом.

Информация, содержащаяся в документах, может быть изменена без предварительного уведомления и мы не гарантируем ее безошибочность. Если вы обнаружите ошибки, нарушение авторских прав, пожалуйста, сообщите нам об этом.

Отказ от ответственности за содержание документа, продукты и услуги третьих лиц:

ООО «Tantor Labs» и связанные лица не несут ответственности и прямо отказываются от любых гарантий любого рода, включая потерю дохода, нанесенные прямым или непрямым, специальным или случайным использованием документа. ООО «Tantor Labs» и связанные лица не несут ответственности за любые убытки, издержки или ущерб, возникшие в результате использования информации, содержащейся в документе или использования сторонних ссылок, продуктов или услуг.

Авторское право © 2025, ООО «Tantor Labs»

Создал: Олег Иванов

!

Создан: 10 ноября 2025 г.

По вопросам обучения обращайтесь: edu@tantorlabs.ru



Обзор курса

Платформа Tantor 6

Предварительная подготовка

- администрирование любой СУБД семейства PostgreSQL или знания эквивалентные курсу "Tantor: Администрирование PostgreSQL"
- Курс предназначен:
 - Администраторам баз данных PostgreSQL
 - Специалистам по мониторингу и эксплуатации
 - Инженерам технической поддержки

Предварительная подготовка

Для успешного прохождения курса достаточно опыта администрирования любой СУБД семейства PostgreSQL или знаний в объеме курса "Tantor: Администрирование PostgreSQL".

В курсе будут рассмотрены возможности Платформы Tantor.

Для успешного прохождения курса рекомендуется слушать инструктора, если будут возникать вопросы задавать их, выполнять практические задания.

Материалы курса

- Учебное пособие в форме книги в формате pdf, которое содержит теоретическую часть курса
- Практические задания
- Виртуальная машина для выполнения практических заданий

Материалы курса

К материалам курса относятся:

- 1) Учебное пособие в форме книги в формате pdf, которое содержит теоретическую часть курса.
- 2) Практические задания в форме книги в формате pdf.
- 3) Виртуальная машина с установленными в операционной системе Astra Linux: Платформой Tantor версии 6, СУБД Tantor Postgres версии 17.5. Может предоставляться доступ к виртуальной машине на время курса или образ в формате ova. Образ виртуальной машины можно использовать с Oracle VirtualBox версии 6.1 и выше.

Разделы курса

1. Возможности Платформы Tantor
2. Архитектура Платформы Тантор (4 части)
3. Мониторинг (8 частей)
4. Конфигурирование и обслуживание PostgreSQL (7 частей)
8. Установка Платформы, Prometheus, Grafana (5 частей)

Разделы курса

1. Возможности Платформы Tantor
2. Архитектура Платформы Тантор (4 части)
3. Мониторинг (8 частей)
4. Конфигурирование и обслуживание PostgreSQL (7 частей)
8. Установка Платформы, Prometheus, Grafana (5 частей)

О курсе

- очное или дистанционное обучение с инструктором:
 - › продолжительность 2 дня
 - › начало в 10:00
 - › перерыв на обед 13:00-14:00
 - › окончание до 17:00

О курсе

Курс предназначен для очного или дистанционного обучения с инструктором. Курс состоит из теоретической части - глав, практических упражнений и перерывов. Перерывы совмещаются с практическими упражнениями, которые слушатели выполняют самостоятельно на подготовленной для курса виртуальной машине.

Примерное расписание:

- 1) начало в 10:00
- 2) перерыв на обед 13:00-14:00. Начало обеда может сдвигаться на полчаса в диапазоне от 12:30 до 13:30, так как обычно совмещается с перерывом между главами.
- 3) окончание теоретической части до 17:00.

Курс состоит из теоретической части (глав) и практических заданий. Длительность глав примерно 20-40 минут. Точное время начала изложения глав и времени на практические задания определяет инструктор. Длительность выполнения упражнений может отличаться у разных слушателей и это не влияет на эффективность усвоения материала курса. Доделывать упражнения можно в перерывах между главами или в конце каждого дня. Порядок следования глав и упражнений на эффективность усвоения материала курса не влияет. Проверка выполнения заданий не производится. Для успешного усвоения материала курса достаточно:

- 1) слушать инструктора, просматривая в процессе изложения инструктора текст на слайдах и под слайдом
- 2) задавать инструктору вопросы, если возникает внутреннее несогласие (появляются вопросы)
- 3) выполнять практические задания и читать текст в практических заданиях

В материалы курса входят:

- 1) учебник в формате `pdf`
- 2) практические задания в формате `pdf`
- 3) образ виртуальной машины в формате `ova`

О компании Тантор Лабс

- с 2016 года на международном рынке
- Платформа Tantor доступна клиентам с 2020 года
- с 2021 года на российском рынке
- разработка СУБД Tantor Postgres
- разработка Платформы Tantor для мониторинга и управления СУБД семейства PostgreSQL, а также кластеров Patroni
- многолетний опыт эксплуатации высоконагруженных систем
- входит в Группу Астра

О компании Тантор

С 2016 года команда Тантор работала на международном рынке поддержки эксплуатации СУБД PostgreSQL и обслуживала клиентов из Европы, Северной и Южной Америки, Ближнего Востока. Команда Тантор разработала программное обеспечение Платформа Tantor и в последующем создала СУБД Tantor Postgres, основанную на программном коде свободно распространяемой СУБД PostgreSQL.

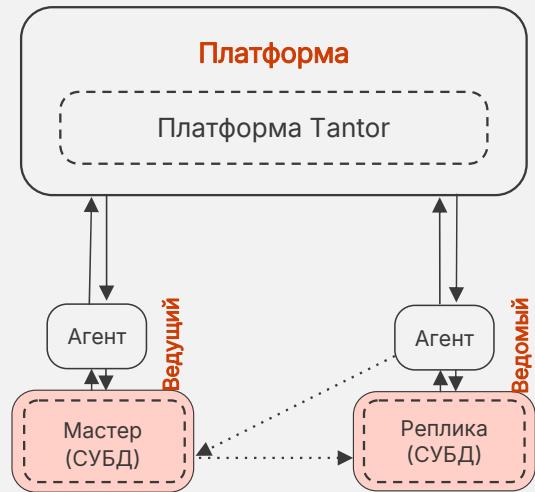
Платформа Tantor доступна клиентам с 2020 года

В 2021 году компания полностью переориентировалась на российский рынок, где сконцентрировала свои основные направления деятельности на проектирование и разработку СУБД Tantor Postgres, а также развитие Платформы Tantor - инструмента управления и мониторинга БД, основанных на PostgreSQL.

Проектирование и разработка продуктов основывается на накопленном многолетнем опыте в эксплуатации высоконагруженных программных систем в государственном и частном секторах. В конце 2022 года компания вошла в Группу Астра 

Платформа Tantor

- программное обеспечение для управления большим количеством СУБД и кластеров Patroni
- управляет СУБД Tantor Postgres и форками PostgreSQL
- сбор показателей работы экземпляров PostgreSQL, хранение и обработка показателей, рекомендации по настройке производительности
- интеграция с почтовыми системами, службами каталогов, мессенджерами



Платформа Tantor

Платформа Tantor - программное обеспечение для управления и мониторинга СУБД Tantor Postgres, любых форков PostgreSQL, кластеров Patroni. Позволяет удобно управлять большим числом экземпляров PostgreSQL.

Относится к классу программных продуктов, в который входит Oracle Enterprise Manager Cloud Control.

Преимущества использования Платформы Tantor:

1. Сбор показателей работы экземпляров PostgreSQL, хранение и обработка показателей, рекомендации по настройке производительности
2. Интуитивно понятный и функциональный графический интерфейс позволяет сосредоточиться на показателях работы экземпляров PostgreSQL
3. Графический интерфейс продуман до мелочей и эстетически приятен, что повышает удовлетворённость от повседневного использования Платформы
4. Графические элементы (размер, цвета, толщина линий, шрифты) оптимизированы для удобства восприятия при долговременной работе
5. Информация, отображаемая Платформой, не перегружена неважными деталями, информация подаётся максимально просто и понятно, что позволяет сосредоточиться на важных метриках работы экземпляров PostgreSQL
6. Автоматизирует рутинные задачи, повышая эффективность работы и снижая вероятность ошибок
7. Управляет не только СУБД Tantor Postgres, но и другими СУБД семейства PostgreSQL
8. Интеграция с почтовыми системами, службами каталогов, мессенджерами, Prometheus, Grafana
9. Простое внедрение: развертывание и ввод СУБД под обслуживание Платформой Tantor с помощью Ansible
10. Гибкое лицензирование: доступны постоянные (без ограничения срока) и срочные (на 1, 2, 3 года) лицензии.

Платформа Tantor DLH

Тантор Лабс также выпускает Платформу Tantor DLH - программное обеспечение, позволяющее организовать процесс трансформации и загрузки данных по логике Extract Transform Load или Extract Load Transform в СУБД Tantor Postgres для организации хранилищ и витрин данных. Относится к классу программных продуктов, в который входит Oracle Data Integrator.

СУБД Tantor Postgres

Tantor BE



Новые возможности и доработки по сравнению с PostgreSQL, техническая поддержка

Tantor SE



СУБД Enterprise-уровня, подходит для наиболее нагруженных OLTP-систем или КХД размером до 100ТБ

Tantor SE 1C



СУБД для высоких нагрузок, оптимизированная и одобрена для работы с приложениями 1С

Tantor PipelineDB



Расширение, позволяющее непрерывно обрабатывать данные

В составе Tantor xData



Максимальная версия СУБД, оптимизированная для работы с 1С



Учебный курс "Платформа Tantor 6" Тантор Лабс © 2025 www.tantolabs.ru



14

СУБД Tantor Postgres

СУБД Tantor Postgres - реляционная база данных семейства PostgreSQL с повышенной производительностью и стабильностью. Выпускается в нескольких редакциях (сборках): BE (Basic Edition), SE (Special Edition), SE 1C, **Certified Исполнение 1** (сертифицированные SE и SE 1C) и **Certified Исполнение 2** (Сертифицированный BE). Редакция Special Edition высоконагруженных OLTP-систем и хранилищ данных размерами до 100ТБ. Редакция Special Edition 1C для приложений 1С.

Для всех редакций доступна техническая поддержка, помощь в построении архитектурных решений, миграции с СУБД других производителей. Программное обеспечение Тантор Лабс включено в "Единый реестр российских программ для электронных вычислительных машин и баз данных".

При приобретении СУБД Tantor Postgres **бесплатно предоставляется лицензия** на Платформу Tantor для управления приобретенными СУБД Tantor Postgres.

Tantor xData

- Программно-Аппаратный Комплекс, с высокой производительностью, отказоустойчивостью, безопасностью
- версия 2A, 2Y - процессоры AMD EPYC и x86-64
- версия 2B - процессоры Baikal-S
- высокая производительность и масштабируемость
- снижение затрат на инфраструктуру и администрирование
- в состав входит СУБД Tantor Postgres и Платформа Tantor



Tantor xData

Программно-аппаратный комплекс (ПАК) Tantor XData обеспечивает рабочие нагрузки большого масштаба и критичности с высокой производительностью и доступностью. Консолидация различных рабочих нагрузок Tantor Postgres SE на машине баз данных XData в корпоративных центрах обработки данных помогает организациям повысить операционную эффективность, сократить объем администрирования и снизить затраты.

Программно-аппаратный комплекс (ПАК) Tantor XData предназначен для миграции с комплексов иностранных производителей и обеспечивает аналогичную нагрузочную способность. Является заменой высоконагруженным СУБД размером до ~50Тб на один экземпляр, обслуживающие нагрузку типа OLTP, работающие на программно-аппаратных комплексах иностранных производителей. Для СУБД, обслуживающих хранилища данных размером до ~120Тб на один экземпляр.

Является заменой для тяжёлых ERP от 1С при миграции с СУБД иностранных производителей. Позволяет консолидировать несколько СУБД в одном ПАК. Может использоваться при миграции с SAP на 1С:ERP.

Предназначен для создания облачных платформ.

Преимуществом при эксплуатации xData является наличие в составе ПАК удобной графической системы мониторинга работы СУБД: Платформы Tantor.

С 2025 года выпускается **вторая** версия ПАК:

xData 2A - на процессорах x86-64 на основе серверов Aquarius.

xData 2Y - на процессорах x86-64 на основе серверов Yadro.

xData 2B - на процессорах Baikal-S на основе серверов Элпитех.

Доработка расширений PostgreSQL

- 1. pg_cluster
- 2. pg_anon
- 3. pg_perfbench
- 4. ansible_tantor_agent
- 5. pg_configurator
- 6. pg_store_plans
- 7. ldap2pg
- 8. citus
- 9. wal-g
- 10. odyssey
- 11. plantuner
- 12. pg_orchestrator
- 13. pgtools
- 14. pipelinedb

Доработка расширений PostgreSQL

Сотрудники Тантор Лабс дорабатывают и создают расширения для СУБД PostgreSQL.

Репозитории расширений: <https://github.com/orgs/TantorLabs>

Список расширений:

1. pg_cluster
2. pg_anon
3. pg_perfbench MIT License
4. ansible_tantor_agent MIT License
5. pg_configurator MIT License
6. pg_store_plans
7. ldap2pg PostgreSQL License
8. citus GNU Affero General Public License v3.0
9. wal-g Apache License, Version 2.0 (Izo - GPL 3.0+)
10. odyssey BSD 3-Clause "New" or "Revised" License
11. plantuner
12. pg_orchestrator MIT License
13. pgtools
14. pipelinedb Apache License 2.0

Конференции PGBootCamp

- Тантор Лабс принимает участие в организации конференций
- Конференция PGBootCamp проводилась:
 - › Екатеринбург 10 апреля 2025
 - › Казань 17 сентября 2024
 - › Минск 16 апреля 2024
 - › Москва 5 октября 2023

Конференции PGBootCamp

Тантор Лабс является активным участником организации конференций сообщества PostgreSQL в рамках глобальной инициативы PG BootCamp.

Участие в конференции бесплатно и возможно онлайн и офлайн: <https://pgbootcamp.ru/>
Можно стать докладчиком на конференции.

Материалы докладов конференций в открытом доступе: <https://github.com/PGBootCamp>

Выступления <https://www.youtube.com/@PGBootCampRussia>

Конференция PGBootCamp проводилась:

Екатеринбург 10 апреля 2025

Казань 17 сентября 2024

Минск 16 апреля 2024

Москва 5 октября 2023

Конференция Tantor JAM проводилась в Москве 19 сентября 2025 года, материалы выступлений: <https://tantorlabs.ru/jam-2025> и <https://tantorlabs.ru/jam-2024>



1

Обзор

Возможности Платформы Tantor

Возможности Платформы Tantor

- Dashboard (панель): графики для наиболее важных метрик
- Конфигуратор: рекомендации по установке значений параметров PostgreSQL
- Анализатор планов запросов Tensor: встроенный анализатор планов запросов с рекомендациями исправления запросов и визуализацией планов
- Аналитика: анализ диагностических журналов PostgreSQL
- Управление кластерами Patroni: просмотр и изменение параметров, мониторинг репликации и событий failover
- Интеграция с системами
 - уведомлений
 - мониторинга: по REST API (Swagger/OpenAPI)
 - службами каталогов по протоколу LDAP
 - системами резервирования: RuBackup и Backman

Возможности Платформы Tantor

Dashboard (панель): готовые графики для наиболее важных метрик: таких как соединения с базами данных, транзакции, попадание в буферный кэш, WAL, репликация, контрольные точки, блокировки; CPU, RAM, IO, место в файловых системах, сеть. Группировка и фильтрация метрик по экземплярам и пространствам.

Конфигуратор: рекомендации по установке значений параметров конфигурации PostgreSQL

Внесение изменений в значения параметров: генерация команд ALTER SYSTEM, хранение истории изменений, отмена и применение через агентов Платформы к экземплярам

Встроенный анализатор планов запросов от Tensor: В Платформу встроен лицензированный анализатор планов запросов от компании Tensor с рекомендациями исправления запросов и использования индексов, который не требует доступа в интернет и передачи запросов во внешние системы.

Повседневное обслуживание: планирование запуска VACUUM/VACUUM FULL/REINDEX/ANALYZE

Мониторинг и уведомления: установка пороговых значений (предупреждение, критическое значение, значение восстановления) для метрик, настройка уведомлений о превышении, маршрутизация получателей уведомлений по уровню важности

Интеграция с системами уведомлений: Мессенджеры (Telegram/e-mail), Triafly BI (экспорт для расширенной аналитики), SIEM (security information and event management, всего лишь передача уведомлений службе syslog).

Интеграция с системами мониторинга: по REST API (OpenAPI, тестирование на странице Swagger UI)

Интеграция со службами каталогов по протоколу LDAP: любые, если известны параметры, в документации даны примеры для FreeIPA, AldPro, Active Directory

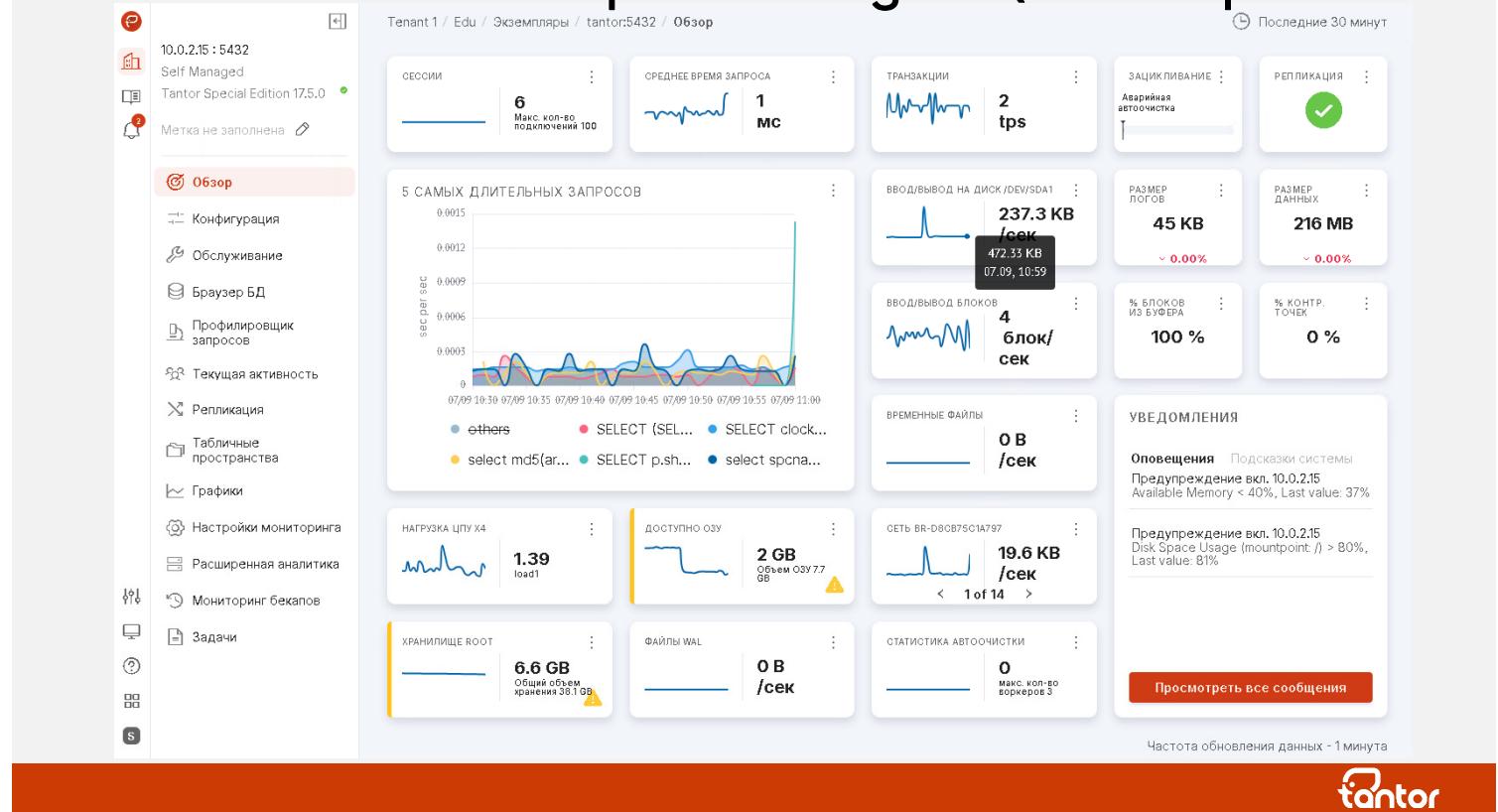
Управление кластерами Patroni: просмотр и изменение параметров, мониторинг репликации и событий failover

Интеграция с системами резервирования: RuBackup и Backman

Аналитика: анализ диагностических журналов PostgreSQL

Управление Tantor xData

Работа с экземплярами PostgreSQL: Обзор



The screenshot shows the PostgreSQL instance overview dashboard. On the left, a sidebar menu lists 13 items: Конфигурация, Обслуживание, Браузер БД, Профилировщик запросов, Текущая активность, Репликация, Табличные пространства, Графики, Настройки мониторинга, Расширенная аналитика, Мониторинг бэкапов, Задачи, and a pinned note: Метка не заполнена.

The main dashboard area contains several cards:

- СЕССИИ:** 6 (Макс. кол-во подключений 100)
- СРЕДНЕЕ ВРЕМЯ ЗАПРОСА:** 1 мс
- ТРАНЗАКЦИИ:** 2 tps
- ЗАСИКЛИВАНИЕ:** Аварийная автоочистка
- РЕПЛИКАЦИЯ:**
- 5 САМЫХ ДЛИТЕЛЬНЫХ ЗАПРОСОВ:** A line chart showing query duration over time. Legend: ethers (blue), SELECT (SEL...) (red), SELECT clock... (dark blue), select md5(ar... (yellow), SELECT p.sh... (cyan), select spcna... (dark blue).
- ВВОД/ВЫВОД НА ДИСК /DEV/SDA1:** 237.3 KB/сек (472.35 KB, 07.09, 10:59)
- РАЗМЕР ЛОГОВ:** 45 KB
- РАЗМЕР ДАННЫХ:** 216 MB
- ВВОД/ВЫВОД БЛОКОВ:** 4 блок/сек
- % БЛОКОВ ИЗ БУФЕРЯ:** 100 %
- % КОНТР. ТОЧЕК:** 0 %
- ВРЕМЕННЫЕ ФАЙЛЫ:** 0 B/сек
- УВЕДОМЛЕНИЯ:** Оповещения: Подсказки системы, Предупреждение вкл. 10.0.2.15 Available Memory < 40%, Last value: 37%.
- НАГРУЗКА ЦПУ X4:** 1.39 (load1)
- ДОСТУПНО ОЗУ:** 2 GB (Объем ОЗУ 77 GB,
- СЕТЬ BR-D8CB75C1A797:** 19.6 KB/сек (1 of 14)
- ХРАНИЛИЩЕ ROOT:** 6.6 GB (Общий объем хранения 38.1 GB,
- ФАЙЛЫ WAL:** 0 B/сек
- СТАТИСТИКА АВТООЧИСТИКИ:** 0 (макс. кол-во воркеров 3)

At the bottom right: **Просмотреть все сообщения** (View all messages) and Частота обновления данных - 1 минута (Data refresh rate - 1 minute).

Работа с экземплярами PostgreSQL: Обзор

Страница "Обзор" экземпляра - основная, с которой начинают работу с экземпляром. На ней отображаются "плитки" (прямоугольные элементы графического интерфейса) с основными показателями работы экземпляра PostgreSQL.

В левой части показывается меню из 13 пунктов (**для реплик PostgreSQL пункт "Обслуживание" не показывается**) в 6 версии Платформы. В 5 версии Платформы было 12 пунктов меню, в 6 версии в меню был добавлен пункт: "Мониторинг бэкапов".

Меню закреплено. Если нужно спрятать меню, чтобы освободить пространство на экране, нужно кликнуть на стрелку в прямоугольнике, которая находится сверху справа на плитке меню:

В предыдущих версиях Платформы для "схлопывания" меню до пиктограмм использовался значок кнопки, который располагался внизу меню:

ЗАДАЧИ



Набор "плиток" на странице "Обзор экземпляра" нельзя изменить. Это можно сделать на странице "Графики", которая изначально пуста.

<https://docs.tantorlabs.ru/tp/6.1/instances/overview.html>

Расширение pg_stat_statements

Расширение pg_stat_statements

Для того, чтобы в плашках "5 САМЫХ ДЛИТЕЛЬНЫХ ЗАПРОСОВ" и "СРЕДНЕЕ ВРЕМЯ ЗАПРОСА" отображались данные, нужно, чтобы в параметре `shared_preload_libraries` была загружена библиотека `pg_stat_statements` и в мониторимых базах данных было установлено расширение: `create extension pg_stat_statements;`

При добавлении экземпляра в Агент Платформы, проверяется наличие расширений:

```
root@tantor:~# pmaagent instances add --name 'Patroni1' --environment 'SERVER' --type 'PostgreSQL' --db-host '10.0.2.15' --db-port '5434' --db-user 'pma_user' --db-pass 'pma_user' --db-admin-user 'postgres' --db-admin-pass 'postgres' --patroni-cluster-hosts '172.18.0.2,172.18.0.7,172.18.0.8' --patroni-user 'appuser' --patroni-password 'appass' --patroni-port '8008' --confirm
#####
#          PMA Agent. Version 6.0.0.          #
#          Add PostgreSQL and Patroni instances #
#          Local date/time: Thursday, 02-Oct-25 00:14:20 MSK #
#####
...
- Please make sure the instance of PostgreSQL that is listening the port [5434]
- is properly configured, and its client authentication (pg_hba) is consistent.
- Testing database connection using provided host [10.0.2.15]... [OK]
- Checking if the PostgreSQL instance is in recovery mode... [OK]
- Checking if the [postgresql-contrib] packaged is installed..... [OK]
- Checking if the [pg_stat_statements] extension is installed..... [WARN]
- Could not detect availability of the extension [pg_stat_statements].
Need to create the extension.
- Checking if the [pg_stat_statements] extension is installed..... [WARN]
- Could not detect availability of the extension [pg_stat_statements].
Need to create the extension.
- Testing connection to database with provided parameters... [OK]
...
```

Набор "плиток" на странице "Обзор экземпляра" нельзя изменить. Это можно сделать на странице "Графики", которая изначально пуста.

По умолчанию, отображаются актуальные данные за 30 минут. Интервал можно изменить: можно выбрать интервалы от 30 минут до 7 дней.

Выбрать можно только интервалы: 30 минут, час, 3, 8, 12, 24 часа, 7 дней. Другие интервалы указать нельзя.

Работа с экземплярами: Графики

10.0.2.15 : 5432
Self Managed
Tantor Special Edition 17.5.0

Метка не заполнена

Обзор
Конфигурация
Обслуживание
Браузер БД
Профильтровщик запросов
Текущая активность
Репликация
Табличные пространства
Графики
Настройки мониторинга

Tenant 1 / Edu / Экземпляры / tantor:5432 / Графики Последние 30 минут

СИСТЕМА	POSTGRES	ПЛАТФОРМА
<input checked="" type="checkbox"/> Loadaverage	<input checked="" type="checkbox"/> Transactions per second	<input checked="" type="checkbox"/> Tablespace size
<input checked="" type="checkbox"/> CPU Usage	<input checked="" type="checkbox"/> Sessions	<input type="checkbox"/> Blocks Hit vs Read per second
<input type="checkbox"/> Context switches and forks per second	<input type="checkbox"/> Checkpoints	<input checked="" type="checkbox"/> WAL Files size
<input type="checkbox"/> Memory usage	<input type="checkbox"/> WAL Files written rate	<input type="checkbox"/> Written buffers
<input type="checkbox"/> Swap usage	<input checked="" type="checkbox"/> Waiting Locks	<input type="checkbox"/> Log size
<input checked="" type="checkbox"/> Filesystems size	<input type="checkbox"/> Temporary Files Count	<input type="checkbox"/> Autovacuum statistic
<input type="checkbox"/> Filesystems usage	<input type="checkbox"/> Tuples per second	<input type="checkbox"/> Deadlocks
<input type="checkbox"/> Network	<input type="checkbox"/> Wraparound	<input type="checkbox"/> Log file number
<input type="checkbox"/> Disk I/O	<input type="checkbox"/> Replication lag	
<input type="checkbox"/> Processes		

Предустановленные настройки: [Все](#) [Производительность](#) [Блокировки](#) [Размер](#)

Применить

tantor

22

Работа с экземплярами: Графики

Графики метрик показываются один за другим. Размер графиков не настраивается. Выбор метрик для отображения на странице "Графики" сохраняется до закрытия окна браузера, то есть до завершения веб-сессии.

Метрики разбиты на три группы: СИСТЕМА, POSTGRES, ПЛАТФОРМА. Список метрик не меняется.

По умолчанию, отображаются актуальные данные за 30 минут. Интервал можно изменить, как и на странице "Обзор":

можно выбрать интервалы от 30 минут до 7 дней.

В дополнение, можно выбрать дату и время, с которого начинается отрисовка на графиках с точностью до минуты.

Дату можно выбрать **не дальше, чем на 7 дней** от текущей даты.

От 01/09/2025 00:00 До 01/09/2025 08:00

Последние 30 минут	От *	01/09/2025 00:00
Последний час	Период *	8 часов
Последние 3 часа		
Последние 8 часов		
Последние 12 часов		
Последний день		
Последние 7 дней		

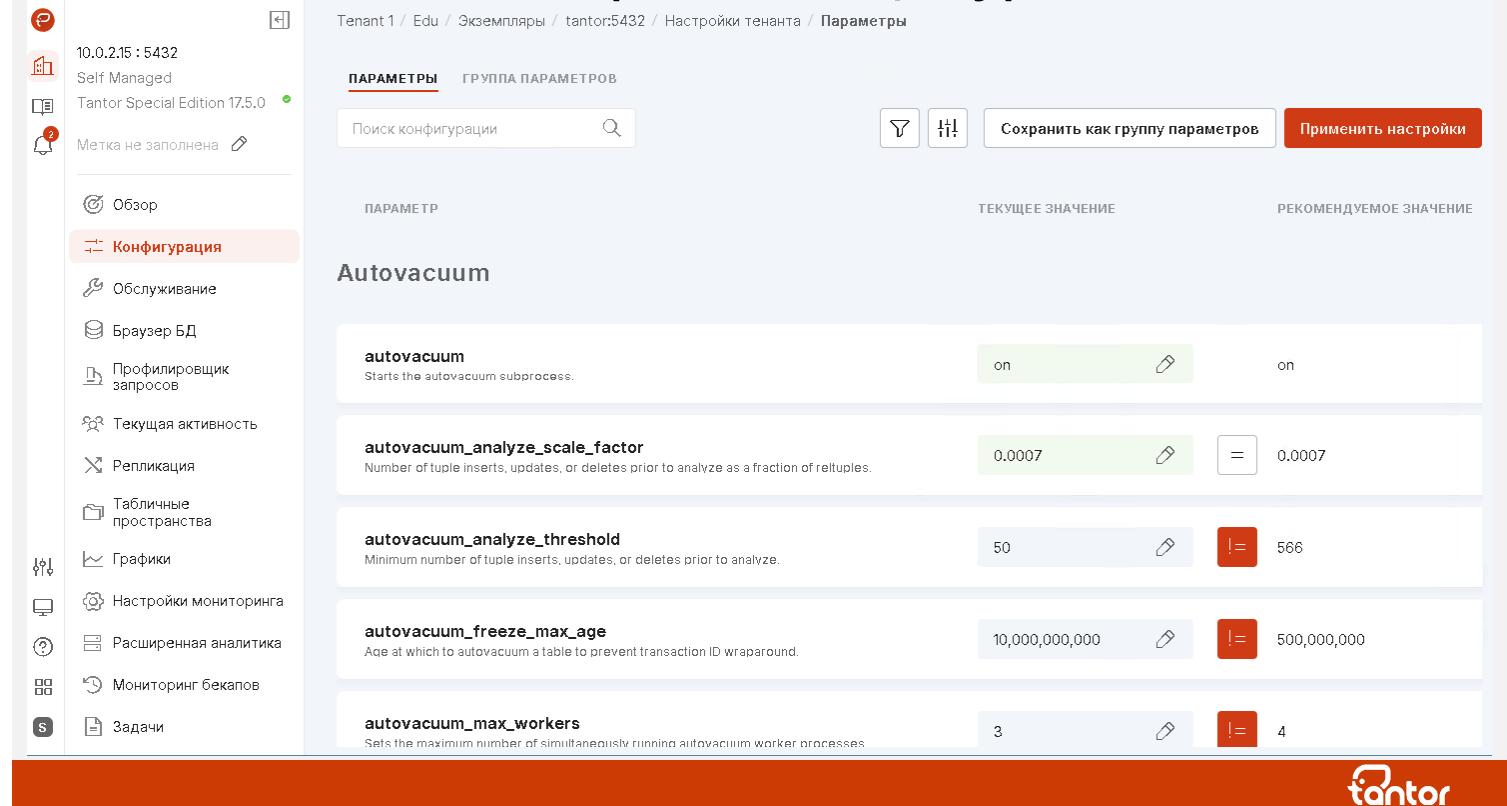
Применить

Предназначение этих метрик - расследование инцидентов, произошедших в прошлом. При возникновении проблемы, можно воспользоваться графиками для сопоставления параметров на том интервале, когда возникла проблема.

Платформа может сохранять данные метрик дольше 7 дней. Настройка очистки накопленных данных делается на странице https://хост_Платформы/pg-monitor/cleardbfile

<https://docs.tantorlabs.ru/tp/6.1/instances/charts.html>

Работа с экземплярами: Конфигурация



The screenshot shows the Tantor PostgreSQL configuration interface. On the left, a sidebar lists various management options: Обслуживание, Браузер БД, Профилировщик запросов, Текущая активность, Репликация, Табличные пространства, Графики, Настройки мониторинга, Расширенная аналитика, Мониторинг бекапов, and Задачи. The 'Конфигурация' option is selected. The main area is titled 'ПАРАМЕТРЫ' (PARAMETERS) and 'ГРУППА ПАРАМЕТРОВ' (PARAMETER GROUP). It features a search bar, a 'Сохранить как группу параметров' (Save as parameter group) button, and a 'Применить настройки' (Apply settings) button. The 'Autowacuum' group is selected, displaying the following parameters:

ПАРАМЕТР	ТЕКУЩЕЕ ЗНАЧЕНИЕ	РЕКОМЕНДУЕМОЕ ЗНАЧЕНИЕ
autovacuum Starts the autovacuum subprocess.	on	on
autovacuum_analyze_scale_factor Number of tuple inserts, updates, or deletes prior to analyze as a fraction of rextuples.	0.0007	0.0007
autovacuum_analyze_threshold Minimum number of tuple inserts, updates, or deletes prior to analyze.	50	566
autovacuum_freeze_max_age Age at which to autovacuum a table to prevent transaction ID wraparound.	10,000,000,000	500,000,000
autovacuum_max_workers Sets the maximum number of simultaneously running autovacuum worker processes.	3	4

Учебный курс "Платформа Tantor 6" Тантор Лабс © 2025 www.tantolabs.ru

tantor

23

Работа с экземплярами: Конфигурация

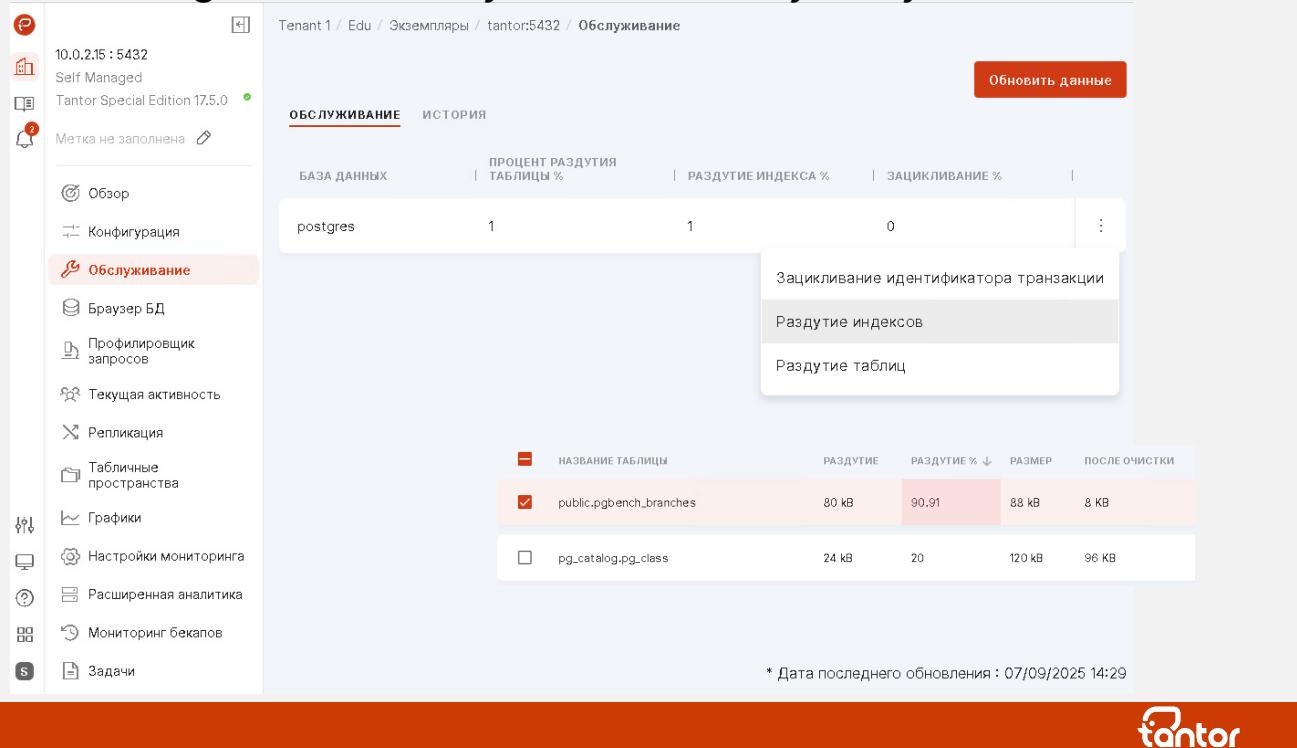
На странице "Конфигурация" показываются параметры экземпляра PostgreSQL и даются рекомендации по установке значений параметров. Если текущее значение параметра отличается от рекомендуемого конфигуратором, встроенным в Платформу Tantor, то на это указывает значок "**!=**".

Можно создать группу параметров, связать её с несколькими экземплярами и применять значения параметров к ним.

<https://docs.tantolabs.ru/tp/6.1/instances/configurations.html>

Работа с экземплярами: Обслуживание

- у реплик PostgreSQL этот пункт меню отсутствует



Название таблицы	Раздутье	Раздутье %	Размер	После очистки	Строки	Последняя автоочистка
public.pgbench_branches	80 kB	90.91	88 kB	8 kB	1	-
pg_catalog.pg_class	24 kB	20	120 kB	96 kB	454	-
pg_catalog.pg_attribute	32 kB	6.25	512 kB	480 kB	3480	-
pg_catalog.pg_depend	8192 bytes	6.25	128 kB	120 kB	1989	-
public.pgbench_accounts	816 kB	5.93	13 MB	13 MB	100000	-
pg_catalog.pg_proc	40 kB	4.1	4 kB	3363 kB	3363	-
pg_catalog.pg_description	16 kB	4.1	4 kB	5165 kB	5165	-
public.pgbench_history	48 kB	1.3	4 kB	69376 kB	69376	-

Работа с экземплярами: Обслуживание

Для реплик PostgreSQL пункт "Обслуживание" не показывается. По наличию этого пункта можно отличить ведущий (мастер, основной, primary) кластер от реплики (резервного, standby).

На странице "Обслуживание" всего три действия по обслуживанию баз данных:

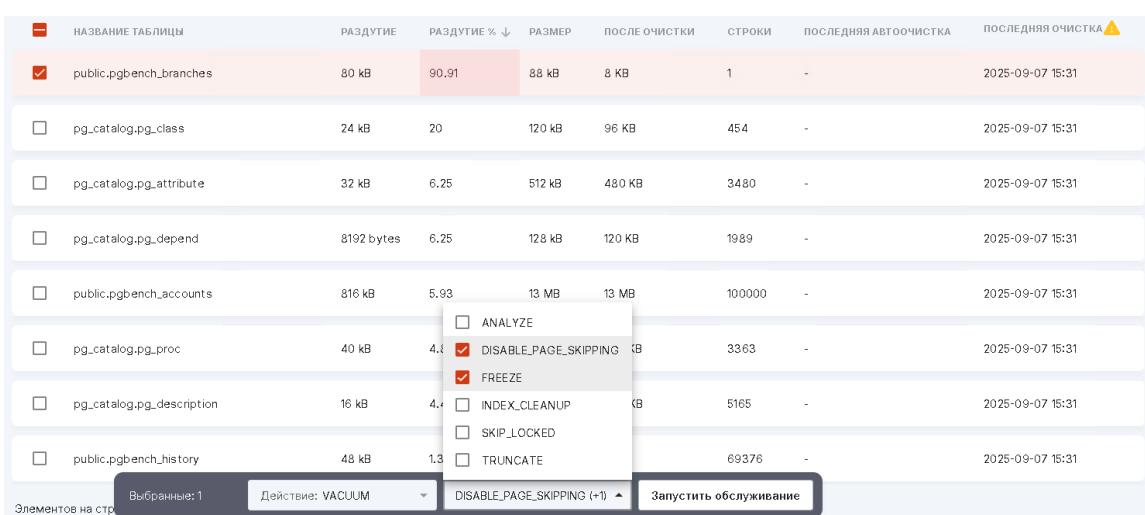
- 1) Зацикливание идентификатора транзакции,
- 2) Раздутье индексов,
- 3) Раздутье таблиц.

Для таблиц можно выполнить VACUUM, VACUUM FULL, ANALYZE.

Для индексов REINDEX и REINDEX CONCURRENTLY.

Для зацикливания VACUUM FULL (опционально ANALYZE), VACUUM FREEZE, INDEX_CLEANUP, SKIP_LOCKED, DISABLE_PAGE_SKIPPING, TRUNCATE.

Основная задача страницы - показать список объектов и процент раздутья:



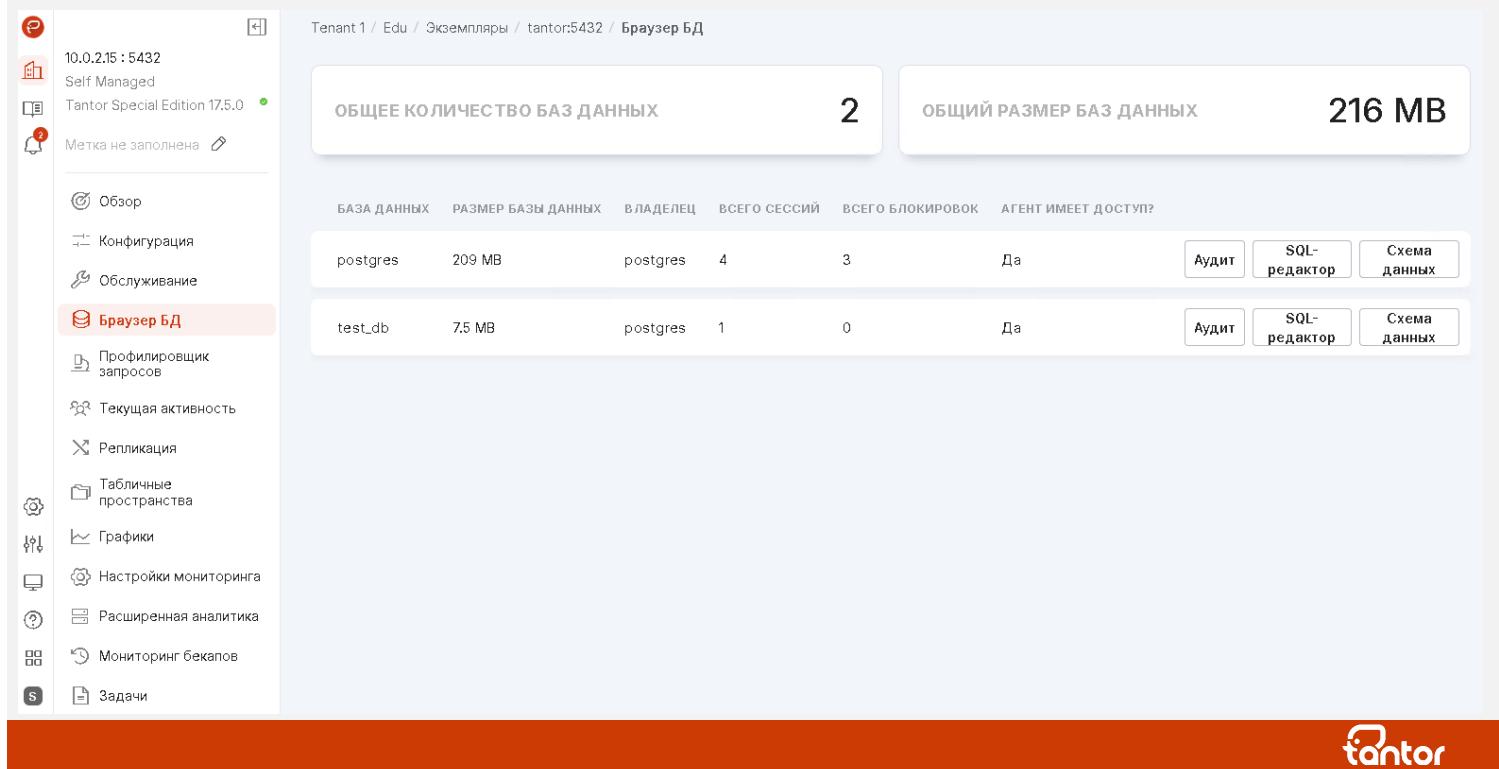
Название таблицы	Раздутье	Раздутье %	Размер	После очистки	Строки	Последняя автоочистка
public.pgbench_branches	80 kB	90.91	88 kB	8 kB	1	-
pg_catalog.pg_class	24 kB	20	120 kB	96 kB	454	-
pg_catalog.pg_attribute	32 kB	6.25	512 kB	480 kB	3480	-
pg_catalog.pg_depend	8192 bytes	6.25	128 kB	120 kB	1989	-
public.pgbench_accounts	816 kB	5.93	13 MB	13 MB	100000	-
pg_catalog.pg_proc	40 kB	4.1	4 kB	3363 kB	3363	-
pg_catalog.pg_description	16 kB	4.1	4 kB	5165 kB	5165	-
public.pgbench_history	48 kB	1.3	4 kB	69376 kB	69376	-

В разделе нет возможности послать задачу по расписанию, только немедленно.

Для того, чтобы база данных появилась в списке, её мониторинг нужно включить в разделе меню "Настройки мониторинга".

<https://docs.tantorlabs.ru/tp/6.1/instances/maintenance.html>

Работа с экземплярами: Браузер БД



The screenshot shows the Tantor Special Edition 17.5.0 interface. The left sidebar has a 'Браузер БД' (Database Browser) section highlighted. The main area displays two databases: 'postgres' (209 MB, owner 'postgres', 4 sessions, 3 locks, Audit, SQL-редактор, Схема данных) and 'test_db' (7.5 MB, owner 'postgres', 1 session, 0 locks, Audit, SQL-редактор, Схема данных). The top navigation bar shows 'Tenant 1 / Edu / Экземпляры / tantor:5432 / Браузер БД'.

БАЗА ДАННЫХ	РАЗМЕР БАЗЫ ДАННЫХ	ВЛАДЕЛЕЦ	ВСЕГО СЕССИЙ	ВСЕГО БЛОКИРОВОК	АГЕНТ ИМЕЕТ ДОСТУП?
postgres	209 MB	postgres	4	3	Да
test_db	7.5 MB	postgres	1	0	Да

Учебный курс "Платформа Tantor 6" Тантор Лабс © 2025 www.tantolabs.ru

Работа с экземплярами: Браузер БД

На странице "Браузер БД" три кнопки у каждой из баз данных: Аудит, SQL-редактор, Схема данных.

https://docs.tantolabs.ru/tp/6.1/instances/DB_browser.html

Браузер БД -> Аудит

Tenant 1 / Edu / Экземпляры / tantor:5432 / Браузер БД / postgres

🕒 Последние 30 минут

РАЗМЕР БАЗЫ ДАННЫХ	ВСЕГО ТАБЛИЦ	ВСЕГО ФУНКЦИЙ	ВСЕГО СХЕМ
281 МВ	7	27	1

Владелец	postgres	Расширения	plpgsql(1.0), pg_stat_statements(1.11), pg_qualstats(2.1...)
Кодировка	UTF8	Всего индексов	9
Табличное пространство	pg_default	Индексы	hash(1), btree(8)
Ограничение по кол-ву подключений	No Limit	функции	c(23), sql(3), plpgsql(1)

ПОЛУЧЕННЫЕ СТРОКИ :

ВОЗВРАЩЕННЫЕ СТРОКИ :

ВЫПОЛНЕНО ТРАНЗАКЦИЙ :

ОТКАЧЕНО ТРАНЗАКЦИЙ :

ВЗАЙМНЫЕ БЛОКИРОВКИ :

ВСТАВЛЕННЫЕ СТРОКИ :

ОБНОВЛЕННЫЕ СТРОКИ :

УДАЛЕННЫЕ СТРОКИ :

ЗАПИСАНО ВРЕМЕННЫХ ФАЙЛОВ :

СОЗДАНО ВРЕМЕННЫХ ФАЙЛОВ :

% БЛОКОВ ИЗ БУФЕРА :

ЗАЦИКЛИВАНИЕ Аварийная автоочистка :

Последний собранный 21/10/2025 18:35

[Скачать отчет](#) [Обновить данные](#)

ПРОВЕРКА СОСТОЯНИЯ

- Самые большие таблицы
- Таблицы с индексами максимального размера
- Топ записываемых таблиц
- Топ читаемых таблиц
- Таблицы с наибольшим раздутьем
- Таблицы с наибольшим кол-вом последовательных сканирований
- Таблицы с недопустимыми индексами
- Таблицы с неиспользуемыми индексами
- Таблицы с кандидатами для частичных индексов
- Индексы таблиц с значениями NULL > 50%
- Таблицы с избыточными индексами
- Таблицы FK без индекса
- Таблицы у которых тип данных внешнего ключа отличен от исходного
- Таблицы с низким коэффициентом обновления

tantor

26

Работа с экземплярами: Браузер БД -> Аудит

На странице "Браузер БД" три кнопки у каждой из баз данных: Аудит, SQL-редактор, Схема данных.

Страница Аудит - одна из наиболее полезных. На этой странице проверяется структура объектов базы данных, выдаются типичные проблемы. Даются рекомендации по устранению проблем. Можно запустить команды Analyze и Vacuum Full.

Tenant 1 / Edu / Экземпляры / tantor:5432 / Браузер БД / postgres / stock_items

ОБЗОР **СТАТИСТИКА**

ВЛАДЕЛЕЦ	ТИП ТАБЛИЦЫ	OID ТАБЛИЦЫ	СТРОКА (ПРОГНОЗИРУЕМАЯ)	КОЭФФ. ЗАПОЛНЕНИЯ
postgres	Normal	16,649	1,607,333	100
НАСЛЕДУЕТ ТАБЛИЦЫ	ИМЕЕТ TOAST?	СИСТЕМНАЯ ТАБЛИЦА?	ТАБЛИЧНОЕ ПРОСТРАНСТВО	КАРТА ВИДИМОСТИ%
false	true	false	pg_default	0.15

7 из 10 Найдены проблемы в этой таблице

Таблицы у которых тип данных внешнего ключа отличен от исходного

Описание	Рекомендации
Foreign Key name: stock_items_fk01 • Field(s): order_items_1_id • Field(s) type(s): int4 • Referenced table: stock_items • Referenced table field(s): id • Referenced table field(s) type(s): int8	Необходимо изменить типы полей внешних ключей, чтобы избежать проблем с производительностью и приведением типов в запросах. Вы можете изменить типы полей внешнего ключа с помощью команды: ALTER TABLE stock_items ALTER COLUMN order_items_1_id TYPE int8.

Последний собранный 21/10/2025 18:35

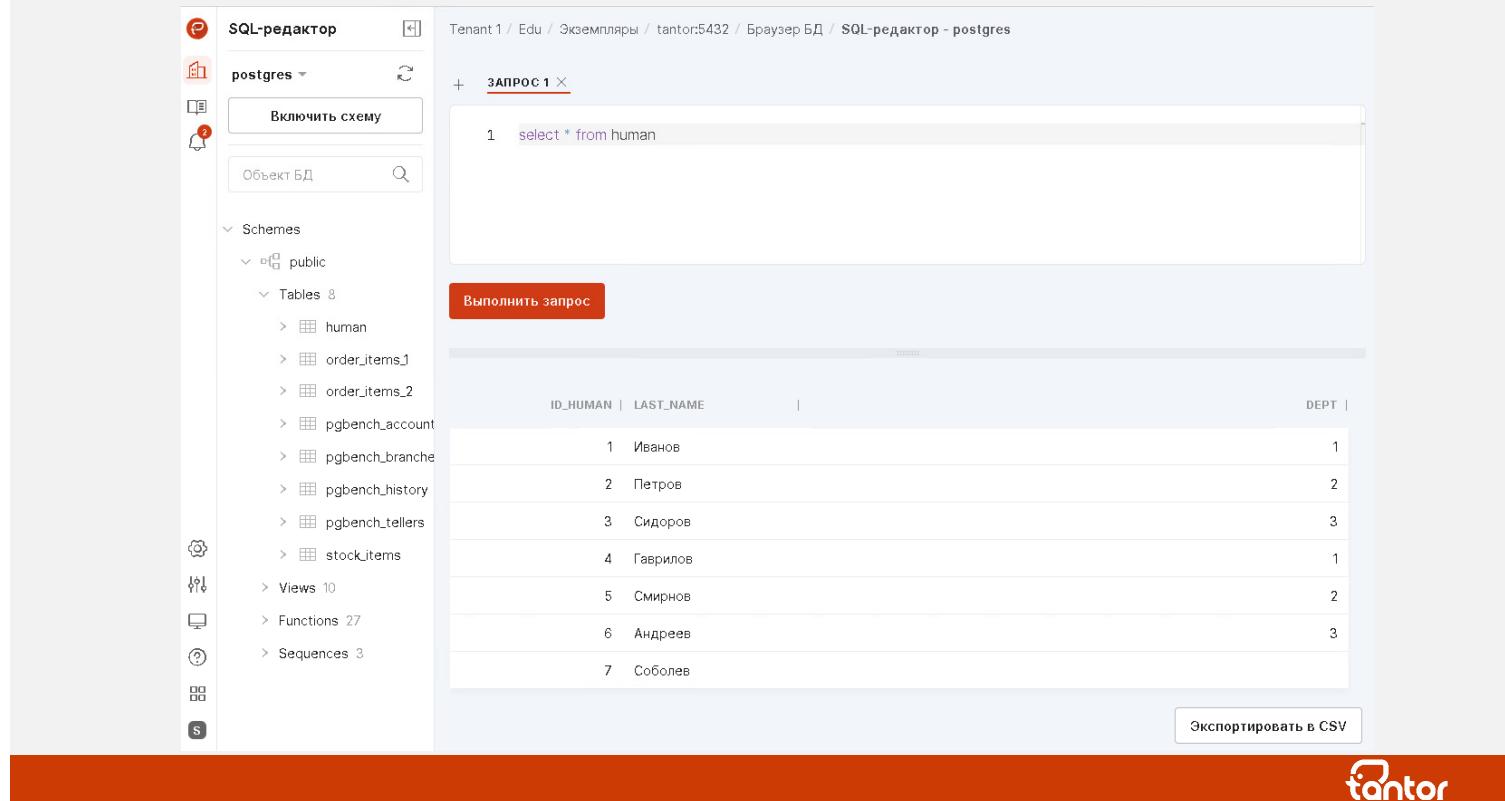
[Скачать отчет](#) [Обновить данные](#)

ПРОВЕРКА СОСТОЯНИЯ

- Самые большие таблицы
- Таблицы с индексами максимального размера
- Топ записываемых таблиц
- Топ читаемых таблиц
- Таблицы с наибольшим раздутьем
- Таблицы с наибольшим кол-вом последовательных сканирований
- Таблицы с недопустимыми индексами
- Таблицы с неиспользуемыми индексами
- Таблицы с кандидатами для частичных индексов
- Индексы таблиц с значениями NULL > 50%
- stock_items** Таблицы с избыточными индексами

Ваш отчет формируется... отличен от исходного

Браузер БД -> SQL-редактор и Схема



The screenshot shows the Tantor PostgreSQL browser interface. On the left, a sidebar displays the database schema. It shows a 'Schemas' section with 'public' expanded, revealing 'Tables' (8), 'Views' (10), 'Functions' (27), and 'Sequences' (3). A 'postgres' connection is selected. A 'Включить схему' (Enable schema) button is visible. On the right, the main area has a title 'ЗАПРОС 1' (Query 1) and contains the SQL command 'select * from human'. Below the command is a 'Выполнить запрос' (Execute query) button. The results are displayed in a table with columns 'ID_HUMAN', 'LAST_NAME', and 'DEPT'. The data is as follows:

ID_HUMAN	LAST_NAME	DEPT
1	Иванов	1
2	Петров	2
3	Сидоров	3
4	Гаврилов	1
5	Смирнов	2
6	Андреев	3
7	Соболев	

At the bottom right, there is a 'Экспортировать в CSV' (Export to CSV) button. The footer includes the Tantor logo and the text 'Учебный курс "Платформа Tantor 6" Тантор Лабс © 2025 www.tantolabs.ru'.

Браузер БД -> SQL-редактор и Схема

На странице SQL-редактора можно выполнять команды SQL, набирая их в окне браузера. При наборе команд есть автоматические подсказки.

На этих страницах можно переключаться между SQL-редактором и Схемой.

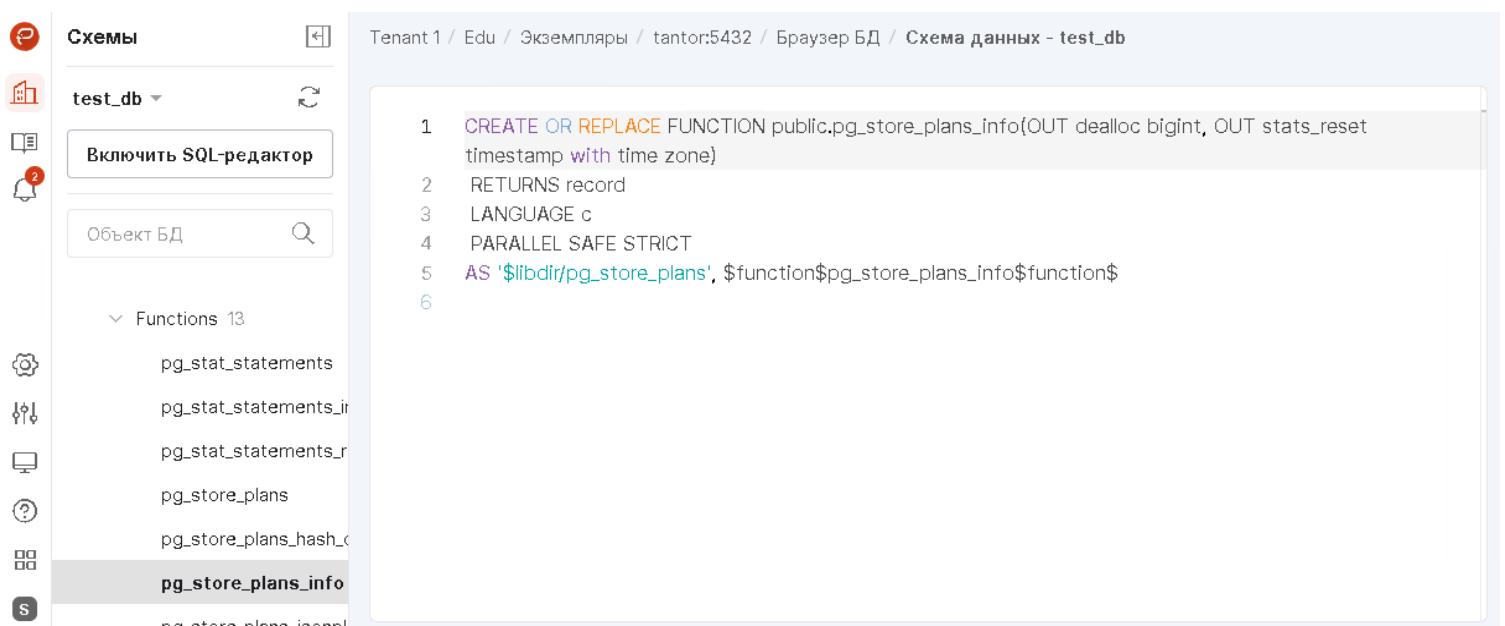
Схема показывает структуру выбранной таблицы, последовательности или команду создания представления, подпрограммы (функции или процедуры).

На странице в двух местах показывается название базы данных, с которой создана сессия.

Пользователь, под которым была создана сессия на странице не показывается.

Значком "+" можно открыть несколько окон для редактирования и выполнения запросов.

Значком "x" можно закрыть окна ненужных запросов. Окно единственного запроса не закрывается.



The screenshot shows the Tantor PostgreSQL browser interface. On the left, a sidebar displays the database schema. It shows a 'Functions' section with '13' items, including 'pg_stat_statements', 'pg_stat_statements_it', 'pg_stat_statements_r', 'pg_store_plans', 'pg_store_plans_hash', and 'pg_store_plans_info'. The 'pg_store_plans_info' function is currently selected. A 'Схемы' (Schemas) button is visible. On the right, the main area shows the function definition:

```
1 CREATE OR REPLACE FUNCTION public.pg_store_plans_info(OUT dealloc bigint, OUT stats_reset timestamp with time zone)
2 RETURNS record
3 LANGUAGE c
4 PARALLEL SAFE STRICT
5 AS '$libdir/pg_store_plans', $function$pg_store_plans_info$function$
```

Работа с экземплярами: Профилировщик запросов

The screenshot shows the Tantor Query Profiler interface. On the left is a sidebar with various monitoring and configuration options. The main area features a line graph titled 'Среднее время запроса' (Average Request Time) showing the trend over time. A callout box highlights a peak at 7/09 22:02 with a value of 3.05 ms. Below the graph is a table listing queries with metrics like duration, calls, and CPU time. The table is sorted by 'Самые высокие' (Highest). At the bottom, there are pagination controls and a search bar.

tantor

28

Работа с экземплярами: Профилировщик запросов

При клике на строку запроса появится страница с деталями выполнения запроса. Под графиком есть 7 точек, кликнув на которые можно переключить график на отображение:

Time Query/Second, Calls/Second, Rows/Second, CPU Time/ Second, IO Time/Second, Durtied Blocks/Second, Temp Blocks(Write)/Second.

Есть выбор интервала отображения от 30 минут до 7 дней:

This screenshot shows a detailed view of a query's execution. On the left is a sidebar with monitoring and configuration options. The main area has a 'STATISTIKA' tab selected, showing a 'CALLS/SECOND' graph with several sharp peaks. To the right is a 'Планы' (Plans) section. A large panel on the right displays the query's text and its execution plan. The query text is:

```
SELECT
  clock_timestamp(),
  *
FROM
  pg_stat_get_activity ($1)
```

Профильтр запросов -> Планы

- В Платформу Tantor встроен полнофункциональный анализатор запросов Tensor
- Запросы не передаются на внешние сервера

Профильтр запросов -> Планы

Для того, чтобы планы показывались, нужно, чтобы в кластере были установлены библиотеки `pg_stat_statements` и `pg_store_plans`:

```
alter system set shared_preload_libraries = pg_stat_statements, pg_qualstats, pg_store_plans, pg_prewarm, pg_stat_kcache, auto_explain ;
```

В базе данных были установлены расширения:

```
create extension if not exists pg_stat_statements;
create extension if not exists pg_store_plans;
```

Показываться будут только запросы, длительность которых превышает значение параметра `pg_store_plans.min_duration`. Значение можно установить командой:

```
alter system set pg_store_plans.min_duration = '0s';
```

Профильтр обеспечивает мониторинг параметров выполнения запросов и их планов в выбранном временном промежутке с использованием скользящего окна. Его цель - проведение анализа и выявление проблемных запросов в базе данных. Профильтр не только отслеживает параметры выполнения, но и проводит анализ планов запросов, предоставляем подсказки для оптимизации. Это инструмент, который помогает идентифицировать и решать возможные проблемы с производительностью запросов.

При клике на вкладке "ПЛАНЫ" откроется страница со списком планов:

Работа с экземплярами: Текущая активность

- можно выбирать столбцы, которые отображать:

БАЗА ДАННЫХ	ИСПОЛЬЗУЕТСЯ	ИСПОЛЬЗУЕТСЯ %	МАКС. КОЛ-ВО СЕССИЙ	АКТИВНО	IDLE	IDLE IN TX	IDLE IN TX (ABORTED)	ОТКЛЮЧЕНО	НЕТ ПРИВИЛЕГИЙ	ОЖИДАНИЕ
postgres	3	3	100	0	3	0	0	0	0	0
test_db	1	1	100	0	1	0	0	0	0	0
template1	0	0	100	0	0	0	0	0	0	0

Учебный курс "Платформа Tantor 6" Тантор Лабс © 2025 www.tantolabs.ru

tantor

30

Работа с экземплярами: Текущая активность

Кликнув на строку с именем базы данных, откроется список процессов. На странице три вкладки: ВЫПОЛНЕНИЕ, ОЖИДАНИЕ, ЗАБЛОКИРОВАНО. Показывается текст запроса, который выполняется в сессии. Нажав на кнопку "Завершить" сессия будет прервана вызовом функции `pg_terminate_backend(PID)`. Отображаемые столбцы можно выбирать, сортировать строки по столбцам.

Порядок отображения столбцов можно менять перетаскивая их мышкой.

Кнопкой "Заморозить" отключается автоматическое обновление страницы (раз в 5 секунд).

PID	ИМЯ ПОЛЬЗОВАТЕЛЯ	ЦПУ	ОЗУ %	ЧТЕНИЕ (Б/С)	ЗАПИСЬ (Б/С)	ОЖИДАНИЕ I/O	ОЖИДАНИЕ	СОСТОЯНИЕ	ДЛЯТЕЛЬНОСТЬ	ЗАПРОС
2872532	postgres	0	0.16	0	0	Нет	Нет	idle	19.25 min	<code>alter system set pg_st...</code>
2865576	postgres	0	0.26	0	0	Нет	Нет	Waiting		<code>SELECT * ...</code>
2865575	postgres	0	0.23	0	0	Нет	Нет	Waiting		<code>clock_timestamp() ...</code>

Есть возможность визуализации блокировок в виде дерева, что позволяет легко определить, какая транзакция блокирует другие транзакции.

* Частота обновления данных - 5 сек

Репликация и Табличные пространства

The screenshot shows the Tantor 6 interface with two main sections:

- Replication:** Shows a table with one row for a physical replication slot named 'replica'.
- Tables:** Shows a table with two rows: 'pg_default' and 'pg_global'.

Both sections include a sidebar with the following items:

- 10.0.2.15 : 5432 (Self Managed, Tantor Special Edition 17.5.0)
- Метка не заполнена
- Обзор
- Конфигурация
- Браузер БД

Учебный курс "Платформа Tantor 6" Тантор Лабс © 2025 www.tantolabs.ru



31

Репликация и Табличные пространства

В этих двух пунктах меню отображаются некоторые свойства этих объектов.

Физические реплики мониторятся Платформой Tantor и у них есть свои страницы "Работа с экземплярами". Меню "Репликация" у физических реплик отлично от мастера и показывает "СТАТУС" репликации и адрес, откуда реплика получает журналы.

Пример данных "ИНФОРМАЦИИ О ПОДКЛЮЧЕНИИ", при установленных значениях параметров на реплике:

```
primary_conninfo = 'user=postgres port=5432'  
primary_slot_name = 'replica'
```

По ссылке из поля "Основной экземпляр" можно перейти на страницу управления экземпляром мастера:

The screenshot shows the Tantor 6 interface with the 'Replication' section for Tenant 1:

СТАТУС

Статус:	N/A
Основной экземпляр:	Tantor_Pg (17.5.0):10.0.2.15:5433

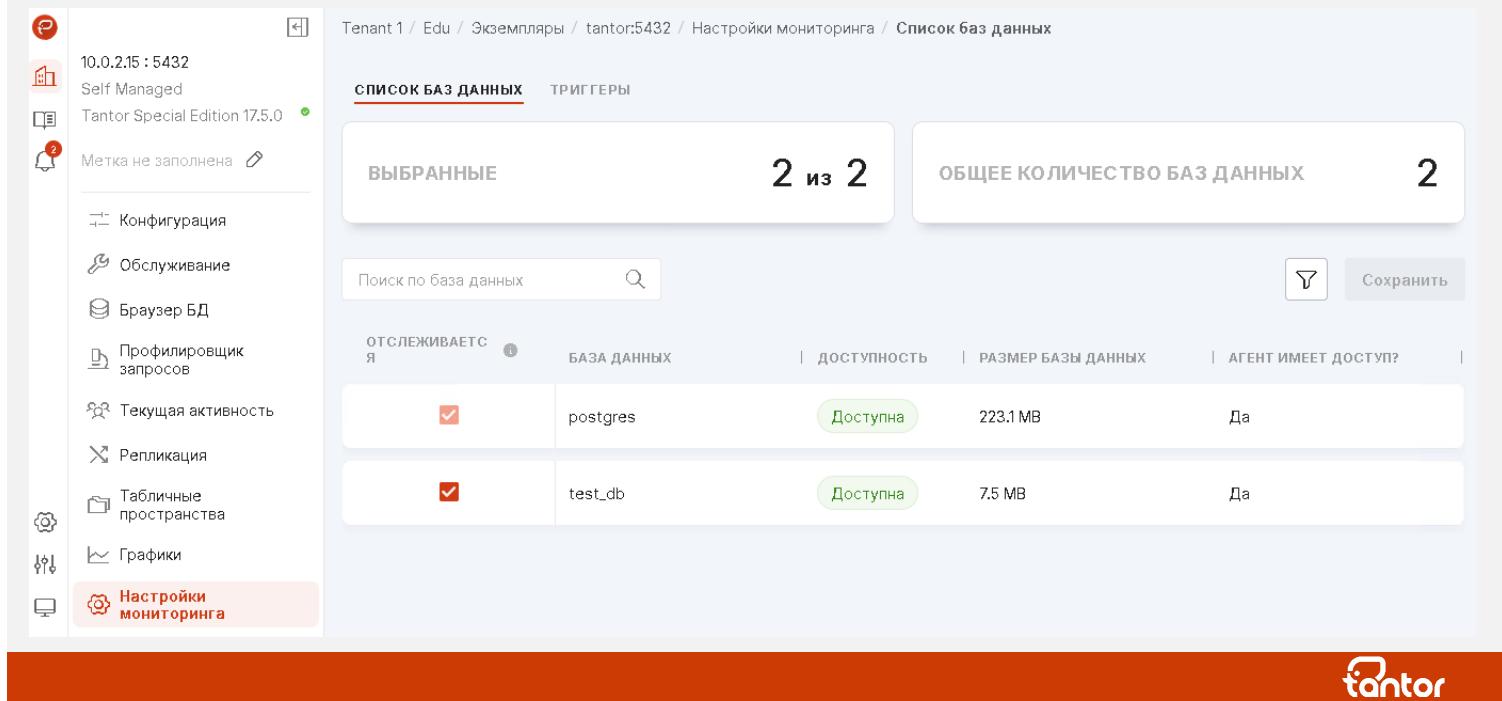
ИНФОРМАЦИЯ О ПОДКЛЮЧЕНИИ:

user:	postgres
passfile:	/var/lib/postgresql/.pgpass
channel_binding:	prefer
dbname:	replication
port:	5432
fallback_application_name:	walreceiver
sslmode:	prefer
sslnegotiation:	postgres
sslcompression:	0
sslcertmode:	allow
sslsni:	1
ssl_min_protocol_version:	TLSv1.2
gssencmode:	prefer
krbsrvname:	postgres
gssdelegation:	0
compression:	off
target_session_attrs:	any
load_balance_hosts:	disable

Sidebar items:

- 10.0.2.15 : 5433 (Self Managed, Tantor Special Edition 17.5.0)
- Метка не заполнена
- Обзор
- Конфигурация
- Браузер БД
- Профиляровщик запросов
- Текущая активность
- Репликация** (highlighted)
- Табличные пространства
- Графики
- Настройки мониторинга

Работа с экземплярами: Настройки мониторинга



The screenshot shows the 'Monitoring Configuration' section of the Tantor interface. On the left, a sidebar lists various monitoring categories: Configuration, Maintenance, Browser DB, Profiler, Current Activity, Replication, Tablespaces, and Monitoring Configuration (which is highlighted in red). The main content area is titled 'Список баз данных' (List of databases) and 'ТРИГГЕРЫ' (Triggers). It displays two selected databases: 'postgres' and 'test_db', both marked as 'Available' (Доступна) with sizes of 223.1 MB and 7.5 MB respectively. A summary at the top right shows '2 из 2' selected and 'ОБЩЕЕ КОЛИЧЕСТВО БАЗ ДАННЫХ' (Total number of databases) as '2'. A 'Сохранить' (Save) button is visible on the right. The bottom of the page includes a footer with the Tantor logo and the text 'Учебный курс "Платформа Tantor 6" Тантор Лабс © 2025 www.tantolabs.ru'.

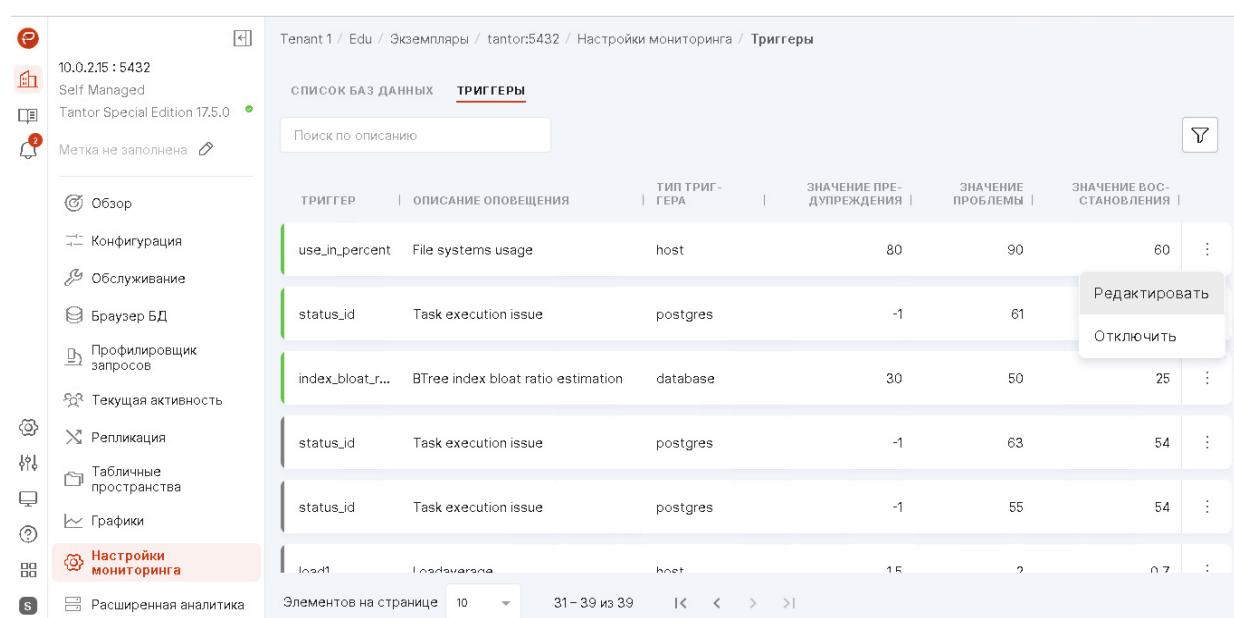
Работа с экземплярами: Настройки мониторинга

На этой странице можно выбрать базы данных, наблюдение за метриками которых будет производиться агентом Платформы Tantor. По умолчанию, выбрана та база данных, которая была указана как база для подключения агента Платформы при его конфигурировании. Остальные базы данных нужно выбирать отдельно. Выбранные базы будут выводиться в пунктах меню: Обслуживание и Браузер БД.

В вкладке "ТРИГГЕРЫ" выдаются 39 метрик, за которыми наблюдает Платформа Tantor. У каждой метрики три пороговых значения: предупреждение, критическое предупреждение, восстановление.

У большинства метрик есть возможность поменять пороговые значения, кликнув на три точки и выбрав "Редактировать".

По умолчанию, мониторинг всех метрик включён. Такие метрики обозначаются зелёной линией. Мониторинг метрик можно отключить. Отключённые метрики обозначаются серой линией:



The screenshot shows the 'Monitoring Configuration' section of the Tantor interface. The sidebar and main layout are identical to the previous screenshot, showing the 'Monitoring Configuration' section. The main content area is titled 'Список баз данных' (List of databases) and 'ТРИГГЕРЫ' (Triggers). It displays a list of triggers with their descriptions, types, and threshold values. The 'status_id' trigger for 'Task execution issue' is selected, showing its details: type 'postgres', threshold values 61, 63, and 54. A context menu is open over this row, with options 'Редактировать' (Edit) and 'Отключить' (Disable). The bottom of the page includes a footer with the Tantor logo and the text 'Учебный курс "Платформа Tantor 6" Тантор Лабс © 2025 www.tantolabs.ru'.

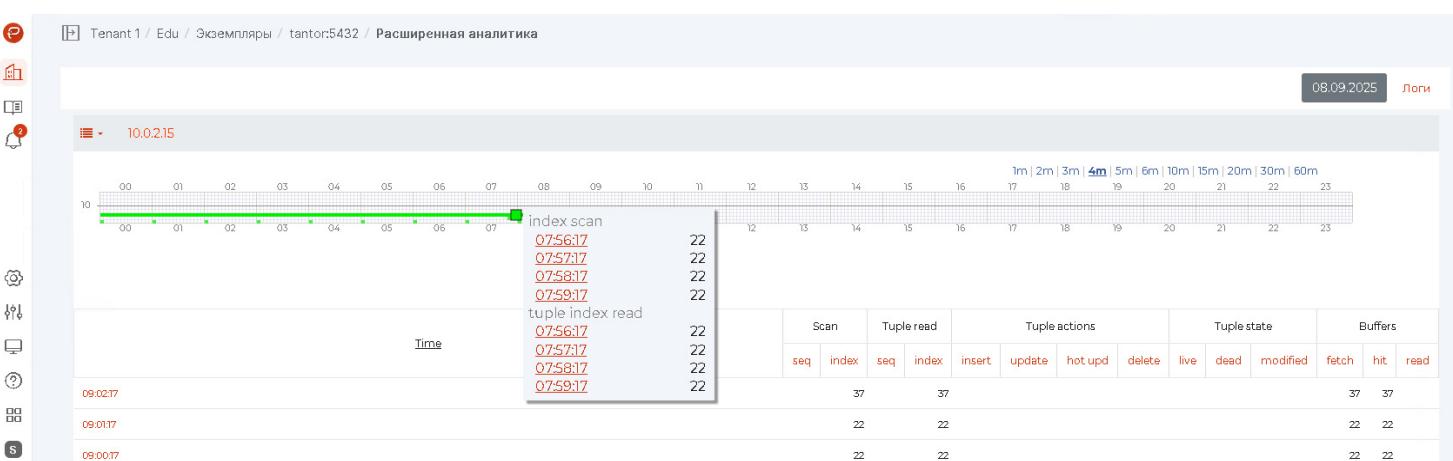
Работа с экземплярами: Расширенная аналитика

Работа с экземплярами: Расширенная аналитика

Анализируется текстовый диагностический лог кластера. На этой странице много ссылок, по которым можно посмотреть то, что выводится в лог кластера, накопительной статистики и то, что собирается расширениями `pg_stat_statements`, `pg_store_plans`, `auto_explain`.

Для сбора данных нужно установить расширения, задать формат названия файла журнала и другие параметры конфигурации кластера PostgreSQL.

Расширенная аналитика разработана российской компанией Тензор и является частью Платформы Tantor.



https://docs.tantorlabs.ru/tp/6.1/instances/pg_monitor/pg_monitor.html

Работа с экземплярами: Задачи

The screenshot shows the Tantor Special Edition 17.5.0 interface. On the left, a sidebar displays system status (10.0.2.15 : 5432, Self Managed, Tantor Special Edition 17.5.0) and navigation links (Overview, Configuration, Monitoring, Browser, Request Profiler). The main area shows a list of tasks for tenant 1. The first task, 'job2', is selected and marked as 'Successful'. The second task, 'job1', is marked as 'Error'. Below the list are buttons for 'Select', 'Enable', 'Disable', 'Run Now', and 'Skip'. A second screenshot shows a detailed view of task 'job2', listing parameters (SELECT 1) and type (sql). The footer includes the Tantor logo and copyright information.

Работа с экземплярами: Задачи

На этой странице можно добавлять задачи, выполняемые по расписанию и указывать имя и пароль пользователя в операционной системе для запуска команд в linux и имя и пароль для запуска SQL-команд в базе данных. После добавления задачи нужно указать действия, которые будут выполняться. Действиями могут быть команды в linux и базе данных. Имя базы данных указывается при добавлении действия.

Добавив действие, можно запустить задачу немедленно. Для того, чтобы на странице со списком задач появились кнопки "Включить, Отключить, Запустить сейчас, Пропустить", нужно галочкой выбрать задачу.

<https://docs.tantorlabs.ru/tp/6.1/instances/tasks.html>

Для страницы конкретной задачи аналогично, чтобы появились кнопки "Включить", "Отключить", нужно выбрать галочкой действие.

<https://docs.tantorlabs.ru/tp/6.1/instances/actions.html>

Со страницы конкретной задачи можно перейти в "Журнал событий", кликнув на такой же надпись.

The screenshot shows the 'Event Log' page for event ID 286. The left panel displays event details: created on 08/09/2025 12:27:07, type TASK, resource INSTANCE, resource ID 3, and created by Unknown. The right panel is titled 'Информация о событии' (Information about the event) and lists event parameters: Action 1 (Id: 5f113e92-ffc1-4690-843c-943db49c8f4a, Name: job1, StatusID: 54, StatusName: Success, Command: select 1, DbType: postgres, Type: sql, OnError: fail, IsDisabled: false, ExecutionResult: [1]), and Sql (5f113e92-ffc1-4690-843c-943db49c8f4a, job1, 54, Success, select 1, postgres, sql, fail, false, [1]). The footer includes the Tantor logo and copyright information.

Модули Платформы Tantor

The screenshot shows the 'Modules' page of the Tantor Platform. On the left, there is a vertical toolbar with icons for search, refresh, and other system functions. The main area displays a grid of modules and their sub-items. The modules are:

- Анализ плана запросов
- Анонимайзер
- SQL Редактор
- Схема данных
- Группы параметров
- Резервные копии
- Задачи
- Схема данных
- Группы параметров
- Резервные копии
- Задачи
- XData
- Swagger
- XData
- Swagger

Below the grid, there is a section for 'External modules' containing 'XData' and 'Swagger'. At the bottom, there is a 'All modules' section. The 'SQL Редактор' module is currently selected, indicated by a blue border around its card.

Модули Платформы Tantor

Для открытия списка модулей нужно кликнуть на значок четыре квадрата слева. Откроется всплывающее меню со списком модулей. Если выбрать "Все модули", то откроется страница с плашками модулей. В версии 6.1 Платформы 9 модулей.

"SQL Редактор, Схема данных, Группы параметров, Мониторинг бэкапов, Задачи" могут открываться через страницу экземпляра PostgreSQL.

Модуль "Анализ планов запросов" позволяет вставить план запроса и визуализировать его. План не передаётся на внешние сервера, анализируется программным обеспечением Tensor, встроенным в Платформу Tantor.

"Swagger API" всего лишь открывает в новом окне браузера ссылку https://имя_хоста-Платформы/docs/ в которой высвечиваются страницы Swagger UI, описывающий (документирующий) REST API Платформы. Для доступа к этой странице аутентификация не нужна.

Оповещения

Tenant 1 / Оповещения

ИД ОПОВЕЩЕНИЯ	ВРЕМЯ СОЗДАНИЯ	РАБОЧЕЕ ПРОСТРАНСТВО	РЕСУРС	КРИТИЧНОСТЬ	ТРИГГЕР	СТАТУС
646	08/09/2025 12:15:26	Edu	10.0.2.15	Проблема	Task execution failed.	Открыто
621	06/09/2025 08:38:00	Edu	10.0.2.15	Проблема	Available Memory < 20%, Last value: 19%	Открыто
622	06/09/2025 08:38:00	Edu	1	Оповещения	Task execution failed. (scheduler_task_id: 2)	Открыто

Элементов на странице: 25 | 1–25 из 63 | < < > >|

1 Оповещения

Task execution failed. (scheduler_task_id: 2)

08/09/2025 12:15
Ресурс: Имя хоста: tantor
Источник: Edu

Available Memory < 20%, Last value: 19% (08/09/2025 03:45)

06/09/2025 08:38
Все оповещения

tantor

36

Оповещения

При нажатии на значок колокольчика появляется всплывающее окно с оповещениями (алёртами, уведомлениями). Если в этом окне кликнуть на "Все оповещения", то откроется страница со списком всех оповещений, в том числе "закрытых" (неактивных). Оповещение можно закрыть вручную, оно может закрыться автоматически при снижении значения метрики ниже уровня "восстановления".

Уведомления настраиваются на странице "Настройки тенанта".

https://docs.tantorlabs.ru/tp/6.1/instances/ug_alerts.html

Практика 1. Вход в Платформу

1. Запустите браузер Firefox. Войдите в Платформу:
логин: student@student.ru
пароль: Student123!
2. Выполните выход из Платформы и снова войдите под тем же пользователем.
3. В меню профиля пользователя Платформы найдите "Лицензионное соглашение"
4. В меню профиля пользователя Платформы смените язык пользователяского интерфейса на английский, потом на русский
5. Поменяйте тему на тёмную, затем светлую, затем верните системную
6. Перейдите на страницу со списком тенантов и откройте тенант Tenant 1

Практика 1

Учётные записи пользователей операционной системы:

astra/astra
postgres/postgres
root/root

1. Запустите браузер Firefox. Войдите в Платформу:
логин: student@student.ru
пароль: Student123!
2. Выполните выход из Платформы и снова войдите под тем же пользователем.
3. В меню профиля пользователя Платформы найдите "Лицензионное соглашение"
4. В меню профиля пользователя Платформы смените язык пользователяского интерфейса на английский, потом на русский
5. Поменяйте тему на тёмную, затем светлую, затем верните системную
6. Перейдите на страницу со списком тенантов и откройте тенант Tenant 1



2

Подготовка к использованию

Архитектура Платформы Tantor

Виды установки Платформы Tantor

- Виды установки:
 - › типичная в контейнерах docker на одном хосте
 - › отказоустойчивая вне контейнеров docker
- Операционная система:
 - › Astra Linux версий 1.7.5+, 1.8
 - › RedOS 7.3+
 - › Centos 7+
- Платформа версии 6.0 поддерживает версии PostgreSQL от 10 до 17.x версии

Экземпляров PostgreSQL	ядер CPU	RAM (GB)	занимаемое место (SSD)
10	4	8	100 GB
100	12	24	500 GB
1000*	18	32	1-2 ТВ
5000*	32	64	5-10 ТВ

Виды установки Платформы Tantor

Типичная установка программного обеспечения выполняется на один хост или виртуальную машину с операционной системой:

Astra Linux версий 1.7.5+, 1.8

RedOS 7.3+

Centos 7+

Платформа устанавливается в виде набора docker-контейнеров. Минимальные версии, которые должны быть установлены в операционной системе:

docker -v

Docker version 20.10.13

docker-compose -v

Docker Compose version v1.29.2

Типичная установка позволяет обслуживать до 1000 экземпляров PostgreSQL. Для обслуживания большего числа экземпляров Платформу нужно устанавливать в режиме отказоустойчивости. Установку в режиме отказоустойчивости и миграцию на новые версии помогает выполнить техническая поддержка Тантор Лабс. Примерная схема установки описана в документации по 4 версии: https://docs.tantorlabs.ru/tp/4.2/vm_installation.html

При типичной установке миграция на новые версии Платформы Tantor выполняется стандартным инсталлятором с минимальным пространством доступа к Платформе.

Платформа версии 6.0 поддерживает версии PostgreSQL от 10 до 17.x версии. Более новые версии PostgreSQL поддерживаются более новыми версиями Платформы.

Если после перезагрузки хоста Платформа не доступна и страница входа выдаёт "**Network error**", то достаточно перезапустить контейнер:

docker restart reverse-proxy

и выполнить команду:

sudo crontab -e

которая откроет в редакторе файл /var/spool/cron/crontab/root

в него в самый конец нужно добавить строку:

@reboot /usr/bin/docker restart reverse-proxy

Директива **@reboot** в конце файла запускает команду перезапуска контейнера `reverse-proxy` после того, как `linux` запустится и сетевые интерфейсы будут инициализированы.

Хост Платформы должен иметь полное (Fully Qualified Domain Name) имя, которое должно разрешаться в IP-адрес на хостах, где будут установлены агенты Платформы.

Агенты Платформы Tantor

- на хосте агента должны быть открыты порты TCP 443 и 4222 в сторону хоста с установленной Платформой
- имя хоста Платформы должно разрешаться в IP на хосте агента
- запускается в фоновом режиме как служба `systemd`
- выполняется с правами пользователя операционной системы `postgres`
- скачивается в виде пакета `deb` или `rpm` с хоста Платформы
- агент при выполнении операций учитывает нагрузку на хост и подождёт перед выполнением операции, если хост перегружен

Агенты Платформы Tantor

Для мониторинга и управления экземплярами PostgreSQL нужно установить программу-агента. Это небольшая, нересурсоёмкая программа, которая запускается в фоновом режиме как служба `systemd`. Агент, обычно, скачивается в виде пакета `deb` или `rpm` с хоста Платформы. Доступ в интернет не обязателен. Агент не использует `docker`. Инсталлятор использует утилиту `systemctl` для добавления службы агента в автозапуск. Отсутствие утилиты не препятствует установке агента, он может запускаться любыми способами.

Агент **выполняется с правами пользователя операционной системы `postgres`**, что упрощает взаимодействие с экземпляром PostgreSQL и улучшает безопасность, так как не используются права `root`.

Для корректной работы Платформы на хосте агента должны быть **открыты порты TCP 443, 4222 в сторону хоста с установленной Платформой** для подсоединения агента.

Агент подсоединяется к Платформе используя имя хоста Платформы. Имя должно разрешаться а IP-адрес.

После того, как агент установлен, Платформа начинает получать метрики, получает возможность применять конфигурации, выполнять регламентные работы и прочее.

Агент - это легковесное приложение, написанное на языке `Golang`. Агент отправляет только изменившиеся данные на основе срезов и предвыборки данных, что снижает объем сетевого трафика и нагрузку на Платформу.

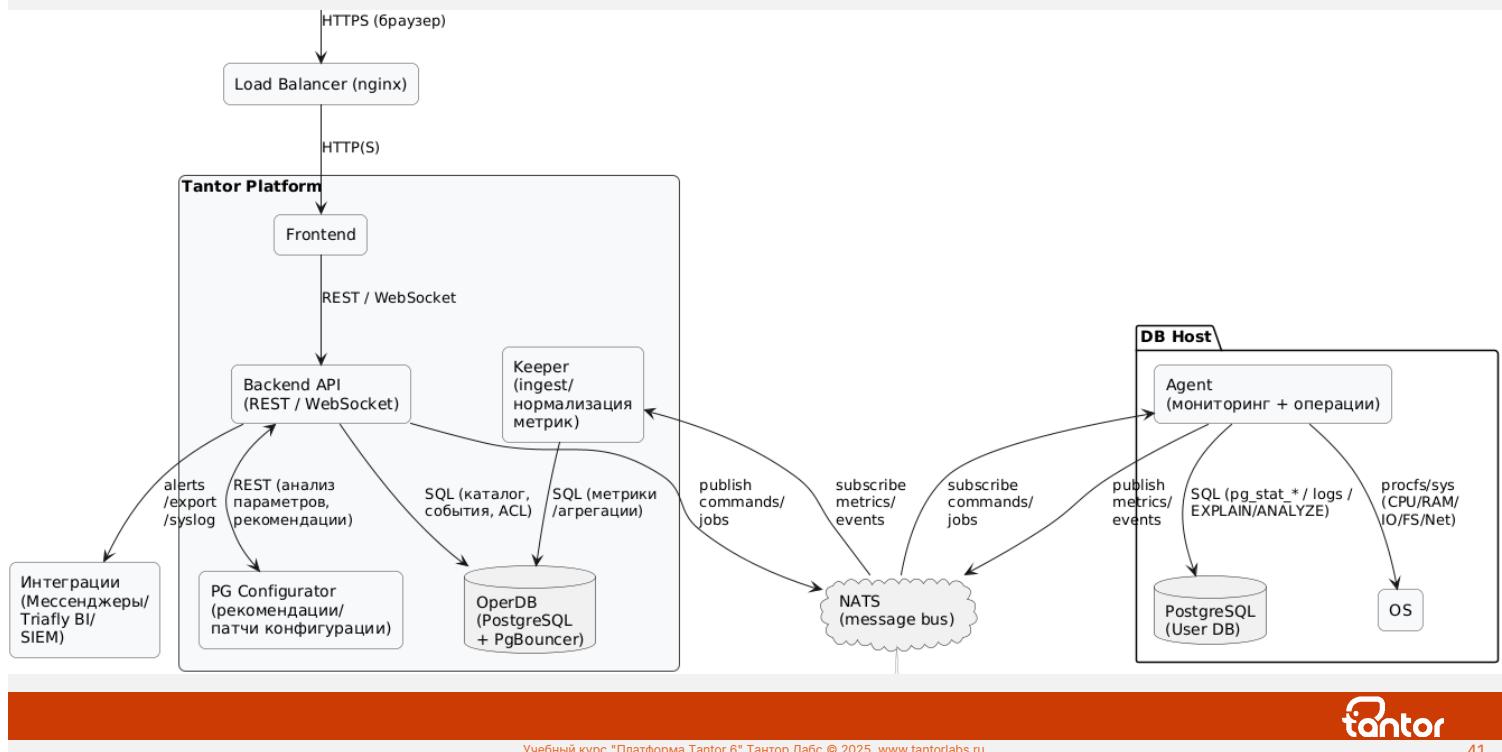
Агенты обновляются на новые версии независимо от обновления Платформы, что минимизирует простоту в работе Платформы.

Агент при выполнении операций учитывает нагрузку на хост и подождёт перед выполнением операции, если хост перегружен.

Клиентская часть представляет собой веб-интерфейс, который позволяет управлять базами данных в браузере. В адресной строке браузера достаточно ввести FQDN (полное доменное имя) хоста с установленной Платформой. Браузер подсоединится на порт 443, который является стандартным для протокола `https`. Появится страница аутентификации.

Платформа написана на языке `Golang`, конфигуратор на `Python`, фронтенд на `JavaScript` (фреймворк `Angular`). Внутри Платформы используется СУБД Tantor Postgres с расширением `PipelineDB` для обработки "на лету" потока метрик из экземпляров PostgreSQL, которыми управляет.

Архитектура Платформы Tantor



Учебный курс "Платформа Tantor 6" Тантор Лабс © 2025 www.tantolabs.ru

Архитектура Платформы Tantor

Агенты собирают метрики PostgreSQL из `pg_stat_*`, `pg_stat_statements`, `pg_store_plans`, диагностического лога кластера, метрики `linux` (CPU, RAM, HDD, сеть), выполняют операции обслуживания (VACUUM, ANALYZE, командные файлы) заданные через Платформу администратором и по расписанию, применяют конфигурации и публикуют изменения метрик в шину сообщений NATS.

Шина сообщений (NATS). Асинхронный обмен данными: topics/streams: metrics (телеметрия), commands (управляющие задания), events (события), связь источников/потребителей, буферизация и горизонтальное масштабирование. Взаимодействие выполняется по протоколу HTTPS.

Keeper (быстрые вставки(ingest)/очереди метрик). Принимает и нормализует потоки метрик из NATS, агрегирует и записывает их в хранилище Платформы.

Backend (REST/WebSocket API). Бизнес-логика: каталог экземпляров PostgreSQL, политики доступа, алёрты, задания, интеграции. Отдаёт данные UI и управляет агентами (через NATS). API позволяет Платформе интегрироваться со внешними системами.

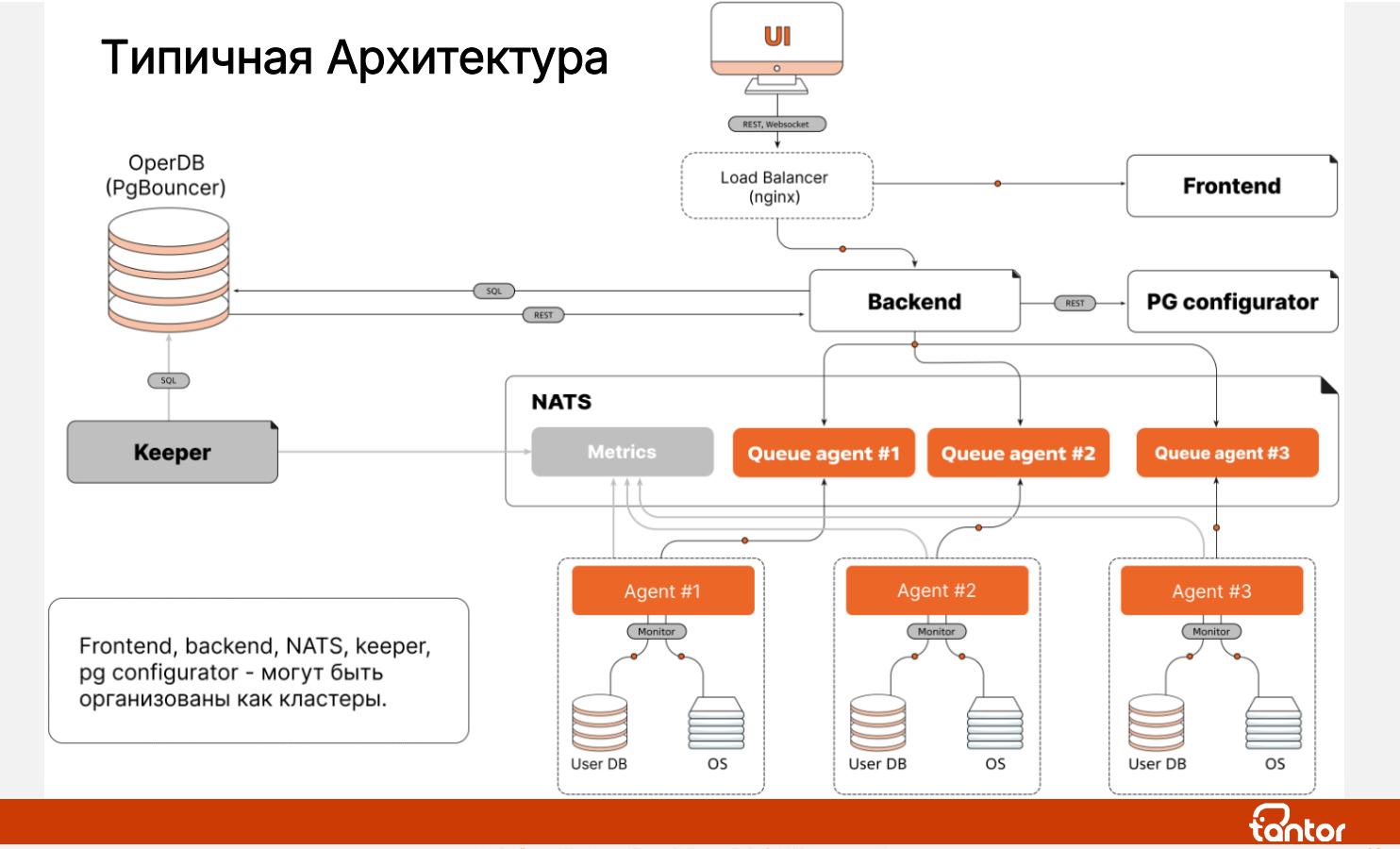
pg_configurator (advice engine) - сервис рекомендаций по параметрам PostgreSQL, профилям нагрузки, автовакуума, настройкам памяти и I/O. Возвращает рекомендации и готовые патчи конфигурации.

OperDB (PostgreSQL, центральное хранилище). Хранит метрики, события, конфигурации, модели пространств/ролей, историю рекомендаций и выполненных задач. Доступ - через PgBouncer.

Web UI (SPA) - панели мониторинга, профили запросов, журнал событий, конструктор алертов, каталог БД, работа с пространствами/пользователями и пр.

Интеграции. Мессенджеры (Telegram, Mattermost, e-mail): доставка уведомлений о событиях; Triafly BI: экспорт исторических данных мониторинга для долговременной аналитики и построения произвольной отчетности; отправка событий/инцидентов по Syslog (это называют "SIEM", Security Information and Event Management).

Типичная Архитектура



Учебный курс "Платформа Tantor 6" Тантор Лабс © 2025 www.tantolabs.ru

Типичная Архитектура

Клиент-серверная архитектура: Платформа основана на клиент-серверной архитектуре, в которой клиентская часть представляет собой веб-интерфейс, а серверная часть состоит из нескольких модулей, написанных на языках Golang, Python и JS Angular.

Сбор метрик: сбор метрик осуществляется с помощью Агента. Агент наблюдает за базой данных и операционной системой. Он собирает информацию о метриках из администрируемой базы данных и отправляет их через брокер сообщений NATS на backend-сервер. Агент также имеет возможность отправлять только изменившиеся данные на основе срезов и предагрегации данных для снижения объема сетевого трафика и нагрузки на сервер.

Хранение метрик: метрики хранятся в основанной на PostgreSQL СУБД, которая была дополнена специальными расширениями для обработки большого количества метрик «на лету». Полученные данные сохраняются в OperDB и могут быть распределены по пространствам.

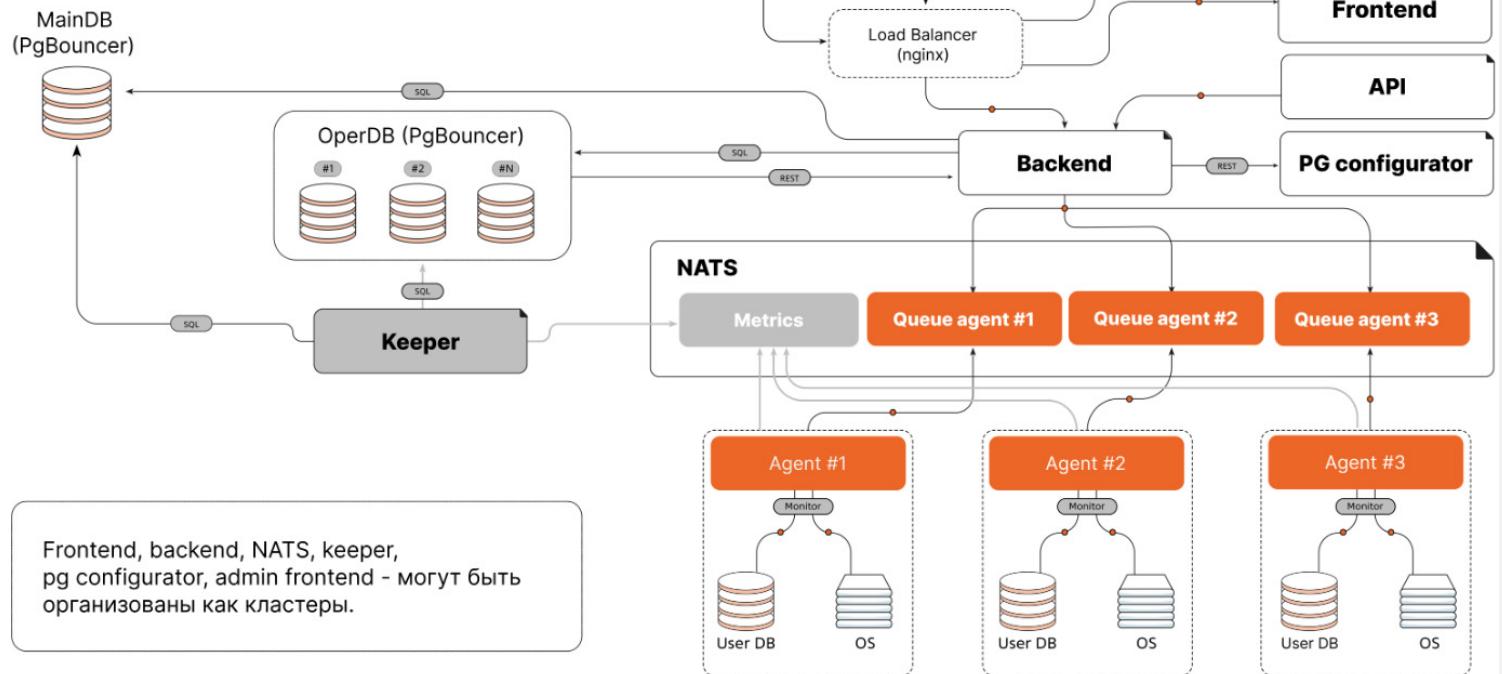
Сбор и хранение журналов СУБД: с помощью дополнительного сервиса pg-monitor возможно настроить сбор и аналитику журналов СУБД с последующей визуализацией в графическом интерфейсе. Этот сервис является опциональным, так как требует дополнительное соединение по протоколу ssh к наблюдаемым экземплярам.

Масштабирование: Платформа поддерживает горизонтальное масштабирование большинства компонентов.

Отказоустойчивость: Платформа обеспечивает отказоустойчивость путем дублирования основных компонентов.

Безопасность: среда исполнения Агента не требует наличия дополнительных открытых сетевых портов. Однако Агенту необходимо иметь сетевое подключение до портов NATS как точки обмена данными с серверной частью Платформы. Также Агент выполняется от имени пользователя postgres, что с одной стороны позволяет иметь возможность глубоко интегрироваться с СУБД, а с другой — минимизирует его влияние на среду операционной системы.

Масштабируемая архитектура



Учебный курс "Платформа Tantor 6" Тантор Лабс © 2025 www.tantolabs.ru

Масштабируемая архитектура

Масштабируемый вариант установки используется, если Платформа работает более с чем 1000 экземплярами PostgreSQL. Экземпляры разделены между несколькими OperDB.

Масштабируемость Платформы основана на Python-скрипте Patroni. Patroni позволяет преобразовать OperDB в высокодоступный кластер с автоматическим контролируемым и аварийным переключением. OperDB использует расширения для обработки большого числа метрик:

```
sudo docker exec -it operdb psql -U postgres -p 5432 pma -c "\dx"
List of installed extensions
  Name           | Description
-----+-----
btree_gin        | support for indexing common datatypes in GIN
http            | HTTP client for PostgreSQL
pg_columnar     | Hydra Columnar extension
pg_cron          | Job scheduler for PostgreSQL
pg_framework    | Tantor PG Framework
pg_stat_statements | track execution statistics of all SQL statements
pg_store_plans  | track plan statistics of all SQL statements executed
pg_throttle      | throttle resources usage
pgcrypto          | cryptographic functions
pipelinedb       | PipelineDB
plpgsql          | PL/pgSQL procedural language
plpython3u        | PL/Python3U untrusted procedural language
uuid-ossp        | generate universally unique identifiers (UUIDs)
```

В Платформе Tantor любой архитектуры есть модули:

pg_anon - утилита и расширение для анонимизации (маскировки) данных в управляемых СУБД.
pg-explain - модуль расширенной аналитики.

Swagger - набор инструментов для создания, генерации кода REST-клиентов и документирования REST-API по спецификации OpenAPI.

Агент

- Агент периодически (с частотой раз в несколько секунд) соединяется с базой данных мониторируемого экземпляра PostgreSQL
- для установки переменных окружения агента используется файл /etc/sysconfig/pmaagent:

```
cat /etc/sysconfig/pmaagent
# This file can be used to set additional environment variables for
# the pmaagent process, or pass additional options to the pmaagent
# executable.
```

- для перезапуска агента и добавления службы используются команды:

```
systemctl daemon-reload
systemctl restart pmaagent
systemctl enable pmaagent
```

Агент

Агент периодически (с частотой раз в несколько секунд) соединяется с базой данных мониторируемого экземпляра PostgreSQL. Агент использует строку подключения из файла:

```
cat /var/lib/pma/agent/agent.yml | grep connectionString
connectionString:
```

```
postgresql://%s:%s@%s:%d/%s?application_name=%s&default_query_exec_mode=cache_describe
```

Для связи по API с кластером Patroni агент может аутентифицироваться по сертификату SSL (TLS).

Полученные агентом метрики передаются Платформе Tantor.

Файл службы агента:

```
cat /etc/systemd/system/multi-user.target.wants/pmaagent.service
[Unit]
Description=PMA Agent
After=network.target
[Service]
Restart=on-failure
Type=notify
EnvironmentFile=/etc/sysconfig/pmaagent
User=postgres
ExecStart=/usr/sbin/pmaagent
ExecReload=/bin/kill -HUP $MAINPID
[Install]
```

Агент работает под пользователем операционной системы **postgres**.

Агент написан на языке Go и использует драйвер **pgx**, а не **libpq**. Поэтому, не все параметры, которые работают в **libpq**, работают в **pgx**. Драйвер **pgx** (**ParseEnvLibpq**) распознаёт только переменные окружения: PGHOST PGPORT PGDATABASE PGUSER PGPASSWORD PGSSLMODE PGSSLCERT PGSSLKEY PGSSLROOTCERT PGAPPNAME PGCONNECT_TIMEOUT PGTARGETSESSIONATTRS.

Не работает:

PGOPTIONS="-c log_disconnections=off", также, параметр options нельзя передать в строке соединения.

Файлы Агента

- Исполняемый файл агента:
 - /usr/sbin/pmaagent.
- Файл описателя службы агента:
 - /usr/lib/systemd/system/pmaagent.service
- Директория агента:
 - /var/lib/pma/agent
- В директории лежит конфигурационный файл агента:
 - /var/lib/pma/agent/agent.yml
- В конфигурационном файле путь к файлу лога:
 - path: /var/lib/pma/agent/logs
 - file: pmaagent.log
- В конфигурационном файле адрес Платформы:
 - endpoint: tcp://хост:4222

Файлы Агента

Исполняемый файл агента: /usr/sbin/pmaagent.

Файл описателя службы агента: /usr/lib/systemd/system/pmaagent.service

Директория агента: /var/lib/pma/agent

В директории лежит конфигурационный файл агента: /var/lib/pma/agent/agent.yml

В конфигурационном файле путь к файлу лога:

path: /var/lib/pma/agent/logs

file: pmaagent.log

указана конечная точка Платформы, куда агент передаёт метрики:

endpoint: tcp://education.tantolabs.ru:4222

Список параметров командной строки агента: /usr/sbin/pmaagent --help

Пример команд:

completion Generate the autocompletion script for the specified shell

configure Configure the pmaagent

deregister Deregistering the pmaagent

help Help about any command

instances Manage instances

register Registering the pmaagent

start Start the PMA Agent (default command)

version Print version information and quit

Список контейнеров

- Платформа состоит из 15 контейнеров
 - **три контейнера** содержат СУБД PostgreSQL

Список контейнеров

Список контейнеров из которых состоит Платформа: `docker ps`

```
  NAMES          PORTS
pg-explain
pg-monitor_dispatcher
pg-monitor_collector
reverse-proxy    80->80, 443->443, 4222->4222, 5666->5666, 8443->8443
keeper
pg_configurator 7777
frontend          80
pg_anon           8000
swagger_ui        80, 8080
backend
nats
pg_bouncer        6432
operdb            5432
tns_db            5432
maindb            5432
```

Войти в шелл контейнера можно командой: `docker exec -it pg_configurator sh`

Посмотреть размеры образов: `docker images`

Посмотреть конфигурацию контейнера: `docker container inspect reverse-proxy`

Посмотреть лог контейнера: `docker logs operdb --tail 10`

Посмотреть версию СУБД: `docker exec -it tns_db psql -U monitor -p 5432 -d pg-monitor -c "select version();"`

Два контейнера содержат СУБД Tantor PostgreSQL, один (Тензор) PostgreSQL

Посмотреть список портов: `docker port reverse-proxy`

Чтобы команду docker можно было запускать не только под root: `usermod -aG docker astra usermod -aG docker postgres`

Перезапуск Платформы: `systemctl restart docker`

Перезапуск платформы: `systemctl restart dockers`
Перезапуск сладьного контейнера: `docker restart pg_moniter_collector`

Перезапуск отдельного контейнера: `docker restart pg_monitors_collector`
Пересоздание контейнеров без потери конфигурации и данных, собранных в Платформе:

Пересоздание контейнера

```
cd /opt/tantor/eco  
docker-compose --env-file /opt/tantor/eco/pla
```

```
docker-compose --env-file /opt/tantor/.env down
```

```
/opt/tantor/eco/docker-compose.yml down  
docker: > rm -f /opt/tantor/eco/docker-compose.yml
```

```
docker-compose --env-file /opt/tantor/etc/tantor.env
```



2-2

Интеграция - Swagger UI

Swagger UI

- В Платформе есть интерфейс (API) для взаимодействия с внешними системами по протоколу REST
- Платформа имеет методы для работы с агентами, экземплярами, группами, метриками, тенантами, пользователями, пространствами
- REST-интерфейс описан в формате OpenAPI
- Swagger UI - набор веб-страниц, который позволяет отправлять запросы к API прямо из браузера и тестировать авторизацию через токены Платформы

Swagger UI

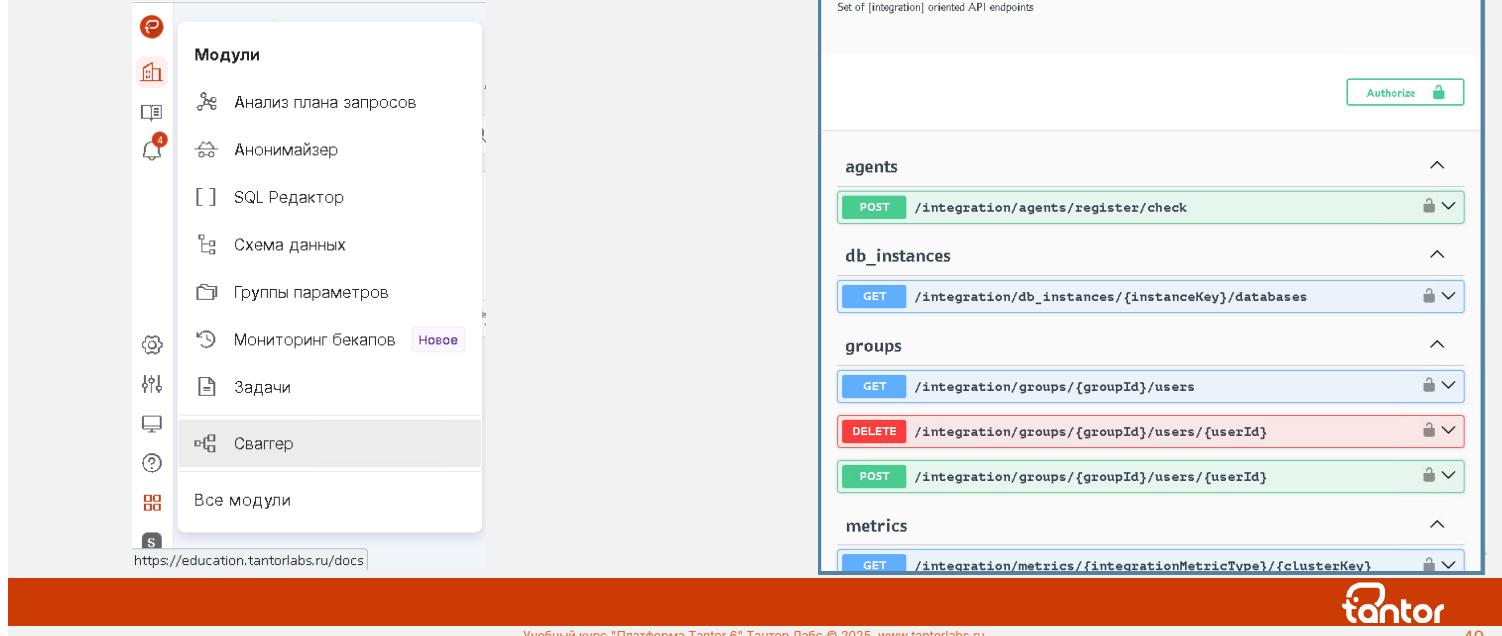
В Платформе есть интерфейс (API) для взаимодействия с внешними системами по протоколу REST. Платформа имеет методы для работы с агентами, экземплярами, группами, метриками, тенантами, пользователями, пространствами. REST-интерфейс описан в формате OpenAPI.

API (Application Programming Interface) определяет правила, каким образом одна программа может запрашивать данные или использовать функционал другой программы. REST (Representational State Transfer) - протокол взаимодействия через запросы по протоколу HTTP методами (GET, POST, PUT, DELETE, INFO и другим) с адресами (конечными точками) представленными в виде URL. OpenAPI описывает структуру JSON или YAML документа, содержащего описание конечных точек REST API, обеспечивает единообразие и стандартизацию документирования REST API.

Swagger использован в Платформе для документирования конечных точек REST Платформы на основе стандарта OpenAPI. Документация Swagger позволяет понять принцип работы API другим разработчикам, которые будут создавать клиентов для взаимодействия с Платформой по протоколу REST.

Страница Swagger UI

- Swagger UI доступен по ссылке <https://хост/docs/>
- Ссылка есть в меню Модули:



The screenshot shows the Swagger UI interface for a PMA Server. On the left, a sidebar titled 'Модули' (Modules) lists several tools: Анализ плана запросов (Query Plan Analysis), Анонимайзер (Anonymizer), SQL Редактор (SQL Editor), Схема данных (Data Schema), Группы параметров (Parameter Groups), Мониторинг бекапов (Backup Monitoring), Задачи (Tasks), Сваггер (Swagger), and Все модули (All modules). The 'Swagger' item is highlighted. The main content area is titled 'PMA Server 6.0.0 OAS 3.0' and shows a 'Set of [integration] oriented API endpoints'. It includes sections for 'agents', 'db_instances', 'groups', and 'metrics', each with a list of API endpoints with their methods, URLs, and authorization status (e.g., POST /integration/agents/register/check, GET /integration/db_instances/{instanceKey}/databases, DELETE /integration/groups/{groupId}/users/{userId}, POST /integration/groups/{groupId}/users/{userId}).

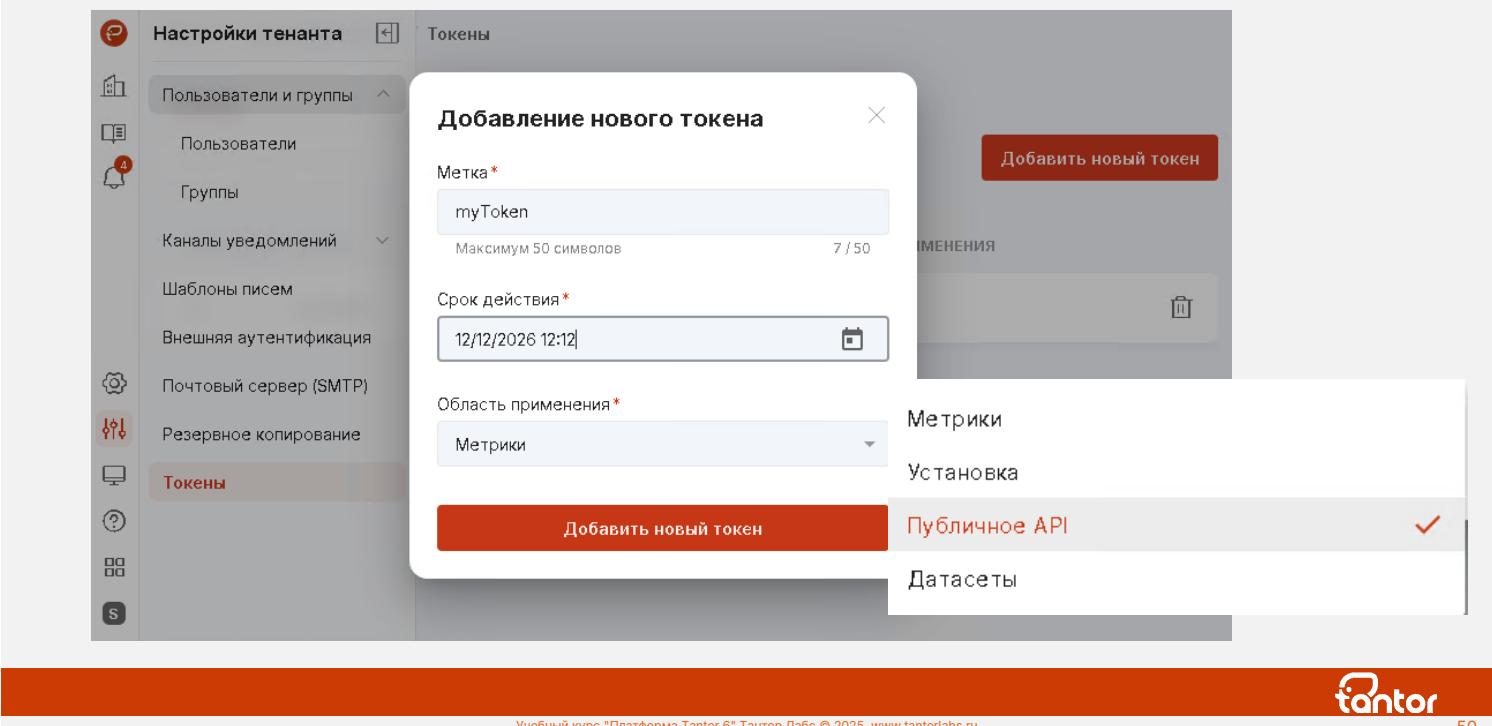
Страница Swagger UI

Swagger UI - набор веб-страниц для визуализации конечных точек REST, написанных в спецификации OpenAPI. Swagger UI позволяет отправлять запросы к API прямо из браузера и тестировать авторизацию через токены Платформы. Swagger UI доступен по ссылке <https://хоста/docs/>. Для доступа к этой странице аутентификация не нужна. Ссылку можно открыть перейдя в любой из тенантов и кликнув на значок "Модули" слева внизу страницы. Значок второй снизу в виде четырёх квадратов.

https://docs.tantorlabs.ru/tp/6.1/admin/swagger_ui.html

Токены

- используются для доступа к функционалу Платформы Tantor



Добавление нового токена

Метка*
myToken

Максимум 50 символов
7 / 50

Срок действия*
12/12/2026 12:12

Область применения*

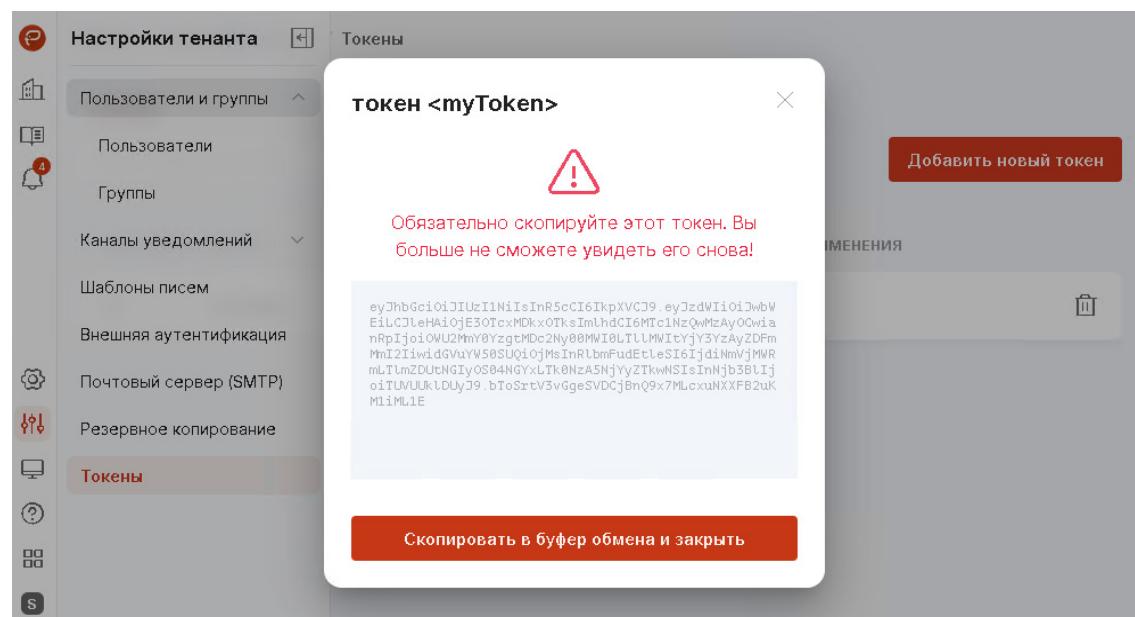
Метрики
Установка
Публичное API
Датасеты

Добавить новый токен

Токены

Токены используются для доступа к функционалу Платформы Tantor по REST API.

Ссылку "Токены" можно открыть перейдя в любой из тенантов и кликнув на значок "Настройки тенанта" в левой панели. Откроется список токенов. Для добавления нового токена нужно нажать красную кнопку "Добавить новый токен". Появится окно с токеном, который нужно скопировать в буфер обмена. **После закрытия окна доступа к токену не будет, токен можно будет только удалить.** Метка токена не уникальна и может повторяться.



токен <myToken>

Обязательно скопируйте этот токен. Вы больше не сможете увидеть его снова!

eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9. eyJzdWIiOiCiwbWEiLCJleHAiOiE3OTcxMDkxOTk-eImhdC16MTc1NzQwMzAyO0wi.a...
nRpijoiOWU2Mny8YzgjMdc2Ny00MWI0LTUUMW1cyjY3Y2ayZDFmMni2iwiwdgvuuyW58sUqi0jMsInR1bmFuadteisI6i1jaiNmVjMWRmlt1mzDUEtNGiyOS84NGYxLTK0hASNyvZTkwnSIsInNjb3BLijoiUVUkUkUDUyJ9.bTo3rtV3vGgeSVDCjBnQ9x7MLcxuNXFB2uKM1iM1E

Скопировать в буфер обмена и закрыть

Конечные точки: [agents](#), [db_instances](#), [groups](#), [metrics](#), [tenant](#), [users](#), [workspaces](#), [datasets](#).

Чтобы протестировать доступ к конечным точкам через Swagger API нужно нажать кнопку Authorize и ввести токен, соответствующего конечной точке типа.

Токен типа "установка" нужен для использования методов [agents](#). Для установки Агентов.

Токен типа "метрики" нужен для использования методов [metrics](#).

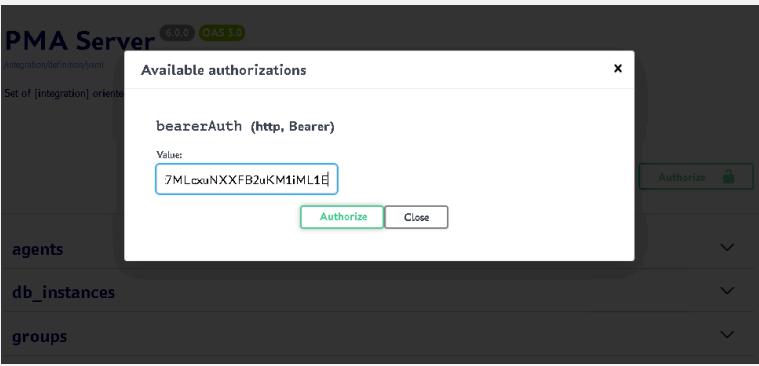
Токен типа "датасеты" нужен для использования методов [datasets](#). Для Triafly.

Токен типа "публичное API" нужен для использования всех остальных API-методов.

https://docs.tantorlabs.ru/tp/6.1/admin/admin_tokens.html

Ввод токена в Swagger UI

- токен вводится на странице Swagger API
- после ввода токена можно посылать запросы по протоколу REST



Execute Clear

Responses

Curl

```
curl -X 'GET' \
  'https://education.tantortools.ru/integration/users?page=1&limit=9' \
  -H 'accept: application/json' \
  -H 'Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpxVCJ9eyJzdWIiOiJwbWEiLC31eHaijE...
```

Request URL

```
https://education.tantortools.ru/integration/users?page=1&limit=9
```

Server response

Code Details

200 Response body

```
{ "pager": { "currentPageId": 1, "currentPageRecordsNumber": 2, "totalRecords": 4, "recordsPerPage": 9, "totalPages": 1 }, "results": [ { "userType": "internal", "userID": 3, "userName": "user", "login": "user@dba1.ru", "userEmail": "user@dba1.ru", "isActive": true, "isDefPass": false, "authTypeID": 0, "userProps": { "timeZone": "Europe/Moscow", "languageCode": "en" }, "userVector": null, "termsAndPrivacy": "2025-09-02T18:54:52.122", "passwordHash": "0001-01-01T00:00:00Z", "groupCat": 1 } ] }
```

Response headers

```
content-length: 765
content-type: application/json
date: Tue, 29 Sep 2025 07:57:49 GMT
server: nginx
vary: Origin
x-firefox-spdy: h2
```

Токены

После получения токена, его можно ввести в Swagger UI, нажав кнопку "Authorize". После чего выбрать REST метод, для доступа к которому подходит токен, заполнить значения параметров метода и нажать кнопку Execute. Swagger UI пошлет запрос в Платформу Tantor, получит и отобразит результат.

Вместе с результатом отображается и сам запрос в виде параметров утилиты командной строки curl:

```
curl -X 'GET' 'https://хост/integration/users?page=1&limit=9'
-H 'accept: application/json'
-H 'Authorization: Bearer eyJhbG...Xysg'
```

Результат выдается в том формате, в котором его выдаёт служба REST. Пример вывода списка пользователей тенанта, запрошенных токеном тип "публичное API" в формате JSON:

```
{
...
"results": [
{
  "userType": "internal",
  "userID": 3,
  "userName": "user",
  "login": "user@dba1.ru",
  "userEmail": "user@dba1.ru",
  "isActive": true,
  "isDefPass": false,
  "authTypeID": 0,
  ...
}
```

Для диагностики Swagger UI показывает заголовки протокола HTTP:

```
content-length: 765
content-type: application/json
server: nginx
vary: Origin
x-firefox-spdy: h2
```

Практика 2. Swagger UI

1. Создайте в тенанте токен типа "Публичное API"
2. Откройте страницу Swagger UI
3. Вставьте токен на странице
4. Выполните вызов по протоколу REST для получения списка пользователей тенанта
5. Выполните вызов по протоколу REST из командной строки утилитой `curl`

Практика 2

1. Создайте в тенанте токен типа "Публичное API"
2. Откройте страницу Swagger UI
3. Вставьте токен на странице
4. Выполните вызов по протоколу REST для получения списка пользователей тенанта.
5. Выполните вызов по протоколу REST из командной строки утилитой `curl`



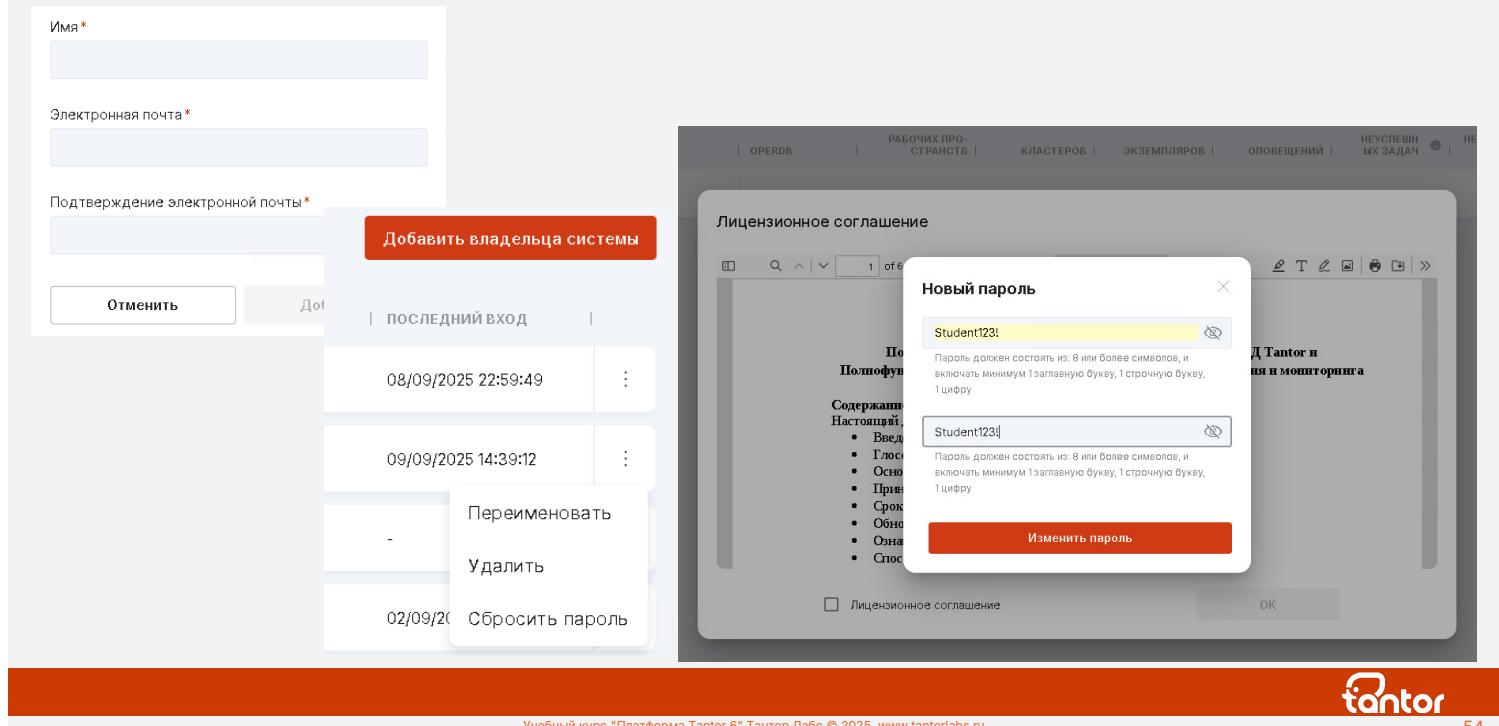
3

Платформа

Начало работы с Платформой

Первый вход в Платформу

- для создания пользователя нужно указать его имя и email



Первый вход в Платформу

Для создания пользователя нужно указать его имя и email. Email будет использоваться для аутентификации пользователя. Имя не уникально, может быть изменено в любой момент.

После создания пользователя, ему можно установить пароль выбрав "Сбросить пароль" в контекстном меню. Пароль может быть автоматически сгенерирован или установлен вручную. Если настроен адрес почтового сервера, то на email автоматически будет выслано письмо с паролем для входа в Платформу. На этот адрес также будут направляться уведомления Платформы. Если посылка письма будет неудачна, то будет выдано предупреждение, которое можно проигнорировать. Пароль при этом будет установлен.

При первом входе нового пользователя будет выдано окно смены пароля, придётся установить новый пароль. Выполняются проверки на "слишком распространённые" пароли, также на сложность пароля. Например, пароль "Student12!" и "Owner123!" слишком распространены, а "Owner12!" и "Student123!" не слишком.

После успешной смены пароля будет предложено прочесть и принять лицензионное соглашение.

Лицензия на использование Платформы

- меню "Настройки системы" -> "Лицензия"

Лицензия на использование Платформы

Лицензию на использование Платформы можно посмотреть в меню "Настройки системы" -> "Лицензия". Сразу после установки Платформы используется 30-дневная пробная лицензия. Меню "Настройки системы" открывается при клике на значок шестерёнки слева внизу страницы.

Начальная страница Платформы - список тенантов

- меню владельца системы:

ИМЯ ТЕНАНТА	OPERDB	РАБОЧИХ ПРОСТРАНСТВ	КЛАСТЕРОВ	ЭКЗЕМПЛИЯРОВ	ОПОВЕЩЕНИЙ	НЕУСПЕШНЫХ ЗАДАЧ	НЕУСПЕШНЫХ БЭКАПОВ
Tenant 1	rma	1	0	2	4	1	0

- меню владельца или пользователя тенанта:

ИМЯ ТЕНАНТА	РАБОЧИХ ПРОСТРАНСТВ	КЛАСТЕРОВ	ЭКЗЕМПЛИЯРОВ	ОПОВЕЩЕНИЙ	НЕУСПЕШНЫХ ЗАДАЧ	НЕУСПЕШНЫХ БЭКАПОВ
Tenant 1	1	0	2	4	1	0

Учебный курс "Платформа Tantor 6" Тантор Лабс © 2025 www.tantolabs.ru

Начальная страница Платформы - список тенантов

Для "владельцев системы" (Платформы); "владельцев тенанта", пользователей тенанта начальной страницей является список тенантов. Внешний вид для владельцев Платформы и пользователей немного отличается. У владельцев есть меню "Настройки системы" (значок шестерёнки слева), выдаётся название OperDb (БД Платформы Тантор), которая указывалась при создании тенанта, в контекстном меню есть пункт "Удалить тенант", "Переименовать", "Расширенная информация".

Значок "шестерёнка" справа вверху над таблицей со списком тенантов - это универсальный элемент таблицы, позволяющий выбирать столбцы для отображения в таблице.

Значок "три точки" справа на каждой строке таблицы - это универсальный элемент таблицы, по которому вызывается контекстное меню.

Для возврата на страницу со списком тенантов с любой страницы Платформы, достаточно кликнуть на значок листочка в круге (логотип Тантор) слева вверху страницы.

Тенант (владение) - логический объект, который содержит пространства (workspaces). В пространствах содержатся объекты управления: экземпляры, обслуживающие кластера баз данных PostgreSQL и кластера Patroni. Тенанты имеют владельцев и пользователей. В списке владельцев тенанта отображаются владельцы системы.

Число неуспешных задач и бэкапов отображается за последние сутки, от чем написано во всплывающем окне, которое появится, если навести мышь на букву "i" в кружочке:

ИМЯ ТЕНАНТА	OPERDB	РАБОЧИХ ПРОСТРАНСТВ	КЛАСТЕРОВ	ЭКЗЕМПЛИЯРОВ	ОПОВЕЩЕНИЙ	НЕУСПЕШНЫХ ЗАДАЧ	НЕУСПЕШНЫХ БЭКАПОВ
Tenant 1	rma	1	0	2	4	1	0

Тенант: Расширенная информация

- на странице "Расширенная информация" показан список владельцев тенанта, на ней можно добавить владельца

Добавление Владельца тенанта
«Tenant 1»

Имя*
Имя

Электронная почта*
Введите адрес электронной почты, чтобы пригласить

Подтверждение электронной почты*
Введите адрес электронной почты, чтобы пригласить

Отменить Добавить владельца

ПОСЛЕДНИЙ ВХОД В ТЕНАНТ

ПОСЛЕДНИЙ ВХОД В ТЕНАНТ	ВРЕМЯ	ДЕЙСТВИЯ
02/09/2025 21:37:41		
09/09/2025 16:49:42		
09/09/2025 13:29:58		

tantor 57

Тенант: Расширенная информация

Тенант (владение) - логический объект, который содержит пространства (workspaces). В пространствах содержатся объекты управления: экземпляры, обслуживающие кластера баз данных PostgreSQL и кластера Patroni. Тенанты имеют локальных для тенантов владельцев и пользователей, пространства, экземпляры и собственные настройки.

При добавлении владельца тенанта на странице "Расширенная информация" пользователь, если он не существует в списке пользователей тенанта, будет добавлен в этот список и получит права владельца тенанта.

При удалении владельца системы он полностью удаляется из Платформы, даже если он есть в списке владельцев тенантов.

В списке владельцев тенанта также отображаются владельцы системы. Владельца тенанта можно убрать из списка владельцев, тогда он станет обычным пользователем тенанта. Его можно снова добавить в список владельцев тенанта, пароль при этом не поменяется.

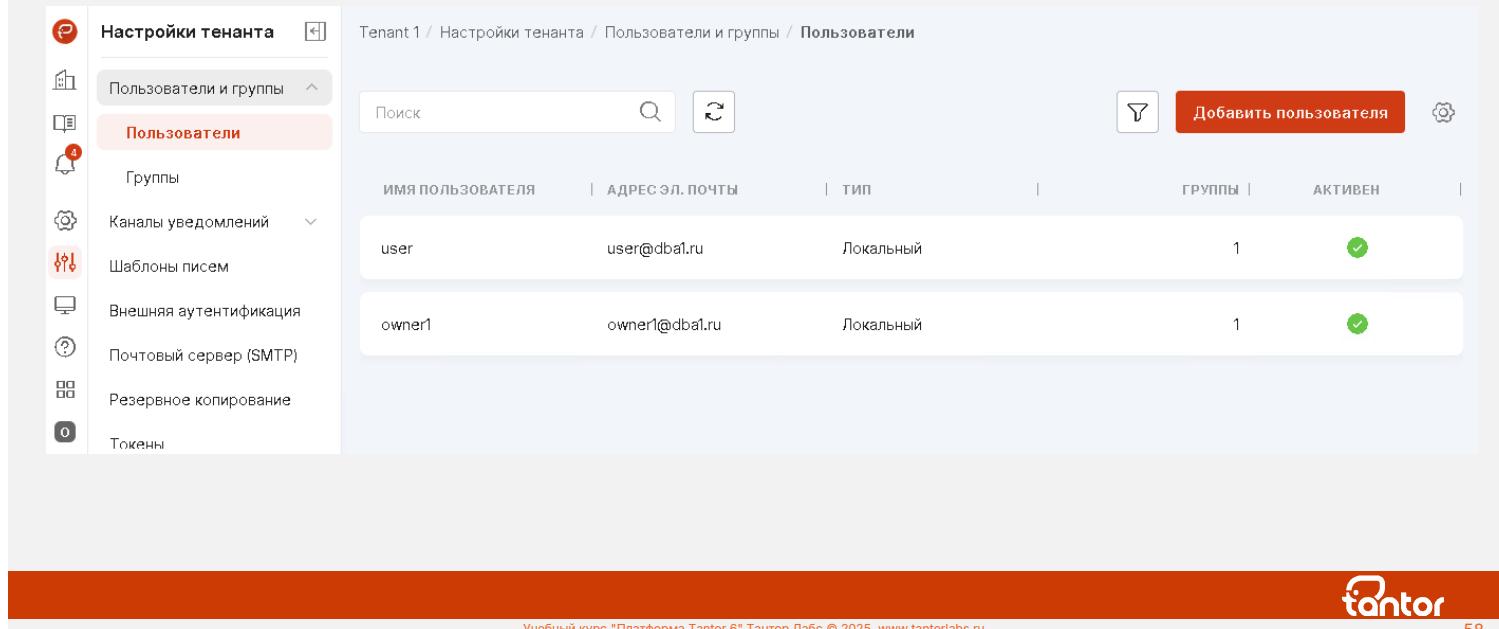
Изменения прав, добавление в группы действуют немедленно, переподсоединяться не нужно. Если пользователь обратится к странице, на которую нет прав выдастся предупреждение, будет выполнен логаут и выдана страница логина.

Добавить пользователя тенанта в список владельцев системы нельзя - будет выдана ошибка "Пользователь с такой почтой уже существует".

Добавить владельца системы в список пользователей тенанта нельзя - будет выдана ошибка "Данный адрес электронной почты уже используется, пожалуйста, попробуйте другой адрес электронной почты".

Настройки тенанта: Пользователи

- на странице "Пользователи" показан список пользователей, в том числе владельцев тенанта, на ней можно добавить локального пользователя тенанта



Имя пользователя	Адрес эл. почты	Тип	Группы	Активен
user	user@dbat.ru	Локальный	1	<input checked="" type="checkbox"/>
owner1	owner1@dbat.ru	Локальный	1	<input checked="" type="checkbox"/>

Настройки тенанта: Пользователи

На странице "Пользователи" показан список пользователей, в том числе владельцев тенанта, на ней можно добавить пользователя тенанта. Пользователь будет иметь тип "локальный". Ещё есть "внешний". Хоть пользователь и локальный, но его можно добавить в другие тенанты, пароль и имя при этом сохраняются.

Переименовать пользователя можно на странице любого тенанта, где он есть в списке пользователей.

Смена статуса активен/неактивен действует на **все** тенанты.

При удалении локального пользователя из любого тенанта он удаляется из всех тенантов и Платформы **полностью**.

Пользователя тенанта можно добавить на странице "Настройки тенанта" (значок "ползунки" на левой полосе меню) -> "Пользователи".

По умолчанию, только что созданный созданный пользователь (если он не в списке владельцев тенанта) не имеет прав на пространства тенанта.

Права на пространства и их содержимое даётся группам, в которые можно включать локальных пользователей тенанта. Напрямую пользователям эти права не даются.

Страница пользователя Платформы

- на странице пользователя владелец системы или тенанта может переименовать, удалить, изменить пароль и поменять статус активен/неактивен выбранному пользователю:

Настройки тенанта

Пользователи и группы

Пользователи

Группы

Каналы уведомлений

Шаблоны писем

Внешняя аутентификация

Почтовый сервер (SMTP)

Резервное копирование

Информация

группы

Удалить

Изменить пароль

Переименовать

Имя пользователя: owner1

Адрес эл. почты: owner1@dbaf1.ru

Тип: Локальный

Активен:

Учебный курс "Платформа Tantor 6" Тантор Лабс © 2025 www.tantorlabs.ru

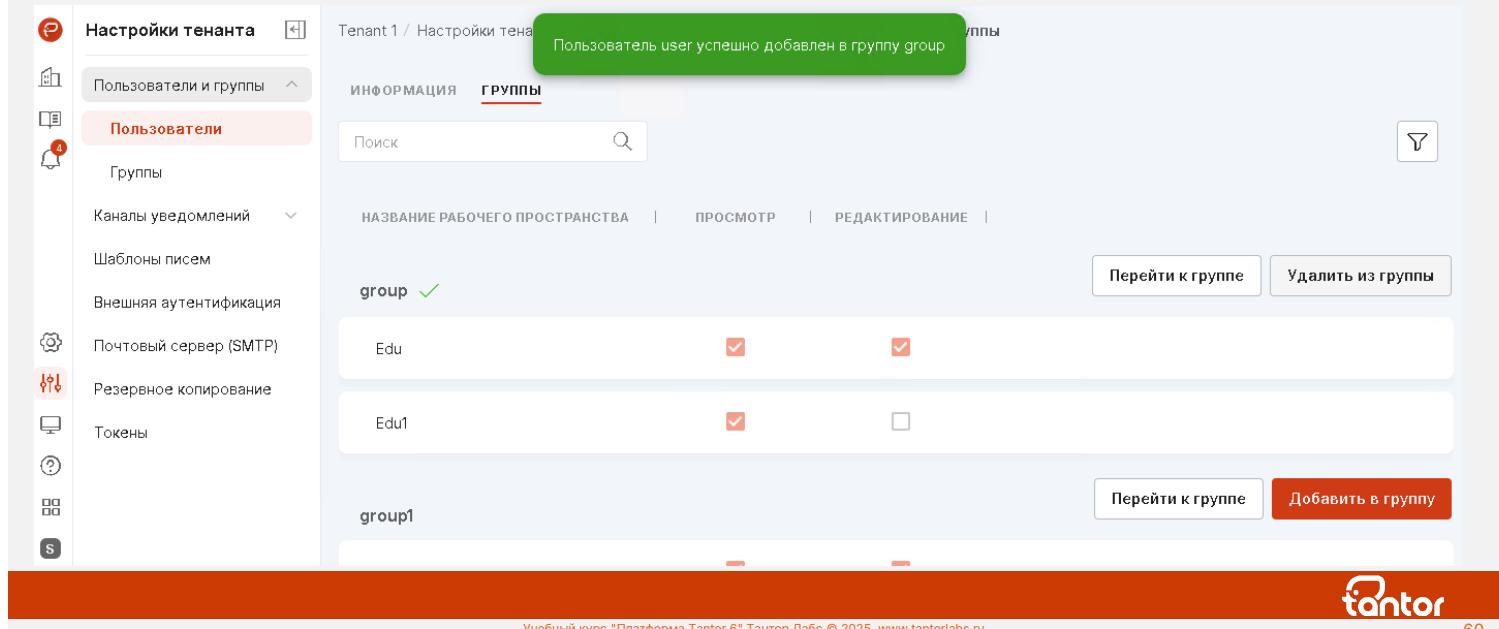
Страница пользователя Платформы

Кликнув на имя пользователя откроется страница со свойствами пользователя, на которой владелец системы или тенанта может переименовать, удалить, изменить пароль и поменять статус активен/неактивен выбранному пользователю.

Неактивный пользователь при вводе правильного и неправильного пароля получит ошибку "Account suspended. Please contact administrator". Так как отличить правильный или не правильный пароль по этому сообщению нельзя, это защищает от подбора пароля неактивных пользователей, что указывает на безопасность Платформы в мелочах.

Включение пользователя в группы

- выполняется на вкладке ГРУППЫ на странице пользователя
- владельцу тенанта нельзя поменять членство в группах и статус активен/неактивен



Настройки тенанта

Пользователи и группы

Пользователи

Группы

Назначение рабочего пространства

Просмотр

Редактирование

group ✓

Edu

Edu1

group1

Перейти к группе

Удалить из группы

Добавить в группу

tantor

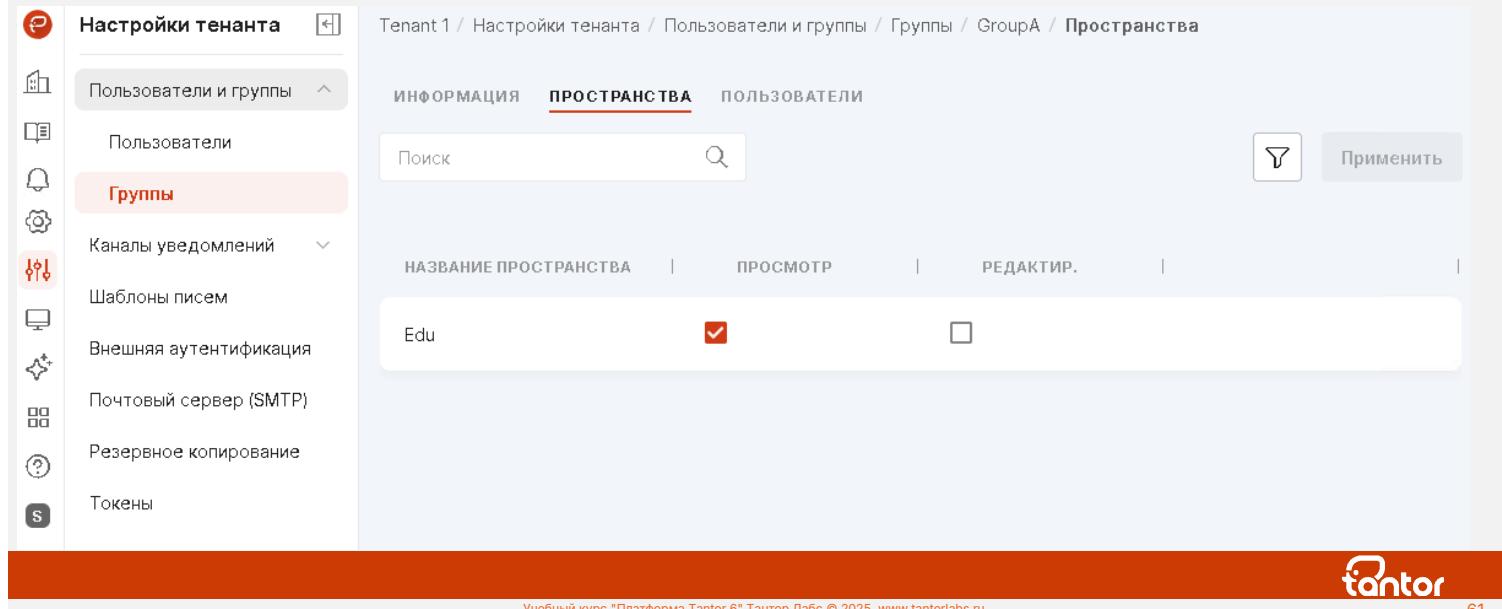
60

Включение пользователя в группы

Пока пользователь является владельцем тенанта его статус активен/неактивен нельзя поменять и нельзя изменить членство в группах. Соответствующие кнопки и ползунок для владельцев тенанта неактивны.

Настройки тенанта: Группы

- на странице "Группы" на вкладке ПРОСТРАНСТВА добавляются права на просмотр (read) и изменения (write) в экземплярах, включенных в рабочее пространство:



Настройки тенанта / Tenant 1 / Настройки тенанта / Пользователи и группы / Группы / GroupA / Пространства

ИНФОРМАЦИЯ ПРОСТРАНСТВА ПОЛЬЗОВАТЕЛИ

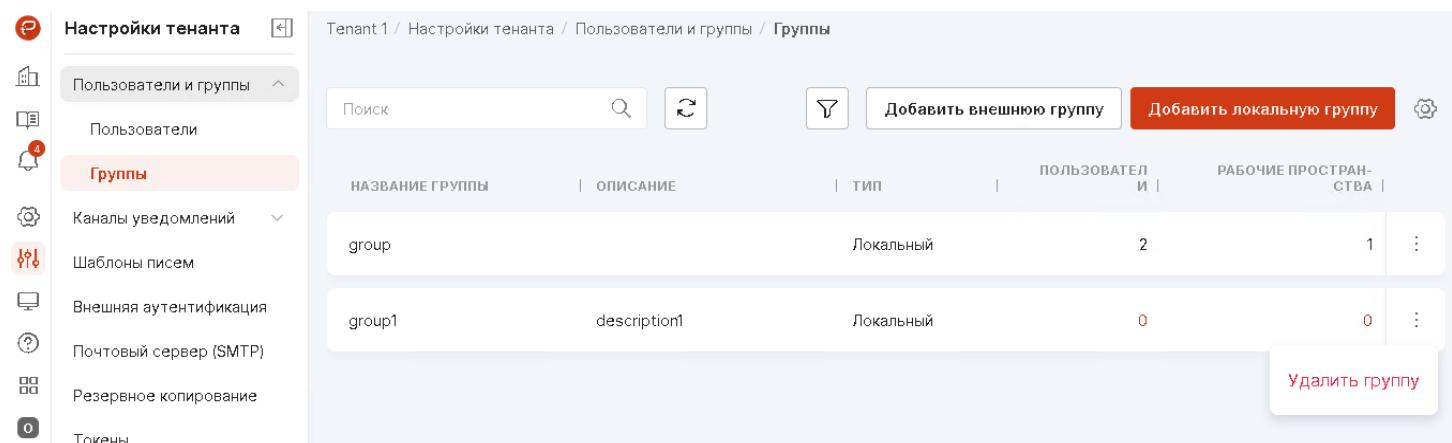
Поиск

НАЗВАНИЕ ПРОСТРАНСТВА | ПРОСМОТР | РЕДАКТИР.

НАЗВАНИЕ ПРОСТРАНСТВА	ПРОСМОТР	РЕДАКТИР.
Edu	<input checked="" type="checkbox"/>	<input type="checkbox"/>

Настройки тенанта: Группы

Для выдачи прав на пространства используются группы внутри тенанта. Добавить и удалить группу можно на странице "Группы":



Настройки тенанта / Tenant 1 / Настройки тенанта / Пользователи и группы / Группы

Поиск

НАЗВАНИЕ ГРУППЫ	ОПИСАНИЕ	ТИП	ПОЛЬЗОВАТЕЛИ	РАБОЧЕЕ ПРОСТРАНСТВО
group		Локальный	2	1
group1	description1	Локальный	0	0

Кликнув на группу откроется страница с четырьмя свойствами группы, которые и так видны в списке: название, описание, тип, количество пользователей. Но эта страница примечательна вкладкой: ПРОСТРАНСТВА, на которой можно добавить группе права на просмотр (read) и редактирование (write) содержимого пространств. При установке прав на редактирование (write) автоматически устанавливается право на просмотр.

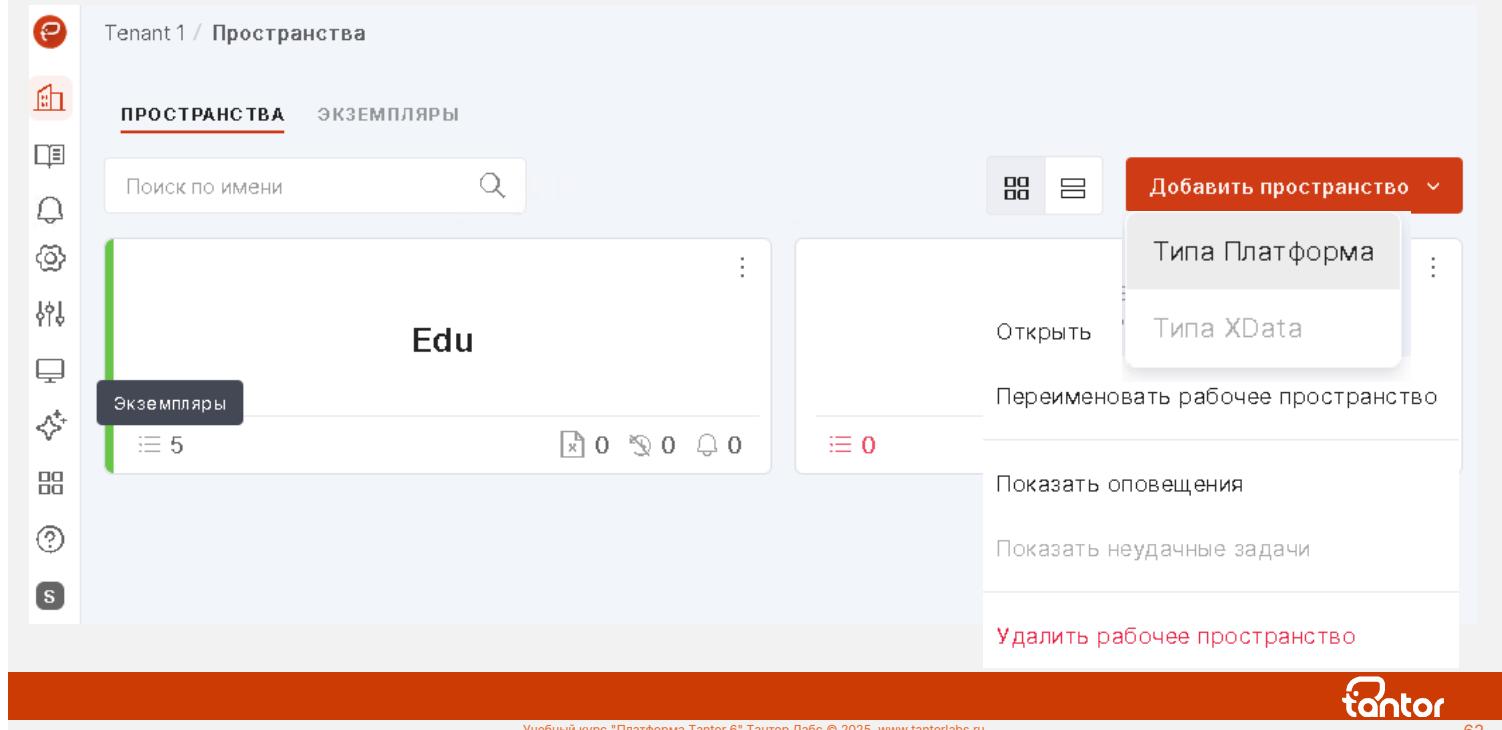
Право на "Редактирование" в предыдущих версиях Платформы называлось "Администратор пространства".

Право на "Просмотр" в предыдущих версиях Платформы называлось "Пользователь с правами просмотра" (Viewer).

вкладка ПОЛЬЗОВАТЕЛИ просто перенаправляет на страницу "Пользователи".

Тенант: Пространства (Workspaces)

- содержит экземпляры PostgreSQL и кластера Patroni



Tenant 1 / Пространства

ПРОСТРАНСТВА ЭКЗЕМПЛЯРЫ

Поиск по имени

Добавить пространство

Типа Платформа

Типа XData

Открыть

Переименовать рабочее пространство

Показать оповещения

Показать неудачные задачи

Удалить рабочее пространство

Edu

Экземпляры

5

0 0 0

0

tantor 62

Тенант: Пространства (Workspaces)

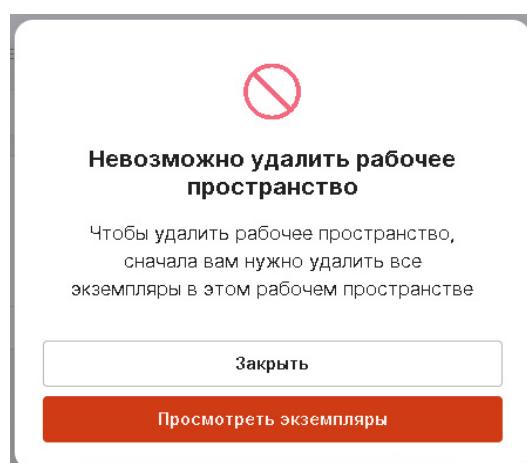
Пространства (workspaces) - логический объект, который содержит объекты управления: экземпляры, обслуживающие кластера баз данных PostgreSQL и кластера Patroni. На конкретное рабочее пространство можно дать привилегию группе: просмотр (read) и редактирование (write).

При добавлении рабочего пространства достаточно указать его название, которое должно быть уникальным в пределах тенанта, регистр букв важен. Можно создать несколько рабочих пространств с одинаковым названием, но разным регистром букв: "Edu1" и "edu1".

Пиктограммы снизу плашек более понятны в строковом отображении (следующая страница учебника) обозначают число: экземпляров "список" (2), невыполненных по расписанию задач планировщика "листок с крестиком" (1), невыполненных бэкапов "перечеркнутый полукруг" (0), число незакрытых оповещений "колокольчик" (4).

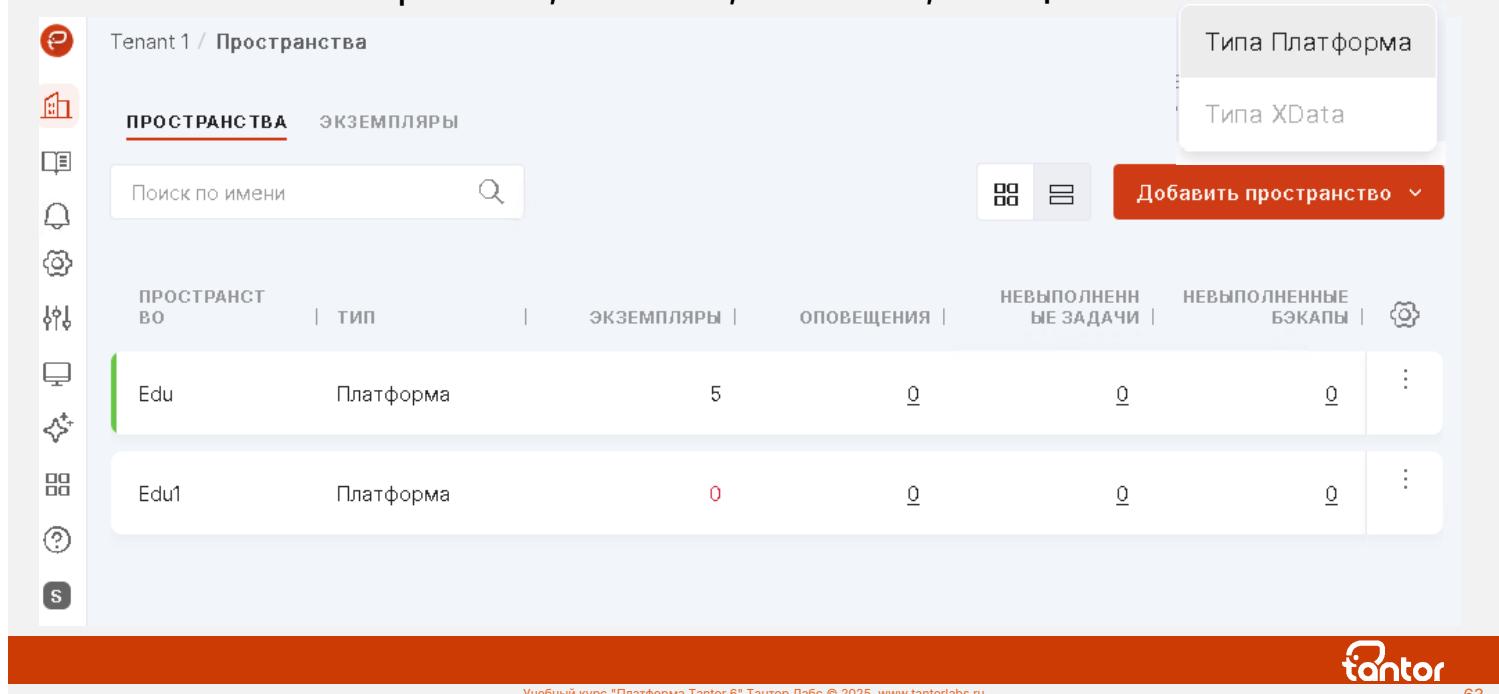
При нажатии на значок "три точки" в правом верхнем углу плашки появится всплывающее меню с пунктами: Открыть, Переименовать, Показать оповещения, Показать неудачные задачи, Удалить.

Удалить рабочее пространство можно только, если в нём нет экземпляров, иначе выдастся ошибка:



Тенант: Пространства

- линия слева от названия отражает состояние пространства и может быть: красной, желтой, зелёной, бесцветной



НАЗВАНИЕ	ТИП	ЭКЗЕМПЛЯРЫ	ОПОВЕЩЕНИЯ	НЕВЫПОЛНЕННЫЕ ЗАДАЧИ	НЕВЫПОЛНЕННЫЕ БЭКАПЫ	...
Edu	Платформа	5	0	0	0	...
Edu1	Платформа	0	0	0	0	...

Тенант: Пространства

Нажав на значок две линии слева от красной кнопки, можно поменять отображение списка пространств с плашек на строки, а нажав на значок четыре квадрата рядом вернуть обратно отображение в виде плашек.

Значок четыре квадрата слева внизу страницы означает совсем другое - он открывает меню со списком модулей.

Цветовая индикация состояния пространства:

Красная линия - в пространствах есть экземпляры PostgreSQL и существуют оповещения с уровнем важности "Проблема", касающиеся одного **или** нескольких экземпляров пространства.

Желтая линия - в пространствах есть экземпляры PostgreSQL и существуют оповещения с уровнем важности "Предупреждение", касающиеся одного **или** нескольких экземпляров пространства.

Зеленая линия - в пространствах есть экземпляры PostgreSQL, нет ошибок или предупреждений, касающихся хотя бы одного из экземпляров пространства.

Если полоска бесцветная (не видна), то в пространствах нет экземпляров.

При нажатии на оповещения откроется общая страница оповещений тенанта, на которую можно попасть кликнув на значок колокольчика в левой полосе страницы, но с включённым фильтром по названию пространства.

Практика 3. Тенанты и пространства

1. Создайте тенант с названием Tenant 2
2. В созданном тенанте создайте три пространства с названиями Workspace 1, Workspace 2, Workspace 3 типа Платформа
4. Поменяйте название Workspace 2 на Reanmed.
5. Удалите пространство Renamed
6. В настройках Tenant 1 найдите список пользователей
7. Найдите список владельцев тенанта Tenant 1
8. Найдите список владельцев системы
9. Посмотрите срок действия лицензии на использование Платформы и число лицензированных ядер процессоров
10. Посмотрите ядра процессоров, обслуживающих какие экземпляры PostgreSQL учитывается в лицензии

Практика 3

1. Создайте тенант с названием Tenant 2
2. В созданном тенанте создайте три пространства с названиями Workspace 1, Workspace 2, Workspace 3 типа Платформа
4. Поменяйте название Workspace 2 на Reanmed.
5. Удалите пространство Renamed
6. В настройках Tenant 1 найдите список пользователей
7. Найдите список владельцев тенанта Tenant 1
8. Найдите список владельцев системы
9. Посмотрите срок действия лицензии на использование Платформы и число лицензированных ядер процессоров
10. Посмотрите ядра процессоров, обслуживающих какие экземпляры PostgreSQL учитывается в лицензии



4

Платформа

Разграничение доступа

Виды пользователей Платформы Tantor

- **Владельцы системы (System owners)** - полные права на все действия в Платформе во всех тенантах
- **Владельцы тенанта (Tenant owners)** - локальные пользователи, которые имеют полный доступ к одному или нескольким тенантам
 - Внутри других тенатов могут не быть владельцами, а быть обычными локальными пользователями, права которым даются только через группы
- **Внешние пользователи и внешние группы** - импортированные по протоколу LDAP в Платформу
 - при входе в Платформу указывают "логин", а не email
 - аутентификацию будет выполнять LDAP-сервер
- В Платформе есть **локальные группы**

Виды пользователей Платформы Tantor

Владельцы системы (System owners) - полные права на все действия в Платформе во всех тенантах. Не могут быть добавлены как пользователи, в том числе владельцы тенантов. По умолчанию, присутствуют в списке владельцев всех тенантов, которые существовали при создании нового владельца системы. Если создать тенант, то в списке владельцев будет только владелец системы, который создал этот тенант.

Владельцы тенанта (Tenant owners) - локальные пользователи, которые имеют полный доступ к одному или нескольким тенантам и пространствам и экземплярам внутри этих тенантов. Внутри других тенатов могут не быть владельцами, а быть обычными локальными пользователями, права которым даются только через группы.

Локальные пользователи - не добавленные в список владельцев тенанта пользователи. В других тенантах могут быть добавлены в список владельцев тенанта. Пользователя можно убрать из списка владельцев тенанта, при этом он не будет удалён из Платформы и удаление не повлияет на права в других тенантах.

После добавления пользователя в Платформу как обычного или владельца тенанта, этот пользователь не будет выводиться в списке пользователей других тенантов. Его можно добавить в другие тенанты, это будет тот же самый пользователь Платформы, его имя и пароль не останутся прежними. Однако, находясь в любом тенанте можно изменить имя, пароль, статус активности пользователя и это подействует и будет видно во всех тенантах. Например, пользователь станет неактивен во всех тенантах и не сможет подсоединиться к Платформе. Добавив пользователя в тенант удалить его из этого тенанта и оставить в других где он есть невозможно, он будет высовчиваться в тех тенантах, в которых был когда-либо добавлен.

Удаление пользователя из списка пользователей находясь в любом тенанте полностью удалит пользователя из Платформы и из всех тенантов. Чтобы удалить пользователя нужно установить ему статус неактивен. Чтобы установить статус неактивен, нужно найти его в списке пользователей тенанта, где он не является владельцем и там можно будет поменять статус активен/неактивен.

Внешние пользователи и внешние группы - импортированные по протоколу LDAP в Платформу. После импорта внешние пользователи не связаны с локальными группами и локальными пользователями. Внешние пользователи при входе в Платформу указывают "логин", а не email. В Платформе есть **локальные группы**. Пользователи могут находиться в нескольких группах с разными правами.

Группы пользователей

- В Платформе есть **локальные группы**
- Пользователи могут находиться в нескольких группах с разными правами
- Права на пространства пользователям (кроме владельцев системы и тенантов) даются только через группы
- В группы добавляются одно или несколько пространств
 - На каждое пространство устанавливается привилегия просмотр (read) и/или редактирование (write)
- Одного пользователя можно добавить в несколько групп
 - Будет действовать объединение всех прав этих нескольких групп

Группы пользователей

В Платформе есть **локальные группы**. Пользователи могут находиться в нескольких группах с разными правами. Права на пространства пользователям (кроме владельцев системы и тенантов) даются только через группы. В группы добавляются одно или несколько пространств. На каждое пространство устанавливается привилегия просмотр (read) и/или редактирование (write).

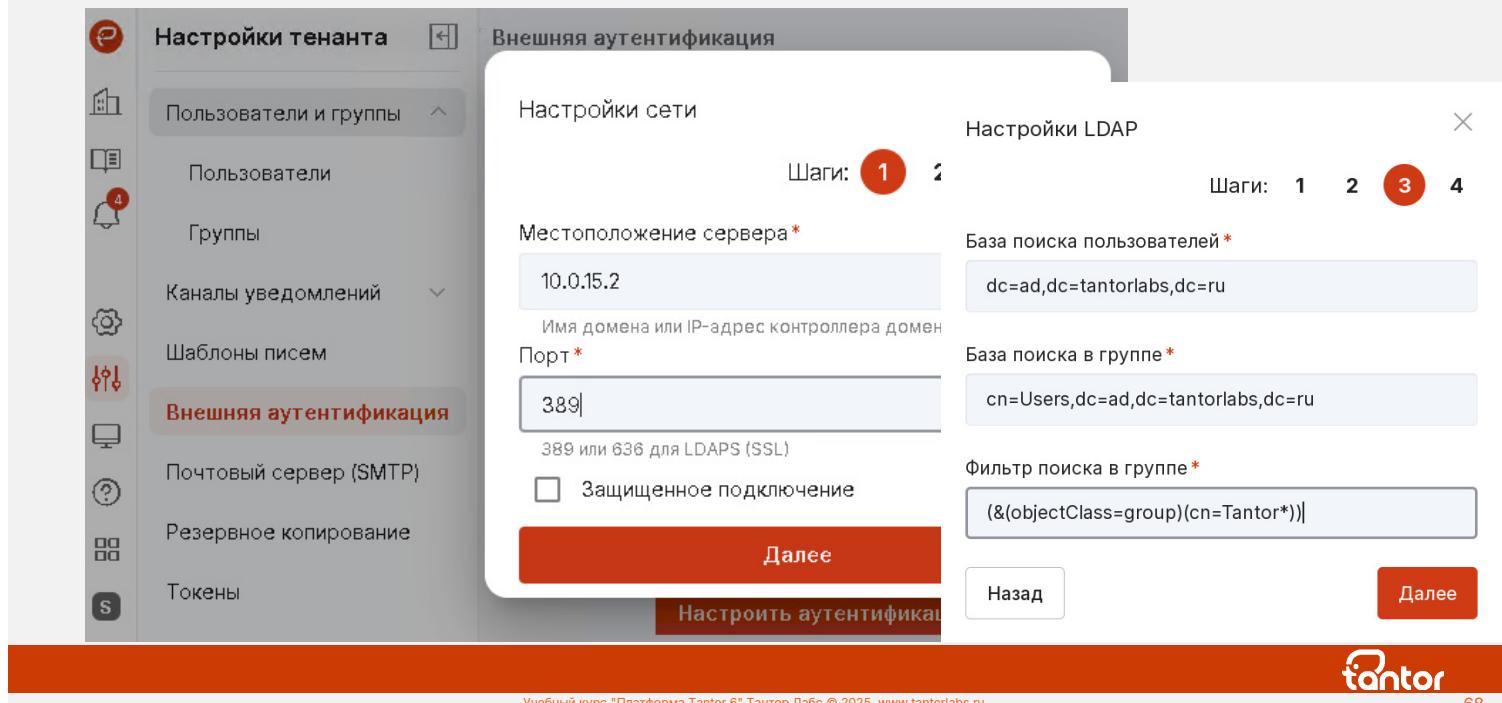
Одно пространство (workspace) может быть в нескольких группах своего тенанта с разными или одинаковыми правами.

Одного пользователя можно добавить в несколько групп. Будет действовать объединение всех прав этих нескольких групп.

Внешние группы импортируются по протоколу LDAP в Платформу.

Внешняя аутентификация

- используется протокол LDAP



Настройки тенанта

Пользователи и группы

Пользователи

Группы

Каналы уведомлений

Шаблоны писем

Внешняя аутентификация

Почтовый сервер (SMTP)

Резервное копирование

Токены

Настройки сети

Настройки LDAP

Шаги: 1 2 3 4

Местоположение сервера *

10.0.15.2

Имя домена или IP-адрес контроллера домен

Порт *

389

389 или 636 для LDAPS (SSL)

Защищенное подключение

Далее

Назад

База поиска пользователей *

dc=ad,dc=tantorlabs,dc=ru

База поиска в группе *

cn=Users,dc=ad,dc=tantorlabs,dc=ru

Фильтр поиска в группе *

(&(objectClass=group)(cn=Tantor*))

Далее

tantor

68

Внешняя аутентификация

В Платформе есть возможность не создавать пользователей Платформы, а импортировать их по LDAP-протоколу. Например, из служб Active Directory, FreeIPA, ALD pro, параметры подключения к которым приведены в документации Платформы Tantor. Для других LDAP-каталогов нужно знать и указать путь к списку пользователей и групп в LDAP-каталоге, шаблон поискового запроса на членство пользователя в группах, название атрибутов класса пользователя: логин, имя, фамилия, email.

Внешние пользователи будут аутентифицироваться внешним LDAP-сервером. Внешние группы также могут импортироваться, единственное их названия и описание не меняются на страницах Платформы.

Тип у таких пользователей и групп будет отображаться как "Внешний", а не "Локальный". При удалении LDAP-каталога, удаляются все импортированные из него пользователи и группы. Преимущество внешних пользователей в том, что их не нужно создавать, управлять паролями и членством пользователей в группах. Это выполняется в LDAP-сервере.



Удалить интеграцию с LDAP

<https://docs.tantorlabs.ru/tp/6.1/admin/ldap.html>

Это действие удалит с платформы всех пользователей и группы, связанные с LDAP. Удаленные ресурсы не могут быть восстановлены.

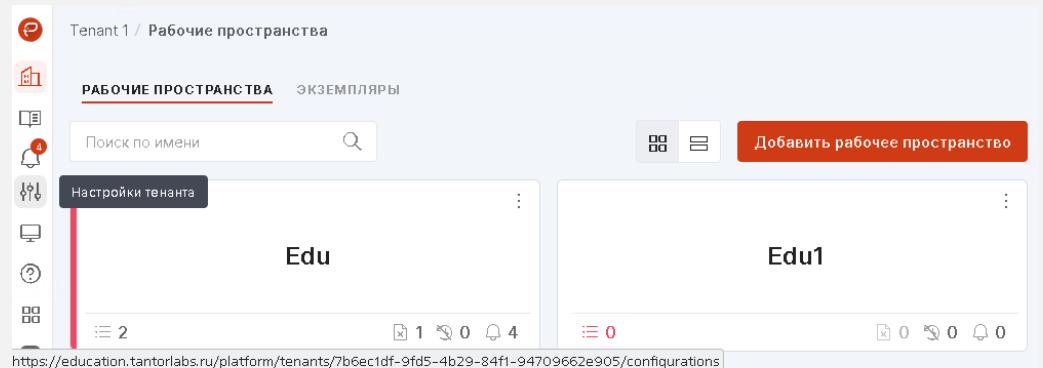
Для подтверждения введите "удалить" *

Отменить

Удалить

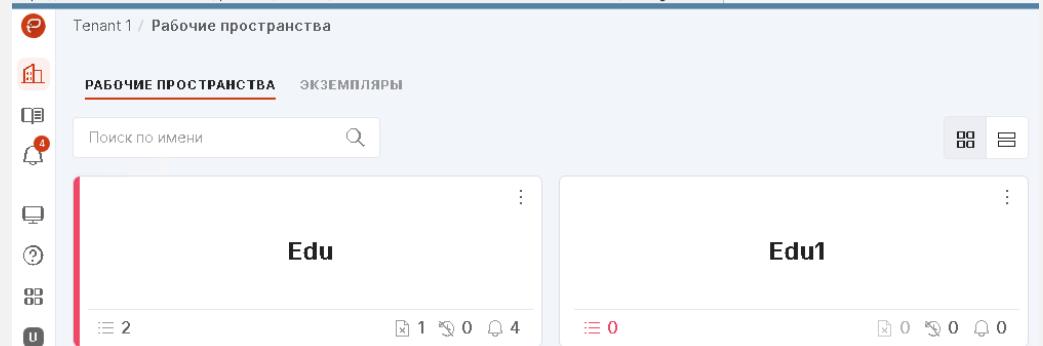
Страница тенанта

- владельца тенанта:



Screenshot of the Tenant 1 workspace list. The interface includes a sidebar with icons for Tenant, Home, Configuration, and Notifications (4). The main header says "Рабочие пространства" (Workspaces) and "Экземпляры" (Instances). A search bar says "Поиск по имени" (Search by name). A red button at the top right says "Добавить рабочее пространство" (Add workspace). The workspace list shows two entries: "Edu" (2 instances) and "Edu1" (0 instances). Each entry has a "Настройки тенанта" (Tenant settings) button. The URL in the address bar is <https://education.tantolabs.ru/platform/tenants/7b6ec1df-9fd5-4b29-84f1-94709662e905/configurations>.

- обычного пользователя:



Screenshot of the Tenant 1 workspace list for a regular user. The interface is identical to the owner's view, showing the same sidebar, header, and workspace list. The workspace list shows "Edu" (2 instances) and "Edu1" (0 instances). The URL in the address bar is <https://education.tantolabs.ru/platform/tenants/7b6ec1df-9fd5-4b29-84f1-94709662e905/configurations>.

Страница тенанта

На странице тенанта выводится список рабочих пространств.

У пользователя с правом владения тенантом на странице есть графические элементы:

- 1) красная кнопка с надписью "Добавить рабочее пространство" ("Add workspace")
- 2) значок "Настройки тенанта" на левой линейке меню.

Отображение списка в виде плашек (прямоугольников) можно поменять на отображение в виде строк, нажав на значок с двумя полосками справа вверху страницы, слева от красной кнопки.

Страница тенанта

- право write:

Tenant 1 / Рабочие пространства

РАБОЧИЕ ПРОСТРАНСТВА ЭКЗЕМПЛЯРЫ

Поиск по имени

РАБОЧЕЕ ПРОСТРАНСТВО ЭКЗЕМПЛЯРЫ ОПОВЕЩЕНИЯ

Имя	Кол-во	Кол-во	Кол-во	Кол-во
Edu	2	6	0	0
Edu1	0	0	0	0

- право read:

Tenant 1 / Рабочие пространства

РАБОЧИЕ ПРОСТРАНСТВА ЭКЗЕМПЛЯРЫ

Поиск по имени

РАБОЧЕЕ ПРОСТРАНСТВО ЭКЗЕМПЛЯРЫ ОПОВЕЩЕНИЯ НЕВЫПОЛНЕННЫЕ ЗАДАЧИ

Имя	Кол-во	Кол-во	Кол-во
Edu	2	6	1
Edu1	0	0	0

Страница тенанта

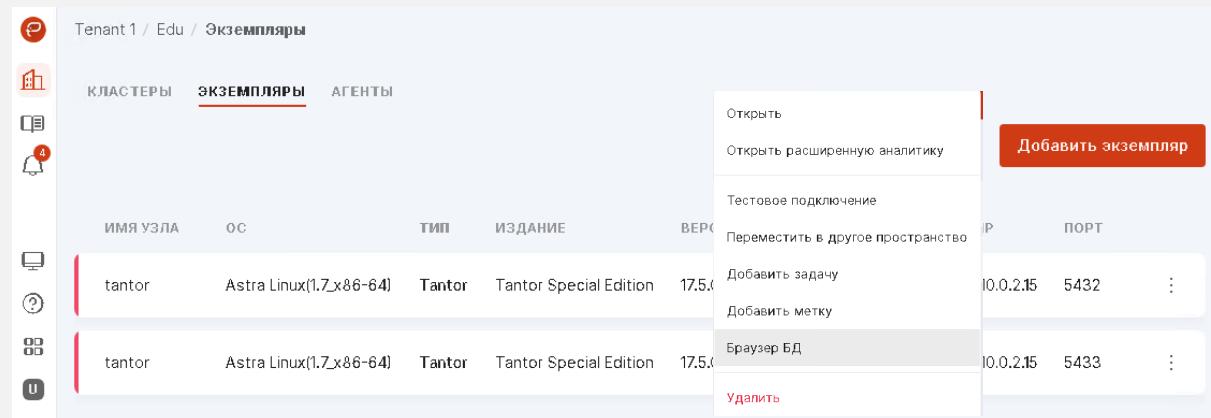
На слайде показано отображение рабочих пространств в виде списка для обычных пользователей (не владельцев).

Пространства, на которые не даны привилегии не отображаются в списке. Всего есть два вида прав: read и/или write, которые даются через группы пользователей.

Если нажать на значок три точки в правой части строки или плашки пространства, то откроется всплывающее меню. Для права read в меню три строки: Открыть, Показать оповещения, Показать неудачные задачи. Для права write+read (без read не устанавливается) дополнительно выдаются: Переименование пространства, Удалить пространство.

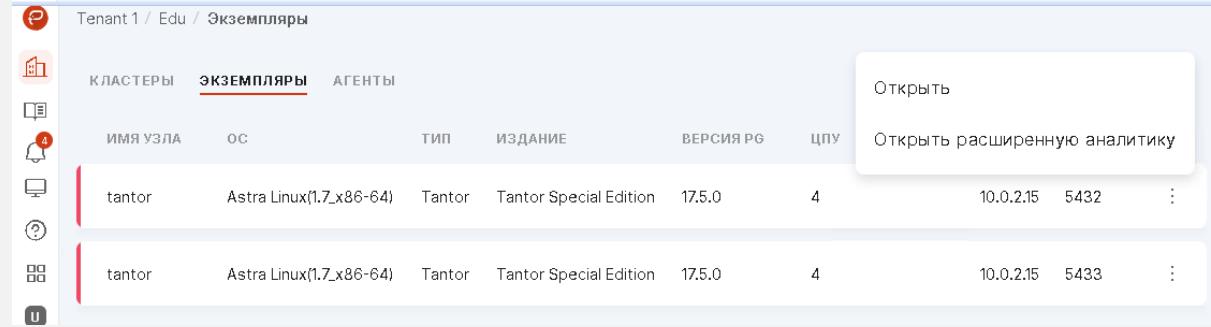
Страница пространств

- write:



ИМЯ УЗЛА	ОС	ТИП	ИЗДАНИЕ	ВЕРСИЯ PG	ЦПУ	ПОРТ	Действия	
tantor	Astra Linux(1.7_x86-64)	Tantor	Tantor Special Edition	17.5.0	4	10.0.2.15	5432	⋮
tantor	Astra Linux(1.7_x86-64)	Tantor	Tantor Special Edition	17.5.0	4	10.0.2.15	5433	⋮

- read:



ИМЯ УЗЛА	ОС	ТИП	ИЗДАНИЕ	ВЕРСИЯ PG	ЦПУ	ПОРТ	Действия	
tantor	Astra Linux(1.7_x86-64)	Tantor	Tantor Special Edition	17.5.0	4	10.0.2.15	5432	⋮
tantor	Astra Linux(1.7_x86-64)	Tantor	Tantor Special Edition	17.5.0	4	10.0.2.15	5433	⋮

Страница пространств

На странице пространства есть три вкладки: КЛАСТЕРЫ, ЭКЗЕМПЛЯРЫ АГЕНТЫ.

У пользователя с правом write отображается красная кнопка "Добавить экземпляр" (на вкладке ЭКЗЕМПЛЯРЫ) или "Добавить кластер Patroni" (на вкладке КЛАСТЕРЫ). Страница на вкладке АГЕНТЫ не различаются.

При клике на три точки, у пользователя с правом write во всплывающем меню есть дополнительные пункты:

1) Тестовое подключение - агент попытается подключиться к базе данных экземпляра под своей ролью кластера PostgreSQL и выдаст результат во всплывающем окне.

2) Переместить в другое пространство

3) Добавить задачу

4) Добавить метку

4) Браузер БД - открывает всплывающее окно для ввода деталей подсоединения к экземпляру. После нажатия на красную кнопку "Подключиться" открывается страница "Схема данных", с возможностью открыть "SQL-редактор".

5) Удалить.

У пользователя с правом read выдаются пункты

1) Открыть

2) Открыть расширенную аналитику.

Подключение к БД

Выберите БД или введите название *

postgres

Пользователь для подключения к БД *

postgres

Пароль для подключения к БД *

.....

Отменить

Подключиться

Страница экземпляра

- у write есть пункт меню:
- Настройки мониторинга
- также есть дополнительные кнопки и графические элементы на страницах

Учебный курс "Платформа Tantor 6" Тантор Лабс © 2025 www.tantolabs.ru

Страница экземпляра

Страница экземпляра - основная, с которой работает администратор баз данных (DBA). У пользователя с правом write на экземпляр, в меню дополнительный пункт "Настройки мониторинга", а также больше графических элементов на других страницах.

В примере на слайде открыт пункт меню "Обслуживание". У пользователя с правом write есть красная кнопка "Собрать данные".

Все отличия нет смысла перечислять и запоминать. Достаточно знать, что пользователь с правом read может просматривать данные по работе экземпляра и работе Платформы с экземпляром (уведомления), но не может вносить изменения в работу экземпляра PostgreSQL.

Право read предназначено для мониторинга, что является частью функционала Платформы Tantor. Право write - для управления кластерами баз данных PostgreSQL и Patroni.

Практика 4

1. Войдите во вкладку управления пользователями
2. Создайте нескольких пользователей, измените имя пользователя, удалите пользователя
3. В контекстном меню настроек войдите во вкладку управления группами пользователей, создайте несколько групп пользователей, измените имя группы
4. Распределите пользователей между группами, соотнесите группы пользователей с пространствами и уровнями полномочий

Практика 4

1. Войдите во вкладку управления пользователями
2. Создайте нескольких пользователей, измените имя пользователя, удалите пользователя.
3. В контекстном меню настроек войдите во вкладку управления группами пользователей, создайте несколько групп пользователей, измените имя группы
4. Распределите пользователей между группами, соотнесите группы пользователей с пространствами и уровнями полномочий



5

Экземпляры PostgreSQL

Мониторинг



Панель управления Платформой

- 1 - переход к списку тенантов
- 2 - переход к пространствам тенанта
- 3 - логи событий, относящихся к выбранному тенанту
- 4 - открывает страницу "Оповещения"
- 5 - смена темы, позволяет выбрать одну из тем:
 - светлая, тёмная, системная
 - значок меняется в зависимости от выбранной темы
- 6 - помощь, позволяет перейти к Базе знаний и документации Платформы, а также открывает окно с информацией об установленной версии Платформы
- 7 - модули Платформы
- 8 - аккаунт
 - на значке отображается первая буква имени пользователя

Панель управления Платформой

Появилась в 6 версии Платформы, располагается в левой части страниц.

1 - переход к списку тенантов

2 - переход к пространствам тенанта

3 - журнал событий, открывает страницу с логами событий, относящихся к данному тенанту

4 - открывает окно "Оповещения"

5 - Смена темы, позволяет выбрать одну из тем оформления Платформы:

светлая;

тёмная;

системная, автоматически меняет оформление Платформы на светлое или тёмное в зависимости от темы на вашем ПК.

6 - Помощь, позволяет перейти к Базе знаний и документации Платформы, а также открывает окно с информацией об установленной версии Платформы.

7 - Модули Платформы

8 - Аккаунт

https://docs.tantorlabs.ru/tp/6.1/instances/ug_platform_control_panel.html

Экземпляры PostgreSQL

- СПИСОК пространств тенанта:

- СПИСОК ЭКЗЕМПЛЯРОВ ВО ВСЕХ ПРОСТРАНСТВАХ ТЕНАНТА:

Экземпляры PostgreSQL

На стартовой странице Платформы, которая по умолчанию показывается после аутентификации, выдаётся список тенантов.

Если кликнуть на тенант, то откроется страница с пространствами, входящими в этот тенант и на которые есть права у пользователя.

На второй вкладке страницы "Экземпляры" выдаются все экземпляры, входящие в пространства данного тенанта. На этой вкладке можно выделить экземпляры, привязать группу параметров конфигурации (одну на выбор из списка) и синхронизировать (применить значения группы параметров к экземпляру) параметры выбранных экземпляров с их группами параметров.

Экземпляр с привязанной группой отображается зеленой галочкой в столбце "ПРИВЯЗ." Экземпляр с параметрами конфигурации, значения которых синхронизированы с группой параметров отображается зеленой галочкой в столбце "СИНХР." После нажатия на кнопку "Синхронизировать с группой параметров" появится окно со списком параметров: старые, новые значения и нужен ли restart экземпляра. Если новые значения неверные, можно отказаться от синхронизации и разбираться с группой параметров.

Цветовой индикатор статуса экземпляра:

красная линия - у экземпляра есть критический алёрт (оповещение);

жёлтая линия - у экземпляра есть предупреждающий алёрт;

зелёная линия - у экземпляра нет алёртов.

https://docs.tantolabs.ru/tp/6.1/instances/admin_instances.html

Страница тенанта со списком экземпляров

- на странице три вкладки: КЛАСТЕРЫ, ЭКЗЕМПЛЯРЫ, АГЕНТЫ

ИМЯ УЗЛА	ОС	ТИП	ИЗДАНИЕ	ВЕРСИЯ PG	ЦПУ	МЕТКА	IP	ПОРТ
tantor	Astra Linux(1.7.x86-64)	Tantor	Tantor Special Edition	17.5.0	4		10.0.2.15	5432
tantor	Astra Linux(1.7.x86-64)	Tantor	Tantor Special Edition	17.5.0	4		10.0.2.15	5433

Страница тенанта со списком экземпляров

Если кликнуть на рабочее пространство на странице тенанта, то откроется страница со списком экземпляров в этом пространстве. На странице три вкладки: КЛАСТЕРЫ (Patroni), ЭКЗЕМПЛЯРЫ (PostgreSQL), АГЕНТЫ (Платформы). В таблице отображаются основные характеристики этих объектов управления.

При нажатии на значок три точки появится контекстное меню. Значения пунктов меню:

"Тестовое подключение" позволяет проверить, может ли Платформа подключиться к экземпляру. Если есть проблемы с подключением, открывается окно с описанием проблемы.

"Добавить метку" - позволяет добавить метку (описание) экземпляра, чтобы идентифицировать его. Например, "Primary" или "для тестов".

Если экземпляр входит в кластер Patroni, то во всплывающем меню добавятся пункты:

"Перезагрузить", "Повторная инициализация", "Перезапустить" и не будет пункта "Переместить в другое пространство". Пункты добавлены так как, если экземпляр не сможет перезапуститься, то это будет проблема Patroni. Если экземпляр не входит в кластер Patroni, то при наличии ошибки в параметрах конфигурации экземпляра не запустится и для исправления параметров нужно будет вручную редактировать файлы параметров конфигурации, например, `postgresql.auto.conf`. В Платформе 6 версии Агент Платформы меняет параметры конфигурации командой `ALTER SYSTEM SET`, которая не работает на остановленном экземпляре. Функционала хранения версий файлов параметров конфигурации нет. В СУБД Tantor Postgres 17.5 появилась утилита `pg_disg_setup.py`, но утилита хранит только историю параметров расширений. Платформа 6 версии не использует эту утилиту, да и для ванильных версий PostgreSQL её нет.

Проблемы с параметрами могут быть, например, в том, что изменения 8 параметров конфигурации мастера передаются через WAL на реплики, где процесс `startup` проверяет, что значения 5 параметров на реплике не меньше, чем на мастере. Если это не так, `startup` останавливает применение WAL, а при остановке экземпляра реплики, экземпляр не запускается. Поэтому, перед остановкой и перезапуском экземпляра PostgreSQL стоит визуально проверить параметры конфигурации хотя бы в `postgresql.auto.conf` и иметь возможность его редактировать. Другими словами, иметь терминальную сессию с хостом экземпляра, а имея такую сессию легко перезапустить экземпляр в этой сессии.

<https://habr.com/ru/articles/903030/>

https://docs.tantorlabs.ru/tp/6.1/instances/instances_list.html

Пространство: Агенты

- на странице АГЕНТЫ можно получить команды обновления Агента:

The screenshot shows the Tantor platform interface. On the left, there is a sidebar with icons for Tenant, Edu, and Agents. The 'Agents' tab is selected. Below it, there are tabs for КЛАСТЕРЫ (Clusters), ЭКЗЕМПЛЯРЫ (Instances), and АГЕНТЫ (Agents). A search bar 'Поиск по конечной точке' (Search by endpoint) is present. The main area displays a table of agents with columns: КЛЮЧ (Key), ЭКЗЕМПЛЯРЫ (Instances), СТАТУС (Status), КОНЕЧНАЯ ТОЧКА (Endpoint), and ВЕРСИЯ (Version). One agent entry is selected, showing details: 43lWMZpQQ6h8pwSdgKz3yfiQENyfKWCW/qhU/8jilV81XHY+PCzdaA, 2 Connected Instance, 10.0.2.15, 5.3.0. A modal dialog box titled 'Обновить Агент' (Update Agent) is open. It contains a dropdown for 'Выберите ваш дистрибутив Linux:' (Select your Linux distribution) with options: Ubuntu, Rocky, Red Hat, RedOS, Fedora, Debian, CentOS, and Astra Linux. The 'Astra Linux' option is selected. A checkbox 'У меня нет доступа к Интернету, используйте локальную/оффлайн установку' (I have no internet access, use local/offline installation) is checked. Below the dropdown, there are two sections: 'Обновить Агент' (Update Agent) with the command 'curl "https://education.tantorlabs.ru:8443/installations/3116d60760?hash=cHRRjAZtXcocomRtl9Uyl' and 'apt-get install -y /tmp/pmaagent.deb'; and 'Запустить Агент (пример только для Systemd)' (Start Agent (example only for Systemd)) with the command 'systemctl daemon-reload && systemctl restart pmaagent'. At the bottom of the dialog are 'Отменить' (Cancel) and 'Готово' (Done) buttons. A 'Закрыть' (Close) button is located at the bottom right of the modal. The footer of the page includes the Tantor logo and the text 'Учебный курс "Платформа Tantor 6" Тантор Лабс © 2025 www.tantorlabs.ru'.

Пространство: Агенты

На странице АГЕНТЫ можно получить команды для обновления агента до текущей версии Платформы. На то, что версия Агента не соответствует текущей версии Платформы указывает треугольник красного цвета рядом с версией Агента. При использовании Агента более старой версии новый функционал будет отсутствовать. Например, будут отсутствовать пункты меню, которые относятся к новшествам. Старый функционал при этом будет работать.

Цвет вертикальной линии слева у Агентов означает:

зелёный - Агент доступен;

красный - Агент не доступен.

Удалить Агент можно только в том случае, если он не обслуживает ни один экземпляр.

Пункт меню "Список экземпляров" нужен, чтобы определить, какие экземпляры обслуживает Агент.

https://docs.tantorlabs.ru/tp/6.1/instances/ug_agents.html

The screenshot shows the 'Update Agent' dialog box. At the top, it says 'Выберите ваш дистрибутив Linux:' (Select your Linux distribution). A dropdown menu is open, showing options: Ubuntu, Rocky, Red Hat, RedOS, Fedora, Debian, CentOS, and Astra Linux. The 'Astra Linux' option is selected. To the right of the dropdown, there are two dropdowns: 'Astra Linux' and '1.6+'. Below the dropdowns, there is a checkbox 'У меня нет доступа к Интернету, используйте локальную/оффлайн установку' (I have no internet access, use local/offline installation), which is checked. The 'Обновить Агент' (Update Agent) section contains the command 'curl "https://education.tantorlabs.ru:8443/installations/3116d60760?hash=cHRRjAZtXcocomRtl9Uyl' and 'apt-get install -y /tmp/pmaagent.deb'. The 'Запустить Агент (пример только для Systemd)' (Start Agent (example only for Systemd)) section contains the command 'systemctl daemon-reload && systemctl restart pmaagent'. At the bottom are 'Отменить' (Cancel) and 'Готово' (Done) buttons. A 'Закрыть' (Close) button is located at the bottom right of the dialog.

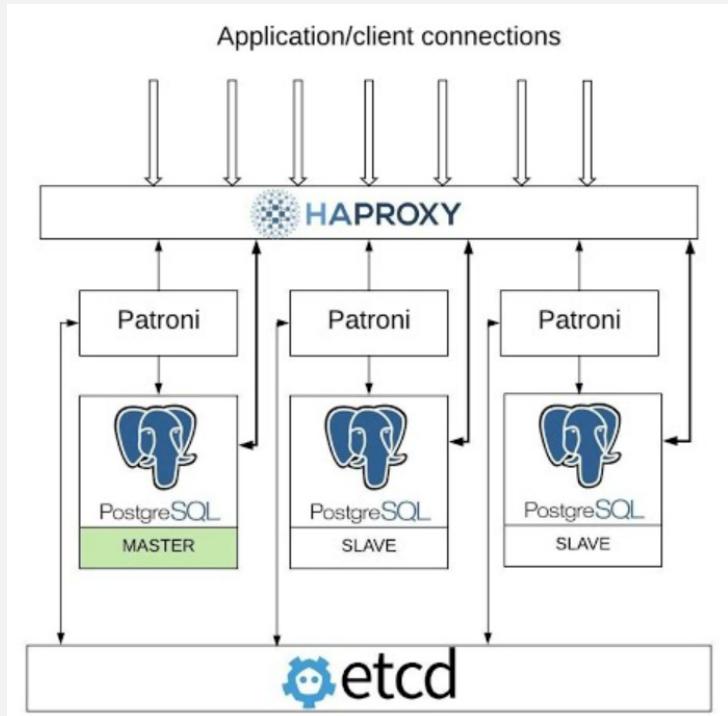


5-2

Patroni

Пространство: Кластеры Patroni

- Patroni - программа для управления кластером баз данных PostgreSQL и его физическими репликами
- Patroni в случае недоступности мастера, автоматически продвигает одну из реплик до мастера и начинает восстановление сбоявшего мастера
- Patroni хранит свои данные в распределенном хранилище конфигурации etcd



Пространство: Кластеры Patroni

Кластер баз данных PostgreSQL обслуживается только одним экземпляром PostgreSQL.

Patroni - это программа для управления кластером баз данных PostgreSQL и его физическими репликами. Patroni занимается мониторингом доступности мастера и, в случае его недоступности, автоматически продвигает (promote) одну из реплик до мастера и автоматически начинает восстановление сбоявшего мастера. Patroni может выполнять автоматическое или ручное продвижение и хранит свои данные в распределенном хранилище конфигурации etcd. Для подсоединения клиентов к кластеру PostgreSQL, обычно, используется балансировщик нагрузки (HAproxy), который подсоединяет клиента к текущему мастеру.

Patroni текущего мастера периодически обновляет статус ключа "leader" в etcd, имеющий срок жизни (TTL, Time To Live). Если срок истёк, реплики соревнуются за роль "leader". Patroni того экземпляра, который создал ключ "leader" выполняет promote реплики до мастера и перенастраивает на остальных экземплярах маршруты потоковой репликации, чтобы реплики забирали WAL с нового мастера. Бывшего мастера пытается восстановить в виде новой реплики утилитой pg_rewind. Также может добавлять ("инициализировать") новые реплики или пересоздавать существующие, используя утилиту pg_basebackup.

Переоценивать Patroni не стоит, качество программного обеспечения PostgreSQL существенно выше, чем etcd и Patroni.

Платформа может управлять экземплярами PostgreSQL, входящими в "кластер Patroni".

Для добавления "кластера Patroni" сначала нужно установить и настроить на несколько узлов (хостов, виртуальных машин):

- 1) кластер etcd (три хоста с тремя IP адресами)
- 2) язык Python, так как Patroni написан на этом языке
- 3) Patroni
- 4) PostgreSQL.

После запуска и конфигурирования Patroni экземпляры PostgreSQL можно добавить в Платформу Tantor. При добавлении экземпляра нужно установить флагок "Экземпляр является членом кластера Patroni". После добавления экземпляра в списке кластеров появится кластер Patroni.

https://docs.tantolabs.ru/tp/6.1/instances/ug_clusters_pages.html

Пространство: КЛАСТЕРЫ

- отображает кластеры Patroni:

Tenant 1 / Cluster / Кластеры

КЛАСТЕРЫ ЭКЗЕМПЛЯРЫ АГЕНТЫ

1 Поиск по имени 2 Добавить кластер Patroni 3 4 main 4 partial 4.0.5 6 11.5 GB 41.3 GB / 49.1 GB 2 / 2

Имя Кластера Состояние Patroni Кол-во ЦПУ Объем ОЗУ Диск основного Экземпляра Экземпляры

Открыть Мониторинг Конфигурация Пауза/Обслуживание

- цвет линии слева - состояние кластера Patroni:
- зелёная** - запущен, работает без ошибок
- жёлтая** - используется асинхронная репликация или не работает какой-то экземпляр реплики
- красная** - не запущен экземпляр мастера (primary) или не запущена реплика
- фиолетовая** - на кластере проводится обслуживание

Пространство: КЛАСТЕРЫ

1 - поле для поиска кластера по столбцу ИМЯ

2 - кнопка красного цвета, открывает окно добавления нового кластера

3 - столбцы с информацией о кластере:

"PATRONI" - версия Patroni;

"КОЛ-ВО ЦПУ" - общее число ядер процессоров на всём кластере Patroni;

"ОБЪЕМ ОЗУ" - общий объём оперативной памяти, доступной на всём кластере Patroni;

"ДИСК ОСНОВНОГО ЭКЗЕМПЛЯРА" - размер занятого/общего дискового пространства на экземпляре мастера (primary, основного, лидера) PostgreSQL;

"ЭКЗЕМПЛЯРЫ" - число экземпляров PostgreSQL в кластере Patroni.

Цвет линии слева отражает состояние кластера Patroni:

зелёная - запущен, работает без ошибок;

жёлтая - используется асинхронная репликация или не работает какой-то экземпляр реплики;

красная - не запущен экземпляр мастера (primary) или не запущена реплика;

фиолетовая - на кластере проводится обслуживание.

Если кликнуть на имя кластера, то отобразится страница кластера Patroni.

Если кликнуть на три точки справа строки, появится всплывающее меню. В этом меню можно поставить Patroni на паузу и перейти на страницу просмотра и редактирования параметров конфигурации экземпляров PostgreSQL "Конфигурация" ("Cluster config").

Страница кластера Patroni: ОБЗОР И УПРАВЛЕНИЕ

Tenant 1 / Cluster / Кластеры / main / Обзор и управление

ОБЗОР И УПРАВЛЕНИЕ		МОНИТОРИНГ		СОСТОЯНИЕ КЛАСТЕРА				ПОДРОБНОСТИ КЛАСТЕРА			
1	ВЕРСИЯ PATRONI	2	ВЕРСИЯ POSTGRES	ОБЪЕМ ОЗУ	КОЛ-ВО ЦПУ	ДИСК ОСНОВНОГО ЭКЗЕМПЛЯРА	ЭКЗЕМПЛЯРЫ	РЕПЛИКИ	3	4	5
1	4.0.5	2	16.8.0	11.5 GB	6	49.1 GB	3 / 3	2 / 2	3	4	5
										6	7

Конфигурация кластера

2

Пауза/Обслуживание

3

Страница кластера Patroni: ОБЗОР И УПРАВЛЕНИЕ

На странице две вкладки: "ОБЗОР И УПРАВЛЕНИЕ" и "МОНИТОРИНГ".

1- характеристики кластера

2 - открывает страницу с параметрами конфигурации экземпляров PostgreSQL, входящие в кластер Patroni. Эти параметры конфигурации одинаковые для всех экземпляров.

3 - "Пауза/Обслуживание" - приостанавливает активность Patroni. Во время паузы(обслуживания) Patroni не будет пытаться менять роли экземплярам, переинициализировать кластера PostgreSQL.

4 - состояние экземпляров PostgreSQL. Один из экземпляров - primary (основной, ведущий, мастер), остальные экземпляры - физические реплики.

"TTL" - время в секундах. Если по истечении этого времени мастер не обновит статус, начнутся выборы нового мастера (лидера).

По нажатию на три точки справа появится всплывающее меню:

Перезагрузить (Reload) - отправляет экземпляру команду reload, перечитать файлы параметров конфигурации без остановки экземпляра;

Повторная инициализация (Reinitialize) – заново инициализирует экземпляр;

Перезапустить (Restart) - останавливает и заново запускает экземпляр;

Переключиться (Switchover) - делает экземпляр мастером.

Ограничения для "Переключения":

Если есть синхронная реплика (параметр `synchronous_mode = true`), то переключиться можно только на нее. Если синхронной реплики нет (параметр `synchronous_mode = false`), то можно выполнить переключение на любую работающую реплику.

Страница кластера Patroni: МОНИТОРИНГ

Tenant 1 / Cluster / Кластеры / main / Мониторинг

ОБЗОР И УПРАВЛЕНИЕ МОНИТОРИНГ

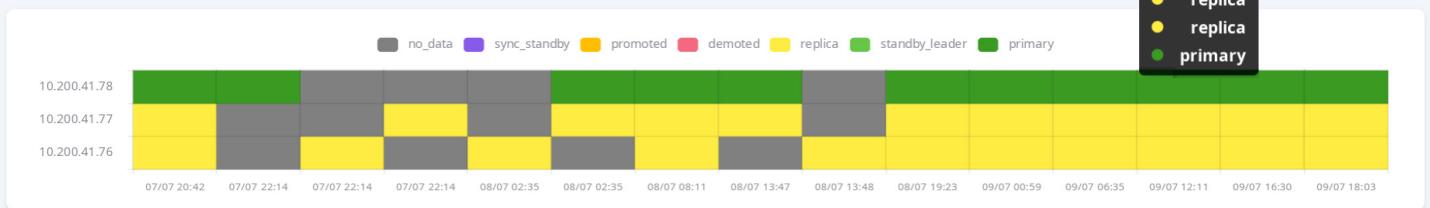
Patroni кластер

partial since 2025.07.08 14:51

Интервал выборки исторических данных

Последние 7 дней

Изменение ролей экземпляра 1



История изменений статуса узла кластера 2

ДАТА	ВРЕМЯ	10.200.41.76	10.200.41.78	10.200.41.77
08.07.2025	14:51	streaming	running	streaming
08.07.2025	14:51	streaming	running	streaming

Страница кластера Patroni: МОНИТОРИНГ

На вкладке "МОНИТОРИНГ"

Изменение ролей экземпляра - график, показывающий историю смены ролей экземпляров. На этом графике можно увидеть, как менялись роли экземпляров кластера за выбранный период. Период можно выбрать справа вверху страницы от 30 минут до 7 дней. если провести мышкой по графику, появится всплывающее окно, в котором будут отображаться роли экземпляров.

История изменений статуса узла кластера - таблица с историей изменения ролей (мастер или реплика) за отображаемый на странице период времени.

Страница кластера Patroni: Конфигурация

Tenant 1 / Cluster / Кластеры / main / Конфигурация

ПАРАМЕТРЫ PATRONI ПАРАМЕТРЫ POSTGRESQL

Информация о настройках

Patroni

failsafe_mode
Enables or disables the failsafe mode

loop_wait
Interval between cycles of the Patroni loop

3. Параметры, измененные вручную на сервере в файлах конфигурации postgresql (postgresql.conf/postgresql.auto.conf), не будут отображаться на этом экране. Вы можете видеть их только на экране конфигурации экземпляра.

4. Интерфейс корректно работает с Patroni версий от 3.3.0 до 4.0.5

Больше не показывать

OK

ТЕКУЩЕЕ ЗНАЧЕНИЕ РЕКОМЕНДУЕМОЕ ЗНАЧЕНИЕ

on 105

Применить настройки

Страница кластера Patroni: Конфигурация

На странице "Конфигурация" (Cluster config) кластера Patroni две вкладки: "ПАРАМЕТРЫ PATRONI" и "ПАРАМЕТРЫ POSTGRESQL".

При переходе на страницу появляется всплывающее окно с предупреждением, что Платформа считывает только те параметры, которые действуют, а не параметры из файлов конфигурации. Если поставить галочку в чекбоксе "Больше не показывать", то окно не будет автоматически появляться при переходе на страницу. Окно можно вернуть, на жёлтый треугольник левее значка фильтрации (воронки).

Если значения параметров на экземплярах PostgreSQL различаются, то такие параметры не показываются. Пример параметров, которые будут отличаться на мастере и репликах: `transaction_read_only` и `in_hot_standby` (на репликах `on`, на мастере `off`), `primary_conninfo`, `primary_slot_name` и не только.

Параметры Patroni, которые нельзя отредактировать через Платформу:

`pg_ident`, `pg_hba`, `ignore_slots`, `slots`, `pause`, `config_file`, `data_directory`, `transaction_read_only`.

Если поменять значения параметров, кнопка "Применить настройки" станет активной и её можно будет нажать. По нажатии на кнопку появится окно со списком новых значений и кнопкой "Применить и перезагрузить".

Страница кластера Patroni: Применение изменений

Tenant 1 / Cluster / Кластеры / main / Конфигурация

ПАРАМЕТРЫ PATRONI ПАРАМЕТРЫ POSTGRESQL

Поиск конфигурации 🔍

⚠ 💡 ⚡ Выбрать рекомендуемые настройки Применить настройки

Применение конфигурации ×

1 Поиск 🔍

	СТАРОЕ ЗНАЧЕНИЕ	НОВОЕ ЗНАЧЕНИЕ	В ОЖИДАНИИ
autovacuum_analyze_threshold	59	53	2 Перезагрузить
shared_buffers	857 MB	855MB	3 Перезапустить кластер

Отменить Применить и перезагрузить

autovacuum_freeze_max_age
Age at which to autovacuum a table to prevent transaction ID wraparound.

2,000,000,000 ✍ != 500,000,000 ℹ

tantor

85

Страница кластера Patroni: Применение изменений

Если поменять значения параметров, кнопка "Применить настройки" станет активной и её можно будет нажать. По нажатии на кнопку появится окно со списком новых значений и кнопкой "Применить и перезагрузить".

"Перезагрузить" (Reload) означает **перечитать** файлы конфигурации (`pg_reload_conf()`). Параметры, которые требуют перезапуска экземпляра ("перезапустить кластер") будут только сохранены в файле `postgresql.auto.conf`, экземпляры перезапускаться не будут.

Через Платформу можно перезапустить по одному экземпляры PostgreSQL, если они входят в кластер Patroni, но лучше перезапускать вручную в командной строке хоста экземпляра.

Практика 5а. Управление кластером Patroni

1. Посмотрите содержимое страниц пространств, экземпляров, агентов, кластера Patroni
2. Поставьте кластер Patroni на обслуживание
3. Выполните переключение на синхронную реплику
4. Пересоздайте асинхронную реплику
5. Посмотрите страницу с историей переключения ролей экземпляров кластера Patroni
6. Посмотрите страницу параметров Patroni

Практика 5б. Обновление агентов

1. Посмотрите страницу Репликация и Графики на примере экземпляров под управлением Patroni

Практика 5с. Обновление агентов

1. Обновите агенты Платформы

Практика 5а

1. Посмотрите содержимое страниц пространств, экземпляров, агентов, кластера Patroni
2. Поставьте кластер Patroni на обслуживание
3. Выполните переключение на синхронную реплику
4. Пересоздайте асинхронную реплику
5. Посмотрите страницу с историей переключения ролей экземпляров кластера Patroni
6. Посмотрите страницу параметров Patroni

Практика 5б

1. Посмотрите страницу Репликация и Графики на примере экземпляров под управлением Patroni

Практика 5с

1. Обновите агенты Платформы



Экземпляры PostgreSQL

Страница "Обзор экземпляра"

Страница Обзор

The screenshot shows the 'Overview' page of the Tantor Special Edition 17.5.0 monitoring interface. At the top, it displays the IP address and port: 10.0.2.15:5432, and the status: Self Managed, Tantor Special Edition 17.5.0. A red warning icon is present. The top right corner shows a 'Last 30 minutes' summary with a green checkmark. The left sidebar contains a navigation menu with items like 'Overview', 'Configuration', 'Monitoring', 'Browser', 'Profiler', 'Current Activity', 'Replication', 'Tablespaces', 'Graphs', 'Monitoring Settings', 'Extended Analytics', 'Beacon Monitoring', and 'Tasks'. A red warning icon is also on the sidebar. The main content area is divided into several cards:

- СЕССИИ:** 56 Max. кол-во подключений из 100. Среднее время запроса: 20.46 ms. A tooltip says: 'Нажмите, чтобы показать предупреждение' (Click to show warning).
- ТРАНЗАКЦИИ:** 471 tps.
- ЗАСИЛКИВАНИЕ:** Аварийная автоочистка.
- РЕПЛИКАЦИЯ:** Green checkmark.
- 5 САМЫХ ДЛЯТЕЛЬНЫХ ЗАПРОСОВ:** A chart showing the duration of the top 5 longest queries. The x-axis is time from 12/09 22:53 to 12/09 23:23. The y-axis is duration in seconds (secs) from 0 to 50. Legend: ethers (blue), INSERT INT... (red), SELECT ab... (blue), UPDATE pg... (yellow), UPDATE pg... (teal), UPDATE pg... (dark blue).
- ВВОД/ВЫВОД НА ДИСК /DEV/SDA1:** 3.8 MB/сек.
- РАЗМЕР ЛОГОВ:** 38 KB.
- РАЗМЕР ДАННЫХ:** 231 MB.
- ВВОД/ВЫВОД БЛОКОВ:** 94 К блок/сек.
- РАЗМЕР БЛОКОВ ИЗ БУФЕРА:** 100 %.
- РАЗМЕР ТОЧЕК:** 0 %.
- ВРЕМЕННЫЕ ФАЙЛЫ:** 0 B/сек.
- УВЕДОМЛЕНИЯ:** Оповещения, Подсказки системы.
- Проблема вкл. 10.0.2.15:** CPU Load Average AVG 1 minutes > 200%, Last value: 259.00%
- Проблема вкл. 10.0.2.15:** Total Waiting sessions > 10, Last value: 44.
- НАГРУЗКА ЦП X4:** 10.35 load1. A red warning icon is present.
- ДОСТУПНО ОЗУ:** 3.3 GB Объем ОЗУ 7.8 GB.
- СЕТЬ BR-D8CB75C1A797:** 35.2 KB/сек. A red warning icon is present. A tooltip says: '1 of 14'.
- ХРАНИЛИЩЕ ROOT:** 12.2 GB Общий объем хранения 38.1 GB.
- ФАЙЛЫ WAL:** 341 KB/сек.
- СТАТИСТИКА АВТООЧИСТКИ:** 0 макс. кол-во воркеров.

A red button at the bottom right says 'Просмотреть все сообщения' (View all messages). At the very bottom, it says 'Частота обновления данных - 1 минута' (Data refresh rate - 1 minute).

Страница Обзор

Страница "Обзор" экземпляра - основная, с которой начинают работу с экземпляром. В левой верхней части значок, кликнув на который можно скрыть меню, увеличив площадь в окне браузера: вернуть меню можно кликнув на значок: Чуть ниже отображается IP адрес и порт экземпляра PostgreSQL: 10.0.2.15:5432

Ниже - тип, сборка и версия СУБД

Ниже - метка и значок карандаша, кликнув на который можно отредактировать метку.

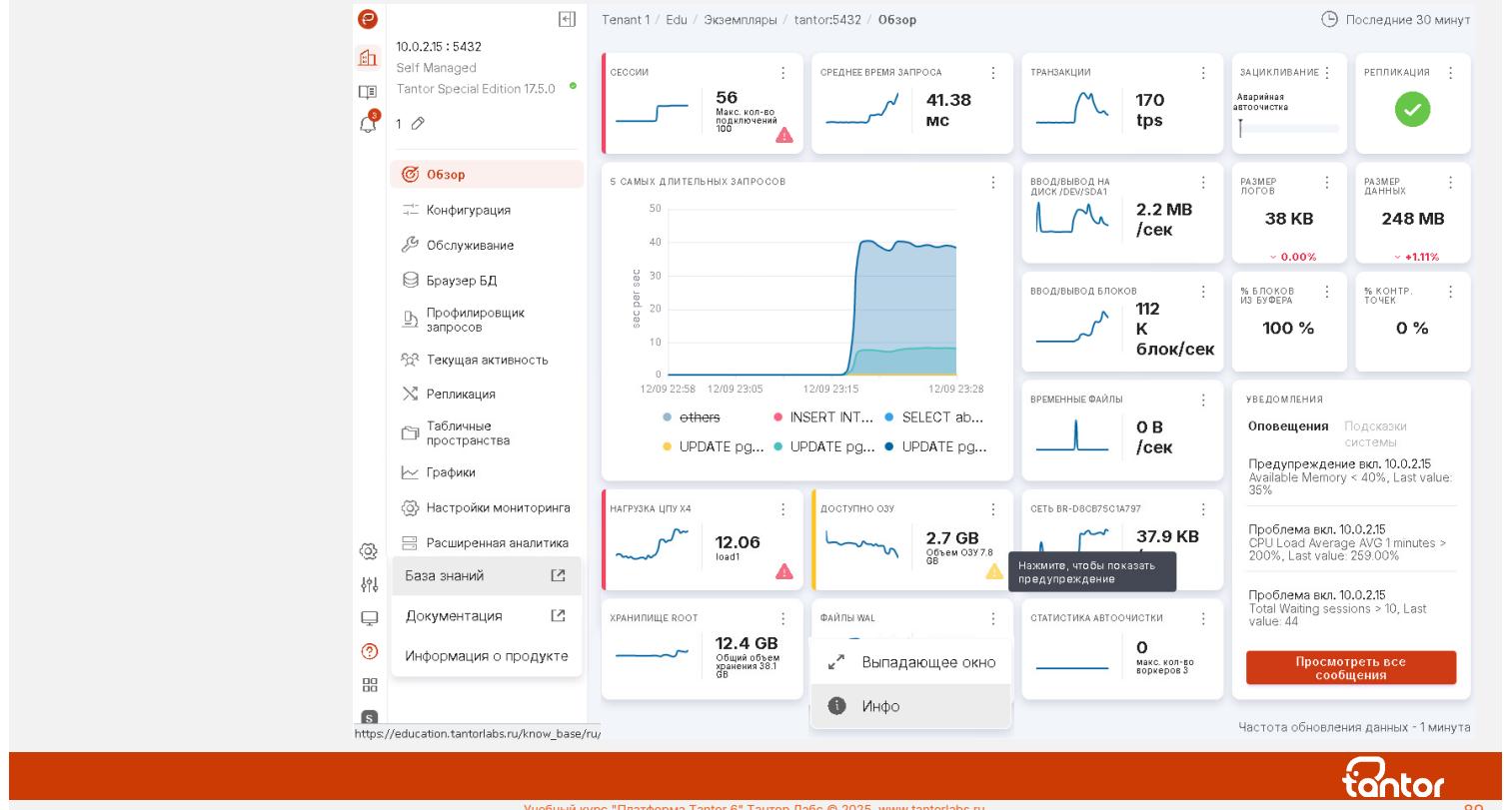
У двух плиток на слайде отображается красный треугольник, который на странице браузера немного мерцает: , при наведении курсора на треугольник появляется окно с текстом:

"Нажмите, чтобы показать предупреждения". При клике на треугольник откроется страница уведомлений. Красный цвет означает "проблема", жёлтый - "предупреждение":

При наличии проблем и предупреждений на плитке появляется линия слева, цвет которой такой же, как у треугольника.

<https://docs.tantorlabs.ru/tp/6.1/instances/overview.html>

Страница Обзор



Страница Обзор

На этом слайде у третьей плитки появился желтый прямоугольник и жёлтая линия. Добавилось предупреждение о том, что оперативной памяти стало мало.

На плитке УВЕДОМЛЕНИЯ (справа внизу) добавилась строка с этим предупреждением. Стока добавлена над предыдущими оповещениями с уровнем важности "Проблема".

В правом верхнем углу плиток есть значок с тремя точками, при нажатии на который откроются две опции:

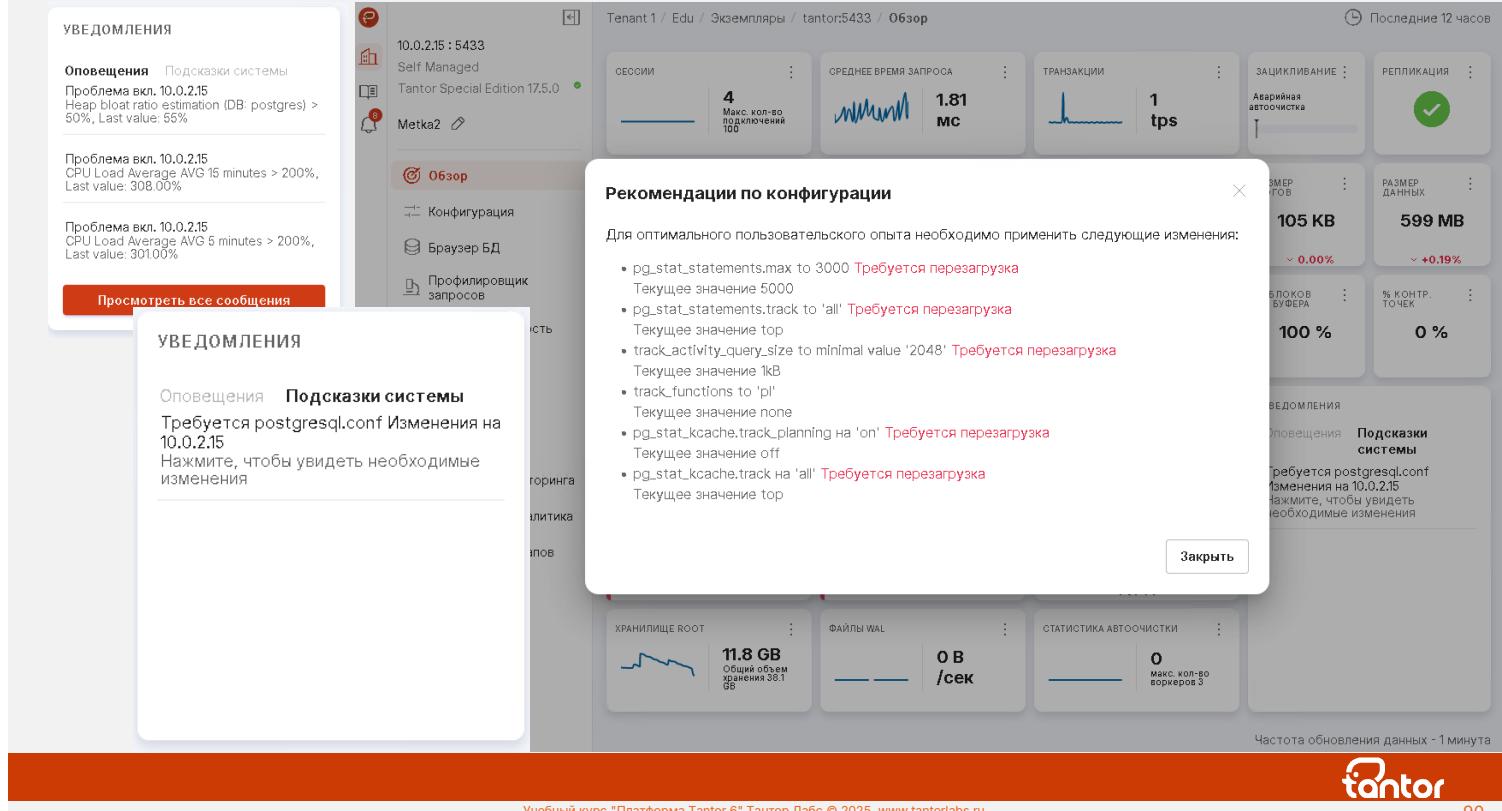
Выпадающее окно/Popout - откроется всплывающее окно с увеличенным графиком. В этом окне могут быть настройки отображения графика, а также диапазоны отображения данных от 30 минут до 7 дней.

Инфо/Info - всплывающее окно с техническим описанием того, что отображается на графике и ссылкой "Подробнее", по нажатию на которую в отдельном окне браузера появится страница Базы знаний Тантор о том, что отображается на графике.

База знаний Тантор установлена локально в Платформе и не требует доступа в интернет. База знаний доступна по адресу https://хост_Платформы/know_base/ru/

Ссылка на Базу знаний Тантор есть под значком "?" на левой вертикальной панели меню.

Страница Обзор -> УВЕДОМЛЕНИЯ -> Подсказки системы



The screenshot shows the Tantor 6 monitoring interface. On the left, there are two tabs: "Оповещения" (Notifications) and "Подсказки системы" (System Recommendations). The "Подсказки системы" tab is active, indicated by a red background. A tooltip box is overlaid on the interface, titled "Рекомендации по конфигурации" (Configuration recommendations). It contains a list of configuration items with their current values and a note that they require a restart. For example, it says "Требуется перезагрузка" (Requires restart) for "pg_stat_statements.max to 3000". The main dashboard shows various metrics like CPU load, memory usage, and transaction rates. The bottom right corner of the interface shows the Tantor logo and the number 90.

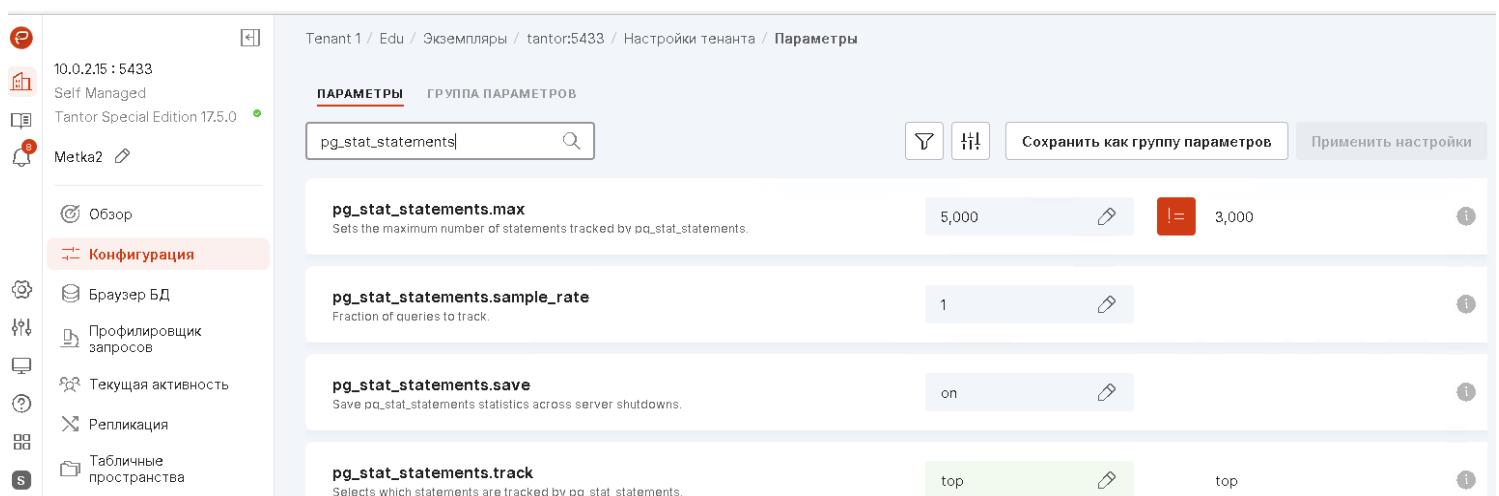
Страница Обзор -> УВЕДОМЛЕНИЯ -> Подсказки системы

На плитке УВЕДОМЛЕНИЯ две вкладки: "Оповещения" и "Подсказки системы". Активная вкладка отображается чёрным цветом, неактивная серым. Серый цвет путает: кажется, что вкладка некликабельна. Однако, кликнуть на вкладку серого цвета можно, содержимое плитки поменяется и высветятся рекомендации.

На вкладке "Подсказки системы" также есть текст серого цвета, но он уже прямо призывает: "Нажмите, чтобы увидеть необходимые изменения". При клике на текст, появится всплывающее окно "Рекомендации по конфигурации".

Рекомендации можно скопировать в буфер обмена или записать. Эти рекомендации **не** видны на странице "Параметры" в рекомендованных значениях, которая открывается через пункт меню "Конфигурация", не видны ни на странице "Оповещения", ни на "Журнал событий", то есть кроме всплывающего окна рекомендации больше **нигде не видны**.

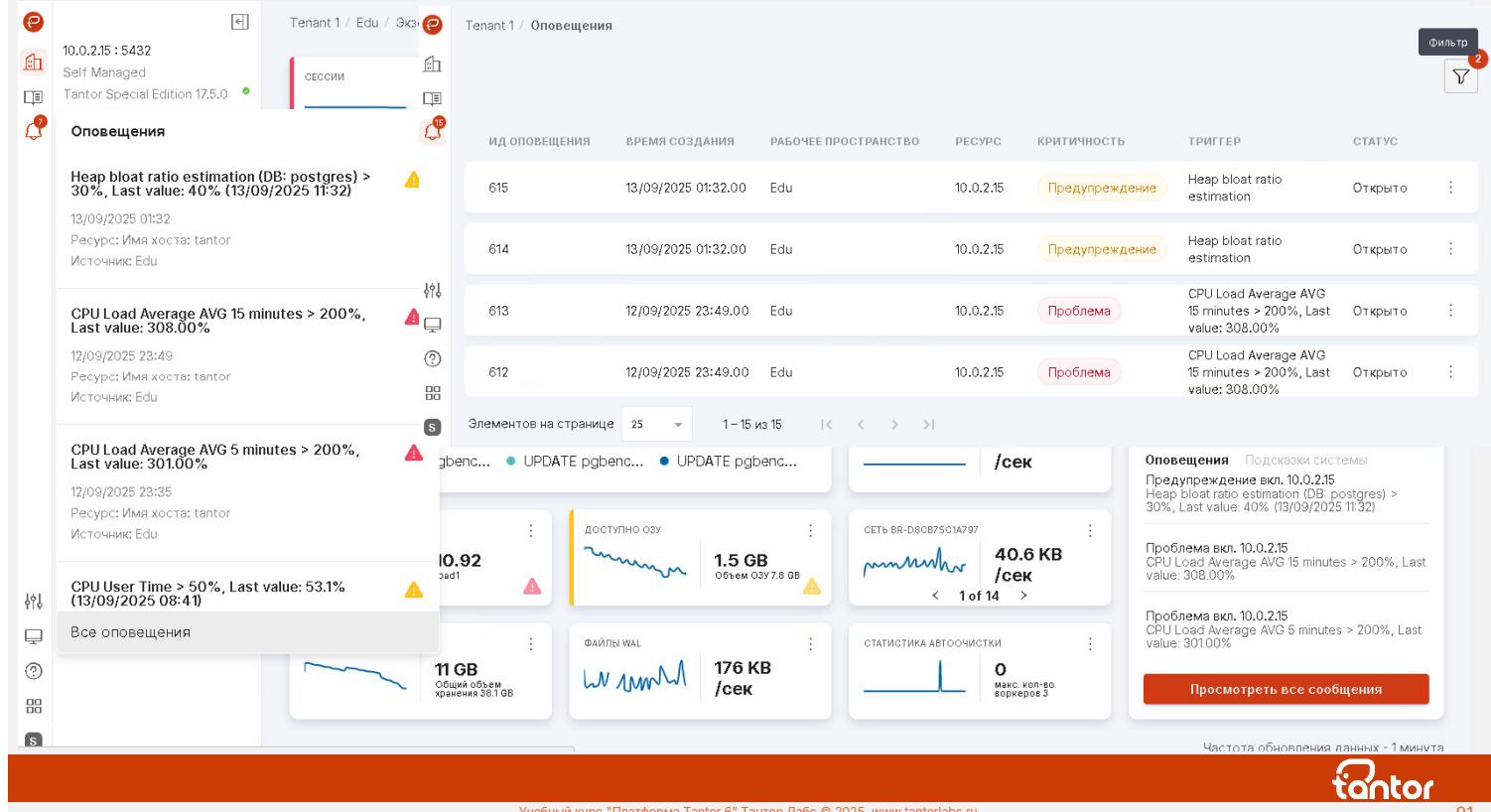
Поэтому, записав рекомендуемые значения, их можно ввести вручную на странице "Конфигурация":



The screenshot shows the "Настройки тенанта / Параметры" (Tenant settings / Parameters) page. The left sidebar has a "Конфигурация" tab selected. The main area shows a list of parameters with their current values and recommended changes. For "pg_stat_statements.max", the current value is 5,000 and the recommendation is 3,000. For "pg_stat_statements.sample_rate", the current value is 1 and the recommendation is 1. For "pg_stat_statements.save", the current value is "on" and the recommendation is "on". For "pg_stat_statements.track", the current value is "top" and the recommendation is "top". Each parameter has a "Сохранить как группу параметров" (Save as group) button and a "Применить настройки" (Apply settings) button.

Значение параметра `pg_stat_statements.max` из рекомендаций соответствует рекомендуемому конфигуратором Платформы. Значение `pg_stat_statements.track` в конфигураторе отсутствуют. Рекомендации конфигуратора консервативны.

Оповещения (сообщения, уведомления)



10.0.2.15 : 5432 Tenant 1 / Edu / Экз. Тенант 1 / Оповещения

Self Managed Tenant Special Edition 17.5.0

Оповещения

Heap bloat ratio estimation (DB: postgres) > 30%, Last value: 40% (13/09/2025 11:32)
13/09/2025 01:32
Ресурс: Имя хоста: tantor
Источник: Edu

CPU Load Average AVG 15 minutes > 200%, Last value: 308.00%
12/09/2025 23:49
Ресурс: Имя хоста: tantor
Источник: Edu

CPU Load Average AVG 5 minutes > 200%, Last value: 301.00%
12/09/2025 23:35
Ресурс: Имя хоста: tantor
Источник: Edu

CPU User Time > 50%, Last value: 53.1% (13/09/2025 08:41)
10.92
11 GB
Файлы WAL
176 KB /сек
Доступно озу
1.5 GB
Объем ОЗУ 7.8 GB
СЕТЬ BR-D8CB75C1A797
40.6 KB /сек
Статистика автоочистки
0 макс. кол-во воркеров 3

Элементов на странице: 25 1 - 15 из 15

ИД ОПОВЕЩЕНИЯ ВРЕМЯ СОЗДАНИЯ РАБОЧЕЕ ПРОСТРАНСТВО РЕСУРС КРИТИЧНОСТЬ ТРИГГЕР СТАТУС

ИД ОПОВЕЩЕНИЯ	ВРЕМЯ СОЗДАНИЯ	РАБОЧЕЕ ПРОСТРАНСТВО	РЕСУРС	КРИТИЧНОСТЬ	ТРИГГЕР	СТАТУС
615	13/09/2025 01:32:00	Edu	10.0.2.15	Предупреждение	Heap bloat ratio estimation	Открыто
614	13/09/2025 01:32:00	Edu	10.0.2.15	Предупреждение	Heap bloat ratio estimation	Открыто
613	12/09/2025 23:49:00	Edu	10.0.2.15	Проблема	CPU Load Average AVG 15 minutes > 200%, Last value: 308.00%	Открыто
612	12/09/2025 23:49:00	Edu	10.0.2.15	Проблема	CPU Load Average AVG 5 minutes > 200%, Last value: 308.00%	Открыто

Оповещения Подсказки системы
Предупреждение вкл. 10.0.2.15
Heap bloat ratio estimation (DB: postgres) > 30%, Last value: 40% (13/09/2025 11:32)

Проблема вкл. 10.0.2.15
CPU Load Average AVG 15 minutes > 200%, Last value: 308.00%

Проблема вкл. 10.0.2.15
CPU Load Average AVG 5 minutes > 200%, Last value: 301.00%

Просмотреть все сообщения

Частота обновления логов < 1 минута

tantor 91

Оповещения (сообщения, уведомления)

По нажатию на красную кнопку "Посмотреть все сообщения" откроется страница "Оповещения".

Также, на эту страницу можно перейти кликнув на значок колокольчика на левой панели меню и в открывшемся всплывающем окне кликнуть на текст "Все оповещения". Однако, при таком переходе может быть установлен фильтр на сообщения. Например, будут показываться сообщения со статусом = "открыто". На наличие фильтра укажет красный кружок рядом со значком "воронки" в правом верхнем углу страницы "Оповещения":

Число в кружке - это количество условий фильтрации.

При клике на значок воронки откроется всплывающее окно, в котором можно поменять условия фильтрации:

Фильтры

Рабочее пространство

Edu

Критичность

Открыто

Статус

Открыто

Закрыто - Обработано вручную

Закрыто - Автоматическое восстановление

Временной интервал

Открыто

Закрыто - Обработано вручную

Закрыто - Автоматическое восстановление

Очистить фильтры

Экземпляр: Сессии

Информация



Экземпляр: Сессии

При нажатии на три точки в верхней части плитки, появится меню с пунктом **"Выпадающее окно"**, нажав на который появится всплывающее окно (панель) с графиком большего размера, чем на плитке.

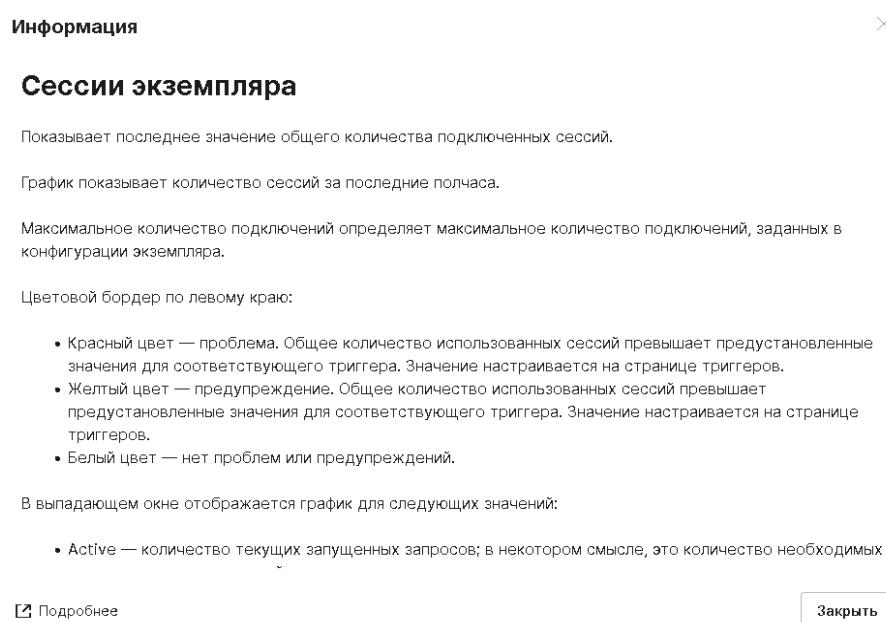
В окне Сессии показывается текущее число сессий, максимальное, которое определяется параметром конфигурации PostgreSQL `max_connections`. Изменение этого параметра требует рестарт экземпляра. Значение параметра на репликах должно быть не меньше, чем на мастере, иначе реплика приостановит применение WAL. Значение по умолчанию 100.

В примере на слайде показана работа экземпляра с запущенным тестом с 50 сессиями:

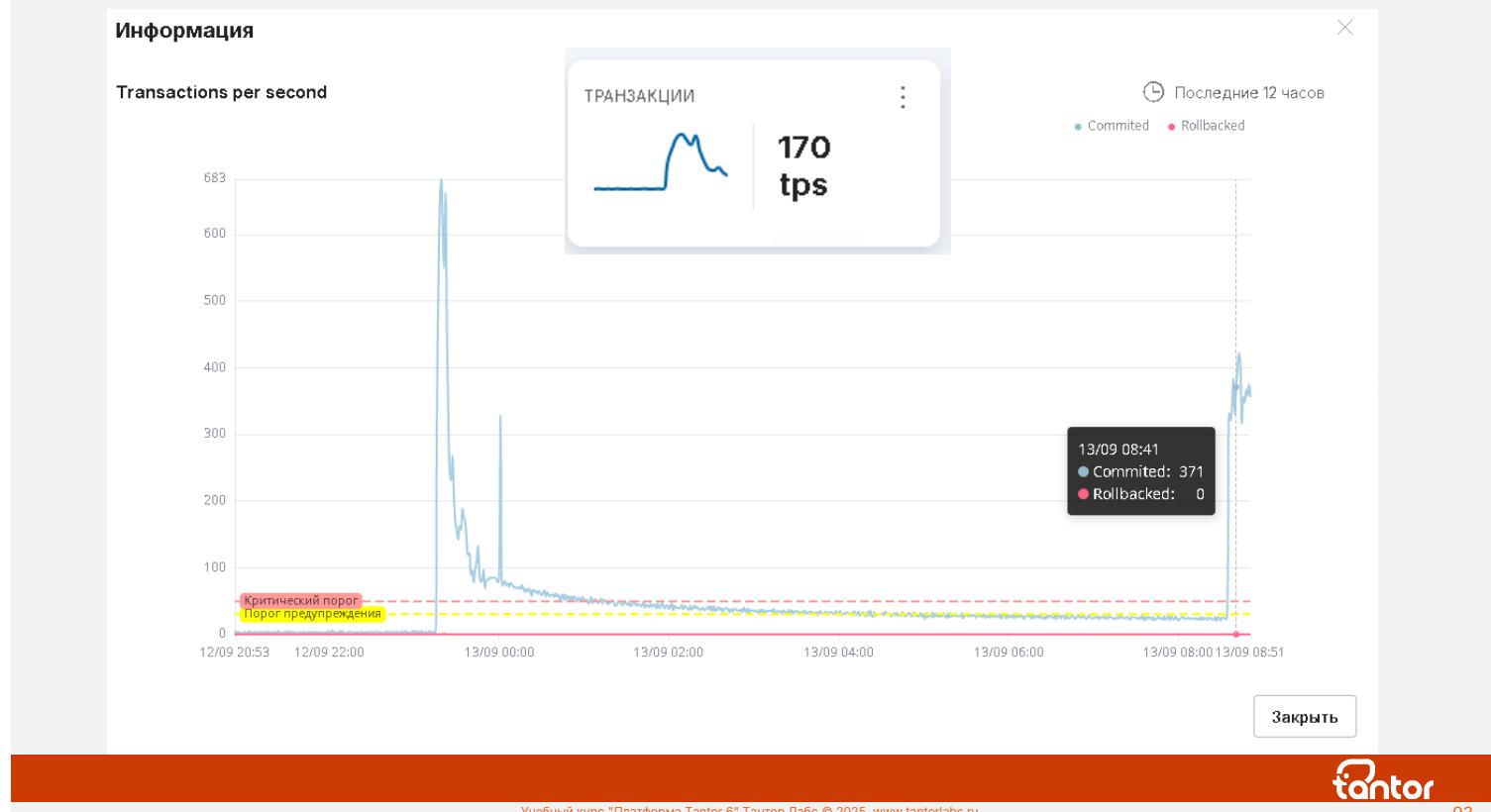
`pgbench -T 6000 -P 10 -c 50`

Цвета линий графика показаны в верхней части окна: активные сессии - синий (в среднем, 10), ожидающие (в среднем, 40) - красный. Если кликнуть на другие типы сессий, она появится на графике. В примере на слайде показаны простаивающие (idle) сессии и простаивающие в открытой транзакции (Idle in transaction).

При выборе в меню **"(i) Инфо"**, появится описание того, что отображается в окне:



Экземпляр: Транзакции



Экземпляр: Транзакции

В окне транзакции показываются зафиксированные и прерванные (отмененные, откаченные) транзакции.

В примере на слайде показывается график изменения TPS стандартно теста, запущенного командой: `pgbench -T 6000 -P 10 -c 50`

но с открытым моментальным снимком в сессии с той же базой данных:

```
begin transaction isolation level repeatable read;
select 1;
```

На графике видно, что сначала TPS около 670, потом довольно быстро снижались до небольших значений. После того как транзакция была завершена и моментальный снимок исчез, горизонт базы данных перестал удерживаться и TPS увеличился до значений около 400.

Желтая и красная линии - это предустановленные значения для триггера `xact_rollback` (Rolledback transactions) по числу транзакций, откатываемых в секунду:

Tenant 1 / Edu / Экземпляры / tantor:5432 / Настройки мониторинга / Триггеры

СПИСОК БАЗ ДАННЫХ

ТРИГГЕРЫ

tran

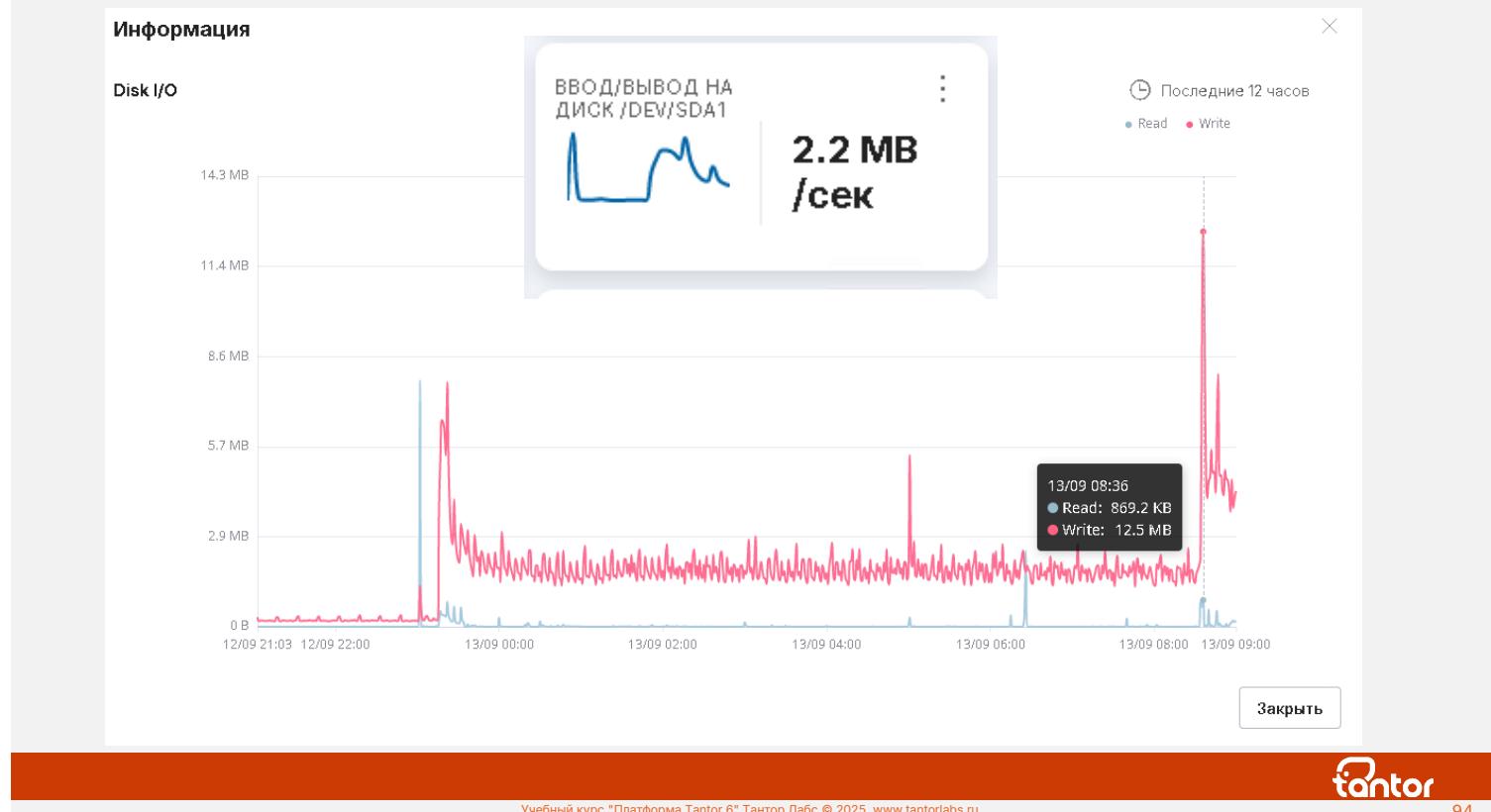


ТРИГГЕР	ОПИСАНИЕ ОПОВЕЩЕНИЯ	ТИП ТРИГГЕРА	ЗНАЧЕНИЕ ПРЕДУПРЕЖДЕНИЯ	ЗНАЧЕНИЕ ПРЕДУПРЕЖДЕНИЯ	ЗНАЧЕНИЕ ВОССТАНОВЛЕНИЯ
---------	---------------------	--------------	-------------------------	-------------------------	-------------------------

xact_rollback	Rollback transactions	host	30	50	20
---------------	-----------------------	------	----	----	----



Экземпляр: Ввод/Вывод на диск



Экземпляр: Ввод/Вывод на диск

Число показывает последнее значение скорости передачи данных между диском и оперативной памятью.

График показывает скорость передачи данных между диском и оперативной памятью за выбранный интервал. В отображается два графика для:

Read - синий цвет, скорость чтения с диска.

Write - красный цвет, скорость записи.

Экземпляр: Ввод/Вывод блоков



Экземпляр: Ввод/Вывод блоков

Показывает последнее значение скорости, с которой процессы экземпляра отправляют блоки (по умолчанию 8Кб) из буферного кэша экземпляра PostgreSQL в страницочный кэш linux или запрашивают чтение блоков файлов данных (доступ к которым идёт через буферный кэш).

В окне отображается график для следующих метрик:

Dirtied (голубой) - число блоков в кэше буферов, содержимое которых поменялось с последней завершенной контрольной точки или загрузки в буферный кэш.

Read (красный) - число чтений блоков из буферного кэша.

Write (синий) - число изменений блоков в буферном кэше.

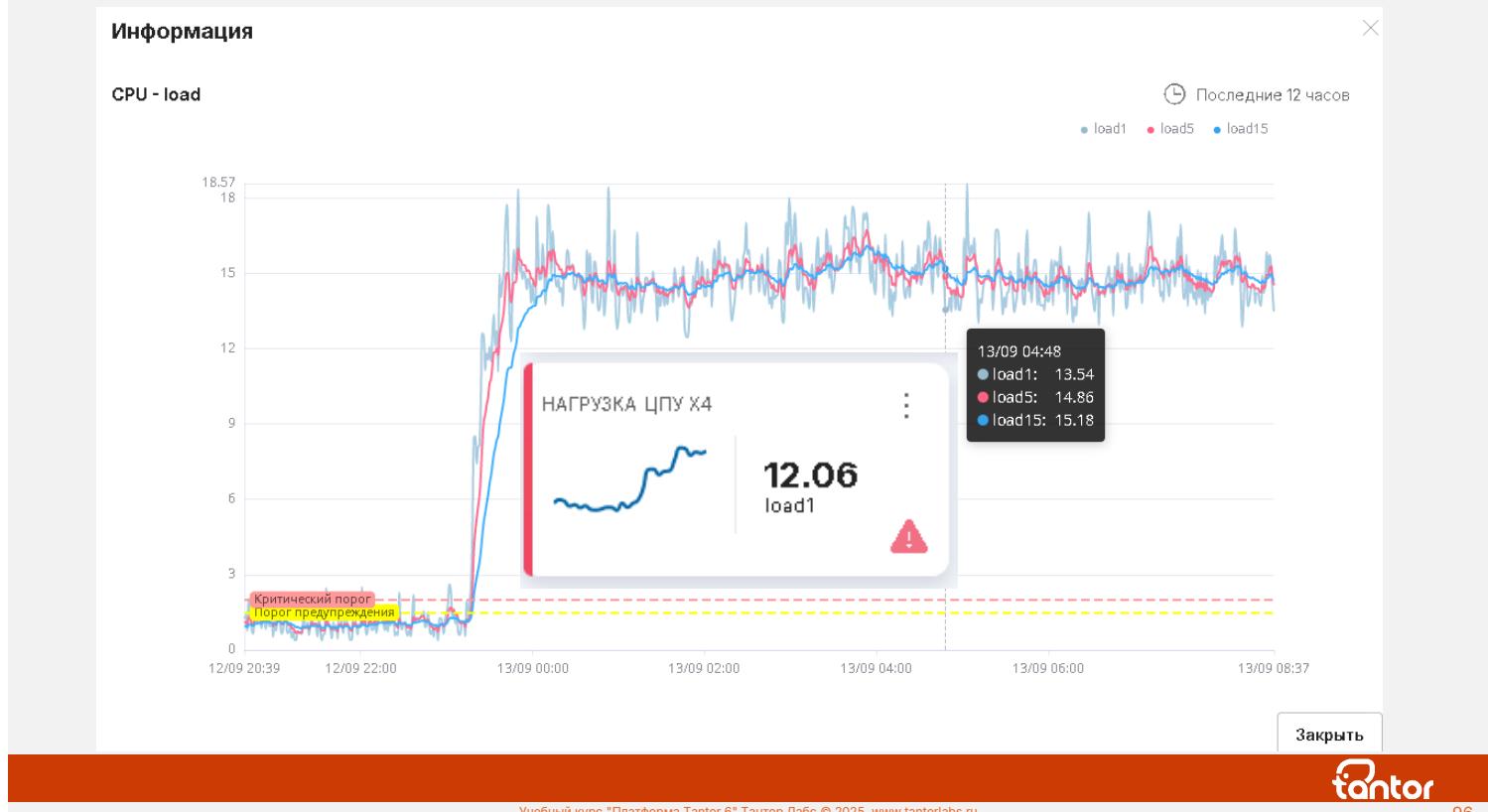
Temp Read (оранжевый) - число чтений блоков из локальных кэшей серверных процессов, которые используется для временных таблиц, индексов.

Temp Write (зелёный) - число изменений блоков в локальных кэшах серверных процессов, которые используются для временных таблиц, индексов.

На графике можно убрать или добавить отображение указанных метрик, масштаб графика при этом может измениться. В примере на слайде убран график красного цвета и зеленый график Dirtied проявил значения около 40. При отображении красного графика зеленый не был заметен. На красном графике значения около 16000 блоков в секунду и постепенно увеличиваются, при том, что TPS в то же время уменьшались. Серверным процессам приходилось читать всё больше блоков таблиц, чтобы добраться до актуальных версий.

После завершения транзакции, удерживавшей моментальный снимок, был всплеск записи и скорость чтения блоков уменьшилась до 20000 блоков в секунду.

Экземпляр: Нагрузка CPU



Экземпляр: Нагрузка CPU

НАГРУЗКА ЦПУ X4. после "X" в названии плашки число ядер, например, «НАГРУЗКА ЦПУ X4» - **четыре** ядра.

Показывает популярную (среди администраторов linux) метрику "CPU load".

На плашке отображается метрика **load1** - нагрузка за последнюю минуту.

В окне графики для трёх метрик:

load1 - загрузка за интервал в 1 минуту;

load5 - загрузка за интервал в 5 минут;

load15 - загрузка за интервал в 15 минут.

Цвет линии на левом крае плашки:

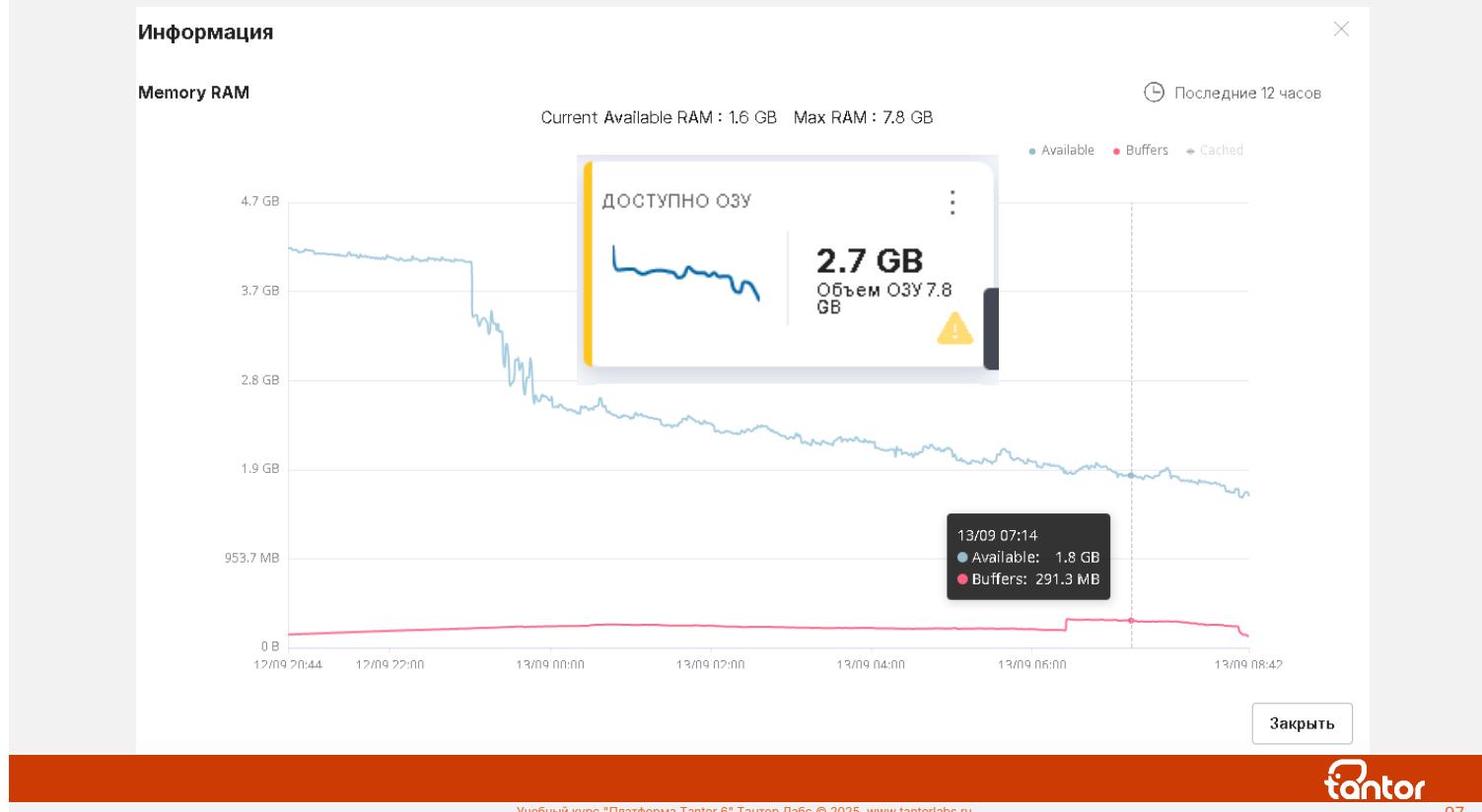
Красный - проблема. Загрузка процессора превысила пороговое значение. Значение настраивается на странице триггеров.

Желтый - предупреждение:

СПИСОК БАЗ ДАННЫХ ТРИГГЕРЫ

ТРИГГЕР	ОПИСАНИЕ ОПОВЕЩЕНИЯ	ТИП ТРИГГЕРА	ЗНАЧЕНИЕ ПРЕДУПРЕЖДЕНИЯ	ЗНАЧЕНИЕ ПРОБЛЕМЫ	ЗНАЧЕНИЕ ВОССТАНОВЛЕНИЯ
load1	Loadaverage	host	1.5	2	0.7
load5	Loadaverage	host	1	2	0.7
load15	Loadaverage	host	1	2	0.7

Экземпляр: Доступно ОЗУ



Экземпляр: Доступно ОЗУ

Плашка показывает текущее значение объема доступной оперативной памяти. На плашке показывается "Объем ОЗУ" (11.5Гб) - общий объем памяти в linux.

На графике отображаются:

Available - оценка объема памяти, доступной для приложений без своппинга;

Buffers - память, используемая буферами ядра linux;

Cached - память, используемая файловым кэшем linux и SLAB.

Цвет линии на левом крае плашки:

Красный - проблема.

Желтый - предупреждение:

СПИСОК БАЗ ДАННЫХ		ТРИГГЕРЫ				
mem						
ТРИГГЕР	ОПИСАНИЕ ОПОВЕЩЕНИЯ	ТИП ТРИГГЕРА	ЗНАЧЕНИЕ ПРЕДУПРЕЖДЕНИЯ	ЗНАЧЕНИЕ ПРОБЛЕМЫ	ЗНАЧЕНИЕ ВОССТАНОВЛЕНИЯ	
available_bytes_pct	Memory available	host	40	20	50	...

Пороговые значения 40/20/50 можно поменять на 30/10/40 или 50/30/60:

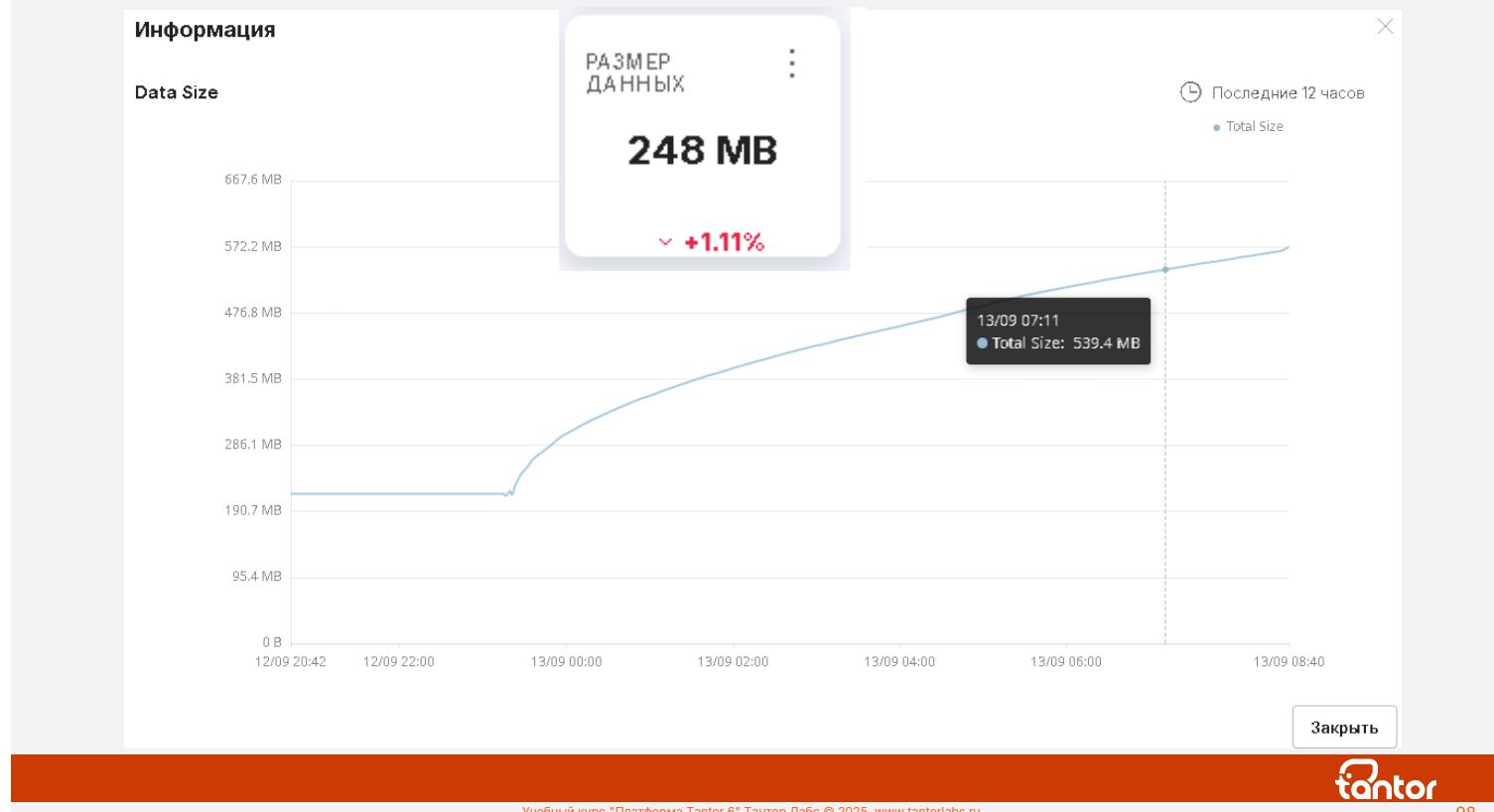
Редактирование триггера

Вы меняете триггер: **Memory available**

ПРЕДУПРЕЖДЕНИЕ	ПРОБЛЕМА	ВОССТАНОВЛЕНИЕ	
<input type="radio"/> 30	10	40	
<input checked="" type="radio"/> 40	20	50	(Default)
<input type="radio"/> 50	30	60	

Отменить Сохранить

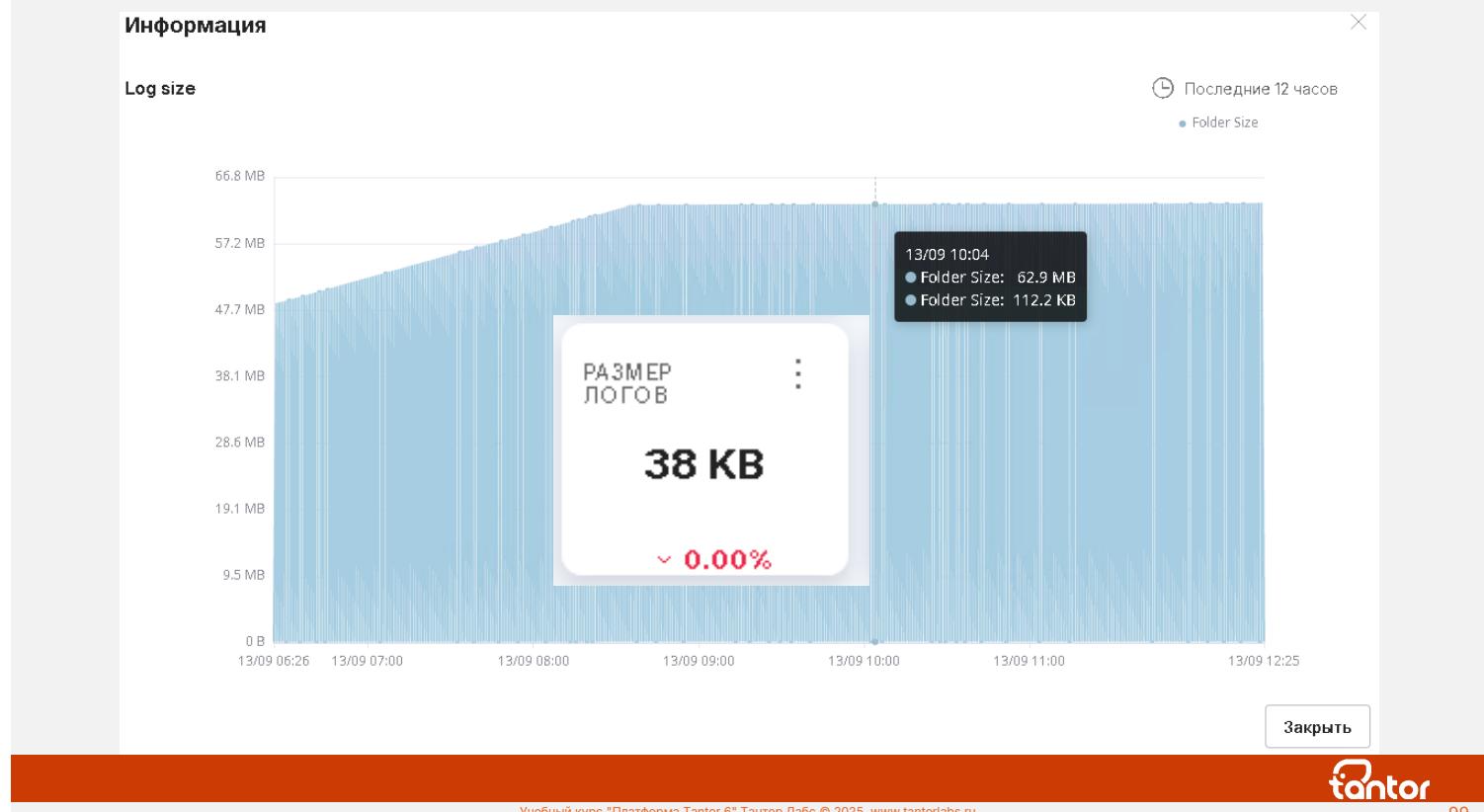
Экземпляр: Размер данных



Экземпляр: Размер данных

Плашка показывает сумму размеров всех баз данных кластера PostgreSQL. В размер не входят WAL-файлы и другие служебные файлы.

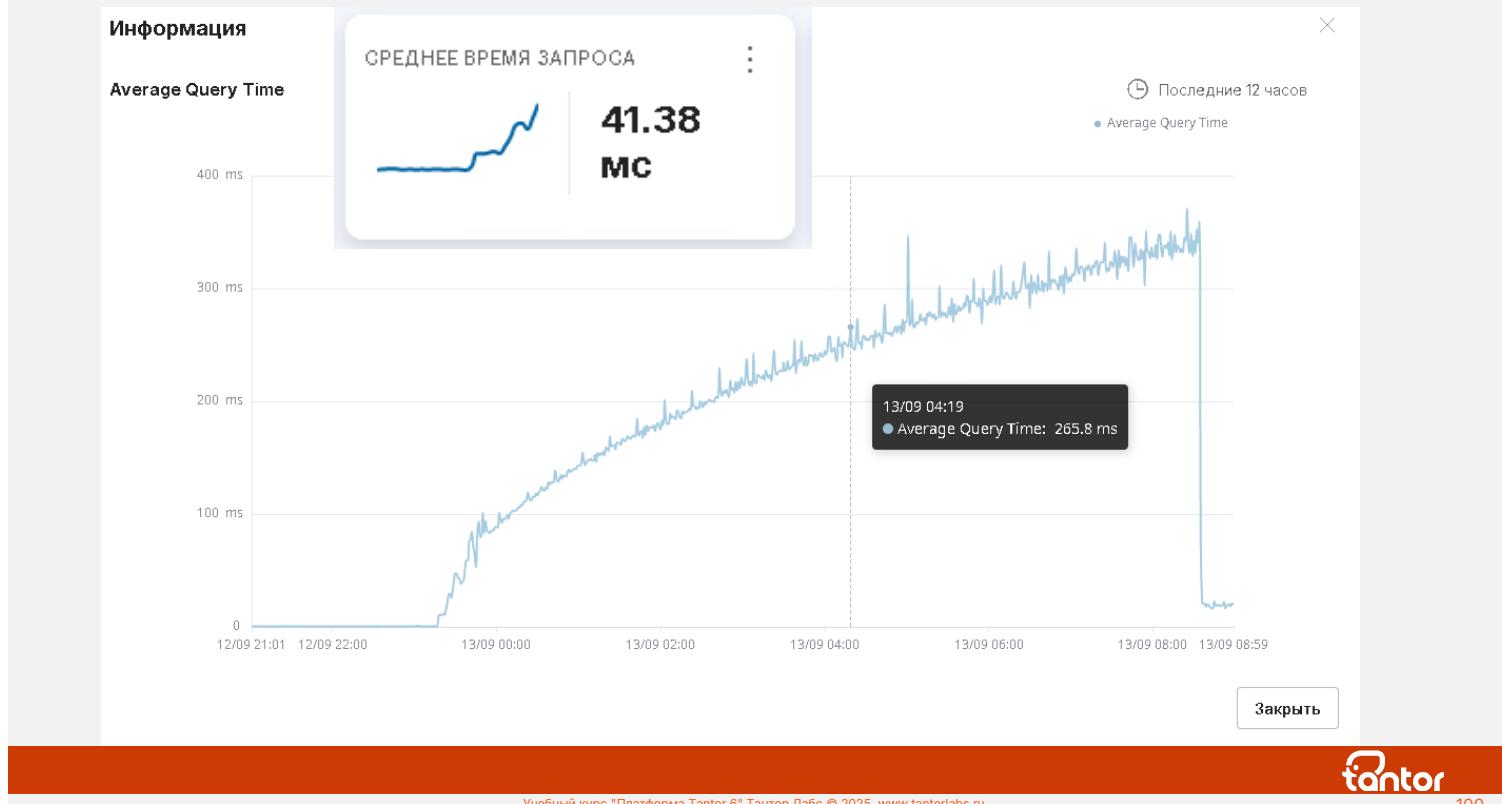
Экземпляр: Размер диагностических логов



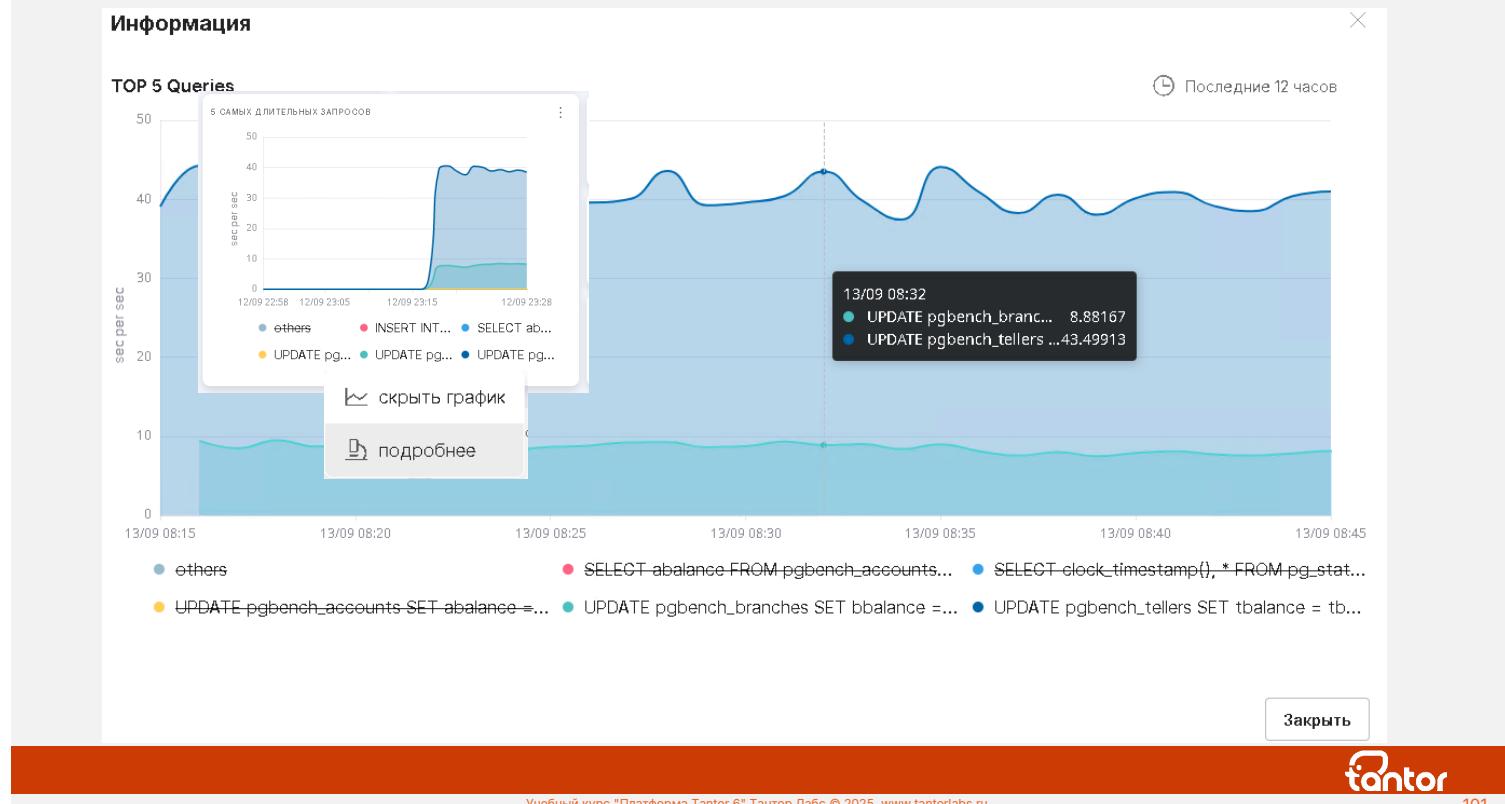
Экземпляр: Размер диагностических логов

Плашка показывает размер файлов диагностических логов кластера PostgreSQL. Это **не** размер директории pg_wal с WAL-файлами.

Экземпляр: Среднее время запроса



Экземпляр: 5 самых длительных запросов

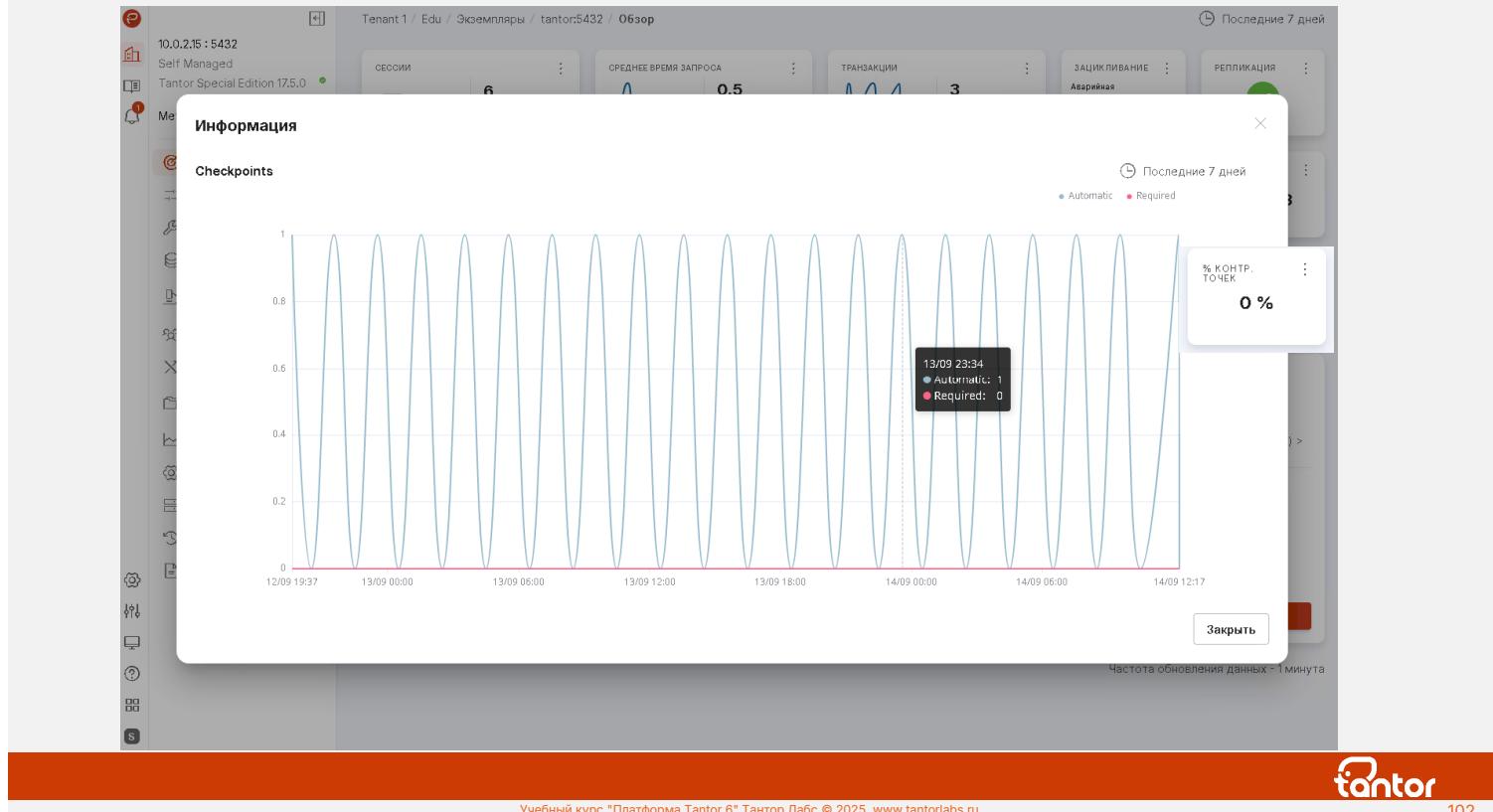


Экземпляр: 5 самых длительных запросов

"TOP 5 Queries" - среднее время выполнения пяти самых долгих запросов и "others" (остальных запросов).

При нажатии на плашке на запрос, появляется всплывающее меню с пунктами: "скрыть график" (скрывает линию запроса на графике), "подробнее" - открывает страницу профилирования этого запроса.

Экземпляр: % КОНТР. ТОЧЕК



Экземпляр: % внеочередных контрольных точек

% КОНТР. ТОЧЕК - % внеочередных контрольных точек. На плашке показывается процент контрольных точек, который был выполнен **НЕ** по времени, а "по требованию" (required). Такое происходит, если экземпляр стал генерировать большой объем журнальных данных так, что если выполнять контрольную точку по времени, то был бы превышен объем, заданный параметром `max_wal_size`. Чтобы не допустить этого, контрольная точка по времени, а она в 90% вероятности выполняется (если параметр `checkpoint_completion_target = 0.9`), завершается без задержек (с максимальной скоростью и нагрузкой на I/O) и начинается контрольная точка по требованию.

Если такое происходит нечасто и вызвано загрузками или реорганизацией данных, то это нормально. Если процент внеочередных (не по времени) контрольных точек всегда больше нуля, то стоит увеличить значение `max_wal_size`. Значение `checkpoint_timeout` уменьшать не стоит, оно выбирается из других принципов.

На графике отображается:

Automatic - контрольная точка, созданная автоматически.

Required - созданная суперпользователем вручную.

Если вынужденные (Required) контрольные точки составляют более 25% в час, по умолчанию будет создано предупреждение; если более 50% - сообщение о проблеме.

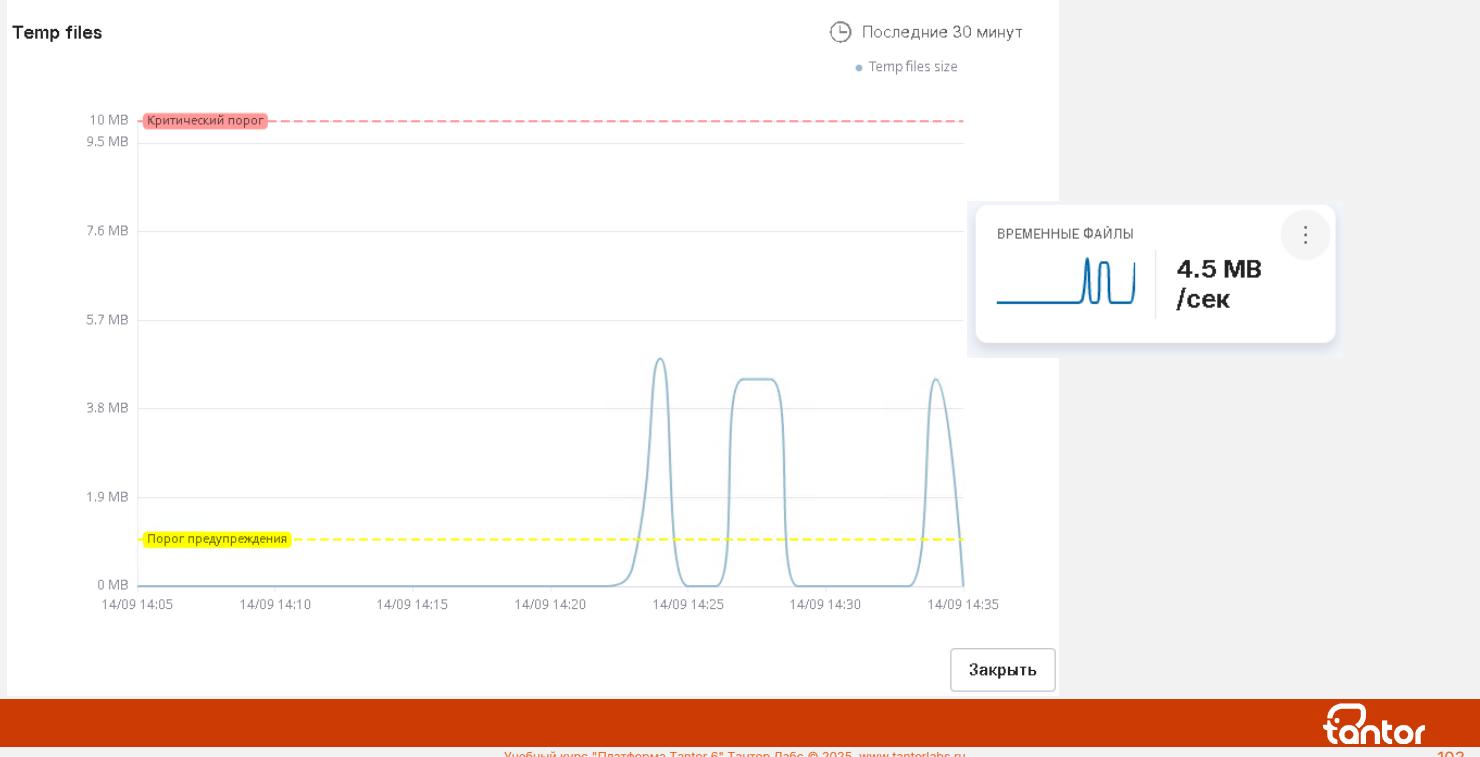
Пороговые значения выбраны эмпирически, чтобы уменьшить вероятность возникновения предупреждений при наличии массовых загрузок или типичных задач по реорганизации данных.

Оповещение автоматически закроется, когда процент внеочередных контрольных точек станет меньше 10%:

СПИСОК БАЗ ДАННЫХ		ТРИГГЕРЫ			
check					
ТРИГГЕР	ОПИСАНИЕ ОПОВЕЩЕНИЯ	ТИП ТРИГГЕРА	ЗНАЧЕНИЕ ПРЕДУПРЕЖДЕНИЯ	ЗНАЧЕНИЕ ПРОБЛЕМЫ	ЗНАЧЕНИЕ ВОССТАНОВЛЕНИЯ
checkpoint_ratio	Checkpoints ratio	postgres	75	50	90

Экземпляр: Временные файлы

Информация



Экземпляр: Временные файлы

временные файлы – объем, записанный на диск при выполнении запросов, включая работу как с временными таблицами, так и сохранение данных при нехватке памяти (параметры для каждого процесса, а не для экземпляра: `work_mem`, `hash_mem_multiplier`, `maintenance_work_mem`, `temp_buffers`). Ограничение на размер временных файлов для каждого процесса задаётся параметром `temp_file_limit`. Временные файлы создаются в табличных пространствах, названия которых можно указать в параметре `temp_tablespaces`, минимальный размер временных файлов, создание которых будет логироваться в диагностический журнал, задаётся параметром `log_temp_files`.

Плашка показывает скорость увеличения размера временных файлов.

График также показывает **скорость увеличения размера** временных файлов (`Temp files size`), а не общий размер временных файлов.

Триггер также установлен на **скорость увеличения размера** временных файлов:

Редактирование триггера

Вы меняете триггер: `Temporary files size (delta)`

ПРЕДУПРЕЖДЕНИЕ

ПРОБЛЕМА

ВОССТАНОВЛЕНИЕ



0.5MB

5MB

0.3MB



1MB

10MB

0.8MB

(Default)



10MB

50MB

5MB

Отменить

Сохранить

Пороговые значения можно поменять с 1/10/0.8 на: 0.5/5/0.3 или 10/50/5.

Экземпляр: Временные файлы (событии)

КРИТИЧНОСТЬ | ТРИГГЕР | СТАТУС | ДЛИТЕЛЬНОСТЬ

WARNING	Temporary files size (delta) (dbname: postgres) > 1МБ (1048580 bytes), Last value: 4МБ (4666667 bytes)	Open	00:02:35						
Рабочее пространство: Edu	Создано: 14/09/2025 14:41:00	ВРЕМЕННЫЕ ФАЙЛЫ	4.5 МВ/сек						
Ресурс: PG 17.5.0: tantor:5432:postgres	Метрика: temp_bytes	УВЕДОМЛЕНИЯ							
Источник: DATABASE:postgres	Значение: 4666667	Оповещения Подсказки системы							
ОБЗОР СВЯЗАННЫЕ ОПОВЕЩЕНИЯ ЖУРНАЛ СОБЫТИЙ		Предупреждение вкл. 10.0.2.15 Temporary files size (delta) (dbname: postgres) > 1МБ (1048580 bytes), Last value: 4МБ (4666667 bytes)							
КРИТИЧНОСТЬ ТРИГГЕР СТАТУС ДЛИТЕЛЬНОСТЬ	Temporary files size (delta) (dbname: postgres) > 1МБ (1048580 bytes), Last value: 4МБ (4666667 bytes) Closed - Auto Recovered 00:07:00								
СОЗДАНО В ТИП СОБЫТИЯ ДАННЫЕ СОБЫТИЯ	<table border="1"><tr><td>14/09/2025 14:41:00</td><td>CREATE ALERT</td><td>Temporary files size (delta) (dbname: postgres) > 1МБ (1048580 bytes), Last value: 4МБ (4666667 bytes)</td></tr><tr><td>14/09/2025 14:48:00</td><td>CLOSE ALERT</td><td>Temporary files size (delta) (dbname: postgres) > 1МБ (1048580 bytes), Last value: 4МБ (4666667 bytes)</td></tr></table>			14/09/2025 14:41:00	CREATE ALERT	Temporary files size (delta) (dbname: postgres) > 1МБ (1048580 bytes), Last value: 4МБ (4666667 bytes)	14/09/2025 14:48:00	CLOSE ALERT	Temporary files size (delta) (dbname: postgres) > 1МБ (1048580 bytes), Last value: 4МБ (4666667 bytes)
14/09/2025 14:41:00	CREATE ALERT	Temporary files size (delta) (dbname: postgres) > 1МБ (1048580 bytes), Last value: 4МБ (4666667 bytes)							
14/09/2025 14:48:00	CLOSE ALERT	Temporary files size (delta) (dbname: postgres) > 1МБ (1048580 bytes), Last value: 4МБ (4666667 bytes)							

Создано: 14/09/2025 14:41:00 | Тип события: CREATE ALERT | Данные события: Temporary files size (delta) (dbname: postgres) > 1МБ (1048580 bytes), Last value: 4МБ (4666667 bytes)

Создано: 14/09/2025 14:48:00 | Тип события: CLOSE ALERT | Данные события: Temporary files size (delta) (dbname: postgres) > 1МБ (1048580 bytes), Last value: 4МБ (4666667 bytes)

Экземпляр: Временные файлы

Пример команд для тестирования создания и увеличения размера временных файлов:

```
create temp table temp1 (id integer);
insert into temp1 select * from generate_series(1, 20000000);
select pg_table_size('temp1');
pg_table_size
-----
725155840
```

Всплески на графике - вставки по 20млн. строк во временную таблицу. Размер временной таблицы и её временных файлов будет расти. После первой вставки оповещение не создавалось, так как вставки проходили в локальный кэш буферов процесса.

В примере на слайде превышение скорости увеличения размера временных файлов таблицы длилось 2 минуты. Через 5 минут после окончания (через 7 минут после начала возникновения оповещения) увеличения размера временных файлов оповещение было автоматически закрыто.

Интервал времени восстановления - 5 минут.

Экземпляр: Сеть

Информация

Network traffic

СЕТЬ BR-D8CB75C1A797

15.5 kB
/сек

< 1 of 14 >

1.8 kB
1.5 kB
1.2 kB
900 B
600 B
300 B
0 B

14/09 01:03 14/09 01:05 14/09 01:10 14/09 01:15 14/09 01:20 14/09 01:25 14/09 01:31

Последние 30 минут

eth0
br-d8cb75c1a797
docker0
eth0
lo
veth1aaa7af
veth4bebfd0
veth5c9de7c

Закрыть

tantor

105

Учебный курс "Платформа Tantor 6" Тантор Лабс © 2025 www.tantolabs.ru

Экземпляр: Сеть

График показывает сетевой трафик выбранного сетевого интерфейса. Выбрать интерфейс можно с помощью угловых стрелок на плашке, а в окне графика - в правом верхнем углу.

В окне выдаются графики **Received (голубой цвет линии)** и **Sent (красный цвет линии)** в kB (или более крупных единиц измерения) в секунду.

Кроме рассмотренных в этой главе, ещё имеются плашки и графики:

% блоков из буфера - процент "попадания в буферный кэш экземпляра" (cache hit). У метрики `hit_read_ratio` есть пороговые значения: 90/80/95 и они не меняются.

зацикливание (wraparound). Прогресс-бар показывает число использованных номеров транзакций в текущем цикле. Маркер на индикаторе покажет ожидаемое начало аварийного автovакуума. Для СУБД с 32-битным счетчиком (ванильный PostgreSQL, Tantor BE, Tantor Free) число транзакций в цикле небольшое, около 4млрд. Для СУБД с 64-битным счетчиком (Tantor Postgres SE и SE1C), зацикливание не актуально. У этой плашки нет графика.

репликация - графика нет, на плашке может отражаться зелёный кружок - репликация работает без сбоев, красный треугольник - проблема с репликацией, серый кружок - реплик нет.

статистика автovакуума - число активных рабочих процессов автovакуума (autovacuum workers). Число процессов рассчитывает процесс autovacuum launcher, но это число ограничено значением параметра `autovacuum_max_workers`.

Практика 6

1. Откройте страницу "Обзор"/"Overview" экземпляра
2. Посмотрите число сетевых интерфейсов
3. Посмотрите выпадающее окно ДОСТУПНО ОЗУ
4. Поменяйте интервал отображения с 30 минут на 7 дней

Практика 5

1. Откройте страницу "Обзор"/"Overview" экземпляра
2. Посмотрите число сетевых интерфейсов
3. Посмотрите выпадающее окно ДОСТУПНО ОЗУ
4. Поменяйте интервал отображения с 30 минут на 7 дней



7

Экземпляры PostgreSQL

Оповещения

Оповещения

Tenant 1 / Рабочие пространства

РАБОЧИЕ ПРОСТРАНСТВА ЭКЗЕМПЛЯРЫ

Помощь имени

Оповещения

Оповещения

Disk Space Usage (mountpoint: /) > 80%, Last value: 90% ⚠

14/09/2025 13:48
Ресурс: Имя хоста: tantor
Источник: Edu

Heap bloat ratio estimation (DB: postgres) > 30%, Last value: 40% (13/09/2025 11:32) ⚠

13/09/2025 01:22
Ресурс: Имя хоста: tantor
Источник: Edu

Heap bloat ratio estimation (DB: postgres) > 30%, Last value: 40% (13/09/2025 11:32) ⚠

13/09/2025 01:22
Ресурс: Имя хоста: tantor
Источник: Edu

Все оповещения

ОПОВЕЩЕНИЯ | НЕВЫПОЛНЕННЫЕ

3

ХРАНИЛИЩЕ ROOT

Нажмите, чтобы показать предупреждение ⚠ 3 объем хранения 58.1GB ⚠

УВЕДОМЛЕНИЯ

Оповещения Подсказки системы

Предупреждение вкл. 10.0.2.15 Disk Space Usage (mountpoint: /) > 80%, Last value: 90%

Предупреждение вкл. 10.0.2.15 Heap bloat ratio estimation (DB: postgres) > 30%, Last value: 40% (13/09/2025 11:32)

Просмотреть все сообщения

Оповещения

Для уведомления о том, что происходит в Платформе используется Журнал событий.

Для уведомления о том, что происходит в работе экземпляров PostgreSQL или хостов используются Оповещения (Alerts).

Оповещения связаны с тенантом. После выбора тенанта в левом вертикальном меню появляется значок колокольчика, при клике на который открывается всплывающее окно с открытыми оповещениями, внизу страницы ссылка "Все оповещения", при клике на которую откроется страница "Оповещения" со списком оповещений текущего пространства.

Если есть не закрытые оповещения, то на колокольчике в красном кружке отображается их число.

Колокольчик с числом оповещений всегда остается на том же месте, пока вы находитесь в тенанте, какую бы страницу Платформы вы не открыли, благодаря этому, вы не упустите появление новых оповещений.

Оповещения нужны для быстрого уведомления о проблемах и предупреждениях.

Страница со списком оповещений

ID ОПОВЕЩЕНИЯ	ВРЕМЯ СОЗДАНИЯ	РАБОЧЕЕ ПРОСТРАНСТВО	РЕСУРС	КРИТИЧНОСТЬ	ТРИГГЕР	СТАТУС	⋮
652	14/09/2025 13:48:00	Edu	10.0.2.15	Предупреждение	Disk Space Usage	Открыто	⋮
614	13/09/2025 01:32:00	Edu	10.0.2.15	Предупреждение	Heap bloat ratio	Открыто	⋮

Открыто ✓

Закрыто - Обработано вручную

Закрыто - Автоматическое восстановление

Открыто

Закрыто - Обработано вручную

Закрыто - Автоматическое

--

Предупреждение

Проблема

Не определено

OK

Фильтры

Рабочее пространство: Edu

Критичность:

Статус: Открыто

Временной интервал:

Очистить фильтры

Учебный курс "Платформа Tantor 6" Тантор Лабс © 2025 www.tantolabs.ru

tantor

109

Страница со списком оповещений

Попасть на страницу Оповещения можно с других страниц Платформы, а не только кликнув на значок колокольчика. Кликнув на цифру в столбце ОПОВЕЩЕНИЯ на странице тенанта; нажав на красную кнопку "Просмотреть все сообщения" на странице Обзор экземпляра. В этой плашке показываются только оповещения, относящиеся к выбранному экземпляру. При открытии страницы со списком оповещений может быть установлен фильтр на оповещения, но его можно убрать.

Также можно переходить на страницу с конкретным оповещением кликнув на полоску или красный или треугольный значок в плашках на странице "Обзор" экземпляра.

В примере на слайде страница "Оповещения" была открыта через значок колокольчика, поэтому был выставлен фильтр по 2 параметрам (красный кружок у значка воронки справа вверху страницы): по пространству и статусу "открыто". Нажав на значок воронки можно поменять условия фильтрации сообщений.

Критичность может быть: Предупреждение, проблема, Не определено, OK.

Статусов может быть пять: один Открыто и Закрыто одним из четырёх способов.

Кроме фильтрации можно отсортировать оповещения, кликнув на название столбца таблицы со списком оповещений.

Нажав на три точки, появится меню с пунктами "Подробнее" и "Закрыть".

Страница оповещения: ОБЗОР

Tenant 1 / Оповещения / ИД оповещения 652

Закрыть

ОБЗОР СВЯЗАННЫЕ ОПОВЕЩЕНИЯ ЖУРНАЛ СОБЫТИЙ

КРИТИЧНОСТЬ ТРИГГЕР СТАТУС ДЛИТЕЛЬНОСТЬ

WARNING Disk Space Usage (mountpoint: /) > 80%, Last value: 90% Open 07:44:16

Рабочее пр... Edu Создано : 14/09/2025 13:48:00

Ресурс : HOST: tantor(mounted on:/) Метрика : use_in_percent

Источник : MOUNTPOINT:root Значение : 90

Закрытие оповещения

Решение (необязательно)

Отменить Закрыть оповещение

Страница оповещения: ОБЗОР

При нажатии в списке оповещений а строку или выборе в меню "Подробнее" или при клике на треугольный значок на странице Обзор экземпляра откроется страница с описанием конкретного оповещения.

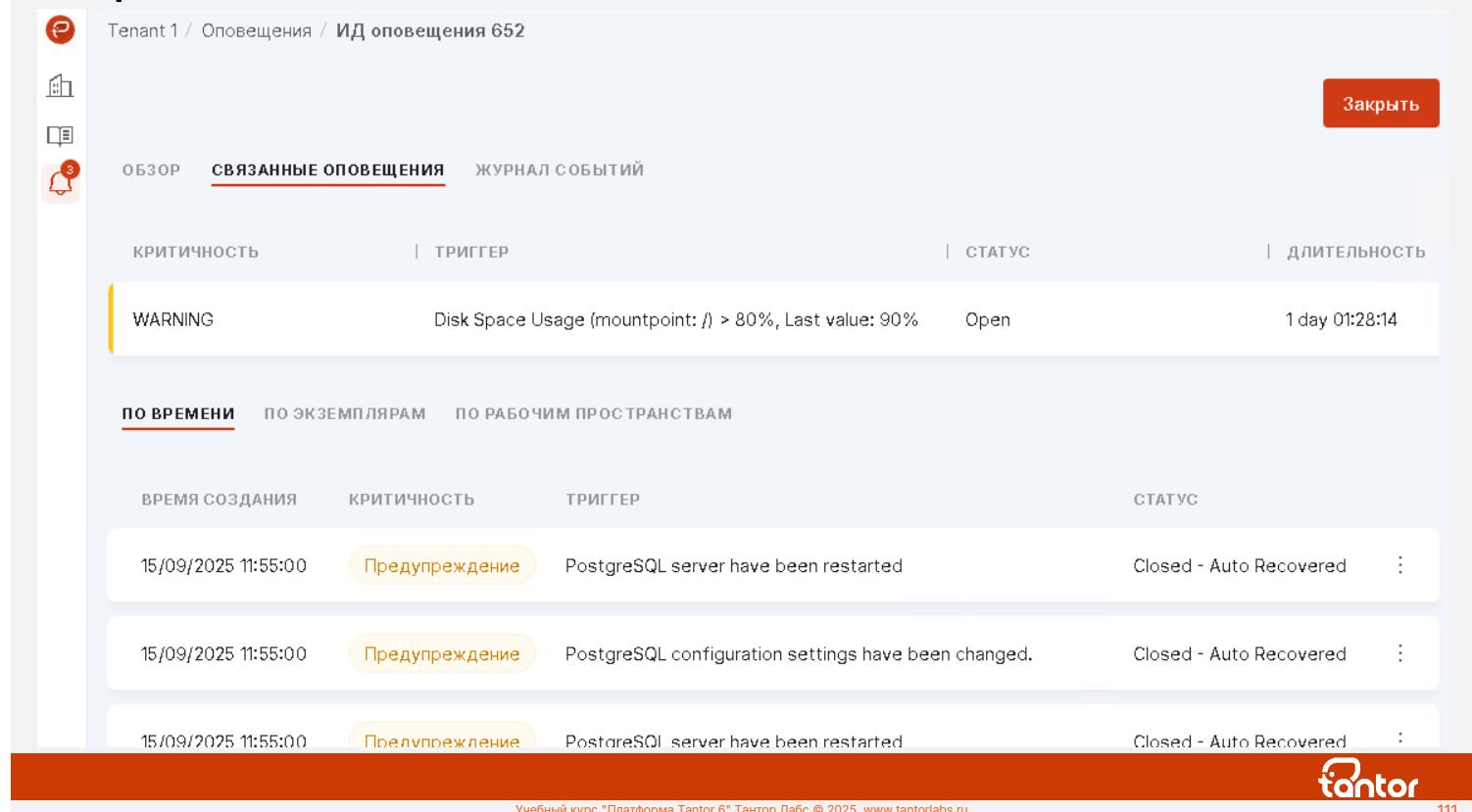
Если выбрать "Закрыть", то появится всплывающее окно, где дополнительно (необязательно) можно написать текст о причине закрытия или закрыть без комментария.

На странице три вкладки: ОБЗОР, СВЯЗАННЫЕ ОПОВЕЩЕНИЯ, ЖУРНАЛ СОБЫТИЙ.

В Платформе есть возможность увидеть список всех типов событий, которые могут вызвать оповещения. Для этого можно перейти в пункт меню "Настройка Мониторинга" (Monitoring Config) и перейти во вкладку "Триггеры". Пороговые значения некоторых их триггеров можно редактировать. Обычно, предлагается три комбинации пороговых значений, чтобы регулировать (повышать или снижать) чувствительность к событию.

Агент постоянно собирает метрики с наблюдаемой базы данных и отправляет их в Платформу, где они обрабатываются. Оповещения срабатывают, если метрики по тем или иным параметрам превышают пороговые значения. Превышение этих значений выявляет триггер.

Страница оповещения: СВЯЗАННЫЕ ОПОВЕЩЕНИЯ



Tenant 1 / Оповещения / ИД оповещения 652

Закрыть

ОБЗОР СВЯЗАННЫЕ ОПОВЕЩЕНИЯ ЖУРНАЛ СОБЫТИЙ

КРИТИЧНОСТЬ | ТРИГГЕР | СТАТУС | ДЛИТЕЛЬНОСТЬ

КРИТИЧНОСТЬ	ТРИГГЕР	СТАТУС	ДЛИТЕЛЬНОСТЬ
WARNING	Disk Space Usage (mountpoint: /) > 80%, Last value: 90%	Open	1 day 01:28:14

ПО ВРЕМЕНИ ПО ЭКЗЕМПЛЯРАМ ПО РАБОЧИМ ПРОСТРАНСТВАМ

ВРЕМЯ СОЗДАНИЯ	КРИТИЧНОСТЬ	ТРИГГЕР	СТАТУС
15/09/2025 11:55:00	Предупреждение	PostgreSQL server have been restarted	Closed - Auto Recovered
15/09/2025 11:55:00	Предупреждение	PostgreSQL configuration settings have been changed.	Closed - Auto Recovered
15/09/2025 11:55:00	Предупреждение	PostgreSQL server have been restarted	Closed - Auto Recovered

tantor 111

Страница оповещения: СВЯЗАННЫЕ ОПОВЕЩЕНИЯ

На вкладке СВЯЗАННЫЕ ОПОВЕЩЕНИЯ после описания оповещения есть три вкладки ПО ВРЕМЕНИ, ПО ЭКЗЕМПЛЯРАМ, ПО ПРОСТРАНСТВАМ.

По этим характеристикам высвечиваются оповещения. Однако, для долгоживущих оповещений (на слайде 1 день 1:28:14) выдается много оповещений, логически не связанных с исходным. Избежать этого нельзя, так как есть опасность отфильтровать связанные оповещения, поэтому выдаются все оповещения: за интервал пока оповещение было открыто, возникших на том же экземпляре или экземплярах пространства.

Страница оповещения: ЖУРНАЛ СОБЫТИЙ

Tenant 1 / Оповещения / Ид оповещения 652

Закрыть

ОБЗОР СВЯЗАННЫЕ ОПОВЕЩЕНИЯ ЖУРНАЛ СОБЫТИЙ

КРИТИЧНОСТЬ ТРИГГЕР СТАТУС ДЛЯ СОБЫТИЙ

WARNING Disk Space Usage (mountpoint: /) > 80%, Last value: 9... Closed - Manually Acknowledged 1 day 01:43:47

СОЗДАНО В ТИП СОБЫТИЯ ДАННЫЕ СОБЫТИЯ

14/09/2025 13:48:00 CREATE ALERT Disk Space Usage (mountpoint: /) > 80%, Last value: 90%

15/09/2025 15:31:47 CLOSE ALERT Resolved by student

Рабочее прост... Edu Создано : 14/09/2025 13:48:00 Разрешено в : 15/09/2025 15:31:47

Ресурс : HOST: tantor(mounted on:/) Метрика : use_in_percent Кем разрешено : student

Источник : MOUNTPOINT:root Значение : 90 Комментарий : закрыто вручную (для примера)

Страница оповещения: ЖУРНАЛ СОБЫТИЙ

На вкладке ЖУРНАЛ СОБЫТИЙ после описания оповещения выводится история статусов оповещения. В примере на слайде оповещение было закрыто вручную.

При закрытии в поле "решение (необязательно)" был вставлен текст, но на вкладке ЖУРНАЛ СОБЫТИЙ текст не выводится. Видно только, что оповещение "Resolved by student", где student - имя пользователя Платформы.

Текст решения выводится только на вкладке ОБЗОР в поле "комментарий".

Поменять статус с "закрыто" на "открыто" нельзя.

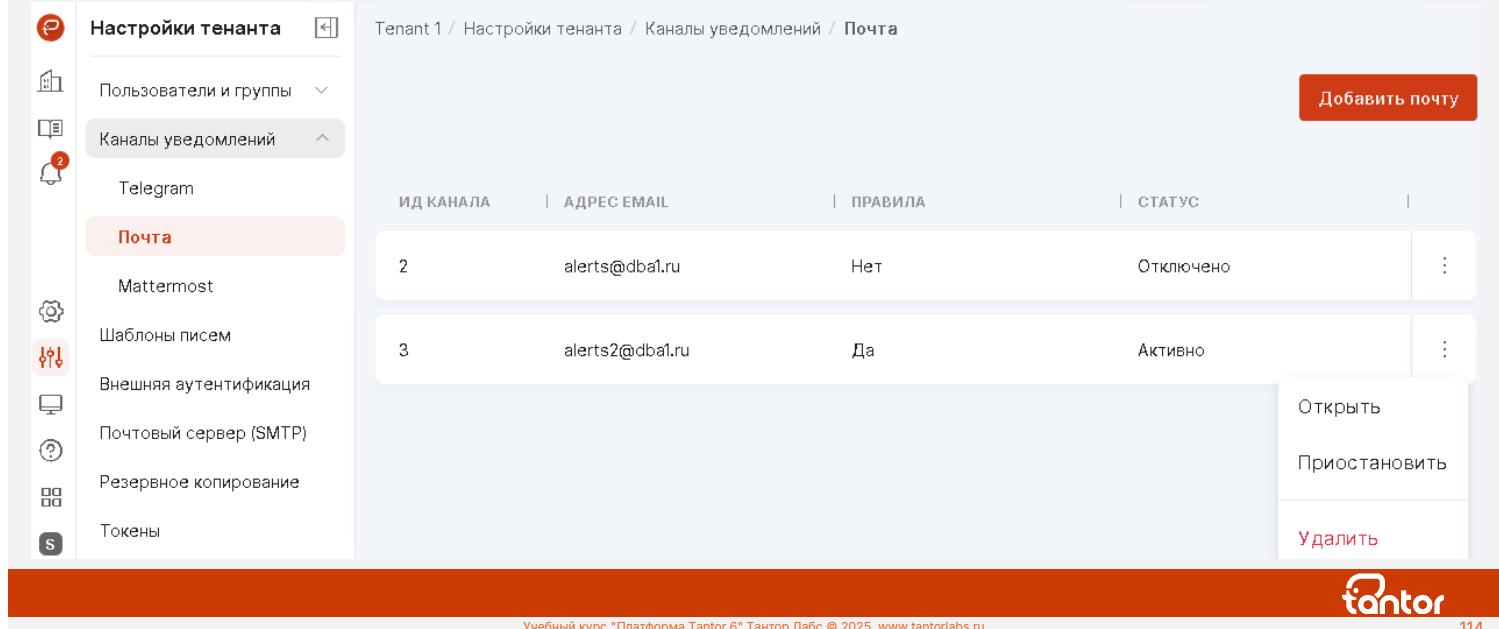


7-2

Уведомления через e-mail

Интеграция со службами сообщений

- уведомления об оповещениях могут передаваться по:
 - e-mail, Telegram, Mattermost



The screenshot shows the 'Email' channel configuration in the Tantor 6 platform. The left sidebar has a 'Tenant 1' icon and a 'Настройки тенанта' (Tenant Settings) section. Under 'Каналы уведомлений' (Notification Channels), 'Почта' (Email) is selected. The main area shows a table of email channels:

ИД КАНАЛА	АДРЕС EMAIL	ПРАВИЛА	СТАТУС
2	alerts@dbat.ru	Нет	Отключено
3	alerts2@dbat.ru	Да	Активно

On the right, for the active channel (ID 3), there is a context menu with options: 'Открыть' (Open), 'Приостановить' (Suspend), and 'Удалить' (Delete). A red 'Добавить почту' (Add Email) button is at the top right of the main area.

Интеграция со службами сообщений

О появлении оповещений можно посыпать уведомления. В Платформе 6 версии поддерживается передача уведомлений:

1. по e-mail
2. в чат Telegram
3. в чат Mattermost.

Это позволяет оперативно получать информацию о происходящих событиях от Платформы.
https://docs.tantorlabs.ru/tp/6.1/admin/admin_reports.html

Настройки тенанта: Почтовый сервер (SMTP)

The screenshot shows the Tenant Settings interface with the 'Настройки тенанта' (Tenant Settings) tab selected. In the left sidebar, 'Пользователи и группы' (Users and groups) is expanded, showing 'Пользователи' (Users), 'Группы' (Groups), and 'Каналы уведомлений' (Notification channels). The 'Почтовый сервер (SMTP)' (SMTP server) section is highlighted. On the right, the 'Настройки тенанта / Настройки тенанта / Почтовый сервер (SMTP)' path is visible. A 'Удалить' (Delete) button, a 'Протестировать' (Test) button, and a 'Редактировать' (Edit) button are at the top right. The main area displays the following configuration:

SMTP сервер	smtp.yandex.com
SMTP порт	465
Использовать SSL/TLS	Используется
Имя пользователя	oleg.ivanov@tantorlabs.ru

Below this, a green box shows a checkmark and the text 'Тестовое письмо успешно отправлено' (Test message successfully sent). A red box shows the text 'Не удалось проверить подключение к SMTP-серверу' (Failed to check connection to SMTP server).

A modal window titled 'Настройка почтового сервера' (SMTP server configuration) is open on the right. It contains the same configuration fields with their current values: 'SMTP сервер' (smtp.yandex.com), 'SMTP порт' (465), 'Использовать SSL/TLS' (checked), 'Имя пользователя' (oleg.ivanov@tantorlabs.ru), and a password field. The modal also includes a note 'Почтовый сервер требует авторизации' (SMTP server requires authentication) and 'Пароль' (Password) with a red asterisk. At the bottom are 'Отменить' (Cancel) and 'Сохранить' (Save) buttons.

Учебный курс "Платформа Tantor 6" Тантор Лабс © 2025 www.tantorlabs.ru

tantor

115

Настройки тенанта: Почтовый сервер (SMTP)

Подключение к почтовому серверу настраивается:

- на странице "**Настройки системы**" - для отправки владельцам системы писем о выдаче прав владельца системы или о смене пароля владельца системы.
- на странице "**Настройки тенанта**" - для отправки писем о событиях тенанта: сообщений о смене пароля пользователя с правами владельца тенанта, локальных пользователей, а также уведомлений о возникновении оповещений.

На страницах указываются детали подсоединения к почтовому серверу по протоколу SMTP (Simple Mail Transfer Protocol). По этому протоколу отсылаются почтовые сообщения уведомлений.

Для добавления почтового сервера нужно ввести имя его хоста, порт для протокола SMTP (обычно 25 или 465), имя для аутентификации отправителя и пароль. Галочка SSL/TLS не играет роли, так как Платформа отправляет письма, переключая почтовую сессию на протокол TLS.

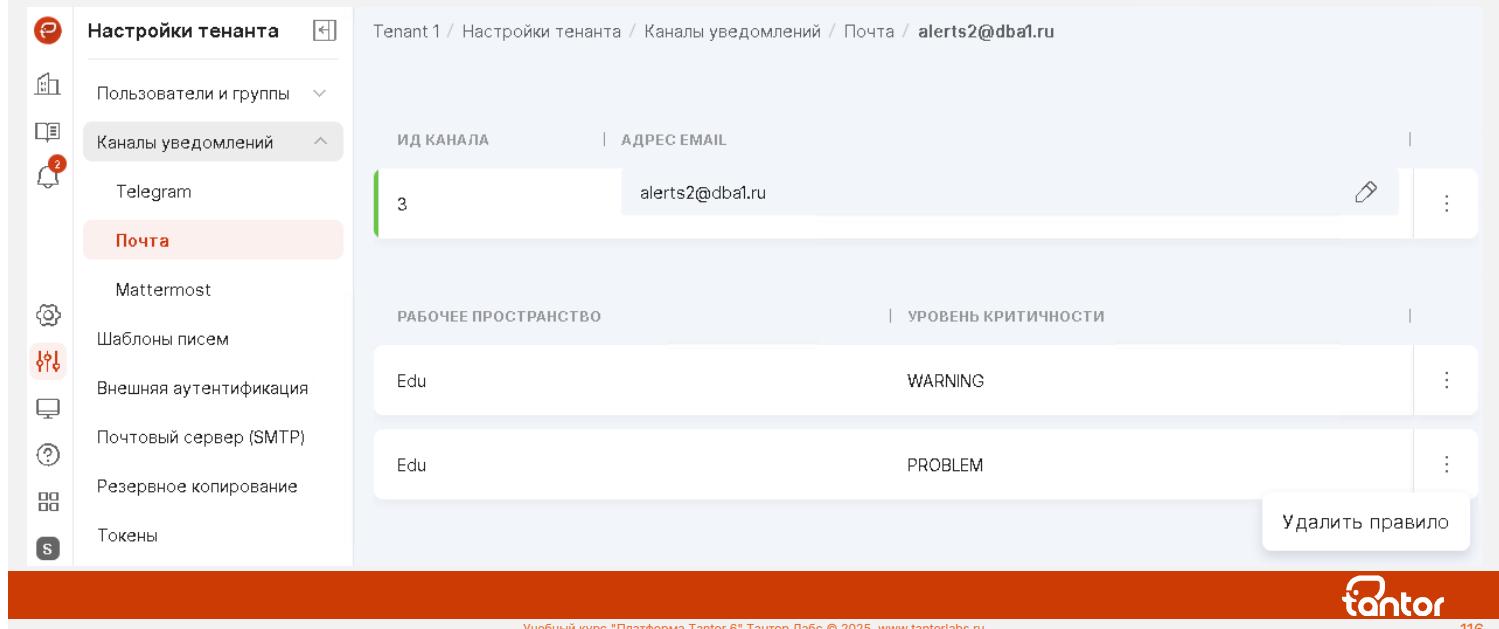
После добавления деталей почтового сервера, можно протестировать отправку письма. Для этого нужно нажать на кнопку "Протестировать", в появившемся окне указать e-mail получателя письма.

Если почтовый сервер примет сообщение, то на несколько секунд появится зелёное окно с текстом "Тестовое письмо успешно отправлено". Это будет означать, что настройка SMTP сервера выполнена корректно. При любой ошибке отправки письма (неверные имя/пароль, сервер использует протокол TLS 1.0, сервер не понимает команду STARTTLS) выдаётся красное всплывающее окно с одинаковым текстом "Не удалось проверить подключение к SMTP-серверу" без указания причины.

<https://docs.tantorlabs.ru/tp/6.1/admin/smtp.html>

Интеграция со службами сообщений

- В правилах уведомления можно указать:
 - пространство
 - уровень важности (Проблема, Предупреждение)



Настройки тенанта

Tenant 1 / Настройки тенанта / Каналы уведомлений / Почта / alerts2@dbal.ru

ИД КАНАЛА | АДРЕС EMAIL

3 | alerts2@dbal.ru

РАБОЧЕЕ ПРОСТРАНСТВО | УРОВЕНЬ КРИТИЧНОСТИ

Edu | WARNING

Edu | PROBLEM

Удалить правило

Пользователи и группы

Каналы уведомлений

Telegram

Почта

Mattermost

Шаблоны писем

Внешняя аутентификация

Почтовый сервер (SMTP)

Резервное копирование

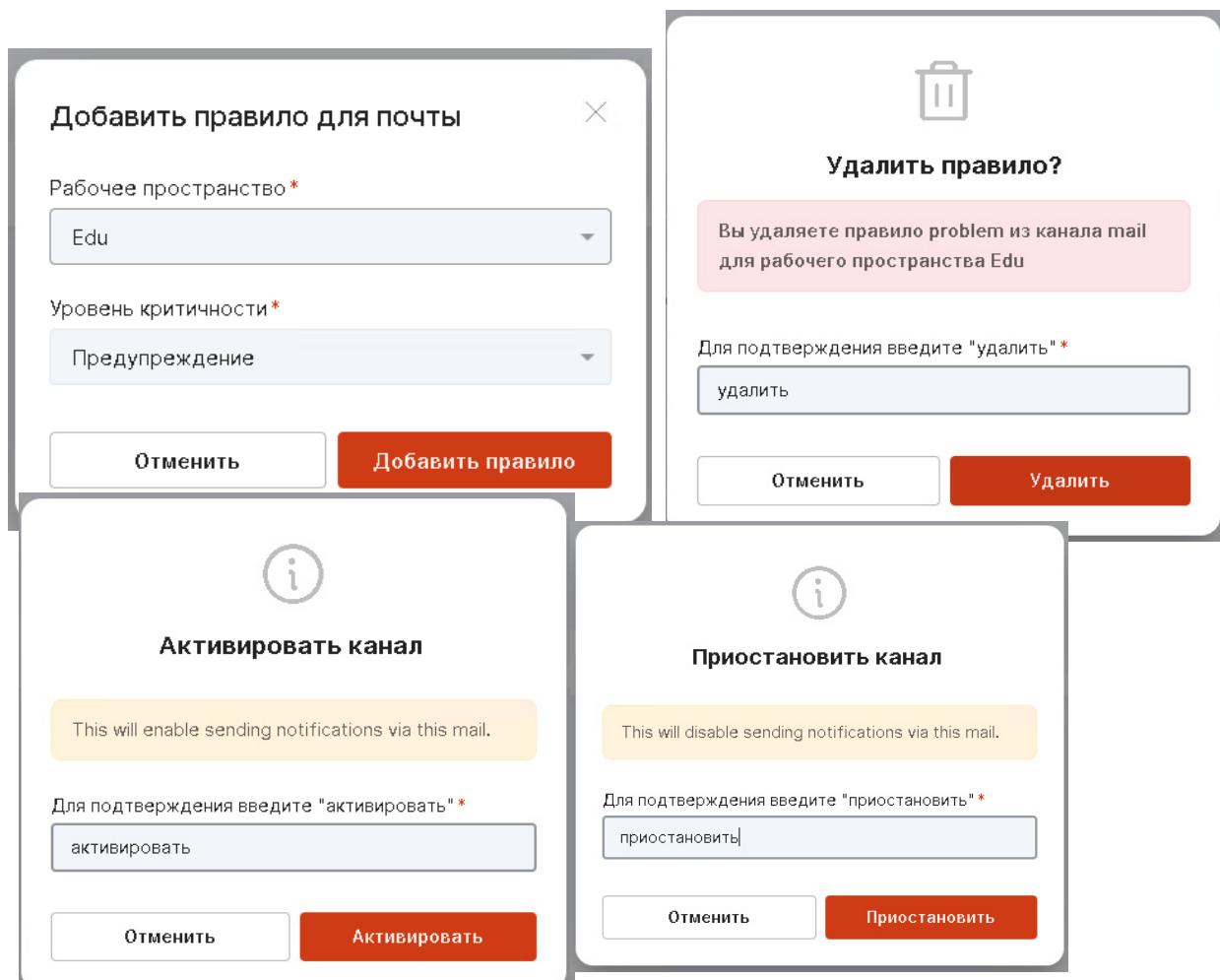
Токены

tantor

Учебный курс "Платформа Tantor 6" Тантор Лабс © 2025 www.tantolabs.ru

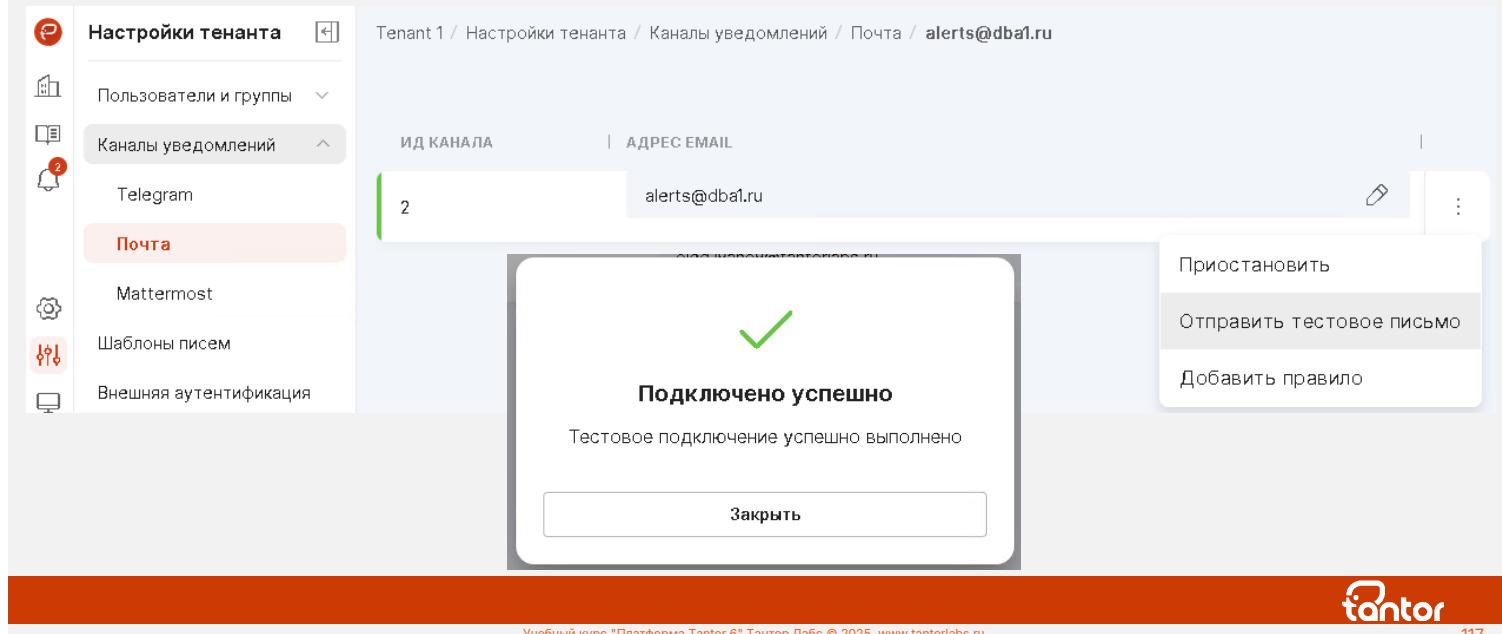
Интеграция со службами сообщений

Можно посыпать уведомления разной важности на разные e-mail. Для этого нужно добавить два e-mail и добавить правила. В правиле к первому e-mail указать уровень важности(критичности) "Предупреждение", а в правиле ко второму e-mail уровень важности "Проблема". Канал передачи (e-mail) можно приостановить и активировать. В приостановленных каналах менять правила нельзя, в активированных можно. Удалить можно любой канал.



Отправка тестового письма

- Для отправки тестового письма на странице "Настройки тенанта": должен быть сконфигурирован "Почтовый сервер (SMTP)"



The screenshot shows the 'Настройки тенанта' (Tenant Settings) page. On the left, a sidebar lists 'Пользователи и группы', 'Каналы уведомлений' (selected), 'Telegram', 'Почта' (selected), 'Mattermost', 'Шаблоны писем', and 'Внешняя аутентификация'. The main content area shows a table with one row for a mail channel. The table has columns 'ИД КАНАЛА' (ID) and 'АДРЕС EMAIL' (Email Address). The row contains the value '2' in the ID column and 'alerts@dba1.ru' in the email column. To the right of the table is a context menu with options: 'Приостановить' (Suspend), 'Отправить тестовое письмо' (Send test email), and 'Добавить правило' (Add rule). A modal window in the center says 'ПОДКЛЮЧЕНИЕ УСПЕШНО' (Connection successful) with a green checkmark and the message 'Тестовое подключение успешно выполнено' (Test connection successfully completed). A 'Закрыть' (Close) button is at the bottom of the modal. The top navigation bar shows 'Tenant 1 / Настройки тенанта / Каналы уведомлений / Почта / alerts@dba1.ru'.

Отправка тестового письма

Для отправки тестового письма на странице "Настройки тенанта": должен быть сконфигурирован "Почтовый сервер (SMTP)".

Кликнув на три точки справа в строке, появится контекстное меню с пунктом "Отправить тестовое письмо". Нажав на него, будет сделана попытка отправки почтового сообщения. Если сообщение будет успешно принято SMTP-сервером, то появится окно "Подключено успешно".

https://docs.tantorlabs.ru/tp/6.1/admin/reports_email.html

Шаблоны писем

Настройки тенанта

Пользователи и группы

Пользователи

Группы

Каналы уведомлений

Шаблоны писем

Внешняя аутентификация

Почтовый сервер (SMTP)

Резервное копирование

Токены

Название шаблона

Оповещения	Предпросмотр	Редактировать	Проверить
Добавлено ЦПУ	Предпросмотр	Редактировать	Проверить
Восстановление пароля	Предпросмотр	Редактировать	Проверить
Подтверждение настроек email	Предпросмотр	Редактировать	Проверить
Новый пользователь	Предпросмотр	Редактировать	Проверить
Проблема с оплатой	Предпросмотр	Редактировать	Проверить
Anonymizer Operation Status	Предпросмотр	Редактировать	Проверить
Регулярные уведомления	Предпросмотр	Редактировать	Проверить
Сброс пароля	Предпросмотр	Редактировать	Проверить
Owner permissions granted for Tenant	Предпросмотр	Редактировать	Проверить
User permissions granted for Tenant	Предпросмотр	Редактировать	Проверить

tantor 118

Шаблоны писем

В Платформе 6 версии есть 11 шаблонов писем для уведомлений о событиях тенанта. Шаблонов для писем на странице "Настройки системы" нет. Названия шаблонов:

Оповещения

Добавлено ЦПУ

Восстановление пароля

Подтверждение настроек email

Новый пользователь

Проблема с оплатой

Anonymizer Operation Status

Регулярные уведомления

Сброс пароля

Owner permissions granted for Tenant

User permissions granted for Tenant.

Список шаблонов не расширяется (в него нельзя добавить еще один шаблон).

Редактор шаблона писем

Your password was changed to newpassword.

If you did not make this change or believe an unauthorized person has accessed your account, please reset your password immediately [here](#).

If you have any problems, please contact us at support@tantorlabs.ru

Good Luck!

TANTORLABS team

Настройки тенанта Tenant 1 / Настройки тенанта / Шаблоны писем

Предпросмотр Сохранить

ПОЛЯ ШАБЛОНА:

{SENDER_NAME}
{NEW_PASSWORD}
{PRODUCT_DOMAIN}
{PLATFORM_DOMAIN}
{ENDPOINT_PATH}

Редактор шаблона писем

Можно отредактировать шаблоны писем.

В правой части списка мы видим команды, можем нажать на "Предпросмотр" того, как будет выглядеть тело письма.

Можно отредактировать тему (Subject) письма. В заголовке нельзя использовать "ПОЛЯ ШАБЛОНА" - переменные замены.

Переменные замены будут заменены текстом с данными, которые были доступны Платформе на момент отправки письма.

Формат тела письма - **HTML**.

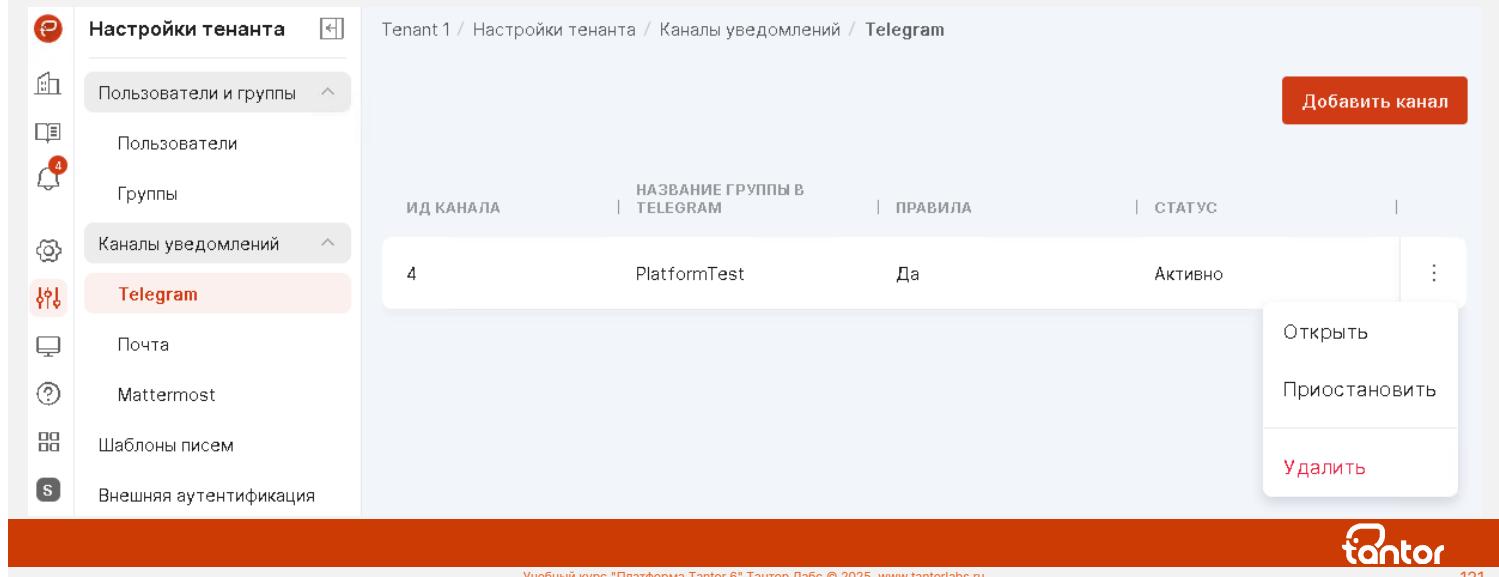


7-3

Уведомления через Telegram и Mattermost

Посылка сообщений в канал Telegram

- хост Платформы должен иметь доступ к telegram.org
- Оповещения посылаются по всем каналам (e-mail, telegram) в соответствии с правилами



Настройки тенанта Tenant 1 / Настройки тенанта / Каналы уведомлений / Telegram

Добавить канал

ИД КАНАЛА	НАЗВАНИЕ ГРУППЫ В TELEGRAM	ПРАВИЛА	СТАТУС
4	PlatformTest	Да	Активно

Контекстное меню для канала PlatformTest:

- Открыть
- Приостановить
- Удалить

Посылка сообщений в канал Telegram

Для настройки и корректной работы интеграции с Telegram нужен доступ к хосту telegram.org из Платформы.

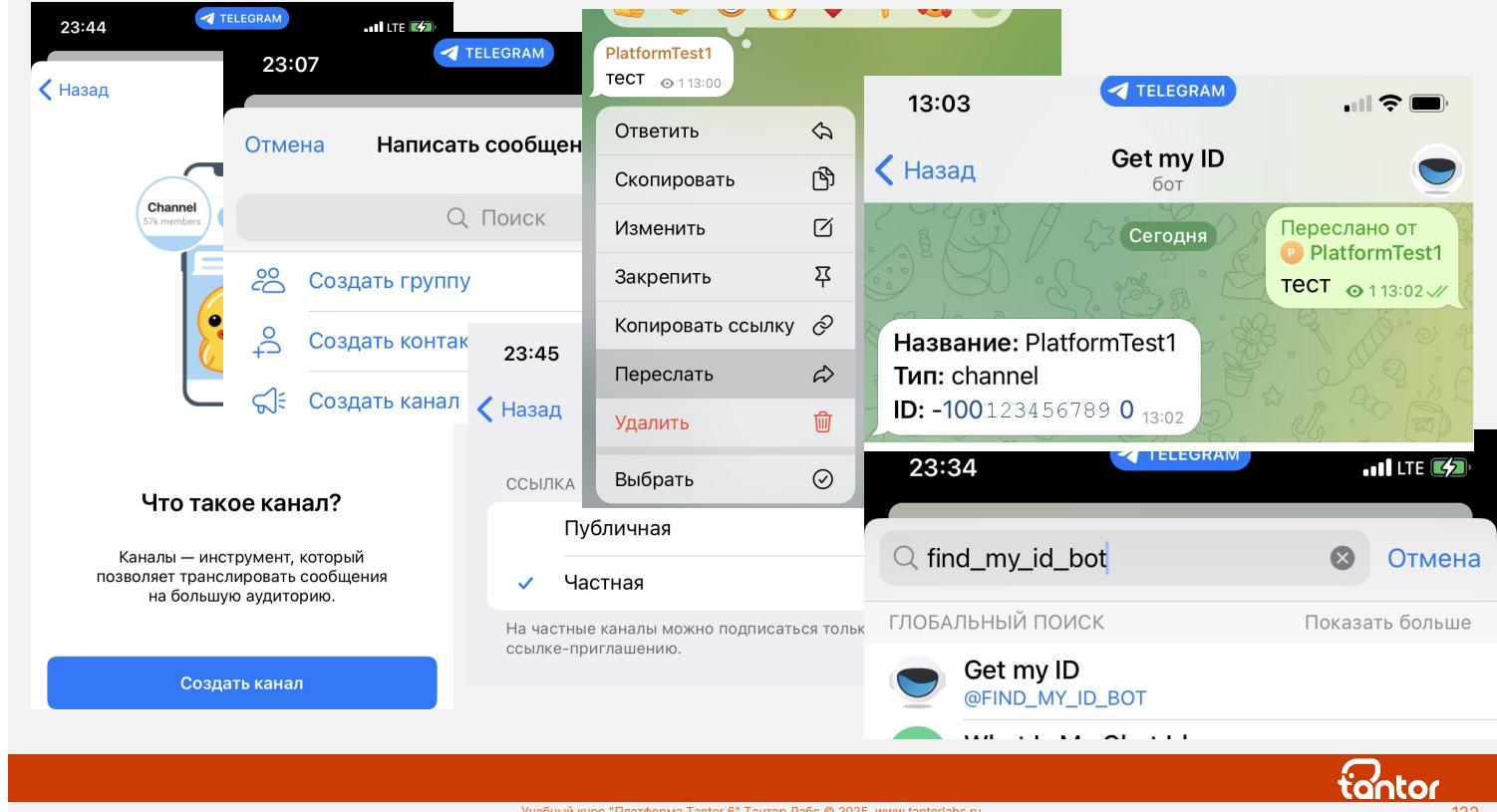
Оповещения посылаются по всем **активным** каналам (трёх типов e-mail, telegram, mattermost) в соответствии с правилами, установленными для каналов. То есть получая уведомления по e-mail, сконфигурировав telegram уведомления будут отправляться и на e-mail и в telegram-канал. Правила позволяют настроить передачу сообщений по пространствам и уровням важности (WARNING и PROBLEM). Параметров "тип сообщений" или "'экземпляр PostgreSQL" в правилах нет.

Если правил нет, то уведомления не посылаются.

"Приостановить" в выпадающем меню - отключит отправку уведомлений по каналу. Статус канала будет отображаться как "Отключено", а не "**Активно**".

https://docs.tantorlabs.ru/tp/6.1/admin/reports_telegram.html

Создание канала Telegram



Создание канала Telegram

В приложении Telegram нужно:

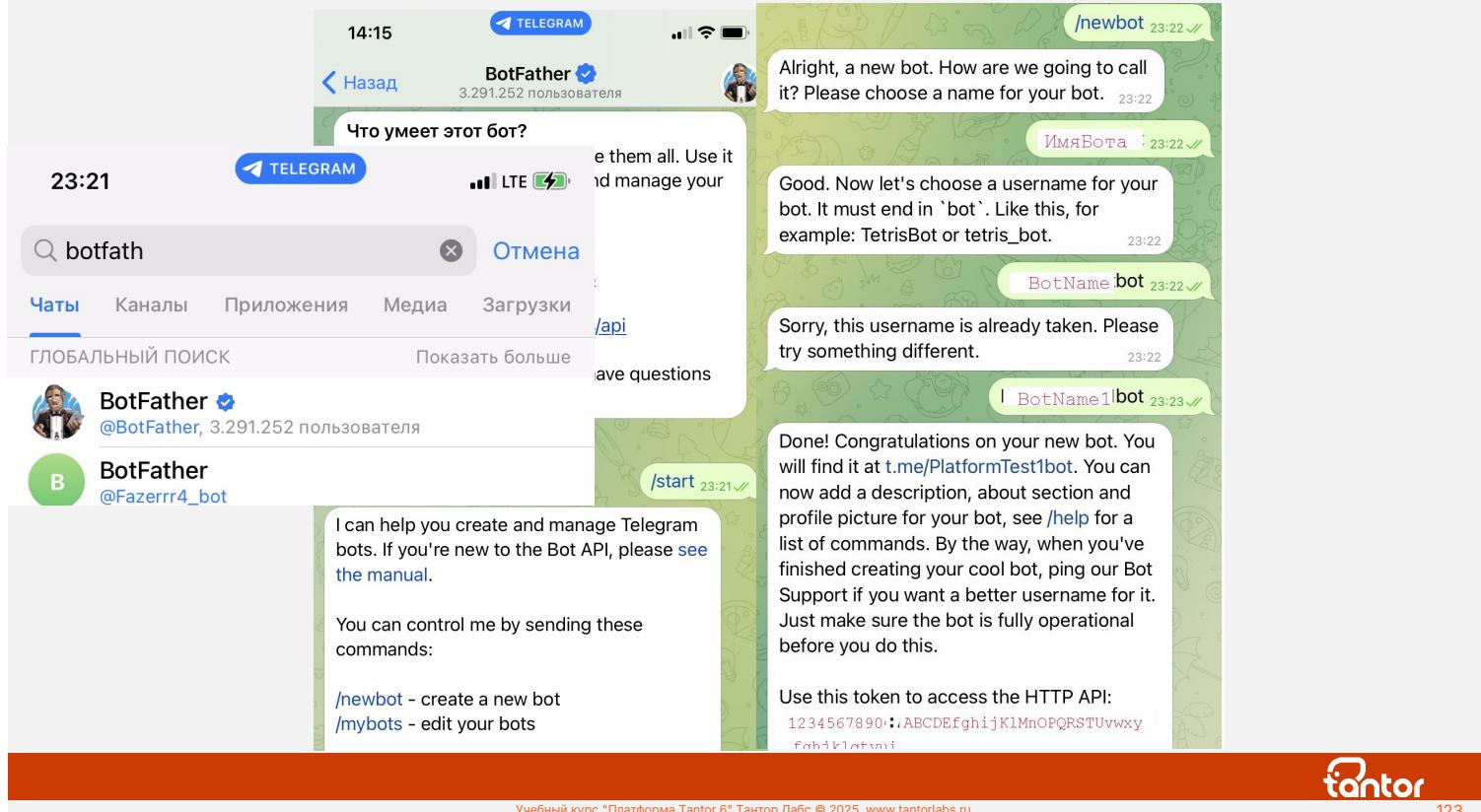
Создать частный канал (НЕ группу и НЕ чат).

В приложении Telegram выберите "Создать канал". Введите имя канала. В окне "Типа канала" отобразится "Частная" и ссылка-приглашение. Частный канал означает, что добавляться в него смогут только по этой ссылке. Эта ссылка Платформой не используется, её можно передавать пользователям Платформы, которые хотят получать уведомления. Пользователь Telegram, создавший канал является его владельцем, может добавлять/удалять администраторов канала и удалять канал. Если нажать "Далее", то появится окно "Контакты", в котором Telegram склоняет к приглашению в канал контактов. Если нажать "Далее" откроется окно канала с сообщением "Канал создан" и поле для публикации сообщений в канале.

Получить идентификатор созданного частного канала из 13 чисел с помощью бота **@FIND_MY_ID_BOT**

Напишите любое сообщение и опубликуйте в канале. Нажмите на сообщение и выберите в выпадающем меню "Переслать". В строке поиска наберите @find_my_id_bot и выберите имя "Get my ID". Сверьтесь с названием и аватаром бота. Вместо бота "Get my ID" можно использовать и другие боты, которые могут выдать идентификатор канала. Идентификатор канала это 13 цифр со значком минус в начале: "-100xxxxxxxxxx". Запишите этот номер, его нужно будет ввести при добавлении канала в Платформе.

Создание бота для канала Telegram



Создание бота для канала Telegram

Бот нужен для того, чтобы Платформа через него публиковала сообщения в канале. Пользователи Telegram привязаны к номерам телефонов и Платформа не может публиковать сообщения как пользователь (человек) Telegram.

Создать бота с помощью бота @BotFather, получить API Token бота.

Найдите бот по имени "BotFather". Не ошибитесь в выборе бота: У правильного бота больше 3 миллионов пользователей и синяя галочка.

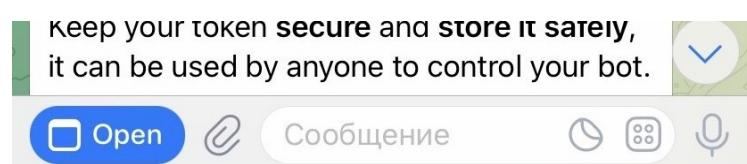
В окне чата с ботом BotFather наберите:

/start, высветятся команды

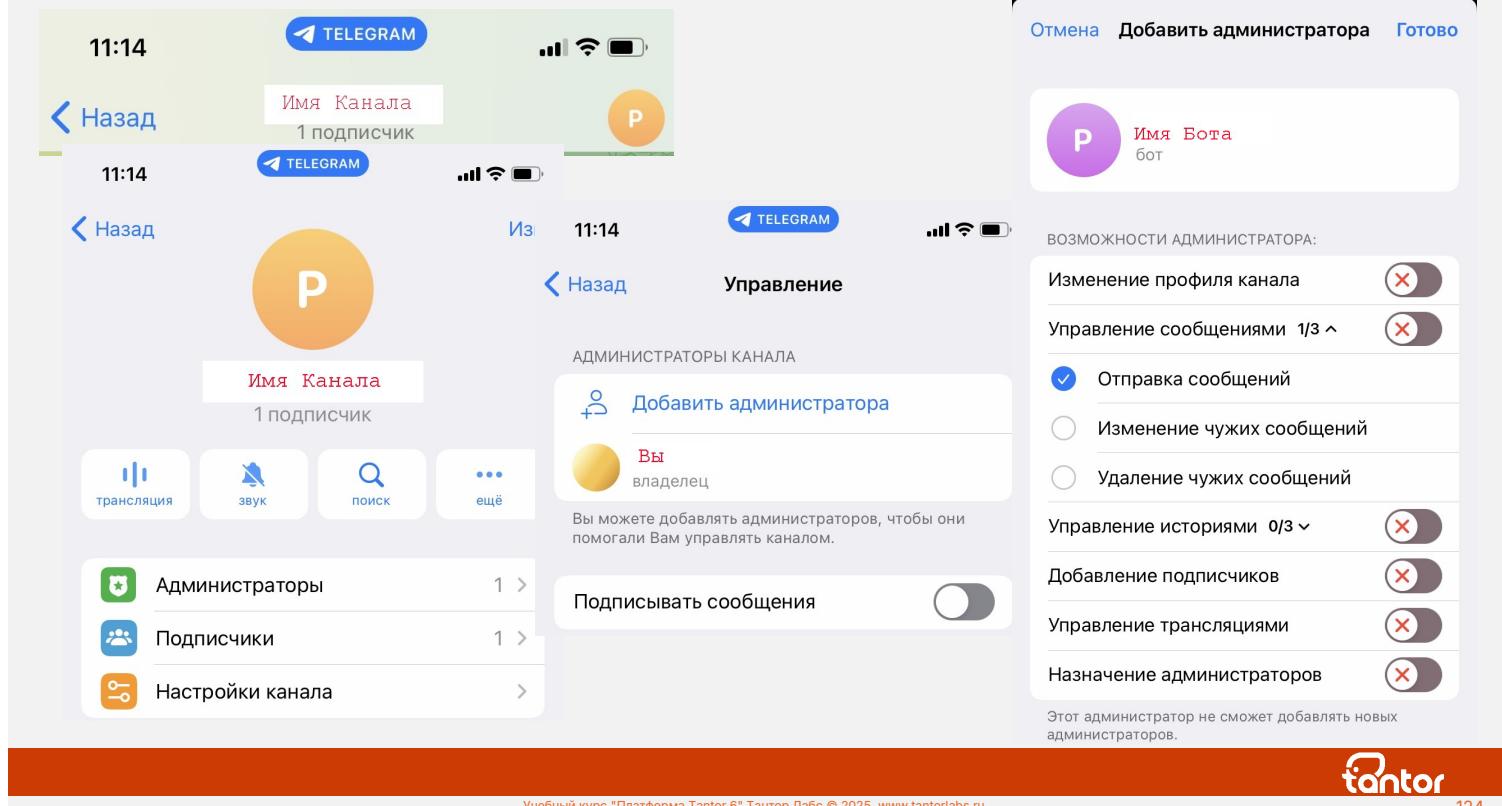
наберите /newbot, выдастся сообщение о том, что надо придумать и ввести имя вашему боту

Имя не уникальное, оно будет высвечиваться у вас в Telegram. Дальше будет предложено придумать username боту. username должно оканчиваться на "bot" и быть уникальным.

Придумайте username, которое ещё не занято. Можно набрать несколько раз. Как только введённое имя окажется незанятым, ваш бот будет создан и в выданном тексте будет API Token. Его нужно сохранить. Токен можно в будущем снова получить, можно отозвать и создать новый. Также можно будет удалить бота, если он вам не нужен. В приложении Telegram можно нажать на кнопку Open и давать команды BotFather через встроенное приложение в Telegram:



Добавление бота в канал Telegram



Добавление бота в канал Telegram

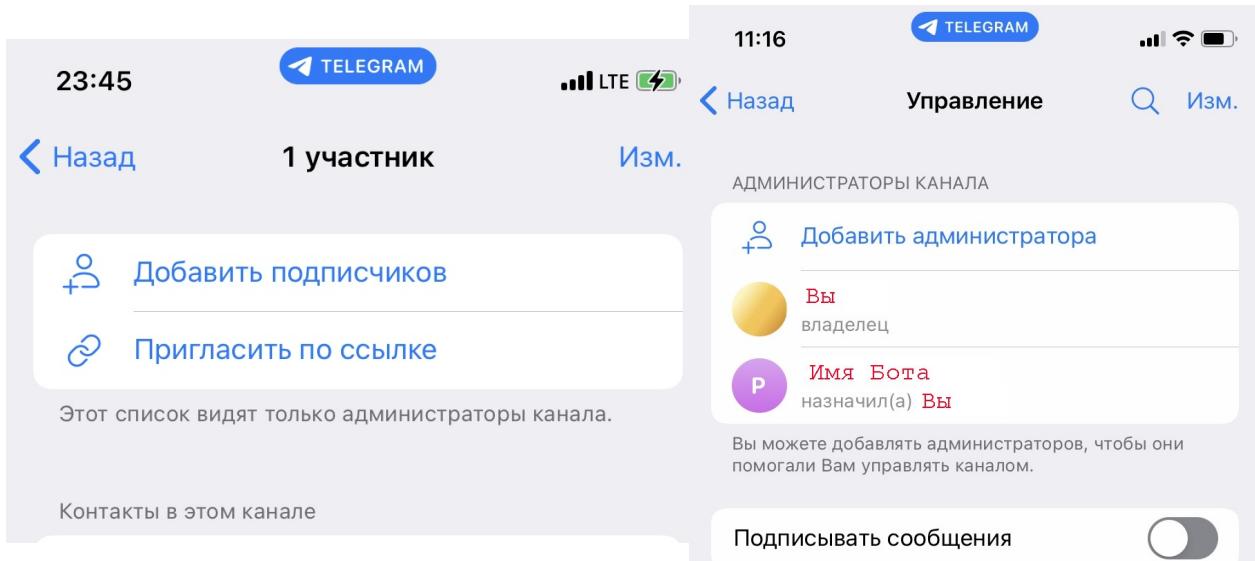
Добавить созданного бота как администратора в канал с единственным правом - публикации сообщений (остальные права не нужно давать).

Для добавления нужно открыть канал, кликнуть на заголовок с именем канала. Выбрать "Администраторы". В открывшемся окне "Управление" выбрать "Добавить администратора". Найти имя вашего бота в окне глобального поиска. Не ошибайтесь в имени бота при поиске, а то добавите чужого бота. Выбрав имя откроется окно с правами. Нужно убрать все права, кроме "Отправка сообщений" и нажать ссылку в верхнем правом углу "Готово".

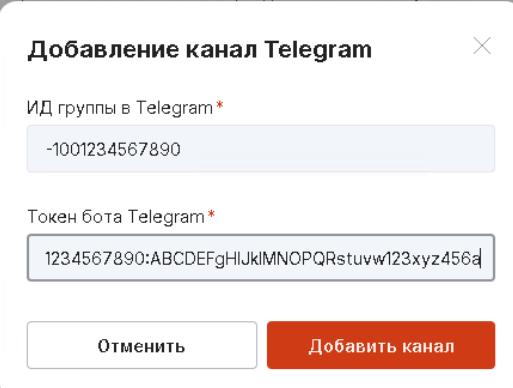
Это все действия, которые нужно выполнить в приложении Telegram. В канал можно приглашать пользователей, которые хотят получать сообщения от Платформы. Так как канал частный, то он не будет виден в глобальном поиске и без ссылки или добавления в канал, сторонние пользователи Telegram не смогут добавиться в канал.

Созданный канал можно удалить, имя канала освободится. Боту можно поменять API Token.

Как создатель каналы вы являетесь владельцем канала и можете удалить его. Бота можно из списка администраторов канала:



Создание канала уведомлений в Платформе

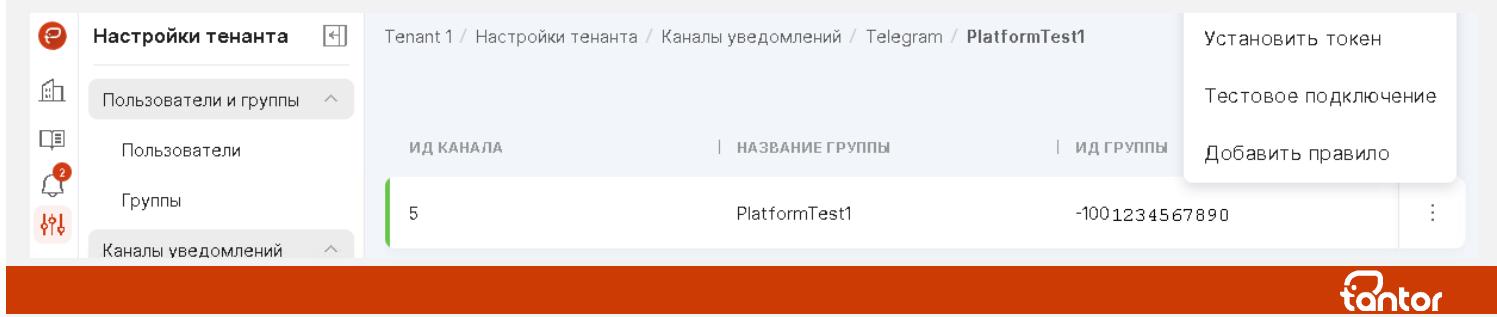


Добавление канала Telegram

ИД группы в Telegram*
-1001234567890

Токен бота Telegram*
1234567890:ABCDEFgHIJklMNOPQRstuvwxyz123xyz4567

Добавить канал



ИД КАНАЛА	НАЗВАНИЕ ГРУППЫ В TELEGRAM	ИД ГРУППЫ
5	PlatformTest1	-1001234567890

Установить токен

Тестовое подключение

Добавить правило

Создание канала уведомлений в Платформе

Создать в Платформе канал уведомлений, указав идентификатор канала и API Token бота, которые были получены в приложении Telegram.

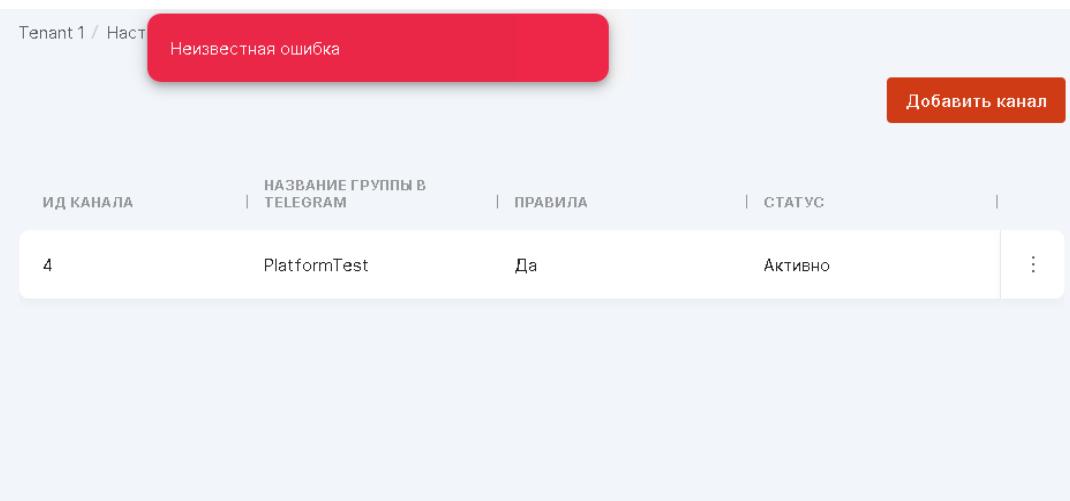
Для этого выбрать в левой вертикальной полоске меню значок "Настройки тенанта" -> "Каналы уведомлений" -> "Telegram" и на открывшейся странице со списком каналов Telegram нажать кнопку в правом верхнем углу "Добавить канал". Появится всплывающее окно "Добавление канала Telegram".

После ввода идентификатора и токена нажать красную кнопку "Добавить канал" и канал будет добавлен в список каналов тенанта.

Дальше нажать справа в строке канала три точки и выбрать "Открыть". Появится страница описания канала. На странице отображается идентификатор канала, но API Token не отображается. Если вы в BotFather отзовёте токен (Revoke current token), то будет создан **новый** API Token. Все клиенты со старым токеном не смогут взаимодействовать с ботом. Чтобы Платформа смогла передавать сообщения в канал, нужно будет выбрать в меню "Установить токен" и ввести новый API Token.

Для проверки того, что Платформа может посыпать сообщения в канал, можно нажать в меню на пункт "Тестовое подключение".

Если удалить бота из администраторов канала или сам канал, то зелёная плоска с лева в строке не будет показываться и при тестировании будет выдана ошибка:



Неизвестная ошибка

ИД КАНАЛА	НАЗВАНИЕ ГРУППЫ В TELEGRAM	ПРАВИЛА	СТАТУС
4	PlatformTest	Да	Активно

Добавить канал

Добавление правил для канала Telegram в Платформе

Настройки тенанта

Пользователи и группы

Пользователи

Группы

Каналы уведомлений

Настройки тенанта

Пользователи и группы

Пользователи

Группы

Каналы уведомлений

Telegram

Почта

Mattermost

Шаблоны писем

Внешняя аутентификация

Добавление правила для канала уведомления PlatformTest1

Рабочее пространство*

Edu

Уровень критичности*

Проблема

Отменить

Добавить правило

ИД КАНАЛА

5

ИД ГРУППЫ

-100 1234567890

Test1

ИД КАНАЛА

5

ИД ГРУППЫ

-100 1234567890

РАБОЧЕЕ ПРОСТРАНСТВО | УРОВЕНЬ КРИТИЧНОСТИ

Edu | PROBLEM

Edu | WARNING

Удалить правило

tantor

126

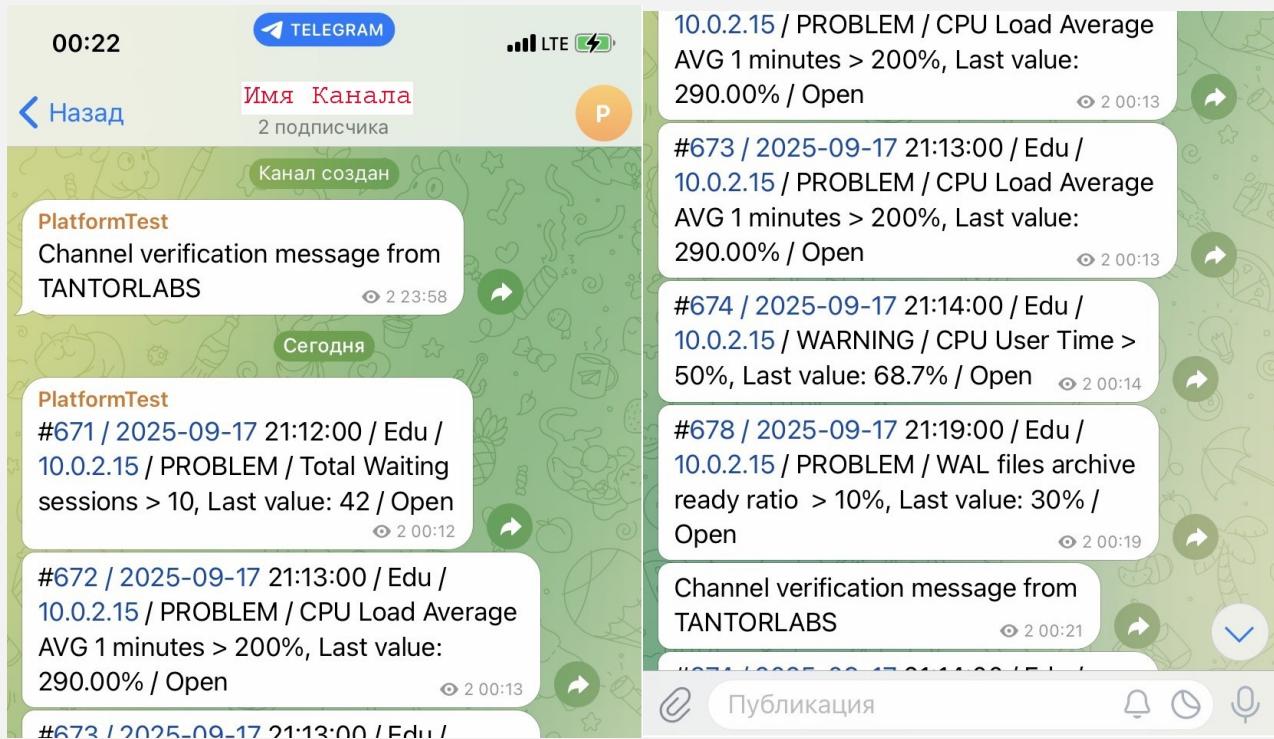
Добавление правил для канала Telegram в Платформе

Добавить правило для отсылки уведомлений в канал Telegram для оповещений с уровнем важности Проблема (PROBLEM) и/или Предупреждение (WARNING).

В меню справа выбрать "Добавить правило" и во всплывающем окне выбрать пространство в тенанте и уровень важности. Выбрать несколько пространств и уровней нельзя, придётся добавлять по одному.

Добавленные правила появятся на странице канала в виде списка под строкой с параметрами канала уведомлений. В меню, которое вызывается при клике на три точки справа на строке правила есть единственный пункт "Удалить правило".

Пример сообщений в канале Telegram



Пример сообщений в канале Telegram

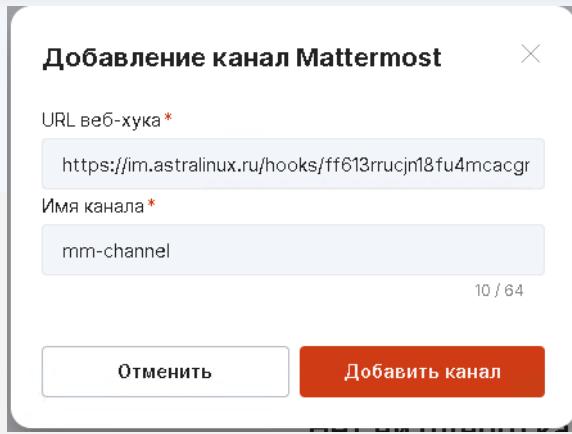
Первое сообщение - результат выбора в меню пункта "Тестовое подключение".

Шаблоны сообщений для Telegram не редактируются. В сообщении выводится время во временной зоне Платформы с точностью до секунд. Время такое же как на странице "Оповещения" Платформы.

В примере на слайде в канале два подписчика: владелец канала и бот.

Числа в сообщениях отображаются синим цветом, это особенность Telegram, кликать на числа синего цвета не нужно.

Mattermost



Настройки тенанта

Пользователи и группы

Пользователи

Группы

Каналы уведомлений

Telegram

Почта

Mattermost

Шаблоны писем

Канал успешно активирован

Добавить канал

ИД КАНАЛА	ИМЯ КАНАЛА	URL ВЕБ-ХУКА	СТАТУС
6	mm-channel	https://im.astralinux.ru/hooks/ff613rrucjn18fu4mcacgn41tr	Активно

Добавление канала Mattermost

URL веб-хука *

https://im.astralinux.ru/hooks/ff613rrucjn18fu4mcacgr

Имя канала *

mm-channel

10 / 64

Отменить Добавить канал

Открыть

Приостановить

Удалить

Учебный курс "Платформа Tantor 6" Тантор Лабс © 2025 www.tantolabs.ru

tantor

128

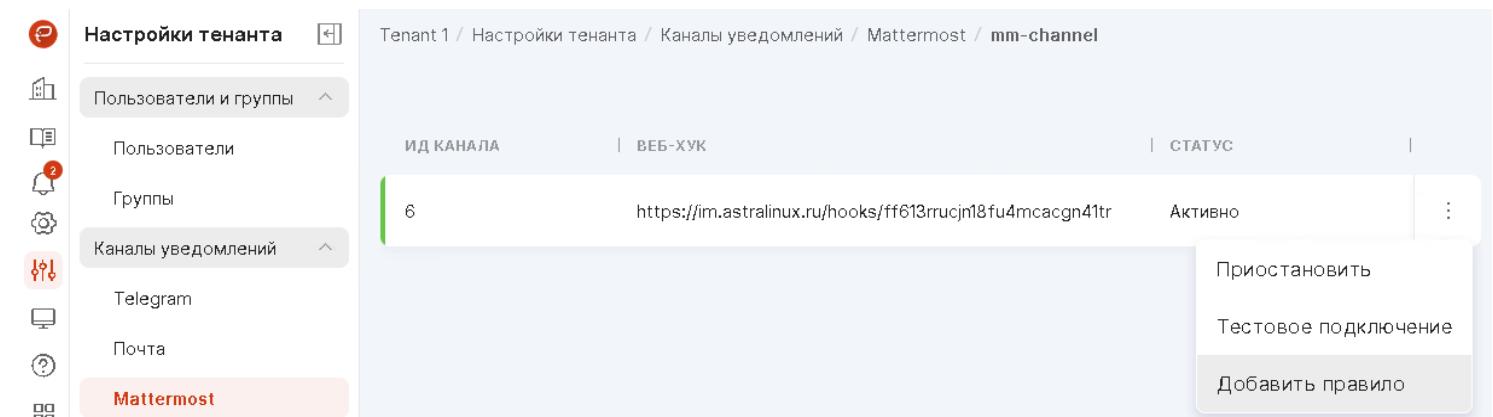
Mattermost

Mattermost - мессенджер с открытым исходным кодом для использования в организациях и компаниях. Является открытой альтернативой Slack и Microsoft Teams. Mattermost используют Группа Компаний Астра, Boeing, Tesla, ЦЕРН. Позволяет выполнить миграцию со службы уведомлений Slack. Mattermost имеет клиентское приложение под iOS, доступное в AppStore.

При добавлении канала Mattermost на странице Платформы достаточно указать "URL веб-хука" и канала, для которого этот веб-хук создан. Канал может быть приватным.

Веб-хук для канала создаётся в настройки канала Mattermost в меню "Интеграции" -> "Входящие веб-хуки" -> кнопка "Добавить входящий веб-хук".

В остальном, **работа с каналами Mattermost в Платформе аналогична Telegram**. Например, так же нужно добавлять правила для оповещений:



Настройки тенанта

Пользователи и группы

Пользователи

Группы

Каналы уведомлений

Telegram

Почта

Mattermost

Tenant 1 / Настройки тенанта / Каналы уведомлений / Mattermost / mm-channel

ИД КАНАЛА	ВЕБ-ХУК	СТАТУС
6	https://im.astralinux.ru/hooks/ff613rrucjn18fu4mcacgn41tr	Активно

Приостановить

Тестовое подключение

Добавить правило

Во всплывающем меню, в отличие от меню для каналов Telegram, нет пункта "Установить токен", так как в Mattermost нет токенов.

При выборе Приостановить, Активировать и при добавлении канала появляются всплывающие сообщения зелёного цвета с текстом об успешной приостановке, активации или добавлении канала.

https://docs.tantolabs.ru/tp/6.1/admin/reports_mattermost.html

Практика 7

1. Посмотрите уведомления (оповещения)
 2. Посмотрите какие параметры есть для фильтрации оповещений
 3. Закройте оповещение вручную
 4. Посмотрите страницу оповещения и вкладки на ней
 5. Посмотрите содержимое вкладки связанных оповещений
 6. Посмотрите настройки каналов (способов) оповещений типа Telegram
 7. Добавьте и удалите правило оповещения
 8. Приостановите и активируйте канал типа Telegram
 9. Посмотрите настройки для каналов типа e-mail
 10. В настройках мониторинга посмотрите содержимое вкладки ТРИГГЕРА
 11. Уменьшите чувствительность триггера свободной оперативной памяти
 12. Запустите скрипт для создания нагрузки к следующей практике
- ```
pgbench -T 600 -P 60 -c 5 -f /opt/workload/custom_workload.sql
```

## Практика 7

1. Посмотрите уведомления (оповещения)
  2. Посмотрите какие параметры есть для фильтрации оповещений
  3. Закройте оповещение вручную
  4. Посмотрите страницу оповещения и вкладки на ней
  5. Посмотрите содержимое вкладки связанных оповещений
  6. Посмотрите настройки каналов (способов) оповещений типа Telegram
  7. Добавьте и удалите правило оповещения
  8. Приостановите и активируйте канал типа Telegram
  9. Посмотрите настройки для каналов типа e-mail
  10. В настройках мониторинга посмотрите содержимое вкладки ТРИГГЕРА
  11. Уменьшите чувствительность триггера свободной оперативной памяти (Memory available)
  12. Запустите скрипт для создания нагрузки к следующей практике
- ```
pgbench -T 600 -P 60 -c 5 -f /opt/workload/custom_workload.sql
```

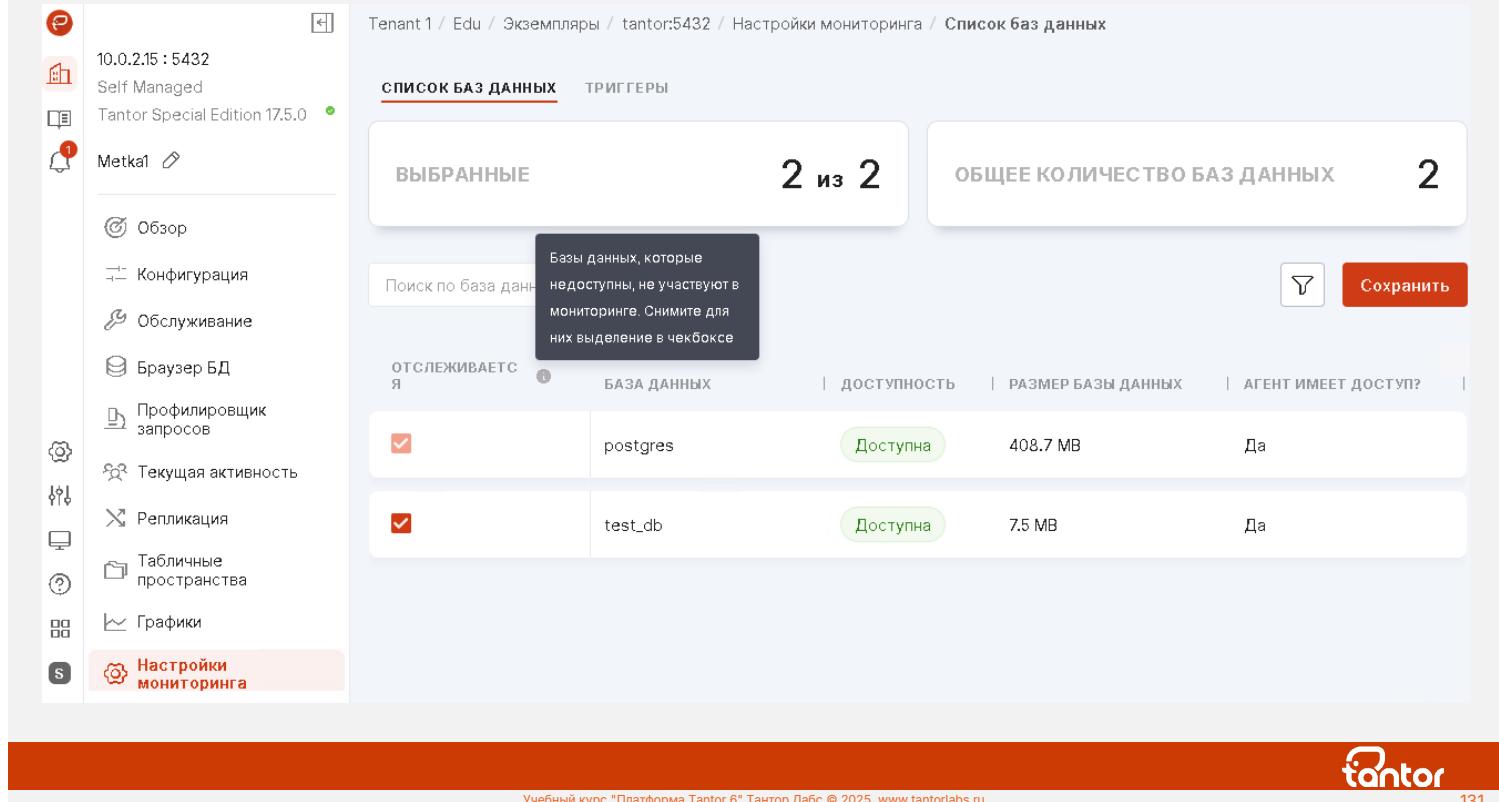


8

Экземпляры PostgreSQL

Профиliровщик запросов

Экземпляр: Настройки мониторинга



10.0.2.15 : 5432
Self Managed
Tantor Special Edition 17.5.0

Metka1

Обзор
Конфигурация
Обслуживание
Браузер БД
Профилировщик запросов
Текущая активность
Репликация
Табличные пространства
Графики
Настройки мониторинга

Список баз данных ТРИГГЕРЫ

ВЫБРАННЫЕ 2 из 2

ОБЩЕЕ КОЛИЧЕСТВО БАЗ ДАННЫХ 2

Базы данных, которые недоступны, не участвуют в мониторинге. Снимите для них выделение в чекбоксе

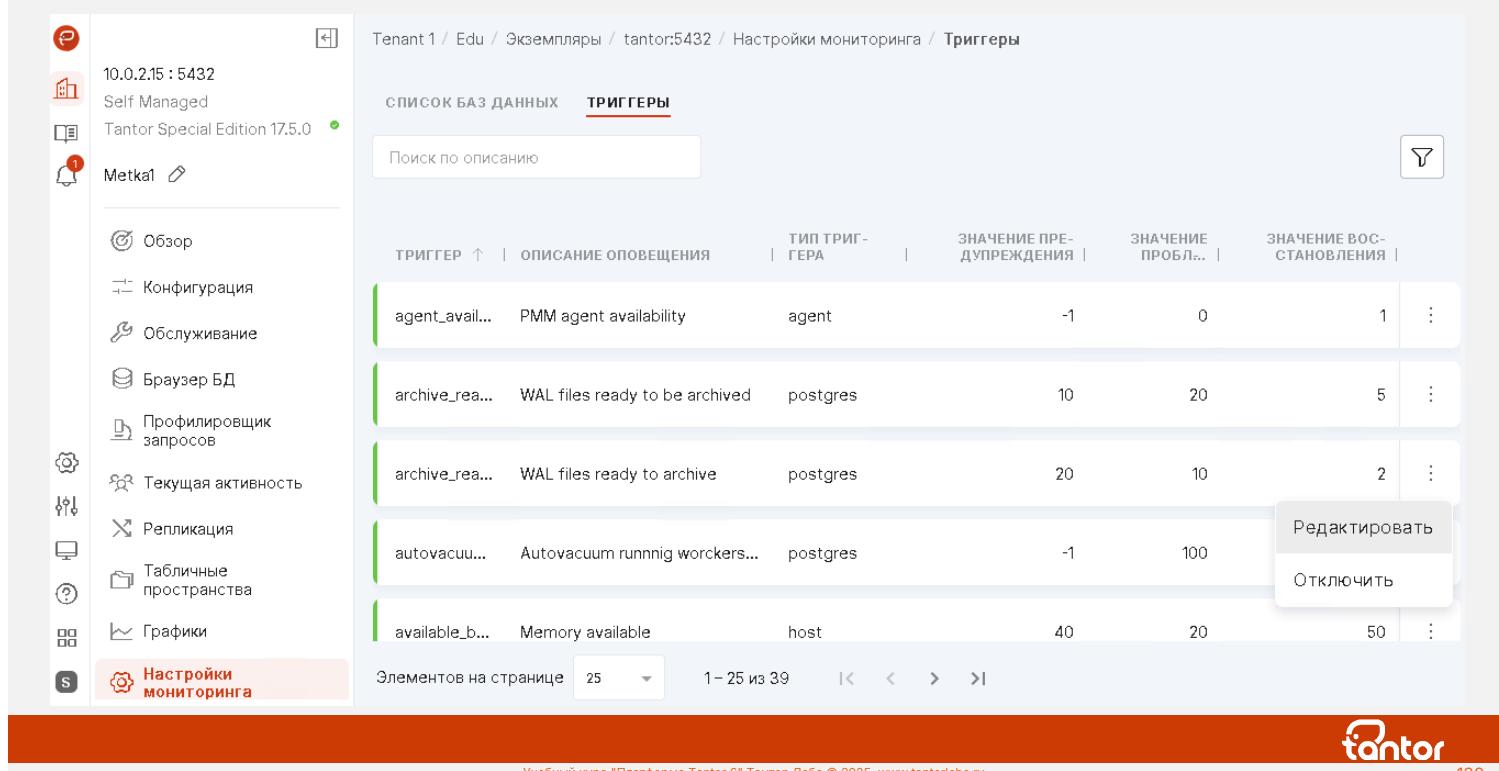
ОТСЛЕЖИВАЕТСЯ	БАЗА ДАННЫХ	ДОСТУПНОСТЬ	РАЗМЕР БАЗЫ ДАННЫХ	АГЕНТ ИМЕЕТ ДОСТУП?
<input checked="" type="checkbox"/>	postgres	Доступна	408.7 MB	Да
<input checked="" type="checkbox"/>	test_db	Доступна	7.5 MB	Да

Учебный курс "Платформа Tantor 6" Тантор Лабс © 2025 www.tantolabs.ru

Экземпляр: Настройки мониторинга

На странице "Настройки мониторинга" экземпляра выбираются базы данных, по которым Агент Платформы собирает метрики. По умолчанию, выбрана только одна база данных, имя которой указывалось для подсоединения Агента при добавление экземпляра в Платформу. Отключить мониторинг этой базы данных (убрать галочку в столбце "Отслеживается") нельзя.

Экземпляр: Настройки мониторинга -> ТРИГГЕРЫ



The screenshot shows the 'Monitoring Configuration' section of the Tantor 6 interface. On the left, a sidebar lists various monitoring categories: Overview, Configuration, Maintenance, Browser DB, Profiler, Current Activity, Replication, Tablespaces, and Graphs. The 'Monitoring Configuration' icon is highlighted. The main area is titled 'СПИСОК БАЗ ДАННЫХ' (List of databases) and 'ТРИГГЕРЫ' (Triggers). It includes a search bar and a table with the following data:

ТРИГГЕР	ОПИСАНИЕ ОПОВЕЩЕНИЯ	ТИП ТРИГГЕРА	ЗНАЧЕНИЕ ПРЕДУПРЕЖДЕНИЯ	ЗНАЧЕНИЕ ПРОБЛ...	ЗНАЧЕНИЕ ВОССТАНОВЛЕНИЯ
agent_avail...	PMM agent availability	agent	-1	0	1
archive_rea...	WAL files ready to be archived	postgres	10	20	5
archive_rea...	WAL files ready to archive	postgres	20	10	2
autovacuu...	Autovacuum running workers...	postgres	-1	100	50
available_b...	Memory available	host	40	20	50

A context menu is open over the fourth row (autovacuum trigger), showing options 'Редактировать' (Edit) and 'Отключить' (Disable). The bottom of the screen shows navigation buttons and a footer with the Tantor logo and copyright information.

Экземпляр: Настройки мониторинга -> ТРИГГЕРЫ

На второй вкладке "ТРИГГЕРЫ" отображается список метрик с их пороговыми значениями. Если собранные Агентом метрики выходят за границы значений триггеров, создаётся оповещение.

Список метрик не меняется. У части метрик нельзя менять пороговые значения. У метрик, у которых можно поменять пороговые значения отображается пункт "Редактировать" во всплывающем меню. Меню вызывается нажатием на три точки справа.

Если триггер редактируемый, то обычно, даётся на выбор три набора пороговых значений. Числа в этих пороговых значениях нельзя менять.

https://docs.tantorlabs.ru/tp/6.1/instances/monitoring_config.html

Экземпляр: Профилировщик запросов

The screenshot shows the Tantor Query Profiler interface. On the left is a sidebar with various monitoring and configuration options. The main area features a line graph titled 'Среднее время запроса' (Average query time) showing fluctuations over time. Below the graph is a search bar and a sorting dropdown. A table lists the top queries with columns for Hash, Database/User, Total Time, Calls, Average Time, Rows, and Blocks. Each query row includes a detailed view button. The bottom of the interface has pagination controls and a footer with copyright information.

ХЭШ ЗАПРОСА	БАЗА ДАННЫХ/ПОЛЬЗОВАТЕЛЬ	ОБЩЕЕ ВРЕМЯ	ВЫЗОВЫ	СРЕДНЕЕ ВРЕМЯ	СТРОКИ	ЗАПИСАНО ВРЕМЕННЫХ БЛОКОВ	ТРЕНД
+ 7e0438aa...	postgres/pma_user	00:00:00	31	15.258	0	0	
+ b4d6d44d...	postgres/pma_user	00:00:00	31	9.677	62	0	

Экземпляр: Профилировщик запросов

"Профилировщик запросов" (Query Profiler) предназначен для профилирования (оценки процесса выполнения) запросов. Профилировщик использует статистику расширения `pg_stat_statements`. Данные из представления `pg_stat_statements` группируются, чтобы не было дубликатов одинаковых запросов. На каждой итерации сбора статистики делается выборка только 50 самых длительных запросов.

В столбце ТРЕНД показывается уменьшенный и сжатый график, который отобразится на основной панели. Чем чаще и дольше выполнялся запрос, тем больше кривая ТРЕНД соответствует графику. Для длительности 1 миллисекунда (как в примере на слайде) ТРЕНД не похож на график, по которому он рисуется.

В меню "три точки" у каждого запроса всего один пункт: "Подробнее". Этот пункт открывает то же самое, как если кликнуть на саму строчку запроса.

При нажатии на значок "+" в левой части, строка "раскроется" и покажет запрос, а значок заменится на "-":

The screenshot shows a detailed view of a query from the previous list. The top part is identical to the previous screenshot. Below it, the query text is displayed in a code editor-like interface with syntax highlighting. The bottom part shows the same table and pagination controls.

```
SELECT
  substring(pgs.query, $3, $4) AS query,
  p.plan,
  p.calls AS plan_calls,
  pgs.calls AS stmt_calls,
  p.total_time,
  p.rows,
  p.shared_kills
```

Экземпляр: Профилировщик запросов: СТАТИСТИКА

10.0.2.15 : 5432
Self Managed
Tantor Special Edition 17.5.0

Metab1

Обзор

Конфигурация

Обслуживание

Браузер БД

Профилировщик запросов

Текущая активность

Репликация

Табличные пространства

Графики

Настройки мониторинга

Расширенная аналитика

Мониторинг бекапов

Задачи

Статистика

Планы

TIME QUERY/SECOND

0.0007
0.0006
0.0005
0.0004
0.0003
0.0002
0.0001
0

20/09 17:39 20/09 17:45 20/09 17:50 20/09 17:55 20/09 18:00 20/09 18:05 20/09 18:08

Хэш запроса: 7e0438aa9db9d1d18164bf8b6899ed2c

База данных: postgres

Пользователь: rma_user

Общее время (HH:MM:SS): 00:00:00

Вызовы: 30

Среднее время: 15.467 ms

Возвращенные строки: 0

Время ЦПУ: 0.464 sec

Время I/O: 0 sec

Прочитано временных блоков: 0 blocks

Записать временные блоки: 0 blocks

Текст запроса:

```
SELECT
  substring(pgs.query, $3, $4) AS query,
  p.plan,
  p.calls AS plan_calls,
  pgs.calls AS stmt_calls,
  p.total_time,
  p.rows,
  p.shared_blk_hit,
  p.shared_blk_read,
  p.shared_blk_dirtied,
  p.shared_blk_written,
  p.local_blk_hit,
  p.local_blk_read,
  p.local_blk_dirtied,
  p.local_blk_written,
```

Последние 30 минут

Последний час

Последние 3 часа

Последние 8 часов

Последние 12 часов

Последний день

Последние 7 дней

От *

Выберите дату

Период *

30 минут

Применить

Экземпляр: Профилировщик запросов: СТАТИСТИКА

При клике на строчку запроса, откроется новая страница на вкладке СТАТИСТИКА по этому запросу.

В левой части страницы - статистика, в правой части - Текст запроса.

И на предыдущей и на этой странице справа вверху можно выбрать интервал отображения собранных данных от 30 минут до 7 дней.

На графике отображается метрика, которая указана слева вверху графика. В примере на слайде: TIME QUERY/SECOND.

Под графиком 7 точек, на которые можно кликать и будут открываться графики метрик:

1. Time Query/Second,
2. Calls/Second,
3. Rows/Second,
4. CPU Time/ Second,
5. IO Time/Second,
6. Dirtied Blocks/Second,
7. Temp Blocks(Write)/Second

При клике на вкладке "ПЛАНЫ" откроется страница со списком планов.

Для того, чтобы планы показывались, а не выдавалось "Данные не найдены", нужно, чтобы

1) в кластере были установлены библиотеки pg_stat_statements и pg_store_plans:

```
alter system set shared_preload_libraries = 'pg_stat_statements, pg_qualstats, pg_store_plans, pg_prewarm, pg_stat_kcache, auto_explain';
```

2) в базе данных были установлены расширения:

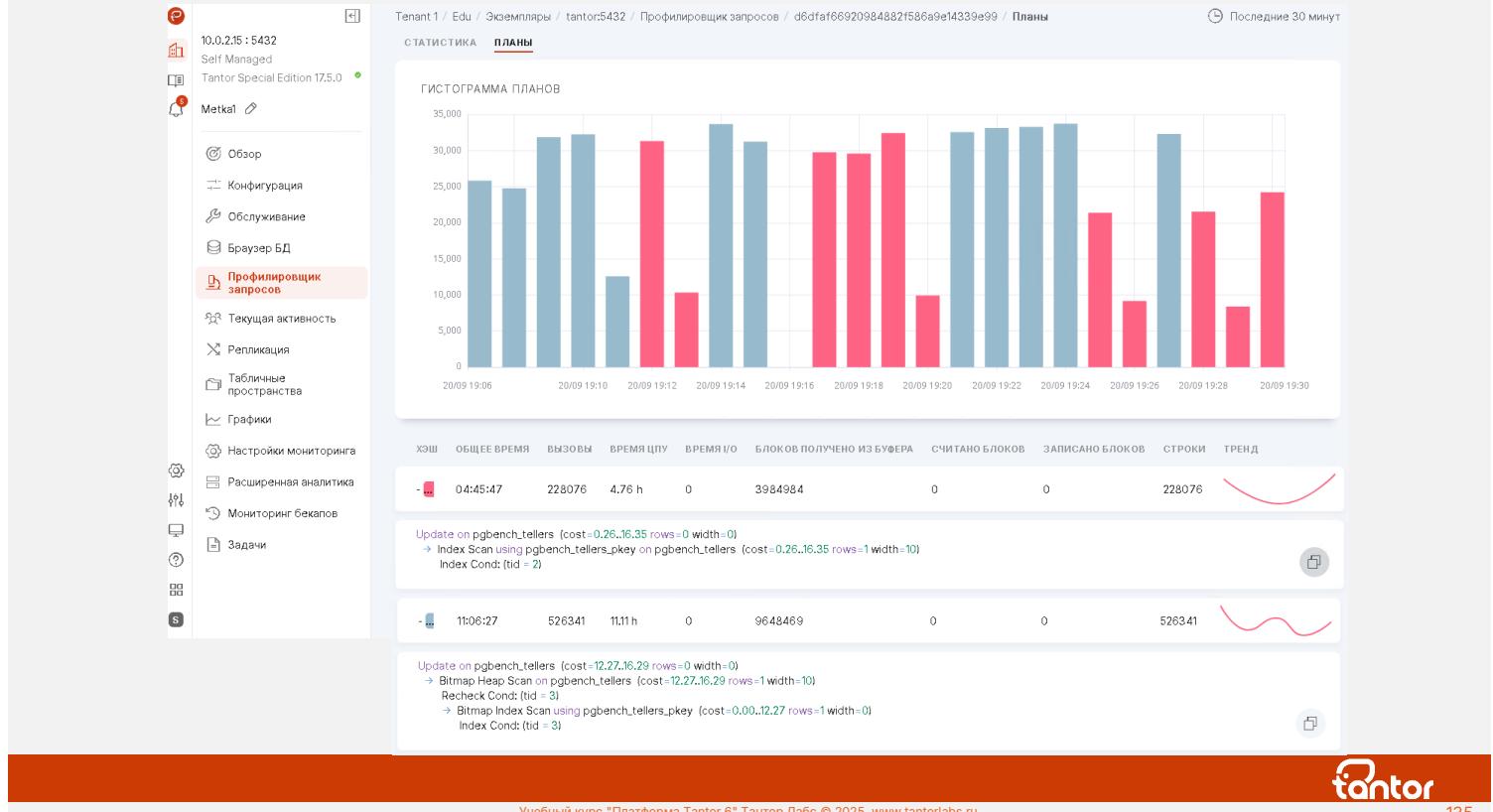
```
create extension if not exists pg_stat_statements;
create extension if not exists pg_store_plans;
```

3) показываться будут только запросы, длительность которых превышает значение параметра pg_store_plans.min_duration. Значение можно установить командой:

```
alter system set pg_store_plans.min_duration = '10s';
```

Цель профилировщика - проведение анализа и выявление проблемных запросов в базе данных. Профилировщик не только отслеживает параметры выполнения, но и проводит анализ планов запросов, предоставляя подсказки для оптимизации.

Экземпляр: Профилировщик запросов: ПЛАНЫ



Учебный курс "Платформа Tantor 6" Тантор Лабс © 2025 www.tantolabs.ru

tantor

135

Экземпляр: Профилировщик запросов: ПЛАНЫ

При клике на вкладке "ПЛАНЫ" откроется страница ГИСТОГРАММА ПЛАНОВ. На гистограмме разными цветами показываются разные планы.

На временной шкале открывается гистограмма, которая показывает с какими планами выполнялся выбранный запрос за указанное время.

Один и тот же запрос может выполняться разными планами.

Под гистограммой можно кликнуть на три точки, подсвеченные цветом столбцов гистограммы и появится план выполнения.

В примере видны два разных плана для запроса

UPDATE pgbench_tellers SET tbalance = tbalance + \$1 WHERE tid = \$2

Запрос на этой странице не виден, он будет виден на вкладке СТАТИСТИКА.

Число столбцов гистограммы зависит от частоты выполнения запроса.

При нажатии на значок с двумя квадратиками справа от плана, план копируется в буфер обмена.

Экземпляр: Профилировщик запросов: простой план

Экземпляр: Профилировщик запросов: простой план

При клике на любое место строки плана, кроме трёх точек, откроется страница с деталями плана выполнения запроса.

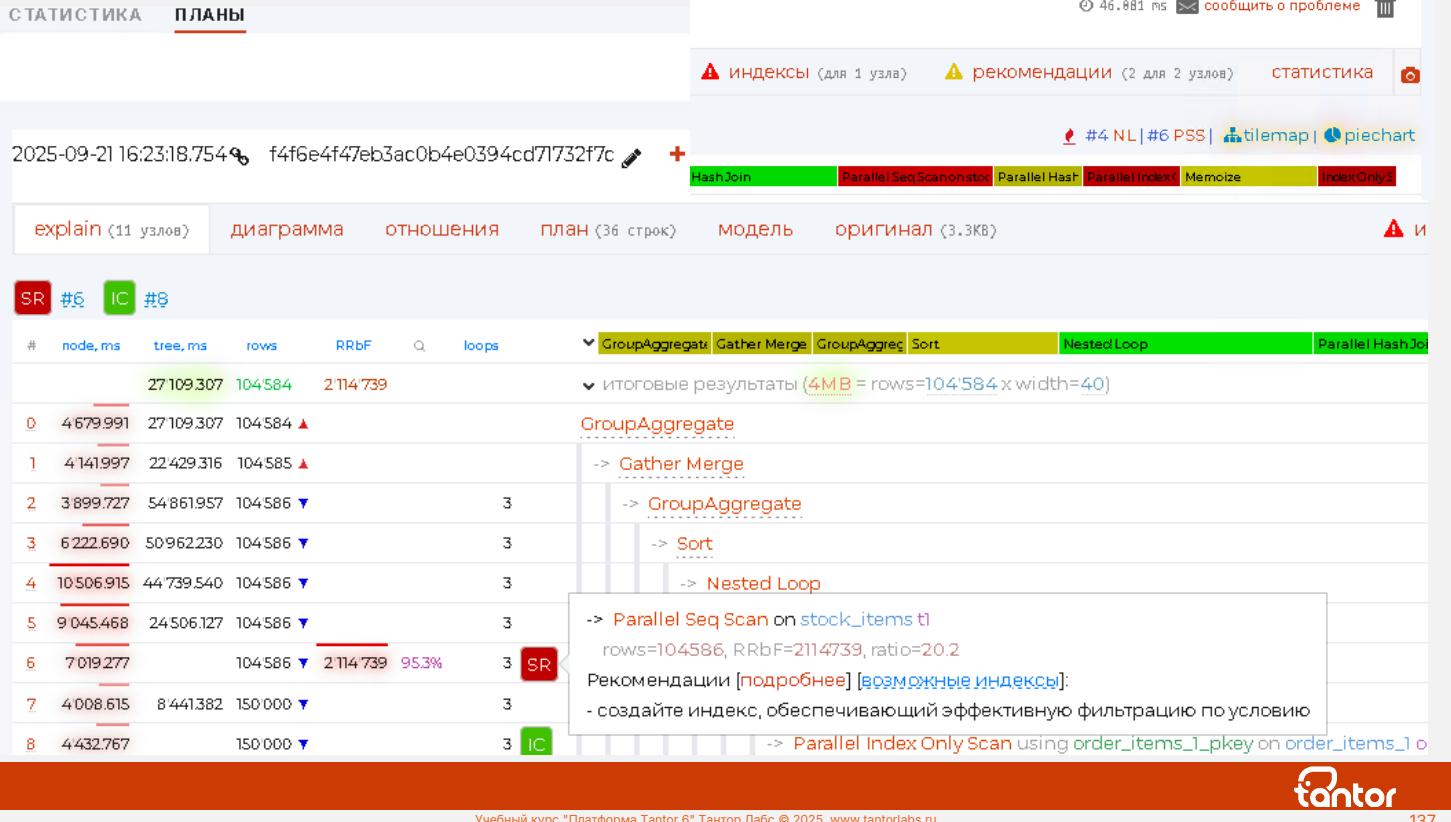
На странице 6 вкладок:

explain
диаграмма
отношения
план
модель
оригинал

В примере на слайде показан план запроса, который не является долго выполняющимся (pg_store_plans.min_duration), по нему не собраны данные и рекомендаций нет. На картинке нет многих столбцов, которые появляются у долгих запросов.

https://docs.tantorlabs.ru/tp/6.1/instances/query_profiler.html

Экземпляр: Профилировщик запросов: долгий запрос



The screenshot shows a detailed query plan for a long-running query. The plan is visualized as a tree structure with various stages: GroupAggregate, Gather Merge, GroupAggregate, Sort, Nested Loop, Parallel Seq Scan, and Index Only Scan. The 'Parallel Hash Joins' tab is selected. A tooltip provides recommendations for creating an index on the 'order_items_1_pkey' column to improve filtering. The interface includes a navigation bar with tabs like 'explain', 'диаграмма', 'отношения', 'план', 'модель', 'оригинал', and 'статистика'.

Экземпляр: Профилировщик запросов: долгий запрос

Для долгих запросов картинка имеет большое число столбцов.

Для создания тестовой нагрузки в виртуальной машине курса можно использовать команды:

`psql -f /opt/workload/custom_init.sql`

`pgbench -T 60000 -P 10 -c 5 -f custom_workload.sql`

При проведении мышью над большинством визуальных элементов страницы появляются всплывающие окна с подсказками.

Во врезке на слайде - продолжение заголовка таблицы, где справа есть ссылки на tilemap и piechart.

Страница создаётся анализатором планов запросов, аналогичном [explain.tensor.ru](#). Анализатор интегрирован в Платформу Tantor и позволяет проводить анализ в рамках Платформы **без отправки** запросов и данных во внешние сервисы. Платформа Tantor самостоятельно собирает и хранит запросы во внутренней базе данных инсталляции Платформы.

Сервис позволяет:

Визуализировать любые планы запросов PostgreSQL, скопированные из любых источников и вставленные в окно, которое открывается из меню "Модули"→"Анализатор плана запросов"

Отображать планы запросов, собранных опцией "Расширенная аналитика" платформы Tantor и показываемые в меню "Проблемные запросы".

Отображать планы запросов, собранных платформой Tantor с помощью **расширения pg_store_plans** на странице "Профилировщик запросов".

при включенном параметре `track_io_timing` в плане отображаются показатели времени, затраченного на операции ввода-вывода (average IO после "итоговые результаты").

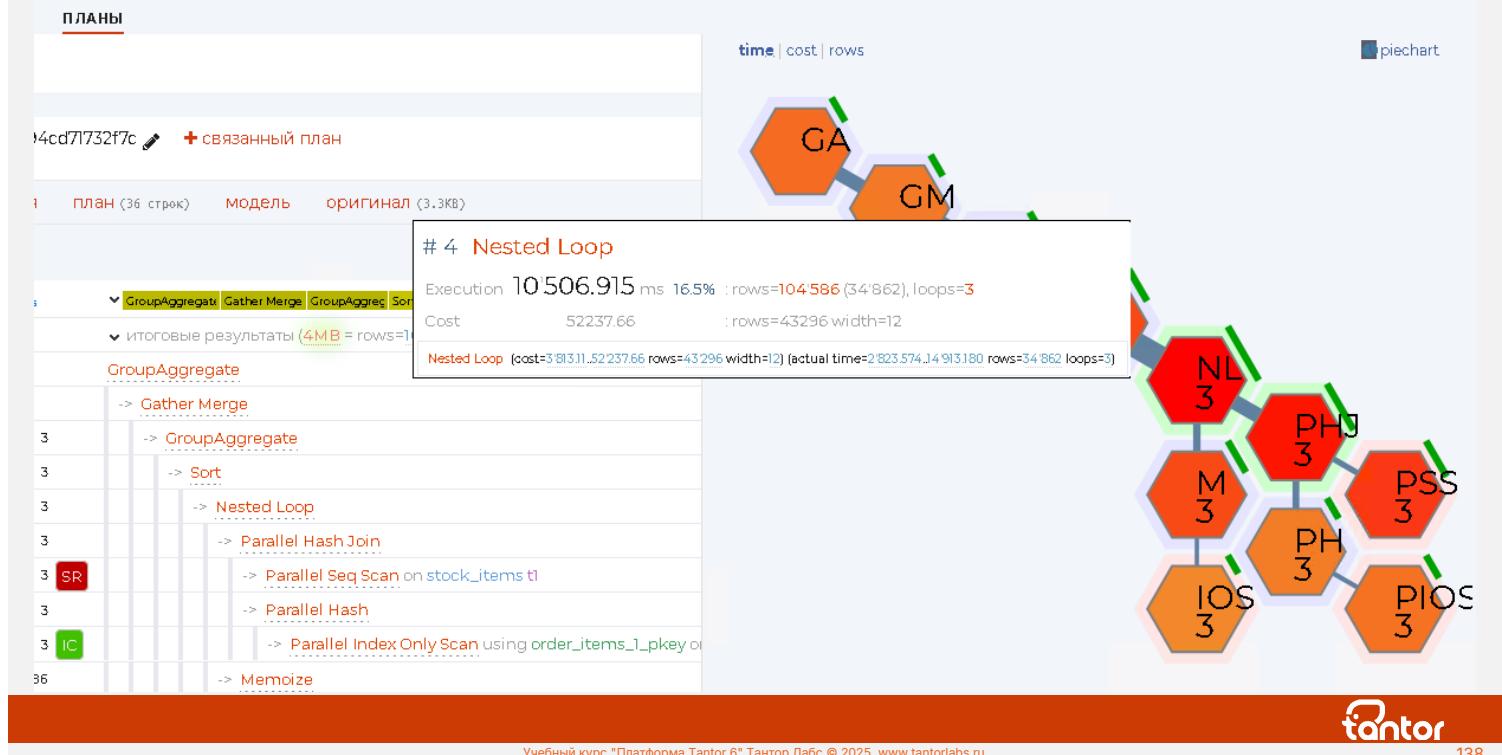
Статьи с описанием визуального интерфейса Tensor:

<https://habr.com/ru/company/tensor/blog/677948/>

<https://habr.com/ru/company/tensor/blog/477624/>

Экземпляр: Профилировщик запросов: tilemap

/ Экземпляры / tantor:5432 / Профилировщик запросов / 1689e981d413fc3061f551d1d9ccee23 / Планы / 11cb655f9b242e88378da2a48e181c91



Экземпляр: Профилировщик запросов: tilemap

Пример tilemap, который появляется справа на плане.

На диаграмме сверху 3 вкладки: time (открыта по умолчанию), cost и rows.

Режим rows (ссылка "rows" сверху над tilemap) позволяет мгновенно оценить, в каком сегменте плана генерируется или фильтруется чересчур много записей.

Более насыщенным (красным) цветом подсвечиваются узлы, на которых было отфильтровано (Rows Removed by Filter ...) наибольшее число строк.

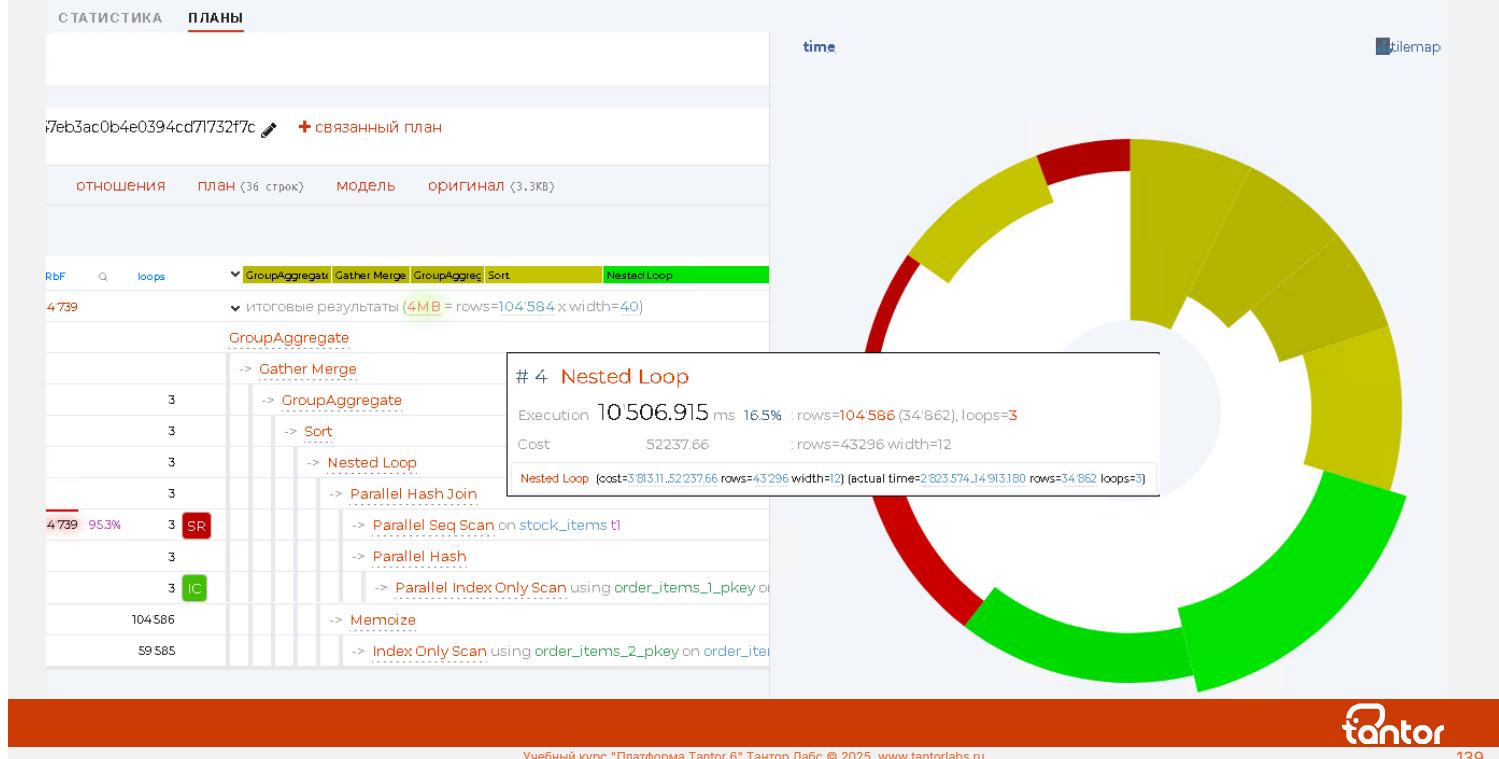
Визуальная ширина линий между шестиугольниками пропорциональна числу строк, которые были переданы вверх по дереву плана.

Чем ярче узел и толще его линии (связи, ветви), тем он более важен и на него стоит обратить внимание.

При проведении мышью над шестиугольниками показываются части плана, к которым эти элементы относятся. В примере - шестиугольник NL3 (Nested Loop).

Экземпляр: Профилировщик запросов: piechart

Tenant 1 / Edu / Экземпляры / tantor:5432 / Профилировщик запросов / 1689e981d413fc3061f551d1d9cc6e23 / Планы / 11cb655f9b242e88378da2a48e181c91



Экземпляр: Профилировщик запросов: piechart

Пример piechart (круговая диаграмма), который появляется справа на плане.

Круговая диаграмма, представляет долю времени, занимаемую узлом плана запроса по отношению к длительности выполнения запроса.

При проведении мышью над элементами показываются части плана, к которым эти элементы относятся. В примере - зелёный сектор.

Круговая диаграмма полезна, когда план запроса содержит много узлов и даже сокращенная форма плана занимает большую площадь. Она позволяет обратить внимание на узлы плана, на выполнение которых тратится больше всего времени.

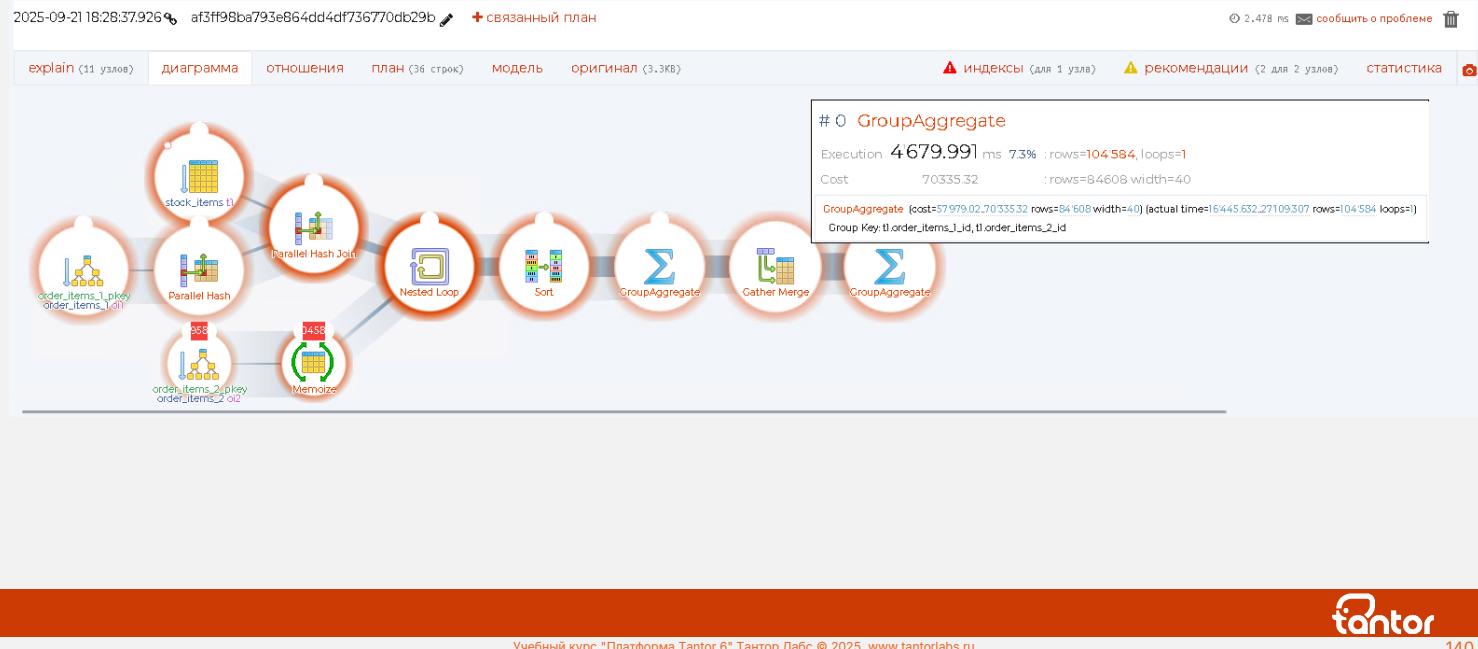
Цвет на диаграмме и на полоске заголовков столбцов:

если доминируют красные сегменты, то тратится много времени на чтение данных (Seq Scan, Index Scan, Bitmap Heap Scan), если желтые - на обработку (Aggregate, Unique), а зелёные - это соединения (Join).

Экземпляр: Профилировщик запросов: диаграмма

Tenant 1 / Edu / Экземпляры / tantor:5432 / Профилировщик запросов / 1689e981d413fc3061f551d1d9cc0e23 / Планы / 11cb655f9b242e88378da2a48e181c91

СТАТИСТИКА ПЛАНЫ



Экземпляр: Профилировщик запросов: диаграмма

Piechart (круговая диаграмма) не показывают полную цепочку вложений подузлов CTE/InitPlain/SubPlan - их можно увидеть только на **диаграмме выполнения**.

Экземпляр: Профилировщик запросов: рекомендации

диаграмма отношения план (36 строк) модель оригинал (3.3КБ) ▲ индексы (для 1 узла) ▲ рекомендации (2 для 2 узлов)

Сводно

6 Parallel Seq Scan on stock_items t1

Execution **7'019.277** ms 11.0% : rows=104 586 (34'862) RRB=2'114'739 (704'913), loops=3

Cost 31250.78 : rows=43296 width=12

Parallel Seq Scan on stock_items t1 (cost=0.00,31'250.78 rows=43296 width=12) (actual time=0.669.2339.759 rows=34'862 loops=3)

Filter: ((optcounter > 14'277) AND (optcounter < 15'220))

Rows Removed by Filter: 704'913

IC SR

8 Parallel Index Only Scan using order_items_1_pkey on order_items_1 oi1

Execution **4'432.767** ms 7.0% : rows=150 000 (50'000), loops=3

Cost 3031.43 : rows=62500 width=8

Parallel Index Only Scan using order_items_1_pkey on order_items_1 oi1 (cost=0.42,3'031.43 rows=62500 width=8) (actual time=0.058.1477.589 rows=50'000 loops=3)

Heap Fetches: 0

IC SR

IC -> Parallel Index Only Scan using order_items_1_pkey on order_items_1 oi1

rows=150000

Рекомендации [подробнее]:

- обратите внимание, что индекс используется только для упорядоченного **чтения, без ограничений по ключу**

IC

Экземпляр: Профилировщик запросов: рекомендации

Иконки рекомендаций в виде прямоугольников (IC - Index Create, SR - Service Recommendation) присутствуют на плане. При наведении на иконки открываются всплывающие окна с рекомендациями.

Также открыть страницу рекомендаций можно кликнув на ссылки "индексы" и "рекомендации". В рекомендации по индексам даётся команда создания индекса, которую можно скопировать в буфер обмена:

диаграмма отношения план (36 строк) модель оригинал (3.3КБ) ▲ индексы (для 1 узла)

Сводно

6 Parallel Seq Scan on stock_items t1

Execution **7'019.277** ms 11.0% : rows=104 586 (34'862) RRB=2'114'739 (704'913), loops=3

Cost 31250.78 : rows=43296 width=12

Parallel Seq Scan on stock_items t1 (cost=0.00,31'250.78 rows=43296 width=12) (actual time=0.669.2339.759 rows=34'862 loops=3)

Filter: ((optcounter > 14'277) AND (optcounter < 15'220))

Rows Removed by Filter: 704'913

stock_items

IC -- optcounter :: integer(>,<)
CREATE INDEX CONCURRENTLY "stock_items-f75b2a99"
ON stock_items(optcounter);

IC

Экземпляр: Профилировщик запросов: статистика

СТАТИСТИКА ПЛАНЫ

2025-09-21 18:28:37.926 af3ff98ba793e864dd4df736770db29b Связанный план

2.478 ms сообщить о проблеме

explain (11 узлов)	диаграмма	отношения	план (36 строк)	модель	оригинал (3.3Кб)	индексы (для 1 узла)	рекомендации (2 для 2 узлов)	статистика	изменить
время	%	rows	RRbF	%	loops	тип узла	таблица	индекс	
10506915	16.5	104586			3				4
9045468	14.2	104586			3				5
8579718	13.5	209170			4				6
7019277	11.0	104586	2114739	95.3	3		stock_items		7
6222690	9.8	104586			3				8
6151398	9.7	104586			104586				9
4432767	7.0	150000			3		order_items_1	order_items_1_pkey	10
4141997	6.5	104585							11
4008615	6.3	150000			3				12
3575100	5.6	59585			59585		order_items_2	order_items_2_pkey	

6 Parallel Seq Scan on stock_items t1

Execution 7'019.277 ms 11.0% :rows=104586 (34.862) RRbF=2'114.739 (704.913), loops=3

Cost 31250.78 :rows=43296 width=12

Parallel Seq Scan on stock_items t1 (cost=0.00..31250.78 rows=43296 width=12) (actual time=0.669..2339.759 rows=34862 loops=3)
Filter: (optcounter > 14277) AND (optcounter < 15220)
Rows Removed by Filter: 704.913

142

Учебный курс "Платформа Tantor 6" Тантор Лабс © 2025 www.tantorlabs.ru

Экземпляр: Профилировщик запросов: статистика

При клике справа вверху "статистика", откроется страница с показателями выполнения: временем, числом строк, RRbF (Rows Removed by Filter), типом узла, таблицей, индексом и ссылкой на номера узлов плана. При наведении на ссылку с номером узла плана появится всплывающее окно с частью плана.



8-2

Расширенная аналитика

Экземпляр: Расширенная аналитика

The screenshot shows the Tantor 6 platform interface. On the left, there is a sidebar with various icons and a list of items: Metka1, Графики, Настройки мониторинга, Расширенная аналитика (which is highlighted in red), Мониторинг бекапов, and Задачи. The main content area is titled 'ЭКЗЕМПЛЯРЫ' and shows a table with two rows of data. The columns are: ИМЯ УЗЛА, ОС, ТИП, ИЗДАНИЕ, ВЕРСИЯ РБ, and ЦПУ. The data is as follows:

ИМЯ УЗЛА	ОС	ТИП	ИЗДАНИЕ	ВЕРСИЯ РБ	ЦПУ
tantor	Astra Linux(1.7_x86-64)	Tantor	Tantor Special Edition	17.5.0	4
tantor	Astra Linux(1.7_x86-64)	Tantor	Tantor Special Edition	17.5.0	4

On the right, there is a context menu with the following options: Открыть (Open), Открыть расширенную аналитику (Open extended analytics), Тестовое подключение (Test connection), Переместить в другое пространство (Move to another space), Добавить задачу (Add task), Редактировать метку (Edit tag), Браузер БД (DB browser), and Удалить (Delete). At the bottom right of the main content area, there is a table titled 'PSA' with two rows of data:

PID	dbname	backend	xact	host / apname	client	changed	query	state
1004	postgres	2025-09-19 16h 04m 59s 615.427ms	eebad04b504e [136]	127.0.0.1	16h 04m 59s 615.331ms	16h 04m 59s 615.336ms	active	SE
1005	postgres	2025-09-19 21:00:34.826515	0712:41384573 pg-monitor.pgstat	127.0.0.1	07:12:41.384669	07:12:41.384664	idle	

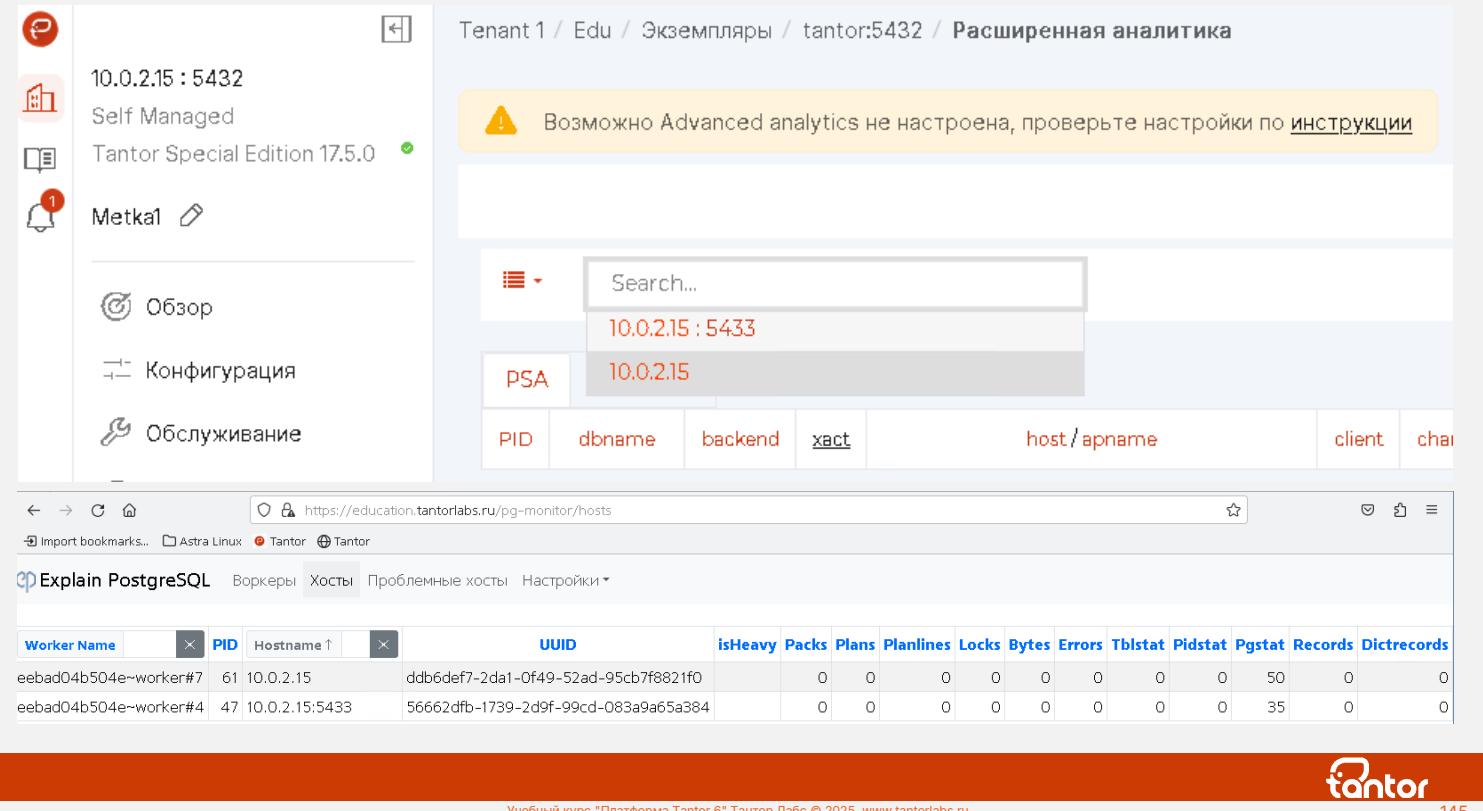
At the bottom of the interface, there is a footer bar with the Tantor logo and the text 'Учебный курс "Платформа Tantor 6" Тантор Лабс © 2025 www.tantolabs.ru' and a page number '144'.

Экземпляр: Расширенная аналитика

Открыть страницу "Расширенная аналитика" можно:

1. на странице Экземпляры из выпадающего меню, которое открывается при нажатии на значок три точки справа.
 2. Открыв экземпляр и выбрав в меню Экземпляра пункт "Расширенная аналитика"
- В примере на слайде на странице есть полоса прокрутки. Полоса появилась потому, что страница имеет большую длину.

Экземпляр: Проверка настройки аналитики



Worker Name	PID	Hostname	UUID	isHeavy	Packs	Plans	Planlines	Locks	Bytes	Errors	Tblstat	Pidstat	Pgstat	Records	Dictrecords
eebad04b504e~worker#7	61	10.0.2.15	ddb6def7-2da1-0f49-52ad-95cb7f8821f0	0	0	0	0	0	0	0	0	0	50	0	0
eebad04b504e~worker#4	47	10.0.2.15:5433	56662dfb-1739-2d9f-99cd-083a9a65a384	0	0	0	0	0	0	0	0	0	35	0	0

Экземпляр: Проверка настройки аналитики

Расширенная аналитика читает текстовые файлы диагностических логов кластера баз данных PostgreSQL, разбирает их содержимое и загружает строки в базу данных Платформы. Если загрузка не выполнена или в журнале нет данных, то на страницах аналитики не будет данных.

Этим занимается служба "pg-monitor", которая работает в докер-контейнерах

pg_monitor_dispatcher и **pg_monitor_collector**. Служба использует соединение по протоколу ssh от хоста Платформы до хоста с экземплярами PostgreSQL. Для этого нужна дополнительная ручная настройка клиента ssh.

Для проверки того, что журнал читается, можно зайти на страницу аналитики, кликнуть слева вверху страницы на IP адрес. Появится список IP-адресов экземпляров, которые зарегистрированы в Агенте хоста. Порт по умолчанию 5432 не отображается. Нестандартный порт 5432 отображается после двоеточия. Признака реплика это или мастер нет.

Если после клика на IP-адрес ниже поля поиска "Search..." выдаётся строка на жёлтом фоне, то диагностический файл лога не был прочитан, нужно проверить по разделу документации настроены ли ключи ssh на хосте Платформы и скопированы ли они на хост Агента; параметры конфигурации мониторируемого кластера PostgreSQL.

Вторая проверка - открыть адрес https://хост_Платформы/pg-monitor/hosts

Если в столбце Hostname нет адреса экземпляра, то Расширенная аналитика не работает, так как нет рабочего процесса, который собирает аналитику с экземпляра. Скорее всего не работает контейнер **pg_monitor_dispatcher**, в котором работают эти процессы. Можно запустить контейнеры:

```
docker start pg_monitor_dispatcher
```

```
docker start pg_monitor_collector
```

и проверить их статус:

```
docker ps -a | grep monitor | grep Up
```

Экземпляр: страница pg-monitor

The screenshot shows the Explain PostgreSQL pg-monitor interface. The main content area displays a table of worker statistics for processes with PIDs 32, 33, 44, 45, 47, 49, and 61. The table includes columns for Name, Pid, Tasks, Heavy, Plans, Planlines, Packs, Heartbeat, CPU Busy, Locks, Bytes, Errors, Tblstat, Pidstat, Pgstat, Records, Dicts, Latency, and Pool status (Status, Queue, Size). The workers are mostly idle (0 tasks, 0 heavy, 0 plans, 0 planlines, 0 packs). The heartbeat column shows times ranging from 6 ms to 1681 ms. The CPU Busy column is mostly at 0%. The Locks, Bytes, Errors, and Tblstat columns are mostly at 0. The Pidstat, Pgstat, Records, Dicts, and Latency columns are mostly at 0. The Pool status is mostly OFF. The URL in the address bar is https://education.tantorlabs.ru/pg-monitor/cleardb/.

Explain PostgreSQL Воркеры Хосты Проблемные хосты Очистка БД

Name	Pid	Tasks	Heavy	Plans	Planlines	Packs	Heartbeat	CPU Busy	Locks	Bytes	Errors	Tblstat	Pidstat	Pgstat	Records	Dicts	Latency	Pool		
																		Status	Queue	Size
eebad04b504e~worker#0	32	0	0	0	0	0	1681 ms	0 %	0	0	0	0	0	0	0	0	3.27 ms	OFF	0	0
eebad04b504e~worker#1	33	0	0	0	0	0	844 ms	0 %	0	0	0	0	0	0	0	0	3.48 ms	OFF	0	0
eebad04b504e~worker#2	44	0	0	0	0	0	1259 ms	0 %	0	0	0	0	0	0	0	0	3.98 ms	OFF	0	0
eebad04b504e~worker#3	45	0	0	0	0	0	1226 ms	0 %	0	0	0	0	0	0	0	0	4.02 ms	OFF	0	0
eebad04b504e~worker#4	47	1	0	0	0	0	6 ms	0 %	0	0	0	0	0	35	0	0	4.25 ms	OFF	0	0
eebad04b504e~worker#5	49	0	0	0	0	0	1244 ms	0 %	0	0	0	0	0	0	0	0	3.13 ms	OFF	0	0
eebad04b504e~worker#6	55	← → C ⌂	🔒	https://education.tantorlabs.ru/pg-monitor/cleardb/														OFF	0	0
eebad04b504e~worker#7	61	Import bookmarks...	Astra Linux	Tantor	Tantor													OFF	0	0

Explain PostgreSQL Воркеры Хосты Проблемные хосты Очистка БД

Частота проверки: 1 час

По свободному месту По глубине архива

Оставлять не менее: %

Полных дней: 8

Reset Save

Экземпляр: страница pg-monitor

Страница доступна по адресу https://хост_Платформы/pg-monitor

На вкладке "очистка БД" можно настроить условия очистки накопленных Платформой данных по метрикам мониторируемых экземпляров. По умолчанию 7 дней.
На вкладке "Воркеры" отображается статистика рабочих процессов.

Экземпляр: Расширенная аналитика

10.0.2.15 : 5432
Self Managed
Tantor Special Edition 17.5.0
Метка не заполнена

10.0.2.15 : 5432 / Edu / Экземпляры / tantor:5432 / Расширенная аналитика / Расширенная аналитика

21.09.2025 Логи

Search...
10.0.2.15 : 5432
PSA 10.0.2.15
17:02:20 17:02:20 укажите фильтр... x

результаты	PID	dbname	backend	xact	query	state	query
130	8	idle = 2 idle in transaction = 1 wait	33ms 3h 01m 04s 743.917ms pg_bench	1m 04s 743.917ms active	UPDATE stock_items SET descr = 'updated at ' now(), amount = amount - 14:0115.256.083		
46	46	idle	1m 01s 256.083ms pg_bench	3h 6s 424.139ms active	select order_items_1_id, order_items_2_id, max(amount) as max_amount 14:0213.575.861		
57	57	idle	1m 01s 256.083ms pg_bench	3h 6s 424.139ms wait	select order_items_1_id, order_items_2_id, max(amount) as max_amount 14:0213.575.861 IPC : MessageQueueSend		
Блокировки	1	deadlock	25ms 1m 01s 256.083ms pg_bench	3h 6s 424.139ms active	select order_items_1_id, order_items_2_id, max(amount) as max_amount 14:0213.575.861		
	6	lock	511ms 1m 01s 256.083ms pg_bench	3h 6s 424.139ms wait	select order_items_1_id, order_items_2_id, max(amount) as max_amount 14:0213.575.861 IPC : MessageQueueSend		
Ошибки	1	FATAL	35ms 1m 01s 256.083ms pg_bench	3h 6s 424.139ms active	select order_items_1_id, order_items_2_id, max(amount) as max_amount 14:0213.575.861		
	1	ERROR	38ms 1m 01s 256.083ms pg_bench	1m 59s 995.638ms active	INSERT INTO stock_items (order_items_1_id, optocounter, amount, order_it 14:0220.004362		
		WARNING	1m 59s 995.638ms pg_bench	1m 59s 995.638ms active	UPDATE stock_items SET descr = 'updated at ' now(), amount = amount - 14:0220.329.638		
Системные действия	1	запросы	38ms 1m 01s 256.083ms pg_bench	9m 50s 36.821ms active	select order_items_1_id, order_items_2_id, max(amount) as max_amount 14:0229.963.179		
	1	процессы	65ms 1m 01s 256.083ms pg_bench	9m 50s 36.821ms active	select order_items_1_id, order_items_2_id, max(amount) as max_amount 14:0229.963.179		
Статистика по таблицам	1	ANALYZE	79ms 1m 01s 256.083ms pg_bench	9m 50s 36.821ms active	select order_items_1_id, order_items_2_id, max(amount) as max_amount 14:0229.963.179		
	3	VACUUM	89ms 1m 01s 256.083ms pg_bench	9m 50s 36.821ms active	select order_items_1_id, order_items_2_id, max(amount) as max_amount 14:0229.963.179		
	3	CHECKPOINT	65ms 1m 01s 256.083ms pg_bench	1m 49s 732.001ms idle in transaction	SELECT balance FROM pg_bench_accounts WHERE aid = 39514; 14:0230.267999		
	1	PG config	1m 49s 732.001ms pg_bench				

Экземпляр: Расширенная аналитика

Если кликнуть на IP-адрес, откроется выпадающий список с IP-адресами и портами экземпляров PostgreSQL.

Дальше можно кликнуть слева вверху страницы на значок с тремя полосками (три полоски напоминают бургер, поэтому называют "бургер-меню"), откроется выпадающее меню, относящееся к выбранному экземпляру PostgreSQL.

В выпадающем меню в правой части строк синим цветом местами присутствуют числа. Те строки, где есть числа кликабельны. Пункты меню, где чисел нет - не кликабельны. Чисел нет, если в диагностическом логе экземпляра отсутствуют сообщения, соответствующие пунктам меню или если по любой причине служба pg-monitor не может получить по ssh файл диагностического журнала и разобрать его содержимое.

Выбрав на странице пункт меню, страница будет отображать данные, соответствующие этому пункту, пока не будет выбран другой пункт меню. Даже если в меню экземпляра кликнуть на "Расширенная аналитика", то всё равно будет показываться предыдущий пункт меню страницы.

На части страниц меню Расширенной аналитики отсутствует. На таких страницах для возврата на предыдущие страницы придётся использовать средства браузера (кнопку "Back"/"Назад"). Либо кликать на какой-нибудь пункт меню экземпляра, а потом на пункт меню "Расширенная аналитика".

Мегазапросы - запросы, превышающие объем 1МБ входящего или исходящего трафика и превышающие время `log_min_duration_statement` или `pg_store_plans.min_duration`.

Справа вверху выбирается дата, за которую в меню будут отображаться числа запросов и ошибок. Дата меняется в полночь и в полночь в меню страницы числа исчезнут, пока не появятся запросы, соответствующие пункту меню.

https://docs.tantorlabs.ru/tp/6.1/instances/pg_monitor/pg_m_page_control_panel.html

Расширенная аналитика: ошибки в логе PostgreSQL

The screenshot shows a PostgreSQL log analysis interface. At the top, a sidebar titled 'Errors' lists 'FATAL' (1), 'ERROR', and 'WARNING'. The main area shows a log entry for '10.0.2.15 / FATAL' with the message 'terminating connection due to administrator command'. Below this, there are three tabs: 'by classes', 'by applications', and 'by time'. The 'by time' tab is selected, showing a timeline from 13:42:31 to 13:42:31. The log entry is highlighted in red. The bottom of the interface shows a summary of the error details: Timestamp (2025-09-21 13:42:30.671 +00), Host (10.0.2.15), Database (test_db), PID (3556), Client (127.0.0.1:46009), UUID (a8a00a70-96d7-11f0-987d-2fb26cf75931), Application (eebad04b504e [47] : pg-monitor.tblstat), User (----), vXID (2/0), and txID (----). The bottom right corner shows the date 13:42:31 and a small icon.

Расширенная аналитика: ошибки в логе PostgreSQL

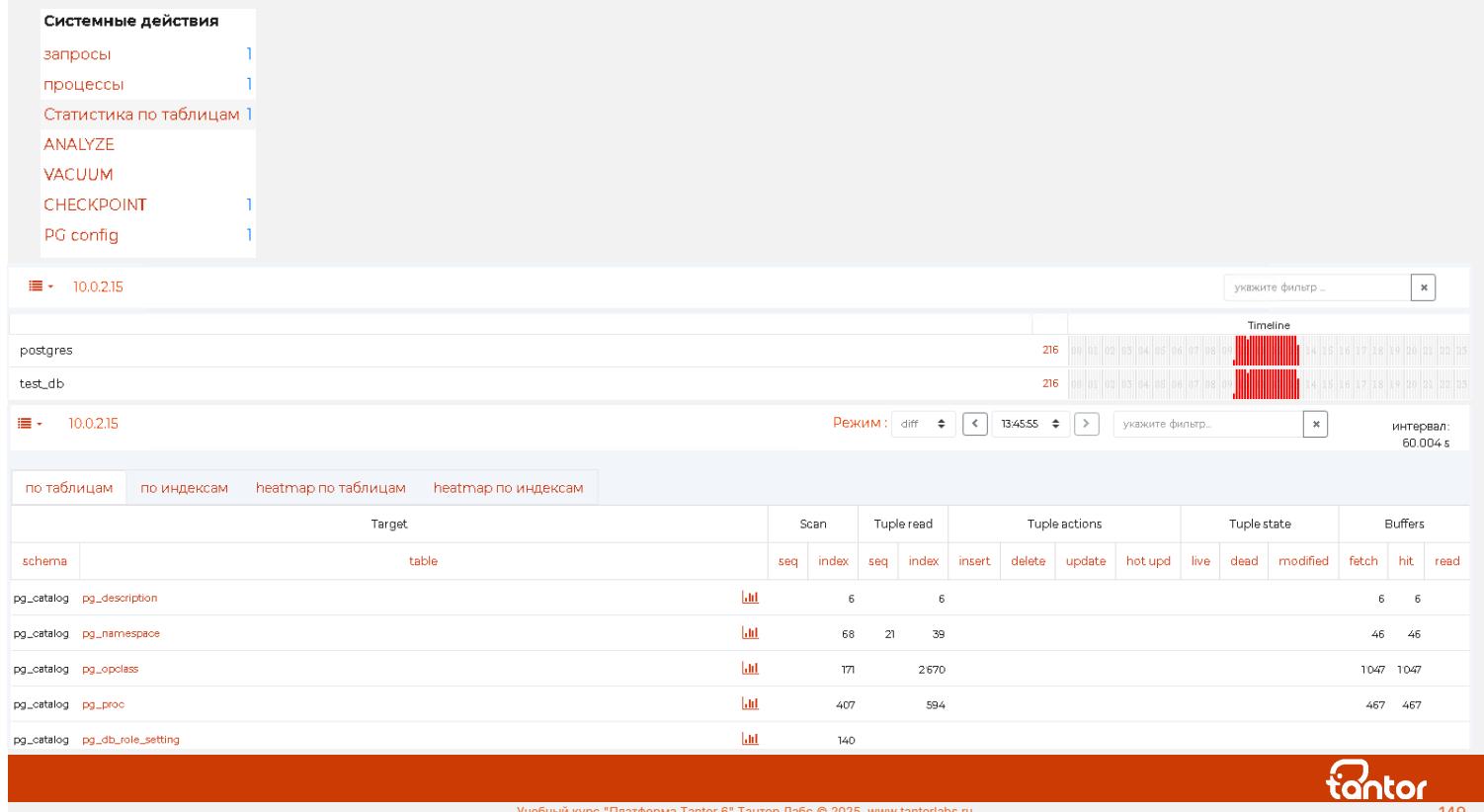
При клике в меню страницы на строку, где число ошибок непусто (на слайде строка ERROR 1) откроется страница с тремя вкладками. На страницах времени в столбце last кликабельно (в примере на слайде 13:42:31). При клике на времени откроется страница с визуализацией строки из диагностического лога.

last - время, в которое последний раз встречалась ошибка в выбранную дату. Если нажать на ссылку поля **last**, будет открыта страница, где будут отражены детали ошибки.

Timeline - универсальный графический элемент, красными вертикальными линиями показывающий, когда встречалась ошибка. Во всей длине одной диаграммы находятся одни сутки, разделённые на секции по часам.

На вкладке **by time** отображается выбор времени справа вверху. Эта вкладка полезна при большом числе сообщений в логе.

Расширенная аналитика: статистика по таблицам



Расширенная аналитика: статистика по таблицам

При выборе в меню страницы пункта "Статистика по таблицам" откроется список мониторуемых баз данных экземпляра PostgreSQL. В примере две базы данных: `postgres` и `test_db`.

Справа вверху есть текстовое поле поиска "укажите фильтр"/"specify filter", в котором можно набрать буквосочетание для первого столбца - имени базы данных. Поле поиска (фильтр) есть на многих страницах и оно выполняет поиск по одному из отображаемых на странице столбцов с текстом. Поиск выполняется по введённому буквосочетанию в одном из текстовых столбцов на странице. В каком столбце не описывается, нужно догадаться, это может быть и первый и последний столбец на странице. В примере можно набрать `os` и в списке останется строка по базе `postgres`.

На странице кликабельно только число 216. При клике на число открывается страница с деталями по таблицам и индексам.

Вкладки `heatmap` (карта нагрева) и пункт меню "процессы" заполняются при использовании расширения `pgpro_stat_activity`.

Расширенная аналитика: взаимоблокировки

Timestamp:	2025-09-21 16:54:54.162 +00	UUID:	8d13ea30-96f2-11f0-95b4-a10832a1b96	LOG	lock
Host:	10.0.2.15	Application:	pgbench: pgbench	DETAIL	unparsed
Database:	postgres	User:	----	CONTEXT	CONTEXT
PID:	9888	vXID:	33/332	STATEMENT	query: raw
Client:	[local]	txID:	----	ERROR	ERROR
				DETAIL	unparsed
				HINT	unparsed
				CONTEXT	CONTEXT
				STATEMENT	query: raw
					query: parsed

10.0.2.15 / deadlock-detect

укажите фильтр ...

*x

Учебный курс "Платформа Tantor 6" Тантор Лабс © 2025 www.tantorlabs.ru

tantor

150

Расширенная аналитика: взаимоблокировки

При клике в меню на deadlock, появится страница со списком взаимоблокировок. На этой странице 4 вкладки: по типам, по приложениям, по времени, графики.

На странице со списком взаимоблокировок можно кликнуть на значение в столбце last и откроется страница с деталями команды, которая была прервана.

Страницы с просто блокировками имеют похожее содержимое. По умолчанию, показывается строка с суммарным временем ожидания получения блокировки. В этой строке столбец "кол-во" не пуст (в примере в столбце число 6). Кликнув на строку можно раскрыть список блокировок, у них в столбце "кол-во" пусто и размер шрифта меньше:

Timestamp: 2025-03-21 16:55:21.074 +0000

44141: 9d1fb0bc0-96f2-11f0-95b4-af10832a1b96

LOG lock

Host: 10.0.2.15

Application: pgbench: pgbench

CONTEXT STATEMENT query: raw

Database: postgres

User: -----

LOG query : parsed

PID: 9888

vXID: 33/3

query::profile
LSC

```
LOG      1 process 9888 acquired ShareLock on transaction 141869 after 18314.165 ms
CONTEXT  2   while updating tuple (969,109) in relation "stock_items"
STATEMENT 3   UPDATE stock_items SET descr = 'updated at'|| now(), amount = amount + 1
             4   WHERE order_items_lid in (
             5       SELECT T.order_items_lid from(
             6           SELECT floor(random()*150000)::integer as order_items_lid, generate_series(1,200)
             7       )T
             8   );
             9   UPDATE
            10  stock_items
            11  SET
            12  descr = 'updated at'|| now()
            13  , amount = amount + 1
            14  WHERE
            15  order_items_lid IN /
```

Расширенная аналитика: Проблемные запросы

Проблемные запросы 3428

Задачи по паттерну ef0b5ce9-c33b-755c-e732-54eb5a3d5a14

Дата	Срок	Задача
Ссылка на задачу/ошибку		
<input type="text" value="вставьте ссылку"/> <input type="button" value="+"/>		

10.0.2.15 / Планы / Режим:

по шаблонам по моделям по приложениям по объектам базы (операции) по объектам базы (ресурсы) по динамическим объектам (операции) по динамическим объектам (ресурсы)

по триггерам пиковые цепочки по времени и ресурсам

Шаблон	метод	app	кол-во	sum, мс	avg, мс	bufmem	bufdsk	buftmp	%	last	Timeline
_ad7c6622	pgbench		707	33 424 721.999	47 276.834	101 554.598	23 541.927		18.8	2011.51	
_afc84abb	pgbench		88	4 640 736.268	52 735.640	764.801	1147.306	627.175	69.9	1726.28	
_745d0af5	pgbench, pgsql		2	943	2 158 133.665	2288.583	89 216.725	7219.808	75	2012.07	
_5a3d5a14	pgbench		588	1819 742.732	3 094.801	57515.935	9 389.254		14.0	2012.11	

Update on stock_items
-> Nested Loop
-> HashAggregate
-> Subquery Scan on t
-> ProjectSet
-> Result
-> IndexScan using stock_items_idx02 on stock_items

app	кол-во	sum, мс	avg, мс	bufmem	bufdsk	buftmp	%	last	Timeline
_e43debd5	506	1054 352.455	2 083.701	7 410.004	5 065.270		40.6	2012.13	
_798eab79	267	970 756.342	3 635.792	29 488.076	4 958.695		14.4	1959.00	
_8c1cd1e3	18	689 301.348	38 294.519	153.058	97.633		38.9	1700.06	

151

Учебный курс "Платформа Tantor 6" Тантор Лабс © 2025 www.tantolabs.ru

Расширенная аналитика: Проблемные запросы

Проблемные запросы - те, которые попали в диагностический лог (например, длительность превышает `log_min_duration_statement`) или их длительность превышает `pg_store_plans.min_duration`.

При наведении на строку слева у первого столбца появляется всплывающий значок из четырех горизонтальных полосок . При наведении на него появляется всплывающее окно "Задачи по паттерну", в котором можно добавить http/https ссылки на любые страницы. Например, на страницу задачи (task, ticket) в корпоративной программе отслеживания задач. После добавления ссылки можно самостоятельно указать дату когда задача была создана и срок решения.

У запросов с добавленными ссылками, между первым и вторым столбцами значок с четырьмя полосками:

Задачи по паттерну efdaelc6-cd29-ce86-431b-9a7bad7c6622

Дата	Срок	Задача
2025-09-22	не задан	http://jira.tantolabs.ru/PRODUCT/123
Ссылка на зад <input type="button" value="Редактировать"/> <input type="button" value="+"/>		
<input type="text" value="вставьте ссылку"/> <input type="button" value="+"/>		

`last` - время, в которое запрос с данным идентификатором последний раз выполнялся в выбранную дату. Если нажать на ссылку поля `last`, будет открыта страница "Запрос из архива", где будут отражены детали запроса.

`Timeline` - график интенсивности появления проблемных запросов. Используется универсальный графический элемент, вертикальными полосками показывающий, когда встречался запрос. В каждой полоске 10 минут. Высота полосок - число запросов в промежутке времени. Во всей длине одной диаграммы находятся **одни сутки** (дата выбирается справа вверху страницы), разделённые на секции по часам. Цвет показывает относительную длительность запросов. Если запросы быстрые, то они зелёного цвета, менее быстрые - жёлтого, а самые медленные красного цвета.

https://docs.tantolabs.ru/tp/6.1/instances/pg_monitor/pg_m_miscellaneous_control_elementes.html

Расширенная аналитика: PSA (pg_stat_activity)

PSA	PSA apps	active = 4 idle = 3 wait = 4 всего = 11								
PID	dbname	backend	xact	host / apname			client	changed	query	state
2952	postgres		2025-09-21	2025-09-21	pg_bench			2025-09-21	2025-09-21	active
				21:35:13.323465	21:57:20.815557			21:57:20.815564	21:57:20.815557	
2949	postgres		2025-09-21	2025-09-21	pg_bench			2025-09-21	2025-09-21	active
				21:35:13.313848	21:58:19.399494			21:58:19.399501	21:58:19.399494	
2950	postgres		2025-09-21	2025-09-21	pg_bench			2025-09-21	2025-09-21	active
										UPDATE stock_items \$
PSA	PSA apps	active = 4 idle = 3 wait = 4 всего = 11								
PID	dbname	backend	xact	host / apname			client	changed	query	state
15636	postgres		2025-09-21	2025-09-21	pgsql			2025-09-21	2025-09-21	idle
				21:55:26.563447				21:55:27:985226	21:55:27:981000	
15860	postgres		2025-09-21	2025-09-21	eebad04b504e [294]			127.0.0.1	2025-09-21	idle
										WITH nspjs AS (SELECT)
PSA	PSA apps	active = 4 idle = 3 wait = 4 всего = 11								
PID	dbname	backend	xact	host / apname			client	changed	query	state
2954	postgres		2025-09-21	2025-09-21	pg_bench			2025-09-21	2025-09-21	wait
			21:35:13.337000	21:37:07:391117				21:37:07:391124	21:37:07:391117	
2953	postgres		2025-09-21	2025-09-21	pg_bench			2025-09-21	2025-09-21	wait
			21:35:13.332264	21:37:13.629715				21:37:13.629000	21:37:13.629715	
16506	postgres		2025-09-21	2025-09-21	pg_bench			2025-09-21	2025-09-21	wait
			21:56:50.096545	21:56:50.053686				21:56:50.286272	21:56:50.053686	select order_items_l_id
										IPC : MessageQueueSend

Расширенная аналитика: PSA (pg_stat_activity)

PSA (pg_stat_activity) - состояние запросов из представления pg_stat_activity.

PSA apps - мониторинг по названиям приложений

PSA	PSA apps	active = 4 idle = 3 wait = 4 всего = 11								
			apname	active	idle	idle-tx	wait	всего	xact	
	pg_bench			3	—	—	4	7	2025-09-21	DELETE FROM stock_items
									21:37:07:391117	
	pg-monitor.pgstat			1	—	—	—	1	2025-09-21	SELECT clock_timestamp(),
									21:58:38.550141	
PSA	PSA apps	active = 4 idle = 3 wait = 4 всего = 11								
			apname	active	idle	idle-tx	wait	всего	xact	
	pg-monitor.tblstat			—	2	—	—	2		WITH nspjs AS (SELECT)
	pgsql			—	1	—	—	1		select pg_reload_conf();
	pg_bench			3	—	—	4	7	2025-09-21	DELETE FROM stock_items
									21:37:07:391117	
PSA	PSA apps	active = 4 idle = 3 wait = 4 всего = 11								
			apname	active	idle	idle-tx	wait	всего	xact	
	pg_bench			3	—	—	4	7	2025-09-21	DELETE FROM stock_items
									21:37:07:391117	
	pgsql			—	1	—	—	1		select pg_reload_conf();
	pg-monitor.pgstat			1	—	—	—	1	2025-09-21	SELECT clock_timestamp(),
									21:58:38.550141	
	pg-monitor.tblstat			—	2	—	—	2		WITH nspjs AS (SELECT json

Расширенная аналитика: CHECKPOINT

10.0.2.15 / CHECKPOINT

по времени

	buffers		xlog			time, sec			sync			size, KB		params
	wrote	%	added	removed	recycled	write	sync	total	files	longest, sec	avg, sec	distance	estimate	
0129:46	2439	14.0				33	37.852	1.148	39.135	12	0.889	0.096	541069	543077
0128:58	3458	21.0				33	41.428	1.263	42.897	13	0.701	0.098	540882	543300
0127:45	2469	15.0				33	44.144	1.148	45.467	17	0.896	0.068	540666	543569
0126:46	2191	13.0				33	32.864	0.279	33.342	11	0.128	0.026	540699	543891
0125:47	2190	13.0				33	51.888	1.106	53.149	13	0.330	0.085	539994	544246
0124:46	2198	13.0				33	51.620	0.966	52.878	13	0.600	0.075	541437	544719
0123:30	2205	13.0				33	48.902	0.715	49.761	17	0.563	0.043	540258	545083
0122:11	2307	14.0				33	25.457	1.103	26.629	10	0.496	0.111	540942	545619
0121:39	1784	10.0				33	48.876	0.086	49.169	10	0.059	0.009	540566	546139
0120:34	3126	19.0				33	101.514	2.901	104.808	12	2.277	0.242	540392	546758
0118:33	4555	27.0				33	24.927	0.540	25.574	12	0.210	0.045	541168	547466
0117:45	793	4.0				33	54.258	1.294	55.649	14	0.632	0.093	540214	548166
0116:26	1639	10.0				33	34.608	0.991	35.676	15	0.627	0.067	540905	549049
0115:42	1792	10.0				33	47.887	1.910	49.909	10	0.965	0.191	540911	549954

tantor

153

Учебный курс "Платформа Tantor 6" Тантор Лабс © 2025 www.tantorlabs.ru

Расширенная аналитика: CHECKPOINT

Статистика сообщений о контрольных точках из диагностического журнала PostgreSQL.
При клике на ссылку в первом столбце, откроется запись из журнала

Timestamp:	2025-09-22 01:21:11.349 +00	UUID:	7c40fffd0-9739-11f0-a608-834159251afc	Host:	10.0.2.15	Application:	:	User:	---	VXID:	---	TXID:	---	
LOG	2025-09-22 01:21:11.349 MSK [2837] [] LOG: checkpoint complete: wrote 2307 buffers (14.1%), 0 WAL file(s) added, 0 removed, 33 recycled; write=25.457 s, sync=1.103 s, total=26.629 s, sync files=10, longest=0.496 s, average=0.111 s, distance=5409													
	42 kB, estimate=545619 kB; lsn=18/282C608, redo lsn=17/E4075700													
LOG	2025-09-22 01:21:11.349 MSK [2837] [] LOG: checkpoint complete: wrote 2307 buffers (14.1%), 0 WAL file(s) added, 0 removed, 33 recycled; write=25.457 s, sync=1.103 s, total=26.629 s, sync files=10, longest=0.496 s, average=0.111 s, distance=5409													
	42 kB, estimate=545619 kB; lsn=18/282C608, redo lsn=17/E4075700													
из которой можно, кликнув на Timestamp, посмотреть сообщения в журнале вокруг сообщения о контрольной точке:														
Параметры просмотра лога:	10.0.2.15 10.0.2.15	PID:	012034299											
2025-09-22 01:17:45.481	2837	---	---	---	---	---	---	---	---	---	---	---	---	
LOG	2025-09-22 01:17:45.481 MSK [2837] [] LOG: checkpoint complete: wrote 793 buffers (4.8%), 0 WAL file(s) added, 0 removed, 33 recycled; write=54.258 s, sync=1.294 s, total=55.649 s, sync files=14, longest=0.632 s, average=0.093 s, distance=540214 kB, estimate=548166 kB; lsn=17/7E6AD860, redo lsn=17/60016168													
2025-09-22 01:16:49.833	2837	---	---	---	---	---	---	---	---	---	---	---	---	
LOG	checkpoint starting: wal													
2025-09-22 01:18:07.281	2837	---	---	---	---	---	---	---	---	---	---	---	---	
LOG	checkpoint starting: wal													
2025-09-22 01:17:45.481	2837	---	---	---	---	---	---	---	---	---	---	---	---	
LOG	2025-09-22 01:17:45.481 MSK [2837] [] LOG: checkpoint complete: wrote 793 buffers (4.8%), 0 WAL file(s) added, 0 removed, 33 recycled; write=54.258 s, sync=1.294 s, total=55.649 s, sync files=14, longest=0.632 s, average=0.093 s, distance=540214 kB, estimate=548166 kB; lsn=17/7E6AD860, redo lsn=17/60016168													
2025-09-22 01:18:09.678	2950	2/133	postgres	---	---	pgbench								2025-09-22 01:
LOG	select order_items_1_id, order_items_2_id, max(amount) as max_amount from stock_items t1 join order_items_1 o1 on t1.order_items_1_id = o1.id join order_items_2 o2 on t1.order_items_2_id = o2.id where optcounter > 14078 and optcounter < 15877 group by order_items_1_id, order_items_2_id;													
LOG	Finalize GroupAggregate (cost=116967.83..167581.12 rows=294299 width=40) (actual time=82315.958..133107.529 rows=440578 loops=1)													перейти к анализу
	Group Key: t1.order_items_1_id, t1.order_items_2_id													
	Buffers: shared hit=47039 read=40217 dirtied=450 written=1718, temp read=1241 written=1247													
	I/O Timings: shared read=915.338 write=231.993, temp read=59.674 write=59.604													
	-> Gather Merge (cost=116967.83..167581.43 rows=358626 width=40) (actual time=82315.910..118558.372 rows=440578 loops=1)													
	Workers Planned: 2													
	Workers Launched: 0													

В которых можно кликнув "перейти к анализу" перейти на страницу анализа плана запроса.

Расширенная аналитика: Логи

Параметры просмотра лога: 10.0.2.15:100215 PID 012034:299

2025-09-22 01:17:45.481 2837 ---- ---- ----

LOG 2025-09-22 01:17:45.481 MSK [2837] | LOG: checkpoint complete: wrote 793 buffers (4.8%), 0 WAL file(s) added, 0 removed, 33 recycled; write=54.258 s, sync=1.294 s, total=55.649 s; sync files=14, longest=0.632 s, average=0.093 s, distance=540214 kB, estimate=548166 kB; lsn=T7/7E6AD860, redo lsn=T7/60016168

2025-09-22 01:16:49.833 2837 ---- ---- ----

2025-09-22 01:18:07.281 2837 ---- ---- ----

LOG checkpoint starting: wal

2025-09-22 01:17:45.481 2837 ---- ---- ----

LOG 2025-09-22 01:17:45.481 MSK [2837] | LOG: checkpoint complete: wrote 793 buffers (4.8%), 0 WAL file(s) added, 0 removed, 33 recycled; write=54.258 s, sync=1.294 s, total=55.649 s; sync files=14, longest=0.632 s, average=0.093 s, distance=540214 kB, estimate=548166 kB; lsn=T7/7E6AD860, redo lsn=T7/60016168

2025-09-22 01:18:09.678 2950 2/133 postgres ---- pgbench

2025-09-22 01:17:45.481 2837 ---- ---- ----

LOG select order_items_1_id, order_items_2_id, max(amount) as max_amount
from stock_items t1
join order_items_1 o1 on t1.order_items_1_id = o1.id
join order_items_2 o2 on t1.order_items_2_id = o2.id
where optcounter > 14078 and optcounter < 15877
group by order_items_1_id, order_items_2_id;

LOG Finalize GroupAggregate (cost=116967.83..167581.12 rows=294299 width=40) (actual time=82315.958..133107.529 rows=440578 loops=1)
Group Key: t1.order_items_1_id, t1.order_items_2_id
Buffers: shared hit=47039 read=40217 dirtied=450 written=1718, temp read=1241 written=1247
I/O Timings: shared read=915.338 write=231.993, temp read=59.674 write=59.604
> Gather Merge (cost=116967.83..161948.43 rows=358626 width=40) (actual time=82315.910..118558.372 rows=440578 loops=1)
Workers Planned: 2
Workers Launched: 0

22.09.2025 Логи

укажите фильтр...

2025-09-22 01:17:45.481 2837 ---- ---- ----

Расширенная аналитика: Логи

Справа вверху страниц есть ссылка "Логи", кликнув на которую, откроется просмотр диагностического лога кластера баз данных PostgreSQL.

Если в сообщении присутствует идентификатор запроса, то у сообщения есть ссылка "перейти к анализу", по которой можно перейти на страницу анализа плана запроса.

Нас странице можно выбрать время, на которое надо показать сообщения и/или PID процесса, чьи сообщения надо показать. PID процесса, по умолчанию, присутствует в сообщениях текстового файла лога в квадратных скобках.

Расширенная аналитика: Мегазапросы

Проблемные запросы 263

Мегазапросы	кол-во	duration	execetime	diff	+%	last
результаты	82	622 251.799	202 902.152	419 349.647	106.7	00:55:09
параметры		12 246.002	1398.194	10 847.808	675.8	00:55:09
запросы		8 249.976	1100.120	7 149.856	549.9	00:55:03
diff	61	9 755.481	3 378.048	5 877.433	74.0	00:55:00

10.0.2.15 / resultset

по приложениям по времени heatmap

Метод

pgbench	кол-во	bytes	avg
pgbench	82	911 745 960	11'118'853
pgbench		12 305 120	0128:40
pgbench		13 420 880	0124:36
pgbench		17 145 440	0123:06

Timeline

911 745 960 11'118'853 0128:40 12 305 120 0128:40 13 420 880 0124:36 17 145 440 0123:06

Учебный курс "Платформа Tantor 6" Тантор Лабс © 2025 www.tantolabs.ru

Расширенная аналитика: Мегазапросы

Мегазапросы – запросы, превышающие объем 1МБ входящего или исходящего трафика и превышающие время log_min_duration_statement или pg_store_plans.min_duration.

Виды мегазапросов:

resultset – входящий трафик к серверному процессу (тело запроса и его переменных привязки), столбцы: Метод (название приложения), кол-во, **bytes**, **avg**, **last**, Timeline. Кликнув на поле last у запроса, откроется страница его плана.

params – исходящий трафик от серверного процесса клиенту (результат запроса);

query – тест запросов

diff – сравнение времени выполнения запросов (пример в правом верхнем углу слайда), столбцы: Метод (название приложения), кол-во, **duration**, **execetime**, **diff**, **+%**, last, Timeline.

На страницах показываются запросы, сгруппированные по приложениям (параметр application_name, если приложение его заполняет). Кликнув на строку, "раскроются" запросы, выполнявшиеся в сессиях с названием этого приложения. Кликнув на поле last у запроса, откроется запись диагностического лога с последним выполнением запроса.

Не во всех вкладках могут присутствовать данные. Например, вкладка heatmap заполняется при использовании pgpro_stat_activity. В текущей версии pg_stat_activity данных для этой вкладки нет. Платформа Tantor поддерживает оба расширения.

https://docs.tantolabs.ru/tp/6.1/instances/pg_monitor/pg_m_Mega-queries.html

Расширенная аналитика: PG config

параметр	значение	описание	изменен
Расположения файлов			
config_file	⚠ /var/lib/postgresql/tantor-se-17/data/postgresql.conf	Определяет основной файл конфигурации	2025-08-13 11:33:34
data_directory	⚠ /var/lib/postgresql/tantor-se-17/data	Определяет каталог данных сервера.	2025-08-13 11:33:34
hba_file	⚠ /var/lib/postgresql/tantor-se-17/data/pg_hba.conf	Задаёт путь к файлу конфигурации "hba".	2025-08-13 11:33:34
ident_file	⚠ /var/lib/postgresql/tantor-se-17/data/pg_ident.conf	Задаёт путь к файлу конфигурации "ident".	2025-08-13 11:33:34
Подключения и аутентификация			
authentication_timeout	1min	Ограничивает время, за которое клиент до	2025-08-12 13:50:10
изменен			
укажите фильтр ...			
10.0.2.15			
изменен			
укажите фильтр ...			
параметры	изменения		
изменен			
укажите фильтр ...			

Учебный курс "Платформа Tantor 6" Тантор Лабс © 2025 www.tantolabs.ru

Расширенная аналитика: PG config

Самый последний пункт меню: **PG config**. При клике на пункт меню открывается страница со списком всех параметров конфигурации экземпляра, в том числе параметров всех загруженных библиотек. Обычно, библиотеки загружаются параметром `shared_preload_libraries`.

Параметры выводятся в разделах по своим категориям.

Желтый треугольник указывает на то, что параметр может иметь значение, отличное от значения по умолчанию.

На странице две вкладки: "параметры" и "изменения".

Во вкладке "изменения" выводится история изменений параметров. Сверху - самые свежие изменения. В примере на слайде последним менялось значение параметра `auto_explain.log_min_duration`, сначала на 1s (в 21:53), а через две минуты (в 21:55) значение было изменено на 100s.

Практика 8

1. Откройте "Профилизатор запросов"/"Query Profiler"
2. В списке запросов, отсортируйте запросы по параметру **Записано временных блоков или Строки**
3. Откройте один из запросов и перейдите на вкладку **ПЛАНЫ**
4. Откройте какой-нибудь план запроса и посмотрите страницу плана, диаграммы, статистику, рекомендации
5. Откройте **Расширенную аналитику**, найдите запрос, который вернул больше 2Мб данных и посмотрите его план
6. Посмотрите диагностический лог экземпляра PostgreSQL на хосте 10.0.2.15 порту 5432
7. Посмотрите страницу истории изменения параметров конфигурации (PG config)
8. Посмотрите страницу диагностики Расширенной аналитики (pg-monitor)

Практика 8

1. Откройте "Профилизатор запросов"/"Query Profiler"
2. В списке запросов, отсортируйте запросы по параметру **Записано временных блоков или Строки**
3. Откройте один из запросов и перейдите на вкладку **ПЛАНЫ**
4. Откройте какой-нибудь план запроса и посмотрите страницу плана, диаграммы, статистику, рекомендации
5. Откройте **Расширенную аналитику**, найдите запрос, который вернул больше 2Мб данных и посмотрите его план
6. Посмотрите диагностический лог экземпляра PostgreSQL на хосте 10.0.2.15 порту 5432
7. Посмотрите страницу истории изменения параметров конфигурации (PG config)
8. Посмотрите страницу диагностики Расширенной аналитики (pg-monitor)



9

Экземпляры PostgreSQL

Конфигурирование и обслуживание PostgreSQL

Текущая активность

The screenshot shows the Tantor 6 PostgreSQL monitoring interface. At the top, a navigation bar indicates the path: Tenant 1 / Edu / Экземпляры / tantor:5432 / Текущая активность / Пользователи. The main area displays two cards: 'ПОЛЬЗОВАТЕЛИ' (52/100) and 'СИСТЕМА' (10/100). A sidebar on the left lists databases: postgres, template1, and test_db. The 'Текущая активность' (Current Activity) option is highlighted. A modal window titled 'Колонки таблицы' (Table Columns) is open, listing various connection states with checkboxes. A note in the modal says 'ЗАРЕЗЕРВИРОВАНО ДЛЯ СУПЕРПОЛЬЗОВАТЕЛЯ' (Reserved for Superuser) with a count of 3. Below the cards, a table provides detailed statistics for each database.

БАЗА ДАННЫХ	ИСПОЛЬЗУЕТСЯ	ИСПОЛЬЗУЕТСЯ %	МАКС. КОЛ-ВО СЕССИЙ	АКТИВНО	↑ IDLE	IDLE IN TX	IDLE IN TX (ABORTED)	ОТКЛЮЧЕНО	НЕТ ПРИВИЛЕГИЙ	ОЖИДАНИЕ
postgres	10	10	100	4	0	0	0	0	0	0
template1	6	6	100	1	0	0	0	0	0	0

Текущая активность

На странице "Текущая активность" отражаются сгруппированные данные по процессам экземпляра. Они актуальны: запрашиваются при открытии страницы обновляются раз в пять секунд.

Данные можно обновлять чаще, нажав на квадратик со стрелками справа от поля "Поиск".

Нажав на значок шестерёнки справа вверху таблицы, откроется всплывающее окно со списком столбцов таблицы, где можно скрыть столбцы.

На странице две вкладки: ПОЛЬЗОВАТЕЛИ (сессии с базами данных) и СИСТЕМА (фоновые процессы экземпляра в операционной системе). Для фоновых процессов, не связанных с базами данных, имя базы данных в столбце БАЗА ДАННЫХ пусто (в примере на слайде последняя строка в таблице).

Максимальное число сессий определяется параметром `max_connections` (по умолчанию 100). Число зарезервированных соединений определяется параметрами `superuser_reserved_connections` (по умолчанию 3) и `reserved_connections` (по умолчанию 0).

Текущая активность: Завершить процесс

Tenant 1 / Edu / Экземпляры / tantor:5432 / Текущая активность / Пользователи / postgres

ВЫПОЛНЕНИЕ ОЖИДАНИЕ ЗАБЛОКИРОВАНО

Поиск по pid

PID	Имя пользователя	ЦПУ	ОЗУ %	ЧТЕНИЕ (Б/С)	ЗАПИСЬ (Б/С)	ОЖИДАНИЕ И/О	ОЖИДАНИЕ	СОСТОЯНИЕ	ДЛИТЕЛЬНОСТЬ	ЗАПРОС	ЗАВЕРШИТЬ
6414	postgres	0	0.29	0	0	Нет	Нет	idle	8.14 min	select fas	<input type="button" value="Завершить"/>
22150	postgres	0	0.29	0	0	Нет	Нет	idle	2.72 sec	UPDATE st	<input type="button" value="Завершить"/>
21701	postgres	0	0.29	0	0	Нет	Нет	idle	1.19 sec	UPDATE pr	<input type="button" value="Завершить"/>

Рядом с каждым процессом есть красная кнопка с изображением крестика. Кликнув на неё, вы увидите детальную информацию о процессе и окно подтверждения завершения.

Детальная информация о процессе 21701:

Рид	21708	UPDATE pgbench_tellers SET tbalance = tbalance + 1498 WHERE tid = 5;
Имя пользователя	postgres	
ЦПУ	0.2 %	
ОЗУ %	1.91 %	
Чтение И/О	0 Б/сек	
Запись И/О	0 Б/сек	
Ожидание И/О	Нет	
Ожидание	Да	
Состояние	active	
Длительность	5.18 sec	

Окно подтверждения завершения:

Завершить процесс?

Процесс №21708 будет завершен

Для подтверждения введите "завершить"*

tantor

160

Текущая активность: Завершить процесс

Кликнув на строку, появится страница со списком сессий базы данных или фоновых процессов. Страница перерисовывается (обновляется) часто и чтобы приостановить ее перерисовку можно нажать на красную кнопку "Заморозить", кнопка поменяет название на "Отменить заморозку". При нажатии на кнопку "Завершить" появится окно с подтверждением. Сессия прерывается вызовом функции `pg_terminate_backend` (PID).

На слайде показан пример списка серверных процессов (Пользователи вверху на breadcrumb). Пример фоновых процессов (Система):

ВЫПОЛНЕНИЕ ОЖИДАНИЕ ЗАБЛОКИРОВАНО

Поиск по pid

PID	Имя пользователя	ОЗУ %	СОСТОЯНИЕ	ДЛИТЕЛЬНОСТЬ	ЗАПРОС	ЗАВЕРШИТЬ
1106		0.38	unknown	7.07 h	autovacuum launcher	<input type="button" value="Завершить"/>
1111	postgres	0.1	unknown	7.07 h	logical replication launcher	<input type="button" value="Завершить"/>
1059		1.82	unknown	7.07 h	checkpointer	<input type="button" value="Завершить"/>
1060		1.79	unknown	7.07 h	background writer	<input type="button" value="Завершить"/>
1105		0.13	unknown	7.07 h	walwriter	<input type="button" value="Завершить"/>
2012	postgres	0.16	active	7.07 h	START_REPLICATION SLOT "replica" 19/37000000 TIMELINE 1	<input type="button" value="Завершить"/>

* Частота обновления данных - 5 сек

При нажатии на кнопку "Завершить", вызывается функция, но фоновые процессы `checkpointer`, `bgwriter`, `walwriter` не останавливаются:

```
select pg_terminate_backend(1059);
```

WARNING: PID 1059 is not a PostgreSQL backend process
pg_terminate_backend

Текущая активность: ОЖИДАНИЕ и ЗАБЛОКИРОВАНО

PID	ИМЯ ПОЛЬЗОВАТЕЛЯ	ОЖИДАНИЕ I/O	ОЖИДАНИЕ	СОСТОЯНИЕ	ДЛИТЕЛЬНОСТЬ	ЗАПРОС	
10896	postgres	Нет	Нет	idle in transaction	9.93 min	update t set n=2 where n=2;	<button>Завершить</button>
14171	postgres	Нет	Да	active	9.7 min	update t set n=2 where n=2;	<button>Завершить</button>
15408	postgres	Нет	Да	active	9.41 min	update t set n=1 where n=1;	<button>Завершить</button>

ВЫПОЛНЕНИЕ	ОЖИДАНИЕ	ЗАБЛОКИРОВАНО
Поиск по pid	<input type="text"/>	

PID	ИМЯ ПОЛЬЗОВАТЕЛЯ	РЕЖИМ БЛОКИРОВКИ	ТИП БЛОКИРОВКИ	СОСТОЯНИЕ	ЗАПРОС	
10896	postgres	ExclusiveLock	transactionid	idle in transaction	update t set n=2 where n=2;	<button>Завершить</button>
14171	postgres	ShareLock	transactionid	active	update t set n=2 where n=2;	<button>Завершить</button>

ВЫПОЛНЕНИЕ	ОЖИДАНИЕ	ЗАБЛОКИРОВАНО
Поиск по pid	<input type="text"/>	

Текущая активность: ОЖИДАНИЕ и ЗАБЛОКИРОВАНО

Для просмотра дерева блокировок можно создать таблицу с двумя строками и три сессии:

```
create table t (n numeric);
insert into t values (1);
insert into t values (2);
1 begin;
1 update t set n=1 where n=1;
2 begin;
2 update t set n=2 where n=2;
1 update t set n=2 where n=2;
3 update t set n=1 where n=1;
```

Сессии 1 (PID=14171) и 3 (PID=15408) заблокированы, сессия 2 (PID=10896) нет.

Если кликнуть на строку базы данных, с которой работают сессии, то появится таблица со списками сессий на закладке ВЫПОЛНЕНИЕ. На этой странице есть ещё две вкладки: ОЖИДАНИЕ и ЗАБЛОКИРОВАНО. Часть столбцов можно убрать, нажав на значок шестеренки.

Если нажать на сессию на вкладке ВЫПОЛНЕНИЕ, то появится неинформативное окно с командой и деталями ее выполнения, которые и так отражаются в столбцах на странице ВЫПОЛНЕНИЕ.

Если же нажать на сессию на вкладке ОЖИДАНИЕ или ЗАБЛОКИРОВАНО, то появится совсем другая страница. На этих вкладках отображаются только сессии, которые простоявают и не могут работать и нужно определить почему, кто эти сессии заблокировал или чего они ждут.

ОЖИДАНИЕ - это, например, ожидание ответа от дисковой системы, а ЗАБЛОКИРОВАНО - это ожидание получения блокировки строки, которая заблокирована другой сессией.

Текущая активность: дерево блокировок

Показать все деревья блокировки

Завершить

PID	ВРЕМЯ	ИЗМЕНЕНИЕ	ПОЛЬЗОВА...	СТАТУС	РЕЖИМ	ЗАПРОС	ИН...
10896	02:42:09	02:41:50	postgres	idle in tran...		update t set...	
14171	02:42:39	02:41:37	postgres	active	exclusive	update t set... info	

Tenant 1 / Edu / Экземпляры / tantor:5432 / Текущая активность / Пользователи / postgres / 14171 / reverse

Показать все деревья блокировки

PID	ВРЕМЯ	ИЗМЕНЕНИЕ	ПОЛЬЗОВА...	СТАТУС	РЕЖИМ	ЗАПРОС	ИН...
10896	00:21:40	00:21:21	postgres	idle in trans...		update t set...	
14171	00:22:10	00:21:08	postgres	active	exclusive	update t set... info	
15408	00:20:56	00:20:50	postgres	active	exclusive	update t set... info	

Pid	14171
Заблокированные PID	
Имя пользователя	postgres
База данных	postgres
ЦПУ	0 %
ОЗУ %	0.26 %
Чтение I/O	0 B/sec
Запись I/O	0 B/sec
Ожидание I/O	Нет
Ожидание	Да
Состояние	active
Длительность	22.58 min

Учебный курс "Платформа Tantor 6" Тантор Лабс © 2025 www.tantolabs.ru

tantor

162

Текущая активность: дерево блокировок

Находясь на вкладке ОЖИДАНИЕ или ЗАБЛОКИРОВАНО можно кликнуть на сессию и появится страница со списком процессов и запросов, в процессе выполнения которых они находятся.

При открытии сессии со страницы ОЖИДАНИЕ на открывшейся странице вверху будет присутствовать чекбокс "Показать все деревья блокировки". Если нажать на него, то добавятся процессы из очереди ожидания.

При открытии сессии со страницы ЗАБЛОКИРОВАНО чекбокса нет, так как и так высветятся все заблокированные сессии и сессии, ожидающие команды от клиента (idle in transaction).

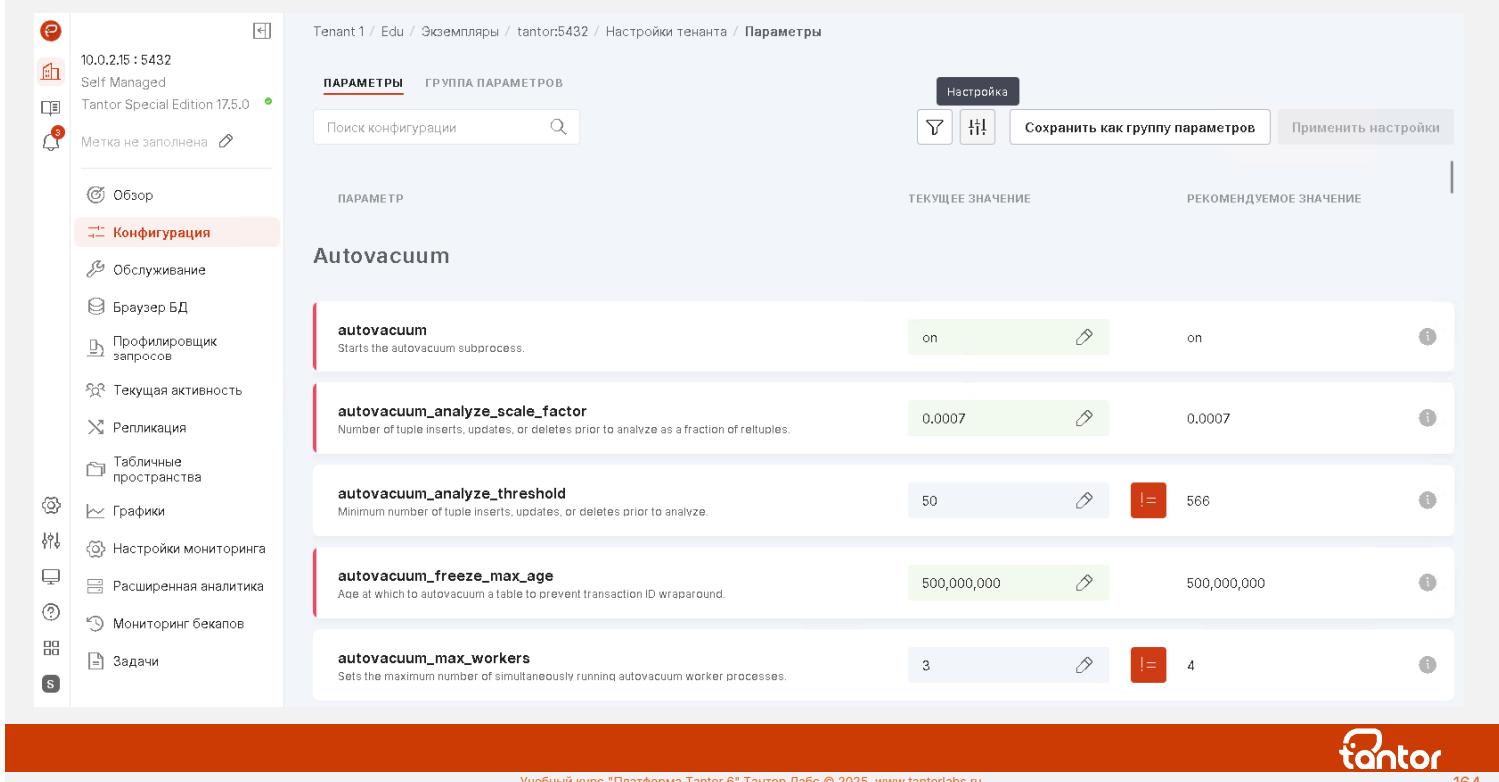
Команда в сессии не видна, только начало команды, но если кликнуть на строку, то справа откроется окно, в котором будет полностью видна сама команда и детали её выполнения, а также кнопка "Завершить", которой можно прервать сессию (завершить работу серверного процесса, обслуживающего сессию). В верхнем правом углу появившегося окна есть значок "x", нажав на который можно закрыть появившееся окно.



9-2

Конфигурация экземпляра PostgreSQL

Экземпляр: Конфигурация



The screenshot shows the Tantor PostgreSQL configuration interface. On the left, a sidebar lists various management tasks: Обзор, Конфигурация (selected), Обслуживание, Браузер БД, Профилировщик запросов, Текущая активность, Репликация, Табличные пространства, Графики, Настройки мониторинга, Расширенная аналитика, Мониторинг бекапов, and Задачи. The main content area is titled 'ПАРАМЕТРЫ ГРУППА ПАРАМЕТРОВ' (Parameters Group of Parameters). It shows a search bar 'Поиск конфигурации' and buttons 'Настройка' (Configure), 'Сохранить как группу параметров' (Save as group of parameters), and 'Применить настройки' (Apply settings). The 'ПАРАМЕТР' (Parameter) column lists 'autovacuum', 'autovacuum_analyze_scale_factor', 'autovacuum_analyze_threshold', 'autovacuum_freeze_max_age', and 'autovacuum_max_workers'. The 'ТЕКУЩЕЕ ЗНАЧЕНИЕ' (Current Value) column shows the current values: 'on', '0.0007', '50', '500,000,000', and '3'. The 'РЕКОМЕНДУЕМОЕ ЗНАЧЕНИЕ' (Recommended Value) column shows the recommended values: 'on', '0.0007', '566', '500,000,000', and '4'. A red exclamation mark icon is present in the 'autovacuum_analyze_threshold' row, indicating a difference between the current and recommended values.

Экземпляр: Конфигурация

На странице Конфигурация показаны параметры экземпляра PostgreSQL. Параметры хранятся в файлах `postgresql.conf` и `postgresql.auto.conf`

Существует больше 370 параметров плюс параметры библиотек.

Если текущее значение параметра отличается от рекомендуемого конфигуратором, встроенным в Платформу Tantor, то на это указывает знак `!=`.

Платформа собирает метрики с экземпляра PostgreSQL и примерно 25 из этих метрик поступает в конфигуратор. Конфигуратор анализирует метрики и на их основе предлагает внести изменения в параметры конфигурации. Число параметров, по которым даются рекомендации, от 80 до 100 (зависит от версии PostgreSQL).

Можно создать группу параметров, связать её с несколькими экземплярами и применять значения параметров к ним.

В столбце ТЕКУЩЕЕ ЗНАЧЕНИЕ показывается действующее значение параметра, а не то, которое сохранено в файле параметров `postgresql.auto.conf`. **Зеленоватый фон** в этом столбце означает, что Платформа рекомендует значение для этого параметра и значение равно рекомендованному Платформой.

Красная полоса слева от названия параметра означает, что значение параметра присутствует в файлах конфигурации в раскомментированном виде.

Убрать красную полосу можно только вручную командой `alter system reset` параметр или убрав или закомментировав строку в файлах параметров.

На слайде три параметра имеют **красную полосу**. Содержимое файла:

```
tail -n 3 $PGDATA/postgresql.auto.conf
autovacuum_analyze_scale_factor = '0.0007' #значение по умолчанию 0.1
autovacuum_freeze_max_age = '500000000' #значение по умолчанию 10000000000
autovacuum = on #является значением по умолчанию
```

<https://docs.tantorlabs.ru/tp/6.1/instances/configurations.html>

Экземпляр: изменение параметров конфигурации

The screenshot shows the Tantor Special Edition 17.5.0 interface. In the top right, a red warning box says: "Чтобы применить все изменения, перезапустите экземпляр" (To apply all changes, restart the instance). Below it, a green success message says: "Конфигурации успешно обновлены. Перейти в список экземпляров" (Configuration successfully updated. Go to the list of instances). The main table shows configuration changes:

	СТАРОЕ ЗНАЧЕНИЕ	НОВОЕ ЗНАЧЕНИЕ	В ОЖИДАНИИ
autovacuum_analyze_scale_factor	0.1	0.0007	Перезагрузить
autovacuum_freeze_max_age	10000000000	5000000000	Перезапустить экземпляр

Buttons at the bottom right of the modal are "Отменить" (Cancel) and "Применить и перезагрузить" (Apply and restart). The left sidebar shows navigation: "Настройки мониторинга", "Расширенная аналитика", "Мониторинг бекапов", and "Задачи". The bottom right corner shows the Tantor logo and the number 165.

Экземпляр: изменение параметров конфигурации

Платформа меняет параметры командой `alter system set` параметр = значение. Изменения сохраняются в файле `postgresql.auto.conf`.

После изменения параметров кнопка "Применить настройки" становится активной, красного цвета. Если нажать на эту кнопку появится всплывающее окно, в котором будут показаны применяемые параметры, у каждого параметра - достаточно ли перечитать файлы параметров или нужен перезапуск экземпляра.

Если перезапуск экземпляра не нужен, то появится всплывающее зелёное окно с текстом: "Конфигурации успешно обновлены. Перейти в список экземпляров".

Если какой-то параметр без перезапуска экземпляра не применился, то появится всплывающее красное окно с текстом: "Чтобы применить все изменения, перезапустите экземпляр" и на странице останется красный треугольник левее значка с воронкой.

При последующих заходах на страницу "Конфигурация" будет появляться это красное всплывающее окно. После рестарта экземпляра страница показывает неактуальные значения минуту.

Перезапустить экземпляр средствами Платформы, если экземпляр не входит в кластер Patroni нельзя.

Часть параметров доступна только для чтения. У таких параметров в окне со значением нет иконки карандаша. Эти параметры нельзя изменить напрямую.

Историю изменений параметров конфигурации экземпляра можно посмотреть на странице [Расширенная аналитика -> PG config -> изменения](#).

Параметры конфигурации: настройка и фильтр

Настройка

Фильтры

Имеет рекомендуемое значение

По умолчанию

Измененные значения

Изменения ожидают перезагрузки

Изменения ожидают перезапуска

Отличается от рекомендуемого

Не примененные значения

Категория

Auto vacuum

Client Connection Defaults / Locale and Formatting

Client Connection Defaults / Other Defaults

Client Connection Defaults / Shared Library Preloading

Client Connection Defaults /

Фильтры

Статус значения

Категория

Очистить

Настройка

Автоматическая настройка

Ручная настройка

Тип управления нагрузкой

Автоматическая настройка

Тип нагрузки

OLAP

OLTP

Mixed

ERP1C

Тип диска

SSD

SSD

HDD

Network Storage

Параметры конфигурации: Фильтры и Настройка

Параметров больше 370. Кнопка Фильтры позволяет отфильтровать параметры, которые будут показываться на странице по Статусам и Категориям, что уменьшит число параметров, показываемых на странице.

Настройка влияет на рекомендации конфигуратора. В настройках можно установить:

Тип управления нагрузкой:

Автоматическая настройка - Платформа в автоматическом режиме подбирает профиль нагрузки наблюдаемой БД. Текущий автоматически подобранный профиль указан в селекторе "Тип нагрузки" серого цвета (не меняется).

Ручная настройка - позволяет вручную указать тип нагрузки для экземпляра.

OLAP (OnLine Analytical Processing) - базы данных, обслуживающие аналитические приложения (DSS, Descision Support System), хранилища (DW, Data Warehouses). В таких приложениях преимущественно разовые (ad-hoc), длительные запросы, обрабатывающие много строк.

OLTP (OnLine Transaction Processing) - приложения, обслуживающие повседневную деятельность компаний. Преимущественно короткие запросы и транзакции.

Mixed - смешанная нагрузка.

ERP1C (Enterprise Resource Planning) - базы данных, обслуживающие приложения 1С или аналогичные.

Тип диска:

SSD (Solid State Disks, твердотельные накопители, NVMe)

HDD (Жёсткие, ротационные, магнитные диски, винчестеры)

Network storage (NAS, Network Attached Storage, подключаемые системы хранения).

Описание параметров

autovacuum_vacuum_cost_delay
Vacuum cost delay in milliseconds, for autovacuum.

Информация

autovacuum_vacuum_cost_delay

Информация

Категория:	Autovacuum
Тип параметра:	real
Ед. измерения:	ms
Значение по умолчанию:	2
Минимальное значение:	-1
Максимальное значение:	100
Контекст:	sighup
Перезапуск:	нет

Описание

На основе стоимости (cost) устанавливает для операций автоочистки задержку в миллисекундах. Если значение указано без единиц измерения, то оно считается заданным в миллисекундах. Параметр может быть установлен только в файле `postgresql.conf` или в командной строке сервера. Для отдельных таблиц настройка может быть переопределена через изменение параметров хранения таблицы. Если автоочистка слишком сильно влияет на производительность выполняющихся запросов, вы можете увеличить это значение до 50 мс, но это приведёт к увеличению времени выполнения отдельных задач очистки.

[Официальная документация](#)

[Подробнее](#) [Закрыть](#)

Информация

autovacuum_vacuum_cost_delay (floating point) #

Указывает значение задержки на основе стоимости, которое будет использоваться в автоматических операциях `VACUUM`. Если указано значение -1, будет использовано обычное значение `vacuum_cost_delay`. Если это значение указано без единиц измерения, оно принимается в миллисекундах. Значение по умолчанию -2 миллисекунды. Этот параметр может быть установлен только в файле `postgresql.conf` или в командной строке сервера; но настройка может быть переопределена для отдельных таблиц путем изменения параметров хранения таблицы.

autovacuum_vacuum_cost_limit (integer) #

Указывает значение ограничения стоимости, которое будет использоваться в автоматических операциях `VACUUM`. Если указано значение -1 (которое является значением по умолчанию), будет использовано обычное значение `vacuum_cost_limit`. Обратите внимание, что значение распределяется пропорционально среди работающих рабочих процессов автоочистки, если их больше одного, так чтобы сумма ограниченной для каждого рабочего процесса не превышала значение этой переменной. Этот параметр может быть установлен только в файле `postgresql.conf` или в командной строке сервера; но настройку можно переопределить для отдельных таблиц, изменив параметры хранения таблицы.

[Назад](#) [Наверх](#) [Начало](#) [Далее](#)

19.9. Статистика выполнения 19.11. Параметры подключения клиента

[Подробнее](#) [Закрыть](#)

Учебный курс "Платформа Tantor 6" Тантор Лабс © 2025 [www.tantolabs.ru](#)

tantor 167

Описание параметров

При нажатии на значок "i" в правой части строк появится всплывающее окно с информационной справкой о параметре конфигурации (пример слева на слайде).

При нажатии на ссылку "Подробнее" в новой вкладке браузера или окне браузера откроется тот же текст из Базы Знаний Платформы.

Если кликнуть внизу окна с информационной справкой на ссылку Официальная документация, то в всплывающем окне откроется страница документации о параметре (пример справа на слайде).

Группы параметров конфигурации и модификаторы

- Группа параметров - набор почти всех параметров конфигурации
 - К экземпляру можно привязать только одну группу параметров
 - Одна группа параметров может быть привязана к нескольким экземплярам
 - Значения параметров в системных группах менять нельзя
 - Значения параметров в пользовательской группе параметров можно менять
 - С экземплярами Patroni группы параметров не работают
- Модификаторы - ноль или более параметров конфигурации, которым можно установить значения
 - Модификаторы привязываются к экземплярам
 - Отношение между модификаторами и экземплярами многое ко многому

Группы параметров конфигурации и модификаторы

Группа параметров - набор почти всех параметров конфигурации. "Почти" потому, что в новых версиях могут появляться новые параметры, а какие-то параметры удаляются при выходе новой основной версии PostgreSQL.

Группы параметров доступны только для отдельных экземпляров. С экземплярами под управлением Patroni функционал групп параметров не работает.

К экземпляру можно привязать только одну группу параметров. Одна группа параметров может быть привязана к нескольким экземплярам.

Значения параметров в **пользовательской** группе параметров можно менять. Для применения изменённых значений ко всем привязанным экземплярам, нужно выполнить "синхронизацию".

В Платформе имеются предустановленные ("системные") группы параметров для каждой мажорной версии PostgreSQL и Tantor Postgres. Значения параметров в этих группах - это значения по умолчанию в конкретной мажорной версии и форке PostgreSQL. **Значения параметров в системных группах менять нельзя**, поэтому смысла привязывать системные группы к экземплярам нет. Смысл системных групп - перечислить параметры, которые есть в версии и форке PostgreSQL, чтобы на основе этих групп создавать собственные (**пользовательские** группы параметров), в которых **можно** менять значения параметров. Также системные группы параметров полезны, чтобы сравнить какие есть отличия в значениях параметров между разными версиями и форками PostgreSQL.

Модификаторы - ноль или более параметров конфигурации, которым можно установить значения. Модификаторы привязываются к экземплярам. Отношение между модификаторами и экземплярами **многое ко многому** (экземпляр может иметь несколько модификаторов, а модификатор может быть привязан к нескольким экземплярам). Привязать модификатор к экземпляру можно только после привязки какой-нибудь группы параметров.

Модуль: Группы параметров

Название	Описание	Семейство	Тип	Создано	Изменено	Экземпляров
<input checked="" type="checkbox"/> PostgreSQL 16 (system config group)	Default setting for PostgreSQL 16	PostgreSQL 16	System	21/09/2024 21:16	21/09/2024 21:16	0
<input checked="" type="checkbox"/> PostgreSQL 17 (system config group)	Default setting for PostgreSQL 17	PostgreSQL 17	System	05/06/2025 12:49	05/06/2025 12:49	0
<input type="checkbox"/> Tantor 14 (system config group)	Default setting for Tantor 14	Tantor 14	System	21/09/2024 21:17	21/09/2024 21:17	0

Модуль: группы параметров

Группы параметров настраиваются отдельно для каждого тенанта. После выбора тенанта, в левой вертикальной линейке меню можно нажать на значок с четырьмя квадратиками, появится меню Модули, в котором выбрать "Группы параметров". Откроется страница с двумя вкладками: ГРУППЫ и МОДИФИКАТОРЫ.

Семейства параметров - это основные версии от 9 до текущей для PostgreSQL и от 14 до текущей для Tantor Postgres. При создании новой группы указывается семейство.

Типы групп: System и User.

При клике на три точки появится всплывающее меню с пунктами: Открыть, Переименовать, Дублировать, Удалить.

Можно дублировать системную группу, дубль будет пользовательской группой и можно будет устанавливать значения параметров.

В первом столбце таблицы можно галочками выбрать две или три группы, внизу появится всплывающее окно с кнопкой Сравнить. Можно будет сравнить значениями и наличием каких параметров отличаются группы.

https://docs.tantorlabs.ru/tp/6.1/instances/param_groups.html

Сравнение и создание групп параметров

The screenshot shows the DataGrip interface with the 'Сравнение групп параметров' (Compare parameter groups) dialog open. It displays a table of parameters from 'POSTGRESQL 17 (SYSTEM CONFIG GROUP)' and 'POSTGRESQL 16 (SYSTEM CONFIG GROUP)'. The table includes columns for parameter name, value in group 1, and value in group 2. Parameters listed include: application_name, backend_flush_after, bgwriter_flush_after, checkpoint_completion_target, checkpoint_flush_after, and checkpoint_timeout. The 'Создание группы параметров' (Create parameter group) dialog is also visible, showing fields for selecting a family (PostgreSQL 17), choosing a group (PostgreSQL 17 (system config group)), entering a new group name, and providing a description. Buttons for 'Закрыть' (Close), 'Отменить' (Cancel), and 'Создать' (Create) are present.

Сравнение и создание групп параметров

Сравнение показывает различия в значениях параметров и их наличии.

При нажатии на красную кнопку "Создать группу параметров" появится всплывающее окно, в котором можно указать семейство (основная версия) параметров, группу на основе которой создаётся новая (откуда будут взяты начальные значения параметров) и название для создаваемой группы.

Группу можно переименовать, но нельзя поменять название или описание группы параметров, к которой привязан хотя бы один экземпляр.

Создать группу можно дублированием. При дублировании указывается имя новой группы и описание.

Группу можно создать на странице "Конфигурация" экземпляра PostgreSQL, нажав на кнопку "Сохранить как группу параметров".

Примеры всплывающих окон дублирования и сохранения:

Дублирование группы параметров

Название новой группы *

Описание новой группы

0 / 50

Отменить Дублировать

Сохранение как группы параметров

Название новой группы *

конфигуратор может дать рекомендации

Максимум 50 символов 37 / 50

Описание новой группы

описание

Привязать экземпляр к группе параметров

Отменить Добавить

Модификаторы групп параметров

ГРУППЫ МОДИФИКАТОРЫ

Поиск 🔍

Создать модификатор

НАЗВАНИЕ	ОПИСАНИЕ	СОЗДАНО	ИЗМЕНЕНО	ЭКЗЕМПЛИЯРОВ
modifier		24/09/2025 13:26	24/09/2025 13:27	2
modifier1		25/09/2025 13:26		2
modifier2		25/09/2025 13:26		1

Невозможно удалить

Вы не можете удалить модификатор с привязанными экземплярами

Отменить Посмотреть экземпляры

Открыть

Дублировать

Удалить

Модули

Учебный курс "Платформа Tantor 6" Тантор Лабс © 2025 www.tantolabs.ru

Модификаторы групп параметров

Модификаторы - ноль или более параметров конфигурации, которым можно установить значения. Модификаторы привязываются к экземплярам. Отношение между модификаторами и экземплярами **многое ко многому** (экземпляр может иметь несколько модификаторов, а модификатор может быть привязан к нескольким экземплярам). Привязать модификатор к экземпляру можно только после привязки какой-нибудь группы параметров.

Если есть несколько модификаторов с одинаковыми параметрами, но разными значениями, при синхронизации выбирается значение из какого-то модификатора, оно будет показано после нажатия на кнопку "Синхронизировать и применить" вместе со всеми применяемыми параметрами. Критерий выбора - из какого модификатора берётся значение неизвестен в 6 версии.

Удалить модификатор с привязанными экземплярами нельзя, но можно поменять значения параметров конфигурации в модификаторе, добавить и удалить параметры.

Редактирование модификаторов

✓ Отличий в значениях параметров не выявлено. Экземпляры успешно синхронизированы

The screenshot shows a user interface for managing parameters. On the left, there's a sidebar with various icons. The main area has tabs for 'ПАРАМЕТРЫ' (Parameters) and 'ЭКЗЕМПЛЯРЫ' (Instances). A green banner at the top right says '✓ Модификатор успешно обновлён' (Modifier successfully updated). A yellow message box in the center says 'Есть синхронизированные экземпляры. При внесении изменений в этот модификатор произойдет рассинхронизация с ними. Посмотреть экземпляры' (There are synchronized instances. Changes to this modifier will cause resynchronization with them. View instances). Below this is a search bar and a 'Параметр' (Parameter) section. A modal window titled 'Редактирование' (Editing) shows the parameter 'work_mem' with the value '8 MB'. Buttons for 'Отменить' (Cancel) and 'Сохранить' (Save) are at the bottom. To the right, a 'Сохранить изменения?' (Save changes?) dialog box is open, asking 'Это действие приведёт к изменению модификатора и рассинхронизации с экземплярами. Хотите сохранить?' (This action will change the modifier and resynchronize with instances. Do you want to save?). It has a text input 'сохранить' (save) and a 'Сохранить' (Save) button. The footer of the page includes the Tantor logo and the text 'Учебный курс "Платформа Tantor 6" Тантор Лабс © 2025 www.tantolabs.ru'.

Редактирование модификаторов

На странице модификатора можно добавить, удалить, поменять значения параметров.

При добавлении параметра ему нужно установить значение.

Справки по параметрам в этих окнах нет.

Проблема в том, что формат параметров разнообразный, а Платформа принимает не все форматы. Эти окна позволяют установить значения параметров, которые некорректны. После этого синхронизация перестаёт работать. Но хуже того, Группа перестаёт показываться на странице Конфигурация. Для редактирования значений придётся заходить в Модули -> Группы параметров -> МОДИФИКАТОРЫ. После установки значений принимаемых Платформой или удаления параметров на Группы и модификаторы станут отображаться на странице Конфигурация -> ГРУППА ПАРАМЕТРОВ. Примеры ошибок:

The screenshot shows a user interface with error messages. A red banner at the top left says '⚠ Ошибка при добавлении модификатора' (Error adding modifier). A red banner at the top right says 'Не удалось сравнить группы конфигураций' (Failed to compare configuration groups). A red message box at the bottom left says 'Не удалось получить статус синхронизации группы конфигураций' (Failed to get synchronization status for configuration group). A button 'Добавить модификатор' (Add modifier) is at the bottom right. The interface includes tabs for 'ПАРАМЕТРЫ' (Parameters) and 'ГРУППА ПАРАМЕТРОВ' (Configuration Group).

Например, для параметров с единицами измерений "B", "kB", "MB", "GB", and "TB" между числом и единицей измерения обязательно ставить пробел, хотя пробел **не обязателен** в командах:

```
set work_mem to '8mB';
ERROR: invalid value for parameter "work_mem": "8mB"
HINT: Valid units for this parameter are "B", "kB", "MB", "GB", and "TB".
set work_mem to '8MB';
SET
set work_mem to '8 MB';
SET
```

Значения, принимаемые Платформой стоит протестировать на странице Конфигурация, где в окнах изменения значений приведены примеры и справка по форматам значений параметров.

Синхронизация на странице Конфигурация

10.0.2.15 : 5432
Self Managed
Tantor Special Edition 17.5.0
Метка не заполнена

ГРУППА ПАРАМЕТРОВ

Добавить модификатор Синхронизировать и применить

Группа параметров

НАЗВАНИЕ	ОПИСАНИЕ	ПРИВЯЗ.	СИНХР.	ДАТА СИНХР.
группа1		✓		25/09/2025 22:19

Модификаторы, которые переписывают группу параметров

НАЗВАНИЕ	ОПИСАНИЕ
модификатор	

Синхронизация на странице Конфигурация

На странице Экземпляр -> Конфигурация -> ГРУППА ПАРАМЕТРОВ есть кнопка "Синхронизировать и применить". При нажатии на кнопку появится всплывающее окно со списком параметров и их значений, которые будут применены, если нажать красную кнопку "Синхронизировать".

Привязать экземпляр к группе параметров и синхронизировать его - разные понятия.

Привязать экземпляр - значит установить связь между ним и группой параметров в рамках Платформы.

Синхронизировать - значит применить значения группы параметров и модификаторы к привязанному к группе и модификаторам экземпляру. Применяться будут параметры, значения которых отличаются от тех, которые установлены у экземпляра.

Изменения на странице Конфигурация применяется только к текущему экземпляру. К другим экземплярам, с которыми может быть связана группа, изменения не применяются (не выполняется синхронизация). Изменения всегда применяются с учетом модификаторов:

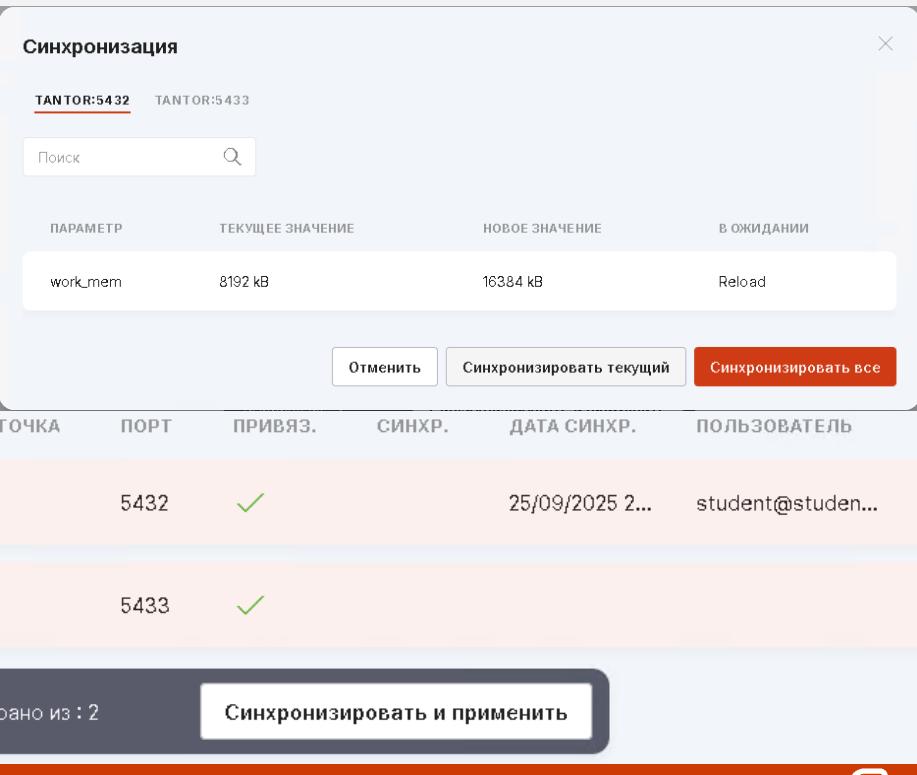
Синхронизация

⚠ К группе параметров группа1 добавлены модификаторы. Модификаторы переписывают параметры группы

ПАРАМЕТР	ТЕКУЩЕЕ ЗНАЧЕНИЕ	НОВОЕ ЗНАЧЕНИЕ	В ОЖИДАНИИ
work_mem	8192 kB	16384 kB	Reload

Отменить Синхронизировать

Синхронизация на странице модуля: Группы параметров



Синхронизация

TANTOR:5432 TANTOR:5433

Tenant 1 / Группы параметров / групп

ПАРАМЕТРЫ ЭКЗЕМПЛЯРЫ

ПАРАМЕТР	ТЕКУЩЕЕ ЗНАЧЕНИЕ	НОВОЕ ЗНАЧЕНИЕ	В ОЖИДАНИИ
work_mem	8192 kB	16384 kB	Reload

Поиск

Имя узла Конечная точка Порт Привяз. Синхр. Дата синхр. Пользователь

tantor 10.0.2.15 5432 ✓ 25/09/2025 2... student@student...

tantor 10.0.2.15 5433 ✓

Выбрано из : 2 Синхронизировать и применить

tantor 174

Учебный курс "Платформа Tantor 6" Тантор Лабс © 2025 www.tantolabs.ru

Синхронизация на странице модуля: Группы параметров

Второй способ синхронизации на странице: Модули -> Группы параметров -> ЭКЗЕМПЛЯРЫ. На этой странице можно выделить галочкой экземпляры, внизу появится всплывающее окно "Синхронизировать и применить". При нажатии на эту кнопку появится всплывающее окно "Синхронизация". В этом окне кнопка "Синхронизировать всё" красного цвета, однако кнопка белого цвета "Синхронизировать текущий" равноправна и может быть нажата вместо красной кнопки. В этом окне вкладки с названиями экземпляров. Можно кликать на названия экземпляров, просматривать параметры, которые будут к ним применены и нажимать на кнопку "Синхронизировать текущий". Значения параметров у разных экземпляров могут быть разными, если с ними связаны разные модификаторы.

Изменения применяются **всегда** с учетом модификаторов, хотя на этой странице модификаторы не упоминаются. Значения, которые применяются, берутся из модификаторов и это те значения, которые будут в реальности применены.

Процесс синхронизации асинхронный и может занимать до 5 минут. До истечения этого времени и не менее минуты на странице Конфигурация будут отображаться прежние значения параметров.



9-3

Параметры конфигурации Patroni



Список кластеров Patroni и их экземпляров

КЛАСТЕРЫ
ЭКЗЕМПЛЯРЫ
АГЕНТЫ

🔍

Добавить кластер Patroni

ИМЯ КЛАСТЕРА	СОСТОЯНИЕ	PATRONI	КОЛ-ВО ЦПУ	ОБЪЕМ ОЗУ	ДИСК ОСНОВНОГО ЭКЗЕМПЛЯРА	ЭКЗЕМПЛЯРЫ
patroni	partial	4.0.4	12	23.3 GB	10.2 GB / 38.1 GB	3 / 3

Tenant 1 / Edu / Кластеры / patroni / Обзор и управление
ОБЗОР И УПРАВЛЕНИЕ
МОНИТОРИНГ

Отправлена команда возобновить на кластер

Отправлена команда пауза на кластер

ВЕРСИЯ PATRONI

ВЕРСИЯ POSTGRES

ОБЪЕМ ОЗУ

КОЛ-ВО ЦПУ

ДИСК ОСНОВНОГО ЭКЗЕМПЛЯРА

ЭКЗЕМПЛЯРЫ

РЕПЛИКИ

4.0.4

17.3

23.3 GB

12

38.1 GB

3 / 3

2 / 2

Возобновить

Конфигурация кластера

Пауза/Обслуживание

Экземпляры кластера
Частота обновления - 1 минута

ИД	СОСТОЯНИЕ	РОЛЬ	ОС	ТИП	ВЕРСИЯ	ЦПУ	МЕТКА	КОНЕЧНАЯ ТОЧКА	ПОРТ	TTL	ЗАДЕРЖКА	⋮
43	streaming	replica	Debian GNU/Linux	Postgres	17.3	4		172.21.1.3	5432	30 0 B		⋮
44	running	primary	Debian GNU/Linux	Postgres	17.3	4		172.21.1.1	5432	30 0 B		⋮

tantor

176

Список кластеров Patroni и их экземпляров

Кластер Patroni это экземпляр PostgreSQL, принимающий изменения (master, primary, основной, ведущий) и несколько физических реплик (standby, резервных). Patroni наблюдает за доступностью экземпляров и, если мастер не доступен, то из одной из реплик делает мастера.

На страницу Cluster Config можно попасть со страницы "Пространства", вкладка КЛАСТЕРЫ. На этой странице кликнуть на строку с именем кластера и в открывшейся странице нажать кнопку "Конфигурация кластера".

Кнопка "Пауза/Обслуживание" приостанавливает работу скрипта Patroni. Скрипт Patroni не будет пытаться что-либо менять в работе экземпляров PostgreSQL.

Кнопка "Возобновить" - возобновит работу скрипта Patroni.

Меню, которое открывается при нажатии на три точки у экземпляра PostgreSQL в списке "Экземпляры кластера":

ИД	СОСТОЯНИЕ	РОЛЬ	ОС	ТИП	ВЕРСИЯ	ЦПУ	МЕТКА	КОНЕЧНАЯ ТОЧКА	ПОРТ	TTL	ЗАДЕРЖКА	⋮
43	streaming	replica	Debian GNU/Linux	Postgres	17.3	4		172.21.1.3	5432	30 0 B		⋮
44	running	primary	Debian GNU/Linux	Postgres	17.3	4		172.21.1.1	5432		Перезагрузить	⋮
42	streaming	replica	Debian GNU/Linux	Postgres	17.3	4		172.21.1.2	5432		Повторная инициализация	⋮

Перезапустить

Повторная инициализация

Переключиться

У мастера (primary) пункты "Повторная инициализация" и "Переключиться" не активны. Переключаться можно только на реплику. Повторно инициализировать мастер нельзя, это можно только после переключения на реплику.

Переключение происходит по правилу: если есть синхронная реплика (synchronous_mode: true), то только её можно сделать мастером. Если синхронной реплики нет, то мастером можно сделать любую доступную реплику.

"Перезагрузить" означает перечитать конфигурацию, без перезапуска экземпляра. "Повторная инициализация" - пересоздание реплики скриптом Patroni.

При нажатии на кнопки "Пауза/Обслуживание" и "Возобновить" подтверждение не запрашивается, сразу появляется зелёный всплывающее сообщение.

https://docs.tantorlabs.ru/tp/6.1/instances/ug_clusters_pages.html

Кластер Patroni: МОНИТОРИНГ

Tenant 1 / Edu / Кластеры / patroni / Мониторинг

ОБЗОР И УПРАВЛЕНИЕ **МОНИТОРИНГ**

Patroni кластер ● partial since 2025.10.03 17:59

Интервал выборки исторических данных 🕒 Последние 30 минут

Изменение ролей экземпляра

22:14

● primary
● replica
● replica

172.21.1.3
172.21.1.2
172.21.1.1

21:58 21:59 22:00 22:01 22:02 22:03 22:04 22:05 22:06 22:07 22:08 22:09 22:10 22:11 22:12 22:13 22:14 22:15 22:16 22:17 22:18

История изменений статуса узла кластера

Результаты не найдены

Попробуйте изменить параметры запроса

tantor 177

Учебный курс "Платформа Tantor 6" Тантор Лабс © 2025 www.tantorlabs.ru

Кластер Patroni: МОНИТОРИНГ

На вкладке МОНИТОРИНГ показывается история смены ролей кластеров PostgreSQL.

Зелёный цвет - роль мастера.

Жёлтый цвет - физическая реплика.

Страница Конфигурация у экземпляра в кластере Patroni

172.21.1.3 : 5432
Patroni 4.0.4 replica
Postgres 17.3
Метка не заполнена

ПАРАМЕТРЫ

Поиск конфигурации

Применить настройки

Этот экземпляр находится в кластере Patroni. Чтобы применить настройки внутри кластера, перейдите на [страницу настройки кластера](#)

ПАРАМЕТР	ТЕКУЩЕЕ ЗНАЧЕНИЕ	РЕКОМЕНДУЕМОЕ ЗНАЧЕНИЕ
autovacuum Starts the autovacuum subprocess.	on	on
autovacuum_analyze_scale_factor Number of tuple inserts, updates, or deletes prior to analyze as a fraction of rextuples.	0.1	0.0007
autovacuum_analyze_threshold Minimum number of tuple inserts, updates, or deletes prior to analyze.	50	566
autovacuum_freeze_max_age Age at which to autovacuum a table to prevent transaction ID wraparound.	200,000,000	500,000,000

Страница Конфигурация у экземпляра в кластере Patroni

Если зайти на страницу конфигурации экземпляра PostgreSQL, который управляется Patroni появится **желтая полоса** с предупреждением и ссылкой на [страницу настройки кластера](#) Patroni.

Страница настройки кластера



Информация о настройках

Tenant 1 / Edu / Кластеры / patroni / Конфигурация



ПАРАМЕТРЫ PATRONI ПАРАМЕТРЫ POSTGRESQL

Поиск конфигурации

ТИП ПАРАМЕТР

Autovacuum



autovacuum

Starts the autovacuum subprocess.



autovacuum_analyze_scale_factor

Number of tuple inserts, updates, or deletes prior to analyze as a fraction.



autovacuum_analyze_threshold

Minimum number of tuple inserts, updates, or deletes prior to analyze.



autovacuum_freeze_max_age

Age at which to autovacuum a table to prevent transaction ID wraparound.

1. Все экземпляры, принадлежащие к кластеру Patroni, настраиваются только на этой странице. Параметры будут применены немедленно ко всем экземплярам кластера.

2. Вкладка параметров PostgreSQL имеет специальные значки, указывающие, где задается этот параметр: на уровне конфигурации Patroni или через конфигурацию PostgreSQL на сервере.



PostgreSQL или Tantor



Patroni

on

?

!= 0.0007

?

!= 566

?

!= 500,000,000

?

3. Параметры, измененные вручную на сервере в файлах конфигурации postgresql (postgresql.conf/postgresql.auto.conf), не будут отображаться на этом экране. Вы можете видеть их только на экране конфигурации экземпляра.

4. Интерфейс корректно работает с Patroni версий от 3.3.0 до 4.0.5

Больше не показывать

Учебный курс "Платформа Tantor 6" Тантор Лабс © 2025 www.tantorlabs.ru



179

Страница настройки кластера

Страница Cluster Config имеет две вкладки: ПАРАМЕТРЫ PARONI и ПАРАМЕТРЫ POSTGRESQL. Вкладка ПАРАМЕТРЫ POSTGRESQL показывает параметры, одинаковые на всех экземплярах PostgreSQL, включённых в кластер Patroni. Если значения параметра на разных экземплярах разные, то страница их не покажет. Например, значение параметра PostgreSQL `transaction_read_only` на реплике будет установлено `on`, а на мастере `off`. Также значения параметров репликации `in_hot_standby`, `primary_conninfo`, `primary_slot_name` отличаются на разных экземплярах.

Также на странице нельзя поменять значения, которые представляют собой сложные структуры данных, например, массивы или вложенные JSON.

Параметры на вкладке ПАРАМЕТРЫ POSTGRESQL можно фильтровать по статусу значения и по типу: Только Patroni (параметры PostgreSQL, который устанавливаются и меняются скриптом Patroni), Только PostgreSQL (скрипт Patroni их не настраивает).

Вкладка ПАРАМЕТРЫ PARONI показывает параметры самого Patroni.

Конфигуратор Платформы может рекомендовать значения параметров на этой вкладке.

Группы параметров и модификаторы для экземпляров в кластере Patroni не могут использоваться.

Параметры Patroni которые не меняются на вкладке: `pg_ident`, `pg_hba`, `ignore_slots`, `slots`, `pause`.

Параметры фильтра на этой вкладке: Параметры Patroni можно фильтровать по статусу значения:

Имеет рекомендуемое значение, Не по умолчанию, Изменённые значения, Изменения ожидают перезагрузки, Изменения ожидают перезапуска.

При открытии страницы Cluster Config появится предупреждение о том, что страницы Платформы работают корректно с Patroni версий от 3.3.0 до 4.0.5 и то, что если поменять вручную (командой `alter system set` или отредактировав файлы) файлы параметров конфигурации экземпляра в составе Patroni, то на странице Cluster Config значения параметров не будут отображаться, но эти значения будут видны на странице Конфигурация этого экземпляра, на которой их поменять будет нельзя. Если выбрать галочку "Больше не показывать", то вернуть это окно можно будет нажав на желтый треугольник.

https://docs.tantorlabs.ru/tp/6.1/instances/ug_cluster_config.html

Практика 9a

1. Откройте страницу **Текущая активность**
2. Создайте в psql сессии с блокировками таблицы и строк. Посмотрите как блокировки отражаются на странице **Текущая активность**. Завершите блокирующие сессии

Практика 9b

1. Откройте пункт меню **Конфигурация**
2. Посмотрите рекомендации конфигуратора по изменению значений параметров
3. Поменяйте значение параметра `hot_standby_feedback` на рекомендованное
4. Верните значение параметра в значение по умолчанию
5. Откройте модуль **Группы параметров** и сравните значения параметров PostgreSQL 18 и Tantor 17

Практика 9a

1. Откройте страницу **Текущая активность**
2. Создайте в psql сессии с блокировками таблицы и строк. Посмотрите как блокировки отражаются на странице **Текущая активность**. Завершите блокирующие сессии

Практика 9b

1. Откройте пункт меню **Конфигурация**
2. Посмотрите рекомендации конфигуратора по изменению значений параметров
3. Поменяйте значение параметра `hot_standby_feedback` на рекомендованное
4. Верните значение параметра в значение по умолчанию
5. Откройте модуль **Группы параметров** и сравните значения параметров PostgreSQL 18 и Tantor 17



Экземпляры PostgreSQL

Браузер БД

Браузер БД

The screenshot shows the Tantor Database Browser interface. On the left, a sidebar menu includes: Обзор, Конфигурация, Обслуживание, **Браузер БД** (highlighted in orange), Профилировщик запросов, Текущая активность, Репликация, Табличные пространства, Графики, **Настройки мониторинга** (highlighted in orange), Расширенная аналитика, Мониторинг бекапов, and Задачи. The main content area displays the 'Общее количество баз данных' (2) and 'Общий размер баз данных' (1438 MB). A table lists two databases: 'postgres' (1.4 GB, owner 'postgres', 2 sessions, 3 locks, access granted) and 'test_db' (7.5 MB, owner 'postgres', 0 sessions, 0 locks, access granted). Below this, a 'Список баз данных' section shows '2 из 2' selected databases: 'postgres' and 'test_db', both marked as 'Доступна' (Available) and with 'Да' (Yes) in the 'АГЕНТ ИМЕЕТ ДОСТУП?' (Agent has access?) column. A 'Сохранить' (Save) button is visible. The footer includes the Tantor logo and the text 'Учебный курс "Платформа Tantor 6" Тантор Лабс © 2025 www.tantolabs.ru'.

Браузер БД

"Браузер БД" - пункт меню Экземпляра. В нём список баз данных, мониторинг которых включен в пункте меню "Настройки мониторинга".

На странице "Браузер БД" справа в строках у каждой базы данных по три кнопки: Аудит, SQL-редактор, Схема данных. При нажатии на кнопки открываются страницы этих инструментов.

ВЛАДЕЛЕЦ
postgres

КОДИРОВКА
UTF8

ТАБЛИЧНОЕ ПРОСТРАНСТВО
pg_default

ОГРАНИЧЕНИЕ ПО КОЛ-ВУ ПОДКЛЮЧЕНИЙ
No Limit

Последний собранный
26/09/2025 13:23

[Обновить данные](#)

ВСЕГО СХЕМ
1

ВСЕГО ТАБЛИЦ
8

ВСЕГО ИНДЕКСОВ
9

ВСЕГО ФУНКЦИЙ
27

РАЗМЕР БАЗЫ ДАННЫХ
1,430 МВ

РАСШИРЕНИЯ
plpgsql(1.0)

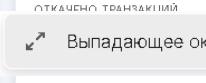
ИНДЕКСЫ
hash(1)

ФУНКЦИИ
c(23)

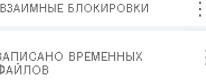
ПОЛУЧЕННЫЕ СТРОКИ

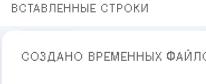

ВОЗВРАЩЕННЫЕ СТРОКИ

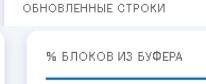

ВЫПОЛНЕНО ТРАНЗАКЦИЙ

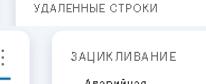

ОТКАНЧЕНО ТРАНЗАКЦИЙ


Выпадающее окно

ВЗАЙМНЫЕ БЛОКИРОВКИ


ВСТАВЛЕННЫЕ СТРОКИ


ОБНОВЛЕННЫЕ СТРОКИ


УДАЛЕННЫЕ СТРОКИ


ЗАПИСАНО ВРЕМЕННЫХ ФАЙЛОВ


СОЗДАНО ВРЕМЕННЫХ ФАЙЛОВ


% БЛОКОВ ИЗ БУФЕРА


ЗАЦИКЛИВАНИЕ
Аварийная автоочистка

ПРОВЕРКА СОСТОЯНИЯ

- Самые большие таблицы
- Таблицы с индексами максимального размера
- Топ записываемых таблиц
- Топ читаемых таблиц
- Таблицы с наибольшим раздутием
- Таблицы с наибольшим кол-вом последовательных сканирований
- Таблицы с недопустимыми индексами
- Таблицы с неиспользуемыми индексами
- Таблицы с кандидатами для частичных индексов
- Индексы таблиц с значениями NULL > 50%
- Таблицы с избыточными индексами
- Таблицы с внешними ключами без индексов
- Таблицы у которых тип данных внешнего ключа отличен от исходного
- Таблицы с низким коэффициентом обновления НОТ



183

Учебный курс "Платформа Tantor 6" Тантор Лабс © 2025 www.tantolabs.ru

Аудит объектов Баз Данных

На странице Аудит в правой верхней части можно нажать на красную кнопку "Собрать данные" и правая часть страницы ПРОВЕРКА СОСТОЯНИЯ заполнится данными аудита (анализа) структур хранения (таблиц и их индексов) базы данных.

У плашек с точками внизу можно кликать на серые точки и содержимое плашк будет меняться. Например, в плашке ИНДЕКСЫ появится: btree(8), что означает, что кроме одного hash-индекса в базе данных ещё есть 8 индексов типа btree:

```
postgres=# \di+
```

List of relations

Schema	Name	Type	Table	Access method	Size
public	order_items_1_pkey	index	order_items_1	btree	3312 kB
public	order_items_2_pkey	index	order_items_2	btree	21 MB
public	pgbench_accounts_pkey	index	pgbench_accounts	btree	4408 kB
public	pgbench_branches_pkey	index	pgbench_branches	btree	16 kB
public	pgbench_tellers_pkey	index	pgbench_tellers	btree	16 kB
public	stock_items_descr_idx	index	stock_items	hash	106 MB
public	stock_items_idx01	index	stock_items	btree	69 MB
public	stock_items_idx02	index	stock_items	btree	78 MB
public	stock_items_pk	index	stock_items	btree	632 MB

(9 rows)

У плашек с графиками справа вверху есть три точки, при нажатии на которые появляется меню со строкой "Выпадающее окно", при нажатии на которую появляется всплывающее окно с увеличенным графиком.

 Выпадающее окно

В фильтре (значок воронки под красной кнопкой) три значения для фильтрации рекомендаций аудита объектов:

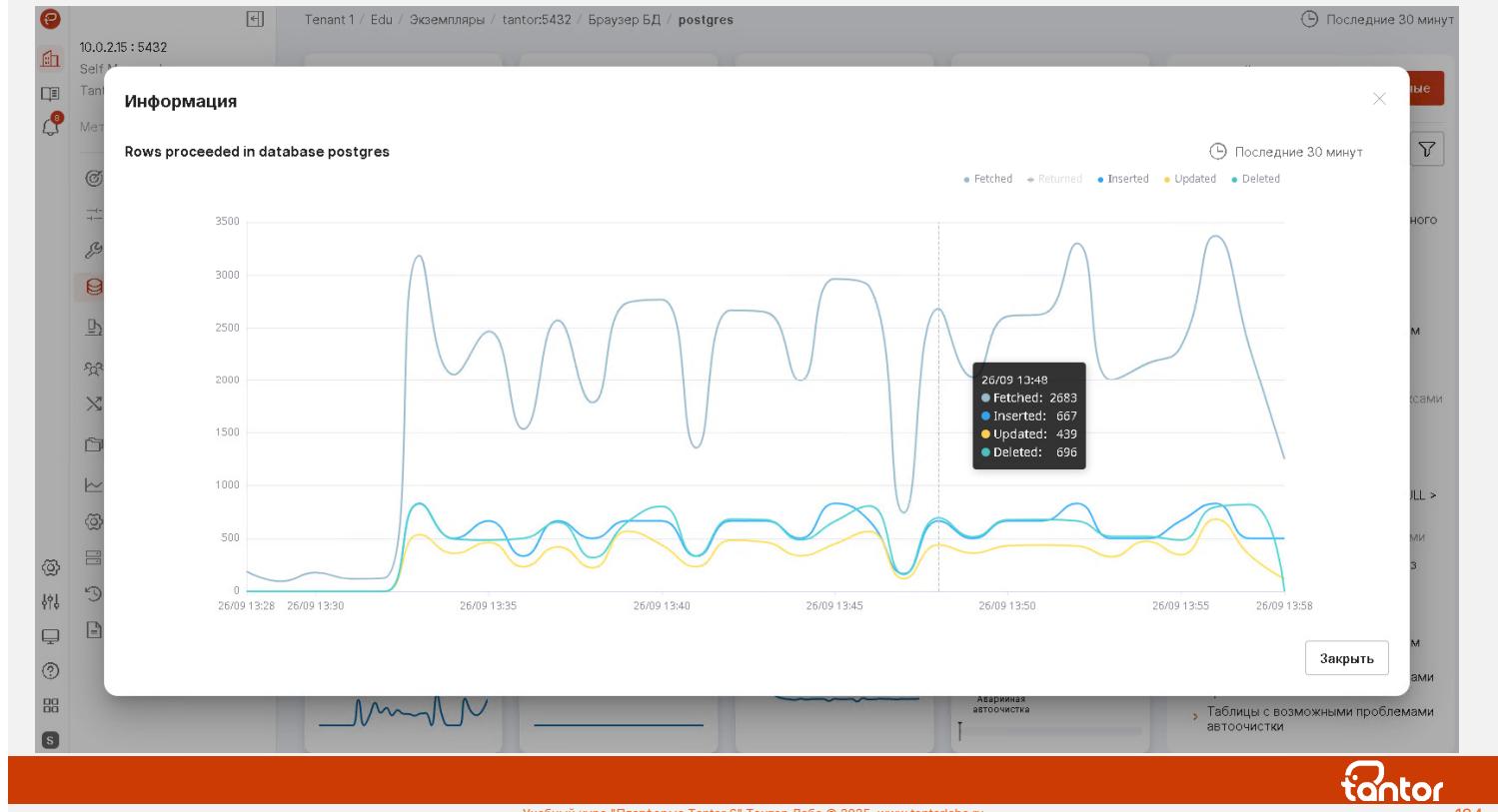
Информация, Предупреждение, Проблема.

Информация

Предупреждение

Проблема

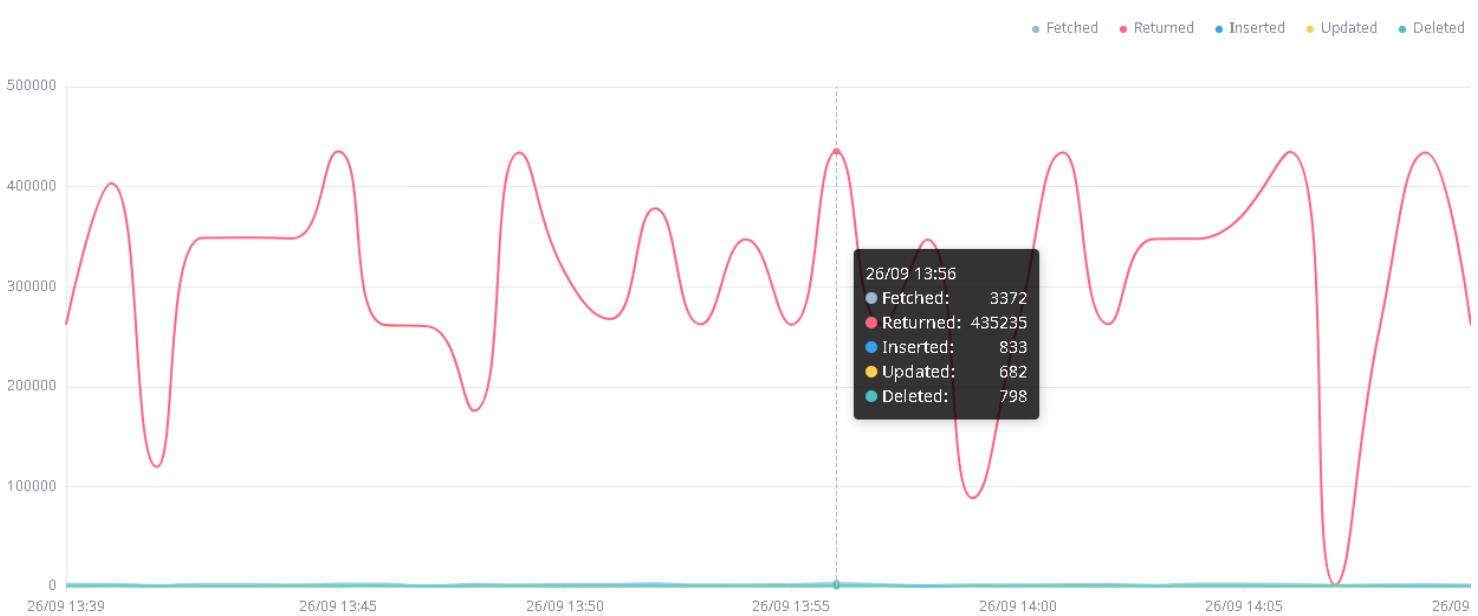
Аудит объектов Баз Данных: графики



Аудит объектов Баз Данных: графики

У плашек: **полученные, возвращенные, вставленные, обновленные, удаленные** строки одно и то же всплывающее окно с 5 линиями на графике.

Если вы видите одну линию, то это означает, что остальные линии около нуля. Например, клиенту возвращается 300 тыс. строк в секунду, а обновлённых, удалённых, выбранных строк 3 тыс. в секунду. Относительно возвращённых (Returned) они невелики и их графики незаметны. Чтобы увидеть их графики, в правой верхней части всплывающего окна нужно кликнуть на Returned, линия вставленных строк исчезнет, масштаб вертикальной оси изменится и станут видны линии вставленных, удаленных и возвращенных строк. Пример, где линии не видны, хотя отображение всех линий включено:



Аудит объектов Баз Данных: Рекомендации

The screenshot shows the Tantor Database Audit interface. On the left is a sidebar with various monitoring and management links. The main area is divided into two tabs: 'ОБЗОР' (Overview) and 'СТАТИСТИКА' (Statistics). The 'ОБЗОР' tab is active, displaying the following table statistics for the 'stock_items' table:

ВЛАДЕЛЕЦ	ТИП ТАБЛИЦЫ	OID ТАБЛИЦЫ	СТРОКА (ПРОГНОЗИРУЕМАЯ)
postgres	Normal	16,649	5,012,969

Below this, there are two rows of metrics:

КОЭФ. ЗАПОЛНЕНИЯ	НАСЛЕДУЕТ ТАБЛИЦЫ	ИМЕЕТ TOAST?	СИСТЕМНАЯ ТАБЛИЦА?
100	false	true	false

Further down, there are two columns: 'Описание' (Description) and 'Рекомендации' (Recommendations). The 'Описание' column lists an index named 'stock_items_descr_idx' with details: Index size: 106 MB, Column: descr, Index Nulls ratio: 0.60, Index tuples read: 0, Index tuples fetch: 0. The 'Рекомендации' column provides a recommendation: "Одностолбцовые индексы с 50% или более значений NULL возможно могут быть заменены частичными индексами для экономии дискового пространства." (One-column indexes with 50% or more NULL values may be replaced by partial indexes to save disk space.)

On the right, a sidebar titled 'ПРОВЕРКА СОСТОЯНИЯ' (Check Status) lists various audit findings, with 'Индексы таблиц с значениями NULL > 50%' (Indexes with NULL values > 50%) being the current focus. A red button at the top right of the sidebar says 'Обновить данные' (Update data).

Аудит объектов Баз Данных: Рекомендации

На странице две вкладки: ОБЗОР и СТАТИСТИКА. В правой части окна "ПРОВЕРКА СОСТОЯНИЯ" (HEALTHCHECKS) выдаются списки таблиц с неоптимальными параметрами хранения. Одна и та же таблица может встречаться в нескольких пунктах. Если кликнуть на название таблицы или индекса, то обновится средняя часть окна и откроются делали хранения этого объекта. В нижней части будут "Рекомендации".

Если таблица встречается в нескольких пунктах, то нажимая на значки "<" и ">" можно проматывать рекомендации по объекту.

В рекомендации, где предлагается выполнить analyze, vacuum, vacuum full внизу появится красная кнопка с названием команды:

The screenshot shows a configuration dialog for 'Выберите операции по обслуживанию' (Select maintenance operations). The 'Действие' (Action) dropdown is set to 'VACUUM', which is highlighted with a red checkmark. Other options include 'ANALYZE', 'VACUUM FULL', and 'None'. Below the dropdown are two checkboxes: 'DISABLE_PAGE_SKIPPING' and 'ANALYZE'. At the bottom are 'Отменить' (Cancel) and 'Готово' (Done) buttons.

On the right, a card titled 'Таблицы с наибольшим раздутием' (Tables with the most bloat) lists a table with 'Раздутие: 49 МБ.' (Bloat: 49 MB). The 'Рекомендации' (Recommendations) section says: 'Необходимо отслеживать результаты проверки состояния.' (It is necessary to monitor the results of the status check.) A red button at the bottom right says 'Запустить Vacuum Full' (Run Vacuum Full).

Below this, a section titled 'Рекомендации' (Recommendations) contains a text block: 'Необходимо изменить типы полей внешних ключей, чтобы избежать проблем с производительностью и приведением типов в запросах. Вы можете изменить типы полей внешнего ключа с помощью команды: ALTER TABLE stock_items ALTER COLUMN order_items_1_id TYPE int8.'

В рекомендации также может быть текст команды:

Необходимо изменить типы полей внешних ключей, чтобы избежать проблем с производительностью и приведением типов в запросах. Вы можете изменить типы полей внешнего ключа с помощью команды: ALTER TABLE stock_items ALTER COLUMN order_items_1_id TYPE int8.

Tenant 1 / Edu / Экземпляры / tantor:5432 / Браузер БД / postgres / **stock_items**

10.0.2.15 : 5432
Self Managed
Tantor Special Edition 17.5.0

Метка не заполнена

Обзор СТАТИСТИКА

РАЗМЕР ТАБЛИЦЫ
Total **1,283 MB**

РАЗДУТИЕ
Size **49 MB**

РАЗМЕР ИНДЕКСОВ & КОЛИЧЕСТВО
Size **908 MB**

ШИРИНА ТАБЛИЦЫ & ДЛИНА
Columns **6**

ОБНОВЛЕНИЕ НОТ & СООТНОШЕНИЕ
0

ОТНОШЕНИЕ СКАНОВ ПОСЛЕ ДОВОДКИ К СКАНОМ ИНДЕКСА
0

ОПЕРАЦИИ НА ЗАПИСЬ
Ratio **0**

СТАТИСТИКА АВТОАНАЛИЗА
Ratio **5951.22**

СТАТИСТИКА АВТОЧИСТКИ
Ratio **109.25**

ОПЕРАЦИИ НА ЧТЕНИЕ
Ratio **99**

СТАТИСТИКА АНАЛИЗОВ
Last Analyze **Никогда**

СТАТИСТИКА ОЧИСТКИ
Last Vacuum **2025-09-26 13:58**

СТОЛБЦЫ ГИСТОГРАММЫ

Название столбца: amount

Учебный курс "Платформа Tantor 6" Тантор Лабс © 2025 www.tantolabs.ru

Последний собранный 26/09/2025 14:14 Обновить данные

ПРОВЕРКА СОСТОЯНИЯ

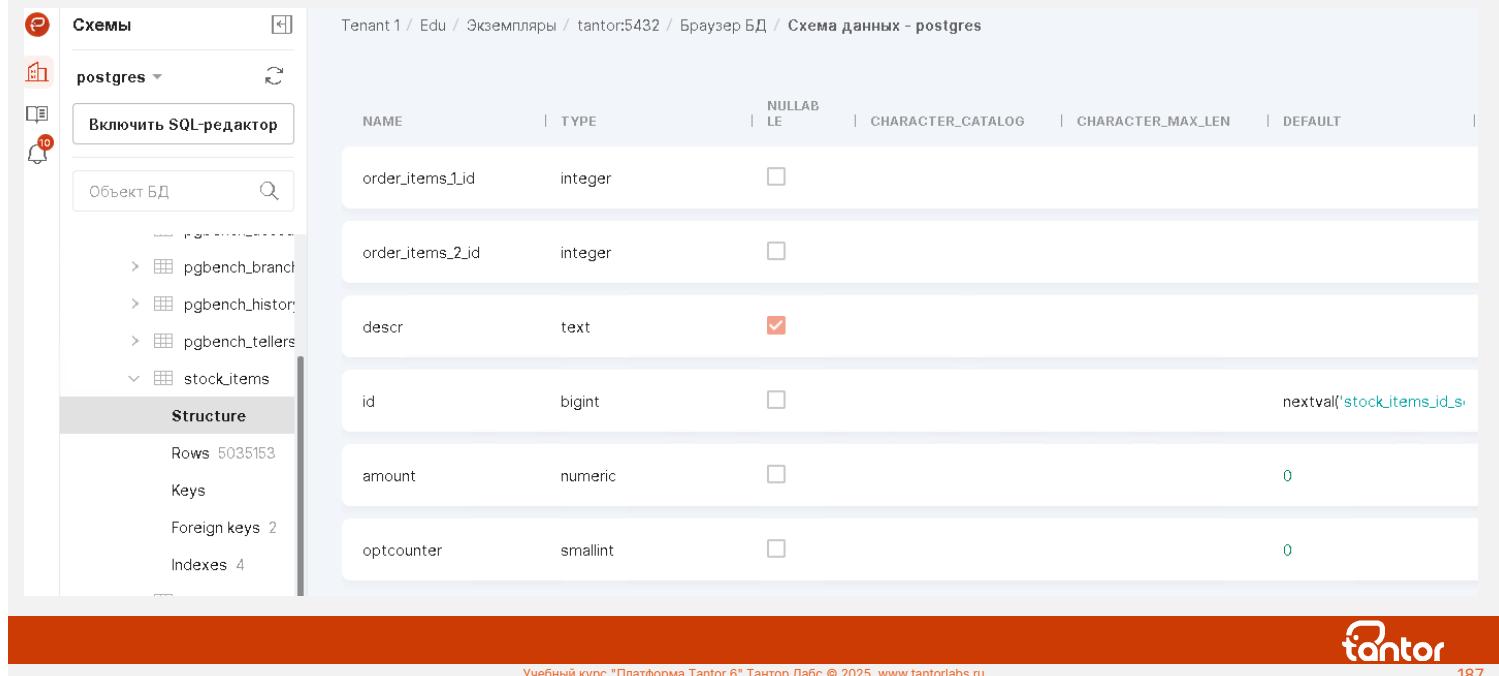
- Самые большие таблицы
- Таблицы с индексами максимального размера
- Топ записываемых таблиц
- Топ читаемых таблиц
- Таблицы с наибольшим раздутьем
- Таблицы с наибольшим кол-вом последовательных сканирований
- Таблицы с недопустимыми индексами
- Таблицы с неиспользуемыми индексами
- Таблицы с кандидатами для частичных индексов
- Индексы таблиц с значениями NULL > 50%
- Таблицы с избыточными индексами
- Таблицы с внешними ключами без индексов
- Таблицы у которых тип данных внешнего ключа отличен от исходного
- Таблицы с низким коэффициентом обновления НОТ
- Таблицы с возможными проблемами при автоанализе

stock_items

Статистика по таблице

При выборе вкладки СТАТИСТИКА откроется страница с накопительной статистикой по таблице. Внизу страницы - гистограмма распределения значений в столбцах таблицы. В выпадающем меню справа вверху плашки графика можно выбрать столбец таблицы, по которому будет показана гистограмма.

Браузер БД: Схема данных



The screenshot shows the PostgreSQL schema browser interface. On the left, a sidebar displays the database structure with a tree view of tables: pgbench_branch, pgbench_history, pgbench_tellers, and stock_items. The stock_items table is selected, showing its structure with columns: order_items_1_id, order_items_2_id, descr, id, amount, and optcounter. The 'descr' column is marked as not nullable (indicated by a red checkmark). The 'id' column has a default value of nextval('stock_items_id_seq'). The 'amount' column has a default value of 0. The 'optcounter' column has a default value of 0. The sidebar also shows 5035153 rows, 2 foreign keys, and 4 indexes.

NAME	TYPE	NULLABLE	CHARACTER_CATALOG	CHARACTER_MAX_LEN	DEFAULT
order_items_1_id	integer	<input type="checkbox"/>			
order_items_2_id	integer	<input type="checkbox"/>			
descr	text	<input checked="" type="checkbox"/>			
id	bigint	<input type="checkbox"/>			nextval('stock_items_id_seq')
amount	numeric	<input type="checkbox"/>			0
optcounter	smallint	<input type="checkbox"/>			0

Браузер БД: Схема данных

Можно посмотреть объекты баз данных, в том числе команды создания индексов, представлений, функций, процедур. Можно посмотреть данные в таблицах.

Браузер БД: SQL-редактор

⚠ Выполнение пользовательского запроса прекращено по тайм-ауту

⚠ Сбой выполнения пользовательского запроса

Tenant 1 / Edu / Экземпляры / tantor:5432 / Браузер БД / SQL-редактор - postgres

+ ЗАПРОС 1 ×

```
1 select * from stock_items limit 3;
```

Выполнить запрос

ID	ORDER_ITEMS_1_ID	ORDER_ITEMS_2_ID	AMOUNT	OPTCOUNTER	DESCR
31187976	8898	7976	255	12622	updated
33620314	115860	314	468	16591	updated
24895112	115830	5112	38	275	updated

Запрос отправлен на сервер

Экспортировать в CSV

tantor 188

Браузер БД: SQL-редактор

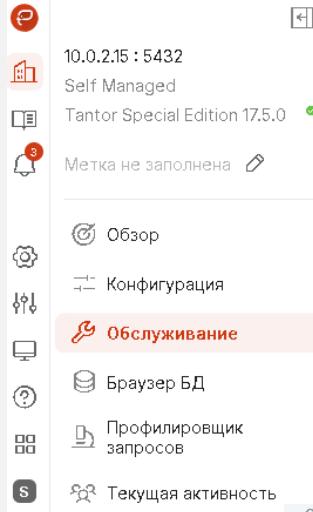
Нажав на кнопку "Включить SQL-редактор" появится окно для ввода команд SQL. Запросы, выдающие большое число строк могут прерваться по таймауту. Можно открыть несколько окон ЗАПРОС и вводить в них разные команды. В редакторе команды, если курсор находится на имени объекта, можно нажать комбинацию клавиш Ctrl+пробел и появится всплывающее окно со списком имён объектов, чтобы можно было выбрать, а не набирать имя на клавиатуре. На странице максимально показывается 1001 строка. Максимальное время выполнения запроса 30 секунд.



10-2

Обслуживание баз данных

Обслуживание баз данных

[Собрать данные](#)

Tenant 1 / Edu / Экземпляры / tantor:5432 / Обслуживание

[Обновить данные](#)[ОБСЛУЖИВАНИЕ](#) [ИСТОРИЯ](#)

БАЗА ДАННЫХ	ПРОЦЕНТ РАЗДУТИЯ ТАБЛИЦЫ %	РАЗДУТИЕ ИНДЕКСА %	ЗАЦИКЛИВАНИЕ %
postgres	13	67	0
test_db	3	7	

Зацикливание идентификатора транзакции

Раздутье индексов

Раздутье таблиц

* Дата последнего обновления : 26/09/2025 19:02

[ОБСЛУЖИВАНИЕ](#) [ИСТОРИЯ](#)

ИД ОПЕРАЦИИ ОБСЛУЖИВАНИЯ	БАЗА ДАННЫХ	СОЗДАНО	СТАТУС
2	postgres	26/09/2025 15:27:55	Completed

[Открыть](#)

190

Обслуживание баз данных

На странице Обслуживание экземпляра PostgreSQL две вкладки: ОБСЛУЖИВАНИЕ и ИСТОРИЯ. Если поля в таблице не заполнены, то нужно нажать на красную кнопку "Собрать данные". Если данные заполнены, то на кнопке будет написано "Обновить данные". На странице список баз данных. Ссылок на вкладке нет, кроме всплывающего меню из трёх пунктов: Зацикливание (Wraparound), Раздутье индексов, Раздутье таблиц.

На вкладке ИСТОРИЯ – история задач обслуживания vacuum, vacuum full, vacuum freeze, analyze, reindex, reindex concurrently, которые вызывались со страницы "Обслуживание" и со страницы "Браузер БД" → "Аудит". С этих страниц можно вызвать задачи обслуживания. Задачи обслуживания с этих страниц посылаются **не** через планировщик Платформы, хотя меню планировщика Платформы называется "Задачи".

На вкладке ИСТОРИЯ тоже нет ссылок, кроме одного пункта в контекстном меню задач "Открыть". Если кликнуть у задачи в истории на этот пункт откроется страница с деталями задачи:

Tenant 1 / Edu / Экземпляры / tantor:5432 / Обслуживание / ИД операции обслуживания 2

ИД операции обслуживания:	2	Создано:	26/09/2025 15:27:55
База данных:	postgres	Кем создано:	student
Действие:	Fix Tables Bloat	Завершено в:	26/09/2025 15:27:56
Статус:	Completed		

ОБСЛУЖИВАЕМЫЕ ОБЪЕКТЫ

ОБЪЕКТ	ДЕЙСТВИЕ	ЗНАЧЕНИЕ ДО	ЗНАЧЕНИЯ ПОСЛЕ / ОШИБОК
Type: Table Table: public.stock_items	ANALYZE SKIP_LOCKED	Table size : Bloat size : Bloat % :	Table size : 375 MB Bloat size : 52 MB Bloat % : 13.81

Запуск задачи обслуживания

Tenant 1 / Edu / Экземпляры / tantor:5432 / Обслуживание / postgres / Раздутие таблиц

Название таблицы	Раздутие	Раздутие %	Размер	После очистки	Строки	Последняя автоочистка	Последняя очистка
public.pgbench_tellers	608 kB	98.7	616 kB	8 kB	10	-	2025-09-26 14:14
public.pgbench_branches	352 kB	97.78	360 kB	8 kB	1	-	2025-09-26 14:14
public.pgbench_accounts	6816 kB	34.51	19 MB	13 MB	100000	-	
pg_catalog.pg_class	24 kB	20	120 kB	96 kB	450	-	

Выбранные: 2 Действие: VACUUM FULL ANALYZE Запустить обслуживание

Запуск задачи обслуживания

На странице "Раздутие таблиц" показывается процент раздутия, то есть на сколько процентов файлы таблицы занимают больше места, чем могли бы. Для уменьшения файлов можно использовать команды vacuum full или cluster.

При выборе таблиц галочками, внизу страницы появится всплывающее окно с выбором команды, параметров команды и кнопкой "Запустить обслуживание".

При нажатии на кнопку появится всплывающее окно с предупреждением, что на время выполнения vacuum full и reindex таблицы и индексы будут заблокированы на время до получения блокировки плюс время выполнения команды.

У команды analyze только одна опция skip_locked. У команды vacuum full тоже одна опция analyze. У команды vacuum много опций, но не все.

Желтый треугольник означает всего лишь предупреждение, что в столбце ПОСЛЕДНЯЯ ОЧИСТКА не учитывается vacuum full, только обычный vacuum.

Розовым цветом в столбце РАЗДУТИЕ % выделяются значения с существенным процентом раздутия. Бледно-розовым цветом выделяются строки, выбранные галочкой.

Статус задач обслуживания

Tenant 1 / Edu / Экземпляры / tantor:5432 / Обслуживание

The screenshot shows the Tantor database management interface. At the top, there are tabs for 'ОБСЛУЖИВАНИЕ' (Maintenance) and 'ИСТОРИЯ' (History). A red button 'Собрать данные' (Collect data) is visible. Below this, a table lists databases: 'postgres' and 'test_db'. A purple notification bar at the bottom of the screen says 'Внимание! Идет техническое обслуживание. Просмотреть детали' (Attention! Technical maintenance is in progress. View details). In the background, a modal window titled 'Статус операции обслуживания' (Maintenance operation status) shows a table with one row: 'public.stock_items_pk' (Operation: REINDEX, Status: In Progress). A progress bar is shown next to the status. At the bottom of this modal are buttons 'Остановить обслуживания' (Stop maintenance) and 'Закрыть' (Close). A large confirmation dialog box is overlaid on the interface, containing an 'i' icon, the title 'Внимание!', a message 'Это действие отменит все операции обслуживания. Вы хотите продолжить?' (This action will cancel all maintenance operations. Do you want to continue?), an input field 'продолжить' (Continue), and buttons 'Отменить' (Cancel) and 'Остановить обслуживания' (Stop maintenance).

Статус задач обслуживания

После запуска процедуры обслуживания в правом нижнем углу появится фиолетовое уведомление: "Внимание! Идет техническое обслуживание. Посмотреть детали":

Внимание! Идет техническое обслуживание. Просмотреть детали

Если кликнуть на ссылку в уведомлении, откроется страница "Статус операции обслуживания", на которой можно узнать на какой стадии находится задача. Задачи могут быть в очереди, уже запущены или завершены. В таблице будет присутствовать столбец "Process" с индикатором, отображающим **прогресс** выполнения операции. Задачу можно прервать, нажав на кнопку "Остановить обслуживание"/"Stop maintenance". Если статус задачи "Начало"/"Started", то процесс уже запущен и не будет прерван.

Уведомление появится у **всех** пользователей, находящихся на странице "Обслуживание" экземпляра, на котором запущены задачи. Если уведомление уже появилось, то оно будет присутствовать на всех страницах, пока задача не будет завершена. Пока задача обслуживания не завершится, меню на странице Обслуживания не активно.

Пока она не завершится другой процесс обслуживания не запустится, на экземпляре может работать одна команда обслуживания.

Обслуживание индексов и счетчика транзакций

Tenant 1 / Edu / Экземпляры / tantor:5432 / Обслуживание / postgres / Раздутие индексов

The screenshot shows the Tantor PostgreSQL management interface. The top section is titled 'Раздутие индексов' (Index Bloat) and displays a table of index fragmentation. The table includes columns: НАЗВАНИЕ ТАБЛИЦЫ (Table Name), НАЗВАНИЕ ИНДЕКСА (Index Name), РАЗДУТИЕ (Bloat), КОЭФФ. РАЗДУТИЯ % (Bloat %), ВСЕГО СВОБОДНО (Free Space), КОЭФФ. ЗАПОЛНЕНИЯ (Fill Factor), and ТРЕБУЕТСЯ АВТОВАКУУМ (Autovacuum Required). It lists three indexes: 'stock_items_pk' (84.34%), 'pgbench_accounts_pkey' (50.09%), and 'order_items_2_pkey' (0.4%). The 'order_items_2_pkey' row has a dropdown menu with 'CONCURRENTLY' and 'INDEX' options, and a button 'Запустить обслуживание' (Run Maintenance). The bottom section is titled 'Зацикливание идентификатора транзакции' (Transaction ID Recycling) and shows a table of transaction IDs. It includes columns: НАЗВАНИЕ ТАБЛИЦЫ (Table Name), ТЕКУЩИЙ XID (Current XID), К ЗАЦИКЛИВАНИЮ (Recycling), К АВАРИЙНОЙ АВТООЧИСТКЕ (Emergency Auto-Clean), and РАЗМЕР (Size). It lists 'pg_catalog.pg_class' (XID 2779998, Recycling 0, Emergency 1, Size 248 kB) and 'pg_catalog.pg_proc' (XID 2779998, Recycling 0, Emergency 1, Size 1272 kB). A sidebar on the left shows system status: 10.0.2.15 : 5432 (Self Managed, Tantor Special Edition 17.5.0), and a 'Обслуживание' (Maintenance) section. The footer includes a navigation bar with 'Обслуживание' (Maintenance) and 'Обзор' (Overview), and a footer note: 'Учебный курс "Платформа Tantor 6" Тантор Лабс © 2025 www.tantolabs.ru'.

Обслуживание индексов, счетчика транзакций и мультитранзакций

На странице "Раздутие индексов" всё аналогично таблицам, только показывается процент раздутия индексов. Для уменьшения размера файлов индекса используются команды `reindex concurrently` или `reindex`.

Несколько `REINDEX` (без `CONCURRENTLY`) могут одновременно работать по одной таблице пересоздавая разные индексы. `REINDEX CONCURRENTLY` может перестраивать только один индекс и не может работать одновременно с другими `REINDEX` по этой же таблице.

При выборе `CONCURRENTLY`, PostgreSQL создаст новый индекс, а после его проверки переключит таблицу на новый индекс, удалив старый.

Команда, которая не может одновременно работать с другой выполняющейся командой будет ждать, либо выдаст ошибку после таймаута, заданного параметром `lock_timeout`.

Розовым цветом в столбце КОЭФФ. РАЗДУТИЯ % выделяются значения с существенным процентом раздутия. Бледно-розовым цветом выделяются строки, выбранные галочкой.

Проценты в столбцах К ЗАЦИКЛИВАНИЮ и К АВАРИЙНОЙ АВТООЧИСТКЕ на основе параметров экземпляра `autovacuum_freeze_max_age` и `vacuum_failsafe_age`.

По умолчанию заморозка начинается, когда она не была выполнена `autovacuum_freeze_max_age` (для PostgreSQL с 32-битным счетчиком транзакций по умолчанию 200 млн.) транзакций, отстоящих от самой последней транзакции. Если заморозка не успеет выполниться и для PostgreSQL с 32-битным счетчиком транзакций пройдёт 2 млрд транзакций, то их номера перестанут выдаваться до тех пор, пока заморозка не будет выполнена.

По умолчанию процессы автовакуума, выполняющие заморозку, работают с задержкой, заданной параметром конфигурации `autovacuum_vacuum_cost_delay`. Они могут не успевать обрабатывать таблицы. Задержку стоит отключить, но по умолчанию она включена.

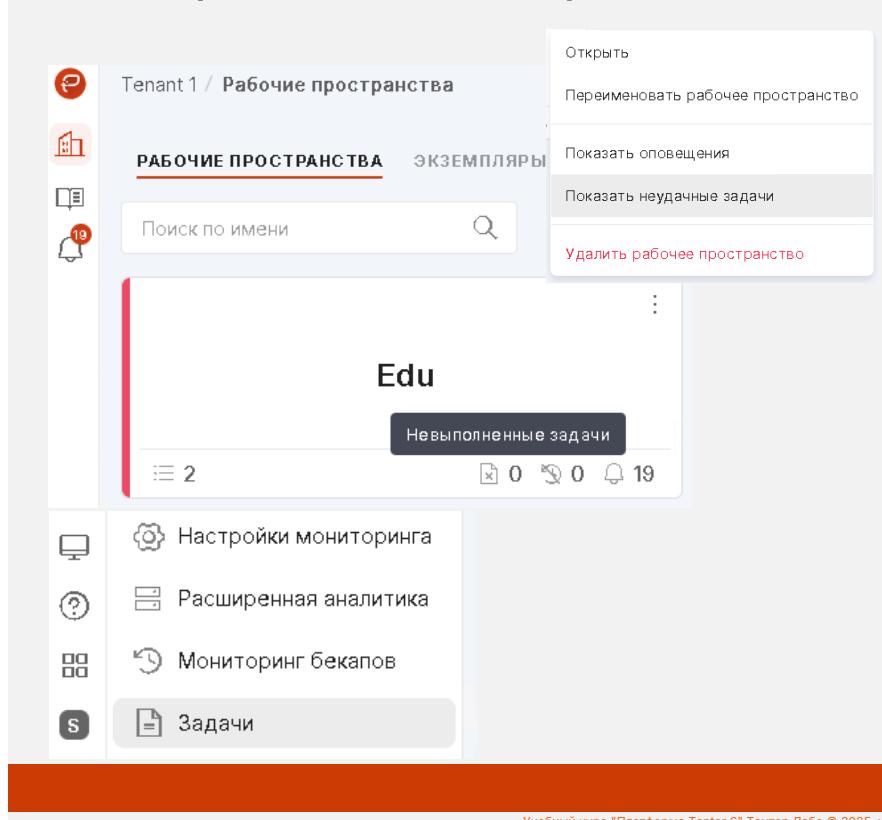
Игнорироваться она начинает только по достижении `vacuum_failsafe_age` транзакций (для PostgreSQL с 32-битным счетчиком транзакций по умолчанию 1 600 000 000), то есть, когда останется 400 млн. транзакций до аварийной остановки экземпляра, которая произойдёт, если заморозка не успеет завершиться. В Tantor Postgres SE и SE 1С используется 64-битный счетчик транзакций и зацикливание счетчика транзакций на уровне экземпляра неактуально, хотя заморозка выполняется.



10-3

Планировщик Платформы Тантор

Планировщик Платформы



Tenant 1 / Рабочие пространства

РАБОЧИЕ ПРОСТРАНСТВА ЭКЗЕМПЛЯРЫ

Поиск по имени

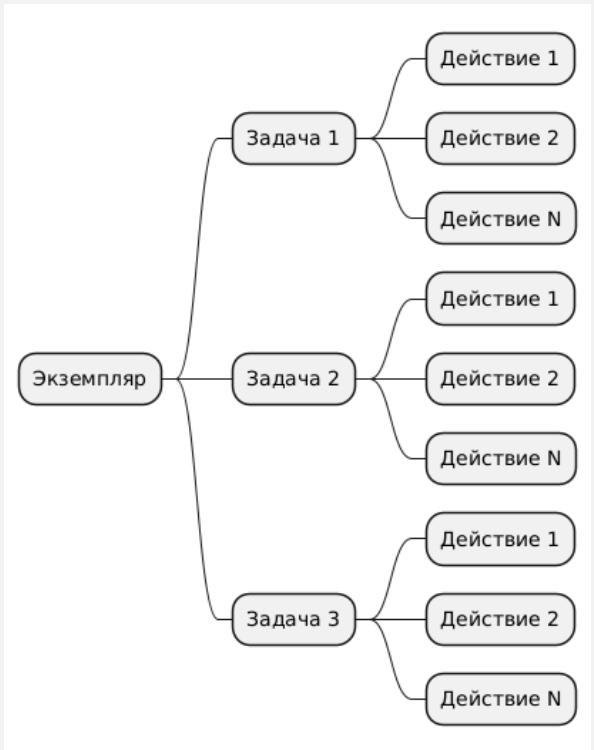
Невыполненные задачи

Настройки мониторинга

Расширенная аналитика

Мониторинг бекапов

Задачи



Учебный курс "Платформа Tantor 6" Тантор Лабс © 2025 www.tantolabs.ru

Планировщик Платформы

Планировщик задач (Job Scheduler) позволяет запускать:

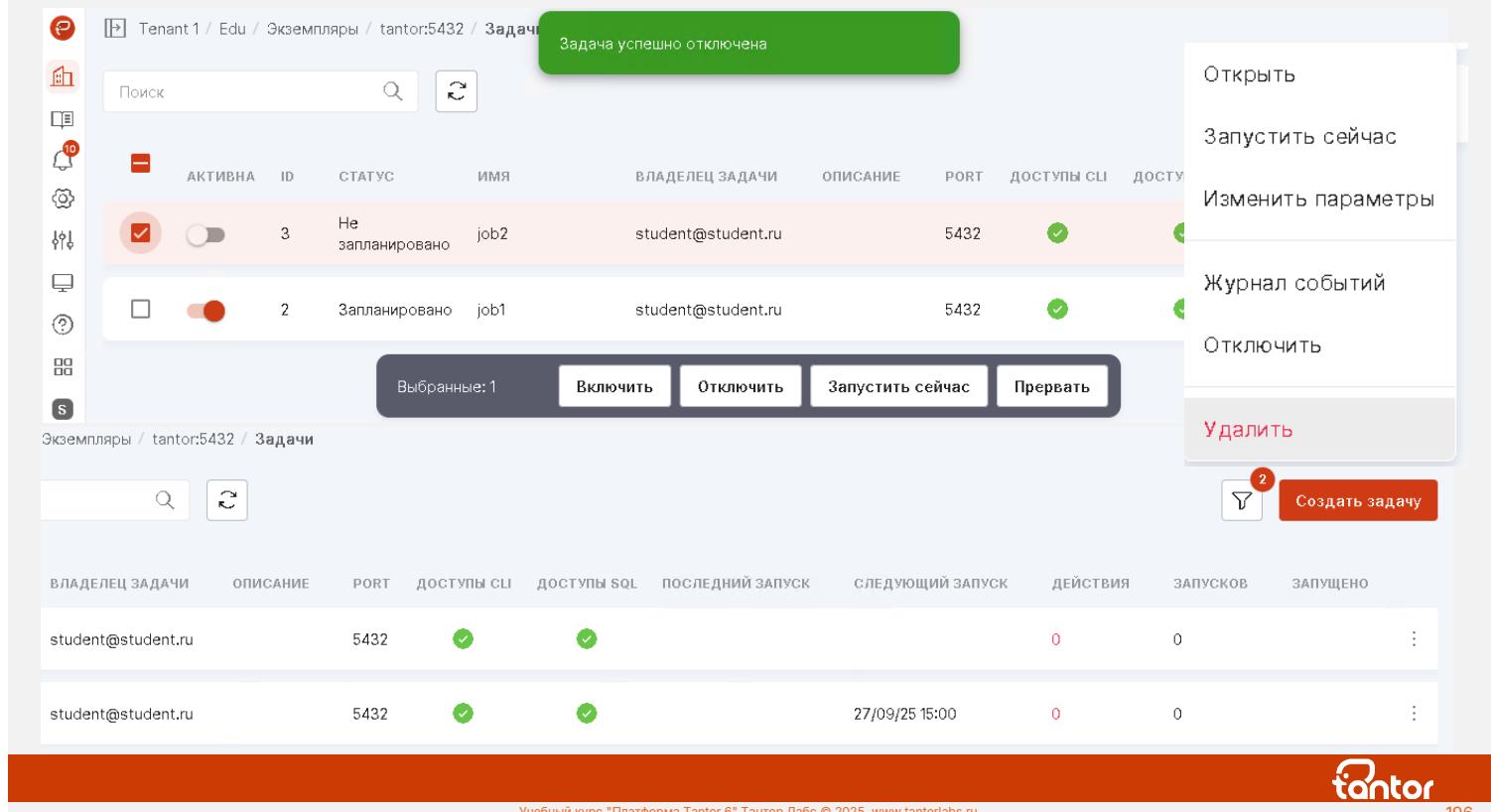
- 1) команды SQL на экземпляре PostgreSQL
- 2) командные скрипты linux на хосте экземпляра.

Перейти на страницу планировщика со списком задач можно:

со страницы пространств: кликнуть на значок листа с крестиком "Невыполненные задачи" или кликнуть на три точки и выбрать в появившемся меню строку "Показать неудачные задачи"; через пункт "Задачи" в самом конце меню экземпляра.

На экземпляре может быть назначено много задач. Задача может состоять из нескольких шагов (действий).

Страница со списком задач



ВЛАДЕЛЕЦ ЗАДАЧИ	ОПИСАНИЕ	PORT	ДОСТУПЫ CLI	ДОСТУПЫ SQL	ПОСЛЕДНИЙ ЗАПУСК	СЛЕДУЮЩИЙ ЗАПУСК	ДЕЙСТВИЯ	ЗАПУСКОВ	ЗАПУЩЕНО
student@student.ru		5432	✓	✓			0	0	
student@student.ru		5432	✓	✓	27/09/25 15:00		0	0	

Страница со списком задач

В таблице со списком задач много столбцов, которые могут не поместиться в окне браузера и можно использовать полосу горизонтальной прокрутки окна браузера.

При выборе галочкой задачи внизу страницы появится меню с кнопками: Включить, Отключить, Запустить сейчас, Прервать.

Ползунком в столбце Активна можно отключать-включать задачу. Если задача неактивна, то она не запустится ни по расписанию, ни вручную.

Меню управления задачей находится в самом конце страницы справа, вызывается при нажатии на значок с тремя точками. Если таблица не помещается в окне браузера, то чтобы добраться до значка нужно воспользоваться полосой прокрутки.

Столбец ДЕЙСТВИЯ - число действий (шагов) в задаче. Ноль действий - задаче нечего выполнять.

Столбец ЗАПУЩЕНО - кто запустил задачу: пользователь или таймер.

В столбце СТАТУС задачи может быть текст:

Запланировано;

Не запланировано;

В процессе - задача сейчас выполняется;

Успешно - успешно выполнена или ошибки на действиях задачи были проигнорированы;

Прервано - была прервана нажатием кнопки "Прервать сейчас";

Тайм-аут - прервана из-за превышения таймаута;

Ошибка - в процессе выполнения была неигнорируемая ошибка;

Ожидание выполнения ;

Неизвестный тип - задача, для которой нельзя определить статус, указывает на то, что что-то не работает.

Добавление задачи

Добавление задачи

Рабочее пространство* Edu

Экземпляр* tantor | 10.0.2.15 | 5432

Введите название задачи* job1

Описание задачи

Пароль* ****

Показать доступы (требуются для запуска)

Доступы CLI

Имя пользователя root

Пароль* ****

Доступы SQL

Имя пользователя postgres

Пароль* *****

Дата и время начала* Выберите дату

Начало должно быть позже текущего времени

Повторить

Количество повторений (0 = бесконечность)* 1

Интервал между запусками* Годы Месяцы Недели Дни Часы Минуты Секунды 24

Сохранить

Отменить Сохранить

SEP 2025

1 2 3 4 5 6

7 8 9 10 11 12 13

14 15 16 17 18 19 20

21 22 23 24 25 26 27

28 29 30

11 : 11

197 tantor

Добавление задачи

Добавить задачу можно на странице "Задачи". Также можно вызывать окно добавления задачи со страницы экземпляров, кликнув на три точки и выбрав в выпадающем меню "Добавить задачу".

При добавлении задачи обязательно указать:

пространство, экземпляр. Они нужны, чтобы определить какой Агент Платформы будет посыпать задачу на выполнение;

1) доступы CLI (имя и пароль пользователя linux) - из под кого будут выполняться действия в linux. Если не указать, то скрипты в linux не смогут выполняться;

2) доступы SQL (имя и пароль пользователя PostgreSQL) - из под кого будут выполняться команды SQL. Если не указать, то команды SQL не смогут выполняться;

3) дату и время, когда задача начнет выполняться. Нужно указать момент в будущем. После создания задачи её можно будет запускать вручную.

Tenant 1 / Edu / Экземпляры

КЛАСТЕРЫ ЭКЗЕМПЛЯРЫ АГЕНТЫ

Добавить экземпляр

ИМЯ УЗЛА	ОС	ТИП	ИЗДАНИЕ	ВЕРСИЯ РБ	ЦПУ	МЕТКА	IP	ПОРТ	⋮
tantor	Astra Linux(1.7_x86-64)	Tantor	Tantor Special Edition	17.5.0	4		10.0.2.15	5432	⋮
tantor	Astra Linux(1.7_x86-64)	Tantor	Tantor Special Edition	17.5.0	4	Открыть			
						Открыть расширенную аналитику			
						Тестовое подключение			
						Переместить в другое пространство			
						Добавить задачу			
						Добавить метку			
						Браузер БД			
						Удалить			

Добавление действий в задачу

Пока нет действий

Создайте хотя бы одно действие для задачи

Добавить действие

Системная команда

SQL

Добавление действия SQL

Все команды будут выполнены от имени postgres

Название действия*

step2

Максимум 50 символов

5 / 50

Выберите БД*

postgres

Введите SQL-запрос:*

select 1;

Отменить

Сохранить

Добавление действия «Системная команда»

Все команды будут выполнены от имени postgres. Обратите внимание, что указывать доступы в команде не нужно

Название действия*

step1

Максимум 50 символов

5 / 50

Команда*

pgbench -T 300 -P 60 -c 5

Максимум 255 символов

25 / 255

Действие в случае ошибки

Остановить задачу



Игнорировать ошибку и продолжить задачу

Отменить

Сохранить

tantor

198

Учебный курс "Платформа Tantor 6" Тантор Лабс © 2025 www.tantorlabs.ru

Добавление действий в задачу

На слайде показы окна добавления двух видов действий: SQL и команд в linux.

Редактирование действий задачи

Tenant 1 / Edu / Экземпляры / tantor:5432 / Задачи / job2

Поиск

↻

✖

Журнал событий

Запустить задачу

Добавить действие

АКТИВНО	СТАТУС	ИМЯ	ПАРАМЕТРЫ	ТИП	⋮
<input checked="" type="checkbox"/>	Created	step1	pgbench - T 300 - P 60 - c 5	command	⋮
<input type="checkbox"/>	Created	step2	SELECT 1;	sql	⋮

Выбранные: 1ВключитьОтключить

Частота обновления данных - 15 сек

Переместить вверх

Переместить вниз

Изменить параметры

Отключить

Удалить действие

Редактирование действий задачи

Если выбрать сортировку по столбцу СТАТУС - отобразится порядок выполнения действий.

В столбце СТАТУС может быть:

Создано - созданные, но еще не выполненные действия;

Ошибка - выполненные с ошибкой;

В процессе - действия в процессе выполнения;

Успешно - успешно выполненные;

Прервано - прерванные в процессе выполнения;

Тайм аут.

Задачи в Журнале событий

Tenant 1 / Edu / Экземпляры / tantor:5432 /

Задача успешно запущена. Перейти в список текущих активностей

Помощь

Поиск

ВЛАДЕЛЕЦ ЗАДАЧИ ОПИСАНИЕ PORT

student@student.ru 5432

student@student.ru 5432

Запустить задачу(и)?

Это запустит 1 задачу.

Для подтверждения введите "продолжить"*

продолжить

Отменить Запустить

Выбранные: 1 Включить Отключить Запустить сейчас Прервать

АПУСК СЛЕДУЮЩИЙ ЗАПУСК ДЕЙСТВИЯ ЗАПУСКОВ ЗАПУЩЕНО

2 227/09/25 15:00 0 2

Открыть

Запустить сейчас

Изменить параметры

Журнал событий

Отключить

Удалить

10.0.2.15 : 5432 Self Managed Tantor Special Edition 17.5.0 Метка не заполнена

Обзор Конфигурация Обслуживание Браузер БД Профилировщик запросов Текущая активность

ПОЛЬЗОВАТЕЛИ СИСТЕМА

ВСЕГО 39/100 ИСПОЛЬЗУЕТСЯ % 39 ЗАРЕЗЕРВИРОВАННО СУПЕРПОЛЬЗОВАТЕЛЕЙ

Поиск

БАЗА ДАННЫХ	ПОЛЬЗУЕТСЯ	ИСПОЛЬЗУЕТСЯ %	МАКС. КОЛ-ВО СЕССИЙ	АКТИВНО	IDLE	IDLE IN TX	IDLE IN TX (ABORTED)	FASTPATH	ОТКЛЮЧЕННО	НЕТ ПРИВИЛЕГИЙ
postgres	39	39	100	11	1	1	0	0	0	0

tantor 200

Задачи в Журнале событий

Активную задачу можно запустить. Либо выбрав галочкой задачу и нажав во всплывающем окне "Запустить сейчас" или кликнуть на три точки справа в строке задачи и в появившемся меню выбрать "Запустить сейчас".

Ненадолго появится сообщение на зелёном фоне со ссылкой на страницу "Текущая активность". Если успеть, то можно кликнуть на ссылку. Если не успеть, то страницу можно открыть в меню Экземпляра. Эта страница удобна для просмотра сессий и блокировок. Для наблюдения за процессом выполнения задачи удобна страница "Журнал событий" (пункт меню на слайде). В нём можно посмотреть сообщения с действий задачи.

Фильтры в Журнале событий

The screenshot shows the Tantor 6 event log interface. On the left, a sidebar has icons for Tenant 1, Journal, and Notifications (2). The main area shows a table of events with columns: Создано в (Created at), ИД События (Event ID), Тип События (Event Type), Ресурс (Resource), and ИД Ресурса (Resource ID). Two events are listed: one with ID 389 and another with ID 388. A filter sidebar on the right is set to 'TASK' for 'Type of event' and '3' for 'Resource ID'. A message at the top says 'Передача данных syslog: не настроена' (Syslog data transmission is not configured). Below the table, a detailed view for event ID 389 shows its creation time (27/09/2025 14:06:08), type (TASK), resource (INSTANCE), resource ID (3), and creator (Unknown). The 'ИНФОРМАЦИЯ О СОБЫТИИ' (Information about the event) section shows StatusID (55), StatusName (Error), and ExecutionResult (Task [job2]:[433a962f-00ba-46e6-b5d7-0b20fe9f1651] has no actions). A red 'Очистить фильтры' (Clear filters) button is at the bottom of the filter sidebar.

Фильтры в Журнале событий

Страница Журнала общая для всех сообщений, а не только для задач. При открытии Журнала через меню задач устанавливается Фильтр по типу события = TASK и экземпляру.

В Журнале показывается результат команд (stdout, errout) для обоих типов команд linux и SQL:

ИНФОРМАЦИЯ О СОБЫТИИ

Action 1	Command
Id :	a0fc5eaf-e932-4a82-982c-88a3338a8554
Name :	step1
StatusID :	54
StatusName :	Success
Command :	pgbench -T 300 -P 60 -c 5
DbName :	
Type :	command
OnError :	fail
IsDisabled :	false
ExecutionResult :	Password: pgbench (17.5) starting vacuum...end. progress: 60.0 s, 1.4 tps, lat 3335.191 ms stddev 3174.612, 0 failed progress: 120.0 s, 1.5 tps, lat 3368.508 ms stddev 3138.413, 0 failed progress: 180.0 s, 1.5 tps, lat 3253.193 ms stddev 3269.200, 0 failed
Action 2	Sql
Id :	cb63cd80-3776-4cd1-af3b-253069dfca5a
Name :	step2
StatusID :	54
StatusName :	Success
Command :	select 1;
DbName :	postgres
Type :	sql
OnError :	fail
IsDisabled :	false
ExecutionResult :	[1]

Экземпляр: страница Репликация у мастера

The screenshot shows the Tantor 6 interface with the 'Replication' page open for a master instance. The left sidebar includes icons for Overview, Configuration, Maintenance, Browser, Profiler, Current Activity, and Replication (which is selected). The main content area has two tabs: 'Горячий резерв' (Hot Standby) and 'Слоты' (Slots). The 'Слоты' tab is active, showing two slots: 'Postgres (17.3):172.21.1.2:5432' and 'Postgres (17.3):172.21.1.3:5432', both in 'streaming' mode with 'async' replication and 0 ms latency. A 'Replication' sub-tab is also visible in the top navigation bar.

Экземпляр: страница Репликация у мастера

На странице Репликация у мастера (Primary, основной, ведущий кластер) две вкладки.

На вкладке СЛОТЫ - список слотов и их статус, на этой вкладке ссылок нет.

На вкладке ГОРЯЧИЙ РЕЗЕРВ показан список реплик. При клике на строке или при выборе в меню "Открыть" появится страница со статистикой слота репликации:

The screenshot shows the detailed Replication page for the slot 'Postgres (17.3):172.21.1.2:5432'. The left sidebar is identical to the previous screenshot. The main content area displays various details about the slot, including its status as 'active', the primary as 'Postgres (17.3):172.21.1.2:5432', the user as 'replicator', and the application as 'b53657ba2894'. It also shows the slot's state as 'Streaming', its mode as 'Async', and its process ID as '73'. On the right, a 'ЗАДЕРЖКИ:' (Latency) section provides metrics: 0 for 'В ожидании:' (Waiting), 0 for 'Запись:' (Write), 0 for 'Синхронизация изменений:' (Change synchronization), 0 for 'Воспроизведение:' (Replay), and 0 for 'Всего:' (Total). The bottom of the page has a 'Служебные' (Utility) section with links for 'Синхронизация' (Sync), 'Воспроизведение' (Replay), and 'Сброс' (Reset).

Экземпляр: пункт меню Репликация на странице реплики

The screenshot shows the Tantor 6 PostgreSQL interface. On the left, a sidebar menu lists various options: Проверка (Check), 172.21.1.3 : 5432 (Patroni 4.0.4 replica, Postgres 17.3), Метка не заполнена (Mark not filled), Обзор (Overview), Конфигурация (Configuration), Браузер БД (DB Browser), Профилировщик запросов (Query Profiler), Текущая активность (Current Activity), **Репликация** (Replication), Табличные пространства (Tablespaces), Графики (Graphs), and Настройки мониторинга (Monitoring Settings). The 'Репликация' option is highlighted with a red box. The main content area is titled 'СТАТУС' (Status). It shows the following details:

Статус:	Active
Основной экземпляр:	Postgres (17.3):172.21.1:5432
Слот репликации:	71F9940F164B
Задержка репликации:	0

On the right, a section titled 'ИНФОРМАЦИЯ О ПОДКЛЮЧЕНИИ' (Connection Information) lists the following connection parameters:

user:	replicator
passfile:	/tmp/pgpass0
channel_binding:	prefer
dbname:	replication
host:	172.21.1
port:	5432
application_name:	71f9940f164b
fallback_application_name:	patroni
sslmode:	prefer
sslnegotiation:	postgres
sslcompression:	0
sslcertmode:	allow
sslsni:	1
ssl_min_protocol_version:	TLSv1.2
gssencmode:	prefer
krbsrvname:	postgres
gssdelegation:	0
target_session_attrs:	any
load_balance_hosts:	disable

Учебный курс "Платформа Tantor 6" Тантор Лабс © 2025 www.tantorlabs.ru

Экземпляр: пункт меню Репликация на странице реплики

У реплик (Standby, резервные, ведомые кластера) на странице Репликация только одна вкладка СТАТУС и содержимое этой страницы другое. На ней детали подключения к источнику WAL.

Настройка тенанта: Резервное копирование

The screenshot shows the Tenant 1 / Настройки тенанта / Резервное копирование page. On the left, a sidebar lists various tenant settings. The 'Резервное копирование' (Backup) item is highlighted. A modal window titled 'Добавление подключения' (Add connection) is open, showing a dropdown for 'Система резервного копирования' (Backup system) with 'Backman' selected. Below it, 'RU_BACKUP' is listed. The 'Порт' (Port) field is empty. The 'Имя пользователя' (User name) and 'Пароль' (Password) fields are also empty. The 'Название базы данных' (Database name) field is empty. At the bottom of the modal are 'Закрыть' (Close) and 'Добавить' (Add) buttons. To the right, a list of connections shows 'rubackup' with an 'Изменить' (Edit) button and a red 'Удалить подключение' (Delete connection) button.

Настройка тенанта: Резервное копирование

В 6 версии Платформы появилась возможность для тенанта добавить одно соединение с сервером резервирования RuBackup и Backman. Это делается на странице **Настройки тенанта -> Резервное копирование**. Если соединение не создано, то на странице будет красная кнопка "Добавить подключение", если создана, то будет строка с параметрами подключения. Через всплывающее меню можно удалить или поменять параметры соединения.

<https://docs.tantorlabs.ru/tp/6.1/admin/backup.html>

Мониторинг бэкапов

The screenshot shows the Tantor 6 platform interface. On the left, there is a sidebar with various icons and a list of modules: Модули, Анализ плана запросов, Анонимайзер, SQL Редактор, Схема данных, Группы параметров, Мониторинг бэкапов (selected), Задачи, Сваггер, and Все модули. The main content area is titled 'Tenant 1 / Рабочие пространства' and shows a table of backup monitoring data. The table has columns: FQDN/IP, РАБОЧЕЕ ПРОСТРАНСТВО, КОЛ-ВО ПОЛНЫХ РК, КОЛ-ВО ИНКРЕМЕНТ. РК, ДАТА ПОСЛЕДНЕЙ РК, РАЗМЕР ПОСЛЕДНЕЙ РК, and СТАТУС ПОСЛЕДНЕЙ РК. There are three rows of data: platform-qa-db-004.ta... (Standalone, 5 full backups, 1 incremental, 0.07/07/2025 21:43, 10.64 MB, Неуспешно), platform-qa-db-001.ta... (Standalone, 18 full backups, 0 incremental, 05/07/2025 14:41, 10.64 MB, Неуспешно), and platform-qa-db-px-001... (Standalone, 0 full backups, 0 incremental, -, -, -). Below the table, there is a section titled 'Текущая активность' with links: Репликация, Табличные пространства, Графики, Настройки мониторинга, Расширенная аналитика, and Мониторинг бэкапов (selected). To the right, there is a modal window titled 'Tenant 1 / Рабочие пространства' with a sub-tab 'РАБОЧИЕ ПРОСТРАНСТВА' selected. It shows a table with a single row for 'Edu' with the status 'Невыполненные бэкапы'. The bottom right corner of the interface features the Tantor logo and the number 205.

Мониторинг бэкапов

На страницу "Мониторинг бэкапов" можно попасть:

- 1) через пункт в меню "Модули"
- 2) через пункт в меню Экземпляра, рядом с пунктом Задачи
- 3) через ссылку "Невыполненные бэкапы" на странице "Пространства" тенанта.

На странице показывается таблица со списком бэкапов. Графические элементы на странице стандартны и встречаются на многих страницах платформы:

- 1 - Поиск по первому столбцу таблицы
- 2 - обновление страницы
- 3 - фильтрация строк, показываемых в таблице
- 4 - выбор столбцов, которые будут отображаться
- 5 - названия столбцов

https://docs.tantolabs.ru/tp/6.1/admin/backup_monitoring.html

Практика 10

- a. Откройте **Браузер БД** и посмотрите результаты проверок и рекомендации на странице **Аудит**
- b. Откройте страницу **Обслуживание** и выполните полное вакуумирование таблицы `stock_items`
- c. Откройте страницу **Задачи** и выполните существующую задачу, затем создайте новую задачу и выполните ее

Практика 10

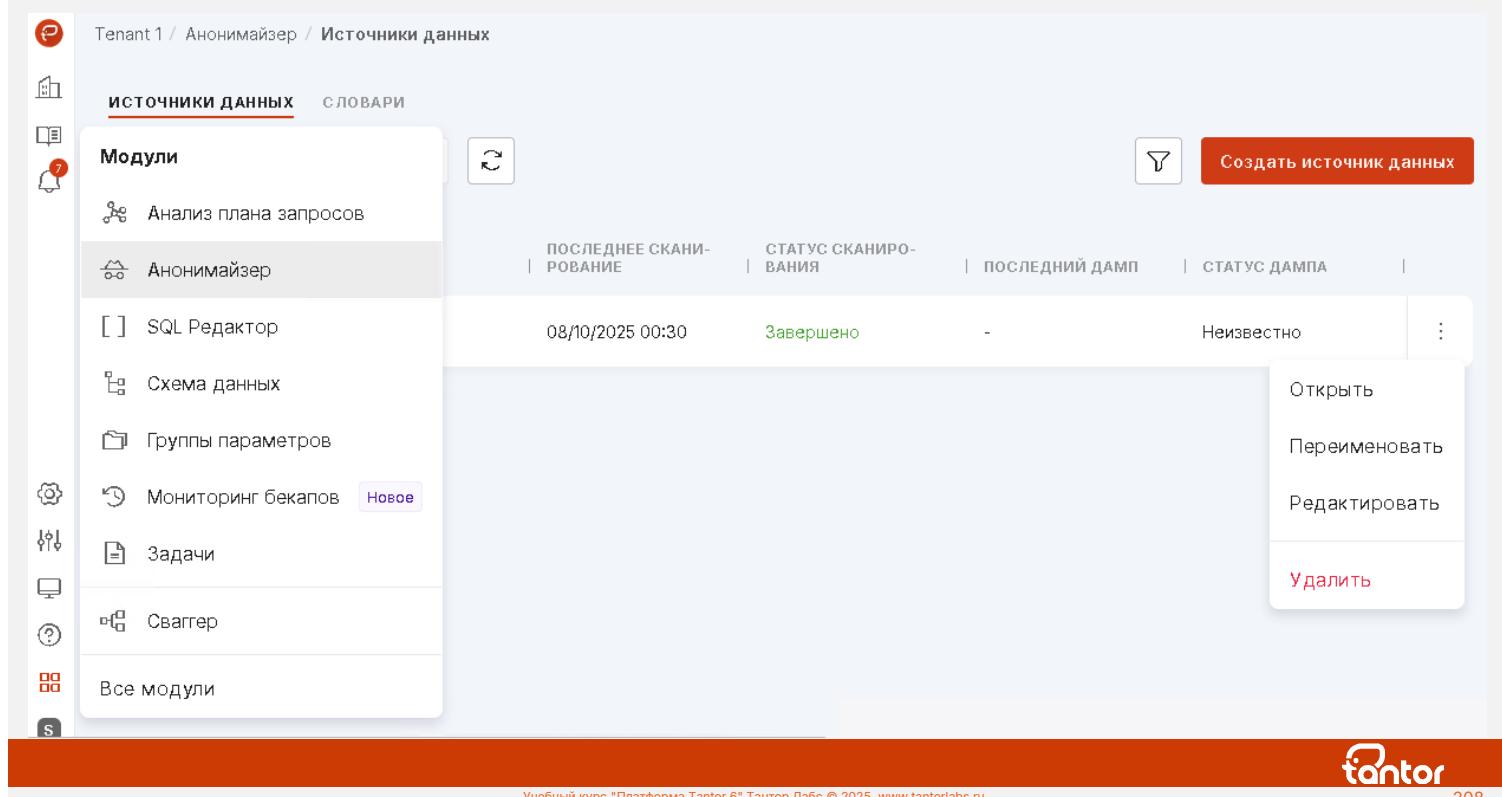
- a. Откройте **Браузер БД** и посмотрите результаты проверок и рекомендации на странице **Аудит**
- b. Откройте страницу **Обслуживание** и выполните полное вакуумирование таблицы `stock_items`
- c. Откройте страницу **Задачи** и выполните существующую задачу, затем создайте новую задачу и выполните ее



Модуль Платформы

Анонимайзер (маскировка данных)

Обзор анонимайзера



ПОСЛЕДНЕЕ СКАНИРОВАНИЕ	СТАТУС СКАНИРОВАНИЯ	ПОСЛЕДНИЙ ДАМП	СТАТУС ДАМПА
08/10/2025 00:30	Завершено	-	Неизвестно

Обзор анонимайзера

Анонимайзер – визуальный интерфейс к утилите pg_anon. Утилита есть как в Платформе Tantor, так и в Tantor Postgres. Анонимизация или маскировка данных – это:

- 1) поиск столбцов, в которых могут храниться конфиденциальные данные. Например, имена, телефоны, баланс счетов, адреса
- 2) Замена данных в таких столбцах на похожие, с сохранением связей между таблицами.
- 3) Выгрузка утилитой pg_dump данных в изменённом виде.

Анонимизация позволяет получить копию (дамп) таблиц, который можно передать подрядчикам для анализа работы программы, работающей с данными. Данные по объему похожи на реальные, но не содержат конфиденциальных данных.

Анонимайзер аналогичен функционалу маскировки данных в Oracle Enterprise Manager Cloud Control.

Ссылка на анонимайзер находится в пункте меню Модули (четыре квадрата слева внизу линейки меню). На странице Анонимайзера две вкладки: ИСТОЧНИКИ ДАННЫХ и СЛОВАРИ.

pg_anon установлен в контейнере pg_anon Платформы. Устанавливать утилиту на хосты баз данных не обязательно.

Можно установить pg_anon и настроить REST-интерфейс по документации:
https://docs.tantolabs.ru/tp/6.1/admin/pg_anon_stateless.html

Словари анонимайзера

Tenant 1 / Анонимайзер / Словари

ИСТОЧНИКИ ДАННЫХ СЛОВАРИ

Поиск по имени

Создать словарь

НАЗВАНИЕ	ТИП	СКАНИРОВАНИЕ	ВАРИАНТ	СОЗДАЛ	ПОСЛЕДНИЕ ИЗМЕНЕНИЯ	⋮
dict2u	Пользовательский	scan2	Несенситивный		08/10/2025 23:17	⋮
meta-dict1	Пользовательский		Мета	student	09/10/2025 00:00	⋮
sens1	Пользовательский		Сенситивный		09/10/2025 00:00	⋮
dict2s	Пользовательский	scan2	Сенситивный		23:10	⋮
dict1	Пользовательский		Сенситивный		12:48	⋮
meta dictionary (system ...	Предустановленный		Мета			⋮

Создание словаря

Название словаря*

Вариант словаря*

Сенситивный

Мета

tantor

209

Словари анонимайзера

Три вида словарей:

Сенситивные - правила, описывающие, в каких таблицах и столбцах находятся конфиденциальные данные и как их менять (анонимизировать). Замена выполняется функциями.

Стандартно поставляемые функции: https://github.com/TantorLabs/pg_anon/blob/master/init.sql

Несенситивные - правила, описывающие таблицы и столбцы, которые не конфиденциальны.

Используются:

- 1) чтобы видеть в одном месте, что не будет изменено анонимизатором
- 2) анонимизатор может использовать этот словарь для ускорения работы: анонимизатор исключает таблицы и столбцы, перечисленные в несенситивном словаре из сканирования
- 3) чтобы скопировать часть JSON-документа в сенситивный словарь по тем таблицам и столбцам, которые нужно анонимизировать, но сканирование не вывело в них конфиденциальных данных

Мета-словари - правила, которые описывают признаки конфиденциальных данных (тип и содержание) данных. Используются на шаге сканирования, на котором создаётся сенситивный словарь. Для сканирования нужен хотя бы один мета-словарь и в Платформе есть предустановленный мета-словарь. Словари можно дублировать и редактировать полученную копию или создать новый. Создать можно мета-словарь и сенситивный. Несенситивный не создается вручную, так как это не имеет смысла: этот словарь не используется анонимизатором и нужен для того, чтобы убедиться, что не будут столбцов, которые содержат конфиденциальные данные, но не будут анонимизированы. Также из несенситивного словаря можно копировать описатели таблиц и столбцов в сенситивный словарь.

Мета-словари



Tenant 1 / Анонимайзер / Словари / **meta-dict1**



meta-dict1 пользовательский мета словарь

[Сохранение как...](#)

[Сохранить изменения](#)

```
1  {
2    "field": {
3      "rules": [
4        "str_regexp" # частичные названия полей (regexp), которые сразу будут признаны сенситивными
5      ],
6      "constants": [
7        "str" # полные названия полей, которые сразу будут признаны сенситивными
8      ],
9    },
10     "skip_rules": [ # таблицы и поля, которые игнорируются (черный список)
11       {
12         "schema": "str" # полное название схемы
13         "table": "str" # полное название таблицы
14         "fields": [
15           "str" # Список полных названий полей
16         ],
17       }
18     ]
19   }
```

tantor

210

Учебный курс "Платформа Tantor 6" Тантор Лабс © 2025 www.tantorlabs.ru

Мета-словари

Словари описываются в формате JSON и удобны для просмотра и редактирования.

Созданный вручную мета-словарь нужно редактировать. В нём нет директив, только шаблон структуры JSON-документа. Создавать собственный мета-словарь необязательно, можно использовать предустановленный meta dictionary (system dictionary), который готов к использованию. Его можно дублировать, а дубль редактировать.

Шаблон мета-словаря, созданного вручную имеет примерно 77 строк.

Шаблон сенситивного словаря

Tenant 1 / Анонимайзер / Словари / sens1

sens1 пользовательский сенситивный словарь

Сохранение как... Сохранить изменения

```
1  {
2    "dictionary": [ # Словарь сенситивных полей. Что будет анонимизироваться. Если анонимизируется база, но поле не указано в этом словаре, то оно будет дампиться "как есть"
3      {
4        "schema": "str" # полное название схемы
5        "schema_mask": "str_regex" # частичное название схемы (regexp)
6        "table": "str" # полное название таблицы
7        "table_mask": "str_regex" # частичное название таблицы (regexp)
8        "fields": { # перечень полей и функции анонимизации для этих полей. Остальные поля в таблице не будут анонимизироваться.
9          Используется либо fields, либо raw_sql
10         "field_name_str": "func_str"
11       }
12     },
13   ],
14   "dictionary_exclude": [ # Исключения из ДАМПА. Какие таблицы/схемы необходимо исключить из дампа полностью
15     {
16       "schema": "str" # полное название схемы
17       "schema_mask": "str_regex" # частичное название схемы (regexp)
18       "table": "str" # полное название таблицы
19       "table_mask": "str_regex" # частичное название таблицы (regexp)
20     }
21   ],
22 }
```

tantor 211

Шаблон сенситивного словаря

Кнопка "Сохранить изменения" становится активной, если внести изменение в текст.

Обычно, сенситивный словарь создается автоматически на основе мета-словаря в результате сканирования. Словарь, созданный вручную, может добавляться к словарю, созданному автоматически, на шаге предпросмотра и дампа (выгрузки).

Шаблон сенситивного словаря, созданного вручную:

```
{  
  "dictionary": [ # Словарь сенситивных полей. Что будет анонимизироваться. Если анонимизируется база, но поле не указано в этом словаре, то оно будет дампиться "как есть"  
    {  
      "schema": "str" # полное название схемы  
      "schema_mask": "str_regex" # частичное название схемы (regexp)  
      "table": "str" # полное название таблицы  
      "table_mask": "str_regex" # частичное название таблицы (regexp)  
      "fields": { # перечень полей и функции анонимизации для этих полей. Остальные поля в таблице не будут анонимизироваться. Используется либо fields, либо raw_sql  
        "field_name_str": "func_str"  
      }  
      "raw_sql": "str" # Сырой SQL запрос, который говорит, как нужно вытянуть таблицу.  
      Внутри можно построить разные джойны и т.п. Либо fields, либо raw_sql  
    },  
    {  
      "dictionary_exclude": [ # Исключения из ДАМПА. Какие таблицы/схемы необходимо исключить из дампа полностью  
        {  
          "schema": "str" # полное название схемы  
          "schema_mask": "str_regex" # частичное название схемы (regexp)  
          "table": "str" # полное название таблицы  
          "table_mask": "str_regex" # частичное название таблицы (regexp)  
        }  
      ]  
    }  
  ]  
}
```

Источники данных

The screenshot shows the Tantor 6 platform interface. On the left, there is a sidebar with various icons and a navigation bar with the text "Tenant 1 / Анонимайзер / Источники данных". The main area is titled "Источники данных" and contains a search bar and a table with columns: НАЗВАНИЕ, БАЗА ДАННЫХ, ПОСЛЕДНЕЕ СКАНИРОВАНИЕ, СТАТУС СКАНИРОВАНИЯ, ПОСЛЕДНИЙ ДАМП, and СТАТУС ДАМПА. A table row shows "primary" as the data source name, "postgres" as the database, "08/10/2025 23:17" as the last scan time, "Завершено" (Completed) as the scan status, "08/10/2025 13:08" as the last dump time, and "Завершено" (Completed) as the dump status. A modal window titled "Создание источника данных" is open, prompting for "Название источника данных*" (primary), "FQDN или IP*" (10.0.2.15), "Порт*" (5432), "Название БД*" (postgres), and an "Email для получения уведомлений о событиях". A note says "Важно! SMTP-сервер должен быть настроен". There is a checkbox for "Дополнительные параметры". At the bottom of the modal are "Отменить" and "Создать" buttons. To the right of the modal is a context menu with options: "Открыть", "Переименовать", "Редактировать", and "Удалить". A "Создать источник данных" button is also visible. The footer of the page includes the Tantor logo and the text "Учебный курс 'Платформа Tantor 6' Тантор Лабс © 2025 www.tantolabs.ru" and "212".

Источники данных

В результате анонимизации создаётся дамп утилитой pg_dump в формате custom. Можно создать дампы: структуры объектов, данных (строк таблиц), структуры+данных. Формат custom читается стандартной утилитой pg_restore, которая может на основе формата custom создать файл дампа в формате plain.

Для анонимизации сначала создается источник данных. Достаточно указать адрес базы данных PostgreSQL. Появится новая запись в списке Источников данных на этой странице. На нее можно кликнуть или выбрать "Открыть" в меню, которое появляется по нажатию на три точки. В этом меню есть пункты: Открыть, Переименовать, Редактировать, Удалить.

После нажатия в всплывающем окне на кнопку "Создать" на странице появится этот созданный источник данных. Можно кликнуть на строку и откроется страница с 5 вкладками, где дальше описываются шаги процесса анонимизации.

Сканирование

Tenant 1 / Анонимайзер / Источники данных / primary / Информация об источнике данных

ИНФОРМАЦИЯ ОБ ИСТОЧНИКЕ ДАННЫХ СКАНИРОВАНИЕ ПРЕДПРОСМОТР ДАМПЫ РЕСТОРЫ

FQDN или IP: 10.0.2.15
Порт: 5432
База данных: postgres
Последнее сканирование: -
Статус сканирования: -
Последний дамп: -
Статус дампа: -
Когда изменил: -
Кто изменил: -
Число процессов: -
Подключений на процесс: -
Путь к pg_dump: -
Email для получения уведомлений о событиях: -

Запустить сканирование

Шаги: 1 2 3 4

Выберите мета-словарь(и):

- Выбрать всё
- meta dictionary (system dictionary)

Назначение сканирования: Частичное
Глубина сканирования: 1
Результат сканирования — словари. Укажите названия
Название сенситивного словаря: dict1
Словарь с таким именем уже существует
Название несенситивного словаря (если нужен): -
Далее

tantor 213

Сканирование

На странице Источника данных 5 вкладок.

Сначала нужно запустить сканирование - выявление столбцов в таблицах, которые будут анонимизированы и определить способ маскировки. На вкладке СКАНИРОВАНИЕ нужно нажать на кнопку "Добавить сканирование". Появится визард, в котором указываются параметры сканирования.

На первом шаге в поле "Название сенситивного словаря" нужно указать название нового "словаря", который будет создан. Этот словарь будет содержать список таблиц, столбцов и функций маскировки. Несенситивный словарь опционален, в нём будут перечислены таблицы и столбцы, которые не будут меняться. **Результатом сканирования будут только сенситивный словарь и опционально несенситивный.**

На втором шаге выбирается мета-словарь. По умолчанию имеется системный словарь и достаточно выбрать его, чтобы перейти на третий шаг.

На третьем и четвертом опциональных шагах можно выбрать уже существующие сенситивные и несенситивные словари. Несенситивные повлияют только тем, что предпросмотр и дамп могут ускориться, так как логике анонимизатора не нужно будет обрабатывать перечисленные в несенситивном словаре столбцы таблиц. Выбор сенситивных словарей повлияет на следующие шаги (предпросмотр и дамп) тем, что принудительно будет считаться конфиденциальным те столбцы, которые в них указаны и будут использоваться правила анонимизации, которые указаны для этих столбцов.

На последнем шаге запускается задача по сканированию.

Для задачи нужно указать имя и пароль для подсоединения к базе. Можно нажать кнопку "Отменить", тогда задача будет создана, но не запущена. Её можно будет запустить позже.

Ввод данных для подключения

Пользователь для подключения к БД *

pma_user

Пароль для подключения к БД *

.....

Отменить

Подключиться

Tenant 1 / Анонимайзер / Источники данных / primary / Сканирование

Результат сканирования

ИНФОРМАЦИЯ О ИСТОЧНИКЕ ДАННЫХ СКАНИРОВАНИЕ ПРЕДПРОСМОТР ДАМПЫ РЕСТОРЫ

Поиск по имени

НАЗВАНИЕ	БАЗА ДАННЫХ	СЛОВАРИ	ГЛУБИНА	СТАТУС	НАЧАЛО	КОНЕЦ	ЗАПУСТИЛ	...
scan1	Tenant 1 / Анонимайзер / Словари / dict2s	meta dictionary (s...)	1000	Несенситивный	08/10/2025 00:28	08/10/2025 00:30	student@student...	<input type="button" value="⋮"/>
scan2	dict2s	пользовательский сенситивный словарь	1000	Сенситивный	08/10/2025 23:16	08/10/2025 23:17	student@student...	<input type="button" value="⋮"/>

Сохранение как...

```

1  {
2    "dictionary": [
3      {
4        "schema": "public",
5        "table": "pgbench_branches",
6        "fields": [
7          "bbalance": "anon_funcs.random_int_between(1, 10)"
8        ],
9      },
10     {
11       "schema": "public",
12       "table": "stock_items",
13       "fields": [
14         "amount": "anon_funcs.noise(\"amount\", 10)"
15       ]
16     }
17   }
18 }
```

Запустить Словари-результаты Переименовать Удалить


tantor

Учебный курс "Платформа Tantor 6" Тантор Лабс © 2025 www.tantolabs.ru

214

Результат сканирования

По результатам сканирования создается сенситивный словарь и, если был указан, то несенситивный. По ссылке в выпадающем меню "Словари-результаты" можно посмотреть список созданных словарей, открыть их и отредактировать их текст. Кнопка "Сохранить изменения" становится активной, если внести изменения в текст словаря.

Также на словари можно перейти через вкладку СЛОВАРИ на первой странице анонимайзера.

Пункт меню "Запустить" активен, если не вводились детали подсоединения к базе данных. Запустить сканирование можно один раз. Сканирование после получения словарей бесполезно и его можно удалить.

Tenant 1 / Анонимайзер / Словари

ИСТОЧНИКИ ДАННЫХ СЛОВАРИ

Поиск по имени

НАЗВАНИЕ	ТИП	СКАНИРОВАНИЕ	ВАРИАНТ	СОЗДАЛ	ПОСЛЕДНИЕ ИЗМЕНЕНИЯ	...
dict2u	Пользовательский	scan2	Несенситивный		08/10/2025 23:17	<input type="button" value="⋮"/>
dict2s	Пользовательский	scan2	Сенситивный		08/10/2025 23:17	<input type="button" value="Открыть"/> <input type="button" value="Переименовать"/> <input type="button" value="Дублировать"/> <input type="button" value="Удалить"/>

Создание предпросмотра

The screenshot shows the Tantor platform interface for creating a preview. The top navigation bar includes 'Tenant 1 / Анонимайзер / Источники данных / primary / Предпросмотр'. The main menu tabs are 'ИНФОРМАЦИЯ О ИСТОЧНИКЕ ДАННЫХ', 'СКАНИРОВАНИЕ', 'ПРЕДПРОСМОТР' (which is underlined in red), 'ДАМПЫ', and 'РЕСТОРЫ'. A search bar with placeholder 'Поиск по имени' and a refresh icon are on the left. A red button 'Новый предпросмотр' is on the right. Below the menu, tabs for 'НАЗВАНИЕ', 'БАЗА ДАННЫХ', 'СЛОВАРИ', and '...' are visible. A list of preview entries shows 'preview1' and 'dict4scan1'. A context menu is open for 'dict4scan1' with options: 'Открыть', 'Переименовать', 'Редактировать', and 'Удалить' (Delete). A modal dialog 'Новый предпросмотр' is open, showing a 'Название предпросмотра*' input field with 'preview2' and a 'Сенситивные словари*' section with checkboxes for 'Выбрать всё' (unchecked), 'dict2s' (checked), and 'dict1' (unchecked). Buttons 'Отменить' and 'Применить' are at the bottom of the dialog. The footer contains the Tantor logo and the text 'Учебный курс "Платформа Tantor 6" Тантор Лабс © 2025 www.tantolabs.ru'.

Создание предпросмотра

Предпросмотр (preview) - визуальное сравнение строк таблиц до анонимизации и после, чтобы проверить как будут выглядеть анонимизированные данные в сравнении с исходными.

Предпросмотр необязательный шаг, но он нетрудоёмкий и лёгок в использовании.

Для создания предпросмотра достаточно указать сенситивный словарь или словари, по правилам которых и будет выполняться анонимизация.

Предпросмотр можно запускать неоднократно. Если удалить словарь, который использует предпросмотр, то предпросмотр не запустится.

Предпросмотр

Сchemes

Объект БД

schemas 1

public

tables 7

order_items_1

order_items_2

pgbench_accounts

pgbench_branches

pgbench_histories

pgbench_tellers

stock_items

amount

descr

optcounter

Элементов на странице 25 1 - 25 из 1,000

ID	ORDER_ITEMS_1_I	ORDER_ITEMS_2_I	AMOUNT	OPTCOUNTER	DESCRIPTOR
1	7953	1	183.0000	19196	None
2	141569	2	323.0000	5009	None
1579506	13855	9506	177.0000	6928	updated at 2025-1...

ID	ORDER_ITEMS_1_I	ORDER_ITEMS_2_I	AMOUNT	OPTCOUNTER	DESCRIPTOR
1	7953	1	386.773367770521	19196	None
2	141569	2	-613.903200198617	5009	None
1579506	13855	9506	1260.12849344841	6928	updated at 2025-1...

Элементов на странице 25 1 - 25 из 1,000

tantor 216

Предпросмотр

При запуске предпросмотра открывается просмотр таблиц, которые попадут в дамп. Данные таблиц показываются в двух вариантах: исходные до анонимизации (сверху) и после (снизу).

Максимальное число строк, которые показываются по каждой таблице 1000.

Максимальное число таблиц при просмотре 5000.

Дамп

Дамп создан успешно

Информация об источнике данных Сканирование Предпросмотр Дампы Ресторы

Поиск по имени

Название | База данных | Статус | Путь | Размер

dump1 postgres Запуск dump1

Создание дампа

Название дампа* dump1

Тип Дампа* Полный

Путь к дампу* dump1

Будет перезаписан если существует

Сенситивные словари*

Выбрать всё dict4scan1 dict1

Создать дамп

```
ls /opt/tantor/eco/pg_anon/dump1
07f7b6866214f4297f1e4a67edec8f76.bin.gz
34ed14e1766e6b41507b217d2f088dfc.bin.gz
427f9ad66a7219208643423cd6b81ab9.bin.gz
43739e7ec0d34bead9530d005fd4e6f.bin.gz
b9e9dce5083f92b06831e9c35674a15e.bin.gz
d5bfc1b3b61ebbc20100f8ae23afa548.bin.gz
f585e47667dff1fd226ea1737e67c811.bin.gz
metadata.json
post_data.backup
pre_data.backup
```

Дамп

Дампы создаются в директории: `/opt/tantor/eco/pg_anon` и принадлежат `root` с правами `rwxr-xr-x`. Можно использовать относительное название директории "dump1" или со слэшем в начале "/dump1".

При указании "../" дамп будет создан внутри контейнера (оверлее). Например, при указании пути к дампу `../../dump3` будет создана [директория](#) в контейнере `pg_anon`:

```
docker exec -it pg_anon ls /pg_anon/dump3
```

Такой дамп будет доступен и его можно будет использовать и удалить на этой странице, но он не будет виден в директории хоста.

При создании дампа нужно указать хотя бы один сенситивный словарь, по правилам которого будут выбираться столбцы и применяться функции, которые изменять значения полей так, чтобы анонимизировать (маскировать) данные - сделать невозможным получение исходных данных.

Загрузка дампа

The screenshot shows the Tantor platform interface. At the top, there is a navigation bar with icons for Tenant 1, Anonymizer, Data Sources, primary, and Dumps. Below the navigation is a header with tabs: ИНФОРМАЦИЯ О ИСТОЧНИКЕ ДАННЫХ, СКАНИРОВАНИЕ, ПРЕДПРОСМОТР, ДАМПЫ (which is underlined in red), and РЕСТОРЫ. A search bar with placeholder 'Поиск по имени' and a magnifying glass icon is on the left. A green button labeled 'Восстановление «dump3» запущено' is in the center. On the right, there is a modal window titled 'Восстановление дампа' with fields for FQDN/IP (10.0.2.15), Port (5432), Database Name (restore_db), User (postgres), and Password (postgres). Below the modal are buttons for 'Отменить' (Cancel) and 'Восстановить' (Restore). To the right of the modal is a sidebar with buttons for 'Новый дамп' (New dump), 'ПУСТИЛ' (Launched), 'Запустить' (Start), 'Переименовать' (Rename), 'Восстановить' (Restore), and 'Удалить' (Delete). The main table below the header shows a single row for 'dump3' with details: база данных 'postgres', статус 'Завершено', путь './dump3', и размер '38 MB'. The footer of the screenshot includes the Tantor logo and the text 'Учебный курс "Платформа Tantor 6" Тантор Лабс © 2025 www.tantorlabs.ru'.

Загрузка дампа

Дампы можно передать тем, кому они нужны в виде файлов.

Если Платформа имеет доступ к базе, в которую нужно загрузить анонимизированные данные, то можно в всплывающем меню выбрать пункт "Восстановить" и указать параметры подсоединения к такой базе данных и дамп будет восстановлен стандартной утилитой `pg_restore` в этой базе данных.

База данных должна существовать.

Созданные восстановления будут видны в закладке РЕСТОРЫ:

The screenshot shows the Tantor platform interface. At the top, there is a navigation bar with icons for Tenant 1, Anonymizer, Data Sources, primary, and Restores. Below the navigation is a header with tabs: ИНФОРМАЦИЯ О ИСТОЧНИКЕ ДАННЫХ, СКАНИРОВАНИЕ, ПРЕДПРОСМОТР, ДАМПЫ, and РЕСТОРЫ (which is underlined in red). A search bar with placeholder 'Поиск по имени' and a magnifying glass icon is on the left. On the right, there is a modal window with a 'Перезапустить' (Restart) button. The main table below the header shows three rows of restoration history: 'dump3' (status: Завершено, started 09/10/2025 14:54, finished 09/10/2025 14:55), 'dump3' (status: Ошибка, started 09/10/2025 14:56, finished -), and 'dump3' (status: Завершено, started 09/10/2025 14:58, finished 09/10/2025 14:58). The footer of the screenshot includes the Tantor logo and the text 'Учебный курс "Платформа Tantor 6" Тантор Лабс © 2025 www.tantorlabs.ru'.

Удалить ресторы нельзя, они для истории. Перезапустить можно через всплывающее меню только те, которые завершились неудачно. При Перезапуске можно указать имя и пароль к базе данных.

Практика 11

Практика 11

1. Откройте модуль Анонимайзера
2. Выполните сканирование для поиска конфиденциальных данных
3. Посмотрите результат сканирования - "сенситивный" словарь
4. Выполните предпросмотр строк таблиц для анонимизации
5. Выгрузите дамп с анонимизированными данными

Практика 11

1. Откройте модуль Анонимайзера
2. Выполните сканирование для поиска конфиденциальных данных
3. Посмотрите результат сканирования - "сенситивный" словарь
4. Выполните предпросмотр строк таблиц для анонимизации
5. Выгрузите дамп с анонимизированными данными



Инсталляция

Установка Платформы Тантор

Способы установки Платформы Tantor

- Способы установки:
 - типичная в контейнерах docker на одном хосте
 - отказоустойчивая вне контейнеров docker
- Операционная система:
 - Astra Linux версий 1.7.5+, 1.8
 - RedOS 7.3+
 - Centos 7+
- Платформа версии 6.0 поддерживает версии PostgreSQL от 10 до 17.x версии

Экземпляров PostgreSQL	ядер CPU	RAM (GB)	занимаемое место (SSD)
10	4	8	100 GB
100	12	24	500 GB
1000*	18	32	1–2 ТВ
5000*	32	64	5–10 ТВ

Способы установки Платформы Tantor

Типичная установка программного обеспечения выполняется на один хост или виртуальную машину с операционной системой:

Astra Linux версий 1.7.5+, 1.8

RedOS 7.3+

Centos 7+

Платформа устанавливается в виде набора docker-контейнеров. В операционной системе должны быть установлены:

Docker Engine 20.10.13+

Docker Compose 1.29.2+

Типичная установка позволяет обслуживать до 1000 экземпляров PostgreSQL. Для обслуживания большего числа экземпляров Платформу нужно устанавливать в режиме отказоустойчивости. Установку в режиме отказоустойчивости и миграцию на новые версии помогает выполнить техническая поддержка Тантор Лабс.

При типичной установке миграция на новые версии Платформы Tantor выполняется стандартным инсталлятором с минимальным пространством доступа к Платформе.

Платформа версии 6.0 поддерживает версии PostgreSQL от 10 до 17.x версии. Более новые версии PostgreSQL поддерживаются более новыми версиями Платформы.

Утилита диагностики Платформы Tantor

```
root@tantor:/opt/tantor/eco6# chmod 700 ./sdc-tantor
root@tantor:/opt/tantor/eco6# ./sdc-tantor
```

SDC-Tantor (версия 4.0.0)

Будет выполнен сбор системной конфигурации и диагностической информации из данной ОС.

Созданный архив может содержать данные, считающиеся конфиденциальными, и его содержимое должно быть проверено, прежде чем передано третьим лицам.

Никаких изменений в конфигурацию системы вноситься не будет.

Нажмите ENTER для продолжения или CTRL-C для выхода **SDC-Report**

Выполнение сбора данных..

```
hostnamectl      commands/hostnamectl
df -h      commands/df_-h
apt list --installed      commands/apt_list_--install
```

...

Сбор данных завершён.

Создание архива..

Архив создан и сохранён в: sdc-report_tantor_202509

Пожалуйста, отправьте этот файл в службы поддержки

```
root@tantor:/opt/tantor/eco6#
```

Commands executed:

- [hostnamectl](#)
- [df-h](#)
- [apt list -installed](#)
- [find /opt/tantor/ -exec ls -dls {} \;](#)
- [tree /opt/tantor/](#)
- [ls -alhR /etc/docker](#)
- [journalctl --no-pager --unit docker](#)
- [docker info](#)
- [docker images](#)
- [docker events --since 24h --until 1s](#)
- [docker ps](#)
- [docker ns -a | grep "nexius * tantorlabs.ru"](#)

Утилита диагностики Платформы Tantor

В директории дистрибутива есть утилита диагностики **sdc-tantor**. Она может запускаться в любое время: до инсталляции или в процессе эксплуатации Платформы. Обычно утилиту используют в случае неудачного завершения установки, обновления Платформы, при обращении в техническую поддержку Tantor.

Утилита создаёт архив с результатами выполнения команд. В архиве html-файл, в котором ссылки на текстовые файлы архива, чтобы удобнее просматривать их с помощью браузера.

https://docs.tantorlabs.ru/tp/6.1/admin/admin_diagnostic_utility.html

Запуск инсталлятора Платформы Tantor

- в документации приводится ссылка на текущую версию дистрибутива Платформы
 - <https://docs.tantorlabs.ru/tp/6.1/admin/wizard.html>
- размер архива примерно 3Гб
- распаковка дистрибутива:
 - tar xzf tantor-eco-6.*.tar.gz
- Запуск инсталлятора:
 - ./installer
- инсталлятор распознаёт установленную Платформу и предложит обновить на новую версию
 - если Платформа не установлена, то предложит установить

Запуск инсталлятора Платформы Tantor

Как скачать дистрибутив описано в документации:

<https://docs.tantorlabs.ru/tp/6.1/admin/wizard.html>

В документации приводится ссылка на **текущую версию** дистрибутива Платформы.

Дистрибутив можно скачать на хост, на котором будет устанавливаться Платформа следующими командами.

переключиться в root, так как инсталлятор запускается из под пользователя root в операционной системе linux:

sudo bash ИЛИ **su -**

создать директорию для дистрибутива:

mkdir /opt/tantor/eco6

перейти в созданную директорию:

cd /opt/tantor/eco6

скачать дистрибутив, размер архива примерно 3Гб:

wget https://public.tantorlabs.ru/tantor-eco-6.0.1.tar.gz -O tantor-eco-6.0.1.tar.gz

распаковать дистрибутив:

tar xzf tantor-eco-6.0.1.tar.gz

Если архив с дистрибутивом не нужен, то его можно удалить командой:

rm -f tantor-eco-6.0.1.tar.gz

Чтобы можно было работать с контейнерами docker не только из-под root опционально можно добавить пользователей в группу docker:

usermod -aG docker astra

usermod -aG docker postgres

Можно поменять разрешения на файл инсталлятора и диагностики, чтобы его мог запускать только root:

chmod 700 ./sdc-tantor ./installer

Запуск инсталлятора:

./installer

Инсталлятор можно запустить перейдя в директорию с файлом инсталлятора и если не переключиться в root, то можно набрать команду: **sudo ./installer**

Инсталлятор распознаёт установленную Платформу и предложит обновить на новую версию. Если Платформа не установлена, то предложит установить.

Обновление Платформы Tantor

```
root@tantor:/opt/tantor/eco6#
./installer start --disable-backup-platform-directory
Writing diagnostic information into the file
[/opt/tantor/eco6/tantor_eco_installation.log]
#####
#           Welcome to Tantor Platform installer.           #
#####

- Checking Tantor Platform dependencies:
- docker installed... [Yes]
- docker installed required version... [Yes]
- docker started... [Yes]
- Do you want to upgrade the Tantor Platform from 6.0.0 to 6.1.0 version?
(the platform will be unavailable during the update) (Y/N): y
- Tantor Platform upgrading
- Tantor Platform will be stopped... [Yes]
- Create a backup Tantor Platform... [No]
- Upgrading...
- Start Tantor Platform... [Yes]
```

Учебный курс "Платформа Tantor 6" Тантор Лабс © 2025 www.tantorlabs.ru



224

Обновление Платформы Tantor

Инсталлятор проверяет свободное место. Если его недостаточно, то выдаст предупреждение:

- Checking available free space... [No]
- There is not enough space to perform the update. The space must be twice the size of the space currently occupied by the platform.
- Continue? (Y/N) :

Если в процессе обновления Платформа будет недоступна, инсталлятор об этом напишет. Обновление минорной версии не занимает много времени (в пределах 5 минут) и обновление заменяет контейнеры с их остановкой. При обновлении мажорной версии обычно создаются новые контейнеры, а контейнеры предыдущей версии останавливаются после создания новых и простой минимизируется. При ошибке запуска контейнеров выдается ошибка:

- Start Tantor Platform... [No]
- Execution error. Could not start Tantor Platform. Additional information can be found in the tantor_eco_installation.log file. Please contact the support team for assistance to install Tantor Platform.

При возникновении ошибок можно перезапустить службу docker:

sudo systemctl restart docker

посмотреть статус контейнеров, обратив внимание на неработающие контейнера. docker не удобен для администрирования, команды посмотреть неработающие контейнеры нет. Можно посмотреть по статусам:

```
docker ps -a --filter "status=exited"
docker ps -a --filter "status=created"
docker ps -a --filter "status=stopped"
```

или полный список: **docker ps -a** и визуально оценить какие контейнеры не работают.

Запустить не работающие, но относящиеся к текущей версии Платформы, командой (пример):

docker start pg_monitor_dispatcher

Стоит проверить статус этого контейнера после миграции, так как он может не запуститься.

Очистка места после обновления

```
...
- Check connection to Tantor Platform... [Yes]
- Upgrading complete!
- Old version: 6.0.0
- New version: 6.1.0
docker ps -a --filter "status=exited"
IMAGE           COMMAND   STATUS    NAMES
tns_migrations:6.1.0  "./flyway.sh" Exited (0)  tns_migrations
maindb:6.1.0        "./flyway.sh" Exited (0)  maindb_migrations
operdb:6.1.0        "./flyway.sh" Exited (0)  operdb_migrations
docker image prune -a -f
Deleted Images:
...
deleted:
sha256:48e3700523371eff5125b34c4cf9e10f1bc4694f4e5ea82bb8099f05a1e6651
Total reclaimed space: 968.7MB
```

Очистка места после обновления

Если на хосте docker используется только [Платформой](#), то пересоздать контейнеры можно сначала удалив их командой:

```
cd /opt/tantor/eco
docker-compose --env-file /opt/tantor/eco/platform.env -f
/opt/tantor/eco/docker-compose.yml down
```

И создать их заново (понадобится доступ в интернет):

```
docker-compose --env-file /opt/tantor/eco/platform.env -f
/opt/tantor/eco/docker-compose.yml up -d
```

Пересоздание контейнеров может использоваться, если нет уверенности, что миграция прошла удачно.

После обновления или пересоздания остаются образы контейнеров прежней версии, которые занимают место. Для версии 6.0.0 это 5,5ГБ. Посмотреть список контейнеров и их размеры можно командой:

```
docker images | grep 6.0.0
```

После проверки, что все контейнеры текущей версии работают можно проверив, что нет контейнеров текущей версии со статусами:

```
docker ps -a --filter "status=exited"
docker ps -a --filter "status=created"
docker ps -a --filter "status=dead"
```

Можно удалить образы и освободить место командами:

```
docker rmi $(docker images | grep 6.0.0 | tr -s ' ' | cut -d ' ' -f 3)
docker container prune -f
docker image prune -a -f
docker system prune -a -f
```

Будут удалены все **не работающие** на момент выдачи команды (по любой причине) контейнеры и их образы. Вместо **6.0.0** нужно подставить старую версию, с которой обновлялись на новую версию.



12-2

Обзор docker и docker-compose

docker и docker-compose

- docker - изолированная виртуальная среда, в которой запущены процессы приложений
- используются образы с установленными и сконфигурированными приложениями
- данные могут храниться в директориях хоста (docker volumes, тома), которые не исчезнут после пересоздания контейнера
- если приложение состоит из нескольких контейнеров, то обычно используют конфигурационный файл docker-compose.yml вместо скрипта создания и конфигурирования контейнеров, сетей, томов

docker и docker-compose

Контейнер docker - это изолированная виртуальная среда (аналог chroot), в которой запущены процессы нескольких приложений, в идеале одного приложения. Аппаратная виртуализация не используется, процессы в контейнерах это процессы операционной системы хоста. В отличие от chroot, изолируется:

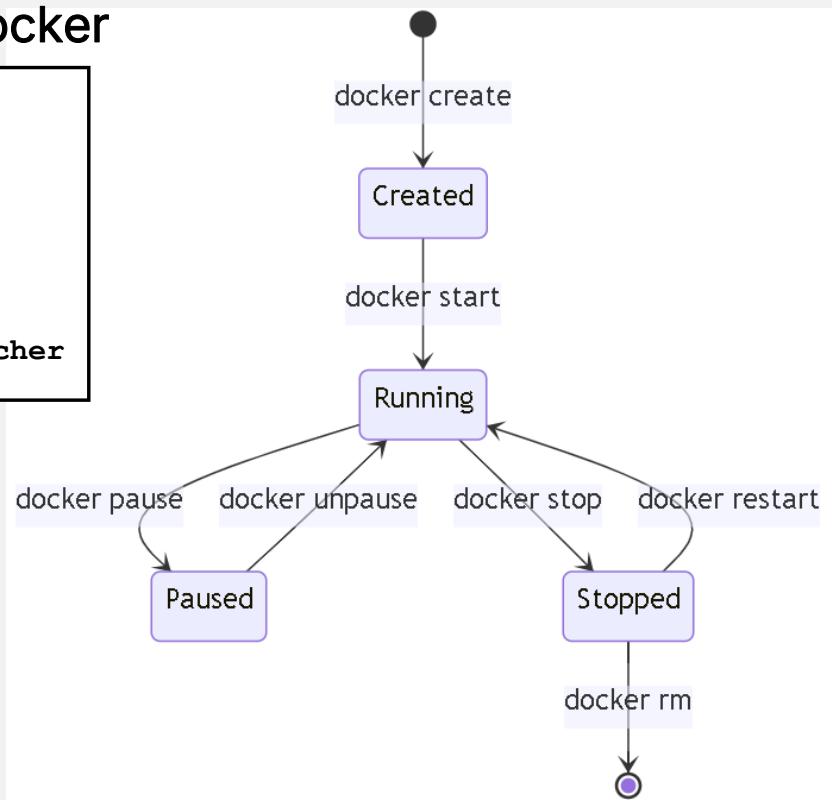
- 1) файловая система, которая делится на слои и экономится место, если нужно запускать много контейнеров, в которых дублируются файлы
- 2) процессы разных контейнеров не могут обмениваться данными через общую память
- 3) сеть.

Докер позволяет упаковать уже установленную, работоспособную программу в образ (image) и распространять так, что она будет выполняться одинаковым образом. Пользователи программы могут запускать несколько экземпляров (контейнеров) программы на основе одного образа (image), вносить в них изменения. Контейнер - набор параметров для запуска на основе образа. Если изменения нежелательны, то можно пересоздать контейнер. Перезапуск контейнеров очень быстрый, в отличие от запуска операционной системы и виртуальных машин. Это позволяет быстро менять версии приложения. С точки зрения использования, контейнер - это готовая к работе виртуальная машина с приложением, нужным ему окружением (переменные окружения, настройки и пакеты linux). Если приложение обрабатывает данные, то данные могут храниться в директориях хоста (docker volumes, тома), которые не исчезнут после пересоздания контейнера. Временные и постоянные данные могут храниться в слоях контейнера, в которых разрешена запись, но данные исчезнут после удаления контейнера с таким слоем.

Если приложение состоит из нескольких слабосвязанных частей (приложений), то для него оптимально иметь несколько образов. Например, если части обновляются с разной частотой, созданы разными разработчиками, им нужно разное окружение и разные библиотеки, нужно несколько экземпляров приложений. Такие приложения обычно распространяются с конфигурационным файлом в формате yaml. Название файла по умолчанию docker-compose.yml В файле описываются образы, параметры контейнеров, тома, сетевые настройки. Без docker-compose для запуска контейнеров приложения нужно было бы написать скрипт с набором команд (создания и настройки сетей и томов) утилиты docker для каждого контейнера. Суть docker-compose - использовать файл в формате yaml, вместо sh-скрипта. Есть публичное хранилище образов - hub.docker.com, что упрощает распространение приложений.

Статусы контейнеров docker

```
# List all containers
docker ps -a
# Filter containers by status
docker ps -f status=running
docker ps -f status=exited
docker ps -f status=created
# Start the container
docker start pg_monitor_dispatcher
```



Статусы контейнеров docker

Контейнер может находиться в состояниях:

Created (Создан) - контейнер инициализирован, но не запущен, docker create

Running (Запущен) - контейнер работает, docker run, docker start

Paused (Приостановлен) - работа контейнера приостановлена, docker pause, docker unpause

Stopped (Остановлен) - контейнер остановлен, но не удален, docker stop, docker kill

Exited (Завершен) - контейнер завершил выполнение, docker ps -a

Командой docker events можно просматривать события контейнеров. Командой docker stats просматривать использование ресурсов контейнерами:

docker stats

NAME	CPU %	MEM USAGE	MEM %	NET I/O	BLOCK I/O	PIDS
keeper	0.00%	19.9MiB	0.25%	1.6MB / 4.1MB	15.7MB / 0B	9
reverse-proxy	0.00%	10.1MiB	0.13%	2.0MB / 2.3MB	6.73MB / 20kB	6
backend	0.47%	34.1MiB	0.43%	0.9MB / 714kB	20MB / 0B	10
pg_explain	28.41%	460.5MiB	5.80%	0.1MB / 124kB	16MB / 4.1kB	64
pg_monitor_collector	10.04%	702.4MiB	8.85%	31.6MB / 14.3MB	7.33MB / 8.1kB	118
pg_monitor_dispatcher	0.00%	45.6MiB	0.58%	1.2MB / 2.4MB	14.6MB / 8.1kB	23
pg_bouncer	0.13%	10.2MiB	0.13%	5.0MB / 5.1MB	9.81MB / 4.1kB	2
nats	0.27%	16.0MiB	0.20%	1.2MB / 1.7MB	10.5MB / 1.3MB	12
pg_anon	0.24%	55.9MiB	0.70%	6.4kB / 0B	606kB / 328kB	6
operdb	2.19%	464.6MiB	5.85%	4.1MB / 745kB	94.4MB / 26.9MB	52
tns_db	0.21%	549.1MiB	6.92%	15MB / 8.2MB	26MB / 89.3MB	275
frontend	0.00%	7.3MiB	0.09%	5.7kB / 0B	3.64MB / 20.5kB	5
maindb	0.16%	112.2MiB	1.41%	211kB / 203kB	13.1MB / 1.4MB	44
pg_configurator	0.01%	28.7MiB	0.36%	6.4kB / 0B	2.29MB / 0B	2
swagger_ui	0.00%	4.4MiB	0.06%	4.6kB / 0B	8.19kB / 1.1MB	5

Выйти из команды можно набрав **ctrl+c**.

Сетевая изоляция контейнеров docker

- типы сетей докера: brige, host, overlay, none
- по умолчанию подсети brige изолированы друг от друга
- сети имеют внутреннюю адресацию **172.x.*.*/16** и извне к таким ip-адресам контейнеров доступ возможен через проброс портов с ip-адресов хоста

```
services:  
  nginx:  
    container_name: reverse-proxy  
    ports:  
      - 80:80  
      - 443:443
```

- для отключения ограничения доступа приложений хоста к портам IP-адресов контейнеров нужно установить в файле службы `/etc/systemd/system/multi-user.target.wants/docker.service`:

```
[Service]  
Environment="DOCKER_INSECURE_NO_IPTABLES_RAW=1"
```

Сетевая изоляция контейнеров docker

Контейнеры могут использовать типы сетей: host (делят ip и порты с хостом, задержка меньше, но вероятны конфликты по ip-портам), overlay (docker-swarm для контейнеров на разных хостах), none. Большинство контейнеров использует тип bridge, для которого:

- 1) docker добавляет правила в **фаервол linux**, которые изолируют сети докера друг от друга
- 2) сети имеют внутреннюю адресацию **172.x.*.*/16** и извне к таким ip-адресам контейнеров доступ возможен через проброс портов (правило PREROUTING, Destination NAT)
- 3) доступ из контейнеров в сеть хоста возможен через трансляцию адресов (правило POSTROUTING, Source NAT). Для проброса указываются **порты на ip-адресах хоста**, которые перенаправляются на **порты ip-адресов контейнера**, которые могут меняться при перезапуске контейнера:

```
services:  
  nginx:  
    container_name: reverse-proxy  
    ports:  
      - 80:80  
      - 443:443
```

Если нужно, чтобы хост имел доступ к портам IP-адресов контейнеров сети докера не были изолированы, можно установить в файле службы докера `/etc/systemd/system/multi-user.target.wants/docker.service`

переменную окружения в разделе Service:

```
[Service]  
Environment="DOCKER_INSECURE_NO_IPTABLES_RAW=1"
```

Для применения параметра нужно перечитать параметры служб и перезапустить контейнеры:

```
systemctl daemon-reload
```

```
systemctl restart docker
```

Начиная с 28 версии докера можно использовать параметр:

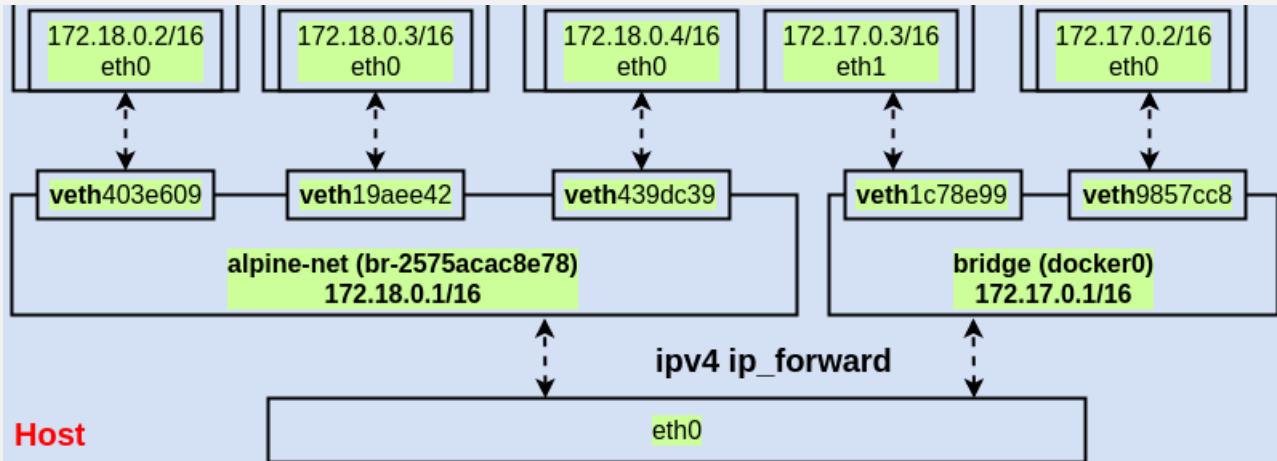
```
cat << EOF > /etc/docker/daemon.json  
  { "allow-direct-routing": true }  
EOF
```

Для применения параметра достаточно перечитать конфигурацию:

```
systemctl reload dockerd
```

Сети docker типа bridge

- внутри контейнера название интерфейса ethX, он соединён с vethX, который через мост brx соединён с veth других контейнеров в сети этого моста
- адрес маршрутизатора в сетях докера типа bridge: 172.x.0.1
- vethX (virtual ethernet) - виртуальная сетевая карта
- bridge - виртуальный switch



Сети docker типа bridge

Название сети, которую используют контейнеры Платформы можно посмотреть командой:

`docker inspect reverse-proxy | grep NetworkMode`

`"NetworkMode": "eco_default",`

Файл `/opt/tantor/eco/docker-compose.yml` не определяет параметры сети и docker создаёт сети самостоятельно, назначает ip-адреса контейнерам по умолчанию из диапазона `172.x.*.*/16`, где X при создании или пересоздании контейнеров каждый раз меняется монотонно возрастающей, чтобы была уникальность и не было конфликтов.

Список сетей, созданных докером можно посмотреть командой:

`docker network ls`

NETWORK ID	NAME	DRIVER	SCOPE
d8cb75c1a797	eco_default	bridge	local

В конце файлов `docker-compose.yml` могут указываться настройки сетей. Например:

`networks:`

```
default:  
  ipam:  
    driver: default  
    config:  
      - subnet: 172.28.0.0/16
```

указывает, что нужно создать сеть и назначить ей диапазон 172.28.0.0/16. При создании новых контейнеров для них создаются подсети и контейнера разных подсетей не могут соединяться друг с другом. Можно добавить контейнер в существующую подсеть командой `docker network connect eco_default имя_контейнера`, но это усложнит сетевую топологию. Можно при создании контейнеров указать, чтобы сеть не создавалась, а контейнерам назначались IP адреса существующих сетей. Контейнера смогут взаимодействовать с контейнерами этой существующей сети. Для этого в секции "networks:" в конце файла `docker-compose.yml` можно указать имя существующей сети и "external: true":

`networks:`

```
имя_сети:  
  name: eco_default  
  external: true
```

Содержимое файла docker-compose.yml

The following keys should be treated as mappings:

- annotations, build.args, build.labels,
- build.extra_hosts, deploy.labels,
- deploy.update_config, deploy.rollback_config,
- deploy.restart_policy, deploy.resources.limits,
- environment, healthcheck, labels, logging.options,
- sysctls, storage_opt, extra_hosts, ulimits.

One exception that applies to `healthcheck` is that the main mapping cannot specify `disable: true` unless the referenced mapping also specifies `disable: true`. Compose returns an error in this case. For example, the following input:

version: '3.8'	services:	app:	image: nginx	healthcheck:	test: ["CMD-SHELL", "curl -f http://localhost exit 1"]	interval: 10s	timeout: 5s	retries: 3	enable: false
----------------	-----------	------	--------------	--------------	---	---------------	-------------	------------	---------------

Table of contents

- Examples
- Simple example
- Advanced example
- Attributes
- annotations
- attach
- build
- blkio_config
- cpu_count
- cpu_percent
- cpu_shares
- cpu_period
- cpu_quota
- cpu_rt_runtime

Содержимое файла docker-compose.yml

Полезно уметь читать содержимое этого файла, чтобы понимать какие порты, сети, тома используются. Описание файла есть в документации

<https://docs.docker.com/reference/compose-file/services/>

В правой части страницы есть ссылки на ключи файла. YAML (ещё один язык разметки, форматирования данных) хранит данные в понятном человеку формате и использует более минималистичный синтаксис, чем JSON. Поддерживаются комментарии (значок `#`); кавычки и апострофы в строках не обязательны. Расширения файлов `yaml` или `yml`. Для отступов используются пробелы, табуляции не стоит использовать. Регистр букв важен.

Данные хранятся в виде пар:

`key: value.`

Последовательности (списки, массивы) хранят набор значений в одном ключе. Обозначаются отступами (одним или несколькими пробелами), тире, пробелом:

`key:`

`- value1`
`- value2`

или квадратными скобками:

`key: [value1, value2]`

Можно хранить иерархии:

`superkey:`

`key:`
`subkey1: 123`
`subkey2: value`

или одной строкой:

`superkey: { key: { subkey: 123 }, subkey2: value }`

Якоря (ссылки `&`) позволяют использовать фрагмент (*) несколько раз в том же файле:

`key1: &anchor1`

`surname: Doe`
`name: John1`

`key2:`

`<<: *anchor1`
`name: John`

Значок `<<` позволяет использовать фрагмент с возможностью переопределения значений.

Файл docker-compose.yml

```
name: application
services:
  node_exporter:
    image: nexus.tantorlabs.ru/repository/docker-private/pg_bouncer:latest
    volumes:
      - '/proc:/host/proc:ro'
      - '/:/rootfs:rw'
    container_name: exporter
    hostname: exporter
    ports:
      - '9100:9100'
      - "127.0.0.1:5000-5010:5000-5010"
    sysctls:
      net.ipv4.ip_unprivileged_port_start: 0
    restart: unless-stopped
    init: true
```

Файл docker-compose.yml

name: myapp # необязательный ключ с названием проекта
services: # корневой параметр, сервис = контейнер
 service_name: # название контейнера
 image: [registry/] [проект/] образ [:тег | @хэш] registry это URL репозитория, если он не указан, то используется публичный репозиторий. Для частных репозиториев используется софт, который сам распространяется как образ докера: Docker Registry (`registry:latest`) или Nexus (`sonatype/nexus3:latest`). Адрес репозитория Платформы: `nexus.tantorlabs.ru` Тег(метка) - это версия, часто используют слово `latest`
 container_name: имя # название контейнера, иначе оно будет сгенерировано автоматически. Также используют, если не хотят, чтобы запускалось больше одного контейнера
 hostname: имя # имя хоста в контейнере, иначе оно будет сгенерировано автоматически и иметь вид `1234c1234a12`
 ports: # перечисление смапированных портов хоста на порты контейнера
 - 1234:1234 # по умолчанию, мапирование на все интерфейсы
 environment: # имена и значения переменных окружения, которые будут установлены внутри контейнера. Может еще использоваться `env_file` - путь к файлу с переменными
 - ПЕРЕМЕННАЯ=ЗНАЧЕНИЕ
 sysctls: # параметры linux, которые задаются sysctl
 net.ipv4.tcp_syncookies: 0 # формат задания параметров
 restart: # что делать докеру, если контейнер сбоянёт. По умолчанию `no` - не рестартовать. `always` - рестартовать. `on-failure[:число_попыток]` - рестартовать, если код выхода скрипта отличен от нуля. `unless-stopped` - рестартовать, но если если контейнер был остановлен, то не запускать
 network_mode: "host" # "none" - изолировать. "host" - доступ к сетевым интерфейсам хоста. "service:[имя_контейнера]" - имя контейнера, к которому дать доступ. "container:[id_контейнера]" - id контейнера, к которому дать доступ.
Если ключ `network_mode` установлен, то нельзя установить ключ `network`, это будет ошибкой.
 init: true # включает службу init с номером PID=1.
<https://docs.docker.com/reference/compose-file/services/>

Файл docker-compose.yml: networks

```
networks:
  default:
    ipv4_address: 172.21.1.1
    aliases:
      - host1
  extra_hosts:
    - "education.tantorlabs.ru:10.0.2.15"
  command:
    - '--path.procfs=/host/proc'
  shm_size: 1gb # размер /dev/shm
  environment:
    TZ: Europe/Moscow
    PATRONI_API_CONNECT_PORT: 8008
    - SHOW=true
  entrypoint: [/bin/sh, /entrypoint.sh]
  depends_on:
    maindb:
      condition: service_healthy
      required: true
  stop_signal: SIGQUIT
```

Файл docker-compose.yml: networks

```
networks: # список названий сетей, к которым подключается контейнер.
  default: # название сети
    ipv4_address: 172.21.1.1
  aliases: # список дополнительных имён хоста, кроме имени службы, которые другие контейнеры могут использовать для обращения к контейнеру
    - host1
networks:# список сетей на уровне файла docker-compose.
  default:
    name: eco_default
    external: true
  extra_hosts: # список имён хостов для добавления в /etc/hosts контейнера
    - "education.tantorlabs.ru:10.0.2.15"
  command: # перекрывает команду по умолчанию, в том числе CMD из образа
    - '--path.procfs=/host/proc'
    - /bin/sh -c 'echo "hello $$HOSTNAME"'
  environment: # переменные окружения. Можно указать в двух форматах (map и array):
    PATRONI_API_CONNECT_PORT: 8008
    - SHOW=true
  entrypoint: [/bin/sh, /entrypoint.sh] # перекрывает команду CMD из образа
  depends_on: # можно задать порядок запуска и остановки контейнеров
    maindb:
      restart: true # (начиная с версии 2.17.0) при ручном рестарте контейнера от которого зависит перезапускает зависимый контейнер. При автоматическом рестарте после die не работает
      condition: service_healthy # контейнер работоспособен. Ещё могут быть значения: service_started - запустился, service_completed_successfully - успешно выполнился
      required: true # (начиная с версии 2.20.0) по умолчанию true. Если установлен в false, то зависимости должны не должны работать (быть остановлены или не доступны)
  stop_signal: SIGQUIT # как останавливать службу. По умолчанию, SIGQUIT (15)
```

Файл docker-compose.yml

```
healthcheck:
  interval: 10s
  retries: 5
  start_period: 10s
  test:
    - CMD-SHELL
    - pg_isready -U postgres -d postgres -h 127.0.0.1
    timeout: 5s
  user: root
  working_dir: /root
  stop_grace_period: 1m30s
  group_add:
    - mail
```

Файл docker-compose.yml

healthcheck: # проверка (начиная с 2.20.2), чтобы сервис считался работоспособным.

```
  interval: 10s
  retries: 5
  start_period: 10s
  test:
    - CMD-SHELL # выполнит /bin/sh, команда передаётся одной строкой
    - pg_isready -U postgres -d postgres -h 127.0.0.1
    timeout: 5s
```

Можно указать команду так:

```
  test: pg_isready -U postgres -d postgres -h 127.0.0.1
  test: ["CMD", "pg_isready", "-U", "postgres", "-d", "postgres", "-h",
"127.0.0.1"]
  test: NONE # отключает тест, который указан в образе, но лучше отключать так:
healthcheck:
  disable: true
  user: root # пользователь, под которым запускать процесс в контейнере. Перекрывает
параметр USER в образе. По умолчанию root.
  working_dir: /root # перекрывает директорию, установленную параметром WORKDIR в
образе
  stop_grace_period: 1m30s # через сколько послать SIGKILL, если контейнер не остановился
сигналом, указанным в stop_signal, по умолчанию 10s
  group_add: # членом каких групп сделать пользователь внутри контейнера
  - mail
```

Файл docker-compose.yml: volumes

```
version: '3.8'
volumes:
  - ./dir:/u01/dir:rw
  - type: volume
    source: db-data
    target: /data
    volume:
      nocopy: true
      subpath: sub
  - type: bind
    source: /var/run/postgres/postgres.sock
    target: /var/run/postgres/postgres.sock
  volumes_from:
    - service_name:ro
    - container:container_name:rw
volumes:
  db-data:
    external: true
```

Файл docker-compose.yml: volumes

```
volumes:
  - ./dir:/u01/dir:rw # Короткий синтаксис. Монтирует ./dir (файл или директорию на хосте) на /u01/dir (соответственно файл или директорию) в режиме чтения-записи (режим по умолчанию). Есть режим ro. Можно использовать относительные пути, относительно директории, в которой находится файл docker-compose.yml. Относительный путь рекомендуется начинать с точки: "./" или двоеточия "../".
  - type: volume # может быть volume, bind, tmpfs, image, pipe, cluster
    source: db-data # путь на хосте к файлу или директории или название в ключе в корневой секции volumes (используется для монтировки тома, параметры которого уже определены, в нескольких контейнерах)
    target: /data # путь в контейнере
    volume:
      nocopy: true # аналог ro (read only)
      subpath: sub # путь внутри образа для монтирования, вместо корня образа
  - type: bind # тип монтировки, не медленнее volume.
    source: /var/run/postgres/postgres.sock
    target: /var/run/postgres/postgres.sock
    create_host_path: true
  volumes_from: # монтирует все тома, указанные в другом контейнере
    - service_name:ro
    - container:container_name:rw
volumes:
  db-data: # название тома
  external: true
  name: actual-name-of-volume # если у смонтированного тома другое название
  db-data1:
    driver_opts:
      type: "nfs"
      o: "addr=10.0.2.15,nolock,soft,rw"
      device:(":/docker/example")
```

Файл /etc/hosts в контейнере docker

- Для добавления имени и IP в файл /etc/hosts контейнера нужно добавить в файл docker-compose.yml параметр:

контейнер:

```
image: ghcr.io/tantorlabs/patroni-docker:postgres-18
volumes:
- ./patroni.yml:/patroni.yml:ro
- ./entrypoint.sh:/entrypoint.sh:rw
- ./patroni-data0:/var/lib/postgresql/patroni/main
networks:
сеть:
  ipv4_address: 172.21.1.1
extra_hosts:
- "education.tantorlabs.ru:10.0.2.15"
- "host1.dba1.ru host2.dba.ru:10.0.2.15" #несколько имён
```

- ИЛИ для одиночного контейнера:

```
docker run --add-host education.tantorlabs.ru:10.0.2.15 -rm -it image_name
```

Файл /etc/hosts в контейнере docker

В docker отсутствует встроенный сервер разрешения имен (DNS), а существующие программы "DNS-сервера" сложны для установки и конфигурирования. Проблема в том, что изменения в файле /etc/hosts контейнера, внесенные в файл командами типа:

```
docker exec -it -u root контейнер echo "172.0.0.1 www.dba1.ru" >> /etc/hosts
```

не сохраняются после перезапуска контейнера, а записи файла хоста не используются контейнерами.

Для добавления имени и IP в файл /etc/hosts контейнера нужно добавить в файл docker-compose.yml параметр:

```
patroni0: &patroni-base
image: ghcr.io/tantorlabs/patroni-docker:postgres-18
entrypoint: [/bin/sh, /entrypoint.sh]
restart: unless-stopped
volumes:
- ./patroni.yml:/patroni.yml:ro
- ./entrypoint.sh:/entrypoint.sh:rw
- ./patroni-data0:/var/lib/postgresql/patroni/main
environment:
PATRONI_API_CONNECT_PORT: 8008
networks:
patroni:
  ipv4_address: 172.21.1.1
extra_hosts:
- "education.tantorlabs.ru:10.0.2.15"
- "host1.dba1.ru host2.dba.ru:10.0.2.15" #несколько имён на один ip адрес
```

При запуске одиночного контейнера:

```
docker run --add-host education.tantorlabs.ru:10.0.2.15 -rm -it image_name
```

<https://docs.docker.com/reference/cli/docker/container/run/#add-host>

Параметры linux, которые можно проверить, если трафик между сетями не проходит:

sysctl -w net.ipv4.ip_forward=1 маршрутизировать трафик между интерфейсами

sysctl -w net.ipv4.conf.all.route_localnet=1 пропускать трафик в локальные сети

Установка и обновление docker и docker-compose

- Платформе Tantor достаточно версии docker не ниже **20.10.13**. Проверить версию можно командами:

```
docker -v
Docker version 25.0.5.astra3, build
docker-compose -v
Docker Compose version 2.20.2.astra2
```

- Установка или обновление docker:

```
apt-get install docker-ce docker-ce-cli containerd.io docker-buildx-plugin
```

- Номер последней версии docker-compose можно посмотреть на странице проекта <https://github.com/docker/compose/releases/latest>
- Установка или обновление docker-compose:

```
curl -SL
https://github.com/docker/compose/releases/download/v2.39.4/docker-compose-
linux-x86_64 -o /usr/local/bin/docker-compose
chmod +x /usr/local/bin/docker-compose
ln -s /usr/local/bin/docker-compose /usr/bin/docker-compose
```

Установка и обновление docker и docker-compose

Платформе Tantor достаточно версии docker не ниже **20.10.13**.

Контейнерам с Prometheus и Grafana могут потребоваться новые (актуальные) версии.

Если ключи репозитариев дистрибутивов не добавлены, то добавить их можно командами:

```
apt-get update
apt-get install ca-certificates curl gnupg
install -m 0755 -d /etc/apt/keyrings
curl -fsSL https://download.docker.com/linux/ubuntu/gpg | gpg --dearmor -o
/etc/apt/keyrings/docker.gpg
chmod a+r /etc/apt/keyrings/docker.gpg
echo "deb [arch=amd64 signed-by=/etc/apt/keyrings/docker.gpg]
https://download.docker.com/linux/debian buster stable" >>
/etc/apt/sources.list.d/docker.list
```

Обновление кэша дистрибутивов и установка компонент docker:

```
apt-get update
apt-get install docker-ce docker-ce-cli containerd.io docker-buildx-plugin
```

Номер последней версии docker-compose можно посмотреть на странице:

<https://github.com/docker/compose/releases/latest>

docker-compose должен быть установлен как исполняемый файл (установки пакета **docker-compose-plugin** не достаточно):

```
curl -SL https://github.com/docker/compose/releases/download/v2.39.4/docker-
compose-linux-x86_64 -o /usr/local/bin/docker-compose
chmod +x /usr/local/bin/docker-compose
ln -s /usr/local/bin/docker-compose /usr/bin/docker-compose
```

Проверить версию можно командами:

```
docker -v
```

Docker version 26.1.4

```
docker-compose -v
```

Docker Compose version v2.39.4

Для перезапуска контейнеров можно перейти в директорию с docker-compose.yml и выполнить команду: **docker-compose restart**

<https://docs.docker.com/reference/compose-file/services/>



12-3

Установка Платформы, Prometheus, Grafana

Установка Prometheus и Grafana

- Создать директории

```
su -  
mkdir -p prometheus/configuration  
mkdir -p prometheus/data  
chown nobody:nogroup prometheus/data
```

- Создать файлы yaml для docker-compose и службы Prometheus node-exporter
- Запустить создание и запуск в фоновом режиме контейнеров:

```
docker-compose -f /root/docker-compose.yml up -d
```

- Проверить, что контейнеры запущены:

```
docker container ps -a | egrep prom\|gra  
grafana/grafana      Up 0.0.0.0:3000->3000/tcp  
prom/prometheus:latest Up 0.0.0.0:9090->9090/tcp  
prom/node-exporter    Up 0.0.0.0:9100->9100/tcp
```

Установка Prometheus и Grafana

Платформа Tantor может поставлять метрики для Prometheus и Grafana. Пример их установки в контейнеры docker приводится в документации:

https://docs.tantorlabs.ru/tp/6.1/admin/monitoring_with_prometheus_and_grafana.html

1) Под пользователем root создаётся файл docker-compose.yml

Содержимое файлов yaml чувствительно к пробелам, как и программы на языке python. Если файл yaml не принимается и выдаются ошибки, нужно проверить и расставить пробелы.

2) Перейти в директорию с файлом docker-compose.yml (в виртуальной машине к курсу это /root) и создать директории:

```
su -  
mkdir -p prometheus/configuration/  
mkdir -p prometheus/data/
```

3) Поменять владельца созданной директории на nobody:nogroup чтобы Prometheus мог читать и писать в директорию:

```
chown nobody:nogroup prometheus/data
```

4) Создать конфигурационный файл для сбора метрик с сервиса prometheus-node-exporter (пробелы важны):

```
cat << EOF > prometheus/configuration/prometheus.yml  
scrape_configs:  
- job_name: 'node'  
  scrape_interval: 5s  
  static_configs:  
  - targets: ['node-exporter:9100']  
EOF
```

5) Выполнить создание и запуск контейнеров в фоновом режиме (-d):

```
docker-compose -f /root/docker-compose.yml up -d
```

6) Проверить, что статус контейнеров Up:

```
docker container ps -a | egrep prom\|gra  
grafana/grafana      Up 0.0.0.0:3000->3000/tcp  
prom/prometheus:latest Up 0.0.0.0:9090->9090/tcp  
prom/node-exporter    Up 0.0.0.0:9100->9100/tcp
```

Контейнеры хранят данные на томах и при пересоздании контейнеров (down, up -d) их настройки сохраняются.

Создание в Grafana соединения с Prometheus

The screenshot shows the Grafana interface with the following details:

- Left sidebar:** Home, Bookmarks, Starred, Dashboards, Explore, Drilldown, Alerting, Connections, Add new connection, Data sources (selected), Administration.
- Top navigation:** Home > Connections > Data sources > prometheus
- Search bar:** Search...
- Right sidebar:** Type: Prometheus, Alerting: Supported, Explore data, Build a dashboard.
- Main content:**
 - Connection settings:** Name: prometheus, Default: checked.
 - Success message:** Successfully queried the Prometheus API.
 - Buttons:** Delete, Save & test.
 - Connection section:** Prometheus server URL: http://10.0.2.15:9090.

Создание в Grafana соединения с Prometheus

Открыть в браузере: <http://localhost:3000> (admin/admin, пароль можно не менять при входе) – Grafana, <http://localhost:9090> – Prometheus

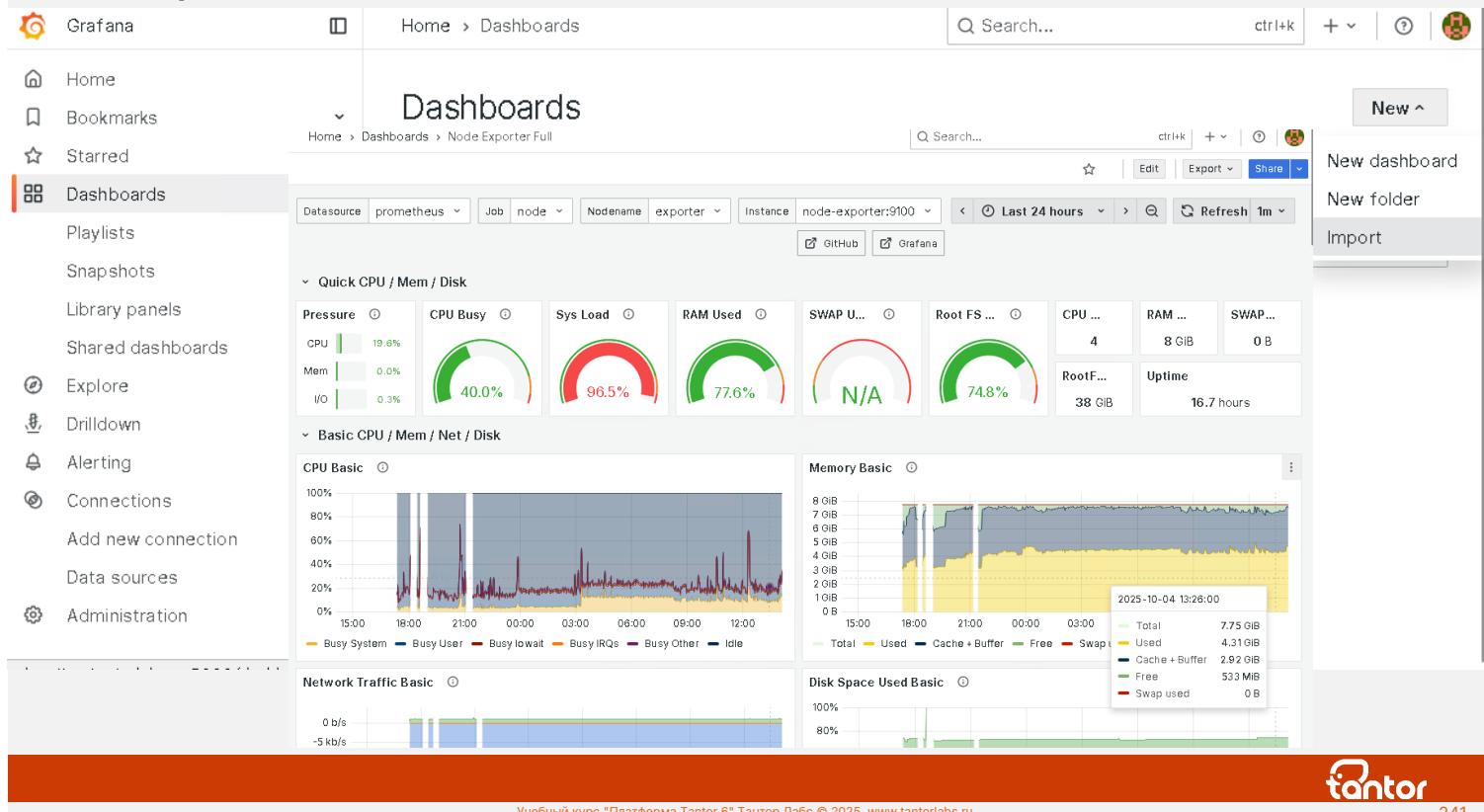
Prometheus получает метрики через службы, работающие в контейнерах "exporter". В примере из документации есть контейнер "node-exporter", собирающий метрики хоста на котором запускается этот контейнер. Этот контейнер доступен по порту <http://10.0.2.15:9100>.

В документации описано как добавить контейнеры для мониторинга экземпляров PostgreSQL на основе образа "[prometheuscommunity/postgres-exporter](#)". Приложение (называют "backend") в контейнере может подсоединяться к экземпляру по сети, а значит контейнер может работать на произвольном хосте, а не только на хосте, где работает экземпляр PostgreSQL. Приложение конфигурируется файлом (queries.yaml), в который можно вставить произвольные SQL-запросы для сбора любых метрик.

Для мониторинга pgbouncer можно использовать контейнер из образа "[spreaker/prometheus-pgbouncer-exporter](#)".

В Grafana можно раскрыть в меню Connections, в нём кликнуть на пункт Data sources. На странице кликнуть на кнопку "Add data source". Появится длинный список программ, из которых Grafana может забирать метрики. Первым в списке идёт Prometheus. Можно кликнуть на него и в поле Prometheus server URL вбить "<http://10.0.2.15:9090>", адрес по которому доступен Prometheus. Нажать на кнопку Save & test и соединение сохранится.

Импорт Dashboard в Grafana



The screenshot shows the Grafana interface with the following details:

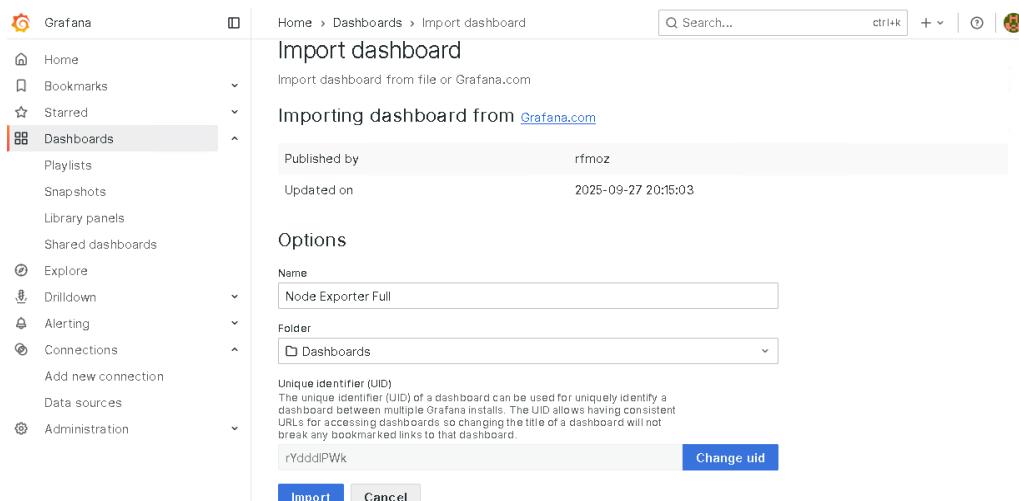
- Header:** Grafana, Home > Dashboards, Search bar, New, Import.
- Left sidebar:** Home, Bookmarks, Starred, Dashboards (selected), Playlists, Snapshots, Library panels, Shared dashboards, Explore, Drilldown, Alerting, Connections, Add new connection, Data sources, Administration.
- Current Dashboard:** Node Exporter Full, Datasource: prometheus, Job: node, Nodename: exporter, Instance: node-exporter:9100, Last 24 hours, Refresh 1m.
- Content:** A dashboard titled "Quick CPU / Mem / Disk" with several gauge charts for CPU, Mem, I/O, RAM, SWAP, and Root FS usage. It also includes a table for CPU, RAM, and SWAP usage and a table for RootFS and Uptime.
- Below:** Basic CPU / Mem / Net / Disk section with two charts: CPU Basic (stacked area chart) and Memory Basic (stacked area chart showing Total, Used, Cache + Buffer, Free, and Swap).
- Bottom:** Network Traffic Basic (line chart) and Disk Space Used Basic (stacked area chart).

tantor

241

Импорт Dashboard в Grafana

Для добавления dashboard (страницы, на которой будут отрисовываться метрики) кликнуть в меню на Dashboards, сна открывшейся странице справа вверху выбрать New -> Import. На открывшейся странице "Import dashboard" ввести число **1860**, нажать кнопку Load. Страница обновиться, на ней нажать синюю кнопку **Import**.

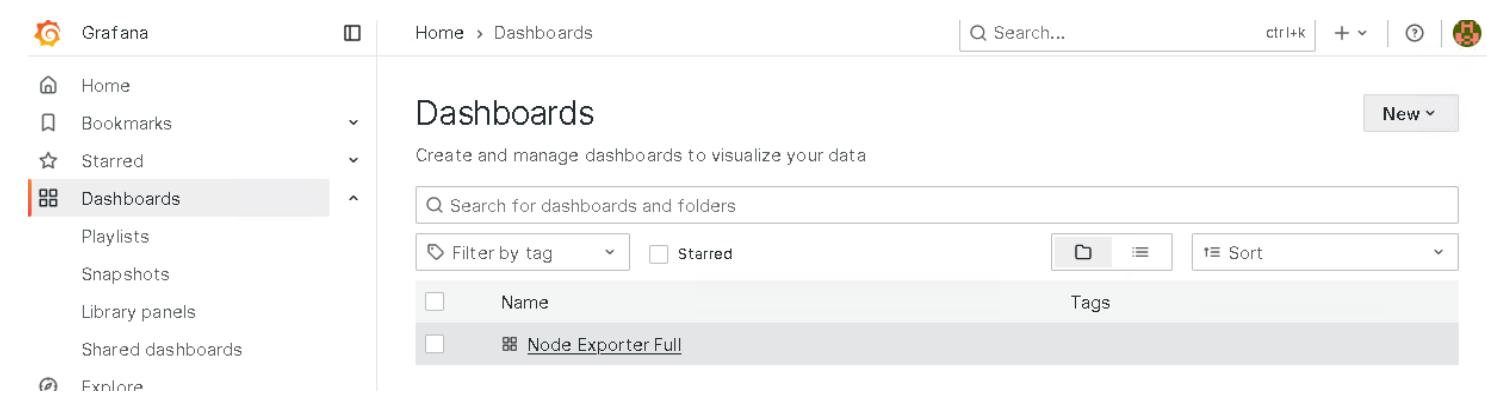


The dialog shows the following information:

- Home > Dashboards > Import dashboard
- Import dashboard from file or Grafana.com
- Importing dashboard from [Grafana.com](#)
- Published by: rfmozz
- Updated on: 2025-09-27 20:15:03
- Options:
 - Name: Node Exporter Full
 - Folder: Dashboards
 - Unique identifier (UID): rYdddIPWk (Change uid button)
- Buttons: Import, Cancel

Появится страница с графиками.

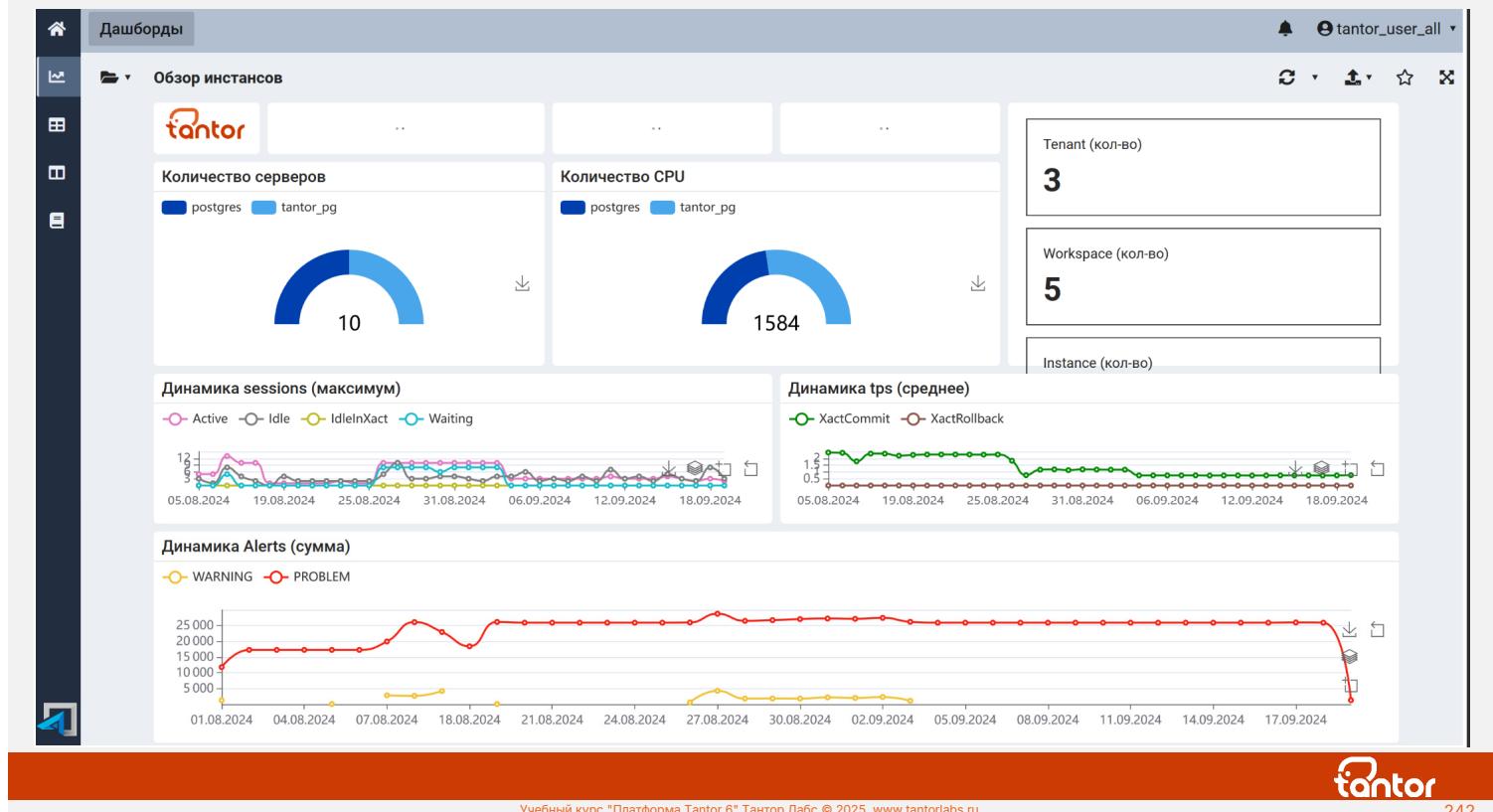
При повторных заходах на страницу Dashboards импортированное будет отображаться в виде строки "Node Exporter Full":



The screenshot shows the Grafana interface with the following details:

- Header:** Grafana, Home > Dashboards, Search bar, New.
- Left sidebar:** Home, Bookmarks, Starred, Dashboards (selected), Playlists, Snapshots, Library panels, Shared dashboards, Explore.
- Content:** Dashboards list with a search bar, filter by tag, and sort buttons. The "Node Exporter Full" dashboard is listed under the "Name" column.

Интеграция с Triafly



Учебный курс "Платформа Tantor 6" Тантор Лабс © 2025 www.tantolabs.ru

tantor
242

Интеграция с Triafly

Платформа хранит данные метрик не более 30 дней. Для хранения и анализ данных за любые интервалы времени Платформа может передавать данные в российскую программу "Триафлай.Аналитика". Эта программа собирает, хранит, анализирует, визуализирует собранные данные и строит по ним отчёты.

Со стороны Платформы Tantor достаточно создать токен типа "Датасеты". Он вводится в Триафлай.Аналитика в меню "Реестры" -> "(Input) Tenant API key" в поля столбца "API key". По умолчанию, данные будут загружаться из Платформы Tantor в Triafly раз в сутки.

Из этих данных создаются отчёты, по которым можно строить графики.

https://docs.tantolabs.ru/tp/6.1/admin/triafly_bi.html

Деинсталляция Платформы

```
docker-compose ls
NAME      STATUS      CONFIG FILES
eco       running(14) /opt/tantor/eco/docker-compose.yml
остановить и удалить контейнеры и сети docker:
docker-compose --env-file /opt/tantor/eco/platform.env -f
/opt/tantor/eco/docker-compose.yml down
удалить контейнеры:
docker rm -vf $(docker ps | grep tantorlabs | cut -d ' ' -f 1)
удалить образы, тома, сети контейнеров:
docker rmi -f $(docker images|grep tantorlabs |tr -s ' '|cut -d ' ' -f 3)
docker volume rm $(docker volume ls -q)
docker network rm $(docker network ls -q)
очистить место:
docker container prune -f
docker image prune -f
docker volume prune -f
docker system prune -f
rm -rf /opt/tantor/eco/
удалить агент Платформы: apt-get remove -y pmaagent
удалить директорию агента:
```

Деинсталляция Платформы

Посмотреть проекты docker-compose: `docker-compose ls`

```
NAME      STATUS      CONFIG FILES
eco       running(14) /opt/tantor/eco/docker-compose.yml
```

Если на хосте docker используется только [Платформой](#), то остановить и удалить контейнеры и сети docker командами: `cd /opt/tantor/eco`

```
docker-compose --env-file /opt/tantor/eco/platform.env -f
/opt/tantor/eco/docker-compose.yml down
```

Если нужно создать заново:

```
docker-compose --env-file /opt/tantor/eco/platform.env -f
/opt/tantor/eco/docker-compose.yml up -d
```

удалить контейнеры [Платформы](#):

```
docker rm -vf $(docker ps | grep tantorlabs | cut -d ' ' -f 1)
```

*если нужно удалить **все** контейнеры:*

```
docker rm -vf $(docker ps -aq)
```

удалить все образы [Платформы](#):

```
docker rmi -f $(docker images|grep tantorlabs |tr -s ' '|cut -d ' ' -f 3)
```

*если нужно удалить **все** образы:*

```
docker rmi -f $(docker images -aq)
```

удалить все тома: `docker volume rm $(docker volume ls -q)`

удалить все сети докера: `docker network rm $(docker network ls -q)`

очистить место от контейнеров, которые остановлены:

```
docker container prune -f
```

```
docker image prune -f
```

```
docker volume prune -f
```

```
docker system prune -f
```

удалить директорию, куда устанавливается Платформа: `rm -rf /opt/tantor/eco`

удалить агент Платформы: `apt-get remove -y pmaagent`

удалить директорию агента: `rm -rf /var/lib/pma`

Установочные файлы

← /opt/tantor/eco6				← /opt/tantor/eco6/images			
.n	Name	Size	Modify time	.n	Name	Size	UP--DIR
/..		UP--DIR	Oct 9 16:28	/..		UP--DIR	
/images		4096	Sep 23 09:50	backend-6.0.1.tar		40199K	
*docker-compose		52985K	Sep 10 12:20	keeper-6.0.1.tar		26785K	
*docker-compose.yml		9791	Sep 23 09:35	maindb-6.0.1.tar		773385K	
*installer		5239044	Sep 10 22:25	maindb_postgres-6.0.1.tar		451816K	
release_tag		6	Sep 10 22:32	nats-6.0.1.tar		23583K	
*sdc-tantor		5038645	Sep 10 22:32	operdb-6.0.1.tar		785064K	
				operdb_postgres-6.0.1.tar		891990K	
				pg-explain-6.0.1.tar		1218M	
				pg-monitor-6.0.1.tar		909854K	
				pg_anon-6.0.1.tar		590733K	
				pg_bouncer-6.0.1.tar		318143K	
				pg_configurator-6.0.1.tar		165041K	
				pmmfront_angmat_tantor-6.0.1.tar		85478K	
				reverse-proxy-6.0.1.tar		527427K	
				swagger-ui-6.0.1.tar		122677K	
				tns_db-6.0.1.tar		272348K	
				tns_migrations-6.0.1.tar		871322K	

```
ls -l /opt/tantor/eco6
ls -l /opt/tantor/eco6/images/
```

Установочные файлы

После разархивирования дистрибутива появятся файлы:

исполняемый файл docker-compose, на случай если он не установлен:

```
/opt/tantor/eco6/docker-compose -v
```

Docker Compose version v2.17.0

Версия может отличаться от той, что установлена:

```
/usr/local/bin/docker-compose -v
```

Docker Compose version v2.39.4

Инсталлятор не будет переустанавливать docker-compose, если он уже установлен.

Файл docker-compose.xml, по которому будут создаваться контейнеры

Утилита диагностики sdc-tantor

Исполняемый файл инсталлятора

Директория images, в которой находятся архивы с образами контейнеров.

<https://docs.tantorlabs.ru/tp/6.1/admin/wizard.html>

Установка Платформы

```
cd /opt/tantor/eco6
sudo /opt/tantor/eco6/installer
astra@tantor:/opt/tantor/eco6$ sudo /opt/tantor/eco6/installer
Writing diagnostic information into the file [/opt/tantor/eco6/tantor_eco_installation.log]
#####
#          Welcome to Tantor Platform installer.          #
#####
- Checking Tantor Platform dependencies:
- docker installed...                                [Yes]
- docker installed required version...               [Yes]
- docker started...                                [Yes]
- Tantor Platform will be installed, version 6.0.1. Proceed? (Y/N) : y
- Tantor Platform installation initial actions...
- Enter Tantor Platform administrator email: student@dbal.ru
- Enter Tantor Platform administrator name: student
- Enter Tantor Platform domain name: education.tantorlabs.ru
- Do you have SSL certificates match domain name education.tantorlabs.ru? (Y/N) : n
- Do you want to create the self-signed certificates for domain name
  education.tantorlabs.ru? (Y/N) : y
#####
- Proceed? (Y/N) : y
```

Установка Платформы

Установка выполняется с правами root. Можно использовать sudo.

Нужно запустить файл `installer`.

`administrator email` будет использоваться как имя для входа в Платформу.

`domain name` это имя хоста, на котором устанавливается Платформа. Имя должно разрешаться в любой из IP-адресов хоста, на котором устанавливается Платформа. Имя будет использоваться Агентами для связи с Платформой по портам 443 и 4222.

Инсталлятор может создать самоподписанные сертификаты SSL(TLS).

Установка Платформы

- Self-signed SSL certificates creation for domain name `education.tantorlabs.ru...`
- Which platform setup deployment you want to install? (1/2/3):
 1. Platform
 2. Platform + pg_explain
 3. Platform + pg_explain + pg_monitor **3**
- Create and enter a password to log in to the platform (8 symbols minimum):
- Re-enter your platform password: **Student123!**
- Create directories `/opt/tantor/eco/nginx/ssl...` [Yes]
- Copy SSL certificates into directory `/opt/tantor/eco/nginx/ssl...` [Yes]
- Copy template `docker-compose.yml` into directory `/opt/tantor/eco/...` [Yes]
- The following configurations will be set:
- Tantor Platform domain name: `education.tantorlabs.ru`
- Tantor Platform administrator email: `student@dbal.ru`
- Tantor Platform administrator name: `student`
- Proceed? (Y/N): **y**
- Change template `/opt/tantor/eco/docker-compose.yml...` [Yes]
- Load docker images... [Yes]
- Start Tantor Platform... [Yes]

Установка Платформы

Инсталлятор предлагает выбрать один из трёх вариантов установки. Введенный пароль не будет меняться при первом входе. Пароль не будет выводиться на экран. Установка займёт примерно 10 минут. Может быть выдано предупреждение о том, что Расширенная аналитика требует дополнительного конфигурирования:

- SSH keys `id_rsa|id_rsa.pub` don't exist in directory `/opt/tantor/eco/ssh`, due to this reason the Advanced Analytics tab will not work properly

```
#####
# Tantor Platform version 6.0.1 is installed and running. #
Please connect to Tantor Platform using:
Tantor Platform URL: https://education.tantorlabs.ru
Tantor Platform administrator email/login: student@dbal.ru
Tantor Platform administrator password: ****
#####
```

Платформа успешно установлена.

После установки можно открыть адрес `https://education.tantorlabs.ru` в браузере. Так как ключ самоподписанный, то браузер выдаст предупреждение, нужно подтвердить, что можно подсоединяться к хосту.

После ввода имени и пароля, появится всплывающее окно с лицензионным соглашением. Слева в конце окна есть галочка о принятии лицензионного соглашения и кнопка ОК.

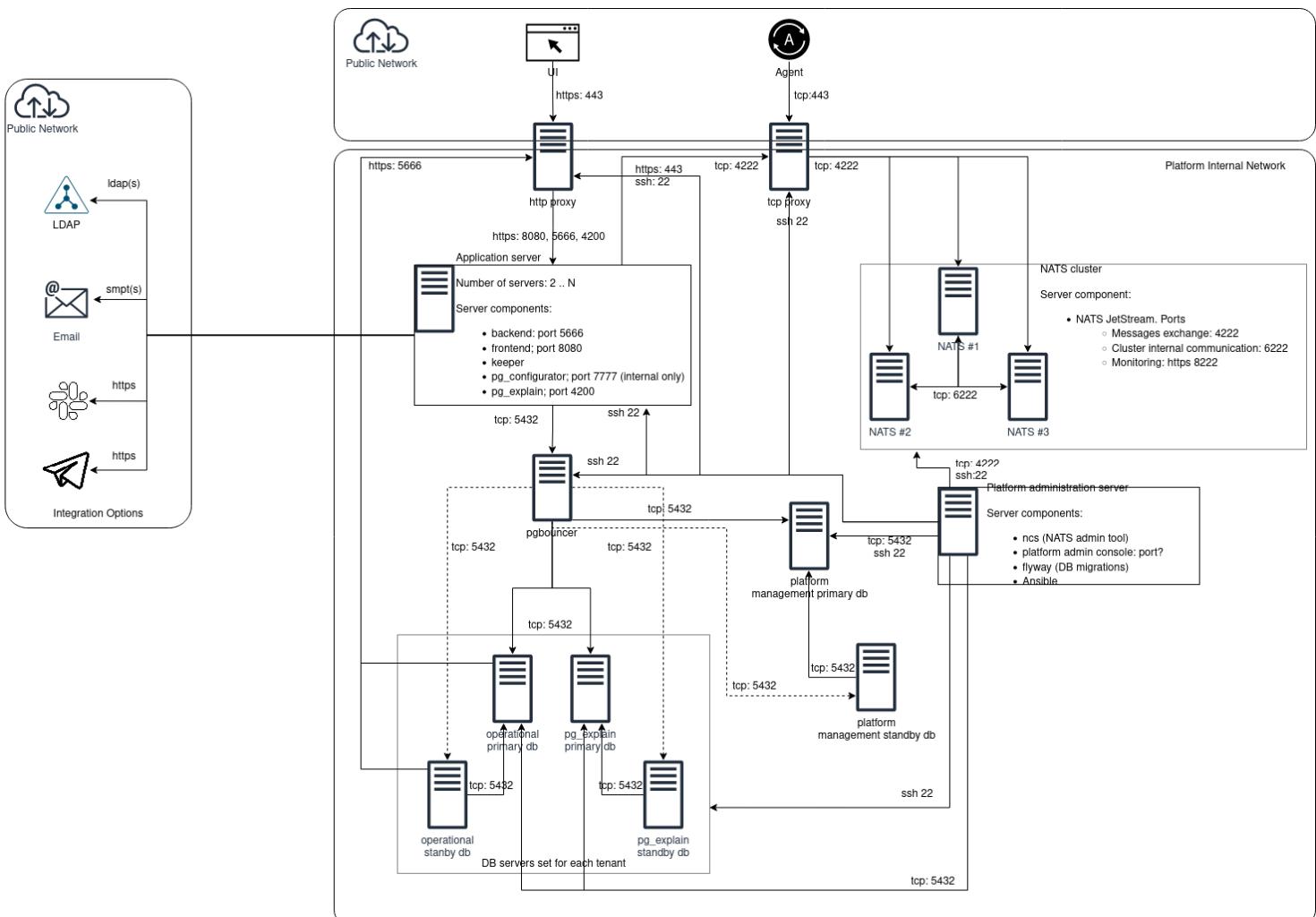
Установка Платформы вне докера

- Установка платформы возможна на любое число серверов (виртуальных машин). Для режима отказоустойчивости (High Availability) рекомендуется использовать не менее 13 хостов или виртуальных машин:
- 2 - OperDB;
- 2 - TensorDB;
- 2 - App (backend, frontend, pgexplain);
- 1 - reverse proxy;
- 1 - Nats nginx proxy;
- 3 - Nats;
- 1 - Admin_server;

Установка Платформы вне докера

Платформа может устанавливаться вне докера на физические хосты и виртуальные машины. При установке используется ansible playbook.

Схема взаимодействия служб Платформы при отказоустойчивой установке:



Установка Платформы вне докера

- Operdb - хранение данных
- Operdb replication - репликация базы данных
- Tensordb - база данных для анализа информации
- Tensordb replication - репликация базы данных для анализа
- Migrations - службы миграции во внутренние базы данных
- Pgbounce - управление пулом соединений внутренних баз данных
- Nats cluster - брокер сообщений для компонентов системы
- Nats nginx proxy - балансировщик соединений к NATS
- Backend (включает в себя keeper, pgconfigurator) - сервер API, keeper - передает данные от агентов во внутренние базы данных, pgconfigurator - конфигурирование PostgreSQL
- pgexplain - сервис анализа запросов
- Frontend - пользовательский интерфейс
- Reverse Proxy - балансировщик соединений

Установка Платформы вне докера

На слайде приведены названия служб Платформы.

На каждом сервере (виртуальной машине) группы должны быть открыты порты. Можно использовать и другие порты. По умолчанию используются:

Operdb: server port 25432

Operdb replica: server port 25432

Tensordb: server port 5432

Tensordb replica: server port 5432

Pgbouncer server port 15432

Nats cluster server port 6222 (each server)

Nats nginx server port 4222, 14222

App server ports: 5666 (backend), 7777 (internal pgconfigurator) 4200 (pgexplain), 8080 (frontend)

Reverse_proxy: 443 (https), 80 (http), 5666 (upstream to backend), 8081 (pmaagent download port), pgalerts 5667.

Помимо начальной установки вне докера можно вынести кластера баз данных PostgreSQL из контейнеров докер на хосты. Общая схема выноса описана в документации к 4 версии:

https://docs.tantorlabs.ru/tp/4.2/db_move.html

Начиная с 5 версии Платформы, установить Платформу в отказоустойчивом режиме помогает техническая поддержка Тантор.



12-4

Установка Агента Платформы Tantor

Установка Агента Платформы

- Способы установки:
 - › в веб-интерфейсе на странице добавления экземпляра (Add instance)
 - › в командной строке
 - › через Ansible
- Агент состоит из файлов:
 - › исполняемый /usr/sbin/pmaagent размером ~30Мб.
 - › /etc/sysconfig/pmaagent для переменных окружения
 - › /etc/logrotate.d/pmaagent параметры ротации логфайла
 - › /usr/lib/systemd/system/pmaagent.service
 - › файлов в директории /var/lib/pma/agent

Установка Агента Платформы

Установка Агента может выполняться:

1) в веб-интерфейсе на странице добавления экземпляра (Add instance)

https://docs.tantorlabs.ru/tp/6.1/instances/instance_adding.html

2) в командной строке

https://docs.tantorlabs.ru/tp/6.1/instances/agent_instance_cli.html

3) через Ansible

https://docs.tantorlabs.ru/tp/6.1/instances/agent_ansible.html

https://github.com/TantorLabs/ansible_tantor_agent/archive/refs/heads/main.zip

Дистрибутив (пакет deb или rpm) доступен по адресу хоста Платформы:

https://хост_Платформы:8443/packages/pmaagent.deb

https://хост_Платформы:8443/packages/pmaagent.rpm

Пример команды для скачивания дистрибутива:

`wget --no-check-certificate`

<https://education.tantorlabs.ru:8443/packages/pmaagent.deb>

Установка выполняется пользователем root или через sudo.

Если установка выполняется из дистрибутива не доступного всем, то команда установки:

`sudo apt-get install ./pmaagent.deb`

выдаст предупреждение, которое можно игнорировать:

N: Download is performed unsandboxed as root as file '/home/astra/pmaagent.deb' couldn't be accessed by user '_apt'. - pkgAcquire::Run (13: Permission denied)

Агент состоит из файлов:

1) исполняемый /usr/sbin/pmaagent размером ~30Мб.

2) /etc/sysconfig/pmaagent в котором можно задать переменные окружения агенту

3) /etc/logrotate.d/pmaagent, в котором указаны параметры ротации логфайла агента.

Настройки по умолчанию: хранить максимум 7 суток до 10 файлов, файлы сжимать.

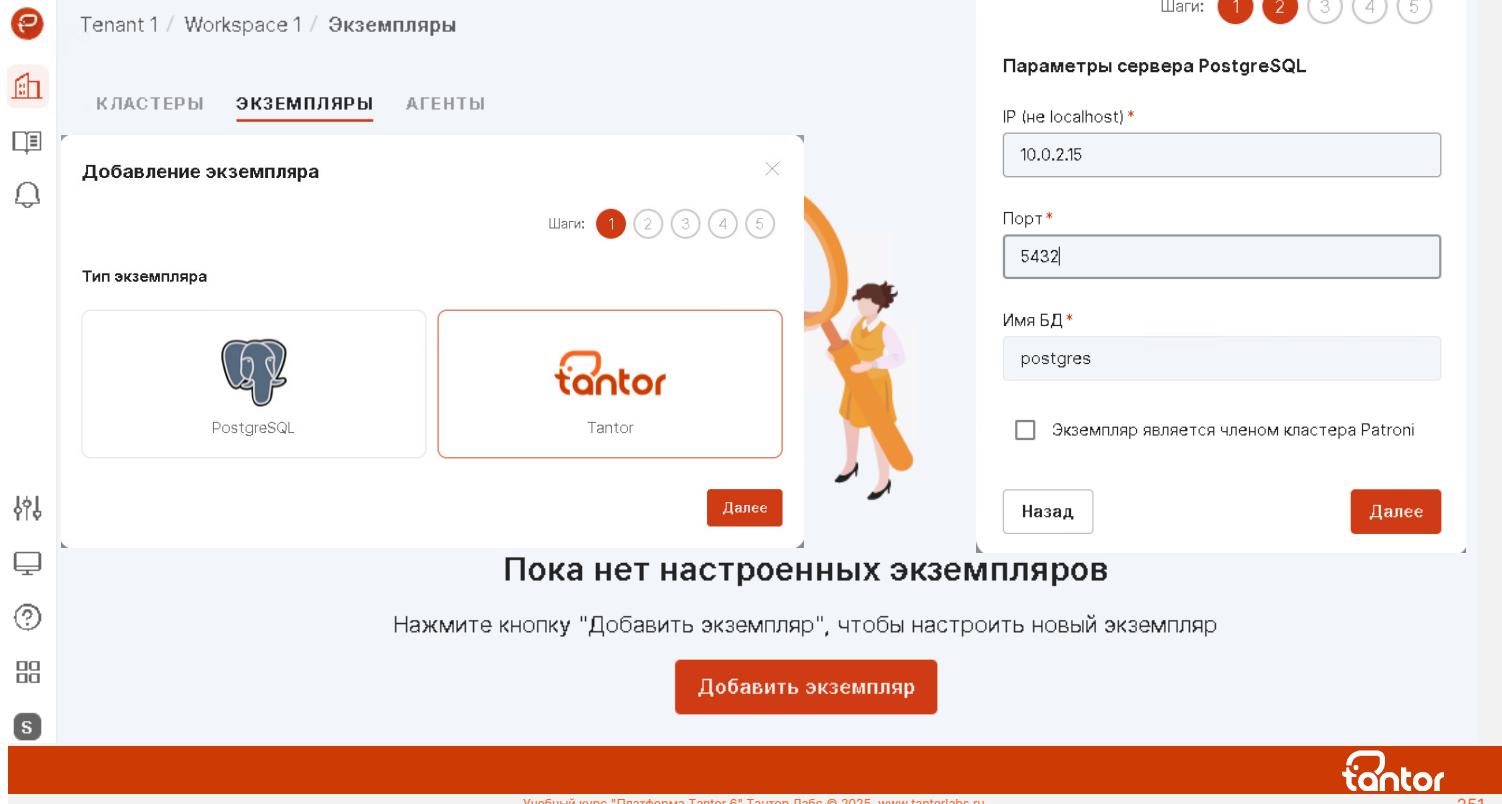
4) /usr/lib/systemd/system/pmaagent.service с параметрами запуска агента через systemd.

Агент запускается под пользователем операционной системы postgres.

5) файлов в директории /var/lib/pma/agent. Файл конфигурации

/var/lib/pma/agent/agent.yml содержит детали подключения к Платформе и создается при конфигурировании агента командой sudo pmaagent configure.

Установка Агента в веб-интерфейсе



Добавление экземпляра

Шаги: 1 2 3 4 5

Тип экземпляра

PostgreSQL Tantor

Далее

Назад

Далее

Пока нет настроенных экземпляров

Нажмите кнопку "Добавить экземпляр", чтобы настроить новый экземпляр

Добавить экземпляр

Параметры сервера PostgreSQL

IP (не localhost) *

10.0.2.15

Порт *

5432

Имя БД *

postgres

Экземпляр является членом кластера Patroni

tantor 251

Установка Агента в веб-интерфейсе

Агент используется для сбора метрик и выполнения команд в базах данных экземпляров PostgreSQL. На одном IP-адресе может работать только один агент.

Установка агента через веб-интерфейс выполняется при добавлении экземпляра PostgreSQL (в том числе, управляемого Patroni).

Нужно на странице Тенант -> Рабочее пространство на вкладке ЭКЗЕМПЛЯРЫ нажать кнопку "Добавить экземпляр". Также можно перейти во складку КЛАСТЕРЫ и нажать кнопку "Добавить кластер Patroni" (более точно было бы "Добавить экземпляр мастера или реплики, входящих в кластер Patroni"). Разница только в том, что на этой странице галочка "Экземпляр является членом кластера Patroni" будет установлена.

После нажатия на кнопку "Добавить экземпляр" появится окно визарда, где на первом шаге выбирается Тип экземпляра: PostgreSQL или Tantor. Разделение сделано для учёта лицензий: все сборки Tantor Postgres включают лицензии на использование Платформы и не исчерпывают число учитываемых ядер процессоров.

Если экземпляр под управлением Patroni, то надо выбрать галочку и указать порт REST-API интерфейса Patroni, доступном на этом IP-адресе (обычно, 8008).

https://docs.tantorlabs.ru/tp/6.1/instances/instance_adding.html

Установка Агента в веб-интерфейсе

Добавление экземпляра

Шаги: 1 2 3 4 5

На IP-адресе 10.0.2.15 нет активного Агента. Пожалуйста, выполните следующие команды на выделенном хосте для установки и активации Агента. Всё команды должны выполняться от имени root-пользователя.

Выберите ваш дистрибутив Linux

Astra Linux

1.6+

У меня нет доступа к Интернету, используйте локальную/оффлайн установку

Скачайте и установите Агент

```
mkdir -p /etc/apt/sources.list.d && sh -c 'echo "deb [arch=amd64] https://nexus.tantorlabs.ru/repo" > /etc/apt/sources.list.d/pmaagent.list' && wget -q -O - https://public.tantorlabs.ru/tantorlabs.ru.asc | apt-key add - & apt-get update & apt-get install -y pmaagent=6.0.1
```



Обновить статус Агента

Агент не активен на IP 10.0.2.15

Настройте Агента

```
su postgres -s /bin/bash -c "/usr/sbin/pmaagent configure --encoded_metadata"
```



Метаданные Агента. Вставьте следующий ключ во время настройки Агента (Показана только первая строка ключа, пожалуйста, используйте кнопку копирования)

```
PvwyzfuPN+eVP0Ohsnqgy2bjmWCPqwhOD6FmWLqoldsdBt+ReHSISouZ8elenA2gAn4NfDqj6b4T...
```



Запустите Агент и включите службу (приведен пример только для Systemd)

```
systemctl daemon-reload && systemctl restart pmaagent && systemctl enable pmaagent
```



Назад

Далее



252

Учебный курс "Платформа Tantor 6" Тантор Лабс © 2025 www.tantorlabs.ru

Установка Агента в веб-интерфейсе

На третьем шаге выбирается операционная система. Выбор определяет набор команд для установки и конфигурирования агента. Дистрибутив агента можно скачать из репозитория nexus.tantorlabs.ru или, если поставить галочку "У меня нет доступа к Интернету, используйте локальную/оффлайн установку".

После установки пакета Агент не запускается, так как его файл конфигурации ещё не был создан. Нажимать кнопку "Обновить статус Агента" не нужно, она не покажет работающий Агент.

Команду `/usr/sbin/pmaagent configure --encoded_metadata` можно выполнять и под пользователем операционной системы `postgres` или `root`:

```
root@tantor:~# pmaagent configure --encoded_metadata
#####
#          PMA Agent. Version 6.0.1.
#      Stage: Agent Metadata based initialization.
# Local date/time: Friday, 10-Oct-25 14:42:45 MSK
#####
- Starting the Agent Metadata based initialization... [OK]
- Please provide the Agent Metadata (available on the installation user interface):
PvwyzfuPN+eVP0Ohsnqgy2bjmWCPqwhOD6FmWLqoldsdBt+ReHSISouZ8elenA2gAn4NfDqj6b4T...
- Saving the configuration data into the file /var/lib/pma/agent/agent.yml]... [OK]
- PMA Agent has been successfully pre-configured. DONE.
```

Агент сконфигурирован, но не запущен и экземпляр не добавлен. Для запуска службы Агента нужно выполнить:

```
root@tantor:~# systemctl daemon-reload
root@tantor:~# systemctl restart pmaagent
root@tantor:~# systemctl enable pmaagent
```

Агент будет запущен. Если на странице не нажимается кнопка "Далее", то это ошибка в веб-интерфейсе и нужно нажать на значок квадратики (копирование текста) и кнопка Далее перестанет показывать всплывающее окно и позволит пройти дальше. В любом случае, можно прервать работу визарда и зайти в него позже, те шаги, которые выполнили выполнить повторно не нужно.

Добавление экземпляра PostgreSQL выполняется в командной строке на четвёртом шаге:

```
root@tantor:~# pmaagent configure --config_postgres -d postgres -p 5432
```

Регистрация Агента в Платформе через веб-интерфейс

```
#####
#          PMA Agent. Version 6.0.1.
#
#          Stage: Agent configuration / on-premise PostgreSQL instance
#
#####
- Please type 'proceed' to continue or 'decline' to cancel: proceed
- Parameters:
+ Type of instance : on-premise PostgreSQL
+ PostgreSQL default host(empty: "unix socket") :
+     listener port : 5432
+ Database name : postgres
+ PostgreSQL user (i.e. super user) : postgres
+ PostgreSQL agent's user name : pma_user (this database user will be
created. Agent connects to the monitored PostgreSQL instance using this user name)
- WARNING: During the process, the PostgreSQL instance will be reconfigured and reloaded.
- Please make sure the instance of PostgreSQL that is listening the port [5432]
- is properly configured, and its client authentication (pg_hba) is consistent.
- Please provide password for database user (i.e: pma_user. Agent uses this
database user to establish database connections): pma_user
- Testing database connection using provided host [...]... [OK]
- Checking if the PostgreSQL instance is in recovery mode... [OK]
- Checking if the [postgresql-contrib] packaged is installed..... [OK]
- Checking if the [pg_stat_statements] extension is installed..... [OK]
- Checking if the [pg_store_plans] extension is installed..... [OK]
```

Регистрация Агента в Платформе через веб-интерфейс

При добавлении экземпляра под управление Агента запрашивается подтверждение, надо набрать **proceed**. Набрать пароль для пользователя (роли) PostgreSQL **pma_user**. Если пользователь существует, то после ошибки создания пользователя нужно набрать что создавать не нужно и что нужно поменять ему пароль. Отказаться от смены пароля нельзя - экземпляр не будет добавлен в Платформу.

- Creating the [pma_user] database user... [FAIL]
- Database user [pma_user] already exists.
- Do you wish to create a new database user? (yes/no): **no**
- Do you wish to change password for pma_user? (yes/no): **yes**
- Checking the pg_hba configuration... [OK]
- Updating the [/var/lib/postgresql/tantor-se-17/data/pg_hba.conf] file [OK]
- Reloading the PostgreSQL configuration... [OK]
- Testing connection to database with provided parameters... [OK]
- Overwriting the [/var/lib/pma/agent/agent.yml] configuration file... [OK]
- Reconfiguration has been completed successfully. OK.

- Please click 'NEXT' in the installation web interface to proceed to step #5



Установка и регистрация Агента в командной строке

The screenshot shows the Tantor platform's tenant settings interface. On the left, there's a sidebar with various icons and a list of settings: Пользователи и группы, Пользователи, Группы, Каналы уведомлений, Шаблоны писем, Внешняя аутентификация, Настройки тенанта (SMTP), Резервное копирование, and Токены (highlighted in red). The main area shows 'Tenant 1 / Настройки тенанта / Токены' with a 'Токены' tab selected. A modal window titled 'Добавление нового токена' is open, showing fields for 'Метка*' (agent), 'Срок действия*' (11/11/2025 00:00), and 'Область применения*' (Установка). At the bottom of the modal is a red 'Добавить новый токен' button. To the right of the modal, a large token value is displayed with a warning: 'Обязательно скопируйте этот токен. Вы больше не сможете увидеть его снова!' (Warning icon). Below the token is a 'Скопировать в буфер обмена и закрыть' button. The main area also shows a message 'Здесь пока нет токенов ...' and a red 'Добавить новый токен' button. The footer of the page includes the Tantor logo and the text 'Учебный курс "Платформа Tantor 6" Тантор Лабс © 2025 www.tantolabs.ru'.

Установка и регистрация Агента в командной строке

Установка Агента через командную строку (CLI) такая же, как и через веб-интерфейс.

Любым способом скачивается и устанавливается Агент:

```
curl "https://education.tantolabs.ru:8443/packages/pmaagent.deb" -o /tmp/pmaagent.deb -k
wget --no-check-certificate
https://education.tantolabs.ru:8443/packages/pmaagent.deb -O /tmp/pmaagent.deb
apt-get autoremove -y
apt-get remove -y pmaagent
apt-get install -y /tmp/pmaagent.deb
```

Для регистрации Агента в Платформе нужен токен. Для этого нужно зайти в веб-интерфейсе Платформы в меню "Настройки тенанта" -> "Токены" -> "Добавить новый токен".

В открывшемся окне ввести название в поле "Метка", указать срок действия, выбрать Область применения "Установка". Появится окно с токеном, который можно скопировать в буфер обмена.

Если токен будет потерян, то восстановить его нельзя, только выпустить новый токен. Прежний токен можно удалить в любой момент и он перестанет действовать.

Токен можно использовать **многократно** до истечения срока действия.

```
root@tantor:~# pmaagent register --access-token
```

```
'eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdWIiOiJwbWEiLCJleHAiOjE3NjI4OTQ3OTksImhdCI6MTc2MDEwODUyMSwianRpIjoizTczMjZhYzgtMDN1NC00MTzkLT11MjEtNjdiODUxOGVkMmI0IiwidGVuYW50SUQiojEsInRlbmFudEtlesI6IjczZGY2YTA1LTc5MTktNGQ5Yi05MDg2LTAzYWRjOGQxZDU1NiIsInNjb3B1Ijoisu5TVEFMCJ9.bksiGsCEV10NN5LRUIy_d6wsAv8xEhhb8myjSEjfuo_ --host
```

```
'10.0.2.15' --workspace-name 'Workspace 1' --endpoint
```

```
'education.tantolabs.ru:4222'
```

Agent successfully registered

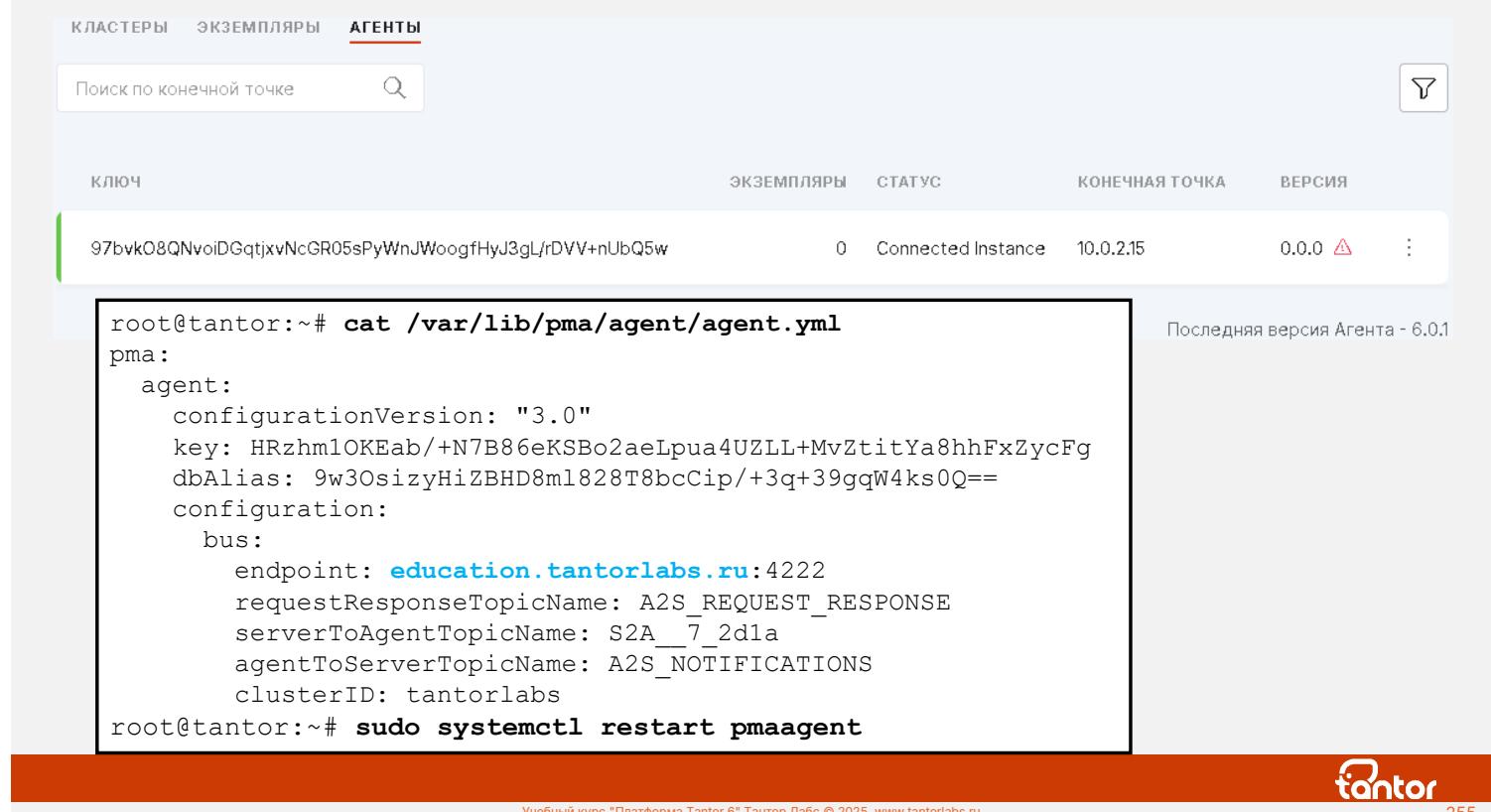
В параметре host лучше указывать IP-адрес сетевого интерфейса хоста. Имя пространства обязательно и чувствительно к регистру. Имя хоста или IP-адрес Платформы и порт обязательны. По указанному имени Агент будет соединяться с Платформой. Значение можно поменять в файле /var/lib/pma/agent/agent.yml

Агент можно дегистрировать, даже если он обслуживает экземпляры PostgreSQL:

```
root@tantor:~# pmaagent deregister --confirm
```

```
Agent successfully deregistered
```

Перезапуск Агента после регистрации в Платформе



КЛАСТЕРЫ ЭКЗЕМПЛЯРЫ АГЕНТЫ

Поиск по конечной точке

КЛЮЧ	ЭКЗЕМПЛЯРЫ	СТАТУС	КОНЕЧНАЯ ТОЧКА	ВЕРСИЯ	⋮
97bvk08QNvoiDGqtjxvNcGR05sPyWnJWoogfHyJ3gL/rDVV+nUbQ5w	0	Connected Instance	10.0.2.15	0.0.0 ⚠	

```
root@tantor:~# cat /var/lib/pma/agent/agent.yml
pma:
  agent:
    configurationVersion: "3.0"
    key: HRzhm10KEab/+N7B86eKSBo2aeLpua4UZLL+MvZtitYa8hhFxZycFg
    dbAlias: 9w3OsizyHiZBHD8ml828T8bcCip/+3q+39gqW4ks0Q==
    configuration:
      bus:
        endpoint: education.tantorlabs.ru:4222
        requestResponseTopicName: A2S_REQUEST_RESPONSE
        serverToAgentTopicName: S2A_7_2d1a
        agentToServerTopicName: A2S_NOTIFICATIONS
        clusterID: tantorlabs
root@tantor:~# sudo systemctl restart pmaagent
```

Последняя версия Агента - 6.0.1

Перезапуск Агента после регистрации в Платформе

Если Агент уже зарегистрирован, повторная регистрация выдаст ошибку:

Agent already exists

После установки Агента и его регистрации в Платформе Агент нужно рестартовать, иначе Платформа будет показывать, что Агент не доступен:

`sudo systemctl restart pmaagent`

Рестартовать Агент можно также после добавления экземпляров. Если в Агенте не зарегистрирован ни один экземпляр, то в веб-интерфейсе Платформы версия Агента показывается как **0.0.0**.

Экземпляры PostgreSQL можно регистрировать в командной строке.

По умолчанию, экземпляр PostgreSQL прослушивает только локальный сетевой интерфейс.

Список интерфейсов для прослушивания можно посмотреть в параметре `listen_addresses`:

`psql -c "\dconfig listen_addresses"`

List of configuration parameters

Parameter	Value
listen_addresses	*

Если пароль пользователю `postgres` кластера PostgreSQL не установлен, его можно установить:

`psql -c "alter user postgres password 'postgres';"`

Регистрация экземпляра PostgreSQL в командной строке

КЛЮЧ	ЭКЗЕМПЛЯРЫ	СТАТУС	КОНЕЧНАЯ ТОЧКА	ВЕРСИЯ	
97bvk08QNvoiDGqtjxvNcGR05sPyWnJWoogfHyJ3gL/rDVV+nUbQ5w	2	Connected Instance	10.0.2.15	6.0.1	⋮

```
pmaagent instances add --name 'Primary' --environment 'SERVER' --type 'TantorDB' --db-host '10.0.2.15' --db-port '5432' --db-user 'pma_user' --db-pass 'pma_user' --db-admin-user 'postgres' --db-admin-pass 'postgres' --confirm
#####
# PMA Agent. Version 6.0.1.
# Add on-premise PostgreSQL instance
#####
- Testing database connection using provided host [10.0.2.15]... [OK]
- Checking if the PostgreSQL instance is in recovery mode... [OK]
- Checking if the [postgresql-contrib] packaged is installed..... [OK]
- Checking if the [pg_stat_statements] extension is installed..... [OK]
- Checking if the [pg_store_plans] extension is installed..... [OK]
- Creating the [pma_user] database user... [FAIL]
- Checking the pg_hba configuration... [OK]
- Updating the [/var/lib/postgresql/tantor-se-17/data/pg_hba.conf] file... [OK]
- Reloading the PostgreSQL configuration... [OK]
- Testing connection to database with provided parameters... [OK]
Instance [9e2a2a6b-56ba-4e14-9b59-bcbb20499fd6] has been added successfully.
```

Регистрация экземпляра PostgreSQL в командной строке

Регистрация экземпляров:

```
pmaagent instances add --name 'Primary' --environment 'SERVER' --type 'TantorDB' --db-host '10.0.2.15' --db-port '5432' --db-user 'pma_user' --db-pass 'pma_user' --db-admin-user 'postgres' --db-admin-pass 'postgres' --confirm
pmaagent instances add --name 'Replica' --environment 'SERVER' --type 'TantorDB' --db-host '10.0.2.15' --db-port '5433' --db-user 'pma_user' --db-pass 'pma_user' --db-admin-user 'postgres' --db-admin-pass 'postgres' --confirm
#####
# PMA Agent. Version 6.0.1.
# Add on-premise PostgreSQL instance
#####
- Testing database connection using provided host [10.0.2.15]... [OK]
- Checking if the PostgreSQL instance is in recovery mode... [OK]
- Checking if the [postgresql-contrib] packaged is installed..... [OK]
- Checking if the [pg_stat_statements] extension is installed..... [OK]
- Checking if the [pg_store_plans] extension is installed..... [OK]
- Creating the [pma_user] database user... [FAIL]
- Checking the pg_hba configuration... [OK]
- Updating the [/var/lib/postgresql/tantor-se-17/data/pg_hba.conf] file... [OK]
- Reloading the PostgreSQL configuration... [OK]
- Testing connection to database with provided parameters... [OK]
Instance [9e2a2a6b-56ba-4e14-9b59-bcbb20499fd6] has been added successfully.
```

Если аутентификация без пароля (trust в pg_hba.conf), то для регистрации экземпляра пароль не требуется и его можно не указывать в параметре **--db-admin-pass**.

В командной строке можно добавлять экземпляры под управлением Patroni.

Если экземпляр зарегистрирован, то команда выдаст ошибку:

```
Failed to add a new instance [Instance with the provided host [10.0.2.15] and port [5433] exists]
```

https://docs.tantorlabs.ru/tp/6.1/instances/agent_instance_cli.html



12-5

Настройка Расширенной аналитики

Конфигурирование экземпляра для Аналитики

- Параметры конфигурации экземпляров для того, чтобы полностью использовать возможности Аналитики:

```
shared_preload_libraries = 'pg_stat_statements, pg_store_plans, auto_explain'
logging_collector = 'on' # коллектор рекомендуется использовать
log_destination = 'stderr' # значение по умолчанию
log_filename = 'postgresql-%F.log' # пример: postgresql-2025-12-01.log
log_line_prefix = '%m [%p:%v] [%d] %r %a' # больше деталей у сообщений
lc_messages = 'en_US.UTF-8' # чтобы не зависеть от других параметров локализации
listen_addresses = '*' # по умолчанию экземпляр прослушивает только localhost
deadlock_timeout = '1s' # значение по умолчанию, можно менять
log_lock_waits = 'on' # логирует события ожидания, больше чем deadlock_timeout
log_min_duration_statement = '10s' # логировать команды, которые работают дольше
pg_store_plans.min_duration = '10s'
auto_explain.log_min_duration = '10s'
auto_explain.log_analyze = 'on'
auto_explain.log_buffers = 'on'
auto_explain.log_nested_statements = 'on' # отслеживать запросы в подпрограммах
track_io_timing = 'on' # если хочется видеть время i/o в планах
```

Конфигурирование экземпляра для Аналитики

Для Расширенной аналитики нужно конфигурирование кластера PostgreSQL и хоста.

Нужно установить параметры конфигурации мониторируемого экземпляра PostgreSQL:

библиотеки, которые использует аналитика. Вместо pg_stat_statements, pg_store_plans можно использовать pgpro_stats. Порядок библиотек важен

```
shared_preload_libraries = 'pg_stat_statements, pg_store_plans, auto_explain'
# местоположение и формат диагностического лога, который ожидает Аналитика. Формат
# оптимален не только для Аналитики
logging_collector = 'on' # коллектор рекомендуется использовать не только для
# Аналитики
log_destination = 'stderr' # значение по умолчанию
log_filename = 'postgresql-%F.log' # создавать раз в сутки, пример: postgresql-2025-12-
# 01.log
log_line_prefix = '%m [%p:%v] [%d] %r %a' # больше деталей у сообщений
lc_messages = 'en_US.UTF-8' # чтобы не зависеть от других параметров локализации
listen_addresses = '*' # по умолчанию экземпляр прослушивает только localhost, нужно
# чтобы позволял подсоединяться по сети
deadlock_timeout = '1s' # значение по умолчанию, можно установить в то значение,
# которое хотите отслеживать
log_lock_waits = 'on' # логирует события ожидания, больше чем deadlock_timeout
log_min_duration_statement = '10s' # логировать команды, которые работают дольше
# установленного этим параметром значения
pg_store_plans.min_duration = '10s'
auto_explain.log_min_duration = '10s'
auto_explain.log_analyze = 'on'
auto_explain.log_buffers = 'on'
auto_explain.log_nested_statements = 'on' # если хочется отслеживать запросы в
# подпрограммах (функциях, процедурах)
track_io_timing = 'on' # если хочется видеть время i/o в планах
```

Конфигурирование доступа к экземпляру для Аналитики

- Соединение к 127.0.0.1 должно быть без пароля:

```
psql -h 127.0.0.1 -U postgres -d postgres
```

- Пример строки в pg_hba.conf, которая дает право подсоединяться без пароля:

```
# IPv4 local connections:
host    all            all            127.0.0.1/32      trust
```

- Местоположение файла:

```
postgres=# \dconfig *hba*
      List of configuration parameters
  Parameter  |          Value
-----+-----
 hba_file   | /var/lib/postgresql/tantor-se-17/data/pg_hba.conf
(1 row)
```

Конфигурирование доступа к экземпляру для Аналитики

Нужно проверить, что подсоединение к адресу 127.0.0.1 базы данных postgres, находящейся в кластере PostgreSQL, с которым хочется чтобы работала Аналитика устанавливается без пароля для пользователя postgres:

```
psql -h 127.0.0.1 -U postgres -d postgres
```

Если соединение не устанавливается, можно проверить содержимое файла pg_hba.conf.

Местоположение файла:

```
postgres=# \dconfig *hba*
      List of configuration parameters
  Parameter  |          Value
-----+-----
 hba_file   | /var/lib/postgresql/tantor-se-17/data/pg_hba.conf
(1 row)
```

Пример строки в файле:

```
# IPv4 local connections:
host    all            all            127.0.0.1/32      trust
```

Пример запроса, показывающего содержимое файла:

```
postgres=# select * from pg_hba_file_rules where address='127.0.0.1' and
database='{all}'\gx
-[ RECORD 1 ]-----
rule_number | 3
file_name   | /var/lib/postgresql/tantor-se-17/data/pg_hba.conf
line_number | 120
type        | host
database   | {all}
user_name   | {all}
address    | 127.0.0.1
netmask    | 255.255.255.255
auth_method| trust
options    |
error      |
```

Конфигурирование хоста для Аналитики

- Соединение к 127.0.0.1 должно быть без пароля:

```
psql -h 127.0.0.1
```

- Пример строки в pg_hba.conf, которая дает право подсоединяться без пароля:

```
# IPv4 local connections:
host    all            all            127.0.0.1/32      trust
```

- Местоположение файла:

```
postgres=# \dconfig *hba*
      List of configuration parameters
  Parameter  |          Value
-----+-----+
 hba_file   | /var/lib/postgresql/tantor-se-17/data/pg_hba.conf
(1 row)
```

Конфигурирование хоста для Аналитики

Для того, чтобы Аналитика могла забирать файлы диагностического журнала кластера PostgreSQL нужно создать ключи под root:

```
astra@tantor:~$ su -
```

```
root@tantor:~# ssh-keygen -t rsa
```

Парольную фразу не вводите, нажмите на приглашения

Enter passphrase (empty for no passphrase):

Enter same passphrase again:

клавишу <enter> на клавиатуре.

Скопируйте ключи в директорию Платформы:

```
root@tantor:~# cp /root/.ssh/id_rsa* /opt/tantor/eco/ssh
```

Скопируйте ключи в домашний каталог пользователя postgres **мониторируемого хоста**:

```
root@tantor:~# ssh-copy-id -i /opt/tantor/eco/ssh/id_rsa
```

```
postgres@education.tantorlabs.ru
```

Выполните подключение с хоста Платформы, пароль **не** должен запрашиваться:

```
root@tantor:~# ssh postgres@education.tantorlabs.ru
```

Проверьте, что на хосте у пользователя postgres используется bash, а не другой интерпретатор:

```
postgres@tantor:~$ echo $SHELL
```

```
/bin/bash
```

```
postgres@tantor:~$ exit
```

```
logout
```

```
Connection to education.tantorlabs.ru closed.
```

Разрешения на файл должны быть 0600. Пример команд, устанавливающих разрешения, которые подходят для ssh:

```
chmod 700 /var/lib/postgresql/.ssh
```

```
chmod 600 /var/lib/postgresql/.ssh/authorized_keys
```

```
chown -R postgres:postgres /var/lib/postgresql/.ssh
```

Перезапустите контейнер Аналитики на хосте Платформы, чтобы ключи были забраны из директории /opt/tantor/eco/ssh:

```
root@tantor:~# docker restart pg_monitor_collector
```

Практика 12

- а. Обновите Платформу на более новую версию
- б. Удалите Платформу и установите более новую версию
- в. Установите Агент Платформы и зарегистрируйте экземпляра Tantor Postgres через веб-интерфейс, а реплику зарегистрируйте через командную строку
- г. Обновите Агентов в docker-контейнерах и зарегистрируйте экземпляры кластера Patroni через командную строку
- д. Настройте Расширенную аналитику, добавив ключи ssh в контейнер pg-monitor

Практика 12

- а. Обновите Платформу на более новую версию
- б. Удалите Платформу и установите более новую версию
- в. Установите Агент Платформы и зарегистрируйте экземпляра Tantor Postgres через веб-интерфейс, а реплику зарегистрируйте через командную строку
- г. Обновите Агентов в docker-контейнерах и зарегистрируйте экземпляры кластера Patroni через командную строку
- д. Настройте Расширенную аналитику, добавив ключи ssh в контейнер pg-monitor