



Application Inspector Enterprise Edition версия 6.2.0

Руководство администратора

© Positive Technologies, 2026.

Настоящий документ является собственностью Positive Technologies и защищен законодательством Российской Федерации и международными соглашениями об авторских правах и интеллектуальной собственности.

Копирование документа либо его фрагментов в любой форме, распространение, в том числе в переводе, а также их передача третьим лицам возможны только с письменного разрешения Positive Technologies.

Документ может быть изменен без предварительного уведомления.

Товарные знаки, использованные в тексте, приведены исключительно в информационных целях, исключительные права на них принадлежат соответствующим правообладателям.

Дата редакции документа: 26.08.2026

Содержание

1.	Об этом документе.....	7
1.1.	Условные обозначения.....	7
1.2.	Другие источники информации о PT AI Enterprise Edition.....	8
2.	О PT AI Enterprise Edition.....	9
2.1.	Архитектура PT AI Enterprise Edition.....	11
2.2.	Поддерживаемые технологии.....	11
3.	Аппаратные и программные требования.....	15
4.	Лицензирование.....	18
5.	Установка PT AI Enterprise Edition.....	19
5.1.	Установка модуля PT AI Enterprise Server.....	19
5.2.	Установка модуля PT AI Enterprise Agent.....	22
5.2.1.	Подготовка к установке.....	22
5.2.2.	Установка модуля PT AI Enterprise Agent.....	23
5.2.3.	Настройка модуля PT AI Enterprise Agent.....	25
5.3.	Действия после установки.....	26
5.3.1.	Установка сертификатов для подключения к внешним ресурсам по SSL.....	26
5.3.2.	Использование внешних баз данных.....	27
5.3.3.	Настройка ротации файлов журнала.....	29
6.	Обновление.....	30
6.1.	Обновление PT AI Enterprise Edition.....	32
6.2.	Обновление внутренних сертификатов.....	33
7.	Администрирование.....	35
7.1.	Резервное копирование и восстановление PT AI Enterprise Server.....	35
7.2.	Журналирование данных с помощью утилиты.....	36
7.3.	Настройка аутентификации через SSO.....	37
7.3.1.	Настройка общих параметров единого входа.....	38
7.3.2.	Настройка подключения по протоколу OpenID Connect.....	38
7.3.3.	Настройка подключения по протоколу SAML 2.0.....	39
7.3.4.	Настройка подключения по протоколу LDAP.....	40
7.3.5.	Настройка соответствия пользователей и ролей.....	41
7.3.6.	Особенности настройки соответствия ролей для LDAP-подключения.....	42
7.4.	Настройка правил анализа.....	43
7.4.1.	Создание правила и группы правил.....	44
7.4.2.	Описание синтаксиса DSL.....	45
7.4.3.	Импорт и экспорт правил анализа.....	48
7.5.	Настройка токенов доступа.....	49
7.5.1.	Создание токена доступа.....	50
7.5.2.	Удаление токена доступа.....	51
7.6.	Активация лицензии.....	51
7.7.	Настройка ИИ-триажа.....	52
7.8.	Выгрузка файлов журнала событий.....	52
7.9.	Настройка прокси-сервера.....	53
7.10.	Настройка почтовых уведомлений.....	54

7.10.1.	Настройка отправки почтовых уведомлений	54
7.10.2.	Изменение параметров почтовых уведомлений	55
7.11.	Настройка шаблонов отчетов	55
7.11.1.	Создание шаблона отчета	56
7.11.2.	Копирование и изменение шаблона отчета	57
7.11.3.	Удаление шаблона отчета	57
7.11.4.	Системные отчеты в PT AI Enterprise Edition	58
7.11.5.	Сравнение отчетов в формате JSON и JSON V2	58
7.11.6.	Дополнительные функции в шаблонах отчетов	62
7.12.	Настройка подключения PT AI Enterprise Edition к Jira	62
7.13.	Работа с пользователями и ролями	63
7.13.1.	О ролях пользователей	63
7.13.2.	Управление ролями и полномочиями	65
7.13.2.1.	Создание пользовательской роли	65
7.13.2.2.	Изменение пользовательской роли	66
7.13.2.3.	Удаление пользовательской роли	66
7.13.3.	Добавление роли пользователю	66
7.13.4.	Блокировка пользователя	67
7.14.	Управление агентами и очередью сканирования	67
7.14.1.	Настройка агентов сканирования	68
7.14.2.	Настройка очереди сканирования	69
7.15.	Мониторинг агента сканирования	70
7.16.	Сбор технических метрик с PT AI Enterprise Server	112
7.16.1.	Метрики Traefik	114
7.16.2.	Метрики RabbitMQ	127
7.16.3.	Метрики Casgate	158
7.16.4.	Метрики .NET-сервисов	177
8.	Интеграция с внешними системами	190
8.1.	Интеграция PT AI Enterprise Edition в сборочный процесс	190
8.1.1.	Об интеграции PT AI Enterprise Edition в сборочный процесс	190
8.1.2.	Установка легкого агента AI.Shell	195
8.1.2.1.	Установка AI.Shell из пакета	196
8.1.2.2.	Установка AI.Shell из инсталлятора для Windows	197
8.1.2.3.	Установка AI.Shell из репозитория	198
8.1.2.4.	Подготовка Docker-образа AI.Shell	198
8.1.3.	Конфигурирование AI.Shell и запуск сканирования	199
8.1.3.1.	Параметры запуска сканирования с помощью AI.Shell	200
8.1.3.2.	Коды возврата	203
8.1.4.	Настройка параметров сканирования в конфигурационном файле	204
8.1.5.	Сборка плагинов	209
8.1.6.	Интеграция с Jenkins	210
8.1.6.1.	Настройка плагина	210
8.1.6.2.	Настройка проектной задачи	214
8.1.6.3.	Создание Groovy-кода для запуска плагина	222
8.1.7.	Интеграция с TeamCity	224

8.1.7.1.	Настройка плагина	224
8.1.7.2.	Настройка проектной задачи	226
8.1.8.	Интеграция с другими CI-системами	234
8.1.8.1.	Установка плагина	235
8.1.8.2.	Подготовка команды запуска сканирования	236
8.1.8.3.	Дополнительные возможности плагина	238
8.1.9.	Файл описания отчетов	240
8.2.	Интеграция PT AI Enterprise Edition со средами разработки	245
8.2.1.	Интеграция с IntelliJ IDEA	245
8.2.1.1.	Установка плагина и анализатора кода	246
8.2.1.2.	Подключение плагина к PT AI Enterprise Server	246
8.2.1.3.	Синхронизация проектов	246
8.2.1.4.	Запуск сканирования в PT AI Enterprise Server	247
8.2.1.5.	Отправка исходного кода в PT AI Enterprise Server	248
8.2.1.6.	Синхронизация результатов сканирования	248
8.2.2.	Интеграция с Visual Studio Code	248
8.2.2.1.	Установка плагина и анализатора кода	249
8.2.2.2.	Подключение плагина к PT AI Enterprise Server	249
8.2.2.3.	Синхронизация проектов	250
8.2.2.4.	Запуск сканирования в PT AI Enterprise Server	250
8.2.2.5.	Отправка исходного кода в PT AI Enterprise Server	251
8.2.2.6.	Синхронизация результатов сканирования	251
8.3.	Интеграция с системой отслеживания ошибок Jira	251
9.	Использование REST API	253
9.1.	Аутентификация по REST API	253
9.2.	Запуск сканирования проекта	254
9.3.	Перенос результатов сканирования между серверами PT AI Enterprise Edition	255
9.4.	Интеграция с сервисом уведомлений	256
9.4.1.	Исходящие сообщения от клиентов	257
9.4.2.	Входящие сообщения: проекты	258
9.4.3.	Входящие сообщения: результаты сканирований	262
9.4.4.	Входящие сообщения: аутентификация	266
9.4.5.	Входящие сообщения: полномочия	268
9.4.6.	Входящие сообщения: параметры сервера	269
9.4.7.	Входящие сообщения: планировщик сканирований	270
9.4.8.	Входящие сообщения: отчеты	271
9.4.9.	Входящие сообщения: результат отправки тестового сообщения	272
10.	Решение проблем	273
10.1.	Нехватка оперативной памяти	273
10.2.	Ошибка миграции данных при большом объеме БД PostgreSQL	274
10.3.	Ошибка «wait-for-healthy-container.sh ai-casgate»	277
10.4.	Остановка сканирования из-за повреждения БД ядра Dependency-check	277
10.5.	Сброс пароля пользователя root	278
10.6.	Проблема подключения к веб-интерфейсу при использовании реверс-прокси	280
10.7.	Ошибка при обновлении до версии 5.2.0 вследствие замены внутренних сертификатов	280

10.8.	Не удается запустить инкрементальное сканирование.....	281
10.9.	Проблема обновления токена доступа к GitLab.....	282
10.10.	Проблема доступа к внутреннему репозиторию	283
11.	О технической поддержке.....	285

1. Об этом документе

Руководство администратора содержит справочную информацию и инструкции по развертыванию, настройке и администрированию Positive Technologies Application Inspector Enterprise Edition (далее также — PT AI Enterprise Edition). Руководство не содержит инструкций по использованию основных функций продукта.

Руководство адресовано специалистам, выполняющим установку, первоначальную настройку и администрирование PT AI Enterprise Edition.

Комплект документации PT AI Enterprise Edition включает в себя следующие документы:

- Этот документ.
- Руководство пользователя — содержит подробную информацию о сценариях работы с продуктом, о настройке функций продукта для решения конкретных задач.
- Руководство по интеграции PT AI Enterprise Edition в сборочный процесс — содержит рекомендации по внедрению анализа кода в сборочный процесс различных CI-систем.

В этом разделе

[Условные обозначения \(см. раздел 1.1\)](#)

[Другие источники информации о PT AI Enterprise Edition \(см. раздел 1.2\)](#)

1.1. Условные обозначения

В документе приняты условные обозначения.

Таблица 1. Условные обозначения

Пример	Описание
Внимание! При выключении модуля снижается уровень защищенности сети	Предупреждения. Содержат информацию о действиях или событиях, которые могут иметь нежелательные последствия
Примечание. Вы можете создать дополнительные отчеты	Примечания. Содержат советы, описания важных частных случаев, дополнительную или справочную информацию, которая может быть полезна при работе с продуктом
► Чтобы открыть файл:	Начало инструкции выделено специальным значком
Нажмите OK	Названия элементов интерфейса (например, кнопок, полей, пунктов меню) выделены полужирным шрифтом
Выполните команду <code>Stop-Service</code>	Текст командной строки, примеры кода, прочие данные, которые нужно ввести с клавиатуры, выделены специальным шрифтом. Также выделены специальным шрифтом имена файлов и пути к файлам и папкам

Пример	Описание
Ctrl+Alt+Delete	Комбинация клавиш. Чтобы использовать комбинацию, клавиши нужно нажимать одновременно
<Название программы>	Переменные заключены в угловые скобки

1.2. Другие источники информации о PT AI Enterprise Edition

Вы можете найти дополнительную информацию о PT AI Enterprise Edition на справочном портале help.ptsecurity.com и на [Позитивном портале](#). Через Позитивный портал вы также можете обратиться [в техническую поддержку \(см. раздел 11\)](#).

2. О PT AI Enterprise Edition

PT AI Enterprise Edition — распределенная система, которая позволяет автоматизировать поиск уязвимостей и признаков недокументированных возможностей (далее — НДВ) в рамках жизненного цикла безопасной разработки и аудита информационной безопасности. В основе работы PT AI Enterprise Edition лежат методы, объединяющие достоинства статического и динамического анализа, а также анализа состава программного обеспечения.

PT AI Enterprise Edition может быть встроен в процесс непрерывной интеграции (англ. continuous integration, CI) на агентах сборки TeamCity, Jenkins и других, что позволяет проверять код на наличие уязвимостей в процессе сборки разрабатываемого продукта.

В PT AI Enterprise Edition реализована ролевая модель доступа. Каждый пользователь имеет соответствующий своей роли уровень доступа к данным и взаимодействует с продуктом с учетом специфики роли.

С помощью PT AI Enterprise Edition разработчики могут писать безопасный код, своевременно устраняя в нем уязвимости, а менеджеры безопасности могут контролировать процесс безопасной разработки.

Использование PT AI Enterprise Edition позволяет повысить качество и сократить сроки разработки и тестирования программного обеспечения, снизить трудоемкость поиска уязвимостей, характерную для ручного анализа.

Преимущества PT AI Enterprise Edition

Преимуществами PT AI Enterprise Edition являются:

- ролевая модель доступа;
- высокая эффективность поиска уязвимостей при низком уровне ложных срабатываний;
- отсутствие необходимости в развертывании приложения;
- наглядная демонстрация уязвимостей;
- сокращение времени проверки кода за счет инкрементального сканирования, учитывающего предыдущие результаты;
- исключение выбранных пользователем уязвимостей из результатов сканирования с помощью добавления комментариев в исходный код.

Ключевые возможности PT AI Enterprise Edition

Ключевыми возможностями PT AI Enterprise Edition являются:

- Реализация функциональных требований к инструментам статического анализа кода, изложенных в Методическом документе «Методика выявления уязвимостей и недекларированных возможностей в программном обеспечении» (для всех поддерживаемых языков, кроме C/C++).
- Поддержка контролируемой сборки программного обеспечения для определения межмодульных связей. Контроль над сборкой реализуется в процессе работы ядра путем отбора проектных файлов, загрузки их зависимостей и последующего построения проектной системы, которая подвергается сканированию.
- Анализ кода на ранних стадиях разработки.
- Автоматическая генерация HTTP-запроса (эксплойта). Эксплойт позволяет проверить найденную уязвимость на развернутом приложении.
- Сканирование запущенного веб-приложения методом черного ящика на тестовом стенде. Сканер анализирует динамические скрипты, формы, параметры, заголовки и прочие входные точки, через которые данные попадают внутрь системы и оказывают на нее негативное воздействие.
- Поддержка пользовательских шаблонов поиска и пользовательских правил, предназначенных для выявления конструкций со специфичной бизнес-логикой или с признаками НДВ.
- Сбор статистических данных о результатах сканирования и найденных уязвимостях.
- Создание задач в Atlassian Jira на исправление уязвимостей, найденных при сканировании.

Поддерживаемые языки программирования

Поддерживаются следующие языки программирования:

- Java
- Scala
- C#
- PHP
- JavaScript/TypeScript
- Python
- C/C++
- Go (Golang)
- Ruby
- Kotlin

- Objective-C
- Swift
- SQL
- Solidity
- 1C
- Dart

В этом разделе

[Архитектура PT AI Enterprise Edition \(см. раздел 2.1\)](#)

[Поддерживаемые технологии \(см. раздел 2.2\)](#)

2.1. Архитектура PT AI Enterprise Edition

PT AI Enterprise Edition состоит из трех отдельно устанавливаемых модулей:

- PT AI Enterprise Server. Центральный модуль, в котором размещен веб-интерфейс. Предоставляет другим модулям и веб-интерфейсу доступ к данным в PT AI Enterprise Edition, управляет этими данными. В состав инсталлятора PT AI Enterprise Server входят:
 - группа сервисов, управляющих работой системы;
 - сервис для работы с сервером очереди сообщений RabbitMQ;
 - база данных PostgreSQL;
 - веб-интерфейс, доступный по URL PT AI Enterprise Server.
- PT AI Enterprise Agent. Проверяет исходный код на наличие уязвимостей и передает результаты сканирования модулю PT AI Enterprise Server. Представляет собой консольное приложение.
- Легкий кросс-платформенный агент AI.Shell. Используется для интеграции с CI-системами и в Docker-контейнере. Подробнее см. в Руководстве по интеграции PT AI Enterprise Edition в сборочный процесс.

2.2. Поддерживаемые технологии

Для оценки защищенности приложений PT AI Enterprise Edition использует три метода анализа: динамический (DAST), статический (SAST) и анализ состава программного обеспечения (SCA).

Динамический анализ защищенности приложения

Dynamic Application Security Testing (DAST). Анализ защищенности развернутого приложения без доступа к его исходному коду.

В PT AI Enterprise Edition DAST реализован ядром WebEngine, которое содержит набор специализированных модулей для обнаружения различных веб-уязвимостей. WebEngine может работать в двух режимах:

- Черный ящик — сканирование развернутого приложения без доступа к его коду.
- Автопроверка — сначала выполняется статический анализ исходного кода (SAST), после чего для найденных уязвимостей генерируются эксплойты. Эти эксплойты автоматически выполняются на вашем развернутом приложении, что позволяет подтвердить или опровергнуть реальное существование уязвимостей.

Статический анализ кода

Static Application Security Testing (SAST). Анализ исходных файлов, байт-кода или артефактов сборки без их исполнения (метод белого ящика). Поскольку сканер работает непосредственно с кодом, для проведения анализа не требуется развертывать приложение, что повышает производительность и упрощает интеграцию в процесс разработки.

SAST является основным методом анализа в PT AI Enterprise Edition. Чтобы обеспечить как можно более полную проверку, SAST-сканирование реализовано несколькими модулями, каждый из которых оптимизирован под определенный тип проверки:

- **Статический анализ кода;**
- **Поиск по шаблонам;**
- **Поиск уязвимостей конфигурации;**
- **Поиск вредоносного кода;**
- **Поиск секретов в коде.**

Модуль **Статический анализ кода** включает в себя основное ядро символьного анализа JSA и ядро taint-анализа Taint. Если для конкретного языка программирования реализован провайдер JSA, то Taint не запускается. Исключение составляет JavaScript/TypeScript — для этих языков можно выбрать, какое ядро использовать. Для повышения качества статического анализа предусмотрена возможность расширения ядра JSA пользовательскими правилами на JSA DSL.

С помощью ядра JSA также осуществляется поиск секретов. Модуль **Поиск секретов в коде** сканирует исходный код на наличие конфиденциальных данных, в частности жестко заданных паролей и комментариев, содержащих секретные данные. Его можно активировать только при включенном одном из модулей — **Статический анализ кода** или **Поиск по шаблонам**.

Модуль **Поиск по шаблонам** использует сигнатурный анализ, основанный на выявлении в программном коде последовательностей, заданных с помощью [правил анализа на специализированном языке PM DSL \(см. раздел 7.4\)](#), или пользовательских Semgrep-правил, которые вы можете задействовать, передав `--semgrep-rules <Путь к правилам Semgrep>` в параметрах запуска модуля.

Модуль **Поиск уязвимостей конфигурации** с помощью ядра Config сканирует конфигурационные файлы проекта и проверяет параметры, влияющие на безопасность приложения, которые заданы неверно или не соответствуют требованиям (по компонентам и всему приложению).

Модуль **Поиск вредоносного кода** построен на собственной модели машинного обучения и предназначен для автоматического выявления в исходном коде скрытой вредоносной логики, которая не относится к заявленной функциональности продукта (например, утечка данных, скрытый удаленный доступ, загрузка и выполнение внешнего кода). Он строит граф вызовов, группирует функции в predetermined активности и с помощью ML-модели оценивает их последовательности, подозрительные комбинации отмечаются как вредоносные.

Анализ состава программного обеспечения

Software Composition Analysis (SCA). Анализ зависимостей, направленный на выявление устаревших и уязвимых сторонних компонентов, проверку их соответствия лицензиям и общим требованиям безопасности. В PT AI Enterprise Edition анализ SCA осуществляют следующие модули:

- **Поиск уязвимых компонентов** — формирует перечень сторонних компонентов, являющихся частью приложения, и сопоставляет этот перечень с базой знаний об известных уязвимостях (CVE, NVD и другие источники).
- **Поиск уязвимых компонентов и их использования** — помимо сопоставления со списком уязвимостей, модуль проверяет, действительно ли приложение использует уязвимые функции стороннего компонента. Это снижает количество ложных срабатываний, поскольку наличие уязвимой библиотеки в проекте не всегда означает возможность ее эксплуатации.

Таблица 2. Технологии, используемые в PT AI Enterprise Edition

Метод анализа		Модуль анализа / Ядро сканирования		Поддерживаемые языки	Расширение базы знаний ядра
Черный ящик	DAST	WebEngine		Анализ проводится по URL развернутого веб-приложения	
Белый ящик	SAST	Статический анализ кода	Taint	JavaScript/TypeScript, Objective-C, Swift, C/C++, Kotlin, SQL, Solidity, 1C, Dart	
			JSA JSA.Net	C#	Правила на JSA DSL

Метод анализа		Модуль анализа / Ядро сканирования			Поддерживаемые языки	Расширение базы знаний ядра	
				JSA.Go	Go		
				JSA.Java	Java		
				JSA.JavaScript	JavaScript		
				JSA.TypeScript	TypeScript		
				JSA.Php	PHP		
				JSA.Python	Python		
				JSA.Ruby	Ruby		
				JSA.Scala	Scala		
		Поиск по шаблонам				Java, C#, PHP, JavaScript/TypeScript, Python, Objective-C, Swift, C/C++, Go, Kotlin, PL/SQL, T-SQL, MySQL, Ruby, Solidity	Правила анализа, написанные с использованием регулярных и DSL-выражений
		Поиск уязвимостей конфигурации				Сканирует конфигурационные файлы, от языка не зависит	
	Поиск секретов в коде				JavaScript/TypeScript, C#, Go, Kotlin, Scala		
	Поиск вредоносного кода				JavaScript/TypeScript, Python		
	SCA	Поиск уязвимых компонентов				Сканирует манифесты, от языка не зависит	
		Поиск уязвимых компонентов и их использования				Java, Go, JavaScript/TypeScript, Python	

3. Аппаратные и программные требования

В этом разделе представлены требования к программному и аппаратному обеспечению для установки и использования PT AI Enterprise Edition.

Таблица 3. Аппаратные и программные требования для PT AI Enterprise Server

Параметр	Минимальное требование
Процессор	x86_64-совместимый 8-ядерный процессор с тактовой частотой 2,4 ГГц
ОЗУ	16 ГБ
Дисковое пространство	200 ГБ
Сетевая карта	10 Мбит/с
ОС	Поддерживаются следующие операционные системы: — Astra Linux 1.7.4 и 2.12; — Astra Linux Special Edition 1.8.1 («Воронеж»); — Astra Linux с OpenSCAP 1.7; — «РЕД ОС» 7.3 и 8.0; — Debian 11.2, 12.1 и 13.0; — Ubuntu 22.04 и 24.04; — Red Hat Enterprise Linux (RHEL) 8.10 и 9.6; — Мос.ОС 15.5; — CentOS 8.5 и 9. Примечание. На более старых версиях ОС стабильная работа не гарантируется
Дополнительное ПО	Docker CE версий 20–29.3.0. Рекомендуется использовать версию 27.1.1, которая не содержит известных критических уязвимостей
Доступные порты	80, 443 для взаимодействия с PT AI Enterprise Agent

Если между PT AI Enterprise Server и клиентом установлен прокси-сервер, необходимо на стороне прокси-сервера разрешить обмен данными по протоколу WebSocket.

На компьютере с модулем PT AI Enterprise Server рекомендуется выделить от одного до двух объемов оперативной памяти под swap-раздел.

Для стабильной работы PT AI Enterprise Edition необходимо использовать валидный TLS-сертификат.

Таблица 4. Аппаратные и программные требования для PT AI Enterprise Agent

Параметр	Минимальное требование
Процессор	x86_64-совместимый 8-ядерный процессор с тактовой частотой 2,4 ГГц
ОЗУ	16 ГБ (при нехватке рекомендуется увеличить до 32 ГБ из-за высокой ресурсоемкости анализа кода)
Дисковое пространство	100 ГБ
Сетевая карта	10 Мбит/с
ОС	Поддерживаются следующие операционные системы: — 64-разрядная версия Windows 10 (кроме обновлений KB5046613 и KB5060533) и 11; — 64-разрядная версия Windows Server 19, 22 и 25; — Astra Linux 1.7.4 и 2.12; — Astra Linux Special Edition 1.8.1 («Воронеж»); — Astra Linux с OpenSCAP 1.7; — «РЕД ОС» 7.3 и 8.0; — Debian 11.2, 12.1 и 13.0; — Ubuntu 22.04 и 24.04; — Red Hat Enterprise Linux (RHEL) 8.10 и 9.6; — Мос.ОС 15.5; — CentOS 8.5 и 9. Примечание. На более старых версиях ОС стабильная работа не гарантируется
Дополнительное ПО	Docker CE версий 20–29.3.0. Рекомендуется использовать версию 27.1.1, которая не содержит известных критических уязвимостей

Таблица 5. Аппаратные и программные требования для AI.Shell

Параметр	Минимальное требование
Процессор	x86_64-совместимый 4-ядерный процессор с тактовой частотой не менее 2 ГГц
ОЗУ	512 МБ
Сетевая карта	10 Мбит/с

Параметр	Минимальное требование
ОС	Поддерживаются следующие операционные системы: — 64-разрядная версия Windows 10 (кроме обновлений KB5046613 и KB5060533); — Astra Linux 1.7.4 и 2.12; — Astra Linux Special Edition 1.8.1 («Воронеж»); — Astra Linux с OpenSCAP 1.7; — «РЕД ОС» 7.3 и 8.0; — Debian 11.2, 12.1 и 13.0; — Ubuntu 22.04 и 24.04; — Red Hat Enterprise Linux (RHEL) 8.10 и 9.6; — Мос.ОС 15.5; — CentOS 8.5 и 9. Примечание. На более старых версиях ОС стабильная работа не гарантируется

Для работы с веб-интерфейсом рекомендуется использовать браузер:

- Microsoft Edge (две последние мажорные версии);
- Google Chrome (две последние версии);
- Firefox ESR (до версии 149.0.0);
- Safari (две последние мажорные версии).

4. Лицензирование

Для работы PT AI Enterprise Edition и защиты всех его модулей от нелегального использования требуется лицензия. Лицензия может быть бессрочной или с ограниченным сроком действия. Приобретенная лицензия определяет:

- набор включенных языков программирования;
- количество доступных проектов;
- количеством агентов сканирования.

Если язык приложения не поддерживается вашей лицензией или превышено количество допустимых проектов, в интерфейсе отображается соответствующее сообщение. Для агентов сканирования такое сообщение не предусмотрено, если их количество больше предусмотренного лицензией, новые агенты не подключаются к системе.

Бессрочная лицензия

Такая лицензия ограничена только сроком поддержки, которая включает техническую поддержку специалистов Positive Technologies, обновление версии продукта, а также обновление базы знаний об уязвимостях. После отключения поддержки вы можете продолжать полноценно работать с проектами и сканированиями.

Лицензия с ограниченным сроком действия

Такая лицензии имеют обозначенный срок действия, который отображается в веб-интерфейсе PT AI Enterprise Edition. По истечении срока действия лицензии активное сканирование останавливается, создание новых проектов, настройка существующих проектов и перезапуск завершенных сканирований становятся недоступными, но вы можете работать с результатами сканирований, полученными ранее.

Примечание. В связи с разницей часовых поясов установленное время истечения лицензии может отличаться от фактического времени в вашем регионе. Расхождение по времени составляет не более суток. Чтобы его избежать, сообщите менеджеру Positive Technologies ваш часовой пояс при оформлении лицензии.

Активация лицензии осуществляется администратором в [веб-интерфейсе программы](#) (см. раздел 7.6).

5. Установка PT AI Enterprise Edition

В этом разделе приводятся инструкции по установке компонентов PT AI Enterprise Edition.

В этом разделе

[Установка модуля PT AI Enterprise Server \(см. раздел 5.1\)](#)

[Установка модуля PT AI Enterprise Agent \(см. раздел 5.2\)](#)

[Действия после установки \(см. раздел 5.3\)](#)

5.1. Установка модуля PT AI Enterprise Server

Установка PT AI Enterprise Server производится на Linux.

Внимание! Для работы PT AI Enterprise Server в операционной системе должен быть установлен компонент Docker CE версий 20–29.3.0 (рекомендуется использовать версию 27.1.1, которая не содержит известных критических уязвимостей). Docker CE, установленный из snap-пакета, не поддерживается.

Внимание! При установке PT AI Enterprise Server в Astra Linux необходимо использовать ядро generic, так как ядро с усиленной защитой (hardened) не позволяет запускать непривилегированные контейнеры.

► Чтобы установить PT AI Enterprise Server:

1. Запустите скрипт установки:

```
sudo sh applicationinspector.<Версия сборки PT AI Enterprise Server>.run
```

или без необходимости подтверждения дальнейших операций:

```
sudo sh applicationinspector.<Версия сборки PT AI Enterprise Server>.run -- --noconfirm
```

При запуске скрипта сформируется файл журнала `/var/log/ptai/install/ptai-install-<Дата и время>.log`, который содержит информационные сообщения от системы, предупреждения и ошибки.

Примечание. Веб-интерфейс PT AI Enterprise Server доступен по адресу сетевого узла (hostname), на котором он развернут. Если вместо имени узла требуется использовать его IP-адрес, необходимо настроить параметр `AdvertiseAddr=<IP-адрес PT AI Enterprise Server>` в файле `/etc/ptai/ptai.ini`. Затем нужно выполнить команду `/opt/ptai/latest/bin/ptaictl reconf` и перезапустить PT AI Enterprise Server командой `ptaictl restart`, чтобы изменения вступили в силу.

2. Введите `y`, чтобы принять условия лицензионного соглашения.
3. Введите `y`, чтобы подтвердить установку PT AI Enterprise Server.

Начнется установка PT AI Enterprise Server.

Примечание. По завершении установки вы можете проверить установленные Docker-контейнеры командой `docker ps -a`. Для мониторинга работы Docker-контейнеров рекомендуется использовать команду `docker stats` или утилиту [ctop](#).

PT AI Enterprise Server установлен. Дистрибутив хранится в каталоге `/opt/ptai`.

Утилита `ptaictl`

Для управления работой PT AI Enterprise Server из консоли используется утилита `ptaictl`. Перечень доступных команд утилиты и их описание приведены в таблице ниже.

По умолчанию утилита `ptaictl` запускается в тихом режиме, без вывода на экран информации от сервисных команд (кроме данных об ошибках). Для отображения подробной информации вы можете указать флаг `VERBOSE` с любым непустым значением. Например, `sudo VERBOSE=1 /opt/ptai/latest/bin/ptaictl stop`.

Таблица 6. Список команд для работы с `ptaictl`

Команда	Описание	Флаги
<code>sudo /opt/ptai/latest/bin/ptaictl stop</code>	Остановка работы PT AI Enterprise Server	<code>-h</code> или <code>--help</code> — справочная информация по использованию команды; <code>--noconfirm</code> — выполнять команду без подтверждения; <code>-l</code> или <code>--log</code> — журналировать в файл результат выполнения команды (например: <code>-l /tmp/log.txt</code>)
<code>sudo /opt/ptai/latest/bin/ptaictl start</code>	Возобновление работы PT AI Enterprise Server	<code>-h</code> или <code>--help</code> — справочная информация по использованию команды; <code>--noconfirm</code> — выполнять команду без подтверждения; <code>-l</code> или <code>--log</code> — журналировать в файл результат выполнения команды (например: <code>-l /tmp/log.txt</code>); <code>--noimgext</code> — не импортировать образы Docker-контейнеров
<code>sudo /opt/ptai/latest/bin/ptaictl restart</code>	Перезапуск PT AI Enterprise Server	<code>-h</code> или <code>--help</code> — справочная информация по использованию команды; <code>--noconfirm</code> — выполнять команду без подтверждения; <code>--noimgext</code> — не импортировать образы Docker-контейнеров

Команда	Описание	Флаги
<code>sudo /opt/ptai/latest/bin/ptaictl reconf</code>	Изменение конфигурации PT AI Enterprise Server	-h или --help — справочная информация по использованию команды; --noconfirm — выполнять команду без подтверждения; -c или --config file — абсолютный путь к файлу ptai.ini (по умолчанию: /etc/ptai/ptai.ini)
<code>sudo /opt/ptai/latest/bin/ptaictl uninstall</code>	Удаление PT AI Enterprise Server	-h или --help — справочная информация по использованию команды; -l или --log — журналировать в файл результат выполнения команды (например: -l /tmp/log.txt)
<code>sudo /opt/ptai/latest/bin/ptaictl restore [VERSION]</code>	Восстановление PT AI Enterprise Server до указанной версии	-h или --help — справочная информация по использованию команды; --noconfirm — выполнять команду без подтверждения
<code>sudo /opt/ptai/latest/bin/ptaictl cert</code>	Замена серверного сертификата и ключа	-h или --help — справочная информация по использованию команды; -c или --cert — путь к файлу с сертификатом в формате PEM; -k или --key — путь к файлу с закрытым ключом в формате PEM, ключ не должен быть зашифрован; --norestart — не выполнять перезапуск Docker-контейнеров
<code>sudo /opt/ptai/latest/bin/ptaictl renew_certs [SERVICES]</code>	Обновление сертификатов внутренних компонентов	-h или --help — справочная информация по использованию команды; --norestart — не выполнять перезапуск Docker-контейнеров
<code>sudo /opt/ptai/latest/bin/ptaictl vacuum</code>	Вакуумирование внутренней БД	-h или --help — справочная информация по использованию команды; --noconfirm — выполнять команду без подтверждения; -l или --log — журналировать в файл результат выполнения команды (например: -l /tmp/log.txt)

При изменении объема оперативной памяти, выделенной для виртуальной машины, необходимо выполнить команду `ptaictl reconf` и перезапустить PT AI Enterprise Server для пересчета коэффициентов потребления оперативной памяти внутри Docker-контейнеров. Перезапуск PT AI Enterprise Server осуществляется путем последовательного выполнения команд `ptaictl stop`, `ptaictl start`.

5.2. Установка модуля PT AI Enterprise Agent

Этот раздел содержит инструкции по подготовке к разворачиванию, установке и настройке модуля PT AI Enterprise Agent.

PT AI Enterprise Agent можно установить на Windows и Linux, но агенты для Linux не поддерживают сканирование C#-приложений, разработанных на платформе .NET Framework до версии 4.8.1 включительно. Для сканирования таких приложений необходимо установить PT AI Enterprise Agent для Windows.

В этом разделе

[Подготовка к установке \(см. раздел 5.2.1\)](#)

[Установка модуля PT AI Enterprise Agent \(см. раздел 5.2.2\)](#)

[Настройка модуля PT AI Enterprise Agent \(см. раздел 5.2.3\)](#)

5.2.1. Подготовка к установке

Чтобы обеспечить стабильную работу модуля PT AI Enterprise Agent в ОС Windows и Linux, необходимо выполнить подготовительные действия.

Windows

Для корректной работы PT AI Enterprise Agent в Windows необходимо установить и настроить следующие программы:

- Для сборки проектов C#: Microsoft .NET 4.7.2 Development Pack.
- Для загрузки зависимостей языка PHP: Composer.
- Для сканирования проектов на языке Go с использованием модуля поиска уязвимых компонентов: Go версии 1.12 и выше. Для модуля «Поиск уязвимых компонентов и их использования» отдельная установка Go не требуется.
- Для сканирования проектов на языках C/C++ и Objective-C: пакеты Windows Kits версии не ниже 10.0.17134.0 и Microsoft Visual C++ SDK версии не ниже 14.28.29333.
- Для сканирования Xamarin-проектов: Microsoft .NET Framework 3.5.1.

В параметрах контроля учетных записей (UAC) Windows рекомендуется отключить запрос разрешений на установку ПО.

Чтобы сообщения и ошибки в языковых параметрах Windows отображались правильно, необходимо включить поддержку Unicode текстов в кодировке UTF-8 или использовать английский язык в качестве языка ОС.

Linux

Для корректного сканирования проектов на языке PHP необходимо в Docker-контейнер с PT AI Enterprise Agent установить Composer, последовательно выполнив следующие команды:

```
docker exec -u root -it ptai-agent bash
apt update
apt install --assume-yes composer
```

Устанавливаемая версия Composer должна совпадать с той, которая используется при разработке и сборке сканируемого приложения.

5.2.2. Установка модуля PT AI Enterprise Agent

Установка PT AI Enterprise Agent возможна как на ОС Linux, так и на Windows.

Установка на Linux

Внимание! Для работы PT AI Enterprise Agent в ОС Linux должен быть установлен компонент Docker CE версий 20–29.3.0 (рекомендуется использовать версию 27.1.1, которая не содержит известных критических уязвимостей).

Внимание! При установке PT AI Enterprise Agent в Astra Linux необходимо использовать ядро generic, так как ядро с усиленной защитой (hardened) не позволяет запускать непривилегированные контейнеры.

► Чтобы установить PT AI Enterprise Agent на Linux:

1. Запустите скрипт установки:


```
sudo ./infr-agent-installer.<Версия PT AI Enterprise Agent> install
```
2. Введите *y*, чтобы принять условия лицензионного соглашения.
3. Введите *y*, чтобы подтвердить установку PT AI Enterprise Agent.
Начнется установка PT AI Enterprise Agent.
4. Выберите способ настройки подключения модуля PT AI Enterprise Agent к PT AI Enterprise Server:
 - если вы хотите настроить подключение к PT AI Enterprise Server в процессе установки, введите *y*;
 - если вы хотите настроить подключение к PT AI Enterprise Server позднее в конфигурационном файле в каталоге `/etc/ptai-agent.conf`, введите *n*.
5. Если вы ввели *y*, укажите следующие параметры:

- URL PT AI Enterprise Server.
- [Токен доступа \(см. раздел 7.5.1\)](#).
- Уникальное имя PT AI Enterprise Agent.

Примечание. Указанное имя будет отображаться в веб-интерфейсе на странице **Агенты сканирования**. Имя Docker-контейнера останется прежним — ptai-agent.

- Имя узла и IP-адрес PT AI Enterprise Server в виде <Имя узла: IP-адрес> (необязательно).

Примечание. PT AI Enterprise Agent, запущенный в Docker-контейнере, обращается к узлу, на котором развернут PT AI Enterprise Server, по его имени (hostname). Если у вас не настроен DNS-сервер для разрешения имен узлов в IP-адреса, вы можете установить соответствие имени узла и IP-адреса с помощью этого параметра.

PT AI Enterprise Agent установлен. Дистрибутив хранится в каталоге /opt/ptai-agent.

Установка на Windows

- ▶ Чтобы установить PT AI Enterprise Agent на Windows:

1. Запустите скрипт установки, выполнив команду в интерфейсе командной строки Windows или в PowerShell от имени администратора:

```
.\infr-agent-installer.<Версия PT AI Enterprise Agent> install
```

2. Введите у, чтобы принять условия лицензионного соглашения.
3. Введите у, чтобы подтвердить установку PT AI Enterprise Agent.

Начнется установка PT AI Enterprise Agent.

4. Выберите способ настройки подключения модуля PT AI Enterprise Agent к PT AI Enterprise Server:

если вы хотите настроить подключение к PT AI Enterprise Server в процессе установки, введите у;

- если вы хотите настроить подключение к PT AI Enterprise Server позднее в конфигурационном файле ptai-agent.conf, введите n.
5. Если вы ввели у, укажите следующие параметры:
 - URL PT AI Enterprise Server.
 - [Токен доступа \(см. раздел 7.5.1\)](#).
 - Уникальное имя PT AI Enterprise Agent.

Примечание. Указанное имя будет отображаться в веб-интерфейсе на странице **Агенты сканирования**. Имя Docker-контейнера останется прежним — ptai-agent.

- Имя узла и IP-адрес PT AI Enterprise Server в виде <Имя узла: IP-адрес> (необязательно).

Примечание. PT AI Enterprise Agent, запущенный в Docker-контейнере, обращается к узлу, на котором развернут PT AI Enterprise Server, по его имени (hostname). Если у вас не настроен DNS-сервер для разрешения имен узлов в IP-адреса, вы можете установить соответствие имени узла и IP-адреса с помощью этого параметра.

Утилита `systemctl`

Для управления работой PT AI Enterprise Agent в Linux используется утилита `systemctl`. С помощью утилиты вы можете:

- останавливать работу сервиса PT AI Enterprise Agent командой `sudo systemctl stop ptai-agent`;
- запускать остановленный сервис PT AI Enterprise Agent командой `sudo systemctl start ptai-agent`;
- перезапускать сервис PT AI Enterprise Agent командой `sudo systemctl restart ptai-agent`.

5.2.3. Настройка модуля PT AI Enterprise Agent

В процессе установки модуля PT AI Enterprise Agent вы настраиваете его подключение к модулю PT AI Enterprise Server по токenu доступа. Если после установки адрес PT AI Enterprise Server или токен изменились, вы можете настроить модуль PT AI Enterprise Agent, указав соответствующие изменения в конфигурационном файле `ptai-agent.conf`.

В конфигурационном файле также можно задать исключения из сканирования и ограничение памяти для Docker-контейнера PT AI Enterprise Agent. По умолчанию контейнер запускается без лимита ОЗУ, что в больших проектах может привести к срабатыванию OOM Killer или к нарушению работы узла с агентом.

► Чтобы настроить модуль PT AI Enterprise Agent:

1. Откройте файл `ptai-agent.conf`:

- Linux — в каталоге `/etc`;
- Windows — по пути `C:\Program Files\Positive Technologies\Application Inspector Agent\ptai-agent.conf`.

2. Настройте подключение к PT AI Enterprise Server:

```
SERVER_URL="<Адрес PT AI Enterprise Server>"
HOSTS="<Имя узла PT AI Enterprise Server:IP-адрес PT AI Enterprise Server>"
AGENT_TOKEN="<Токен доступа>"
AGENT_NAME="<Уникальное имя PT AI Enterprise Agent>"
LOGGING_LEVEL="<Уровень журналирования, возможные значения: debug, info, warn, error>"
```

3. Настройте ограничение памяти (только для агентов на Linux):

```
RAM_LIMIT_ENABLED=true — включить ограничение памяти
RAM_LIMIT="<Размер ограничения, например, 32g>"
```

4. Настройте исключения из сканирования:

`DISABLE_AIIGNORE=<false (учитывать файл .aignore) или true (не учитывать файл .aignore)>`
`DISABLE_GITIGNORE=<false (учитывать файл.gitignore) или true (не учитывать файл .gitignore)>`

Примечание. Если `DISABLE_GITIGNORE=false`, то в `aiproj.json` параметр `SkipGitIgnoreFiles` определяет, будет ли учитываться файл `.gitignore`.

5. Настройте автоматическое обновление исходного кода до последнего коммита текущей ветки (HEAD) при сканировании проекта из Git:

`IGNORE_UPDATE_GIT=<false (обновлять) или true (не обновлять)>`

6. Настройте ротацию файлов журнала:

`LOGGING_ROTATION_ENABLED="true" — включить ротацию`
`LOGGING_ROTATION_MAX_SIZE="<Размер файла в МБ, после которого происходит ротация>"`
`LOGGING_ROTATION_MAX_BACKUPS="<Количество сохраняемых ротированных файлов, лишние будут удаляться>"`
`LOGGING_ROTATION_MAX_AGE="<Срок хранения ротированных файлов в днях>"`
`LOGGING_ROTATION_COMPRESS="<Включить сжатие ротированных файлов, true или false>"`

7. Настройте максимальное время, которое отводится ядру сканирования JSA.Net на восстановление зависимостей при сканировании проектов на C#:

`NUGET_DEPENDENCIES_RESTORE_TIMEOUT="<Время в секундах>"`

8. Перезапустите PT AI Enterprise Agent:

- в Linux выполните команду:
`sudo systemctl restart ptai-agent`
- в Windows перезапустите службу `ptai-agent`.

5.3. Действия после установки

Раздел содержит инструкции, которые необходимо единоразово выполнить после развертывания PT AI Enterprise Edition, чтобы обеспечить полноценную работу с продуктом.

В этом разделе

[Установка сертификатов для подключения к внешним ресурсам по SSL \(см. раздел 5.3.1\)](#)

[Использование внешних баз данных \(см. раздел 5.3.2\)](#)

[Настройка ротации файлов журнала \(см. раздел 5.3.3\)](#)

5.3.1. Установка сертификатов для подключения к внешним ресурсам по SSL

Если для подключения к внешним ресурсам, таким как SSO, Jira и VCS (Git), у вас используется SSL-протокол, то для корректного взаимодействия этих ресурсов с PT AI Enterprise Edition необходимо загрузить в PT AI Enterprise Server один или несколько их корневых сертификатов.

► Чтобы загрузить корневые сертификаты в PT AI Enterprise Server:

1. На компьютере с установленным PT AI Enterprise Server скопируйте PEM-файлы SSL-сертификатов центра сертификации (CA) в следующие каталоги:

`/opt/ptai/_data/volumes/certs/jira/` — для Jira;

`/opt/ptai/_data/volumes/certs/git/` — для Git.

2. Перезапустите PT AI Enterprise Server:

```
sudo /opt/ptai/latest/bin/ptaictl restart --noimgext
```

После перезапуска SSL-сертификаты из PEM-файлов будут добавлены в доверенные в следующих Docker-контейнерах:

- backend — использует SSL-сертификат для подключения к Jira;
- file-content — использует SSL-сертификаты для подключения к Git.

5.3.2. Использование внешних баз данных

В PT AI Enterprise Server предусмотрены две встроенные базы данных. Вы можете использовать свои БД под управлением СУБД PostgreSQL (поддерживаются версии 14.19, 15.14, 16.10 и 17.6).

Рекомендуется настроить подключение внешних баз к PT AI Enterprise Server сразу после установки продукта (модули PT AI Enterprise Agent подключатся автоматически); в противном случае вам необходимо будет самостоятельно выполнить перенос уже существующих в системе данных во внешние БД.

Настройка подключения внешних БД к PT AI Enterprise Server осуществляется в конфигурационном файле `ptai.ini`. Файл содержит два раздела:

- [Database] — БД, в которой хранятся параметры проектов и сканирований, системные параметры (кроме SSO), а также информация об агентах и очереди сканирования. Для этого раздела необходимо в рамках одной СУБД установить две базы и указать их имена в `MainDatabase` и `ScanSchedulerDatabase`.
- [CasgateDatabase] — БД для хранения учетных записей пользователей, их ролей и параметров SSO-подключений.

Примечание. Для PT AI Enterprise Server и всех модулей PT AI Enterprise Agent должен быть открыт доступ к внешним БД.

Внимание! PT AI Enterprise Edition **не выполняет автоматическое резервное копирование и восстановление внешних баз данных (см. раздел 7.1)**, эти операции необходимо осуществлять вручную на стороне внешних БД.

► Чтобы подключить внешние БД к PT AI Enterprise Server:

1. Откройте конфигурационный файл `ptai.ini` в каталоге `/etc/ptai`.
2. В разделе `[Database]` укажите значения параметров для подключения к серверу с СУБД PostgreSQL:

```
Hosts= <IP-адрес или FQDN сервера с БД>:<Порт>,<IP-адрес или FQDN сервера с БД>:<Порт>
MainDatabase= <Имя БД для хранения параметров проектов и сканирований, а также системных параметров (кроме SSO)>
ScanSchedulerDatabase= <Имя БД для хранения информации об агентах и очереди сканирования>
Username= <Имя пользователя>
Password= <Пароль>
SSLMode= <Параметр для подключения по протоколу SSL, возможные значения: disable, require, allow, prefer, verify-ca, verify-full>
SSLCert= <Абсолютный путь к файлу SSL-сертификата>
SSLKey= <Абсолютный путь к файлу закрытого ключа SSL-сертификата>
SSLRootCert= <Абсолютный путь к PEM-файлу SSL-сертификата центра сертификации, параметр обязателен, если DBSSLMode= verify-ca или verify-full>
Timeout= <Тайм-аут подключения к каждому из серверов с БД, по умолчанию 15 секунд>
TargetSessionAttributes= <Предпочтительный режим подключения к целевому серверу с БД, значение по умолчанию – any (используется при указании единственного сервера), если настроено подключение к нескольким серверам, можно использовать все поддерживаемые режимы работы (например, read-write, read-only, primary и другие)>
```

Примечание. Если пароль для подключения к БД содержит спецсимволы, они должны быть корректно экранированы. Правила экранирования такие же, как в языке Bash.

Примечание. Если для подключения к БД используется SSL-протокол, то файлы сертификатов в PT AI Enterprise Server должны быть доступны для чтения пользователю `application-inspector`.

3. Настройте параметры подключения в разделе `[CasgateDatabase]`. Вместо `MainDatabase` и `ScanSchedulerDatabase` укажите `Name= <Имя БД для хранения учетных записей пользователей, их ролей и параметров SSO-подключений>`.
4. Измените конфигурацию PT AI Enterprise Server:


```
sudo /opt/ptai/latest/bin/ptaictl reconf
```
5. Введите `y`, чтобы подтвердить изменение конфигурации.
6. Перезапустите PT AI Enterprise Server:


```
sudo /opt/ptai/latest/bin/ptaictl restart
```
7. Введите `y`, чтобы подтвердить перезапуск.

После перезапуска PT AI Enterprise Server автоматически подключится к внешним БД.

5.3.3. Настройка ротации файлов журнала

Вы можете настраивать параметры ротации файлов журнала. Настройка выполняется отдельно для каждой службы PT AI Enterprise Server, а также для службы управления агентами сканирования PT AI Enterprise Agent. В случае обновления или переустановки модулей требуется повторная настройка ротации.

- Чтобы настроить ротацию файлов журнала для служб PT AI Enterprise Server:

1. На компьютере с установленным PT AI Enterprise Server получите список Docker-контейнеров:

```
docker ps
```

2. Подключитесь к нужному Docker-контейнеру по его идентификатору:

```
docker exec -it <container_id> /bin/bash
```

3. В файле `/app/nlog.config` измените значения параметров:

`archiveAboveSize`: <Размер в байтах, при превышении которого файлы журнала будут автоматически архивироваться>.

`enableArchiveFileCompression`: <Сжимать архивные файлы журнала в ZIP-файлы. Возможные значения: True, False>.

`archiveEvery`: <Автоматически архивировать файлы журнала по истечении указанного времени. Возможные значения: None, Minute, Hour, Day, Month, Year, Sunday, Monday и т. п>.

`maxArchiveDays`: <Максимальное количество дней хранения архивных файлов журнала, по истечении которых файлы будут удалены>.

Примечание. Подробное описание всех параметров приведено в [документации к NLog](#).

- Чтобы настроить ротацию файлов журнала для службы управления агентами сканирования:

1. На компьютере с установленным PT AI Enterprise Agent подключитесь к Docker-контейнеру `ptai-agent`:

```
docker exec -it pta-agent /bin/bash
```

2. В файле `/app/nlog.scheduler.config` измените значения параметров:

`archiveAboveSize`: <Размер в байтах, при превышении которого файлы журнала будут автоматически архивироваться>.

`enableArchiveFileCompression`: <Сжимать архивные файлы журнала в ZIP-файлы. Возможные значения: True, False>.

`archiveEvery`: <Автоматически архивировать файлы журнала по истечении указанного времени. Возможные значения: None, Minute, Hour, Day, Month, Year, Sunday, Monday и т. п>.

`maxArchiveDays`: <Максимальное количество дней хранения архивных файлов журнала, по истечении которых файлы будут удалены>.

Примечание. Подробное описание всех параметров приведено в [документации к NLog](#).

6. Обновление

В этом разделе приводятся инструкции по обновлению PT AI Enterprise Edition.

Для обновления модулей PT AI Enterprise Server и PT AI Enterprise Agent достаточно установить новую версию. Цепочки обновлений для PT AI Enterprise Server приведены в таблице ниже, а PT AI Enterprise Agent можно обновлять напрямую до нужной версии, без последовательного перехода.

Для корректной интеграции IDE-плагинов и плагина Azure DevOps важно, чтобы их версии соответствовали нужной версии PT AI Enterprise Server, поэтому они включены в таблицу.

Для CI/CD-плагинов обеспечена обратная совместимость, и можно использовать последнюю доступную версию.

По завершении обновления всех модулей необходимо также [обновить плагины ptai-ee-tools](#). Они поддерживают обратную совместимость, поэтому вы можете скачать последнюю версию плагина вне зависимости от установленной у вас версии PT AI Enterprise Server. Информация о поддержке новой версии плагина публикуется в [репозитории](#).

Таблица 7. Совместимость компонентов PT AI Enterprise Edition по версиям

Версия PT AI Enterprise Server	Минимальная версия, начиная с которой поддерживается прямое обновление	Версия PT AI Enterprise Agent, совместимая с PT AI Enterprise Server	Версия AI.Shell, совместимая с PT AI Enterprise Server	Версия плагина для Visual Studio Code	Версия плагина для IntelliJ IDEA	Версия плагина для Azure DevOps
6.2.0	4.11.0	6.1.0	6.1.0	2.9.0	2.9.0	Не поддерживается
6.1.0	4.11.0	6.1.0	6.1.0	2.9.0	2.9.0	Не поддерживается
6.0.0	4.11.0	6.0.0	6.0.0	2.9.0	2.9.0	Не поддерживается
5.4.0	4.11.0	5.4.0	5.4.0	2.8.0	2.8.0	1.3.x
5.3.0	4.11.0	5.3.0	5.3.0	2.8.0	2.8.0	1.3.x
5.2.0	4.11.0	5.2.0	5.2.0	2.7.0	2.7.0	1.3.0
5.1.0	4.11.0	5.1.0	5.1.0	2.6.0	2.6.0	1.3.0

Версия PT AI Enterprise Server	Минимальная версия, начиная с которой поддерживается прямое обновление	Версия PT AI Enterprise Agent, совместимая с PT AI Enterprise Server	Версия AI.Shell, совместимая с PT AI Enterprise Server	Версия плагина для Visual Studio Code	Версия плагина для IntelliJ IDEA	Версия плагина для Azure DevOps
5.0.0	4.11.0	5.0.0	5.0.0	2.6.0	2.6.0	1.3.0
4.11.0	4.8.0 и выше	4.11.0	4.11.0	2.5.0	2.5.0	1.2.1
4.10.0	4.8.0 и выше	4.10.0	4.10.0	2.4.2	2.4.2	1.2.1
4.9.1	4.8.0 и выше	4.9.1	4.9.1	2.4.0	2.4.0	1.2.1
4.9.0	4.8.0 и выше	4.9.0	4.9.0	2.4.0	2.4.0	1.2.1
4.8.1	4.7.1 и выше	4.8.1	4.8.1	2.3.1	2.3.1	1.2.1
4.8.0	4.7.1 и выше	4.8.0	4.8.0	2.3.0	2.3.0	1.2.1
4.7.3	4.7.1 и выше	4.7.2	4.7.2	2.2.2, 2.2.3, 2.2.4	2.2.1, 2.2.2	1.2.1
4.7.2	4.7.1	4.7.2	4.7.2	2.2.2, 2.2.3, 2.2.4	2.2.1, 2.2.2	1.2.1
4.7.1	4.6.0 и выше	4.7.1	4.7.1	2.2.2, 2.2.3, 2.2.4	2.2.1, 2.2.2	1.2.1
4.7.0	4.6.0	4.7.0	4.7.0	2.2.0	2.2.0	1.2.1
4.6.0	4.5.0	4.6.0	4.6.0	2.1.0	2.1.0	1.2.1
4.5.0	4.4.0 и выше	4.5.0	4.5.0	2.0.1	2.0.1	—
4.4.1	4.4.0	4.4.1	4.4.1	2.0.0	2.0.0	—
4.4.0	4.3.1	4.4.0	4.4.0	2.0.0	2.0.0	—
4.3.1	4.2.0 и выше	4.3.0, 4.3.1	4.2.0, 4.2.1, 4.3.0, 4.3.1	—	—	—

В этом разделе

[Обновление PT AI Enterprise Edition \(см. раздел 6.1\)](#)

[Обновление внутренних сертификатов \(см. раздел 6.2\)](#)

6.1. Обновление PT AI Enterprise Edition

Внимание! Обновление PT AI Enterprise Server до версии 5.3.0 и выше, если не подключена внешняя БД, может занять значительное время. Это связано с обновлением версии встроенной СУБД PostgreSQL, в ходе которого будет выполнена полная миграция внутренней базы данных. Чтобы во время миграции с версии ниже 5.3.0 не возникла ошибка нехватки дискового пространства, необходимо обеспечить достаточный объем для резервной копии старой БД и размещения новой БД на том же диске: 20 Гб свободного места или 5-кратный размер БД. Текущий размер БД можно проверить с помощью команды: `du -sh /opt/ptai/_data/volumes/postgres/data`.

Примечание. Если вы выполняете обновление на удаленном сервере по SSH, рекомендуется использовать консольные мультиплексоры (например, `tmux` или `GNU Screen`). Они позволяют запускать процессы на удаленном сервере и не прерывать их работу даже при потере SSH-соединения.

Обновление PT AI Enterprise Server

Примечание. Если имя PT AI Enterprise Server содержит заглавные буквы, то при его обновлении с версии ниже 6.0.0 на версию 6.0.0 и выше изменится отпечаток системы: он будет рассчитываться с учетом написания имени сервера в нижнем регистре. После обновления потребуется пересоздать отпечаток и перевыпустить лицензию.

► Чтобы обновить PT AI Enterprise Server:

1. Запустите установочный файл PT AI Enterprise Server:
`sudo sh applicationinspector.<Версия сборки PT AI Enterprise Server>.run`
2. Выполните инструкцию, начиная с шага 2 (см. раздел 5.1).

Примечание. По завершении обновления рекомендуется выполнить вакуумирование внутренней базы данных, указав флаг `--vacuum`, чтобы освободить место для последующих сканирований. Вакуумирование внешней базы данных необходимо выполнять вручную на ее стороне.

Обновление PT AI Enterprise Agent для Linux

► Чтобы обновить PT AI Enterprise Agent:

1. Запустите установочный файл PT AI Enterprise Agent.
`sudo ./infr-agent-installer.<Версия PT AI Enterprise Agent> install`
2. Введите `y`, чтобы принять условия лицензионного соглашения.
3. Введите `y`, чтобы подтвердить удаление предыдущей версии PT AI Enterprise Agent.
4. Введите `y`, чтобы использовать для настройки модуля существующий конфигурационный файл или введите `n`, чтобы создать новую конфигурацию.
5. Если вы ввели `n`, укажите следующие параметры:

- URL PT AI Enterprise Server.
- [Токен доступа \(см. раздел 7.5.1\)](#).
- Уникальное имя PT AI Enterprise Agent.

Обновление PT AI Enterprise Agent для Windows

► Чтобы обновить PT AI Enterprise Agent:

1. Запустите установочный файл PT AI Enterprise Agent от имени администратора:
`.\infr-agent-installer.<Версия PT AI Enterprise Agent> install`
2. Введите `y`, чтобы принять условия лицензионного соглашения.
3. Введите `y`, чтобы подтвердить удаление предыдущей версии PT AI Enterprise Agent.
4. Введите `y`, чтобы создать новый конфигурационный файл `ptai-agent.conf` и импортировать в него параметры для настройки модуля из существующего конфигурационного файла или введите `n`, чтобы создать пустой конфигурационный файл.
5. Если вы ввели `n`, в консоли будет предложено задать новые значения параметров:
 - URL PT AI Enterprise Server.
 - [Токен доступа \(см. раздел 7.5.1\)](#).
 - Уникальное имя PT AI Enterprise Agent.

Если нужно сохранить текущее значение параметра, нажмите Enter без ввода значения.

6. Введите `y`, если нужно удалить старый конфигурационный файл.

6.2. Обновление внутренних сертификатов

При обновлении PT AI Enterprise Edition автоматически обновляются сертификаты внутренних компонентов и серверный сертификат, если срок их действия истек или истечет в течение трех месяцев. Вы также можете при необходимости обновить эти сертификаты вручную, используя команду `renew_certs` утилиты `ptaictl`. Серверный сертификат (tls) будет обновлен только в том случае, если он не был установлен через команду `sudo /opt/ptai/latest/bin/ptaictl cert`.

Формат вызова команды:

```
sudo /opt/ptai/latest/bin/ptaictl renew_certs [SERVICES] [--norestart] [--help],  
где:
```

- `SERVICES` — список компонентов, для которых требуется обновить сертификаты, разделенных запятыми. Возможные значения: `consul`, `vault`, `encrypt`, `vault_client`, `tls`. Обновление выполняется только для сертификатов, автоматически сгенерированных установщиком PT AI Enterprise Edition. Сертификаты, установленные вручную, игнорируются даже если они явно указаны.
- `--norestart` — отмена автоматического перезапуска PT AI Enterprise Edition после обновления сертификатов. Для применения новых сертификатов перезапуск продукта все равно потребует выполнения позже вручную.
- `--help` — справочная информация по использованию команды.

Пример корректного вызова:

```
sudo /opt/ptai/latest/bin/ptaictl renew_certs consul,vault --norestart
```

7. Администрирование

Раздел содержит инструкции для администрирования PT AI Enterprise Edition после установки.

В этом разделе

[Резервное копирование и восстановление PT AI Enterprise Server \(см. раздел 7.1\)](#)

[Журналирование данных с помощью утилиты \(см. раздел 7.2\)](#)

[Настройка аутентификации через SSO \(см. раздел 7.3\)](#)

[Настройка правил анализа \(см. раздел 7.4\)](#)

[Настройка токенов доступа \(см. раздел 7.5\)](#)

[Активация лицензии \(см. раздел 7.6\)](#)

[Настройка ИИ-триажа \(см. раздел 7.7\)](#)

[Выгрузка файлов журнала событий \(см. раздел 7.8\)](#)

[Настройка прокси-сервера \(см. раздел 7.9\)](#)

[Настройка почтовых уведомлений \(см. раздел 7.10\)](#)

[Настройка шаблонов отчетов \(см. раздел 7.11\)](#)

[Настройка подключения PT AI Enterprise Edition к Jira \(см. раздел 7.12\)](#)

[Работа с пользователями и ролями \(см. раздел 7.13\)](#)

[Управление агентами и очередью сканирования \(см. раздел 7.14\)](#)

[Мониторинг агента сканирования \(см. раздел 7.15\)](#)

[Сбор технических метрик с PT AI Enterprise Server \(см. раздел 7.16\)](#)

7.1. Резервное копирование и восстановление PT AI Enterprise Server

Перед началом обновления PT AI Enterprise Server формируется горячая резервная копия его предыдущей версии, которая содержит данные о состоянии приложения, а также локальные базы данных.

Внимание! Внешние базы данных не резервируются, необходимо самостоятельно делать их резервные копии перед установкой новой версии.

Если при обновлении возникли ошибки, предыдущая версия PT AI Enterprise Server будет автоматически восстановлена из горячей резервной копии. Для экземпляров PT AI Enterprise Server, использующих внешние базы данных, откат к прежней версии требует ручного восстановления внешних БД из их собственных резервных копий.

При успешном обновлении горячая резервная копия будет переименована и станет холодной. В системе могут храниться до трех холодных резервных копий, они используются только для ручного восстановления данных.

► Чтобы восстановить PT AI Enterprise Server из холодной резервной копии:

1. Запустите скрипт `ptaictl`.

2. Выполните команду:

```
sudo /opt/ptai/latest/bin/ptaictl restore
```

Примечание. Краткое описание команды доступно в справке: `sudo /opt/ptai/latest/bin/ptaictl restore --help`.

3. Укажите версию, которую необходимо восстановить.

Примечание. Номер версии можно посмотреть в названии резервной копии: `<Версия>_back.tgz`.

7.2. Журналирование данных с помощью утилиты

Иногда для решения проблем, возникающих в процессе работы с PT AI Enterprise Edition, специалистам технической поддержки не хватает подробной информации о среде функционирования продукта и его сервисах. С помощью утилиты `diagtool`, которая поддерживает работу с PT AI Enterprise Server и PT AI Enterprise Agent для Linux, вы можете собрать эти данные, запаковать их в архив и отправить его в службу поддержки Positive Technologies.

Утилита `diagtool` собирает следующие данные:

- информацию об операционной системе;
- информацию о `docker-cli`, `dockerd`, `containerd`;
- конфигурации `dockerd`, `containerd`;
- информацию о Docker-контейнерах;
- файлы журналов `dockerd`, `containerd`, `linux-kernel`;
- файлы журналов установки;
- информацию о сервисах PT AI Enterprise Edition;
- файлы журналов сервисов PT AI Enterprise Edition;
- конфигурацию интегрированных с PT AI Enterprise Edition систем (Jira, SSO-системы);
- конфигурацию сервисов PT AI Enterprise Edition.

► Чтобы собрать данные с помощью утилиты:

1. На компьютере с установленным PT AI Enterprise Server или PT AI Enterprise Agent для Linux перейдите в режим суперпользователя:

```
sudo su
```

2. Запустите утилиту в PT AI Enterprise Server:

```
/opt/ptai/latest/bin/diagtool
```

или в PT AI Enterprise Agent для Linux:

```
/opt/ptai-agent/_latest/etc/bin/diagtool
```

Примечание. При запуске вы можете использовать ключи, перечисленные в таблице ниже.

3. Введите у, чтобы подтвердить добавление данных в архив.

По завершении сбора данных в указанном каталоге будет создан архив для отправки в службу технической поддержки.

Таблица 8. Ключи, используемые при запуске утилиты

Ключ	Описание
-a или --after <Дата в формате гггг-мм-дд>	Дата, начиная с которой нужно собирать данные
-o или --dir <Каталог для сохранения архива с диагностической информацией>	Путь к каталогу с архивом. По умолчанию — /var/tmp
-d или --days <Количество дней>	Количество дней, включая текущий, за которые нужно собрать данные. По умолчанию — 7
--print	Вывод сгенерированного отчета на экран
--errors	Отображение ошибок в процессе сбора данных
-u или --until <Дата в формате гггг-мм-дд>	Дата, до которой включительно нужно собирать данные. По умолчанию используется текущая дата
--noconfirm	Пропуск шага с подтверждением добавления данных в архив
--help	Вывод справки утилиты на экран

7.3. Настройка аутентификации через SSO

В PT AI Enterprise Edition предусмотрена аутентификация и авторизация пользователей через сторонние SSO-системы по протоколам OpenID Connect, SAML 2.0 и LDAP. Такой способ аутентификации и авторизации позволяет не хранить учетные данные пользователей в PT AI Enterprise Edition, а обращаться за ними в SSO-систему, используемую в вашей компании.

Внимание! SSO-система не входит в комплект поставки PT AI Enterprise Edition. Вам необходимо установить и настроить ее самостоятельно. PT AI Enterprise Edition поддерживает работу с SSO-системами Keycloak, OpenLDAP и Blitz. При этом стабильность работы с Blitz не гарантируется.

В этом разделе

[Настройка общих параметров единого входа \(см. раздел 7.3.1\)](#)

[Настройка подключения по протоколу OpenID Connect \(см. раздел 7.3.2\)](#)

[Настройка подключения по протоколу SAML 2.0 \(см. раздел 7.3.3\)](#)

[Настройка подключения по протоколу LDAP \(см. раздел 7.3.4\)](#)

[Настройка соответствия пользователей и ролей \(см. раздел 7.3.5\)](#)

[Особенности настройки соответствия ролей для LDAP-подключения \(см. раздел 7.3.6\)](#)

7.3.1. Настройка общих параметров единого входа

► Чтобы настроить общие параметры единого входа:

1. В главном меню нажмите .
2. В разделе **Единый вход (SSO)** нажмите **Общие параметры**.
3. Задайте время жизни сессии пользователя.
4. Если вы хотите блокировать неактивных пользователей, установите флажок **Блокировка пользователей** и укажите количество дней с последнего входа, по истечении которых пользователь будет отмечен системой как неактивный и заблокирован.
5. Нажмите **Сохранить**.

7.3.2. Настройка подключения по протоколу OpenID Connect

► Чтобы настроить подключение по протоколу OpenID Connect:

1. В главном меню нажмите .
2. В разделе **Единый вход (SSO)** нажмите **Добавить подключение**.
3. В раскрывающемся списке **Протокол** выберите **OpenID Connect**.
4. Введите название протокола, которое будет отображаться во всплывающей подсказке рядом со значком  при входе в веб-интерфейс PT AI Enterprise Edition.

5. В поле **Адрес центра** введите URL хранилища учетных записей пользователей в SSO-системе.
6. В поле **Код клиента** введите идентификатор PT AI Enterprise Edition в вашей SSO-системе.
7. В поле **Ключ клиента** введите ключ для доступа PT AI Enterprise Edition, сгенерированный на стороне вашей SSO-системы.

Примечание. Для корректной работы протокола OpenID Connect необходимо, чтобы на стороне вашей SSO-системы в параметре `grant_type` было указано значение `client_credentials`.

8. Если для подключения к вашей SSO-системе используется протокол HTTPS, в блоке **Защита соединения** загрузите сертификат удостоверяющего центра в PEM-формате.
9. Нажмите **Проверить подключение**.

Если параметры подключения настроены корректно, отобразится оповещение об успешном подключении к серверу SSO-системы.

10. Нажмите **Далее**.

Параметры для подключения настроены, вы можете перейти [к настройке соответствия пользователей \(см. раздел 7.3.5\)](#).

Примечание. При обновлении PT AI Enterprise Edition с предыдущей версии, если у вас настроено подключение по протоколу OpenID Connect, в Keycloak в поле **Valid redirect URIs** необходимо заменить указанное там значение на `https://<Адрес PT AI Enterprise Edition>/*`.

7.3.3. Настройка подключения по протоколу SAML 2.0

- Чтобы настроить подключение по протоколу SAML 2.0:

1. В главном меню нажмите .
2. В разделе **Единый вход (SSO)** нажмите **Добавить подключение**.
3. В раскрывающемся списке **Протокол** выберите **SAML 2.0**.
4. В поле **Название** введите название протокола, которое будет отображаться во всплывающей подсказке рядом со значком  при входе в веб-интерфейс PT AI Enterprise Edition.
5. Введите идентификатор подключаемого приложения (PT AI Enterprise Edition) в SSO-системе (например, `aie-saml2`).
6. В поле **Издатель** введите URL идентификатора SSO-системы.
7. В раскрывающемся списке **Формат идентификатора пользователя** выберите значение атрибута `Format`, которое определит формат идентификатора имени в ответе.

8. В поле **Точка входа в SSO-систему** введите URL страницы авторизации в SSO-системе, куда вы будете перенаправлены при входе в PT AI Enterprise Edition.
9. Если SSO-система проверяет подписи запросов при обращении к ней, в блоке **Запрос** установите флажок **Подписывать запросы**.
10. Загрузите сертификат и закрытый ключ для подписи запросов клиента к SSO-системе в PEM-формате.
11. Выберите алгоритм, с помощью которого подписан сертификат.
12. Если SSO-система подписывает свои ответы, установите флажок **Проверять подписи ответов**.
13. Нажмите **Далее**.

Параметры для подключения настроены, вы можете перейти [к настройке соответствия пользователей \(см. раздел 7.3.5\)](#).

7.3.4. Настройка подключения по протоколу LDAP

► Чтобы настроить подключение по протоколу LDAP:

1. В главном меню нажмите .
2. В разделе **Единый вход (SSO)** нажмите **Добавить подключение**.
3. В раскрывающемся списке **Протокол** выберите **LDAP**.
4. Введите название вашего LDAP-сервера, которое нужно будет выбрать при входе в веб-интерфейс PT AI Enterprise Edition.

Примечание. Максимальное количество символов в поле — 35.

5. Введите адрес LDAP-сервера и порт для подключения к нему.
6. В поле **База поиска пользователей** введите уникальное имя (DN) записи верхнего уровня из вашего LDAP-каталога, чтобы поиск пользователей осуществлялся по всему каталогу.

Примечание. Максимальное количество символов в поле — 100.

7. В поле **Фильтр поиска пользователей** введите атрибуты, ограничивающие область поиска в LDAP-каталоге.

Примечание. Так как поиск пользователей осуществляется по всему LDAP-каталогу, рекомендуется в полях **База поиска пользователей** и **Фильтр поиска пользователей** указывать как можно более точные значения, чтобы в выборку попадали только те пользователи, которые будут работать с PT AI Enterprise Edition. Также необходимо учитывать, что при внесении изменений в эти поля загруженные ранее пользователи не будут удалены и могут задублироваться.

Примечание. Максимальное количество символов в поле — 200.

8. Если для подключения к LDAP-серверу используется защищенное SSL-соединение, установите соответствующий флажок в блоке **Защита соединения**.

Примечание. Для подключения по SSL обычно используется порт 3269.

9. Загрузите сертификат контроллера домена, к которому осуществляется подключение, в PEM-формате.
10. Если LDAP-сервер требует аутентификации PT AI Enterprise Edition в качестве клиента, установите флажок **Криптографическая аутентификация клиента**, а затем загрузите пользовательский сертификат и закрытый ключ в PEM-формате.
11. Введите учетные данные пользователя, который обладает правами на чтение объектов из LDAP-каталога.

Примечание. Максимальное количество символов для логина — 255.

12. Установите флажок **Автоматическая синхронизация**, чтобы отслеживать актуальную информацию о пользователях.
13. Задайте интервал синхронизации, в соответствии с которым PT AI Enterprise Edition будет обращаться к SSO-системе для обновления информации о пользователях.
14. Нажмите **Далее**.

Параметры для подключения настроены, вы можете перейти [к настройке соответствия пользователей \(см. раздел 7.3.5\)](#).

7.3.5. Настройка соответствия пользователей и ролей

Учетная запись пользователя в PT AI Enterprise Edition имеет системные и пользовательские атрибуты. Системные атрибуты (логин, имя пользователя и эл. почта) нельзя изменить, пользовательские атрибуты вы можете добавлять в зависимости от требований вашей организации (например, указывать отдел и должность сотрудника).

Чтобы атрибуты пользователя из SSO-системы синхронизировались с PT AI Enterprise Edition, необходимо настроить их соответствие. С помощью атрибутов вы также можете назначить пользователю из SSO-системы роль в PT AI Enterprise Edition.

► Чтобы настроить соответствие пользователей и ролей:

1. По завершении настройки протокола подключения к SSO-системе на странице **Новое подключение** заполните поля в блоке **Соответствие пользователей**:
 - Введите атрибут из SSO-системы, который будет использован в качестве логина при входе в PT AI Enterprise Edition.
 - Введите атрибут из SSO-системы, который будет использован в качестве имени пользователя.

- Введите атрибут из SSO-системы для электронной почты пользователя, адрес которой будет использован для дополнительной идентификации пользователя в PT AI Enterprise Edition и при отправке отчетов о сканировании.
2. Настройте параметры в блоке **Соответствие ролей**:
 - Введите атрибут из SSO-системы, который будет использован при назначении роли в PT AI Enterprise Edition.
 - Введите значение атрибута.
 - Выберите роль в PT AI Enterprise Edition, которая будет назначена пользователю в соответствии с атрибутом из SSO-системы.

Примечание. Для LDAP-подключения предусмотрен набор функций, которые вы можете использовать [для упрощения поиска пользователей в каталоге Active Directory \(см. раздел 7.3.6\)](#).

Примечание. При первоначальной настройке системы, когда проектов еще нет, вы можете выбрать только роли с глобальными полномочиями, когда проекты будут созданы, вы сможете назначить роль пользователю в конкретном проекте.

3. Если требуется, добавьте еще одно соответствие ролей в сторонней системе и в PT AI Enterprise Edition по кнопке **Добавить соответствие**.

Примечание. Если соответствий много, вы можете искать и фильтровать их по кнопке **Фильтры**.

4. Нажмите **Сохранить**.

7.3.6. Особенности настройки соответствия ролей для LDAP-подключения

Для LDAP-подключения в PT AI Enterprise Edition предусмотрены функции, обеспечивающие удобство поиска пользователей в каталоге Active Directory при настройке соответствия ролей:

- поиск по маске (*) для значений атрибутов;
- рекурсивный поиск пользователей в родительских и дочерних группах;
- поиск по Distinguished Name (DN) с поддержкой маски.

Поиск по маске

Обеспечивает гибкий поиск — вместо точного значения атрибута (`attribute=value`) вы можете указывать только его часть (`attribute=*part*`).

Например, по запросу `ou=*sales*` будут найдены все пользователи, у которых атрибут OU содержит подстроку `sales`.

Ограничения:

- маска может замедлять выполнение запросов в больших каталогах;
- не рекомендуется использовать маску в начале значения (**value*) без дополнительных условий, так как под это условие попадают все пользователи Active Directory;
- маски не работают для атрибутов с бинарными данными (например, `thumbnailPhoto`).

Рекурсивный поиск

Обеспечивает автоматический поиск во всей иерархии групп.

Например, по запросу `memberOf=*ou=marketing*` будут найдены все пользователи, которые:

- состоят непосредственно в группе `marketing`;
- состоят в любой дочерней группе, где OU содержит `marketing`;
- состоят в любой родительской группе, где OU содержит `marketing`;

Поиск по Distinguished Name

Позволяет учитывать сложную иерархическую структуру каталога с родительскими и дочерними группами, которые вы можете указывать в одном запросе.

Например, по запросу `dn=*ou=departments,dc=company,dc=com*` будут найдены все пользователи, чей DN содержит указанный путь в любом месте, а по запросу `dn=cn=user,ou=group,ou=departments,dc=company,dc=com*` будет найден конкретный пользователь, чей DN содержит указанный путь.

Примечание. Атрибут DN необходимо указывать в нижнем регистре. Допускается использование как полного наименования атрибута (`distinguishedName`), так и короткого варианта (`dn`).

7.4. Настройка правил анализа

Правила анализа, написанные с использованием регулярных и DSL-выражений, позволяют выявлять в коде такие недостатки, как хранение конфиденциальных данных в комментариях, применение слабых алгоритмов шифрования и хеширования, а также использование потенциально небезопасных методов. Созданные в этом разделе правила задействуются при сканировании модулем «Поиск по шаблонам».

В этом разделе

[Создание правила и группы правил \(см. раздел 7.4.1\)](#)

[Описание синтаксиса DSL \(см. раздел 7.4.2\)](#)

[Импорт и экспорт правил анализа \(см. раздел 7.4.3\)](#)

7.4.1. Создание правила и группы правил

► Чтобы создать правило анализа:

1. В главном меню нажмите .
2. В разделе **Правила анализа** → **Правила** нажмите **Создать правило**.
3. Введите название правила.

Примечание. Название правила будет отображаться в результатах сканирования как название найденной уязвимости.

4. Выберите одну или несколько групп, к которым относится правило, или создайте новую группу по кнопке внизу раскрывающегося списка.
5. Выберите тип, к которому относится описываемая в правиле уязвимость, или создайте новый по кнопке **Добавить**.
6. Если требуется, добавьте описание пользовательской уязвимости.

Примечание. При описании правила вы можете использовать язык разметки Markdown.

7. Выберите режим анализа:
 - **DSL** — поиск по шаблону на DSL;
 - **Regex** — поиск по регулярному выражению.
8. Если вы выбрали режим **DSL**, укажите один или несколько языков программирования, для которых будет использоваться правило.

Примечание. Вы можете создать универсальное правило для нескольких языков, однако в конкретном проекте будет выполнен анализ только для основного языка, выбранного в параметрах проекта, и для дополнительных языков: JavaScript, TypeScript, SQL и HTML.

9. Если вы выбрали режим **Regex**, укажите одну или несколько масок, в соответствии с которыми будут выбраны файлы для анализа.

Примечание. Поле не является обязательным. Маска позволяет ускорить поиск по регулярным выражениям и снизить количество ложных срабатываний. Несколько масок нужно указывать через точку с запятой.

Примечание. В версии 5.3.0 осуществлен переход на использование glob-шаблонов вместо регулярных выражений. Маски больше не являются рекурсивными по умолчанию. Для поиска файлов в произвольной глубине нужно указывать двойную звездочку, например `**/*.js`, — тогда шаблон будет работать для любой вложенности каталогов.

10. Выберите степень опасности уязвимостей, которые PT AI Enterprise Edition будет искать по этому правилу.
11. Если требуется, укажите любую информацию для связи с вами.

12. Добавьте код правила.

Примечание. Синтаксис DSL приведен в разделе «Описание синтаксиса DSL». Для написания правил с помощью регулярных выражений необходимо использовать [синтаксис RE2](#). Примеры правил анализа можно посмотреть в [GitHub-репозитории](#). Там же хранится [открытая база знаний](#) с правилами, добавленными контрибьюторами из сторонних ИБ-компаний.

Примечание. Поиск по регулярным выражениям не работает с якорем окончания строки (\$) в многострочном режиме. Рекомендуется использовать `\r?$`.

13. Нажмите **Сохранить**.

► Чтобы создать группу РМ-правил:

1. В главном меню нажмите .
2. В разделе **Правила анализа** → **Группы правил** нажмите **Создать группу**.
3. Введите название группы.
4. Выберите правила в списке для включения в группу.
5. Нажмите **Сохранить**.

7.4.2. Описание синтаксиса DSL

С помощью шаблонов на DSL вы можете искать уязвимости в коде более эффективно, чем с помощью регулярных выражений. Регулярные выражения работают с кодом как с текстовой последовательностью и неустойчивы к его форматированию. DSL позволяет создавать шаблоны для поиска блоков кода, которые схожи логически, но содержат разные типы, выражения и функции.

Шаблоны на DSL строятся из выражений, которые содержат следующие лексические элементы:

- идентификаторы;
- ключевые слова;
- спецификаторы;
- литералы.

Идентификаторы (ID) представляют собой названия методов, классов, переменных и различных структур. Идентификатор не должен начинаться с цифры и может содержать только буквы латинского алфавита, цифры, знаки подчеркивания (`_`) и доллара (`$`).

Ключевые слова (keywords) — это набор зарезервированных слов, которые имеют специфичные значения в конкретном языке программирования (например, используются для обозначения класса или функции), а также арифметические операторы и знаки препинания.

Спецификаторы представляют собой служебные элементы DSL, которыми вы можете заменить некоторые значения в коде:

- `#` — любое выражение;
- `#*` или `...` — произвольное количество любых аргументов;
- `<[ ]>` — спецификатор, в котором с помощью регулярных выражений можно задать значения литералов. Список используемых литералов представлен в таблице ниже.

Таблица 9. Литералы, используемые в выражениях

Тип	Описание	Пример
String	Строка. Должна быть заключена в двойные кавычки. Знак обратной косой черты и двойные кавычки должны быть экранированы обратной косой чертой (\)	"password"
Number	Целое, восьмеричное (начиная с 0) или шестнадцатеричное (начиная с 0X) число	12345, 012, 0X123
NumberRange	Диапазон значений данных типа Number	10..20 — диапазон, в котором нижняя граница включается, а верхняя — нет; .. — бесконечный диапазон (ограничен множеством значений типа Int64); 10.. — диапазон, ограниченный снизу (включающий нижнее значение); ..20 — диапазон, ограниченный сверху (не включающий верхнее значение)
Bool	Логическое значение	true, false
Null	Отсутствие данных	null

Из выражений можно составить два типа шаблонов: поиск присваивания и поиск вызова функции или метода.

Поиск присваивания

Общий вид шаблона:

```
expression = expression,
```

где `expression` — это регулярное выражение `<[regexID]>`, идентификатор или литерал.

Например:

- `hardcoded_password = <[".*"]>` // поиск идентификатора `hardcoded_password`, значение которого содержит любой строковый литерал;
- `<[^pass]> = <[".*"]>` // поиск любого идентификатора, который начинается с последовательности `pass`.

Примечание. Для записи регулярных выражений нужно использовать [синтаксис RE2](#).

Поиск вызова функции или метода

Примечание. Ниже приведено описание шаблонов только для поиска вызова функции или метода, но не для их объявления.

Общий вид шаблона:

`expression(arg1, arg2, argN)`

В качестве аргументов в круглых скобках вы можете указать:

- `"value"` — строковый литерал;
- `value` — переменную или нестроковый литерал;
- `<["value"]>` — множество строковых литералов;
- `<[value]>` — множество переменных или нестроковых литералов;
- `#` — один аргумент с любым названием;
- `#*` или `...` — любое количество аргументов.

Например:

- `dangerous_func (#*)` // вызов функции с любым количеством аргументов;
- `dangerous_func (#, \"fake value\")` // вызов функции с первым любым аргументом и вторым обязательным аргументом;
- `dangerous_func (#, \"fake value\", #*)` // вызов функции с первым любым аргументом, вторым обязательным аргументом и далее любой последовательностью необязательных аргументов.

Если у метода явно указан объект, необходимо учитывать, что этот метод может быть вызван как метод объекта, а не как свободная функция. Задать обращение к методу можно с помощью шаблонов вида:

`expression.ID(arg1, arg2, argN)` или `(expression.)?ID(arg1, arg2, argN)`

Например:

- `(#.)?dangerous_func(...)` // вызов метода с любым количеством аргументов, с указанием необязательного объекта.

7.4.3. Импорт и экспорт правил анализа

Вы можете экспортировать правила анализа, созданные в PT AI Enterprise Edition, а также импортировать правила из внешних источников. Эта функциональность позволяет создавать резервные копии правил, переносить правила с одного сервера PT AI Enterprise Edition на другой, а также наполнять базу знаний в [GitHub-репозитории](#).

Импорт правил анализа

► Чтобы импортировать правила анализа:

1. В главном меню нажмите .
2. В разделе **Правила анализа** нажмите **Импортировать**.
3. Загрузите файлы или папку с файлами.

Примечание. Каждый файл должен содержать одно правило, иметь расширение `.pmr1s.json` и не превышать 2 МБ.

4. Если импортируемая папка с файлами может содержать правила, которые уже есть в системе, установите флажок **Импортировать только новые**.
5. Нажмите **Импортировать**.
Начнется загрузка правил в PT AI Enterprise Edition.
6. Если при импорте возникли конфликты с существующими правилами, в раскрывающемся списке **Выберите действие** укажите способ разрешения каждого из конфликтов:
 - **Создать копию.** Правило добавится в систему под новым именем.
 - **Пропустить.** Конфликт будет разрешен в пользу правила, уже существующего в системе.
 - **Заменить.** Импортируемое правило заменит существующее.

Примечание. Список параметров, различающихся в существующем и импортируемом правилах, приведен в раскрывающемся блоке импортируемого правила.

7. После разрешения всех конфликтов нажмите **Импортировать**.
Правила будут добавлены в систему.

Экспорт правил анализа

► Чтобы экспортировать правила анализа:

1. В главном меню нажмите .
2. В разделе **Правила анализа** выберите одно или несколько правил для экспорта.

3. Нажмите **Экспортировать**.
4. Укажите папку, в которой будет сохранено правило с расширением `.pmr1s.json` или ZIP-архив с несколькими правилами.

Начнется выгрузка правил из PT AI Enterprise Edition.

7.5. Настройка токенов доступа

Модуль PT AI Enterprise Server взаимодействует с PT AI Enterprise Agent, легким агентом сканирования AI.Shell, а также с CI-агентами сборки. Для обеспечения доступа и безопасной передачи данных между PT AI Enterprise Server и подключаемыми приложениями в системе предусмотрены два типа токенов доступа:

- для легкого агента и плагинов CI/CD;
- для агента сканирования.

Токены предназначены для решения разных задач и имеют разные наборы прав.

Таблица 10. Права для токенов доступа

Права	Токен для легкого агента и плагинов CI/CD	Токен для агента сканирования
Создание проекта	+	+
Просмотр проекта	+	+
Изменение параметров проекта	+	+
Удаление проекта	+	—
Управление сканированием проекта	+	+
Управление ветками проекта	+	+
Изменения статуса уязвимости	+	—
Изменения статуса уязвимости без комментария	+	—
Просмотр списка агентов	+	+
Просмотр параметров почтовых уведомлений	+	+

Права	Токен для легкого агента и плагинов CI/CD	Токен для агента сканирования
Просмотр пользовательских правил анализа	+	+
Просмотр параметров журналов событий	—	+
Просмотр параметров отчета	+	+
Просмотр очереди сканирования	+	—
Просмотр лицензии	+	+

В этом разделе

[Создание токена доступа \(см. раздел 7.5.1\)](#)

[Удаление токена доступа \(см. раздел 7.5.2\)](#)

7.5.1. Создание токена доступа

► Чтобы создать токен доступа:

1. В главном меню нажмите .
2. В разделе **Токены доступа** нажмите **Создать токен**.
3. Введите уникальное название токена доступа.
4. Установите флажок:
 - **Для легкого агента и плагинов CI/CD**, чтобы разрешить подключение легкого агента AI.Shell и CI-агентов сборки к PT AI Enterprise Server по этому токenu.
 - **Для агента сканирования**, чтобы разрешить подключение PT AI Enterprise Agent к PT AI Enterprise Server по этому токenu.
5. Укажите дату истечения срока действия токена или установите флажок **Бессрочный**.
6. Нажмите **Создать**.
Новый токен доступа сгенерирован.
7. В открывшемся окне скопируйте токен доступа и используйте его при настройке подключения соответствующего приложения к PT AI Enterprise Server.

7.5.2. Удаление токена доступа

► Чтобы удалить токен доступа:

1. В главном меню нажмите .
2. В разделе **Токены доступа** в списке токенов наведите курсор на строку с нужным токеном.
3. Нажмите .
4. В открывшемся окне подтвердите удаление токена по кнопке **Отозвать**.

7.6. Активация лицензии

Вы можете активировать лицензию на сервере лицензий через веб-интерфейс.

Процесс лицензирования состоит из следующих шагов:

1. Вы создаете отпечаток аппаратных характеристик сервера лицензий (фингерпринт) и отправляете его в службу технической поддержки Positive Technologies.
2. Служба технической поддержки Positive Technologies на основании отпечатка создает файл лицензии (ключ) и присылает его вам.
3. Вы добавляете файл лицензии на сервер лицензий и активируете лицензию.

► Чтобы активировать лицензию PT AI Enterprise Edition:

1. В главном меню нажмите .
2. В разделе **Лицензия** нажмите **Создать фингерпринт**.
Файл с отпечатком аппаратных характеристик сервера лицензий загрузится на ваш компьютер.
3. Отправьте загруженный файл в службу технической поддержки Positive Technologies для получения файла лицензии.
4. В блоке **Лицензия** нажмите **Загрузить файл лицензии**.
5. В открывшемся окне выберите файл лицензии, полученный от службы технической поддержки Positive Technologies, и нажмите **Открыть**.

Лицензия будет загружена на сервер лицензий и активирована. В блоке **Лицензия** отобразится информация о лицензии.

7.7. Настройка ИИ-триажа

Сервис ИИ-триажа позволяет упростить и ускорить первичный анализ результатов сканирования за счет использования языковой модели PT Naira. Не применяется для динамического анализа (DAST), анализа состава программного обеспечения (SCA) и при поиске секретов в коде.

ИИ-триаж работает в двух режимах:

- Автоматический — по завершении сканирования система автоматически запускает анализ всех найденных уязвимостей.
- Ручной — пользователь запускает анализ отдельных уязвимостей вручную в карточке уязвимости по кнопке **ИИ-триаж**.

Ручной режим является базовой функциональностью продукта и доступен по умолчанию с некоторыми ограничениями.

Для подключения ИИ-триажа в автоматическом режиме необходимо обратиться в Positive Technologies — это связано с тем, что функциональность находится на стадии закрытого бета-тестирования.

► Чтобы настроить ИИ-триаж:

1. В главном меню нажмите .
2. В разделе **ИИ-триаж** установите флажок **Включить ИИ-триаж**.
3. Убедитесь, что обеспечен сетевой доступ к сервису ИИ-триажа по адресу `update.ptsecurity.ru`.

Примечание. Доступ к сервису можно настроить через [прокси-сервер](#) (см. раздел 7.9), если он подключен на странице **Администрирование**.

4. Дайте согласие на обработку данных.
5. Нажмите **Сохранить**.

7.8. Выгрузка файлов журнала событий

PT AI Enterprise Edition позволяет выгружать файлы журнала с событиями от всех модулей системы. Записи о событиях можно хранить локально или передавать в стороннюю систему.

► Чтобы выгрузить файлы журнала:

1. В главном меню нажмите .
2. В разделе **Журнал событий** в раскрывающемся списке **Мин. уровень журналирования** выберите уровень важности, начиная с которого будут выгружаться файлы журнала.

Примечание. Рекомендуется выбирать уровень **Информация**. Более высокие уровни журналирования, такие как **Отладка** и **Трассировка**, необходимы для диагностики проблем и сбора данных для передачи в службу технической поддержки. Использование этих уровней для постоянной работы может приводить к ухудшению производительности PT AI Enterprise Edition.

3. Нажмите **Сохранить**.
4. По ссылке **Скачать** настройте временной промежуток, за который требуется выгрузить файлы журнала.
5. Нажмите **Скачать**.
Архив с файлами журнала событий будет загружен на ваш компьютер.
6. Если вы хотите хранить записи о событиях в сторонней системе, установите флажок **Передавать события в систему сбора данных**.
Отобразятся параметры настройки сторонней системы.
7. Выберите вариант с системой хранения данных.
8. Введите адрес и порт syslog-сервера или сервера Elasticsearch.
9. В раскрывающемся списке выберите формат сообщений.
10. В раскрывающемся списке **Мин. уровень журналирования в системе сбора данных** выберите уровень важности, начиная с которого события будут отправляться в стороннюю систему.

Примечание. Рекомендуется выбирать уровень **Информация**. Более высокие уровни журналирования, такие как **Отладка** и **Трассировка**, необходимы для диагностики проблем и сбора данных для передачи в службу технической поддержки. Использование этих уровней для постоянной работы может приводить к ухудшению производительности PT AI Enterprise Edition.

Архив с файлами журнала событий будет загружен на выбранный сервер.

11. Нажмите **Сохранить**.

7.9. Настройка прокси-сервера

Некоторые сервисы PT AI Enterprise Edition работают с внешними ресурсами. Если для доступа к внешним ресурсам в вашей сети используется прокси-сервер, вы можете настроить его в веб-интерфейсе.

► Чтобы настроить прокси-сервер:

1. В главном меню нажмите .
2. В разделе **Прокси-сервер** установите флажок **Включить**.
3. В полях **Адрес** и **Порт** укажите IP-адрес и порт прокси-сервера.

4. Установите флажок **Требуется аутентификация**, если для подключения к прокси-серверу необходимо ввести учетные данные.

Отобразятся параметры настройки прокси-сервера.

5. В поле **Имя пользователя** укажите логин для подключения к прокси-серверу.
6. В поле **Пароль** укажите пароль для подключения к прокси-серверу.
7. Нажмите **Сохранить**.

Прокси-сервер настроен. Восстановить предыдущие значения параметров вы можете по кнопке **Сбросить**.

7.10. Настройка почтовых уведомлений

Вы можете настроить отправку отчетов о найденных уязвимостях на электронную почту пользователя с помощью SMTP-сервера.

Настройка отправки почтовых уведомлений состоит из следующих шагов:

- настройка SMTP-сервера;
- настройка шаблона отчета.

В этом разделе

[Настройка отправки почтовых уведомлений \(см. раздел 7.10.1\)](#)

[Изменение параметров почтовых уведомлений \(см. раздел 7.10.2\)](#)

7.10.1. Настройка отправки почтовых уведомлений

- Чтобы настроить отправку почтовых уведомлений:

1. В главном меню нажмите .
2. В разделе **Почтовые уведомления** нажмите **Добавить почтовый профиль**.
3. В блоке **Параметры SMTP-сервера** в поле **Название** введите название почтового профиля, которое будет отображаться в списке почтовых профилей.
4. В поле **Адрес SMTP-сервера** введите FQDN или IP-адрес SMTP-сервера.
5. В поле **Порт** введите порт SMTP-сервера.
6. В поле **Эл. почта отправителя** введите электронный адрес, от имени которого PT AI Enterprise Edition будет посылать уведомления пользователям.
7. Если требуется аутентификация на SMTP-сервере, установите флажок **Требуется аутентификация**.
8. В полях **Логин** и **Пароль** введите учетные данные для аутентификации.

9. Если требуется использовать защищенное SSL-соединение, установите флажок **Включить SSL-шифрование**.
10. Установите флажок **Игнорировать ошибки сертификата**.
PT AI Enterprise Edition будет опрашивать отчет, несмотря на проблемы с SSL-сертификатом.
11. Если требуется, в блоке **Шаблон письма** в поле **Тема** с помощью макросов измените название отчета. Доступные макросы:
 - {{PROJECT}} — название проекта;
 - {{TYPE}} — тип отчета;
 - {{NAME}} — название отчета;
 - {{DT}} — дата и время окончания сканирования;
 - {{LEVEL}} — максимальная опасность найденных уязвимостей.
12. Установите флажок **Сжимать отчет перед отправкой**, если нужно упаковать отчет в ZIP-архив перед отправкой.
Вы можете проверить выполненную настройку, отправив тестовое сообщение на произвольный адрес по кнопке **Отправить тестовое письмо**.
13. Нажмите **Сохранить**.

7.10.2. Изменение параметров почтовых уведомлений

► Чтобы изменить параметры почтовых уведомлений:

1. В главном меню нажмите .
2. В разделе **Почтовые уведомления** в списке почтовых профилей наведите курсор на строку с профилем.
3. Нажмите .

Примечание. Вы также можете удалить почтовый профиль по кнопке .

4. Измените необходимые параметры.
5. Нажмите **Сохранить**.

7.11. Настройка шаблонов отчетов

PT AI Enterprise Edition позволяет выпускать отчеты о найденных уязвимостях по завершении сканирования. Отчеты могут быть пользовательскими и [системными \(см. раздел 7.11.4\)](#). Системные отчеты, в отличие от пользовательских, являются предустановленными, их нельзя изменять и удалять.

В веб-интерфейсе PT AI Enterprise Edition вы можете создавать шаблоны пользовательских отчетов или копировать системные отчеты и на их основе создавать свои шаблоны.

В этом разделе

[Создание шаблона отчета \(см. раздел 7.11.1\)](#)

[Копирование и изменение шаблона отчета \(см. раздел 7.11.2\)](#)

[Удаление шаблона отчета \(см. раздел 7.11.3\)](#)

[Системные отчеты в PT AI Enterprise Edition \(см. раздел 7.11.4\)](#)

[Сравнение отчетов в формате JSON и JSON V2 \(см. раздел 7.11.5\)](#)

[Дополнительные функции в шаблонах отчетов \(см. раздел 7.11.6\)](#)

7.11.1. Создание шаблона отчета

► Чтобы создать шаблон отчета:

1. В главном меню нажмите .
2. В разделе **Шаблоны отчетов** нажмите **Создать шаблон**.
Примечание. Вы также можете загрузить свой шаблон по кнопке **Загрузить шаблон из файла**.
3. В открывшемся окне в поле **Название** введите название отчета, которое будет отображаться в списке шаблонов отчетов.
4. В поле **Имя файла** с помощью макросов задайте имя файла отчета, которое будет отображаться при выпуске отчета. Доступные макросы:
 - {{NAME}} — название отчета;
 - {{PROJECT}} — название проекта;
 - {{TYPE}} — тип отчета;
 - {{DT}} — дата и время окончания сканирования;
 - {{LEVEL}} — максимальная опасность найденных уязвимостей.
5. В поле **Шаблон** введите код шаблона на языке Scriban.

Примечание. Вы можете посмотреть синтаксис языка Scriban по ссылке [Документация по языку шаблонов](#).

Примечание. По ссылке **Модели данных** доступен HTML-документ с описанием входных данных, используемых для формирования отчета: перечень полей, их типы и структура.

6. Если требуется, по кнопке **Предпросмотр** скачайте созданный шаблон перед сохранением в PT AI Enterprise Edition.
7. Нажмите **Сохранить**.

7.11.2. Копирование и изменение шаблона отчета

Вы можете изменять только пользовательские шаблоны отчетов, системные шаблоны изменять нельзя.

- ▶ Чтобы копировать шаблон отчета:

1. В главном меню нажмите .
2. В разделе **Шаблоны отчетов** в списке шаблонов наведите курсор на строку с шаблоном отчета.
3. Нажмите .

Шаблон скопирован и отображается в списке шаблонов как пользовательский. Вы можете изменить его.

- ▶ Чтобы изменить шаблон отчета:

1. В разделе **Шаблоны отчетов** в списке шаблонов наведите курсор на строку с пользовательским шаблоном отчета.
2. Нажмите .
3. Измените необходимые параметры отчета.

Примечание. Вы можете скопировать измененный шаблон по кнопке **Создать копию**.

4. Нажмите **Сохранить**.

7.11.3. Удаление шаблона отчета

Вы можете удалять пользовательские шаблоны, системные шаблоны удалять нельзя.

- ▶ Чтобы удалить шаблон отчета:

1. В главном меню нажмите .
2. В разделе **Шаблоны отчетов** в списке шаблонов наведите курсор на строку с пользовательским шаблоном отчета.
3. Нажмите .
4. В открывшемся окне подтвердите удаление.

7.11.4. Системные отчеты в PT AI Enterprise Edition

В PT AI Enterprise Edition вы можете выпускать следующие типы системных отчетов:

- **Отчет по результатам сканирования.** Отчет в формате HTML, отображающий уязвимости с распределением по опасности и типам, а также уязвимости из списка десяти самых опасных уязвимостей в веб-приложениях по версии OWASP.
- **Отчет в формате SARIF.** Полный JSON-отчет о найденных уязвимостях [в формате SARIF](#).
- **Отчет автопроверки.** HTML-отчет по результатам автоматической проверки.
- **Отчет OWASP Top 10 2021.** Отчет в формате HTML, отображающий уязвимости из списка десяти самых опасных уязвимостей в веб-приложениях по версии OWASP.
- **Отчет PCI DSS 3.2.** Отчет в формате HTML, отображающий оценку выполнения требований стандарта безопасности данных индустрии платежных карт (PCI DSS) в организации.
- **Отчет NIST 800-53 Rev. 4.** HTML-отчет с оценкой выполнения мер обеспечения безопасности и приватности для федеральных информационных систем и организаций.
- **Отчет SANS TOP 25.** Отчет в формате HTML, отображающий уязвимости из списка двадцати пяти самых опасных уязвимостей в приложениях по версии SANS.
- **Отчет OWASP Mobile Top 10 2016.** Отчет в формате HTML, отображающий уязвимости из списка десяти самых опасных уязвимостей в мобильных приложениях по версии OWASP.
- **Отчет ОУД4 (ГОСТ 15408-3).** Отчет в формате HTML, отображающий результаты анализа уязвимостей в соответствии с оценочным уровнем доверия 4 (ОУД4 из [ГОСТ 15408-3](#)). В состав мер доверия по ОУД4 входит компонент доверия AVA_VAN.3, определяющий набор действий по анализу уязвимостей в проекте. PT AI Enterprise Edition позволяет автоматизировать процесс анализа уязвимостей. После проведения анализа по ОУД4 вы можете сформировать соответствующий отчет.
- **Отчет в формате JSON.** Полный отчет о найденных уязвимостях и параметрах проекта в формате JSON.
- **Отчет в формате JSON V2.** Обновленный отчет о найденных уязвимостях и параметрах проекта, представленный в JSON-формате с улучшенной схемой и оптимизированный для внешних интеграций.
- **Отчет для GitLab.** Полный JSON-отчет о найденных уязвимостях, предназначенный для отправки в GitLab, если настроена соответствующая интеграция.

7.11.5. Сравнение отчетов в формате JSON и JSON V2

В версии 6.1.0 появился новый тип системного отчета — JSON V2. В нем оптимизированы структура, состав и наименования полей (подробности см. в таблице ниже). Этот формат рекомендован для интеграции с внешними системами.

Таблица 11. Сравнение отчетов в форматах JSON и JSON V2

Элемент отчета	Отчет в формате JSON	Отчет в формате JSON V2
Стиль написания JSON-ключей	PascalCase	camelCase
Название массива со списком уязвимостей	Items	issues
Отдельный раздел со сводной статистикой о сканировании	—	<pre> "statistic": { "totalFileCount": {{ ScanInfo.TotalFileCount }}, "scannedFileCount": {{ ScanInfo.ScannedFileCount }}, "totalUrlCount": {{ ScanInfo.TotalUrlCount }}, "scannedUrlCount": {{ ScanInfo.ScannedUrlCount }}, "vulnerableFiles": {{ ScanInfo.VulnerableFilesCount }}, "totalVulnerabilityCount": {{ ScanInfo.TotalVulnerabilityCount }}, "acceptedVulnerabilities": {{ ScanInfo.AcceptedVulnerabilitiesCount }}, "discardedVulnerabilities": {{ ScanInfo.DiscardedVulnerabilitiesCount }} }, "suppressedVulnerabilities": {{ ScanInfo.SuppressedVulnerabilitiesCount }}, "newVulnerabilities": {{ ScanInfo.NewVulnerabilitiesCount }}, "previousVulnerabilities": {{ ScanInfo.PreviousVulnerabilitiesCount }}, "policyState": "{{ ScanInfo.PolicyState }}" </pre>

Элемент отчета	Отчет в формате JSON	Отчет в формате JSON V2
		<pre> "nonFilteredVulnerabilitiesCount": {{ ScanInfo.UnfilteredVulnerabilities Count }}}, "nonFilteredAcceptedVulnerabiliti esCount": {{ ScanInfo.UnfilteredAccept edVulnerabilitiesCount }}}, "nonFilteredDiscardedVulnerabiliti esCount": {{ ScanInfo.UnfilteredDiscar dedVulnerabilitiesCount }}}, "nonFilteredSuppressedVulnerabilit iesCount": {{ ScanInfo.UnfilteredSuppr essedVulnerabilitiesCount }}}, "nonFilteredNewVulnerabilitiesCoun t": {{ ScanInfo.UnfilteredNewVulnerabi litiesCount }}}, "nonFilteredPreviousVulnerabilitie sCount": {{ ScanInfo.UnfilteredPreviou sVulnerabilitiesCount }}}, "vulnerableLinks": {{ ScanInfo.Vuln erableLinksCount string.escape }} } </pre>
Координаты уязвимостей	—	<pre> {{- if vuln.StartLine != -1 }} "startLine": {{ vuln.StartLine }}, {{- end }} {{- if vuln.EndLine != -1 }} "endLine": {{ vuln.EndLine }}, {{- end }} {{- if vuln.StartColumn != -1 }} "startColumn": {{ vuln.StartColumn }}, {{- end }} {{- if vuln.EndColumn != -1 }} "endColumn": {{ vuln.EndColumn }}, {{- end }} </pre>

Элемент отчета	Отчет в формате JSON	Отчет в формате JSON V2
Раздел с настройкой представлением параметров сканирования	<pre> "Settings": [{{- for setting in ScanInfo.Settings array.order_by 'Order' }} { "Name": "{{ setting.Name string.escape }}", "Value": "{{ setting.Value string.escape }}", "Order": {{ setting.Order }}, "NeedLineBreak": {{ setting.NeedLineBreak }}, "IsExternalLink": {{ setting.IsExternalLin k }} }{{~ ~}} {{~ if !for.last ~}} {{~ "," ~}} {{~ end ~}} {{- end }}] </pre>	<pre> "settings": {{ ScanInfo.Aiproj }} </pre>

7.11.6. Дополнительные функции в шаблонах отчетов

Для формирования отчетов в PT AI Enterprise Edition используется базовый шаблонизатор. Описание его возможностей доступно по ссылке [Документация по языку шаблонов](#). Однако в работе также применяются дополнительные пользовательские функции, не описанные в документации языка Scriban:

- `json_encode` — кодирует входную строку в формат, соответствующий стандарту JSON.
- `array.order_by` — сортирует элементы массива по указанному полю объектов. Пример: `array.order_by 'DisplayName'`.
- `array.group_by` — группирует элементы массива по значению заданного поля. Пример: `array.group_by 'DisplayName'`.
- `localize` — выполняет локализацию строк на основе внутреннего словаря системы. Добавление пользовательских данных не поддерживается.

7.12. Настройка подключения PT AI Enterprise Edition к Jira

В PT AI Enterprise Edition реализован встроенный плагин для интеграции с трекером задач Jira. Интеграция позволяет создавать в Jira задачи на исправление уязвимостей, найденных при сканировании проекта в PT AI Enterprise Edition.

Вы можете настроить подключение к Jira в веб-интерфейсе.

- Чтобы настроить подключение PT AI Enterprise Edition к Jira:

1. В главном меню нажмите .
2. В разделе **Интеграция с Jira** в блоке **Интеграция с Jira** нажмите **Добавить подключение**.
3. Введите адрес сервера Jira и номер порта для подключения.
4. В раскрывающемся списке **Аутентификация** выберите тип аутентификации в Jira.
5. Если вы выбрали тип аутентификации **Базовая**, укажите следующие параметры:
 - в поле **Имя пользователя** введите учетную запись пользователя в Jira;
 - в поле **Пароль** введите пароль пользователя в Jira.
6. Если вы выбрали тип аутентификации **OAuth 1.0A**, укажите следующие параметры:
 - в поле **Ключ потребителя** введите идентификатор клиента;
 - в поле **Открытый ключ** введите или загрузите по ссылке **Загрузить из файла** открытый RSA-ключ доступа;
 - в поле **Закрытый ключ** введите или загрузите по ссылке **Загрузить из файла** закрытый RSA-ключ доступа.
7. Нажмите **Добавить**.

Подключение PT AI Enterprise Edition к Jira для базового типа аутентификации настроено. Для аутентификации OAuth 1.0A в браузере отобразится страница авторизации в Jira.

8. Укажите учетную запись пользователя и пароль в Jira.
9. Нажмите **Log In**.
10. На открывшейся странице нажмите **Allow**.
11. Вернитесь в веб-интерфейс PT AI Enterprise Edition и нажмите **Продолжить**.

Подключение PT AI Enterprise Edition к Jira для аутентификации OAuth 1.0A настроено.

Вы можете перейти к [настройке проекта в Jira \(см. раздел 8.3\)](#). В этот проект будут поступать задачи на исправление уязвимостей от PT AI Enterprise Edition.

7.13. Работа с пользователями и ролями

Управление пользователями, работающими в PT AI Enterprise Edition, осуществляется на стороне SSO-провайдера. В интерфейсе PT AI Enterprise Edition вы можете назначить пользователям из SSO системные роли или создать свои роли с необходимым набором полномочий.

В общем случае процесс назначения ролей пользователям состоит из следующих этапов:

- Root-пользователь при настройке аутентификации через SSO добавляет администратора.
- Администратор настраивает соответствие атрибутов пользователей из SSO ролям в PT AI Enterprise Edition, таким образом пользователи из SSO автоматически получают одну или несколько системных ролей.
- Администратор при необходимости создает пользовательские роли с глобальными или проектными полномочиями и вручную назначает эти роли пользователям.
- Администратор и менеджер безопасности создают новые проекты. Администратор назначает в них менеджеров проекта вручную или автоматически.
- Менеджер проекта добавляет пользователей в свой проект.

В этом разделе

[О ролях пользователей \(см. раздел 7.13.1\)](#)

[Управление ролями и полномочиями \(см. раздел 7.13.2\)](#)

[Добавление роли пользователю \(см. раздел 7.13.3\)](#)

[Блокировка пользователя \(см. раздел 7.13.4\)](#)

7.13.1. О ролях пользователей

В PT AI Enterprise Edition используется ролевая модель управления доступом. Роль — это набор полномочий, определяющих права доступа к функциям продукта.

По умолчанию в PT AI Enterprise Edition предусмотрены шесть системных ролей с глобальными и проектными полномочиями. Глобальные полномочия наделяют пользователя возможностями по администрированию системы и созданию проектов. Проектные полномочия позволяют выполнять действия только в проекте, на который назначен пользователь. Вы также можете [создавать собственные роли и настраивать для них произвольные комбинации полномочий](#) (см. раздел 7.13.2.1).

Назначение ролей пользователям производится двумя способами:

- Автоматически, за счет настроенного соответствия между ролями в PT AI Enterprise Edition и атрибутами пользователей в SSO-системе, через которую осуществляется аутентификация в PT AI Enterprise Edition.
- Вручную администратором, менеджером проекта или другим пользователем, обладающим полномочиями «Управление пользователями и ролями в проекте» или «Управление пользователями».

Таблица 12. Системные роли и полномочия в PT AI Enterprise Edition

Роль	Полномочия
Администратор	Обладает абсолютными полномочиями в PT AI Enterprise Edition. Имеет доступ ко всем функциям системы, выполняет ее настройку, управляет правами доступа других пользователей. Примечание. Для первоначальной настройки системы и ролей предусмотрен пользователь root. Он обладает полномочиями администратора. Этого пользователя нельзя удалить или изменить его полномочия
Аудитор системы	Просматривает системные параметры, но не обладает полномочиями по их настройке
Аудитор проекта	Просматривает проекты, на которые назначен, но не обладает полномочиями по их настройке. Может управлять статусами найденных уязвимостей и удалять результаты сканирования
Менеджер проекта	Имеет полный доступ к проектам, на которые он назначен
Менеджер безопасности	Создает новые проекты. Просматривает системные параметры, связанные с безопасностью
Разработчик	Обладает полномочиями только в тех проектах, на которые назначен: — просматривает результаты сканирования; — создает задачи в Jira

7.13.2. Управление ролями и полномочиями

PT AI Enterprise Edition поддерживает работу не только с системными ролями, но и позволяет создавать пользовательские роли с глобальными или проектными полномочиями. Пользовательские роли можно изменять и удалять, а системные роли являются неизменяемыми.

В этом разделе

[Создание пользовательской роли \(см. раздел 7.13.2.1\)](#)

[Изменение пользовательской роли \(см. раздел 7.13.2.2\)](#)

[Удаление пользовательской роли \(см. раздел 7.13.2.3\)](#)

7.13.2.1. Создание пользовательской роли

► Чтобы создать пользовательскую роль:

1. В главном меню нажмите .
2. Выберите вкладку **Роли**.
3. На открывшейся странице нажмите кнопку **Добавить**.
В правой части страницы отобразится панель **Новая роль**.
4. Введите название и описание роли.
5. В блоке **Полномочия** установите флажки напротив необходимых полномочий.
6. Нажмите **Создать**.

Если вам нужно создать несколько ролей с одинаковыми или похожими полномочиями, вы можете сделать это через копирование существующих ролей (в том числе системных). В этом случае новая роль получит те же полномочия, что и исходная.

► Чтобы создать копию роли:

1. На странице **Роли** выберите исходную роль в списке.
2. Нажмите **Создать копию**.
В правой части страницы отобразится панель **Новая роль** с тем же набором полномочий, что и у исходной роли.
3. Введите название и описание роли.
4. Нажмите **Создать**.

После создания роли вы можете назначить ее пользователям.

7.13.2.2. Изменение пользовательской роли

► Чтобы изменить пользовательскую роль:

1. В главном меню нажмите .
2. Выберите вкладку **Роли**.
3. На открывшейся странице выберите роль в списке.
4. Нажмите **Изменить**.
5. Внесите изменения и нажмите **Сохранить**.

7.13.2.3. Удаление пользовательской роли

► Чтобы удалить пользовательскую роль:

1. В главном меню нажмите .
2. Выберите вкладку **Роли**.
3. На открывшейся странице выберите роль в списке.
4. Нажмите **Удалить** и подтвердите удаление.

Роль будет удалена у всех пользователей, которым она назначена, доступ пользователей к проектам с этой ролью будет закрыт. Также будут удалены все соответствия с этой ролью в параметрах SSO-подключений.

7.13.3. Добавление роли пользователю

После создания роли вы можете назначить ее пользователю автоматически (через соответствие ролей в PT AI Enterprise Edition и пользователей в SSO-системе) или вручную на странице **Пользователи**.

► Чтобы добавить роль пользователю:

1. В главном меню нажмите .
2. Выберите вкладку **Пользователи**.
3. На открывшейся странице выберите пользователя в списке.
4. Нажмите **Изменить роли**.
5. Назначьте пользователю новую роль по кнопке **Добавить** или измените текущую, выбрав ее в раскрывающемся списке.

Примечание. Роль, назначенную автоматически, можно изменить или удалить только в параметрах соответствия пользователей и ролей (см. раздел 7.3.5).

Примечание. Если пользователю назначены несколько ролей (вручную и автоматически), то итоговый набор полномочий суммируется.

6. Назначьте роль пользователю в отдельных проектах или установите флажок **Все проекты**, чтобы роль распространялась на все проекты, в которых участвует пользователь.

Примечание. Администратор имеет доступ ко всем проектам по умолчанию.

7. Нажмите **Сохранить**.

7.13.4. Блокировка пользователя

Администратор может настроить автоматическую блокировку неактивных пользователей в общих параметрах [единого входа](#) (см. раздел 7.3.1). По истечении заданного срока с момента последнего входа учетная запись пользователя заблокируется, и он не сможет войти в интерфейс PT AI Enterprise Edition.

Помимо автоматической блокировки в PT AI Enterprise Edition предусмотрена возможность ручной блокировки любого пользователя (кроме root-пользователя) в любой момент времени. Эта функциональность также доступна только администратору.

- ▶ Чтобы заблокировать пользователя:

1. В главном меню нажмите .
2. Выберите вкладку **Пользователи**.
3. На открывшейся странице выберите пользователя в списке.
4. Нажмите **Заблокировать**.

Примечание. Нельзя заблокировать самого себя.

- ▶ Чтобы разблокировать пользователя:

1. Нажмите на строку с заблокированным пользователем.
2. Нажмите **Разблокировать** в верхней части страницы.

7.14. Управление агентами и очередью сканирования

Каждый установленный в системе PT AI Enterprise Agent одновременно сканирует только один проект (ветку), остальные проекты помещаются в очередь. Порядок сканирования определяется приоритетом проекта, однако вы можете вручную изменять порядок сканирования, а также удалять проекты из очереди.

Статусы агентов сканирования также можно изменять вручную, например, при необходимости приостановить сканирование для обслуживания агента.

В этом разделе

[Настройка агентов сканирования \(см. раздел 7.14.1\)](#)

[Настройка очереди сканирования \(см. раздел 7.14.2\)](#)

7.14.1. Настройка агентов сканирования

► Чтобы изменить статус агента сканирования:

1. В главном меню нажмите .
2. Выберите вкладку **Агенты сканирования**.

В зависимости от текущего статуса агента в панели инструментов будут доступны различные кнопки.

3. Если требуется, измените состояние агента.

Примечание. При удалении агента из системы он автоматически исключается из списка предпочтительных агентов всех проектов. Если у какого-то проекта удаляемый агент был единственным, на котором могло выполняться сканирование, параметры проекта сбрасываются к значениям по умолчанию: список предпочитаемых агентов становится пустым, а сканирование разрешается на любом доступном агенте.

Таблица 13. Статусы агента сканирования

Состояние агента	Доступные действия	Примечание
Устанавливается	—	Нет доступных действий
Готов к сканированию (агент активен и может брать из очереди новую задачу сканирования)	Поставить на паузу	При нажатии кнопки агент остановится и не будет принимать новые задачи сканирования из очереди
Сканирует (агент сканирует проект в текущий момент)	Поставить на паузу	При нажатии кнопки агент остановится и не будет принимать новые задачи сканирования из очереди, текущее сканирование также будет приостановлено
	Пауза по окончании сканирования	При нажатии кнопки агент завершит текущее сканирование, затем остановится и не будет принимать новые задачи сканирования из очереди

Состояние агента	Доступные действия	Примечание
	Остановить сканирование и убрать из очереди	Кнопка  отображается в строке с агентом. При ее нажатии сканирование на текущем агенте отменится и удалится из очереди
На паузе (агент приостановлен администратором)	Запустить	При нажатии кнопки агент активируется и сможет брать из очереди новую задачу сканирования
	Удалить	При нажатии кнопки агент удалится из списка. Задачи сканирования этому агенту поступать не будут
Пауза по окончании сканирования (агент сканирует проект в текущий момент, но по завершении сканирования остановится и не будет принимать новые задачи сканирования из очереди)	—	Нет доступных действий
Недоступен (нет связи с агентом)	Удалить	При нажатии кнопки агент удалится из списка. Задачи сканирования этому агенту поступать не будут

7.14.2. Настройка очереди сканирования

► Чтобы настроить очередь сканирования:

1. В главном меню нажмите .
2. Выберите вкладку **Агенты сканирования**.
3. В блоке **Очередь сканирования** установите флажок напротив одного или нескольких проектов, которые вы хотите переместить в очереди.
4. В отобразившейся внизу панели нажмите **Переместить** → **В начало** или **Переместить** → **В конец**.

Примечание. Ручное перемещение проекта в очереди имеет более высокий приоритет, чем заданный в параметрах проекта.

5. Если требуется, удалите проект из очереди по кнопке **Убрать из очереди** в панели внизу или по кнопке  напротив названия проекта.

7.15. Мониторинг агента сканирования

PT AI Enterprise Agent может отдавать технические операционные метрики в формате Prometheus.

Базовый набор метрик включает:

- статус агента сканирования;
- статус подключения к PT AI Enterprise Server;
- мониторинг использования диска;
- статус адаптеров;
- статус текущего сканирования.

Все метрики агента имеют общий префикс `ptai_agent_`.

Включение метрик

► Чтобы включить метрики:

1. Откройте файл конфигурации `/etc/ptai-agent.conf` и настройте параметры:


```
METRICS_ENABLED=true
METRICS_ADDR=0.0.0.0:9090
```
2. Сохраните файл.
3. Перезапустите агент сканирования:


```
sudo systemctl restart pta-agent
```
4. Убедитесь, что сервер метрик запущен:


```
curl http://<host>:9090/metrics | head
curl -I http://<host>:9090/readyz # 200 OK, когда агент сканирования готов к работе
```

Таблица 14. Конечные точки

URL	Описание
GET <code>/metrics</code>	Выдает все метрики в формате Prometheus
GET <code>/livez</code>	Возвращает <code>200 OK</code> , пока процесс запущен. Используется в <code>livenessProbe</code> Kubernetes
GET <code>/readyz</code>	Возвращает <code>200 OK</code> , когда агент полностью инициализирован (API-сервер готов принимать запросы); в противном случае — <code>503</code> . Используется в <code>readinessProbe</code> Kubernetes

Сервер метрик запускается до попытки подключения к PT AI Enterprise Server. Поэтому, даже если PT AI Enterprise Server недоступен, запрос `GET /metrics` будет отвечать и покажет реальное состояние агента сканирования.

Таблица 15. Состав метрик

Категория	Метрика	Тип	Метка	Что показывает
Жизненный цикл PT AI Enterprise Agent	ptai_agent_start_time_seconds	Gauge	—	Unix-время запуска агента, Uptime: time() - value
	ptai_agent_ready	Gauge	—	1 — агент готов к работе; 0 — агент еще в процессе инициализации
Статус подключения к PT AI Enterprise Server	ptai_agent_backend_requests_total	Counter	method, code	Общее количество запросов к серверу; code — ok или grpc_<status> (например, grpc_unavailable)
	ptai_agent_backend_last_success_timestamp_seconds	Gauge	method	Unix-время последнего успешного запроса к серверу; при запуске — 0 (не было успешных запросов)
	ptai_agent_heartbeat_failures_total	Counter	—	Количество неудачных heartbeat-сообщений (UpdateStatus). Рост счетчика указывает на деградацию соединения с сервером
Статус токена подключения к PT AI Enterprise Server	ptai_agent_token_valid	Gauge	—	1 — токен действителен; 0 — в данный момент действующего токена нет
	ptai_agent_token_expiry_timestamp_seconds	Gauge	—	Unix-время истечения текущего токена
	ptai_agent_token_refresh_failures_total	Counter	—	Количество неудачных попыток обновления JWT-токена. Рост счетчика указывает на проблемы при обращении к auth-сервису PT AI Enterprise Server
Статус текущего сканирования в PT AI Enterprise Agent	ptai_agent_tasks_received_total	Counter	source, (aie или stdin)	Общее количество пришедших задач
	ptai_agent_tasks_in_flight	Gauge	—	Количество задач, которые обрабатываются в данный момент

Категория	Метрика	Тип	Метка	Что показывает
	ptai_agent_tasks_finished_total	Counter	result (done, failed или aborted)	Общее количество завершённых задач, разбитое по результату. Рост значения метки failed указывает на проблемы в пайплайне или работе адаптеров
	ptai_agent_task_duration_seconds	Histogram	result	Длительность выполнения задачи (от получения до завершения)
	ptai_agent_tasks_active_by_stage	Gauge	stage	Количество активных задач на каждой стадии. Возможные значения stage: Initialize, LoadSources, PrepareVFS, Scanning, SaveLogs, Complete
Статус адаптеров (отдельных процессов: ml-code, jsa, ssa и так далее, запускаемых агентом при каждом сканировании)	ptai_agent_adapter_running	Gauge	adapter	Количество запущенных в данный момент процессов адаптера
	ptai_agent_adapter_runs_total	Counter	adapter, result (ok или error)	Общее количество запусков адаптера
	ptai_agent_adapter_last_failure_timestamp_seconds	Gauge	adapter	Unix-время последнего сбоя конкретного адаптера
Статус диска. Обновляется каждые 30 секунд для трех рабочих каталогов: logs (журналы), source (файлы с исходным кодом проектов), temp (временные файлы)	ptai_agent_workdir_bytes	Gauge	root (logs, source или temp)	Размер рабочего каталога в байтах
	ptai_agent_disk_available_bytes	Gauge	root	Доступное место в файловой системе, где расположен указанный каталог
	ptai_agent_disk_total_bytes	Gauge	root	Общий объем файловой системы, где расположен указанный каталог

Категория	Метрика	Тип	Метка	Что показывает
Состояние узла (load-average/RAM). Данные берутся из /proc/loadavg и /proc/meminfo. Только на Linux	ptai_agent_load_average	Gauge	period (1m, 5m или 15m)	Средняя нагрузка на процессор за указанный интервал. Для оценки нагрузки нужно сравнивать с числом потоков CPU
	ptai_agent_memory_bytes	Gauge	kind (total, available или used)	Объем оперативной памяти узла в байтах. Метка available учитывает reclaimable-cache, used = total-available

Пример конфигурационного файла дашборда для Grafana

Ниже представлен пример конфигурационного файла, который вы можете импортировать в Grafana для отображения получаемых метрик на графиках дашборда.

```
{
  "apiVersion": "dashboard.grafana.app/v2",
  "kind": "Dashboard",
  "metadata": {
    "name": "ptai-agent",
    "namespace": "default",
    "uid": "c62c895e-1fa4-4bdb-8988-dc07a1ee7950",
    "resourceVersion": "1780038921374993",
    "generation": 3,
    "creationTimestamp": "2026-05-29T07:01:19Z",
    "labels": {
      "grafana.app/deprecatedInternalID": "2110119502827520"
    },
    "annotations": {
      "grafana.app/createdBy": "access-policy:service",
      "grafana.app/managedBy": "classic-file-provisioning",
      "grafana.app/managerAllowsEdits": "true",
      "grafana.app/managerId": "Default",
      "grafana.app/sourceChecksum": "bf69f33a31380649b314f4ca3cc380e6",
      "grafana.app/sourcePath": "/etc/grafana/provisioning/dashboards/json/ptai-agent.json",
      "grafana.app/sourceTimestamp": "1780038918000",
      "grafana.app/updatedBy": "access-policy:service",
      "grafana.app/updatedTimestamp": "2026-05-29T07:15:21Z",
      "grafana.app/folder": ""
    }
  },
  "spec": {
```

```

"annotations": [
  {
    "kind": "AnnotationQuery",
    "spec": {
      "query": {
        "kind": "DataQuery",
        "group": "grafana",
        "version": "v0",
        "datasource": {
          "name": "-- Grafana --"
        },
      },
      "spec": {}
    },
    "enable": true,
    "hide": true,
    "iconColor": "rgba(0, 211, 255, 1)",
    "name": "Annotations & Alerts",
    "builtIn": true,
    "legacyOptions": {
      "type": "dashboard"
    }
  }
],
"cursorSync": "Off",
"editable": true,
"elements": {
  "panel-10": {
    "kind": "Panel",
    "spec": {
      "id": 10,
      "title": "Token Refresh Failures",
      "description": "",
      "links": [],
      "data": {
        "kind": "QueryGroup",
        "spec": {
          "queries": [
            {
              "kind": "PanelQuery",
              "spec": {
                "query": {
                  "kind": "DataQuery",
                  "group": "prometheus",
                  "version": "v0",
                  "datasource": {

```

```

        "name": "prometheus"
      },
      "spec": {
        "expr": "ptai_agent_token_refresh_failures_total",
        "legendFormat": "Token Refresh Failures"
      }
    },
    "refId": "A",
    "hidden": false
  }
],
"transformations": [],
"queryOptions": {}
}
},
"vizConfig": {
  "kind": "VizConfig",
  "group": "stat",
  "version": "10.0.0",
  "spec": {
    "options": {
      "colorMode": "value",
      "graphMode": "area",
      "justifyMode": "auto",
      "orientation": "auto",
      "reduceOptions": {
        "calcs": [
          "lastNotNull"
        ],
        "fields": "",
        "values": false
      },
      "textMode": "auto"
    },
    "fieldConfig": {
      "defaults": {
        "thresholds": {
          "mode": "absolute",
          "steps": [
            {
              "value": null,
              "color": "green"
            },
            {
              "value": 1,

```

```

        "color": "red"
      }
    ]
  },
  "color": {
    "mode": "thresholds"
  }
},
"overrides": []
}
}
}
},
"panel-11": {
  "kind": "Panel",
  "spec": {
    "id": 11,
    "title": "Last Backend Success",
    "description": "",
    "links": [],
    "data": {
      "kind": "QueryGroup",
      "spec": {
        "queries": [
          {
            "kind": "PanelQuery",
            "spec": {
              "query": {
                "kind": "DataQuery",
                "group": "prometheus",
                "version": "v0",
                "datasource": {
                  "name": "prometheus"
                },
                "spec": {
                  "expr": "ptai_agent_backend_last_success_timestamp_seconds *
1000",
                  "legendFormat": "{{method}}"
                }
              },
            },
            "refId": "A",
            "hidden": false
          }
        ]
      }
    }
  },

```

```

        "transformations": [],
        "queryOptions": {}
    }
},
"vizConfig": {
    "kind": "VizConfig",
    "group": "stat",
    "version": "10.0.0",
    "spec": {
        "options": {
            "colorMode": "value",
            "graphMode": "none",
            "justifyMode": "auto",
            "orientation": "auto",
            "reduceOptions": {
                "calcs": [
                    "lastNotNull"
                ],
                "fields": "",
                "values": false
            },
            "textMode": "auto"
        },
        "fieldConfig": {
            "defaults": {
                "unit": "dateTimeAsIso",
                "thresholds": {
                    "mode": "absolute",
                    "steps": [
                        {
                            "value": null,
                            "color": "blue"
                        }
                    ]
                },
                "color": {
                    "mode": "thresholds"
                }
            },
            "overrides": []
        }
    }
},
"panel-13": {

```

```

"kind": "Panel",
"spec": {
  "id": 13,
  "title": "Active Tasks by Stage",
  "description": "",
  "links": [],
  "data": {
    "kind": "QueryGroup",
    "spec": {
      "queries": [
        {
          "kind": "PanelQuery",
          "spec": {
            "query": {
              "kind": "DataQuery",
              "group": "prometheus",
              "version": "v0",
              "datasource": {
                "name": "prometheus"
              },
            },
            "spec": {
              "expr": "ptai_agent_tasks_active_by_stage",
              "legendFormat": "{{stage}}"
            }
          },
          "refId": "A",
          "hidden": false
        }
      ],
      "transformations": [],
      "queryOptions": {}
    }
  },
  "vizConfig": {
    "kind": "VizConfig",
    "group": "timeseries",
    "version": "",
    "spec": {
      "options": {
        "legend": {
          "calcs": [],
          "displayMode": "list",
          "placement": "bottom",
          "showLegend": true
        },

```

```

"tooltip": {
  "mode": "single",
  "sort": "none"
},
},
"fieldConfig": {
  "defaults": {
    "thresholds": {
      "mode": "absolute",
      "steps": [
        {
          "value": null,
          "color": "green"
        }
      ]
    },
    "color": {
      "mode": "palette-classic"
    },
    "custom": {
      "axisCenteredZero": false,
      "axisLabel": "",
      "axisPlacement": "auto",
      "barAlignment": 0,
      "drawStyle": "line",
      "fillOpacity": 10,
      "gradientMode": "none",
      "hideFrom": {
        "legend": false,
        "tooltip": false,
        "viz": false
      },
      "lineInterpolation": "linear",
      "lineWidth": 1,
      "pointSize": 5,
      "scaleDistribution": {
        "type": "linear"
      },
      "showPoints": "never",
      "spanNulls": false,
      "stacking": {
        "group": "A",
        "mode": "none"
      },
      "thresholdsStyle": {
        "mode": "off"
      }
    }
  }
}

```

```

        }
      }
    },
    "overrides": []
  }
}
},
"panel-14": {
  "kind": "Panel",
  "spec": {
    "id": 14,
    "title": "Tasks Rate",
    "description": "",
    "links": [],
    "data": {
      "kind": "QueryGroup",
      "spec": {
        "queries": [
          {
            "kind": "PanelQuery",
            "spec": {
              "query": {
                "kind": "DataQuery",
                "group": "prometheus",
                "version": "v0",
                "datasource": {
                  "name": "prometheus"
                },
              },
              "spec": {
                "expr": "rate(ptai_agent_tasks_received_total[5m])",
                "legendFormat": "Received ({{source}})"
              }
            },
            "refId": "A",
            "hidden": false
          }
        ],
      },
    },
    {
      "kind": "PanelQuery",
      "spec": {
        "query": {
          "kind": "DataQuery",
          "group": "prometheus",
          "version": "v0",

```

```

        "datasource": {
            "name": "prometheus"
        },
        "spec": {
            "expr": "rate(ptai_agent_tasks_finished_total[5m])",
            "legendFormat": "Finished ({{result}})"
        }
    },
    "refId": "B",
    "hidden": false
}
],
"transformations": [],
"queryOptions": {}
},
"vizConfig": {
    "kind": "VizConfig",
    "group": "timeseries",
    "version": "",
    "spec": {
        "options": {
            "legend": {
                "calcs": [],
                "displayMode": "list",
                "placement": "bottom",
                "showLegend": true
            },
            "tooltip": {
                "mode": "single",
                "sort": "none"
            }
        },
        "fieldConfig": {
            "defaults": {
                "thresholds": {
                    "mode": "absolute",
                    "steps": [
                        {
                            "value": null,
                            "color": "green"
                        }
                    ]
                }
            },
            "color": {

```

```

        "mode": "palette-classic"
    },
    "custom": {
        "axisCenteredZero": false,
        "axisLabel": "",
        "axisPlacement": "auto",
        "barAlignment": 0,
        "drawStyle": "line",
        "fillOpacity": 10,
        "gradientMode": "none",
        "hideFrom": {
            "legend": false,
            "tooltip": false,
            "viz": false
        },
        "lineInterpolation": "linear",
        "lineWidth": 1,
        "pointSize": 5,
        "scaleDistribution": {
            "type": "linear"
        },
        "showPoints": "never",
        "spanNulls": false,
        "stacking": {
            "group": "A",
            "mode": "none"
        },
        "thresholdsStyle": {
            "mode": "off"
        }
    }
},
"overrides": []
}
}
}
},
"panel-15": {
    "kind": "Panel",
    "spec": {
        "id": 15,
        "title": "Running Adapters",
        "description": "",
        "links": [],
        "data": {

```

```

"kind": "QueryGroup",
"spec": {
  "queries": [
    {
      "kind": "PanelQuery",
      "spec": {
        "query": {
          "kind": "DataQuery",
          "group": "prometheus",
          "version": "v0",
          "datasource": {
            "name": "prometheus"
          },
          "spec": {
            "expr": "ptai_agent_adapter_running",
            "legendFormat": "{{adapter}}"
          }
        },
        "refId": "A",
        "hidden": false
      }
    }
  ],
  "transformations": [],
  "queryOptions": {}
},
"vizConfig": {
  "kind": "VizConfig",
  "group": "timeseries",
  "version": "",
  "spec": {
    "options": {
      "legend": {
        "calcs": [],
        "displayMode": "list",
        "placement": "bottom",
        "showLegend": true
      },
      "tooltip": {
        "mode": "single",
        "sort": "none"
      }
    },
    "fieldConfig": {
      "defaults": {

```

```

    "thresholds": {
      "mode": "absolute",
      "steps": [
        {
          "value": null,
          "color": "green"
        }
      ]
    },
    "color": {
      "mode": "palette-classic"
    },
    "custom": {
      "axisCenteredZero": false,
      "axisLabel": "",
      "axisPlacement": "auto",
      "barAlignment": 0,
      "drawStyle": "line",
      "fillOpacity": 10,
      "gradientMode": "none",
      "hideFrom": {
        "legend": false,
        "tooltip": false,
        "viz": false
      },
      "lineInterpolation": "linear",
      "lineWidth": 1,
      "pointSize": 5,
      "scaleDistribution": {
        "type": "linear"
      },
      "showPoints": "never",
      "spanNulls": false,
      "stacking": {
        "group": "A",
        "mode": "none"
      },
      "thresholdsStyle": {
        "mode": "off"
      }
    }
  },
  "overrides": []
}

```

```

    }
  },
  "panel-17": {
    "kind": "Panel",
    "spec": {
      "id": 17,
      "title": "Workdir Size",
      "description": "",
      "links": [],
      "data": {
        "kind": "QueryGroup",
        "spec": {
          "queries": [
            {
              "kind": "PanelQuery",
              "spec": {
                "query": {
                  "kind": "DataQuery",
                  "group": "prometheus",
                  "version": "v0",
                  "datasource": {
                    "name": "prometheus"
                  },
                },
                "spec": {
                  "expr": "ptai_agent_workdir_bytes",
                  "legendFormat": "{{root}} size"
                }
              },
              "refId": "A",
              "hidden": false
            }
          ],
          "transformations": [],
          "queryOptions": {}
        }
      },
      "vizConfig": {
        "kind": "VizConfig",
        "group": "timeseries",
        "version": "",
        "spec": {
          "options": {
            "legend": {
              "calcs": [],
              "displayMode": "list",

```

```

        "placement": "bottom",
        "showLegend": true
    },
    "tooltip": {
        "mode": "single",
        "sort": "none"
    }
},
"fieldConfig": {
    "defaults": {
        "unit": "bytes",
        "thresholds": {
            "mode": "absolute",
            "steps": [
                {
                    "value": null,
                    "color": "green"
                }
            ]
        },
    },
    "color": {
        "mode": "palette-classic"
    },
    "custom": {
        "axisCenteredZero": false,
        "axisLabel": "",
        "axisPlacement": "auto",
        "barAlignment": 0,
        "drawStyle": "line",
        "fillOpacity": 10,
        "gradientMode": "none",
        "hideFrom": {
            "legend": false,
            "tooltip": false,
            "viz": false
        },
        "lineInterpolation": "linear",
        "lineWidth": 1,
        "pointSize": 5,
        "scaleDistribution": {
            "type": "linear"
        },
        "showPoints": "never",
        "spanNulls": false,
        "stacking": {
            "group": "A",

```

```

        "mode": "none"
      },
      "thresholdsStyle": {
        "mode": "off"
      }
    }
  },
  "overrides": []
}
}
}
},
"panel-18": {
  "kind": "Panel",
  "spec": {
    "id": 18,
    "title": "Disk Space",
    "description": "",
    "links": [],
    "data": {
      "kind": "QueryGroup",
      "spec": {
        "queries": [
          {
            "kind": "PanelQuery",
            "spec": {
              "query": {
                "kind": "DataQuery",
                "group": "prometheus",
                "version": "v0",
                "datasource": {
                  "name": "prometheus"
                },
              },
              "spec": {
                "expr": "ptai_agent_disk_avail_bytes",
                "legendFormat": "{{root}} available"
              }
            },
            "refId": "A",
            "hidden": false
          }
        ]
      }
    },
    {
      "kind": "PanelQuery",
      "spec": {

```

```

        "query": {
            "kind": "DataQuery",
            "group": "prometheus",
            "version": "v0",
            "datasource": {
                "name": "prometheus"
            },
            "spec": {
                "expr": "ptai_agent_disk_total_bytes",
                "legendFormat": "{{root}} total"
            }
        },
        "refId": "B",
        "hidden": false
    }
}
],
"transformations": [],
"queryOptions": {}
}
},
"vizConfig": {
    "kind": "VizConfig",
    "group": "timeseries",
    "version": "",
    "spec": {
        "options": {
            "legend": {
                "calcs": [],
                "displayMode": "list",
                "placement": "bottom",
                "showLegend": true
            },
            "tooltip": {
                "mode": "single",
                "sort": "none"
            }
        }
    },
    "fieldConfig": {
        "defaults": {
            "unit": "bytes",
            "thresholds": {
                "mode": "absolute",
                "steps": [
                    {
                        "value": null,

```

```

        "color": "green"
      }
    ]
  },
  "color": {
    "mode": "palette-classic"
  },
  "custom": {
    "axisCenteredZero": false,
    "axisLabel": "",
    "axisPlacement": "auto",
    "barAlignment": 0,
    "drawStyle": "line",
    "fillOpacity": 10,
    "gradientMode": "none",
    "hideFrom": {
      "legend": false,
      "tooltip": false,
      "viz": false
    },
    "lineInterpolation": "linear",
    "lineWidth": 1,
    "pointSize": 5,
    "scaleDistribution": {
      "type": "linear"
    },
    "showPoints": "never",
    "spanNulls": false,
    "stacking": {
      "group": "A",
      "mode": "none"
    },
    "thresholdsStyle": {
      "mode": "off"
    }
  }
},
"overrides": []
}
}
}
},
"panel-19": {
  "kind": "Panel",
  "spec": {

```

```

    "id": 19,
    "title": "Memory",
    "description": "",
    "links": [],
    "data": {
      "kind": "QueryGroup",
      "spec": {
        "queries": [
          {
            "kind": "PanelQuery",
            "spec": {
              "query": {
                "kind": "DataQuery",
                "group": "prometheus",
                "version": "v0",
                "datasource": {
                  "name": "prometheus"
                },
                "spec": {
                  "expr": "ptai_agent_memory_bytes",
                  "legendFormat": "{{kind}}"
                }
              },
              "refId": "A",
              "hidden": false
            }
          }
        ],
        "transformations": [],
        "queryOptions": {}
      }
    },
    "vizConfig": {
      "kind": "VizConfig",
      "group": "timeseries",
      "version": "",
      "spec": {
        "options": {
          "legend": {
            "calcs": [],
            "displayMode": "list",
            "placement": "bottom",
            "showLegend": true
          },
          "tooltip": {
            "mode": "single",

```

```

        "sort": "none"
    }
},
"fieldConfig": {
    "defaults": {
        "unit": "bytes",
        "thresholds": {
            "mode": "absolute",
            "steps": [
                {
                    "value": null,
                    "color": "green"
                }
            ]
        },
        "color": {
            "mode": "palette-classic"
        },
        "custom": {
            "axisCenteredZero": false,
            "axisLabel": "",
            "axisPlacement": "auto",
            "barAlignment": 0,
            "drawStyle": "line",
            "fillOpacity": 10,
            "gradientMode": "none",
            "hideFrom": {
                "legend": false,
                "tooltip": false,
                "viz": false
            },
            "lineInterpolation": "linear",
            "lineWidth": 1,
            "pointSize": 5,
            "scaleDistribution": {
                "type": "linear"
            },
            "showPoints": "never",
            "spanNulls": false,
            "stacking": {
                "group": "A",
                "mode": "none"
            },
            "thresholdsStyle": {
                "mode": "off"
            }
        }
    }
}

```

```

        }
      },
      "overrides": []
    }
  }
}
},
"panel-2": {
  "kind": "Panel",
  "spec": {
    "id": 2,
    "title": "Agent Ready",
    "description": "",
    "links": [],
    "data": {
      "kind": "QueryGroup",
      "spec": {
        "queries": [
          {
            "kind": "PanelQuery",
            "spec": {
              "query": {
                "kind": "DataQuery",
                "group": "prometheus",
                "version": "v0",
                "datasource": {
                  "name": "prometheus"
                },
                "spec": {
                  "expr": "ptai_agent_ready",
                  "legendFormat": "Ready"
                }
              }
            },
            "refId": "A",
            "hidden": false
          }
        ]
      },
      "transformations": [],
      "queryOptions": {}
    }
  },
  "vizConfig": {
    "kind": "VizConfig",
    "group": "stat",

```

```

"version": "10.0.0",
"spec": {
  "options": {
    "colorMode": "value",
    "graphMode": "none",
    "justifyMode": "auto",
    "orientation": "auto",
    "reduceOptions": {
      "calcs": [
        "lastNotNull"
      ],
      "fields": "",
      "values": false
    },
    "textMode": "auto"
  },
  "fieldConfig": {
    "defaults": {
      "mappings": [
        {
          "type": "value",
          "options": {
            "0": {
              "text": "Not Ready",
              "color": "red",
              "index": 0
            },
            "1": {
              "text": "Ready",
              "color": "green",
              "index": 1
            }
          }
        }
      ]
    },
    "thresholds": {
      "mode": "absolute",
      "steps": [
        {
          "value": null,
          "color": "red"
        },
        {
          "value": 1,
          "color": "green"
        }
      ]
    }
  }
}

```

```

        ]
      },
      "color": {
        "mode": "thresholds"
      }
    },
    "overrides": []
  }
}
},
"panel-3": {
  "kind": "Panel",
  "spec": {
    "id": 3,
    "title": "Token Status",
    "description": "",
    "links": [],
    "data": {
      "kind": "QueryGroup",
      "spec": {
        "queries": [
          {
            "kind": "PanelQuery",
            "spec": {
              "query": {
                "kind": "DataQuery",
                "group": "prometheus",
                "version": "v0",
                "datasource": {
                  "name": "prometheus"
                },
                "spec": {
                  "expr": "ptai_agent_token_valid",
                  "legendFormat": "Token Valid"
                }
              },
              "refId": "A",
              "hidden": false
            }
          }
        ],
        "transformations": [],
        "queryOptions": {}
      }
    }
  }
}

```

```

},
"vizConfig": {
  "kind": "VizConfig",
  "group": "stat",
  "version": "10.0.0",
  "spec": {
    "options": {
      "colorMode": "value",
      "graphMode": "none",
      "justifyMode": "auto",
      "orientation": "auto",
      "reduceOptions": {
        "calcs": [
          "lastNotNull"
        ],
        "fields": "",
        "values": false
      },
      "textMode": "auto"
    },
    "fieldConfig": {
      "defaults": {
        "mappings": [
          {
            "type": "value",
            "options": {
              "0": {
                "text": "Invalid",
                "color": "red",
                "index": 0
              },
              "1": {
                "text": "Valid",
                "color": "green",
                "index": 1
              }
            }
          }
        ]
      },
      "thresholds": {
        "mode": "absolute",
        "steps": [
          {
            "value": null,
            "color": "red"
          }
        ]
      }
    }
  }
}

```

```

        {
            "value": 1,
            "color": "green"
        }
    ]
},
"color": {
    "mode": "thresholds"
}
},
"overrides": []
}
}
}
},
"panel-4": {
    "kind": "Panel",
    "spec": {
        "id": 4,
        "title": "Token Expiry",
        "description": "",
        "links": [],
        "data": {
            "kind": "QueryGroup",
            "spec": {
                "queries": [
                    {
                        "kind": "PanelQuery",
                        "spec": {
                            "query": {
                                "kind": "DataQuery",
                                "group": "prometheus",
                                "version": "v0",
                                "datasource": {
                                    "name": "prometheus"
                                }
                            },
                            "spec": {
                                "expr": "ptai_agent_token_expiry_timestamp_seconds * 1000",
                                "legendFormat": "Token Expires"
                            }
                        },
                        "refId": "A",
                        "hidden": false
                    }
                ]
            }
        }
    }
}

```

```

    ],
    "transformations": [],
    "queryOptions": {}
  }
},
"vizConfig": {
  "kind": "VizConfig",
  "group": "stat",
  "version": "10.0.0",
  "spec": {
    "options": {
      "colorMode": "value",
      "graphMode": "none",
      "justifyMode": "auto",
      "orientation": "auto",
      "reduceOptions": {
        "calcs": [
          "lastNotNull"
        ],
        "fields": "",
        "values": false
      },
      "textMode": "auto"
    },
    "fieldConfig": {
      "defaults": {
        "unit": "dateTimeAsIso",
        "thresholds": {
          "mode": "absolute",
          "steps": [
            {
              "value": null,
              "color": "blue"
            }
          ]
        },
        "color": {
          "mode": "thresholds"
        }
      },
      "overrides": []
    }
  }
}
}
}
},

```

```

"panel-5": {
  "kind": "Panel",
  "spec": {
    "id": 5,
    "title": "Memory Usage",
    "description": "",
    "links": [],
    "data": {
      "kind": "QueryGroup",
      "spec": {
        "queries": [
          {
            "kind": "PanelQuery",
            "spec": {
              "query": {
                "kind": "DataQuery",
                "group": "prometheus",
                "version": "v0",
                "datasource": {
                  "name": "prometheus"
                },
                "spec": {
                  "expr": "ptai_agent_memory_bytes{kind=\"used\"} / on() ptai_a
gent_memory_bytes{kind=\"total\"}",
                  "legendFormat": "Memory Usage"
                }
              },
              "refId": "A",
              "hidden": false
            }
          ]
        },
        "transformations": [],
        "queryOptions": {}
      }
    },
    "vizConfig": {
      "kind": "VizConfig",
      "group": "timeseries",
      "version": "",
      "spec": {
        "options": {
          "legend": {
            "calcs": [],
            "displayMode": "list",
            "placement": "bottom",

```

```

        "showLegend": true
    },
    "tooltip": {
        "mode": "single",
        "sort": "none"
    }
},
"fieldConfig": {
    "defaults": {
        "unit": "percentunit",
        "thresholds": {
            "mode": "absolute",
            "steps": [
                {
                    "value": null,
                    "color": "green"
                }
            ]
        },
    },
    "color": {
        "mode": "palette-classic"
    },
    "custom": {
        "axisCenteredZero": false,
        "axisLabel": "",
        "axisPlacement": "auto",
        "barAlignment": 0,
        "drawStyle": "line",
        "fillOpacity": 10,
        "gradientMode": "none",
        "hideFrom": {
            "legend": false,
            "tooltip": false,
            "viz": false
        },
        "lineInterpolation": "linear",
        "lineWidth": 1,
        "pointSize": 5,
        "scaleDistribution": {
            "type": "linear"
        },
        "showPoints": "never",
        "spanNulls": false,
        "stacking": {
            "group": "A",
            "mode": "none"
        }
    }
}

```

```

        },
        "thresholdsStyle": {
            "mode": "off"
        }
    },
    },
    "overrides": []
}
}
}
},
"panel-6": {
    "kind": "Panel",
    "spec": {
        "id": 6,
        "title": "Load Average",
        "description": "",
        "links": [],
        "data": {
            "kind": "QueryGroup",
            "spec": {
                "queries": [
                    {
                        "kind": "PanelQuery",
                        "spec": {
                            "query": {
                                "kind": "DataQuery",
                                "group": "prometheus",
                                "version": "v0",
                                "datasource": {
                                    "name": "prometheus"
                                },
                                "spec": {
                                    "expr": "ptai_agent_load_average",
                                    "legendFormat": "{{period}}"
                                }
                            },
                            "refId": "A",
                            "hidden": false
                        }
                    }
                ],
                "transformations": [],
                "queryOptions": {}
            }
        }
    }
}

```

```

},
"vizConfig": {
  "kind": "VizConfig",
  "group": "timeseries",
  "version": "",
  "spec": {
    "options": {
      "legend": {
        "calcs": [],
        "displayMode": "list",
        "placement": "bottom",
        "showLegend": true
      },
      "tooltip": {
        "mode": "single",
        "sort": "none"
      }
    }
  },
  "fieldConfig": {
    "defaults": {
      "thresholds": {
        "mode": "absolute",
        "steps": [
          {
            "value": null,
            "color": "green"
          }
        ]
      },
      "color": {
        "mode": "palette-classic"
      },
      "custom": {
        "axisCenteredZero": false,
        "axisLabel": "",
        "axisPlacement": "auto",
        "barAlignment": 0,
        "drawStyle": "line",
        "fillOpacity": 10,
        "gradientMode": "none",
        "hideFrom": {
          "legend": false,
          "tooltip": false,
          "viz": false
        },
        "lineInterpolation": "linear",

```

```

        "lineWidth": 1,
        "pointSize": 5,
        "scaleDistribution": {
            "type": "linear"
        },
        "showPoints": "never",
        "spanNulls": false,
        "stacking": {
            "group": "A",
            "mode": "none"
        },
        "thresholdsStyle": {
            "mode": "off"
        }
    },
    "overrides": []
}
}
}
},
"panel-8": {
    "kind": "Panel",
    "spec": {
        "id": 8,
        "title": "Backend Requests Rate",
        "description": "",
        "links": [],
        "data": {
            "kind": "QueryGroup",
            "spec": {
                "queries": [
                    {
                        "kind": "PanelQuery",
                        "spec": {
                            "query": {
                                "kind": "DataQuery",
                                "group": "prometheus",
                                "version": "v0",
                                "datasource": {
                                    "name": "prometheus"
                                }
                            },
                            "spec": {
                                "expr": "rate(ptai_agent_backend_requests_total[1m])",
                                "legendFormat": "{{method}} ({{code}})"
                            }
                        }
                    }
                ]
            }
        }
    }
}

```

```

        }
      },
      "refId": "A",
      "hidden": false
    }
  ],
  "transformations": [],
  "queryOptions": {}
},
"vizConfig": {
  "kind": "VizConfig",
  "group": "timeseries",
  "version": "",
  "spec": {
    "options": {
      "legend": {
        "calcs": [],
        "displayMode": "list",
        "placement": "bottom",
        "showLegend": true
      },
      "tooltip": {
        "mode": "single",
        "sort": "none"
      }
    },
    "fieldConfig": {
      "defaults": {
        "unit": "reqps",
        "thresholds": {
          "mode": "absolute",
          "steps": [
            {
              "value": null,
              "color": "green"
            }
          ]
        }
      },
      "color": {
        "mode": "palette-classic"
      },
      "custom": {
        "axisCenteredZero": false,
        "axisLabel": ""
      }
    }
  }
}

```

```

    "axisPlacement": "auto",
    "barAlignment": 0,
    "drawStyle": "line",
    "fillOpacity": 10,
    "gradientMode": "none",
    "hideFrom": {
      "legend": false,
      "tooltip": false,
      "viz": false
    },
    "lineInterpolation": "linear",
    "lineWidth": 1,
    "pointSize": 5,
    "scaleDistribution": {
      "type": "linear"
    },
    "showPoints": "never",
    "spanNulls": false,
    "stacking": {
      "group": "A",
      "mode": "none"
    },
    "thresholdsStyle": {
      "mode": "off"
    }
  },
  "overrides": []
},
{
  "panel-9": {
    "kind": "Panel",
    "spec": {
      "id": 9,
      "title": "Heartbeat Failures",
      "description": "",
      "links": [],
      "data": {
        "kind": "QueryGroup",
        "spec": {
          "queries": [
            {
              "kind": "PanelQuery",

```

```

    "spec": {
      "query": {
        "kind": "DataQuery",
        "group": "prometheus",
        "version": "v0",
        "datasource": {
          "name": "prometheus"
        },
        "spec": {
          "expr": "ptai_agent_heartbeat_failures_total",
          "legendFormat": "Heartbeat Failures"
        }
      },
      "refId": "A",
      "hidden": false
    }
  ],
  "transformations": [],
  "queryOptions": {}
},
"vizConfig": {
  "kind": "VizConfig",
  "group": "stat",
  "version": "10.0.0",
  "spec": {
    "options": {
      "colorMode": "value",
      "graphMode": "area",
      "justifyMode": "auto",
      "orientation": "auto",
      "reduceOptions": {
        "calcs": [
          "lastNotNull"
        ],
        "fields": "",
        "values": false
      },
      "textMode": "auto"
    },
    "fieldConfig": {
      "defaults": {
        "thresholds": {
          "mode": "absolute",
          "steps": [

```

```

        {
          "value": null,
          "color": "green"
        },
        {
          "value": 1,
          "color": "red"
        }
      ]
    },
    "color": {
      "mode": "thresholds"
    }
  },
  "overrides": []
}
}
}
}
},
"layout": {
  "kind": "RowLayout",
  "spec": {
    "rows": [
      {
        "kind": "RowLayoutRow",
        "spec": {
          "title": "Agent Status",
          "collapse": false,
          "layout": {
            "kind": "GridLayout",
            "spec": {
              "items": [
                {
                  "kind": "GridLayoutItem",
                  "spec": {
                    "x": 0,
                    "y": 0,
                    "width": 6,
                    "height": 4,
                    "element": {
                      "kind": "ElementReference",
                      "name": "panel-2"
                    }
                  }
                }
              ]
            }
          }
        }
      }
    ]
  }
}

```

```

    },
    {
      "kind": "GridLayoutItem",
      "spec": {
        "x": 6,
        "y": 0,
        "width": 6,
        "height": 4,
        "element": {
          "kind": "ElementReference",
          "name": "panel-3"
        }
      }
    }
  ],
  {
    "kind": "GridLayoutItem",
    "spec": {
      "x": 12,
      "y": 0,
      "width": 12,
      "height": 4,
      "element": {
        "kind": "ElementReference",
        "name": "panel-4"
      }
    }
  }
],
  {
    "kind": "GridLayoutItem",
    "spec": {
      "x": 0,
      "y": 4,
      "width": 12,
      "height": 8,
      "element": {
        "kind": "ElementReference",
        "name": "panel-5"
      }
    }
  }
],
  {
    "kind": "GridLayoutItem",
    "spec": {
      "x": 12,
      "y": 4,
      "width": 12,

```

```

        "height": 8,
        "element": {
            "kind": "ElementReference",
            "name": "panel-6"
        }
    }
}
]
}
}
},
{
    "kind": "RowsLayoutRow",
    "spec": {
        "title": "Backend Connection",
        "collapse": false,
        "layout": {
            "kind": "GridLayout",
            "spec": {
                "items": [
                    {
                        "kind": "GridLayoutItem",
                        "spec": {
                            "x": 0,
                            "y": 0,
                            "width": 12,
                            "height": 8,
                            "element": {
                                "kind": "ElementReference",
                                "name": "panel-8"
                            }
                        }
                    }
                ],
                "x": 12,
                "y": 0,
                "width": 6,
                "height": 4,
                "element": {
                    "kind": "ElementReference",
                    "name": "panel-9"
                }
            }
        }
    }
}

```

```

    },
    {
      "kind": "GridLayoutItem",
      "spec": {
        "x": 18,
        "y": 0,
        "width": 6,
        "height": 4,
        "element": {
          "kind": "ElementReference",
          "name": "panel-10"
        }
      }
    }
  ],
  {
    "kind": "GridLayoutItem",
    "spec": {
      "x": 12,
      "y": 4,
      "width": 12,
      "height": 8,
      "element": {
        "kind": "ElementReference",
        "name": "panel-11"
      }
    }
  }
]
}
},
{
  "kind": "RowsLayoutRow",
  "spec": {
    "title": "Tasks & Adapters",
    "collapse": false,
    "layout": {
      "kind": "GridLayout",
      "spec": {
        "items": [
          {
            "kind": "GridLayoutItem",
            "spec": {
              "x": 0,
              "y": 0,

```

```

        "width": 8,
        "height": 8,
        "element": {
            "kind": "ElementReference",
            "name": "panel-13"
        }
    },
    {
        "kind": "GridLayoutItem",
        "spec": {
            "x": 8,
            "y": 0,
            "width": 8,
            "height": 8,
            "element": {
                "kind": "ElementReference",
                "name": "panel-14"
            }
        }
    },
    {
        "kind": "GridLayoutItem",
        "spec": {
            "x": 16,
            "y": 0,
            "width": 8,
            "height": 8,
            "element": {
                "kind": "ElementReference",
                "name": "panel-15"
            }
        }
    }
]
}
},
{
    "kind": "RowsLayoutRow",
    "spec": {
        "title": "Disk Usage",
        "collapse": false,
        "layout": {
            "kind": "GridLayout",

```

```

"spec": {
  "items": [
    {
      "kind": "GridLayoutItem",
      "spec": {
        "x": 0,
        "y": 0,
        "width": 8,
        "height": 8,
        "element": {
          "kind": "ElementReference",
          "name": "panel-17"
        }
      }
    },
    {
      "kind": "GridLayoutItem",
      "spec": {
        "x": 8,
        "y": 0,
        "width": 8,
        "height": 8,
        "element": {
          "kind": "ElementReference",
          "name": "panel-18"
        }
      }
    },
    {
      "kind": "GridLayoutItem",
      "spec": {
        "x": 16,
        "y": 0,
        "width": 8,
        "height": 8,
        "element": {
          "kind": "ElementReference",
          "name": "panel-19"
        }
      }
    }
  ]
}

```

```

    ]
  },
  "links": [],
  "liveNow": false,
  "preload": false,
  "tags": [
    "ptai",
    "agent"
  ],
  "timeSettings": {
    "from": "now-1h",
    "to": "now",
    "autoRefresh": "10s",
    "autoRefreshIntervals": [
      "5s",
      "10s",
      "30s",
      "1m",
      "5m",
      "15m",
      "30m",
      "1h",
      "2h",
      "1d"
    ],
    "hideTimepicker": false,
    "fiscalYearStartMonth": 0
  },
  "title": "PTAI Agent Monitoring",
  "variables": []
}
}

```

7.16. Сбор технических метрик с PT AI Enterprise Server

PT AI Enterprise Server предоставляет технические метрики всех сервисов в формате Prometheus. Метрики доступны через шлюз Traefik и предназначены для сбора внешними системами мониторинга — Prometheus, VictoriaMetrics и другими совместимыми решениями.

Базовый набор метрик включает:

- HTTP-метрики сервисов;
- gRPC-метрики;
- метрики процессов и среды выполнения.NET;

- метрики брокера RabbitMQ;
- метрики сервиса аутентификации Casgate (SSO);
- метрики шлюза Traefik.

Метрики доступны без аутентификации (анонимный доступ) по путям `/metrics/<Имя_сервиса>`.

Архитектура сбора метрик

Метрики не публикуются отдельными портами во внешнюю сеть. Доступ осуществляется исключительно через Traefik:

1. Запрос приходит на единый адрес шлюза: `https://<host>/metrics/<Имя_сервиса>`.
2. Traefik направляет запрос к соответствующему внутреннему контейнеру, подменяя путь на `/metrics` (для RabbitMQ — `/metrics/per-object`).

В состав дистрибутива PT AI Enterprise Server не входят встроенные Prometheus и Grafana. Необходимо самостоятельно настроить внешнюю систему мониторинга для периодического опроса конечных точек Traefik (`/metrics/<Имя_сервиса>`) и хранения данных, а также импортировать готовые JSON-файлы с панелями в Grafana.

Конечные точки метрик

Все конечные точки доступны через HTTPS и HTTP (через соответствующие точки входа Traefik).

Таблица 16. Конечные точки метрик

Конечная точка	Сервис	Описание метрик
<code>/metrics/auth</code>	Auth	HTTP RED (интенсивность, ошибки, задержка) и активные запросы по контроллерам и действиям, загрузка ЦП, память, сборщик мусора .NET
<code>/metrics/backend</code>	Backend	HTTP RED, gRPC-сервер и gRPC-клиент, процесс и среда выполнения .NET
<code>/metrics/file-content</code>	File Content (FilesStore)	HTTP RED, процесс и среда выполнения .NET
<code>/metrics/notifications</code>	Notifications	HTTP RED, процесс и среда выполнения .NET
<code>/metrics/backend-for-frontend</code>	BFF (Backend for Frontend)	HTTP RED, gRPC-клиент (вызовы к Backend и Scan Scheduler), процесс и среда выполнения .NET

Конечная точка	Сервис	Описание метрик
/metrics/scan-scheduler	Scan Scheduler	HTTP RED, gRPC-сервер (Agents/Queue/AgentsCommunication), процесс и среда выполнения .NET
/metrics/swagger	Swagger-агрегатор	HTTP RED, процесс и среда выполнения .NET
/metrics/rabbitmq	RabbitMQ	Метрики по объектам: очереди, соединения, каналы, память, диск, виртуальная машина Erlang
/metrics/casgate	Casgate (SSO)	Встроенный Prometheus-экспортер Casdoor (/api/metrics): пропускная способность и задержка API, загрузка ЦП, память, среда выполнения Go
/metrics/traefik	Traefik	Точки входа, маршруты, сервисы, HTTP-трафик через шлюз

В этом разделе

[Метрики Traefik \(см. раздел 7.16.1\)](#)

[Метрики RabbitMQ \(см. раздел 7.16.2\)](#)

[Метрики Casgate \(см. раздел 7.16.3\)](#)

[Метрики .NET-сервисов \(см. раздел 7.16.4\)](#)

7.16.1. Метрики Traefik

Traefik — шлюз, через который выполняется доступ ко всем сервисам PT AI Enterprise Server. Метрики позволяют отслеживать входящий HTTP-трафик, маршрутизацию запросов и срок действия TLS-сертификатов.

Таблица 17. Метрики Traefik

Метрика	Тип	Метки	Описание
traefik_entrypoint_requests_total	Counter	status, method, protocol	Общее количество HTTP-запросов, обработанных точкой входа (entrypoint) Traefik
traefik_entrypoint_request_duration_seconds	Histogram	—	Длительность обработки запросов на точке входа

Метрика	Тип	Метки	Описание
traefik_router_requests_total	Counter	—	Общее количество запросов, обработанных конкретным маршрутом
traefik_service_requests_total	Counter	—	Общее количество запросов к вышестоящим сервисам
traefik_open_connections	Gauge	entrypoint, protocol	Открытые соединения по точкам входа и протоколам
traefik_tls_certs_not_after	Gauge	—	Дата и время окончания срока действия TLS-сертификата в Unix-формате

Пример конфигурационного файла дашборда для Grafana

Ниже представлен пример конфигурационного файла, который вы можете импортировать в Grafana для отображения получаемых метрик на графиках дашборда.

```
{
  "annotations": {
    "list": [
      {
        "builtIn": 1,
        "datasource": { "type": "grafana", "uid": "-- Grafana --" },
        "enable": true,
        "hide": true,
        "iconColor": "rgba(0, 211, 255, 1)",
        "name": "Annotations & Alerts",
        "type": "dashboard"
      }
    ]
  },
  "description": "Traefik dashboard prometheus",
  "editable": true,
  "fiscalYearStartMonth": 0,
  "graphTooltip": 1,
  "id": null,
  "links": [],
  "liveNow": false,
  "panels": [
    {
      "collapsed": false,
      "gridPos": { "h": 1, "w": 24, "x": 0, "y": 0 },
      "id": 10,
      "panels": [],
    }
  ]
}
```

```

    "title": "Overview",
    "type": "row"
  },
  {
    "datasource": { "type": "prometheus", "uid": "prometheus" },
    "fieldConfig": {
      "defaults": {
        "color": { "mode": "palette-classic" },
        "unit": "reqps"
      },
      "overrides": []
    },
    "gridPos": { "h": 4, "w": 6, "x": 0, "y": 1 },
    "id": 20,
    "options": {
      "colorMode": "value",
      "graphMode": "area",
      "justifyMode": "auto",
      "orientation": "auto",
      "reduceOptions": { "calcs": ["lastNotNull"], "fields": "", "values": false },
      "showPercentChange": false,
      "textMode": "auto"
    },
    "pluginVersion": "11.5.2",
    "targets": [{
      "expr": "sum(rate(traefik_service_requests_total{service=~\"$service\"}
[ $__rate_interval ]))",
      "legendFormat": "RPS",
      "refId": "A"
    }],
    "title": "Request rate",
    "type": "stat"
  },
  {
    "datasource": { "type": "prometheus", "uid": "prometheus" },
    "fieldConfig": {
      "defaults": {
        "color": { "mode": "palette-classic" },
        "unit": "Bps"
      },
      "overrides": []
    },
    "gridPos": { "h": 4, "w": 6, "x": 6, "y": 1 },
    "id": 21,
    "options": {
      "colorMode": "value",

```

```

    "graphMode": "area",
    "justifyMode": "auto",
    "orientation": "auto",
    "reduceOptions": { "calcs": ["lastNotNull"], "fields": "", "values": false },
    "showPercentChange": false,
    "textMode": "auto"
  },
  "pluginVersion": "11.5.2",
  "targets": [{
    "expr": "sum(rate(traefik_service_responses_bytes_total{service=~\"$service\"}[$__rate_interval]))",
    "legendFormat": "bytes/s",
    "refId": "A"
  }],
  "title": "Response throughput",
  "type": "stat"
},
{
  "datasource": { "type": "prometheus", "uid": "prometheus" },
  "fieldConfig": {
    "defaults": {
      "color": { "mode": "palette-classic" },
      "unit": "ms"
    },
    "overrides": []
  },
  "gridPos": { "h": 4, "w": 6, "x": 12, "y": 1 },
  "id": 22,
  "options": {
    "colorMode": "value",
    "graphMode": "area",
    "justifyMode": "auto",
    "orientation": "auto",
    "reduceOptions": { "calcs": ["lastNotNull"], "fields": "", "values": false },
    "showPercentChange": false,
    "textMode": "auto"
  },
  "pluginVersion": "11.5.2",
  "targets": [{
    "expr": "sum(rate(traefik_service_request_duration_seconds_sum{service=~\"$service\"}[$__rate_interval])) / sum(rate(traefik_service_request_duration_seconds_count{service=~\"$service\"}[$__rate_interval])) * 1000",
    "legendFormat": "avg",
    "refId": "A"
  }],
  "title": "Avg response time",

```

```

    "type": "stat"
  },
  {
    "datasource": { "type": "prometheus", "uid": "prometheus" },
    "fieldConfig": {
      "defaults": {
        "color": { "mode": "thresholds" },
        "max": 1,
        "min": 0,
        "thresholds": {
          "mode": "absolute",
          "steps": [
            { "color": "green", "value": null },
            { "color": "yellow", "value": 0.01 },
            { "color": "red", "value": 0.05 }
          ]
        },
      },
      "unit": "percentunit"
    },
    "overrides": []
  },
  "gridPos": { "h": 4, "w": 6, "x": 18, "y": 1 },
  "id": 23,
  "options": {
    "colorMode": "value",
    "graphMode": "area",
    "justifyMode": "auto",
    "orientation": "auto",
    "reduceOptions": { "calcs": ["lastNotNull"], "fields": "", "values": false },
    "showPercentChange": false,
    "textMode": "auto"
  },
  "pluginVersion": "11.5.2",
  "targets": [{
    "expr": "sum(rate(traefik_service_requests_total{service=~\"$service\"},code=~\\\"5..\\\")
[ $__rate_interval])) / sum(rate(traefik_service_requests_total{service=~\"$service\"}
[ $__rate_interval]))",
    "legendFormat": "5xx",
    "refId": "A"
  }],
  "title": "Error rate (5xx)",
  "type": "stat"
},
{
  "collapsed": false,
  "gridPos": { "h": 1, "w": 24, "x": 0, "y": 5 },

```

```

    "id": 11,
    "panels": [],
    "title": "Requests",
    "type": "row"
  },
  {
    "datasource": { "type": "prometheus", "uid": "prometheus" },
    "fieldConfig": {
      "defaults": {
        "color": { "mode": "palette-classic" },
        "custom": {
          "axisBorderShow": false,
          "axisCenteredZero": false,
          "axisColorMode": "text",
          "axisLabel": "",
          "axisPlacement": "auto",
          "barAlignment": 0,
          "drawStyle": "line",
          "fillOpacity": 20,
          "gradientMode": "opacity",
          "hideFrom": { "legend": false, "tooltip": false, "viz": false },
          "insertNulls": false,
          "lineInterpolation": "smooth",
          "lineWidth": 2,
          "pointSize": 5,
          "scaleDistribution": { "type": "linear" },
          "showPoints": "never",
          "spanNulls": false,
          "stacking": { "group": "A", "mode": "normal" },
          "thresholdsStyle": { "mode": "off" }
        },
        "unit": "reqps"
      },
      "overrides": []
    },
    "gridPos": { "h": 8, "w": 12, "x": 0, "y": 6 },
    "id": 3,
    "options": {
      "legend": { "calcs": ["mean", "max"], "displayMode": "table", "placement": "bottom", "showLegend": true },
      "tooltip": { "mode": "multi", "sort": "desc" }
    },
    "pluginVersion": "11.5.2",
    "targets": [{
      "expr": "sum by (service) (rate(traefik_service_requests_total{service=~\"$service\"}[$__rate_interval]))",
      "legendFormat": "{{service}}",

```

```

    "refId": "A"
  }],
  "title": "Request rate by service",
  "type": "timeseries"
},
{
  "datasource": { "type": "prometheus", "uid": "prometheus" },
  "fieldConfig": {
    "defaults": {
      "color": { "mode": "palette-classic" },
      "custom": {
        "axisBorderShow": false,
        "axisCenteredZero": false,
        "axisColorMode": "text",
        "axisLabel": "",
        "axisPlacement": "auto",
        "barAlignment": 0,
        "drawStyle": "line",
        "fillOpacity": 20,
        "gradientMode": "opacity",
        "hideFrom": { "legend": false, "tooltip": false, "viz": false },
        "insertNulls": false,
        "lineInterpolation": "smooth",
        "lineWidth": 2,
        "pointSize": 5,
        "scaleDistribution": { "type": "linear" },
        "showPoints": "never",
        "spanNulls": false,
        "stacking": { "group": "A", "mode": "normal" },
        "thresholdsStyle": { "mode": "off" }
      },
    },
    "unit": "reqps"
  },
  "overrides": [
    {
      "matcher": { "id": "byRegex", "options": "/200/" },
      "properties": [
        { "id": "color", "value": { "fixedColor": "green", "mode": "fixed" } }
      ]
    },
    {
      "matcher": { "id": "byRegex", "options": "/304/" },
      "properties": [
        { "id": "color", "value": { "fixedColor": "blue", "mode": "fixed" } }
      ]
    },
    {
      "matcher": { "id": "byRegex", "options": "/404/" },
      "properties": [
        { "id": "color", "value": { "fixedColor": "yellow", "mode": "fixed" } }
      ]
    },
    {
      "matcher": { "id": "byRegex", "options": "/503/" },
      "properties": [
        { "id": "color", "value": { "fixedColor": "red", "mode": "fixed" } }
      ]
    }
  ]
},

```

```

    "gridPos": { "h": 8, "w": 12, "x": 12, "y": 6 },
    "id": 5,
    "options": {
      "legend": { "calcs": ["mean", "max"], "displayMode": "table", "placement": "bottom", "showLegend": true },
      "tooltip": { "mode": "multi", "sort": "desc" }
    },
    "pluginVersion": "11.5.2",
    "targets": [{
      "expr": "sum by (code) (rate(traefik_service_requests_total{service=~\"$service\"}[ $__rate_interval ]))",
      "legendFormat": "{{code}}",
      "refId": "A"
    }],
    "title": "Request rate by status code",
    "type": "timeseries"
  },
  {
    "datasource": { "type": "prometheus", "uid": "prometheus" },
    "fieldConfig": {
      "defaults": {
        "color": { "mode": "palette-classic" },
        "custom": { "hideFrom": { "legend": false, "tooltip": false, "viz": false } }
      },
      "unit": "reqps"
    },
    "overrides": []
  },
  {
    "gridPos": { "h": 8, "w": 12, "x": 0, "y": 14 },
    "id": 7,
    "options": {
      "legend": { "displayMode": "table", "placement": "right", "showLegend": true, "values": ["value"] },
      "pieType": "donut",
      "reduceOptions": { "calcs": ["lastNotNull"], "fields": "", "values": false },
      "tooltip": { "mode": "single", "sort": "none" }
    },
    "pluginVersion": "11.5.2",
    "targets": [{
      "expr": "sum by (service) (rate(traefik_service_requests_total{service=~\"$service\"}[ $__rate_interval ]))",
      "legendFormat": "{{service}}",
      "refId": "A"
    }],
    "title": "Requests by service",
    "type": "piechart"
  },

```

```

{
  "datasource": { "type": "prometheus", "uid": "prometheus" },
  "fieldConfig": {
    "defaults": {
      "color": { "mode": "palette-classic" },
      "custom": { "hideFrom": { "legend": false, "tooltip": false, "viz": false }
    },
    "unit": "reqps"
  },
  "overrides": []
},
"gridPos": { "h": 8, "w": 12, "x": 12, "y": 14 },
"id": 8,
"options": {
  "legend": { "displayMode": "table", "placement": "right", "showLegend": true,
"values": [{"value"}] },
  "pieType": "donut",
  "reduceOptions": { "calcs": ["lastNotNull"], "fields": "", "values": false },
  "tooltip": { "mode": "single", "sort": "none" }
},
"pluginVersion": "11.5.2",
"targets": [{
  "expr": "sum by (method) (rate(traefik_service_requests_total{service=~\"$service\"}[$__rate_interval]))",
  "legendFormat": "{{method}}",
  "refId": "A"
}],
"title": "Requests by method",
"type": "piechart"
},
{
  "collapsed": false,
  "gridPos": { "h": 1, "w": 24, "x": 0, "y": 22 },
  "id": 12,
  "panels": [],
  "title": "Responses",
  "type": "row"
},
{
  "datasource": { "type": "prometheus", "uid": "prometheus" },
  "fieldConfig": {
    "defaults": {
      "color": { "mode": "palette-classic" },
      "custom": {
        "axisBorderShow": false,
        "axisCenteredZero": false,
        "axisColorMode": "text",

```

```

    "axisLabel": "",
    "axisPlacement": "auto",
    "barAlignment": 0,
    "drawStyle": "line",
    "fillOpacity": 20,
    "gradientMode": "opacity",
    "hideFrom": { "legend": false, "tooltip": false, "viz": false },
    "insertNulls": false,
    "lineInterpolation": "smooth",
    "lineWidth": 2,
    "pointSize": 5,
    "scaleDistribution": { "type": "linear" },
    "showPoints": "never",
    "spanNulls": false,
    "stacking": { "group": "A", "mode": "normal" },
    "thresholdsStyle": { "mode": "off" }
  },
  "unit": "Bps"
},
"overrides": []
},
"gridPos": { "h": 8, "w": 12, "x": 0, "y": 23 },
"id": 30,
"options": {
  "legend": { "calcs": ["mean", "max"], "displayMode": "table", "placement": "bottom", "showLegend": true },
  "tooltip": { "mode": "multi", "sort": "desc" }
},
"pluginVersion": "11.5.2",
"targets": [{
  "expr": "sum by (service) (rate(traefik_service_responses_bytes_total{service=~\"$service\"}[$__rate_interval]))",
  "legendFormat": "{{service}}",
  "refId": "A"
}],
"title": "Response bytes rate by service",
"type": "timeseries"
},
{
  "datasource": { "type": "prometheus", "uid": "prometheus" },
  "fieldConfig": {
    "defaults": {
      "color": { "mode": "palette-classic" },
      "custom": {
        "axisBorderShow": false,
        "axisCenteredZero": false,
        "axisColorMode": "text",

```

```

    "axisLabel": "",
    "axisPlacement": "auto",
    "barAlignment": 0,
    "drawStyle": "line",
    "fillOpacity": 10,
    "gradientMode": "none",
    "hideFrom": { "legend": false, "tooltip": false, "viz": false },
    "insertNulls": false,
    "lineInterpolation": "smooth",
    "lineWidth": 2,
    "pointSize": 5,
    "scaleDistribution": { "type": "linear" },
    "showPoints": "never",
    "spanNulls": false,
    "stacking": { "group": "A", "mode": "none" },
    "thresholdsStyle": { "mode": "off" }
  },
  "unit": "s"
},
"overrides": []
},
"gridPos": { "h": 8, "w": 12, "x": 12, "y": 23 },
"id": 31,
"options": {
  "legend": { "calcs": ["mean", "max"], "displayMode": "table", "placement": "bottom", "showLegend": true },
  "tooltip": { "mode": "multi", "sort": "desc" }
},
"pluginVersion": "11.5.2",
"targets": [
  {
    "expr": "histogram_quantile(0.50, sum by (le) (rate(traefik_service_request_duration_seconds_bucket{service=~\"$service\"}[$__rate_interval])))",
    "legendFormat": "p50",
    "refId": "A"
  },
  {
    "expr": "histogram_quantile(0.95, sum by (le) (rate(traefik_service_request_duration_seconds_bucket{service=~\"$service\"}[$__rate_interval])))",
    "legendFormat": "p95",
    "refId": "B"
  },
  {
    "expr": "histogram_quantile(0.99, sum by (le) (rate(traefik_service_request_duration_seconds_bucket{service=~\"$service\"}[$__rate_interval])))",
    "legendFormat": "p99",
    "refId": "C"
  }
]

```

```

    }
  ],
  "title": "Response time percentiles",
  "type": "timeseries"
},
{
  "datasource": { "type": "prometheus", "uid": "prometheus" },
  "fieldConfig": {
    "defaults": {
      "color": { "mode": "palette-classic" },
      "custom": { "hideFrom": { "legend": false, "tooltip": false, "viz": false }
    },
    "unit": "short"
  },
  "overrides": []
},
"gridPos": { "h": 8, "w": 12, "x": 0, "y": 31 },
"id": 2,
"options": {
  "legend": { "displayMode": "table", "placement": "right", "showLegend": true,
"values": ["value", "percent"] },
  "pieType": "donut",
  "reduceOptions": { "calcs": ["lastNotNull"], "fields": "", "values": false },
  "tooltip": { "mode": "single", "sort": "none" }
},
"pluginVersion": "11.5.2",
"targets": [{
  "expr": "sum by (code) (rate(traefik_service_requests_total{service=~\"$service\"}[$_rate_interval]))",
  "legendFormat": "{{code}}",
  "refId": "A"
}],
"title": "Response codes ($service)",
"type": "piechart"
},
{
  "datasource": { "type": "prometheus", "uid": "prometheus" },
  "fieldConfig": {
    "defaults": {
      "color": { "mode": "palette-classic" },
      "custom": {
        "axisBorderShow": false,
        "axisCenteredZero": false,
        "axisColorMode": "text",
        "axisLabel": "",
        "axisPlacement": "auto",
        "barAlignment": 0,

```

```

        "drawStyle": "line",
        "fillOpacity": 20,
        "gradientMode": "opacity",
        "hideFrom": { "legend": false, "tooltip": false, "viz": false },
        "insertNulls": false,
        "lineInterpolation": "smooth",
        "lineWidth": 2,
        "pointSize": 5,
        "scaleDistribution": { "type": "linear" },
        "showPoints": "never",
        "spanNulls": false,
        "stacking": { "group": "A", "mode": "normal" },
        "thresholdsStyle": { "mode": "off" }
    },
    "unit": "Bps"
},
"overrides": []
},
"gridPos": { "h": 8, "w": 12, "x": 12, "y": 31 },
"id": 32,
"options": {
    "legend": { "calcs": ["mean", "max"], "displayMode": "table", "placement": "bottom", "showLegend": true },
    "tooltip": { "mode": "multi", "sort": "desc" }
},
"pluginVersion": "11.5.2",
"targets": [{
    "expr": "sum by (service) (rate(traefik_service_requests_bytes_total{service=~\"$service\"}[$__rate_interval]))",
    "legendFormat": "{{service}}",
    "refId": "A"
}],
"title": "Request bytes rate by service",
"type": "timeseries"
}
],
"refresh": "10s",
"schemaVersion": 39,
"tags": ["traefik", "prometheus"],
"templating": {
    "list": [
        {
            "allValue": ".*",
            "current": { "selected": true, "text": "All", "value": "$__all" },
            "datasource": { "type": "prometheus", "uid": "prometheus" },
            "definition": "label_values(traefik_service_requests_total, service)",
            "hide": 0,

```

```

    "includeAll": true,
    "label": "Service",
    "multi": true,
    "name": "service",
    "options": [],
    "query": {
      "qryType": 1,
      "query": "label_values(traefik_service_requests_total, service)",
      "refId": "Prometheus-service-Variable-Query"
    },
    "refresh": 1,
    "regex": "",
    "skipUrlSync": false,
    "sort": 1,
    "type": "query"
  }
]
},
"time": { "from": "now-1h", "to": "now" },
"timepicker": {
  "refresh_intervals": ["5s", "10s", "30s", "1m", "5m", "15m", "30m", "1h"]
},
"timezone": "browser",
"title": "Traefik",
"uid": "traefik",
"version": 14,
"weekStart": ""
}

```

7.16.2. Метрики RabbitMQ

RabbitMQ — брокер сообщений, обеспечивающий взаимодействие между сервисами PT AI Enterprise Server. Метрики позволяют отслеживать глубину очередей, количество потребителей (consumers), интенсивность доставки сообщений и общее состояние брокера.

Таблица 18. Метрики RabbitMQ

Метрика	Тип	Метки	Описание
rabbitmq_queue_messages	Gauge	—	Общее число сообщений в очереди
rabbitmq_queue_messages_ready	Gauge	—	Число сообщений, готовых к доставке
rabbitmq_queue_messages_unacked	Gauge	—	Число сообщений, ожидающих подтверждения (ack)

Метрика	Тип	Метки	Описание
rabbitmq_queue_consumers	Gauge	—	Число потребителей в очереди
rabbitmq_queue_messages_published_total	Counter	—	Общее число сообщений, опубликованных в очередь
rabbitmq_queue_messages_delivered_total	Counter	—	Общее число сообщений, доставленных потребителям
rabbitmq_connections	Gauge	—	Число активных соединений с брокером
rabbitmq_channels	Gauge	—	Число открытых каналов
rabbitmq_global_messages_*	Various	—	Агрегированные метрики по брокеру: received, delivered, acknowledged, redelivered, dead-lettered

Пример конфигурационного файла дашборда для Grafana

Ниже представлен пример конфигурационного файла, который вы можете импортировать в Grafana для отображения получаемых метрик на графиках дашборда.

```
{
  "annotations": {
    "list": [
      {
        "builtIn": 1,
        "datasource": {
          "type": "grafana",
          "uid": "-- Grafana --"
        },
        "enable": true,
        "hide": true,
        "iconColor": "rgba(255, 152, 0, 1)",
        "name": "Annotations & Alerts",
        "type": "dashboard"
      }
    ]
  },
  "description": "RabbitMQ – broker overview and per-queue depth, consumers and rates (job=rabbitmq).",
  "editable": true,
  "fiscalYearStartMonth": 0,
  "graphTooltip": 1,
  "id": null,
```

```

"links": [
  {
    "icon": "external link",
    "tags": [],
    "title": "Services",
    "type": "link",
    "url": "/d/services"
  },
  {
    "icon": "external link",
    "tags": [],
    "title": "PostgreSQL",
    "type": "link",
    "url": "/d/postgres"
  }
],
"liveNow": false,
"panels": [
  {
    "collapsed": false,
    "gridPos": {
      "h": 1,
      "w": 24,
      "x": 0,
      "y": 0
    },
    "id": 100,
    "panels": [],
    "title": "Broker overview",
    "type": "row",
    "description": "Broker-level totals. The Queue filter uses grouped message-
type names."
  },
  {
    "datasource": {
      "type": "prometheus",
      "uid": "prometheus"
    },
    "description": "Prometheus scrape target health.",
    "fieldConfig": {
      "defaults": {
        "color": {
          "mode": "thresholds"
        },
        "mappings": [
          {

```

```

        "options": {
            "0": {
                "color": "red",
                "index": 0,
                "text": "DOWN"
            }
        },
        "type": "value"
    },
    {
        "options": {
            "1": {
                "color": "green",
                "index": 1,
                "text": "UP"
            }
        },
        "type": "value"
    }
],
"thresholds": {
    "mode": "absolute",
    "steps": [
        {
            "color": "red",
            "value": null
        },
        {
            "color": "green",
            "value": 1
        }
    ]
},
"overrides": []
},
"gridPos": {
    "h": 4,
    "w": 3,
    "x": 0,
    "y": 1
},
"id": 1,
"options": {
    "colorMode": "background",
    "graphMode": "none",

```

```

    "justifyMode": "center",
    "orientation": "auto",
    "reduceOptions": {
      "calcs": [
        "lastNotNull"
      ],
      "fields": "",
      "values": false
    },
    "showPercentChange": false,
    "textMode": "value"
  },
  "pluginVersion": "11.5.2",
  "targets": [
    {
      "expr": "up{job=\"rabbitmq\"}",
      "legendFormat": "rabbitmq",
      "refId": "A"
    }
  ],
  "title": "Target",
  "type": "stat"
},
{
  "datasource": {
    "type": "prometheus",
    "uid": "prometheus"
  },
  "fieldConfig": {
    "defaults": {
      "color": {
        "mode": "thresholds"
      },
      "thresholds": {
        "mode": "absolute",
        "steps": [
          {
            "color": "blue",
            "value": null
          }
        ]
      }
    },
    "overrides": []
  },
  "gridPos": {

```

```

    "h": 4,
    "w": 3,
    "x": 3,
    "y": 1
  },
  "id": 2,
  "options": {
    "colorMode": "value",
    "graphMode": "area",
    "justifyMode": "auto",
    "orientation": "auto",
    "reduceOptions": {
      "calcs": [
        "lastNotNull"
      ],
      "fields": "",
      "values": false
    },
    "showPercentChange": false,
    "textMode": "auto"
  },
  "targets": [
    {
      "expr": "sum(rabbitmq_connections{job=\"rabbitmq\"})",
      "legendFormat": "connections",
      "refId": "A"
    }
  ],
  "title": "Connections",
  "type": "stat"
},
{
  "datasource": {
    "type": "prometheus",
    "uid": "prometheus"
  },
  "fieldConfig": {
    "defaults": {
      "color": {
        "mode": "thresholds"
      },
      "thresholds": {
        "mode": "absolute",
        "steps": [
          {
            "color": "blue",

```

```

        "value": null
      }
    ]
  }
},
"overrides": [],
},
"gridPos": {
  "h": 4,
  "w": 3,
  "x": 6,
  "y": 1
},
"id": 3,
"options": {
  "colorMode": "value",
  "graphMode": "area",
  "justifyMode": "auto",
  "orientation": "auto",
  "reduceOptions": {
    "calcs": [
      "lastNotNull"
    ],
    "fields": "",
    "values": false
  },
  "showPercentChange": false,
  "textMode": "auto"
},
"targets": [
  {
    "expr": "sum(rabbitmq_channels{job=\"rabbitmq\"})",
    "legendFormat": "channels",
    "refId": "A"
  }
],
"title": "Channels",
"type": "stat"
},
{
  "datasource": {
    "type": "prometheus",
    "uid": "prometheus"
  },
  "fieldConfig": {
    "defaults": {

```

```

    "color": {
      "mode": "thresholds"
    },
    "thresholds": {
      "mode": "absolute",
      "steps": [
        {
          "color": "blue",
          "value": null
        }
      ]
    },
    "overrides": []
  },
  "gridPos": {
    "h": 4,
    "w": 3,
    "x": 9,
    "y": 1
  },
  "id": 4,
  "options": {
    "colorMode": "value",
    "graphMode": "area",
    "justifyMode": "auto",
    "orientation": "auto",
    "reduceOptions": {
      "calcs": [
        "lastNotNull"
      ],
      "fields": "",
      "values": false
    },
    "showPercentChange": false,
    "textMode": "auto"
  },
  "targets": [
    {
      "expr": "sum(rabbitmq_queues{job=\"rabbitmq\"})",
      "legendFormat": "queues",
      "refId": "A"
    }
  ],
  "title": "Queues",
  "type": "stat"

```

```

    },
    {
      "datasource": {
        "type": "prometheus",
        "uid": "prometheus"
      },
      "description": "Total ready + unacked messages for the selection. Values are for queues grouped by message type: GUID and AI.Enterprise.Notifications.Messages. are stripped, and consumer queues with the same message are summed.",
      "fieldConfig": {
        "defaults": {
          "color": {
            "mode": "thresholds"
          },
          "thresholds": {
            "mode": "absolute",
            "steps": [
              {
                "color": "green",
                "value": null
              },
              {
                "color": "yellow",
                "value": 1000
              },
              {
                "color": "red",
                "value": 10000
              }
            ]
          }
        },
        "overrides": []
      },
      "gridPos": {
        "h": 4,
        "w": 3,
        "x": 15,
        "y": 1
      },
      "id": 6,
      "options": {
        "colorMode": "value",
        "graphMode": "area",
        "justifyMode": "auto",
        "orientation": "auto",
        "reduceOptions": {

```

```

        "calcs": [
            "lastNotNull"
        ],
        "fields": "",
        "values": false
    },
    "showPercentChange": false,
    "textMode": "auto"
},
"targets": [
    {
        "expr": "sum(rabbitmq_queue_messages{job=\"rabbitmq\", vhost=~\"$vhost\", queue=~\"(?:[0-9a-fA-F]{8}-[0-9a-fA-F]{4}-[0-9a-fA-F]{4}-[0-9a-fA-F]{4}-[0-9a-fA-F]{12})-(?:AI[.]Enterprise[.]Notifications[.]Messages[.])?${queue:regex}$\", queue=~\".+\"})",
        "legendFormat": "messages",
        "refId": "A"
    }
],
"title": "Messages (total)",
"type": "stat"
},
{
    "datasource": {
        "type": "prometheus",
        "uid": "prometheus"
    },
    "fieldConfig": {
        "defaults": {
            "color": {
                "mode": "thresholds"
            },
            "thresholds": {
                "mode": "absolute",
                "steps": [
                    {
                        "color": "green",
                        "value": null
                    },
                    {
                        "color": "yellow",
                        "value": 0.7
                    },
                    {
                        "color": "red",
                        "value": 0.9
                    }
                ]
            }
        }
    }
}

```

```

        ],
        },
        "unit": "percentunit",
        "max": 1,
        "min": 0
    },
    "overrides": []
},
"gridPos": {
    "h": 4,
    "w": 3,
    "x": 18,
    "y": 1
},
"id": 7,
"options": {
    "colorMode": "value",
    "graphMode": "area",
    "justifyMode": "auto",
    "orientation": "auto",
    "reduceOptions": {
        "calcs": [
            "lastNotNull"
        ],
        "fields": "",
        "values": false
    },
    "showPercentChange": false,
    "textMode": "auto"
},
"targets": [
    {
        "expr": "max(rabbitmq_process_resident_memory_bytes{job=\"rabbitmq\"} / rabbitmq_resident_memory_limit_bytes{job=\"rabbitmq\"})",
        "legendFormat": "mem",
        "refId": "A"
    }
],
"title": "Memory used",
"type": "stat"
},
{
    "datasource": {
        "type": "prometheus",
        "uid": "prometheus"
    },

```

```

"fieldConfig": {
  "defaults": {
    "color": {
      "mode": "thresholds"
    },
    "thresholds": {
      "mode": "absolute",
      "steps": [
        {
          "color": "blue",
          "value": null
        }
      ]
    },
    "unit": "bytes"
  },
  "overrides": []
},
"gridPos": {
  "h": 4,
  "w": 3,
  "x": 21,
  "y": 1
},
"id": 8,
"options": {
  "colorMode": "value",
  "graphMode": "none",
  "justifyMode": "auto",
  "orientation": "auto",
  "reduceOptions": {
    "calcs": [
      "lastNotNull"
    ],
    "fields": "",
    "values": false
  },
  "showPercentChange": false,
  "textMode": "auto"
},
"targets": [
  {
    "expr": "min(rabbitmq_disk_space_available_bytes{job=\"rabbitmq\"})",
    "legendFormat": "disk",
    "refId": "A"
  }
]

```

```

    ],
    "title": "Disk free",
    "type": "stat"
  },
  {
    "collapsed": false,
    "gridPos": {
      "h": 1,
      "w": 24,
      "x": 0,
      "y": 5
    },
    "id": 102,
    "panels": [],
    "title": "Queue depth over time",
    "type": "row",
    "description": "Values are for queues grouped by message type: GUID and AI.Enterprise.Notifications.Messages. are stripped, and consumer queues with the same message are summed."
  },
  {
    "datasource": {
      "type": "prometheus",
      "uid": "prometheus"
    },
    "description": "Ready messages waiting in each queue. Values are for queues grouped by message type: GUID and AI.Enterprise.Notifications.Messages. are stripped, and consumer queues with the same message are summed.",
    "fieldConfig": {
      "defaults": {
        "color": {
          "mode": "palette-classic"
        },
        "custom": {
          "axisBorderShow": false,
          "axisCenteredZero": false,
          "axisColorMode": "text",
          "axisLabel": "",
          "axisPlacement": "auto",
          "barAlignment": 0,
          "drawStyle": "line",
          "fillOpacity": 10,
          "gradientMode": "none",
          "hideFrom": {
            "legend": false,
            "tooltip": false,
            "viz": false
          }
        }
      }
    }
  }

```

```

    },
    "insertNulls": false,
    "lineInterpolation": "smooth",
    "lineWidth": 2,
    "pointSize": 5,
    "scaleDistribution": {
      "type": "linear"
    },
    "showPoints": "never",
    "spanNulls": false,
    "stacking": {
      "group": "A",
      "mode": "none"
    },
    "thresholdsStyle": {
      "mode": "off"
    }
  },
  "unit": "short"
},
"overrides": []
},
"gridPos": {
  "h": 8,
  "w": 12,
  "x": 0,
  "y": 6
},
"id": 20,
"options": {
  "legend": {
    "calcs": [
      "mean",
      "lastNotNull",
      "max"
    ],
    "displayMode": "table",
    "placement": "bottom",
    "showLegend": true
  },
  "tooltip": {
    "mode": "multi",
    "sort": "desc"
  }
},
"targets": [

```

```

    {
      "expr": "sum by (queue_name) (label_replace(rabbitmq_queue_messages_ready{jo
ob=\"rabbitmq\", vhost=~\"$vhost\", queue=~\"(?:[0-9a-fA-F]{8}-[0-9a-fA-F]{4}-[0-9a-
fA-F]{4}-[0-9a-fA-F]{4}-[0-9a-fA-F]{12})-)?(?:AI[.]Enterprise[.]Notifications[.]Messages[.]?){queue:regex}$\", queue=~\".+
\\\", \\\"queue_name\\\", \\\"$1\\\", \\\"queue\\\", \\\"(?:[0-9a-fA-F]{8}-[0-9a-fA-F]{4}-[0-9a-fA-
F]{4}-[0-9a-fA-F]{4}-[0-9a-fA-F]{12})-)?(?:AI[.]Enterprise[.]Notifications[.]Messages[.]?(.+)\\\")\"),
      \"legendFormat\": \"{queue_name}\",
      \"refId\": \"A\"
    }
  ],
  \"title\": \"Ready messages\",
  \"type\": \"timeseries\"
},
{
  \"datasource\": {
    \"type\": \"prometheus\",
    \"uid\": \"prometheus\"
  },
  \"description\": \"Unacked messages delivered to consumers. Values are for queues
grouped by message type: GUID and AI.Enterprise.Notifications.Messages. are stripped,
and consumer queues with the same message are summed.\",
  \"fieldConfig\": {
    \"defaults\": {
      \"color\": {
        \"mode\": \"palette-classic\"
      },
      \"custom\": {
        \"axisBorderShow\": false,
        \"axisCenteredZero\": false,
        \"axisColorMode\": \"text\",
        \"axisLabel\": \"\",
        \"axisPlacement\": \"auto\",
        \"barAlignment\": 0,
        \"drawStyle\": \"line\",
        \"fillOpacity\": 10,
        \"gradientMode\": \"none\",
        \"hideFrom\": {
          \"legend\": false,
          \"tooltip\": false,
          \"viz\": false
        },
        \"insertNulls\": false,
        \"lineInterpolation\": \"smooth\",
        \"lineWidth\": 2,
        \"pointSize\": 5,
        \"scaleDistribution\": {

```

```

        "type": "linear"
    },
    "showPoints": "never",
    "spanNulls": false,
    "stacking": {
        "group": "A",
        "mode": "none"
    },
    "thresholdsStyle": {
        "mode": "off"
    }
},
"unit": "short"
},
"overrides": []
},
"gridPos": {
    "h": 8,
    "w": 12,
    "x": 12,
    "y": 6
},
"id": 21,
"options": {
    "legend": {
        "calcs": [
            "mean",
            "lastNotNull",
            "max"
        ],
        "displayMode": "table",
        "placement": "bottom",
        "showLegend": true
    },
    "tooltip": {
        "mode": "multi",
        "sort": "desc"
    }
},
"targets": [
    {
        "expr": "sum by (queue_name) (label_replace(rabbitmq_queue_messages_unacked
{job=\"rabbitmq\", vhost=~\"$vhost\", queue=~\"(?:[0-9a-fA-F]{8}-[0-9a-fA-F]{4}-
[0-9a-fA-F]{4}-[0-9a-fA-F]{4}-[0-9a-fA-F]{12}-)?
(?:AI[.]Enterprise[.]Notifications[.]Messages[.]?){queue:regex}$\", queue=~\".+

```

```
\", \"queue_name\\\", \"\\$1\\\", \"queue\\\", \"(?:[0-9a-fA-F]{8}-[0-9a-fA-F]{4}-[0-9a-fA-F]{4}-[0-9a-fA-F]{4}-[0-9a-fA-F]{12})-?(?:AI[.]Enterprise[.]Notifications[.]Messages[.]?)(.+)\\)\")\",
    \"legendFormat\": \"{{queue_name}}\",
    \"refId\": \"A\"
  }
],
\"title\": \"Unacked messages\",
\"type\": \"timeseries\"
},
{
  \"collapsed\": false,
  \"gridPos\": {
    \"h\": 1,
    \"w\": 24,
    \"x\": 0,
    \"y\": 14
  },
  \"id\": 103,
  \"panels\": [],
  \"title\": \"Throughput\",
  \"type\": \"row\",
  \"description\": \"Values are for queues grouped by message type: GUID and AI.Enterprise.Notifications.Messages. are stripped, and consumer queues with the same message are summed.\"
},
{
  \"datasource\": {
    \"type\": \"prometheus\",
    \"uid\": \"prometheus\"
  },
  \"fieldConfig\": {
    \"defaults\": {
      \"color\": {
        \"mode\": \"palette-classic\"
      },
      \"custom\": {
        \"axisBorderShow\": false,
        \"axisCenteredZero\": false,
        \"axisColorMode\": \"text\",
        \"axisLabel\": \"\",
        \"axisPlacement\": \"auto\",
        \"barAlignment\": 0,
        \"drawStyle\": \"line\",
        \"fillOpacity\": 10,
        \"gradientMode\": \"none\",
        \"hideFrom\": {
```

```

        "legend": false,
        "tooltip": false,
        "viz": false
    },
    "insertNulls": false,
    "lineInterpolation": "smooth",
    "lineWidth": 2,
    "pointSize": 5,
    "scaleDistribution": {
        "type": "linear"
    },
    "showPoints": "never",
    "spanNulls": false,
    "stacking": {
        "group": "A",
        "mode": "none"
    },
    "thresholdsStyle": {
        "mode": "off"
    }
},
"unit": "ops"
},
"overrides": []
},
"gridPos": {
    "h": 8,
    "w": 8,
    "x": 0,
    "y": 15
},
"id": 30,
"options": {
    "legend": {
        "calcs": [
            "mean",
            "lastNotNull"
        ],
        "displayMode": "table",
        "placement": "bottom",
        "showLegend": true
    },
    "tooltip": {
        "mode": "multi",
        "sort": "desc"
    }
}

```

```

    },
    "targets": [
      {
        "expr": "sum by (queue_name) (label_replace(rate(rabbitmq_queue_messages_published_total{job=\"rabbitmq\", queue_vhost=~\"$vhost\", queue=~\"(?:[0-9a-fA-F]{8}-[0-9a-fA-F]{4}-[0-9a-fA-F]{4}-[0-9a-fA-F]{4}-[0-9a-fA-F]{12})-(?:AI[.]Enterprise[.]Notifications[.]Messages[.]?${queue:regex}$\", queue=~\".+\\\"}[$__rate_interval])), \"queue_name\", \"$1\", \"queue\", \"(?:[0-9a-fA-F]{8}-[0-9a-fA-F]{4}-[0-9a-fA-F]{4}-[0-9a-fA-F]{4}-[0-9a-fA-F]{12})-(?:AI[.]Enterprise[.]Notifications[.]Messages[.]?(.+)\\\"))\", \"legendFormat\": \"{queue_name}\", \"refId\": \"A\"
      }
    ],
    "title": "Publish rate",
    "type": "timeseries",
    "description": "Publish rate. Values are for queues grouped by message type: GU ID and AI.Enterprise.Notifications.Messages. are stripped, and consumer queues with the same message are summed."
  },
  {
    "datasource": {
      "type": "prometheus",
      "uid": "prometheus"
    },
    "fieldConfig": {
      "defaults": {
        "color": {
          "mode": "palette-classic"
        },
        "custom": {
          "axisBorderShow": false,
          "axisCenteredZero": false,
          "axisColorMode": "text",
          "axisLabel": "",
          "axisPlacement": "auto",
          "barAlignment": 0,
          "drawStyle": "line",
          "fillOpacity": 10,
          "gradientMode": "none",
          "hideFrom": {
            "legend": false,
            "tooltip": false,
            "viz": false
          },
          "insertNulls": false,
          "lineInterpolation": "smooth",
          "lineWidth": 2,

```

```

        "pointSize": 5,
        "scaleDistribution": {
            "type": "linear"
        },
        "showPoints": "never",
        "spanNulls": false,
        "stacking": {
            "group": "A",
            "mode": "none"
        },
        "thresholdsStyle": {
            "mode": "off"
        }
    },
    "unit": "ops"
},
"overrides": []
},
"gridPos": {
    "h": 8,
    "w": 8,
    "x": 8,
    "y": 15
},
"id": 31,
"options": {
    "legend": {
        "calcs": [
            "mean",
            "lastNotNull"
        ],
        "displayMode": "table",
        "placement": "bottom",
        "showLegend": true
    },
    "tooltip": {
        "mode": "multi",
        "sort": "desc"
    }
},
"targets": [
    {
        "expr": "sum by (queue_name) (label_replace(rate(rabbitmq_queue_messages_de
livered_total{job=\"rabbitmq\", vhost=~\"$vhost\", queue=~\"(?:[0-9a-fA-F]{8}-[0-9a-
fA-F]{4}-[0-9a-fA-F]{4}-[0-9a-fA-F]{4}-[0-9a-fA-F]{12}-)?
(?:AI[.]Enterprise[.]Notifications[.]Messages[.])?${queue:regex}$\", queue=~\".+\"}

```

```
[${__rate_interval}], \"queue_name\", \"\$1\", \"queue\", \"(?:[0-9a-fA-F]{8}-[0-9a-fA-F]{4}-[0-9a-fA-F]{4}-[0-9a-fA-F]{4}-[0-9a-fA-F]{12})?
(?:AI[.]Enterprise[.]Notifications[.]Messages[.])?(.+)\")\",
    \"legendFormat\": \"{{queue_name}}\",
    \"refId\": \"A\"
  }
],
  \"title\": \"Deliver rate\",
  \"type\": \"timeseries\",
  \"description\": \"Deliver rate. Values are for queues grouped by message type: GU
ID and AI.Enterprise.Notifications.Messages. are stripped, and consumer queues with t
he same message are summed.\"
},
{
  \"datasource\": {
    \"type\": \"prometheus\",
    \"uid\": \"prometheus\"
  },
  \"fieldConfig\": {
    \"defaults\": {
      \"color\": {
        \"mode\": \"palette-classic\"
      },
      \"custom\": {
        \"axisBorderShow\": false,
        \"axisCenteredZero\": false,
        \"axisColorMode\": \"text\",
        \"axisLabel\": \"\",
        \"axisPlacement\": \"auto\",
        \"barAlignment\": 0,
        \"drawStyle\": \"line\",
        \"fillOpacity\": 10,
        \"gradientMode\": \"none\",
        \"hideFrom\": {
          \"legend\": false,
          \"tooltip\": false,
          \"viz\": false
        },
        \"insertNulls\": false,
        \"lineInterpolation\": \"smooth\",
        \"lineWidth\": 2,
        \"pointSize\": 5,
        \"scaleDistribution\": {
          \"type\": \"linear\"
        },
        \"showPoints\": \"never\",
        \"spanNulls\": false,
```

```

        "stacking": {
            "group": "A",
            "mode": "none"
        },
        "thresholdsStyle": {
            "mode": "off"
        }
    },
    "unit": "ops"
},
"overrides": []
},
"gridPos": {
    "h": 8,
    "w": 8,
    "x": 16,
    "y": 15
},
"id": 32,
"options": {
    "legend": {
        "calcs": [
            "mean",
            "lastNotNull"
        ],
        "displayMode": "table",
        "placement": "bottom",
        "showLegend": true
    },
    "tooltip": {
        "mode": "multi",
        "sort": "desc"
    }
},
"targets": [
    {
        "expr": "sum by (queue_name) (label_replace(rate(rabbitmq_queue_messages_ack_total{job=\"rabbitmq\", vhost=~\"$vhost\", queue=~\"(?:[0-9a-fA-F]{8}-[0-9a-fA-F]{4}-[0-9a-fA-F]{4}-[0-9a-fA-F]{4}-[0-9a-fA-F]{12})-)?(?:AI[.]Enterprise[.]Notifications[.]Messages[.]?${queue:regex}$\", queue=~\".+\"}[$__rate_interval]), \"queue_name\", \"$1\", \"queue\", \"(?:[0-9a-fA-F]{8}-[0-9a-fA-F]{4}-[0-9a-fA-F]{4}-[0-9a-fA-F]{4}-[0-9a-fA-F]{12})-)?(?:AI[.]Enterprise[.]Notifications[.]Messages[.]?(.+)\"))\", \"legendFormat\": \"ack {{queue_name}}\", \"refId\": \"A\"
    },
    {

```

```

        "expr": "sum by (queue_name) (label_replace(rate(rabbitmq_queue_messages_re
delivered_total{job=\"rabbitmq\", vhost=~\"$vhost\", queue=~\"(?:[0-9a-fA-F]{8}-
[0-9a-fA-F]{4}-[0-9a-fA-F]{4}-[0-9a-fA-F]{4}-[0-9a-fA-F]{12})-)?
(?:AI[.]Enterprise[.]Notifications[.]Messages[.]?${queue:regex}$\", queue=~\".+\\\"}
[$__rate_interval]), \"queue_name\", \"$1\", \"queue\", \"(?:[0-9a-fA-F]{8}-[0-9a-fA-
F]{4}-[0-9a-fA-F]{4}-[0-9a-fA-F]{4}-[0-9a-fA-F]{12})-)?
(?:AI[.]Enterprise[.]Notifications[.]Messages[.]?(.+\\\"))\",
        \"legendFormat\": \"redeliver {{queue_name}}\",
        \"refId\": \"B\"
    }
],
    \"title\": \"Ack / redeliver rate\",
    \"type\": \"timeseries\",
    \"description\": \"Ack and redeliver rates. Values are for queues grouped by messa
ge type: GUID and AI.Enterprise.Notifications.Messages. are stripped, and consumer qu
eues with the same message are summed.\"
},
{
    \"collapsed\": false,
    \"gridPos\": {
        \"h\": 1,
        \"w\": 24,
        \"x\": 0,
        \"y\": 23
    },
    \"id\": 101,
    \"panels\": [],
    \"title\": \"All queues\",
    \"type\": \"row\",
    \"description\": \"Original queue names (GUID + full type), one row per physical q
ueue – not grouped.\"
},
{
    \"datasource\": {
        \"type\": \"prometheus\",
        \"uid\": \"prometheus\"
    },
    \"description\": \"Original queue names (GUID + full type), one row per physical q
ueue – not grouped.\",
    \"fieldConfig\": {
        \"defaults\": {
            \"color\": {
                \"mode\": \"thresholds\"
            },
            \"custom\": {
                \"align\": \"auto\",
                \"cellOptions\": {
                    \"type\": \"auto\"
                }
            }
        }
    }
}

```

```

    },
    "inspect": false
  },
  "thresholds": {
    "mode": "absolute",
    "steps": [
      {
        "color": "green",
        "value": null
      }
    ]
  }
},
"overrides": [
  {
    "matcher": {
      "id": "byName",
      "options": "Ready"
    },
    "properties": [
      {
        "id": "thresholds",
        "value": {
          "mode": "absolute",
          "steps": [
            {
              "color": "green",
              "value": null
            },
            {
              "color": "yellow",
              "value": 100
            },
            {
              "color": "red",
              "value": 1000
            }
          ]
        }
      }
    ]
  },
  {
    "id": "custom.cellOptions",
    "value": {
      "type": "color-background"
    }
  }
]

```

```

    ]
  },
  {
    "matcher": {
      "id": "byName",
      "options": "Unacked"
    },
    "properties": [
      {
        "id": "thresholds",
        "value": {
          "mode": "absolute",
          "steps": [
            {
              "color": "green",
              "value": null
            },
            {
              "color": "yellow",
              "value": 50
            },
            {
              "color": "red",
              "value": 500
            }
          ]
        }
      }
    ]
  },
  {
    "id": "custom.cellOptions",
    "value": {
      "type": "color-background"
    }
  }
],
{
  "matcher": {
    "id": "byName",
    "options": "Consumers"
  },
  "properties": [
    {
      "id": "thresholds",
      "value": {
        "mode": "absolute",

```

```

        "steps": [
            {
                "color": "red",
                "value": null
            },
            {
                "color": "green",
                "value": 1
            }
        ]
    },
    {
        "id": "custom.cellOptions",
        "value": {
            "type": "color-background"
        }
    }
]
},
"gridPos": {
    "h": 12,
    "w": 24,
    "x": 0,
    "y": 24
},
"id": 10,
"options": {
    "cellHeight": "sm",
    "footer": {
        "countRows": false,
        "fields": "",
        "reducer": [
            "sum"
        ],
        "show": false
    },
    "showHeader": true,
    "sortBy": [
        {
            "desc": true,
            "displayName": "Total"
        }
    ]
}
]

```

```

    },
    "pluginVersion": "11.5.2",
    "targets": [
        {
            "expr": "rabbitmq_queue_messages{job=\"rabbitmq\", vhost=~\"$vhost\", queue=
=~\"(?:[0-9a-fA-F]{8}-[0-9a-fA-F]{4}-[0-9a-fA-F]{4}-[0-9a-fA-F]{4}-[0-9a-fA-F]{12})-)?
(?:AI[.]Enterprise[.]Notifications[.]Messages[.]?)${queue:regex}$\", queue=~\".+\"}\",
            "format": "table",
            "instant": true,
            "legendFormat": "Total",
            "refId": "A",
            "datasource": {
                "type": "prometheus",
                "uid": "prometheus"
            }
        },
        {
            "expr": "rabbitmq_queue_messages_ready{job=\"rabbitmq\", vhost=~\"$vhost\",
queue=~\"(?:[0-9a-fA-F]{8}-[0-9a-fA-F]{4}-[0-9a-fA-F]{4}-[0-9a-fA-F]{4}-[0-9a-fA-F]{12})-)?
(?:AI[.]Enterprise[.]Notifications[.]Messages[.]?)${queue:regex}$
\", queue=~\".+\"}\",
            "format": "table",
            "instant": true,
            "legendFormat": "Ready",
            "refId": "B",
            "datasource": {
                "type": "prometheus",
                "uid": "prometheus"
            }
        },
        {
            "expr": "rabbitmq_queue_messages_unacked{job=\"rabbitmq\", vhost=~\"$vhost\",
queue=~\"(?:[0-9a-fA-F]{8}-[0-9a-fA-F]{4}-[0-9a-fA-F]{4}-[0-9a-fA-F]{4}-[0-9a-fA-F]{12})-)?
(?:AI[.]Enterprise[.]Notifications[.]Messages[.]?)${queue:regex}$
\", queue=~\".+\"}\",
            "format": "table",
            "instant": true,
            "legendFormat": "Unacked",
            "refId": "C",
            "datasource": {
                "type": "prometheus",
                "uid": "prometheus"
            }
        }
    ]
}

```

```

        "expr": "rabbitmq_queue_consumers{job=\"rabbitmq\", vhost=~\"$vhost\", queue=\\\"(?:[0-9a-fA-F]{8}-[0-9a-fA-F]{4}-[0-9a-fA-F]{4}-[0-9a-fA-F]{4}-[0-9a-fA-F]{12})-(?:AI[.]Enterprise[.]Notifications[.]Messages[.])?${queue:regex}$\\\", queue=~\".+\\\"}\",
        \"format\": \"table\",
        \"instant\": true,
        \"legendFormat\": \"Consumers\",
        \"refId\": \"D\",
        \"datasource\": {
            \"type\": \"prometheus\",
            \"uid\": \"prometheus\"
        }
    },
    {
        \"expr\": \"sum by (queue, queue_vhost) (rate(rabbitmq_queue_messages_published_total{job=\\\"rabbitmq\\\", queue_vhost=~\\\"$vhost\\\", queue=\\\"(?:[0-9a-fA-F]{8}-[0-9a-fA-F]{4}-[0-9a-fA-F]{4}-[0-9a-fA-F]{4}-[0-9a-fA-F]{12})-(?:AI[.]Enterprise[.]Notifications[.]Messages[.])?${queue:regex}$\\\", queue=~\".+\\\"}[$__rate_interval]))\",
        \"format\": \"table\",
        \"instant\": true,
        \"legendFormat\": \"Publish/s\",
        \"refId\": \"E\",
        \"datasource\": {
            \"type\": \"prometheus\",
            \"uid\": \"prometheus\"
        }
    },
    {
        \"expr\": \"sum by (queue, vhost) (rate(rabbitmq_queue_messages_delivered_total{job=\\\"rabbitmq\\\", vhost=~\\\"$vhost\\\", queue=\\\"(?:[0-9a-fA-F]{8}-[0-9a-fA-F]{4}-[0-9a-fA-F]{4}-[0-9a-fA-F]{4}-[0-9a-fA-F]{12})-(?:AI[.]Enterprise[.]Notifications[.]Messages[.])?${queue:regex}$\\\", queue=~\".+\\\"}[$__rate_interval]))\",
        \"format\": \"table\",
        \"instant\": true,
        \"legendFormat\": \"Deliver/s\",
        \"refId\": \"F\",
        \"datasource\": {
            \"type\": \"prometheus\",
            \"uid\": \"prometheus\"
        }
    },
    {
        \"expr\": \"sum by (queue, vhost) (rate(rabbitmq_queue_messages_acked_total{job=\\\"rabbitmq\\\", vhost=~\\\"$vhost\\\", queue=\\\"(?:[0-9a-fA-F]{8}-[0-9a-fA-F]{4}-[0-9a-fA-F]{4}-[0-9a-fA-F]{4}-[0-9a-fA-F]{12})-(?:AI[.]Enterprise[.]Notifications[.]Messages[.])?${queue:regex}$\\\", queue=~\".+\\\"}[$__rate_interval]))\",

```

```

    "format": "table",
    "instant": true,
    "legendFormat": "Ack/s",
    "refId": "G",
    "datasource": {
      "type": "prometheus",
      "uid": "prometheus"
    }
  },
],
"title": "Queues",
"transformations": [
  {
    "id": "seriesToColumns",
    "options": {
      "byField": "queue"
    }
  },
  {
    "id": "organize",
    "options": {
      "excludeByName": {
        "Time": true,
        "Time 1": true,
        "Time 2": true,
        "Time 3": true,
        "Time 4": true,
        "Time 5": true,
        "Time 6": true,
        "__name__": true,
        "__name__ 1": true,
        "__name__ 2": true,
        "__name__ 3": true,
        "__name__ 4": true,
        "__name__ 5": true,
        "__name__ 6": true,
        "instance": true,
        "instance 1": true,
        "instance 2": true,
        "instance 3": true,
        "instance 4": true,
        "instance 5": true,
        "instance 6": true,
        "job": true,
        "job 1": true,
        "job 2": true,

```

```

        "job 3": true,
        "job 4": true,
        "job 5": true,
        "job 6": true,
        "node": true,
        "vhost 1": true,
        "vhost 2": true,
        "vhost 3": true,
        "vhost 4": true,
        "vhost 5": true,
        "vhost 6": true,
        "queue_vhost": true,
        "queue_vhost 1": true,
        "queue_vhost 2": true,
        "queue_name": true
    },
    "renameByName": {
        "queue": "Queue",
        "vhost": "VHost",
        "Value #A": "Total",
        "Value #B": "Ready",
        "Value #C": "Unacked",
        "Value #D": "Consumers",
        "Value #E": "Publish/s",
        "Value #F": "Deliver/s",
        "Value #G": "Ack/s"
    }
}
    },
    "type": "table"
}
],
"refresh": "15s",
"schemaVersion": 39,
"tags": [
    "rabbitmq",
    "ai-enterprise"
],
"templating": {
    "list": [
        {
            "allValue": ".*",
            "current": {
                "selected": true,
                "text": "All",

```

```

        "value": "$__all"
    },
    "datasource": {
        "type": "prometheus",
        "uid": "prometheus"
    },
    "definition": "label_values(rabbitmq_queue_messages{job=\"rabbitmq\"}, vhost)
",
    "includeAll": true,
    "label": "VHost",
    "multi": true,
    "name": "vhost",
    "options": [],
    "query": {
        "qryType": 1,
        "query": "label_values(rabbitmq_queue_messages{job=\"rabbitmq\"}, vhost)",
        "refId": "PrometheusVariableQueryEditor-VariableQuery"
    },
    "refresh": 2,
    "regex": "",
    "sort": 1,
    "type": "query"
},
{
    "allValue": ".*",
    "current": {
        "selected": true,
        "text": "All",
        "value": "$__all"
    },
    "datasource": {
        "type": "prometheus",
        "uid": "prometheus"
    },
    "definition": "label_values(rabbitmq_queue_messages{job=\"rabbitmq\", vhost=~\
\"$vhost\"}, queue)",
    "includeAll": true,
    "label": "Queue",
    "multi": true,
    "name": "queue",
    "options": [],
    "query": {
        "qryType": 1,
        "query": "label_values(rabbitmq_queue_messages{job=\"rabbitmq\", vhost=~\
vhost\"}, queue)",
        "refId": "PrometheusVariableQueryEditor-VariableQuery"
    },

```

```

    "refresh": 2,
    "regex": "/[0-9a-fA-F]{8}-[0-9a-fA-F]{4}-[0-9a-fA-F]{4}-[0-9a-fA-F]{4}-[0-9a-fA-F]{12}-AI\\\.Enterprise\\\.Notifications\\\.Messages\\\.(.+)/",
    "sort": 1,
    "type": "query"
  }
]
},
"time": {
  "from": "now-1h",
  "to": "now"
},
"timepicker": {},
"timezone": "browser",
"title": "RabbitMQ",
"uid": "rabbitmq",
"version": 1,
"weekStart": ""
}

```

7.16.3. Метрики Casgate

Casgate — сервис аутентификации и единого входа (SSO) в PT AI Enterprise Server. Метрики позволяют отслеживать производительность API аутентификации, пропускную способность и потребление ресурсов процессом Casgate.

Таблица 19. Метрики Casgate (SSO)

Метрика	Тип	Метки	Описание
casdoor_api_latency	Histogram	method, path	Распределение времени обработки запросов к API Casgate
casdoor_total_throughput	Gauge	—	Общая пропускная способность API
casdoor_cpu_usage	Gauge	—	Загрузка ЦП процессом Casgate
casdoor_memory_usage	Gauge	—	Потребление памяти процессом Casgate
go_*	Various	—	Метрики среды выполнения Go
process_*	Various	—	Системные метрики процесса (ОС)

Пример конфигурационного файла дашборда для Grafana

Ниже представлен пример конфигурационного файла, который вы можете импортировать в Grafana для отображения получаемых метрик на графиках дашборда.

```
{
  "annotations": {
    "list": [
      {
        "builtIn": 1,
        "datasource": { "type": "grafana", "uid": "-- Grafana --" },
        "enable": true,
        "hide": true,
        "iconColor": "rgba(255, 152, 0, 1)",
        "name": "Annotations & Alerts",
        "type": "dashboard"
      }
    ]
  },
  "description": "Casgate IAM – API latency, throughput, CPU/
memory and Go runtime (job=casgate).",
  "editable": true,
  "fiscalYearStartMonth": 0,
  "graphTooltip": 1,
  "id": null,
  "links": [
    {
      "icon": "external link",
      "tags": [],
      "title": "Services",
      "type": "link",
      "url": "/d/services"
    },
    {
      "icon": "external link",
      "tags": [],
      "title": "Traefik",
      "type": "link",
      "url": "/d/traefik"
    }
  ],
  "liveNow": false,
  "panels": [
    {
      "collapsed": false,
      "gridPos": { "h": 1, "w": 24, "x": 0, "y": 0 },
      "id": 100,
```

```

    "panels": [],
    "title": "Overview",
    "type": "row"
  },
  {
    "datasource": { "type": "prometheus", "uid": "prometheus" },
    "description": "Prometheus scrape target health.",
    "fieldConfig": {
      "defaults": {
        "color": { "mode": "thresholds" },
        "mappings": [
          { "options": { "0": { "color": "red", "index": 0, "text": "DOWN" } }, "ty
pe": "value" },
          { "options": { "1": { "color": "green", "index": 1, "text": "UP" } }, "ty
pe": "value" }
        ],
        "thresholds": {
          "mode": "absolute",
          "steps": [{ "color": "red", "value": null }, { "color": "green", "value":
1 } ]
        }
      },
      "overrides": []
    },
    "gridPos": { "h": 4, "w": 4, "x": 0, "y": 1 },
    "id": 1,
    "options": {
      "colorMode": "background",
      "graphMode": "none",
      "justifyMode": "center",
      "orientation": "auto",
      "reduceOptions": { "calcs": ["lastNotNull"], "fields": "", "values": false },
      "showPercentChange": false,
      "textMode": "value"
    },
    "pluginVersion": "11.5.2",
    "targets": [{ "expr": "up{job=\"casgate\"}", "legendFormat": "casgate", "refId"
: "A" } ],
    "title": "Target",
    "type": "stat"
  },
  {
    "datasource": { "type": "prometheus", "uid": "prometheus" },
    "description": "API calls per second (from latency histogram count).",
    "fieldConfig": {
      "defaults": {
        "color": { "fixedColor": "orange", "mode": "fixed" },

```

```

        "decimals": 2,
        "unit": "reqps"
    },
    "overrides": []
},
"gridPos": { "h": 4, "w": 5, "x": 4, "y": 1 },
"id": 2,
"options": {
    "colorMode": "value",
    "graphMode": "area",
    "justifyMode": "auto",
    "orientation": "auto",
    "reduceOptions": { "calcs": ["lastNotNull"], "fields": "", "values": false },
    "showPercentChange": false,
    "textMode": "auto"
},
"targets": [
    {
        "expr": "sum(rate(casdoor_api_latency_count{job=\"casgate\", path=~\"$path\\", method=~\"$method\"}[$__rate_interval]))",
        "legendFormat": "RPS",
        "refId": "A"
    }
],
"title": "API rate",
"type": "stat"
},
{
    "datasource": { "type": "prometheus", "uid": "prometheus" },
    "description": "Mean API latency across selected endpoints.",
    "fieldConfig": {
        "defaults": {
            "color": { "mode": "thresholds" },
            "decimals": 1,
            "thresholds": {
                "mode": "absolute",
                "steps": [
                    { "color": "green", "value": null },
                    { "color": "yellow", "value": 100 },
                    { "color": "red", "value": 500 }
                ]
            },
            "unit": "ms"
        },
        "overrides": []
    },

```

```

"gridPos": { "h": 4, "w": 5, "x": 9, "y": 1 },
"id": 3,
"options": {
  "colorMode": "value",
  "graphMode": "area",
  "justifyMode": "auto",
  "orientation": "auto",
  "reduceOptions": { "calcs": ["lastNotNull"], "fields": "", "values": false },
  "showPercentChange": false,
  "textMode": "auto"
},
"targets": [
  {
    "expr": "sum(rate(casdoor_api_latency_sum{job=\"casgate\", path=~\"$path\",
method=~\"$method\"}
[$__rate_interval])) / clamp_min(sum(rate(casdoor_api_latency_count{job=\"casgate\",
path=~\"$path\", method=~\"$method\"}[$__rate_interval])), 0.001)",
    "legendFormat": "avg",
    "refId": "A"
  }
],
"title": "Avg latency",
"type": "stat"
},
{
  "datasource": { "type": "prometheus", "uid": "prometheus" },
  "fieldConfig": {
    "defaults": {
      "color": { "mode": "thresholds" },
      "decimals": 1,
      "thresholds": {
        "mode": "absolute",
        "steps": [
          { "color": "green", "value": null },
          { "color": "yellow", "value": 250 },
          { "color": "red", "value": 1000 }
        ]
      }
    },
    "unit": "ms"
  },
  "overrides": []
},
"gridPos": { "h": 4, "w": 5, "x": 14, "y": 1 },
"id": 4,
"options": {
  "colorMode": "value",
  "graphMode": "area",

```

```

    "justifyMode": "auto",
    "orientation": "auto",
    "reduceOptions": { "calcs": ["lastNotNull"], "fields": "", "values": false },
    "showPercentChange": false,
    "textMode": "auto"
  },
  "targets": [
    {
      "expr": "histogram_quantile(0.95, sum by (le) (rate(casdoor_api_latency_bucket{job=\"casgate\", path=~\"$path\", method=~\"$method\"}[$__rate_interval])))",
      "legendFormat": "p95",
      "refId": "A"
    }
  ],
  "title": "p95 latency",
  "type": "stat"
},
{
  "datasource": { "type": "prometheus", "uid": "prometheus" },
  "description": "Casgate-reported heap usage vs total.",
  "fieldConfig": {
    "defaults": {
      "color": { "mode": "thresholds" },
      "decimals": 1,
      "max": 100,
      "min": 0,
      "thresholds": {
        "mode": "absolute",
        "steps": [
          { "color": "green", "value": null },
          { "color": "yellow", "value": 70 },
          { "color": "red", "value": 90 }
        ]
      }
    },
    "unit": "percent"
  },
  "overrides": []
},
{
  "gridPos": { "h": 4, "w": 5, "x": 19, "y": 1 },
  "id": 5,
  "options": {
    "colorMode": "value",
    "graphMode": "area",
    "justifyMode": "auto",
    "orientation": "auto",
    "reduceOptions": { "calcs": ["lastNotNull"], "fields": "", "values": false },

```

```

        "showPercentChange": false,
        "textMode": "auto"
    },
    "targets": [
        {
            "expr": "100 * casdoor_memory_usage{job=\"casgate\", type=\"memoryUsed\"} /
clamp_min(casdoor_memory_usage{job=\"casgate\", type=\"memoryTotal\"}, 1)",
            "legendFormat": "used %",
            "refId": "A"
        }
    ],
    "title": "Memory used",
    "type": "stat"
},
{
    "datasource": { "type": "prometheus", "uid": "prometheus" },
    "fieldConfig": {
        "defaults": {
            "color": { "mode": "palette-classic" },
            "unit": "short"
        },
        "overrides": []
    },
    "gridPos": { "h": 5, "w": 6, "x": 0, "y": 5 },
    "id": 6,
    "options": {
        "colorMode": "value",
        "graphMode": "area",
        "justifyMode": "center",
        "orientation": "auto",
        "reduceOptions": { "calcs": ["lastNotNull"], "fields": "", "values": false },
        "showPercentChange": false,
        "textMode": "auto"
    },
    "targets": [
        {
            "expr": "casdoor_total_throughput{job=\"casgate\"}",
            "legendFormat": "throughput",
            "refId": "A"
        }
    ],
    "title": "Total throughput",
    "type": "stat"
},
{
    "datasource": { "type": "prometheus", "uid": "prometheus" },

```

```

"fieldConfig": {
  "defaults": {
    "color": { "mode": "continuous-GrYlRd" },
    "max": 100,
    "min": 0,
    "unit": "percent"
  },
  "overrides": []
},
"gridPos": { "h": 5, "w": 10, "x": 6, "y": 5 },
"id": 7,
"options": {
  "displayMode": "lcd",
  "legend": { "displayMode": "list", "placement": "bottom", "showLegend": true
},
  "minVizHeight": 10,
  "minVizWidth": 0,
  "namePlacement": "auto",
  "orientation": "horizontal",
  "reduceOptions": { "calcs": ["lastNotNull"], "fields": "", "values": false },
  "showUnfilled": true,
  "valueMode": "color"
},
"targets": [
  {
    "expr": "casdoor_cpu_usage{job=\"casgate\"}",
    "legendFormat": "CPU {{cpuNum}}",
    "refId": "A"
  }
],
"title": "CPU by core",
"type": "bargauge"
},
{
  "datasource": { "type": "prometheus", "uid": "prometheus" },
  "fieldConfig": {
    "defaults": { "color": { "mode": "palette-classic" }, "unit": "bytes" },
    "overrides": []
  },
  "gridPos": { "h": 5, "w": 4, "x": 16, "y": 5 },
  "id": 8,
  "options": {
    "colorMode": "value",
    "graphMode": "none",
    "justifyMode": "auto",
    "orientation": "auto",

```

```

    "reduceOptions": { "calcs": ["lastNotNull"], "fields": "", "values": false },
    "showPercentChange": false,
    "textMode": "auto"
  },
  "targets": [
    {
      "expr": "casdoor_memory_usage{job=\"casgate\", type=\"memoryUsed\"}",
      "legendFormat": "used",
      "refId": "A"
    }
  ],
  "title": "Heap used",
  "type": "stat"
},
{
  "datasource": { "type": "prometheus", "uid": "prometheus" },
  "fieldConfig": {
    "defaults": { "color": { "mode": "palette-classic" }, "unit": "short" },
    "overrides": []
  },
  "gridPos": { "h": 5, "w": 4, "x": 20, "y": 5 },
  "id": 9,
  "options": {
    "colorMode": "value",
    "graphMode": "area",
    "justifyMode": "auto",
    "orientation": "auto",
    "reduceOptions": { "calcs": ["lastNotNull"], "fields": "", "values": false },
    "showPercentChange": false,
    "textMode": "auto"
  },
  "targets": [{ "expr": "go_goroutines{job=\"casgate\"}", "legendFormat": "gorout
ines", "refId": "A" }],
  "title": "Goroutines",
  "type": "stat"
},
{
  "collapsed": false,
  "gridPos": { "h": 1, "w": 24, "x": 0, "y": 10 },
  "id": 101,
  "panels": [],
  "title": "API performance",
  "type": "row"
},
{
  "datasource": { "type": "prometheus", "uid": "prometheus" },

```

```

"fieldConfig": {
  "defaults": {
    "color": { "mode": "palette-classic" },
    "custom": {
      "axisBorderShow": false,
      "axisCenteredZero": false,
      "axisColorMode": "text",
      "axisLabel": "",
      "axisPlacement": "auto",
      "barAlignment": 0,
      "drawStyle": "line",
      "fillOpacity": 25,
      "gradientMode": "opacity",
      "hideFrom": { "legend": false, "tooltip": false, "viz": false },
      "insertNulls": false,
      "lineInterpolation": "smooth",
      "lineWidth": 2,
      "pointSize": 5,
      "scaleDistribution": { "type": "linear" },
      "showPoints": "never",
      "spanNulls": false,
      "stacking": { "group": "A", "mode": "none" },
      "thresholdsStyle": { "mode": "off" }
    },
    "unit": "reqps"
  },
  "overrides": []
},
"gridPos": { "h": 8, "w": 12, "x": 0, "y": 11 },
"id": 10,
"options": {
  "legend": { "calcs": ["mean", "max"], "displayMode": "table", "placement": "bottom", "showLegend": true },
  "tooltip": { "mode": "multi", "sort": "desc" }
},
"targets": [
  {
    "expr": "topk(8, sum by (path) (rate(casdoor_api_latency_count{job=\"casgate\", path=~\"$path\", method=~\"$method\"}[$__rate_interval])))",
    "legendFormat": "{{path}}",
    "refId": "A"
  }
],
"title": "Request rate by path (top 8)",
"type": "timeseries"
},
{

```

```

"datasource": { "type": "prometheus", "uid": "prometheus" },
"fieldConfig": {
  "defaults": {
    "color": { "mode": "palette-classic" },
    "custom": {
      "axisBorderShow": false,
      "axisCenteredZero": false,
      "axisColorMode": "text",
      "axisLabel": "",
      "axisPlacement": "auto",
      "barAlignment": 0,
      "drawStyle": "line",
      "fillOpacity": 20,
      "gradientMode": "scheme",
      "hideFrom": { "legend": false, "tooltip": false, "viz": false },
      "insertNulls": false,
      "lineInterpolation": "smooth",
      "lineWidth": 2,
      "pointSize": 5,
      "scaleDistribution": { "type": "linear" },
      "showPoints": "never",
      "spanNulls": false,
      "stacking": { "group": "A", "mode": "none" },
      "thresholdsStyle": { "mode": "off" }
    },
    "unit": "ms"
  },
  "overrides": []
},
"gridPos": { "h": 8, "w": 12, "x": 12, "y": 11 },
"id": 11,
"options": {
  "legend": { "calcs": ["mean", "max"], "displayMode": "table", "placement": "bottom", "showLegend": true },
  "tooltip": { "mode": "multi", "sort": "desc" }
},
"targets": [
  {
    "expr": "histogram_quantile(0.50, sum by (le) (rate(casdoor_api_latency_bucket{job=\"casgate\", path=~\"$path\", method=~\"$method\"}[$__rate_interval])))",
    "legendFormat": "p50",
    "refId": "A"
  },
  {
    "expr": "histogram_quantile(0.95, sum by (le) (rate(casdoor_api_latency_bucket{job=\"casgate\", path=~\"$path\", method=~\"$method\"}[$__rate_interval])))",
    "legendFormat": "p95",
  }
]

```

```

        "refId": "B"
      },
      {
        "expr": "histogram_quantile(0.99, sum by (le) (rate(casdoor_api_latency_bucket{job=\"casgate\", path=~\"$path\", method=~\"$method\"}[ $__rate_interval ])))",
        "legendFormat": "p99",
        "refId": "C"
      }
    ],
    "title": "Latency percentiles",
    "type": "timeseries"
  },
  {
    "datasource": { "type": "prometheus", "uid": "prometheus" },
    "fieldConfig": {
      "defaults": {
        "color": { "mode": "palette-classic" },
        "custom": {
          "axisBorderShow": false,
          "axisCenteredZero": false,
          "axisColorMode": "text",
          "axisLabel": "",
          "axisPlacement": "auto",
          "barAlignment": 0,
          "drawStyle": "line",
          "fillOpacity": 15,
          "gradientMode": "opacity",
          "hideFrom": { "legend": false, "tooltip": false, "viz": false },
          "insertNulls": false,
          "lineInterpolation": "smooth",
          "lineWidth": 2,
          "pointSize": 5,
          "scaleDistribution": { "type": "linear" },
          "showPoints": "never",
          "spanNulls": false,
          "stacking": { "group": "A", "mode": "none" },
          "thresholdsStyle": { "mode": "off" }
        },
        "unit": "ms"
      },
      "overrides": []
    },
    "gridPos": { "h": 8, "w": 12, "x": 0, "y": 19 },
    "id": 12,
    "options": {
      "legend": { "calcs": ["mean", "max"], "displayMode": "table", "placement": "bottom", "showLegend": true },

```

```

        "tooltip": { "mode": "multi", "sort": "desc" }
    },
    "targets": [
        {
            "expr": "topk(8, sum by (path, method) (rate(casdoor_api_latency_sum{job=\"casgate\", path=~\"$path\", method=~\"$method\"} [$_rate_interval])) / clamp_min(sum by (path, method) (rate(casdoor_api_latency_count{job=\"casgate\", path=~\"$path\", method=~\"$method\"} [$_rate_interval])), 0.001))",
            "legendFormat": "{{method}} {{path}}",
            "refId": "A"
        }
    ],
    "title": "Avg latency by endpoint (top 8)",
    "type": "timeseries"
},
{
    "datasource": { "type": "prometheus", "uid": "prometheus" },
    "fieldConfig": {
        "defaults": {
            "color": { "mode": "palette-classic" },
            "custom": {
                "align": "auto",
                "cellOptions": { "type": "auto" },
                "inspect": false
            },
            "unit": "reqps"
        },
        "overrides": [
            {
                "matcher": { "id": "byName", "options": "Value" },
                "properties": [{ "id": "decimals", "value": 3 }]
            }
        ]
    },
    "gridPos": { "h": 8, "w": 12, "x": 12, "y": 19 },
    "id": 13,
    "options": {
        "cellHeight": "sm",
        "footer": { "countRows": false, "fields": "", "reducer": ["sum"], "show": false },
        "showHeader": true,
        "sortBy": [{ "desc": true, "displayName": "Value" }]
    },
    "pluginVersion": "11.5.2",
    "targets": [
        {

```

```

      "expr": "topk(15, sum by (path, method) (rate(casdoor_api_latency_count{job\n=\n\"casgate\\\", path=~\\\"$path\\\", method=~\\\"$method\\\"}[$__rate_interval])))",
      "format": "table",
      "instant": true,
      "legendFormat": "__auto",
      "refId": "A"
    }
  ],
  "title": "Top endpoints (current)",
  "transformations": [
    {
      "id": "organize", "options": { "excludeByName": { "Time": true }, "indexByN\name": {}, "renameByName": { "Value": "RPS", "method": "Method", "path": "Path" } } }
  ],
  "type": "table"
},
{
  "collapsed": false,
  "gridPos": { "h": 1, "w": 24, "x": 0, "y": 27 },
  "id": 102,
  "panels": [],
  "title": "Resources & runtime",
  "type": "row"
},
{
  "datasource": { "type": "prometheus", "uid": "prometheus" },
  "fieldConfig": {
    "defaults": {
      "color": { "mode": "palette-classic" },
      "custom": {
        "axisBorderShow": false,
        "axisCenteredZero": false,
        "axisColorMode": "text",
        "axisLabel": "",
        "axisPlacement": "auto",
        "barAlignment": 0,
        "drawStyle": "line",
        "fillOpacity": 30,
        "gradientMode": "opacity",
        "hideFrom": { "legend": false, "tooltip": false, "viz": false },
        "insertNulls": false,
        "lineInterpolation": "smooth",
        "lineWidth": 2,
        "pointSize": 5,
        "scaleDistribution": { "type": "linear" },
        "showPoints": "never",
        "spanNulls": false,
        "stacking": { "group": "A", "mode": "none" },

```

```

        "thresholdsStyle": { "mode": "off" }
    },
    "unit": "bytes"
},
"overrides": [
    {
        "matcher": { "id": "byName", "options": "total" },
        "properties": [{ "id": "custom.fillOpacity", "value": 0 }, { "id": "custom.lineStyle", "value": { "dash": [10, 10], "fill": "dash" } } ]
    }
],
"gridPos": { "h": 7, "w": 12, "x": 0, "y": 28 },
"id": 14,
"options": {
    "legend": { "calcs": ["mean", "max"], "displayMode": "list", "placement": "bottom", "showLegend": true },
    "tooltip": { "mode": "multi", "sort": "none" }
},
"targets": [
    {
        "expr": "casdoor_memory_usage{job=\"casgate\", type=\"memoryUsed\"}",
        "legendFormat": "used",
        "refId": "A"
    },
    {
        "expr": "casdoor_memory_usage{job=\"casgate\", type=\"memoryTotal\"}",
        "legendFormat": "total",
        "refId": "B"
    },
    {
        "expr": "process_resident_memory_bytes{job=\"casgate\"}",
        "legendFormat": "RSS (process)",
        "refId": "C"
    }
],
"title": "Memory",
"type": "timeseries"
},
{
    "datasource": { "type": "prometheus", "uid": "prometheus" },
    "fieldConfig": {
        "defaults": {
            "color": { "mode": "palette-classic" },
            "custom": {
                "axisBorderShow": false,
                "axisCenteredZero": false,

```

```

    "axisColorMode": "text",
    "axisLabel": "",
    "axisPlacement": "auto",
    "barAlignment": 0,
    "drawStyle": "line",
    "fillOpacity": 25,
    "gradientMode": "opacity",
    "hideFrom": { "legend": false, "tooltip": false, "viz": false },
    "insertNulls": false,
    "lineInterpolation": "smooth",
    "lineWidth": 2,
    "pointSize": 5,
    "scaleDistribution": { "type": "linear" },
    "showPoints": "never",
    "spanNulls": false,
    "stacking": { "group": "A", "mode": "none" },
    "thresholdsStyle": { "mode": "off" }
  },
  "unit": "percent"
},
"overrides": []
},
"gridPos": { "h": 7, "w": 12, "x": 12, "y": 28 },
"id": 15,
"options": {
  "legend": { "calcs": ["mean", "max"], "displayMode": "list", "placement": "bottom", "showLegend": true },
  "tooltip": { "mode": "multi", "sort": "none" }
},
"targets": [
  {
    "expr": "casdoor_cpu_usage{job=\"casgate\"}",
    "legendFormat": "CPU {{cpuNum}}",
    "refId": "A"
  },
  {
    "expr": "rate(process_cpu_seconds_total{job=\"casgate\"}[$__rate_interval]) * 100",
    "legendFormat": "process CPU %",
    "refId": "B"
  }
],
"title": "CPU",
"type": "timeseries"
},
{
  "datasource": { "type": "prometheus", "uid": "prometheus" },

```

```

"fieldConfig": {
  "defaults": {
    "color": { "mode": "palette-classic" },
    "custom": {
      "axisBorderShow": false,
      "axisCenteredZero": false,
      "axisColorMode": "text",
      "axisLabel": "",
      "axisPlacement": "auto",
      "barAlignment": 0,
      "drawStyle": "line",
      "fillOpacity": 20,
      "gradientMode": "opacity",
      "hideFrom": { "legend": false, "tooltip": false, "viz": false },
      "insertNulls": false,
      "lineInterpolation": "smooth",
      "lineWidth": 2,
      "pointSize": 5,
      "scaleDistribution": { "type": "linear" },
      "showPoints": "never",
      "spanNulls": false,
      "stacking": { "group": "A", "mode": "none" },
      "thresholdsStyle": { "mode": "off" }
    },
    "unit": "bytes"
  },
  "overrides": []
},
"gridPos": { "h": 7, "w": 12, "x": 0, "y": 35 },
"id": 16,
"options": {
  "legend": { "calcs": ["mean", "max"], "displayMode": "list", "placement": "bottom", "showLegend": true },
  "tooltip": { "mode": "multi", "sort": "none" }
},
"targets": [
  {
    "expr": "go_memstats_heap_inuse_bytes{job=\"casgate\"}",
    "legendFormat": "heap inuse",
    "refId": "A"
  },
  {
    "expr": "go_memstats_heap_alloc_bytes{job=\"casgate\"}",
    "legendFormat": "heap alloc",
    "refId": "B"
  }
]

```

```

],
"title": "Go heap",
"type": "timeseries"
},
{
  "datasource": { "type": "prometheus", "uid": "prometheus" },
  "fieldConfig": {
    "defaults": {
      "color": { "mode": "palette-classic" },
      "custom": {
        "axisBorderShow": false,
        "axisCenteredZero": false,
        "axisColorMode": "text",
        "axisLabel": "",
        "axisPlacement": "auto",
        "barAlignment": 0,
        "drawStyle": "line",
        "fillOpacity": 20,
        "gradientMode": "opacity",
        "hideFrom": { "legend": false, "tooltip": false, "viz": false },
        "insertNulls": false,
        "lineInterpolation": "smooth",
        "lineWidth": 2,
        "pointSize": 5,
        "scaleDistribution": { "type": "linear" },
        "showPoints": "never",
        "spanNulls": false,
        "stacking": { "group": "A", "mode": "none" },
        "thresholdsStyle": { "mode": "off" }
      },
      "unit": "short"
    },
    "overrides": []
  },
  "gridPos": { "h": 7, "w": 12, "x": 12, "y": 35 },
  "id": 17,
  "options": {
    "legend": { "calcs": ["mean", "max"], "displayMode": "list", "placement": "bottom", "showLegend": true },
    "tooltip": { "mode": "multi", "sort": "none" }
  },
  "targets": [
    { "expr": "go_goroutines{job=\"casgate\"}", "legendFormat": "goroutines", "refId": "A" },
    { "expr": "go_threads{job=\"casgate\"}", "legendFormat": "OS threads", "refId": "B" }
  ]
}

```

```

        { "expr": "process_open_fds{job=\"casgate\"}", "legendFormat": "open FDs", "refId": "C" }
      ],
      "title": "Go runtime",
      "type": "timeseries"
    }
  ],
  "refresh": "10s",
  "schemaVersion": 41,
  "style": "dark",
  "tags": ["ptai", "casgate", "iam", "prometheus"],
  "templating": {
    "list": [
      {
        "allValue": ".*",
        "current": { "selected": true, "text": "All", "value": "$__all" },
        "datasource": { "type": "prometheus", "uid": "prometheus" },
        "definition": "label_values(casdoor_api_latency_count{job=\"casgate\"}, path)",
        "hide": 0,
        "includeAll": true,
        "label": "API path",
        "multi": true,
        "name": "path",
        "options": [],
        "query": {
          "qryType": 1,
          "query": "label_values(casdoor_api_latency_count{job=\"casgate\"}, path)",
          "refId": "Prometheus-path-Variable-Query"
        },
        "refresh": 2,
        "regex": "",
        "skipUrlSync": false,
        "sort": 1,
        "type": "query"
      },
      {
        "allValue": ".*",
        "current": { "selected": true, "text": "All", "value": "$__all" },
        "datasource": { "type": "prometheus", "uid": "prometheus" },
        "definition": "label_values(casdoor_api_latency_count{job=\"casgate\"}, method)",
        "hide": 0,
        "includeAll": true,
        "label": "HTTP method",
        "multi": true,
        "name": "method",

```

```

    "options": [],
    "query": {
      "qryType": 1,
      "query": "label_values(casdoor_api_latency_count{job=\"casgate\"}, method)"
    },
    "refId": "Prometheus-method-Variable-Query"
  },
  "refresh": 2,
  "regex": "",
  "skipUrlSync": false,
  "sort": 1,
  "type": "query"
}
]
},
"time": { "from": "now-1h", "to": "now" },
"timepicker": {
  "refresh_intervals": ["5s", "10s", "30s", "1m", "5m", "15m", "30m", "1h"]
},
"timezone": "browser",
"title": "Casgate",
"uid": "casgate",
"version": 1,
"weekStart": ""
}

```

7.16.4. Метрики .NET-сервисов

Метрики доступны во всех сервисах PT AI Enterprise Server на платформе .NET: Auth, Backend, File Content, Notifications, BFF, Scan Scheduler, Swagger-агрегатор. Метрики включают статистику HTTP-запросов, gRPC-вызовов (входящих и исходящих), а также потребление ресурсов процессом и средой выполнения .NET.

Таблица 20. Метрики .NET-сервисов

Метрика	Тип	Метки	Описание
http_requests_received_total	Counter	code, method, controller, action, endpoint	Счетчик HTTP-запросов
http_request_duration_seconds	Histogram	—	Распределение времени выполнения HTTP-запросов
http_requests_in_progress	Gauge	—	Число HTTP-запросов в обработке в текущий момент

Метрика	Тип	Метки	Описание
grpc_requests_received_total	Counter	service, method, code	Счетчик входящих gRPC-вызовов на стороне сервера. Доступно для сервисов Backend, Scan Scheduler
grpc_request_duration_seconds	Histogram	—	Распределение времени выполнения входящих gRPC-вызовов на стороне сервера
grpc_client_requests_sent_total	Counter	service, method, code	Счетчик исходящих gRPC-вызовов на стороне клиента. Доступно для сервисов Backend, BFF
grpc_client_request_duration_seconds	Histogram	—	Распределение времени выполнения исходящих gRPC-вызовов на стороне клиента
process_*, dotnet_*	Various	—	Метрики процесса: загрузка ЦП, память, потоки, сборщик мусора .NET, среда выполнения

Пример конфигурационного файла дашборда для Grafana

Ниже представлен пример конфигурационного файла, который вы можете импортировать в Grafana для отображения получаемых метрик на графиках дашборда.

```
{
  "annotations": {
    "list": [
      {
        "builtIn": 1,
        "datasource": { "type": "grafana", "uid": "-- Grafana --" },
        "enable": true,
        "hide": true,
        "iconColor": "rgba(0, 211, 255, 1)",
        "name": "Annotations & Alerts",
        "type": "dashboard"
      }
    ]
  },
  "editable": true,
  "fiscalYearStartMonth": 0,
```

```

"graphTooltip": 1,
"id": null,
"links": [],
"liveNow": false,
"panels": [
  {
    "collapsed": false,
    "gridPos": { "h": 1, "w": 24, "x": 0, "y": 0 },
    "id": 100,
    "panels": [],
    "title": "Overview (RED)",
    "type": "row"
  },
  {
    "datasource": { "type": "prometheus", "uid": "prometheus" },
    "description": "Target scrape health (1 = up).",
    "fieldConfig": {
      "defaults": {
        "color": { "mode": "thresholds" },
        "mappings": [
          { "options": { "0": { "color": "red", "index": 0, "text": "DOWN" } }, "type": "value" },
          { "options": { "1": { "color": "green", "index": 1, "text": "UP" } }, "type": "value" }
        ],
        "thresholds": { "mode": "absolute", "steps": [ { "color": "red", "value": null }, { "color": "green", "value": 1 } ] },
        "unit": "none"
      },
      "overrides": []
    },
    "gridPos": { "h": 4, "w": 4, "x": 0, "y": 1 },
    "id": 1,
    "options": { "colorMode": "background", "graphMode": "none", "justifyMode": "center", "orientation": "auto", "reduceOptions": { "calcs": ["lastNotNull"], "fields": "", "values": false }, "showPercentChange": false, "textMode": "value" },
    "pluginVersion": "11.5.2",
    "targets": [ { "datasource": { "type": "prometheus", "uid": "prometheus" }, "expr": "up{job=~\"$service\"}", "legendFormat": "{{job}}", "refId": "A" } ],
    "title": "Up",
    "type": "stat"
  },
  {
    "datasource": { "type": "prometheus", "uid": "prometheus" },
    "description": "Total HTTP request rate (all status codes).",
    "fieldConfig": {
      "defaults": { "color": { "mode": "palette-classic" }, "unit": "reqps" },

```

```

    "overrides": []
  },
  "gridPos": { "h": 4, "w": 5, "x": 4, "y": 1 },
  "id": 2,
  "options": { "colorMode": "value", "graphMode": "area", "justifyMode": "auto",
"orientation": "auto", "reduceOptions": { "calcs": ["lastNotNull"], "fields": "", "va
lues": false }, "showPercentChange": false, "textMode": "auto" },
  "targets": [{ "expr": "sum(rate(http_requests_received_total{job=~\"$service\\\"}
[$__rate_interval]))", "legendFormat": "RPS", "refId": "A" }],
  "title": "Request rate",
  "type": "stat"
},
{
  "datasource": { "type": "prometheus", "uid": "prometheus" },
  "description": "Share of 5xx responses (errors).",
  "fieldConfig": {
    "defaults": {
      "color": { "mode": "thresholds" },
      "max": 1,
      "min": 0,
      "thresholds": { "mode": "absolute", "steps": [{ "color": "green", "value":
null }, { "color": "yellow", "value": 0.01 }, { "color": "red", "value": 0.05 } ] },
      "unit": "percentunit"
    },
    "overrides": []
  },
  "gridPos": { "h": 4, "w": 5, "x": 9, "y": 1 },
  "id": 3,
  "options": { "colorMode": "value", "graphMode": "area", "justifyMode": "auto",
"orientation": "auto", "reduceOptions": { "calcs": ["lastNotNull"], "fields": "", "va
lues": false }, "showPercentChange": false, "textMode": "auto" },
  "targets": [{
    "expr": "sum(rate(http_requests_received_total{job=~\"$service\\\", code=~\"5..
\\\"}[$__rate_interval])) / sum(rate(http_requests_received_total{job=~\"$service\\\"}
[$__rate_interval]))",
    "legendFormat": "5xx ratio",
    "refId": "A"
  }],
  "title": "Error rate (5xx)",
  "type": "stat"
},
{
  "datasource": { "type": "prometheus", "uid": "prometheus" },
  "fieldConfig": { "defaults": { "color": { "mode": "palette-
classic" }, "unit": "s" }, "overrides": [] },
  "gridPos": { "h": 4, "w": 5, "x": 14, "y": 1 },
  "id": 4,

```

```

      "options": { "colorMode": "value", "graphMode": "area", "justifyMode": "auto",
"orientation": "auto", "reduceOptions": { "calcs": ["lastNotNull"], "fields": "", "va
lues": false }, "showPercentChange": false, "textMode": "auto" },
      "targets": [{
        "expr": "histogram_quantile(0.50, sum by (le) (rate(http_request_duration_sec
onds_bucket{job=~\"$service\"}[$__rate_interval])))",
        "legendFormat": "p50",
        "refId": "A"
      }],
      "title": "Latency p50",
      "type": "stat"
    },
    {
      "datasource": { "type": "prometheus", "uid": "prometheus" },
      "fieldConfig": { "defaults": { "color": { "mode": "palette-
classic" }, "unit": "s" }, "overrides": [] },
      "gridPos": { "h": 4, "w": 5, "x": 19, "y": 1 },
      "id": 5,
      "options": { "colorMode": "value", "graphMode": "area", "justifyMode": "auto",
"orientation": "auto", "reduceOptions": { "calcs": ["lastNotNull"], "fields": "", "va
lues": false }, "showPercentChange": false, "textMode": "auto" },
      "targets": [{
        "expr": "histogram_quantile(0.99, sum by (le) (rate(http_request_duration_sec
onds_bucket{job=~\"$service\"}[$__rate_interval])))",
        "legendFormat": "p99",
        "refId": "A"
      }],
      "title": "Latency p99",
      "type": "stat"
    },
    {
      "collapsed": false,
      "gridPos": { "h": 1, "w": 24, "x": 0, "y": 5 },
      "id": 101,
      "panels": [],
      "title": "http",
      "type": "row"
    },
    {
      "datasource": { "type": "prometheus", "uid": "prometheus" },
      "description": "REST/HTTP API traffic (controller label is set).",
      "fieldConfig": {
        "defaults": {
          "color": { "mode": "palette-classic" },
          "custom": { "axisBorderShow": false, "axisCenteredZero": false, "axisColorM
ode": "text", "axisLabel": "", "axisPlacement": "auto", "barAlignment": 0, "drawStyle
": "line", "fillOpacity": 25, "gradientMode": "opacity", "hideFrom": { "legend": fals
e, "tooltip": false, "viz": false }, "insertNulls": false, "lineInterpolation": "smoo

```

```

th", "lineWidth": 2, "pointSize": 5, "scaleDistribution": { "type": "linear" }, "show
Points": "never", "spanNulls": false, "stacking": { "group": "A", "mode": "normal" },
"thresholdsStyle": { "mode": "off" } },
    "unit": "reqps"
},
"overrides": [
    { "matcher": { "id": "byRegex", "options": "/"
^2.*/" }, "properties": [{ "id": "color", "value": { "fixedColor": "green", "mode": "
fixed" } }] },
    { "matcher": { "id": "byRegex", "options": "/"
^3.*/" }, "properties": [{ "id": "color", "value": { "fixedColor": "blue", "mode": "f
ixed" } }] },
    { "matcher": { "id": "byRegex", "options": "/"
^4.*/" }, "properties": [{ "id": "color", "value": { "fixedColor": "yellow", "mode":
"fixed" } }] },
    { "matcher": { "id": "byRegex", "options": "/"
^5.*/" }, "properties": [{ "id": "color", "value": { "fixedColor": "red", "mode": "fi
xed" } }] }
]
},
"gridPos": { "h": 8, "w": 12, "x": 0, "y": 6 },
"id": 10,
"options": { "legend": { "calcs": ["mean", "max"], "displayMode": "table", "pla
cement": "bottom", "showLegend": true }, "tooltip": { "mode": "multi", "sort": "desc"
} },
"targets": [{
    "expr": "sum by (code) (rate(http_requests_received_total{job=~\"$service\",
controller=~\".+\"}[$_rate_interval]))",
    "legendFormat": "{{code}}",
    "refId": "A"
}],
"title": "HTTP – RPS by status code",
"type": "timeseries"
},
{
    "datasource": { "type": "prometheus", "uid": "prometheus" },
    "fieldConfig": {
        "defaults": {
            "color": { "mode": "palette-classic" },
            "custom": { "axisBorderShow": false, "axisCenteredZero": false, "axisColorM
ode": "text", "axisLabel": "", "axisPlacement": "auto", "barAlignment": 0, "drawStyle
": "line", "fillOpacity": 10, "gradientMode": "none", "hideFrom": { "legend": false,
"tooltip": false, "viz": false }, "insertNulls": false, "lineInterpolation": "smooth"
, "lineWidth": 2, "pointSize": 5, "scaleDistribution": { "type": "linear" }, "showPoi
nts": "never", "spanNulls": false, "stacking": { "group": "A", "mode": "none" }, "thr
esholdsStyle": { "mode": "off" } },
            "unit": "s"
        },
        "overrides": []
    },

```

```

    "gridPos": { "h": 8, "w": 12, "x": 12, "y": 6 },
    "id": 11,
    "options": { "legend": { "calcs": ["mean", "max"], "displayMode": "table", "placement": "bottom", "showLegend": true }, "tooltip": { "mode": "multi", "sort": "desc" } },
    "targets": [
      { "expr": "histogram_quantile(0.50, sum by (le) (rate(http_request_duration_seconds_bucket{job=~\"$service\", controller=~\".+\"}[$__rate_interval])))", "legendFormat": "p50", "refId": "A" },
      { "expr": "histogram_quantile(0.95, sum by (le) (rate(http_request_duration_seconds_bucket{job=~\"$service\", controller=~\".+\"}[$__rate_interval])))", "legendFormat": "p95", "refId": "B" },
      { "expr": "histogram_quantile(0.99, sum by (le) (rate(http_request_duration_seconds_bucket{job=~\"$service\", controller=~\".+\"}[$__rate_interval])))", "legendFormat": "p99", "refId": "C" }
    ],
    "title": "HTTP – latency percentiles",
    "type": "timeseries"
  },
  {
    "collapsed": false,
    "gridPos": { "h": 1, "w": 24, "x": 0, "y": 14 },
    "id": 102,
    "panels": [],
    "title": "grpc",
    "type": "row"
  },
  {
    "datasource": { "type": "prometheus", "uid": "prometheus" },
    "description": "gRPC server request rate by status code (grpc_requests_received_total).",
    "fieldConfig": {
      "defaults": {
        "color": { "mode": "palette-classic" },
        "custom": { "axisBorderShow": false, "axisCenteredZero": false, "axisColorMode": "text", "axisLabel": "", "axisPlacement": "auto", "barAlignment": 0, "drawStyle": "line", "fillOpacity": 25, "gradientMode": "opacity", "hideFrom": { "legend": false, "tooltip": false, "viz": false }, "insertNulls": false, "lineInterpolation": "smooth", "lineWidth": 2, "pointSize": 5, "scaleDistribution": { "type": "linear" }, "showPoints": "never", "spanNulls": false, "stacking": { "group": "A", "mode": "normal" }, "thresholdsStyle": { "mode": "off" } },
        "unit": "reqps"
      },
      "overrides": [
        { "matcher": { "id": "byRegexp", "options": "/OK/" }, "properties": [ { "id": "color", "value": { "fixedColor": "green", "mode": "fixed" } } ] },
      ]
    }
  }
]

```

```

        { "matcher": { "id": "byRegexp", "options": "/InvalidArgument|NotFound|
AlreadyExists|PermissionDenied|Unauthenticated|FailedPrecondition|ResourceExhausted|
Cancelled/" }, "properties": [{ "id": "color", "value": { "fixedColor": "yellow", "mo
de": "fixed" } } ] },
        { "matcher": { "id": "byRegexp", "options": "/Internal|Unknown|Unavailable|
DataLoss|
DeadlineExceeded/" }, "properties": [{ "id": "color", "value": { "fixedColor": "red",
"mode": "fixed" } } ] }
    ]
},
"gridPos": { "h": 8, "w": 12, "x": 0, "y": 15 },
"id": 20,
"options": { "legend": { "calcs": ["mean", "max"], "displayMode": "table", "pla
cement": "bottom", "showLegend": true }, "tooltip": { "mode": "multi", "sort": "desc"
} },
"targets": [{
    "expr": "sum by (code) (rate(grpc_requests_received_total{job=~\"$service\"}
[$__rate_interval]))",
    "legendFormat": "{{code}}",
    "refId": "A"
}],
"title": "gRPC - RPS by status",
"type": "timeseries"
},
{
    "datasource": { "type": "prometheus", "uid": "prometheus" },
    "description": "gRPC server latency percentiles (grpc_request_duration_seconds)
.",
    "fieldConfig": {
        "defaults": {
            "color": { "mode": "palette-classic" },
            "custom": { "axisBorderShow": false, "axisCenteredZero": false, "axisColorM
ode": "text", "axisLabel": "", "axisPlacement": "auto", "barAlignment": 0, "drawStyle
": "line", "fillOpacity": 10, "gradientMode": "none", "hideFrom": { "legend": false,
"tooltip": false, "viz": false }, "insertNulls": false, "lineInterpolation": "smooth"
, "lineWidth": 2, "pointSize": 5, "scaleDistribution": { "type": "linear" }, "showPoi
nts": "never", "spanNulls": false, "stacking": { "group": "A", "mode": "none" }, "thr
esholdsStyle": { "mode": "off" } },
            "unit": "s"
        },
        "overrides": []
    },
    "gridPos": { "h": 8, "w": 12, "x": 12, "y": 15 },
    "id": 21,
    "options": { "legend": { "calcs": ["mean", "max"], "displayMode": "table", "pla
cement": "bottom", "showLegend": true }, "tooltip": { "mode": "multi", "sort": "desc"
} },
    "targets": [

```

```

        { "expr": "histogram_quantile(0.50, sum by (le) (rate(grpc_request_duration_s
econds_bucket{job=~\"$service\"}
[ $__rate_interval ])))", "legendFormat": "p50", "refId": "A" },
        { "expr": "histogram_quantile(0.95, sum by (le) (rate(grpc_request_duration_s
econds_bucket{job=~\"$service\"}
[ $__rate_interval ])))", "legendFormat": "p95", "refId": "B" },
        { "expr": "histogram_quantile(0.99, sum by (le) (rate(grpc_request_duration_s
econds_bucket{job=~\"$service\"}
[ $__rate_interval ])))", "legendFormat": "p99", "refId": "C" }
    ],
    "title": "gRPC – latency percentiles",
    "type": "timeseries"
},
{
    "collapsed": false,
    "gridPos": { "h": 1, "w": 24, "x": 0, "y": 23 },
    "id": 103,
    "panels": [],
    "title": "Process & .NET runtime",
    "type": "row"
},
{
    "datasource": { "type": "prometheus", "uid": "prometheus" },
    "fieldConfig": {
        "defaults": {
            "color": { "mode": "palette-classic" },
            "custom": { "axisBorderShow": false, "axisCenteredZero": false, "axisColorM
ode": "text", "axisLabel": "", "axisPlacement": "auto", "barAlignment": 0, "drawStyle
": "line", "fillOpacity": 20, "gradientMode": "opacity", "hideFrom": { "legend": fals
e, "tooltip": false, "viz": false }, "insertNulls": false, "lineInterpolation": "smoo
th", "lineWidth": 2, "pointSize": 5, "scaleDistribution": { "type": "linear" }, "show
Points": "never", "spanNulls": false, "stacking": { "group": "A", "mode": "none" }, "
thresholdsStyle": { "mode": "off" } },
            "unit": "short",
            "decimals": 2
        },
        "overrides": []
    },
    "gridPos": { "h": 8, "w": 8, "x": 0, "y": 24 },
    "id": 30,
    "options": { "legend": { "calcs": ["mean", "max"], "displayMode": "list", "plac
ement": "bottom", "showLegend": true }, "tooltip": { "mode": "multi", "sort": "desc"
} },
    "targets": [{
        "expr": "rate(process_cpu_seconds_total{job=~\"$service\"}
[ $__rate_interval ])",
        "legendFormat": "{{job}} CPU cores",
        "refId": "A"
    }],

```

```

    "title": "CPU usage (cores)",
    "type": "timeseries"
  },
  {
    "datasource": { "type": "prometheus", "uid": "prometheus" },
    "fieldConfig": {
      "defaults": {
        "color": { "mode": "palette-classic" },
        "custom": { "axisBorderShow": false, "axisCenteredZero": false, "axisColorM
ode": "text", "axisLabel": "", "axisPlacement": "auto", "barAlignment": 0, "drawStyle
": "line", "fillOpacity": 20, "gradientMode": "opacity", "hideFrom": { "legend": fals
e, "tooltip": false, "viz": false }, "insertNulls": false, "lineInterpolation": "smoo
th", "lineWidth": 2, "pointSize": 5, "scaleDistribution": { "type": "linear" }, "show
Points": "never", "spanNulls": false, "stacking": { "group": "A", "mode": "none" }, "
thresholdsStyle": { "mode": "off" } },
        "unit": "bytes"
      },
      "overrides": []
    },
    "gridPos": { "h": 8, "w": 8, "x": 8, "y": 24 },
    "id": 31,
    "options": { "legend": { "calcs": ["mean", "max"], "displayMode": "list", "plac
ement": "bottom", "showLegend": true }, "tooltip": { "mode": "multi", "sort": "desc"
} },
    "targets": [
      { "expr": "process_working_set_bytes{job=~\"$service\"}", "legendFormat": "{
job}} working set", "refId": "A" },
      { "expr": "process_private_memory_bytes{job=~\"$service\"}", "legendFormat":
"{{job}} private", "refId": "B" },
      { "expr": "dotnet_total_memory_bytes{job=~\"$service\"}", "legendFormat": "{
job}} dotnet total", "refId": "C" }
    ],
    "title": "Memory",
    "type": "timeseries"
  },
  {
    "datasource": { "type": "prometheus", "uid": "prometheus" },
    "fieldConfig": {
      "defaults": {
        "color": { "mode": "palette-classic" },
        "custom": { "axisBorderShow": false, "axisCenteredZero": false, "axisColorM
ode": "text", "axisLabel": "", "axisPlacement": "auto", "barAlignment": 0, "drawStyle
": "line", "fillOpacity": 15, "gradientMode": "none", "hideFrom": { "legend": false,
"tooltip": false, "viz": false }, "insertNulls": false, "lineInterpolation": "smooth"
, "lineWidth": 2, "pointSize": 5, "scaleDistribution": { "type": "linear" }, "showPoi
nts": "never", "spanNulls": false, "stacking": { "group": "A", "mode": "none" }, "thr
esholdsStyle": { "mode": "off" } },
        "unit": "ops"
      },
      "overrides": []
    },

```

```

    },
    "gridPos": { "h": 8, "w": 8, "x": 16, "y": 24 },
    "id": 32,
    "options": { "legend": { "calcs": ["mean", "max"], "displayMode": "list", "placement": "bottom", "showLegend": true }, "tooltip": { "mode": "multi", "sort": "desc" } },
    "targets": [{
      "expr": "rate(dotnet_collection_count_total{job=~\"$service\"}[$__rate_interval])",
      "legendFormat": "{{job}} gen {{generation}}",
      "refId": "A"
    }],
    "title": "GC collections / sec",
    "type": "timeseries"
  },
  {
    "datasource": { "type": "prometheus", "uid": "prometheus" },
    "fieldConfig": {
      "defaults": {
        "color": { "mode": "palette-classic" },
        "custom": { "axisBorderShow": false, "axisCenteredZero": false, "axisColorMode": "text", "axisLabel": "", "axisPlacement": "auto", "barAlignment": 0, "drawStyle": "line", "fillOpacity": 10, "gradientMode": "none", "hideFrom": { "legend": false, "tooltip": false, "viz": false }, "insertNulls": false, "lineInterpolation": "smooth", "lineWidth": 2, "pointSize": 5, "scaleDistribution": { "type": "linear" }, "showPoints": "never", "spanNulls": false, "stacking": { "group": "A", "mode": "none" }, "thresholdsStyle": { "mode": "off" } },
        "unit": "short"
      },
      "overrides": []
    },
    "gridPos": { "h": 8, "w": 12, "x": 0, "y": 32 },
    "id": 33,
    "options": { "legend": { "calcs": ["mean", "max"], "displayMode": "list", "placement": "bottom", "showLegend": true }, "tooltip": { "mode": "multi", "sort": "desc" } },
    "targets": [
      { "expr": "process_num_threads{job=~\"$service\"}", "legendFormat": "{{job}} threads", "refId": "A" },
      { "expr": "process_open_handles{job=~\"$service\"}", "legendFormat": "{{job}} handles", "refId": "B" }
    ],
    "title": "Threads & handles",
    "type": "timeseries"
  },
  {
    "datasource": { "type": "prometheus", "uid": "prometheus" },
    "fieldConfig": {
      "defaults": {

```

```

        "color": { "mode": "palette-classic" },
        "custom": { "axisBorderShow": false, "axisCenteredZero": false, "axisColorM
ode": "text", "axisLabel": "", "axisPlacement": "auto", "barAlignment": 0, "drawStyle
": "line", "fillOpacity": 25, "gradientMode": "opacity", "hideFrom": { "legend": fals
e, "tooltip": false, "viz": false }, "insertNulls": false, "lineInterpolation": "smoo
th", "lineWidth": 2, "pointSize": 5, "scaleDistribution": { "type": "linear" }, "show
Points": "never", "spanNulls": false, "stacking": { "group": "A", "mode": "normal" },
"thresholdsStyle": { "mode": "off" } },
        "unit": "reqps"
    },
    "overrides": []
},
"gridPos": { "h": 8, "w": 12, "x": 12, "y": 32 },
"id": 34,
"options": { "legend": { "calcs": ["mean", "max"], "displayMode": "table", "pla
cement": "bottom", "showLegend": true }, "tooltip": { "mode": "multi", "sort": "desc"
} },
"targets": [{
    "expr": "sum by (method) (rate(http_requests_received_total{job=~\"$service\"
}[$__rate_interval]))",
    "legendFormat": "{{method}}",
    "refId": "A"
}],
"title": "All traffic – RPS by HTTP method",
"type": "timeseries"
}
],
"refresh": "10s",
"schemaVersion": 39,
"tags": ["ai", "dotnet", "prometheus", "red"],
"templating": {
    "list": [
        {
            "current": { "selected": true, "text": "backend", "value": "backend" },
            "datasource": { "type": "prometheus", "uid": "prometheus" },
            "definition": "label_values(up{job=~\"backend|auth|file-content|
notifications|scan-scheduler|backend-for-frontend|swagger\"}, job)",
            "hide": 0,
            "includeAll": true,
            "allValue": "backend|auth|file-content|notifications|scan-scheduler|backend-
for-frontend|swagger",
            "label": "Service",
            "multi": false,
            "name": "service",
            "options": [],
            "query": {
                "qryType": 1,
                "query": "label_values(up{job=~\"backend|auth|file-content|notifications|
scan-scheduler|backend-for-frontend|swagger\"}, job)",
            }
        }
    ]
}

```

```

        "refId": "PrometheusVariableQueryEditor-VariableQuery"
    },
    "refresh": 2,
    "regex": "",
    "skipUrlSync": false,
    "sort": 1,
    "type": "query"
}
]
},
"time": { "from": "now-1h", "to": "now" },
"timepicker": { "refresh_intervals": ["5s", "10s", "30s", "1m", "5m"] },
"timezone": "browser",
"title": "Services",
"uid": "services",
"version": 1,
"weekStart": ""
}

```

8. Интеграция с внешними системами

Раздел содержит инструкции по интеграции PT AI Enterprise Edition со средами разработки, сервисами CI/CD, а также системой отслеживания ошибок Jira.

В этом разделе

[Интеграция PT AI Enterprise Edition в сборочный процесс \(см. раздел 8.1\)](#)

[Интеграция PT AI Enterprise Edition со средами разработки \(см. раздел 8.2\)](#)

[Интеграция с системой отслеживания ошибок Jira \(см. раздел 8.3\)](#)

8.1. Интеграция PT AI Enterprise Edition в сборочный процесс

Раздел содержит рекомендации по внедрению анализа кода в сборочный процесс систем Jenkins, TeamCity и других с помощью PT AI Enterprise Edition.

В этом разделе

[Об интеграции PT AI Enterprise Edition в сборочный процесс \(см. раздел 8.1.1\)](#)

[Установка легкого агента AI.Shell \(см. раздел 8.1.2\)](#)

[Конфигурирование AI.Shell и запуск сканирования \(см. раздел 8.1.3\)](#)

[Настройка параметров сканирования в конфигурационном файле \(см. раздел 8.1.4\)](#)

[Сборка плагинов \(см. раздел 8.1.5\)](#)

[Интеграция с Jenkins \(см. раздел 8.1.6\)](#)

[Интеграция с TeamCity \(см. раздел 8.1.7\)](#)

[Интеграция с другими CI-системами \(см. раздел 8.1.8\)](#)

[Файл описания отчетов \(см. раздел 8.1.9\)](#)

8.1.1. Об интеграции PT AI Enterprise Edition в сборочный процесс

Разработка приложения состоит из нескольких этапов: написание и загрузка кода в репозиторий, сборка приложения, его тестирование и публикация. Если вы разрабатываете одно или несколько приложений, эти этапы можно выполнять вручную без больших усилий. Однако, если количество приложений и число релизов постоянно растут, то сборка и тестирование могут отнимать значительную часть времени ваших сотрудников. Для решения этой проблемы используется концепция непрерывной интеграции и доставки (англ. continuous integration/continuous delivery, далее также — CI/CD). Она позволяет автоматизировать

процессы сборки, тестирования и доставки новых модулей проекта разработчикам, аналитикам, инженерам по качеству, конечным пользователям и другим заинтересованным лицам. Концепция CI/CD является частью практики DevSecOps, которая охватывает более широкий аспект разработки приложений по методологии Agile.

SEC

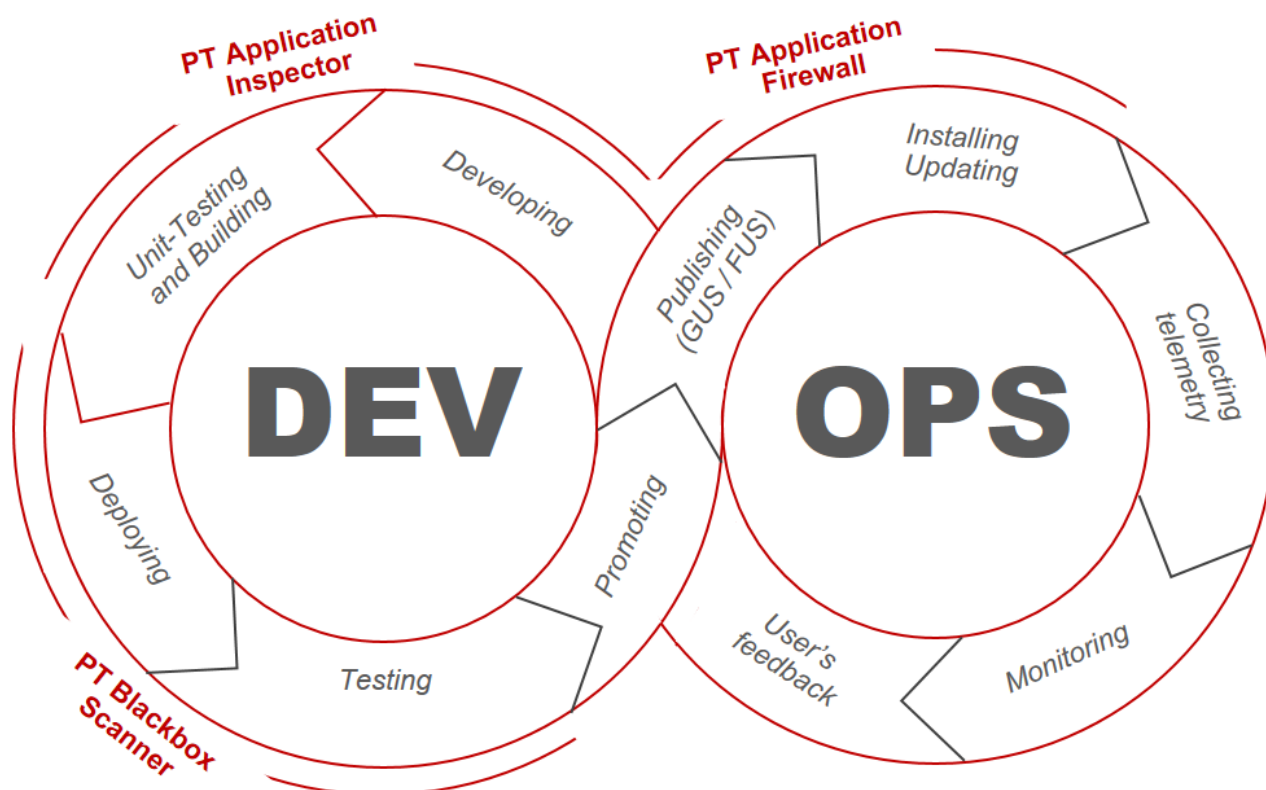


Рисунок 1. Этапы процесса разработки в концепции CI/CD

Концепция CI/CD реализуется при помощи специальных систем, наиболее популярными из которых являются GitLab CI, Jenkins и TeamCity. Эти системы состоят из центрального сервера и агентов сборки, число которых может масштабироваться под изменяющуюся нагрузку. Некоторые системы поддерживают установку плагинов, позволяющих сторонним разработчикам интегрировать свои решения, добавляя новые возможности в процесс сборки. В системах, не поддерживающих плагины, интеграция осуществляется путем вызова оболочки командной строки и запуска в ней исполняемых файлов с передачей требуемых параметров.

Для внедрения анализа кода с помощью PT AI Enterprise Edition в сборочный процесс используются следующие плагины:

- Al.Jenkins.Plugin — плагин для системы Jenkins;
- Al.TeamCity.Plugin — плагин для системы TeamCity;
- Al.Cli.Plugin — кросс-платформенный Java-плагин для запуска из командной строки.

Плагины осуществляют полный цикл взаимодействия с PT AI Enterprise Server: авторизацию, передачу исходного кода, постановку задачи сканирования, отслеживание прогресса ее выполнения и загрузку полученных результатов. На этапе постановки задачи сканирования PT AI Enterprise Server добавляет ее в очередь, а затем передает на выполнение первому освободившемуся агенту сканирования.

Шаги сборки запускаются в последовательности, указанной в параметрах настройки проекта CI/CD-системы. Одним из таких шагов является вызов плагина для анализа кода. В общем случае плагины запускаются как отдельный модуль на агенте сборки CI/CD-системы. Схема взаимодействия PT AI Enterprise Edition с DevOps-инфраструктурой представлена на рисунке ниже.

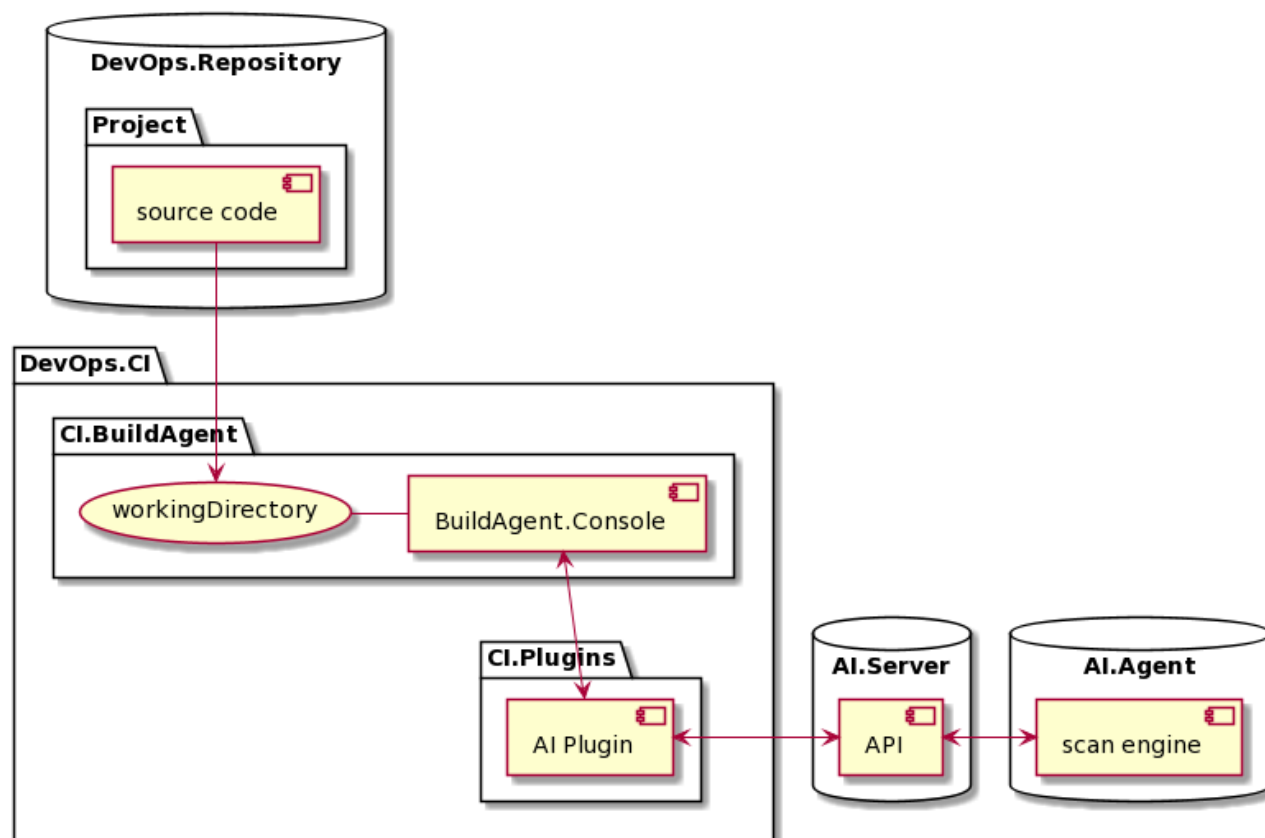


Рисунок 2. Размещение PT AI Enterprise Edition в DevOps-инфраструктуре

Таблица 21. Компоненты на схеме

Название	Описание
DevOps.Repository	Корпоративное хранилище с исходным кодом для сканирования
DevOps.CI	Корпоративная CI-система, в которой запускаются сборки
CI.BuildAgent	Сборочный агент CI-системы. Получает исходный код и предоставляет его для использования плагинам
BuildAgent.Console	Системная консоль сборочного агента
CI.Plugins	Плагины CI-системы
AI.Plugin	Плагин PT AI Enterprise Edition. Отправляет исходный код в PT AI Enterprise Server, отслеживает прогресс и скачивает отчеты
AI.Server	Модуль PT AI Enterprise Server. Принимает исходный код, ставит задачу сканирования модулю PT AI Enterprise Agent и сохраняет результаты
AI.Agent	Модуль PT AI Enterprise Agent. Принимает задачу сканирования от PT AI Enterprise Server и возвращает результаты

Для эффективного выявления уязвимостей с использованием методов статического анализа необходимо обеспечить выполнение следующих условий:

- доступность сторонних компонентов, используемых приложением;
- синтаксическую корректность кода, обеспечивающую возможность его сборки.

Исходя из этих условий, шаг анализа кода в сборочном процессе должен выполняться после завершения задач сборки, как правило, включающих в себя выполнение модульных тестов. В следующих разделах будут представлены примеры сборочных процессов с учетом данных требований.

Добавление нового шага в сборочный процесс увеличивает общее время сборки и анализ кода не является исключением. Очевидно, что длительность шага анализа кода напрямую зависит от того, будет ли сборочный конвейер дожидаться завершения анализа или ограничится запуском задачи сканирования и вернет управление в конвейер. Первый вариант оправдан в случае важных сборок, например, перед выпуском нового релиза продукта, когда необходимо предотвратить попадание уязвимого кода в сборку. В плагинах PT AI Enterprise Edition такой режим работы называется синхронным.

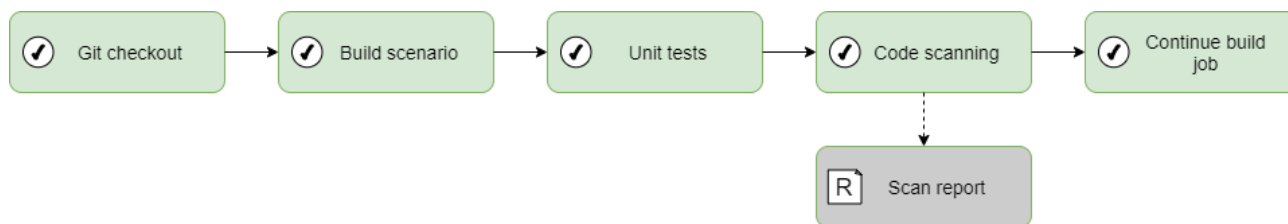


Рисунок 3. Пример сборочного процесса с анализом кода в синхронном режиме

Однако, в повседневных сборках тестовых версий продукта, а также при анализе мердж-реквестов такая задержка приводит к замедлению разработки продукта. Чтобы избежать подобных ситуаций, в плагинах PT AI Enterprise Edition предусмотрена возможность асинхронного запуска, когда плагин передает задачу анализа кода в PT AI Enterprise Server и прекращает свою работу, не дожидаясь завершения задачи, тем самым не увеличивая общее время сборки.

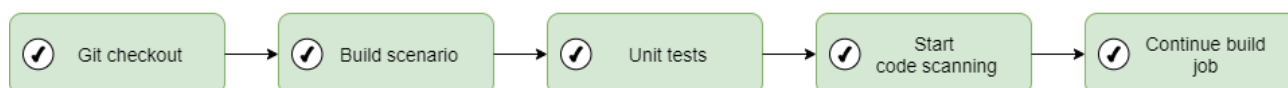


Рисунок 4. Пример сборочного процесса с анализом кода в асинхронном режиме

При асинхронном запуске плагина получение отчетов о сканировании в CI/CD-системе недоступно, но вы можете просмотреть результаты анализа кода в веб-интерфейсе PT AI Enterprise Edition, либо настроить отправку отчетов на электронную почту. Чтобы не терять преимущества асинхронного режима и получать отчеты в виде артефактов сборки, вы можете реализовать асинхронность средствами CI/CD-системы. Для этого необходимо запустить плагин в синхронном режиме в отдельном конвейере параллельно с основным сборочным процессом.

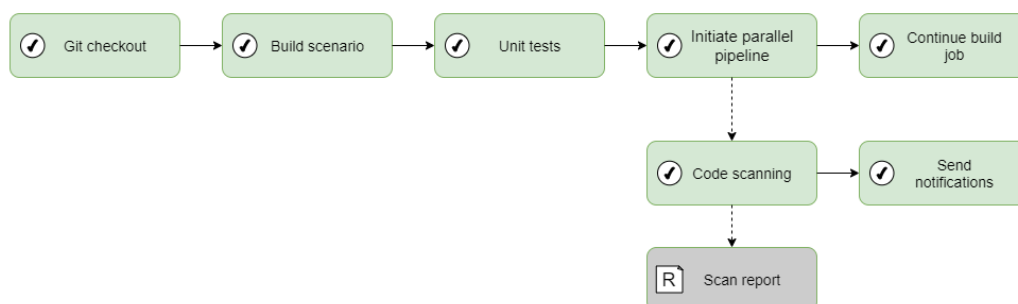


Рисунок 5. Пример сборочного процесса с анализом кода, выполняемым параллельно со сборкой

При синхронном запуске плагина вы можете настраивать политику безопасности (англ. security gates), которая позволяет останавливать сборку при несоблюдении заданных условий. Например, если обнаружена одна или более уязвимостей высокого уровня опасности, плагин завершит свою работу с ошибкой, которая заблокирует дальнейшие шаги сборки или мердж-реквесты. Для продуктовых сборок это поможет предотвратить попадание уязвимой версии приложения на веб-сервер.

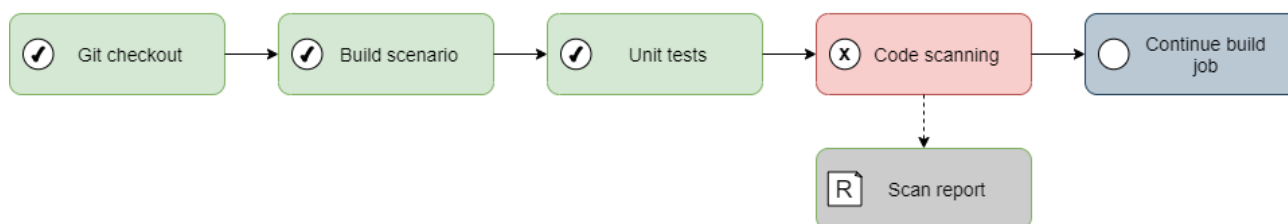


Рисунок 6. Пример блокировки сборочного процесса при нарушении политики безопасности

При запуске плагина параллельно с основным сборочным процессом нарушения политики безопасности обрабатываются отдельно, что позволяет, например, блокировать мердж-реквесты.

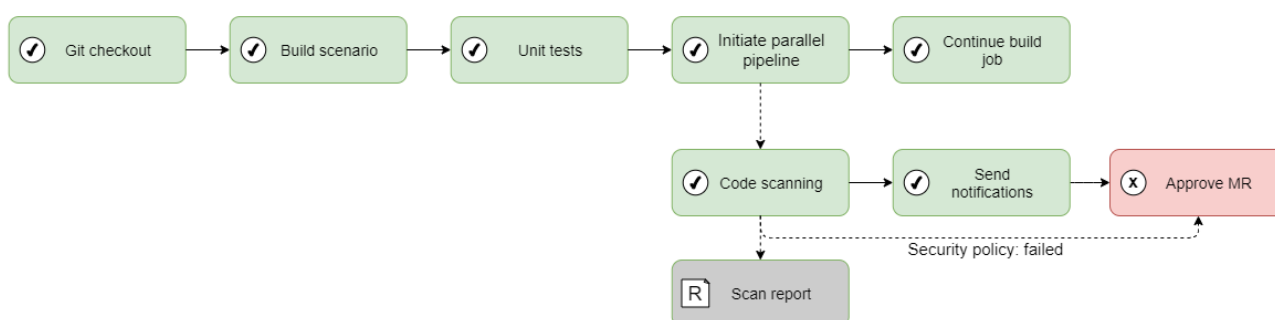


Рисунок 7. Пример сборочного процесса с блокировкой мердж-реквеста при нарушении политики безопасности

Параметры запуска плагина также позволяют:

- ограничивать перечень передаваемых на анализ файлов и папок;
- останавливать сборку при ошибках запуска сканирования кода;
- определять перечень формируемых отчетов о сканировании, которые будут сохранены в виде артефактов сборки.

Далее в документе представлены инструкции по использованию плагинов в CI/CD-системах, а также лучшие практики по интеграции в процессы разработки.

8.1.2. Установка легкого агента AI.Shell

Если CI-процесс выполняется на Linux-агентах или в Docker-контейнерах, необходимо установить на них компонент AI.Shell. AI.Shell — это легкий кросс-платформенный агент, который отправляет задачу сканирования модулю PT AI Enterprise Server, а PT AI Enterprise Server ставит задачу в очередь на выполнение доступному PT AI Enterprise Agent.

В этом разделе

[Установка AI.Shell из пакета \(см. раздел 8.1.2.1\)](#)

[Установка AI.Shell из инсталлятора для Windows \(см. раздел 8.1.2.2\)](#)

[Установка AI.Shell из репозитория \(см. раздел 8.1.2.3\)](#)

[Подготовка Docker-образа AI.Shell \(см. раздел 8.1.2.4\)](#)

8.1.2.1. Установка AI.Shell из пакета

В зависимости от ОС, AI.Shell имеет несколько вариантов поставки.

Таблица 22. Пакеты установки AI.Shell

ОС	Расширение пакета установки
Microsoft Windows (x64)	Стандартный инсталлятор для Microsoft Windows (содержит все файлы приложения, необходимые для работы) или готовый к использованию пакет для Windows Docker, не требующий установки
Alpine (x64) последней версии или LTS	.tar.gz
CentOS (x64) последней версии или LTS	.rpm
Ubuntu (x64) последней версии или LTS	.deb
Debian (x64) последней версии или LTS	.deb
Другие Linux-системы	.tar.gz

Установка AI.Shell из пакетов с расширениями .deb и .rpm

Примечание. Если в ОС не установлена библиотека `libssl`, которая необходима для безопасной передачи данных через интернет и является частью реализации протоколов шифрования SSL и TLS, ее нужно предварительно установить.

► Чтобы установить AI.Shell из пакетов с расширениями `.deb` и `.rpm`:

1. Для `.deb`-пакетов выполните команду:

```
apt-get install <Имя пакета>
```

или

```
dpkg -i <Имя пакета>
```

2. Для `.rpm`-пакетов выполните команду:

```
rpm -i <Имя пакета>
```

Установка AI.Shell из .tar.gz-архива

► Чтобы установить AI.Shell из .tar.gz-архива:

1. Скачайте предоставленный архив и распакуйте его в каталог `/usr/share/aisa`.
2. Выполните команду установки:
`sh /usr/share/aisa/scripts/install.sh`

Примечание. Вы можете удалить AI.Shell командой `sh /usr/share/aisa/scripts/remove.sh`.

8.1.2.2. Установка AI.Shell из инсталлятора для Windows

► Чтобы установить AI.Shell на компьютер:

1. Запустите установочный файл.
2. Выберите язык установки и нажмите **ОК**.
3. В окне с текстом лицензионного соглашения ознакомьтесь с условиями лицензионного соглашения.
4. Выберите **Я принимаю условия соглашения**, если вы согласны со всеми пунктами лицензионного соглашения.
5. Нажмите **Далее**.
6. Выберите папку для установки программы.
7. Нажмите **Далее**.
8. В открывшемся окне настройте подключение AI.Shell к модулю PT AI Enterprise Server по токenu доступа:
 - введите FQDN PT AI Enterprise Server с указанием протокола;
 - введите токен доступа.

Примечание. Для создания токена доступа необходимо по завершении установки модуля PT AI Enterprise Server перейти в веб-интерфейс PT AI Enterprise Edition и выполнить инструкцию в разделе «Создание токена доступа» в Руководстве пользователя.

9. Нажмите **Далее**.
10. Убедитесь в правильном выборе параметров установки и нажмите **Установить**.

Начнется установка программы на ваш компьютер. Дождитесь завершения установки.

11. Нажмите **Завершить**.

8.1.2.3. Установка AI.Shell из репозитория

► Чтобы установить AI.Shell из репозитория:

1. Скачайте и установите открытый ключ для проверки подписи пакета:
`wget -q0 - https://update.ptsecurity.com/packages/PT-public.gpg | sudo apt-key add -`
2. Подключите репозиторий:
`echo 'deb https://update.ptsecurity.com/packages/deb/AI.Shell/release all non-free' | sudo tee -a /etc/apt/sources.list`
3. Обновите список пакетов и установите AI.Shell:
`sudo apt update && sudo apt install aisa`

8.1.2.4. Подготовка Docker-образа AI.Shell

Вы можете использовать готовый Docker-образ AI.Shell для запуска в вашем контейнере.

► Чтобы подготовить Docker-образ:

1. Создайте Dockerfile на основе [примера \(см. раздел 8.1.2.4\)](#), представленного ниже, и укажите в нем следующие аргументы:

SOURCE_TOOL_IMAGE — название базового Docker-образа. При использовании образа из внутреннего Docker-репозитория компании необходимо изменить значение этого параметра.

ROOT_CA_CERT и **INTERMEDIATE_CA_CERT** — содержимое доверенных сертификатов корневого и выпускающего центров сертификации в формате PEM.

PTAI_VERSION — версия PT AI Enterprise Edition, которой соответствует используемый пакет AI.Shell.
2. Запустите сборку Docker-образа (на примере PT AI Enterprise Edition версии 4.11.0):
`ROOT_CA_CERT=`cat ~/x509/ptdemo.local/ptdemo-root-ca.pem.crt`
INTERMEDIATE_CA_CERT=`cat ~/x509/ptdemo.local/ptdemo-intermediate-ca.pem.crt`
PTAI_VERSION=4.11.0.47790
docker build \
--build-arg "ROOT_CA_CERT=$ROOT_CA_CERT" \
--build-arg "INTERMEDIATE_CA_CERT=$INTERMEDIATE_CA_CERT" \
--build-arg "PTAI_VERSION=$PTAI_VERSION" \
--tag aisa:"$PTAI_VERSION".`

Пример Dockerfile

Dockerfile позволяет вам самостоятельно собрать готовый к использованию Docker-образ, содержащий AI.Shell и все необходимые вспомогательные компоненты.

```
ARG SOURCE_TOOL_IMAGE=ubuntu:18.04
FROM ${SOURCE_TOOL_IMAGE}
```

```

ARG PTAI_VERSION=4.6.0.29976

ARG ROOT_CA_CERT
ARG INTERMEDIATE_CA_CERT

# Install external packages
RUN set -ex && \
    apt-get update && \
    apt-get install -y --no-install-recommends ca-certificates gnupg curl apt-utils \
    libssl1.1 libgssapi-krb5-2 libicu60 && \
    echo "${ROOT_CA_CERT}" > /usr/local/share/ca-certificates/root-ca.crt && \
    echo "${INTERMEDIATE_CA_CERT}" > /usr/local/share/ca-certificates/intermediate- \
    ca.crt && \
    update-ca-certificates && \
    apt-get clean && \
    rm -rf /var/lib/apt/lists/*

# Install AISA
RUN set -ex && \
    mkdir -p /etc/apt/keyrings && \
    curl -fsSL https://update.ptsecurity.com/packages/PT-public.gpg | gpg --dearmor \
    -o /etc/apt/keyrings/pt.gpg && \
    echo \
    "deb [arch=$(dpkg --print-architecture) signed-by=/etc/apt/keyrings/pt.gpg] \
    https://update.ptsecurity.com/packages/deb/AI.Shell/release all non-free" | \
    tee /etc/apt/sources.list.d/pt.list > /dev/null && \
    apt-get update && \
    DEBIAN_FRONTEND=noninteractive apt-get install -yq aisa=${PTAI_VERSION}

```

8.1.3. Конфигурирование AI.Shell и запуск сканирования

Вы можете создавать проект и запускать сканирование с помощью AI.Shell из командной строки или на CI-агенте.

- Чтобы запустить задачу сканирования с помощью AI.Shell:

1. Настройте подключение AI.Shell к PT AI Enterprise Server командой:

```
aisa --set-settings -u <Адрес PT AI Enterprise Server> -t <Токен доступа>
```

Примечание. Если запуск сканирования производится на CI-агенте, вы можете не настраивать подключение к PT AI Enterprise Server отдельной командой, а указывать параметры подключения непосредственно при запуске сканирования: `aisa -u <Адрес PT AI Enterprise Server> -t <Токен доступа> --project-name "<Имя проекта>" --scan-target "<Путь к объекту сканирования>".`

2. Если вы хотите запустить сканирование ранее созданного проекта с текущими параметрами, выполните команду:

```
aisa --project-name "<Имя проекта>" --scan-target "<Путь к объекту сканирования>" --branch-name "<Имя ветки>" --scan-label <Метка сканирования>
```

Примечание. Если нужно просканировать основную ветку, параметр `--branch-name` можно не указывать, для создания новой ветки используйте параметр `--create-branch`. Параметром `--scan-label` задается метка, идентифицирующая конкретное сканирование в рамках ветки. В качестве метки можно указать хеш коммита или идентификатор пайплайна.

Примечание. Список всех параметров запуска сканирования приведен в разделе «[Параметры запуска сканирования с помощью AI.Shell \(см. раздел 8.1.3.1\)](#)».

3. Если вы хотите запустить сканирование проекта и одновременно настроить его параметры в конфигурационном файле, выполните команду:

```
aisa --project-settings-file "<Путь к конфигурационному файлу>" --scan-target "<Путь к объекту сканирования>"
```

Примечание. Параметры, указанные в конфигурационном файле, имеют приоритет по отношению к заданным в командной строке.

В этом разделе

[Параметры запуска сканирования с помощью AI.Shell \(см. раздел 8.1.3.1\)](#)

[Коды возврата \(см. раздел 8.1.3.2\)](#)

8.1.3.1. Параметры запуска сканирования с помощью AI.Shell

Ниже приведен список параметров запуска сканирования, доступных в AI.Shell.

Таблица 23. Список параметров для AI.Shell

Параметр	Описание
<code>--set-settings</code>	Настройка параметров для работы AI.Shell. Команда используется вместе с параметрами <code>u</code> и <code>t</code> . Параметры записываются в конфигурационный файл AI.Shell, сканирование при этом не запускается
<code>-u</code>	Адрес PT AI Enterprise Server. Примечание. Параметры <code>-u</code> и <code>-t</code> учитываются только в паре. Если параметры заданы, то сначала происходит валидация этих параметров, а затем сканирование
<code>-t</code>	Токен доступа
<code>--no-wait</code>	Отключение подписки на события задачи сканирования. Если параметр задан, то после отправки сканирования в очередь работа AI.Shell завершится, если не задан — AI.Shell примет от PT AI Enterprise Server все статусы задачи сканирования и прогресс сканирования до момента завершения задачи. Только после этого работа AI.Shell завершится. Примечание. Если задан параметр <code>--no-wait</code> , выпуск отчетов невозможен, даже если заданы параметры создания отчетов <code>--reports-folder</code> и <code>--report</code>
<code>--create-project</code>	Создание проекта, если он не был создан ранее. Используется с параметрами <code>--project-name</code> и <code>--scan-target</code>
<code>--project-name</code>	Регистронезависимое имя проекта сканирования
<code>--scan-target</code>	Путь к папке или файлу приложения для сканирования. Если путь не задан, будет просканирована текущая папка
<code>--dsl-path</code>	Путь к папке с пользовательскими DSL-правилами
<code>--create-branch</code>	Создание ветки
<code>--branch-name</code>	Имя ветки
<code>--status</code>	Подписка на получение информации о сканировании, запущенном с параметром <code>--no-wait</code> . Используется с параметрами <code>--project-id</code> и <code>--scan-result-id</code> . При добавлении параметра <code>--no-wait</code> приходит однократное сообщение о статусе сканирования
<code>--project-id</code>	Уникальный идентификатор проекта
<code>--scan-result-id</code>	Уникальный идентификатор сканирования
<code>--log-level</code>	Уровень журналирования.

Параметр	Описание
	Возможные аргументы (расположены в порядке уменьшения количества выводимой информации): — trace; — debug; — info; — warning; — error; — fatal; — none
<code>--policy-settings-file</code>	Путь к файлу с описанием политик безопасности. Может быть передан для обновления политик безопасности без запуска сканирования. Используется с параметрами <code>--project-name</code> и <code>--scan-off</code>
<code>--full-scan</code>	Принудительный запуск полного сканирования
<code>--file-exclusions</code>	Путь к файлу в формате gitignore, содержащему список объектов, которые нужно исключить из сканирования
<code>--policies-path</code>	Путь к файлу с описанием политик безопасности. Может быть передан при запуске сканирования проекта
<code>--project-settings-file</code>	Путь к конфигурационному файлу
<code>--reports-folder</code>	Путь к папке для сохранения отчетов
<code>--report</code>	Выпуск отчета по окончании сканирования. Параметр содержит название шаблона отчета. Возможные значения параметра: PlainReport, XML, JSON, GitLab, MD, Sarif, AutoCheck, Nist, Oud4, Owaspm, Owaspm, Pcidss, Sans, <Название пользовательского шаблона отчета>. Для выпуска нескольких отчетов вы можете передавать названия их шаблонов через пробел в одном токене (<code>--report json md</code>) или в отдельных (<code>--report json --report md</code>). Названия пользовательских шаблонов, содержащие пробелы, заключаются в двойные кавычки. Например, <code>--report "my template"</code>
<code>--waf-patch</code>	Выпуск патча для PT AF
<code>--help</code>	Отображение всех доступных параметров запуска
<code>--scan-off</code>	Создание проекта по конфигурационному файлу без запуска задачи сканирования. Используется вместе с параметром <code>--project-settings-file</code>

Параметр	Описание
--retry	Параметр, задающий постановку сканирования в очередь, если агент сканирования занят. Ожидание освобождения агента составляет 3600 секунд
--retry-time	Тайм-аут (в секундах), в течение которого Al.Shell ожидает освобождения агента сканирования и пытается поставить новое сканирование в очередь. Максимальное значение — 2 147 483 647, минимальное — 5
--scan-label	Метка для идентификации конкретного сканирования в рамках ветки. Может содержать до 40 символов. Допустимые символы — буквы русского и латинского алфавитов, цифры, дефис, точка и знак подчеркивания

8.1.3.2. Коды возврата

По завершении сканирования отображается код возврата. Ниже приведен список возможных кодов возврата.

Таблица 24. Список кодов возврата

Код	Описание
0	Успешное сканирование
-1	Программа уже запущена
-2	Ошибка запуска дочернего процесса
2	Не найдена папка для сканирования
3	Проблема с лицензией
4	Не найден проект для сканирования
5	Ошибки в параметрах проекта
6	Некритические ошибки сканирования
7	Ошибка отправки отчета на электронную почту
8	Некорректный путь к конфигурационному файлу
9	Некорректный путь к папке отчетов
10	Сработала политика безопасности (сборка не удовлетворяет заданной политике безопасности)
11	Некорректная настройка параметров отчетов
13	Сканирование было удалено
17	Ошибка в политике безопасности

Код	Описание
18	Критическая ошибка модуля PT AI Enterprise Agent
19	Не найден модуль PT AI Enterprise Agent
20	Ошибка загрузки исходных файлов с PT AI Enterprise Server
21	Тайм-аут оповещения PT AI Enterprise Server о нормальной работе PT AI Enterprise Agent
22	Ошибка в процессе обновления
24	Не найден серверный сертификат
25	Недействительный URI PT AI Enterprise Server
27	Версия AI.Shell устарела, доступна новая версия для обновления
28	Отсутствуют активные агенты сканирования
29	Отсутствует токен доступа для AI.Shell
30	Неверный путь к файлу с описанием политики безопасности
31	Сканирование проекта уже запущено
32	Ошибка подключения к PT AI Enterprise Server (например, AI.Shell новой версии установлен на PT AI Enterprise Server старой версии)
33	Ошибка обновления исходных файлов проекта
34	Для запущенного сканирования не выбраны модули сканирования
35	Некорректные параметры черного ящика или автопроверки
36	В конфигурационном файле указана невалидная версия
37	Недопустимый уровень журналирования
38	Некорректное содержимое файла <code>appSettings.json</code> или файл отсутствует
39	Не удалось определить язык программирования
40	Некорректное содержимое файла с исключениями или файл отсутствует
60	Совмещенные коды 6 и 10. Сработала политика безопасности, и в ходе сканирования найдены не критические ошибки
100	Сканирование отменено
1000	Неопознанная ошибка

8.1.4. Настройка параметров сканирования в конфигурационном файле

Вы можете сгенерировать конфигурационный файл с параметрами проекта при запуске сканирования с помощью AI.Shell или создать его заранее и хранить в своем репозитории.

► Чтобы настроить параметры сканирования в конфигурационном файле:

1. Создайте конфигурационный файл в формате JSON в текстовом редакторе.
2. Пропишите в конфигурационном файле параметры сканирования.

Примечание. При написании конфигурационного файла вы можете воспользоваться готовым примером. Схему описания параметров сканирования см. на сайте json.schemastore.org/aiproj-1.11.json.

3. Сохраните конфигурационный файл.

Пример конфигурационного файла

Ниже приведен пример конфигурационного файла, в котором задаются параметры сканирования. Конфигурационный файл поддерживает версионирование. Если указана устаревшая версия, файл будет автоматически преобразован в новый формат, если версия не указана, формат файла по умолчанию считается актуальным.

```
{
  "$schema": "https://json.schemastore.org/aiproj-1.11.json", // JSON-схема,
  описывающая конфигурационный файл
  "Version": "1.11", // Версия конфигурационного файла: 1.10 для PT AI Enterprise
  Edition версии 6.1.0 и выше
  "ProjectName": "Test_Project", // Название проекта
  "ProgrammingLanguages": ["Php", "Java", "CSharp (Windows, Linux)",
  "JavaScript"], // Языки приложения: Java, CSharp (Windows, Linux), CSharp (Windows),
  Php, JavaScript, Python, ObjectiveC, Swift, Solidity, CAndCPlusPlus, Go, Kotlin, Sql,
  Ruby, Scala, OneC, Dart
  "ScanModules": ["StaticCodeAnalysis"], // Модули поиска уязвимостей:
  Configuration, Components, BlackBox, PatternMatching, StaticCodeAnalysis,
  SoftwareCompositionAnalysis, MaliciousCodeDetection, SecretDetection
  "BranchName": "default", // Имя ветки
  "ApplyAllPMRules": false, // Включить все правила анализа
  "SkipGitIgnoreFiles": true, // Не сканировать файлы и папки, указанные
  в .gitignore
  "UseSecurityPolicies": true, // Включить проверку политик безопасности
  "MailingProjectSettings": {
    "Enabled": true, // Включить отправку отчета по почте по окончании
    сканирования
    "MailProfileName": "MailProfile", // Название почтового профиля
    "EmailRecipients": ["user@email.com"] // Список адресатов
  },
  "DotNetSettings": {
    "ProjectType": "Solution", // Тип проекта: Solution
    "SolutionFile": "./file.sln", // Путь к файлу решения или проекта
    "CustomParameters": null, // Дополнительные параметры запуска модуля анализа
    "DownloadDependencies": true, // Загружать зависимости
    "UsePublicAnalysisMethod": true, // Искать от доступных методов public и
    protected
    "DslRulesRelativePath": "" // Путь к DSL-правилам относительно корневой папки
    объекта сканирования
  },
  "GoSettings": {
    "CustomParameters": "--log-level debug", // Дополнительные параметры запуска
    модуля анализа
    "DslRulesRelativePath": "", // Путь к DSL-правилам относительно корневой
    папки объекта сканирования
    "DownloadDependencies": true, // Загружать зависимости
    "UsePublicAnalysisMethod": false // Искать от доступных методов public и
    protected
  },
}
```

```

"JavaScriptSettings": {
    "UsePublicAnalysisMethod": true, // Искать от доступных методов public и
protected
    "DownloadDependencies": true, // Загружать зависимости
    "UseTaintAnalysis": true, // Включить модуль сканирования Taint
    "UseJsaAnalysis": true, // Включить модуль сканирования JSA
    "CustomParameters": "", // Дополнительные параметры запуска модуля анализа
    "DslRulesRelativePath": "" // Путь к DSL-правилам относительно корневой папки
объекта сканирования
},
"JavaSettings": {
    "Parameters": null, // Параметры запуска Java Virtual Machine
    "UnpackUserPackages": false, // Распаковка пользовательских JAR-файлов
    "UserPackagePrefixes": null, // Префиксы пользовательских пакетов
    "Version": "21", // Версия JDK: 8, 11, 17, 21, 25
    "UsePublicAnalysisMethod": true, // Искать от доступных методов public и
protected
    "DownloadDependencies": true, // Загружать зависимости
    "CustomParameters": null, // Дополнительные параметры запуска модуля анализа
    "DslRulesRelativePath": "" // Путь к DSL-правилам относительно корневой папки
объекта сканирования
},
"PhpSettings": {
    "CustomParameters": "--log-level debug", // Дополнительные параметры запуска
модуля анализа
    "DslRulesRelativePath": "", // Путь к DSL-правилам относительно корневой
папки объекта сканирования
    "DownloadDependencies": true, // Загружать зависимости
    "UsePublicAnalysisMethod": false // Искать от доступных методов public и
protected
},
"PmTaintSettings": {
    "UsePublicAnalysisMethod": true, // Искать от доступных методов public и
protected
    "CustomParameters": "--sgrep-rule", // Дополнительные параметры запуска
модуля анализа
    "PMGroups": [] // Список групп правил анализа
},
"PythonSettings": {
    "UsePublicAnalysisMethod": true, // Искать от доступных методов public и
protected
    "DownloadDependencies": false, // Загружать зависимости
    "CustomParameters": "--log-level debug", // Дополнительные параметры запуска
модуля анализа
    "DslRulesRelativePath": "" // Путь к DSL-правилам относительно корневой папки
объекта сканирования
},
"RubySettings": {

```

```

    "UsePublicAnalysisMethod": true, // Искать от доступных методов public и
protected
    "CustomParameters": "--log-level debug", // Дополнительные параметры запуска
модуля анализа
    "DslRulesRelativePath": "" // Путь к DSL-правилам относительно корневой папки
объекта сканирования
},
    "ScaSettings": {
        "CustomParameters": "--sca --log-level debug" // Дополнительные параметры
запуска модуля анализа
        "BuildDependenciesGraph": true // Создание графа зависимостей
    },
    "BlackBoxSettings": {
        "AdditionalHttpHeaders": [{"Key": "cache-control", "Value": "no-cache"}], //
Дополнительные HTTP-заголовки
        "Level": "Normal", // Режим поиска: Fast, Normal, Full
        "RunAutocheckAfterScan": false, // Запускать автоматическую проверку
уязвимостей после сканирования
        "ScanScope": "Folder", // Область сканирования: Folder, Domain, Path
        "Site": "https://test.com/", // Адрес сайта
        "SslCheck": true, // Проверка безопасности HTTPS-соединения
        "Authentication": { // Параметры аутентификации
            "Type": "RawCookie", // Тип аутентификации
            "Cookie": { // Тип аутентификации: Cookie
                "Cookie": "", // Cookie авторизованного пользователя
                "ValidationAddress": "", // Адрес страницы, на которую вы переходите в
случае успешной аутентификации
                "ValidationTemplate": "" // Текст, с помощью которого можно определить,
что аутентификация прошла успешно
            },
            "Form": { // Тип аутентификации: Form
                "FormDetection": "Manual", // Обнаружение формы: Auto, Manual
                "FormAddress": "", // URL страницы, содержащей форму аутентификации
                "FormXPath": "", // XPath формы
                "Login": "", // Логин для аутентификации на сайте
                "LoginKey": "", // Идентификатор для логина
                "Password": "", // Пароль для аутентификации на сайте
                "PasswordKey": "", // Идентификатор для пароля
                "ValidationTemplate": "" // Текст, с помощью которого можно определить,
что аутентификация прошла успешно
            },
            "Http": { // Тип аутентификации: HTTP
                "Login": "", // Логин для аутентификации на сайте
                "Password": "", // Пароль для аутентификации на сайте
                "ValidationAddress": "" // Адрес страницы, на которую вы переходите в
случае успешной аутентификации
            }
        },
    },
},

```

```

    "WhiteListedAddresses": [
        {
            "Address": "*logout*", // Значение адреса из белого списка адресов
            "Format": "Wildcard" // Тип записи адреса: Wildcard, ExactMatch,
RegExp
        }
    ],
    "BlackListedAddresses": [
        {
            "Address": "^setup", // Значение адреса из черного списка адресов
            "Format": "RegExp" // Тип записи адреса: Wildcard, ExactMatch, RegExp
        }
    ],
    "ProxySettings": { // Параметры прокси-сервера
        "Enabled": false,
        "Type": "Http", // Тип подключения к прокси-серверу: Http, Socks4, Socks5
        "Host": "", // IP-адрес или доменное имя прокси-сервера
        "Port": 3128, // Порт прокси-сервера
        "Login": "", // Логин для подключения к прокси-серверу
        "Password": "" // Пароль для подключения к прокси-серверу
    }
}
}

```

8.1.5. Сборка плагинов

Для сборки плагина необходимо предварительно установить JDK версии 8 или 11.

► Чтобы выполнить сборку плагина из исходного кода на узле с доступом в интернет:

1. Загрузите плагины с помощью Git:
`git clone https://github.com/POSIdev-community/ptai-ee-tools.git`
2. В командной строке перейдите в корень каталога с плагином и выполните сборку:
`./gradlew build`
3. Если требуется, для плагина `AI.TeamCity.Plugin` укажите версию `TeamCity`:
`./gradlew build -P teamcityVersion=xxxx.y`

Примечание. Плагин `AI.Jenkins.Plugin` собирается для минимальной поддерживаемой версии Jenkins 2.300 и совместим со всеми более новыми версиями. При сборке менять значение параметра `jenkinsVersion` не требуется.

По завершении сборки плагины будут доступны по следующим путям:

- `ptai-teamcity-plugin\ptai-teamcity-plugin-server\build\distributions\ptai-teamcity-plugin.zip`;
- `ptai-jenkins-plugin\build\libs\ptai-jenkins-plugin.hpi`;
- `ptai-cli-plugin\build\libs\ptai-cli-plugin.jar`

Примечание. Если в процессе сборки вы получили ошибку, добавьте каталог с исходным кодом плагина в список исключений для вашего антивируса и повторите сборку. Если проблема не решится, обратитесь в Positive Technologies.

8.1.6. Интеграция с Jenkins

Для подключения анализа кода к сборке в Jenkins используется плагин `AI.Jenkins.Plugin`. Он устанавливается и настраивается на сервере Jenkins, что позволяет не настраивать каждый агент сборки Jenkins по отдельности.

После установки и настройки плагина вы можете добавить новый шаг сборки для сканирования кода с помощью PT AI Enterprise Edition, указав параметры сканирования, режим работы (синхронный или асинхронный), перечень файлов для анализа и требования к выпускаемым отчетам.

В синхронном режиме вы можете получать отчеты в виде артефактов сборки и скачивать их из Jenkins. Если выпуск отчетов не задан или используется асинхронный режим, вы можете просматривать результаты сканирования в веб-интерфейсе PT AI Enterprise Edition. По завершении сканирования в Jenkins отображается статистика по найденным уязвимостям для выбранной сборки и по проекту целиком.

В этом разделе

[Настройка плагина \(см. раздел 8.1.6.1\)](#)

[Настройка проектной задачи \(см. раздел 8.1.6.2\)](#)

[Создание Groovy-кода для запуска плагина \(см. раздел 8.1.6.3\)](#)

8.1.6.1. Настройка плагина

Настройка плагина `AI.Jenkins.Plugin` состоит из следующих шагов:

1. Получение плагина для вашей версии Jenkins в Positive Technologies или сборка плагина самостоятельно [по инструкции \(см. раздел 8.1.5\)](#).
2. Установка плагина в Jenkins.
3. Настройка подключения плагина к PT AI Enterprise Server.

Установка плагина в Jenkins

► Чтобы загрузить плагин в Jenkins:

1. Войдите в Jenkins под учетной записью администратора системы.
2. В левой части главной страницы выберите раздел **Manage Jenkins**.
3. На открывшейся странице выберите пункт **Manage Plugins**.
4. На странице **Plugin manager** выберите вкладку **Advanced**.
5. В блоке **Upload Plugin** нажмите кнопку **Choose File**.
6. В открывшемся окне выберите файл с плагином и нажмите кнопку **Open**.
7. Нажмите кнопку **Upload**.

Плагин установлен и отображается на странице **Plugin Manager** на вкладке **Installed**.

Внимание! В ранних версиях Jenkins могут возникать проблемы с установкой зависимых плагинов для AI.Jenkins.Plugin. В этом случае необходимо вручную установить зависимые плагины, приведенные в таблице ниже.

Таблица 25. Зависимые плагины для AI.Jenkins.Plugin

Название плагина	Минимальная версия
structs	1.20
workflow-step-api	2.14
configuration-as-code	1.35
token-macro	2.12
credentials	2.3.11

Настройка подключения плагина к PT AI Enterprise Server

► Чтобы настроить подключение плагина к PT AI Enterprise Server:

1. В левой части главной страницы Jenkins выберите раздел **Manage Jenkins**.
2. На открывшейся странице выберите пункт **Configure System**.
3. В списке параметров в блоке **PT AI vulnerability analysis** нажмите кнопку **Add PT AI global configuration**, чтобы добавить новое подключение к PT AI Enterprise Server.

PT AI vulnerability analysis

PT AI global configurations

Configuration name

ptai-scan

PT AI server URL

https://ai.ptsecurity.ru

Credentials

eb79f4b0-7b45-418c-b5e5-13f0e2b5d578 Add

☐ Ignore SSL certificate check

Test PT AI server connection

Delete

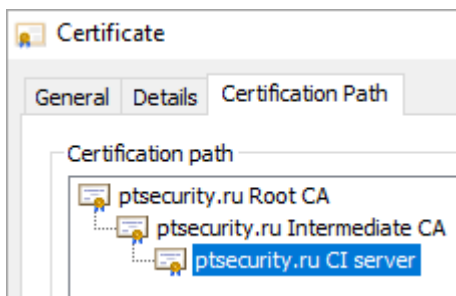
Рисунок 8. Настройка подключения плагина к PT AI Enterprise Server

4. В поле **Configuration name** введите название подключения, которое будет отображаться в веб-интерфейсе Jenkins.
5. В поле **PT AI server URL** введите адрес PT AI Enterprise Server.
6. Нажмите кнопку **Add** рядом с полем **Credentials** и в раскрывшемся меню выберите пункт **Jenkins**.

Откроется окно **Jenkins Credentials Provider: Jenkins**.

- В раскрывающемся списке **Kind** выберите **PT AI server authentication**.
- В поле **PT AI client API token** введите токен доступа для плагинов CI/CD.
- В поле **PT AI server CA certificates** укажите цепочку SSL-сертификатов удостоверяющего центра, выпустившего SSL-сертификат для PT AI Enterprise Server, или сам сертификат PT AI Enterprise Server (при использовании самоподписанного сертификата) в кодировке Base64.

Примечание. Если иерархия сертификатов, используемых для выпуска сертификата PT AI Enterprise Server, содержит Root CA и Intermediate CA как показано на рисунке ниже, необходимо выгрузить эти сертификаты в кодировке Base64, затем в текстовом редакторе соединить их в цепочку в обратном порядке: Intermediate → Root, после чего скопировать и вставить полученную цепочку ключей в поле **PT AI server CA certificates**.



213

10. Проверьте серверный сертификат по кнопке **Test CA certificates**.
11. Нажмите кнопку **Add**.
12. Если вы не указали цепочку SSL-сертификатов, или она не прошла проверку, установите флажок **Ignore SSL certificate check** для игнорирования проблем с сертификатом.

Примечание. В этом случае сохраняется возможность установки защищенного соединения с PT AI Enterprise Server, но не выполняется проверка принадлежности сертификата, предъявленного сервером в ходе установки SSL-соединения, этому серверу. Поэтому не рекомендуется устанавливать флажок **Ignore SSL certificate check** при развертывании в рабочей среде.

13. Нажмите кнопку **Save**.

Подключение плагина к PT AI Enterprise Server настроено. Вы можете перейти к настройке проектной задачи.

8.1.6.2. Настройка проектной задачи

Шаг анализа кода встраивается в существующий процесс сборки. Поскольку эти процессы могут быть устроены по-разному, в инструкции рассматривается пример создания нового Freestyle-проекта с минимальной конфигурацией для проверки кода в PT AI Enterprise Edition.

Если вы работаете с существующим процессом сборки и хотите изменить его, начните выполнение инструкции с шага 6. Если вы описываете процесс сборки с использованием Groovy-скриптов, перейдите к разделу «[Создание Groovy-кода для запуска плагина \(см. раздел 8.1.6.3\)](#)».

Проектная задача создается отдельно для каждого проекта сканирования.

► Чтобы настроить проектную задачу:

1. В левой части главной страницы Jenkins выберите раздел **New Item**.
2. На открывшейся странице выберите пункт **Freestyle project** и укажите в поле название проекта.
3. Нажмите кнопку **OK**.
4. На открывшейся странице в блоке **Source Code Management** в поле **Repository URL** введите адрес репозитория, в котором хранится исходный код сканируемого приложения.

Source Code Management

☐ None

☒ Git

Repositories

Repository URL

https://github.com/ptssdl/App01

Credentials

- none - Add

Рисунок 11. Настройка загрузки исходного кода из Git-репозитория

- В блоке **Build** в меню кнопки **Add build step** добавьте шаги по сборке, юнит-тестированию и загрузке зависимостей, чтобы перед запуском шага анализа кода у вас был гарантированно собранный код и используемые им сторонние компоненты.

Build

Invoke top-level Maven targets

Goals

clean install

Advanced...

Invoke top-level Maven targets

Goals

dependency:copy-dependencies -DoutputDirectory=\"\$WORKSPACE/libs\"

Advanced...

Рисунок 12. Настройка проектной задачи

- Добавьте в сборку шаг с анализом кода, выбрав в раскрывающемся списке **Add build step** шаг **PT AI vulnerability analysis**.
- В раскрывающемся списке **Branch at PT AI Server** выберите:
 - From pipeline environment**, если название сканируемой ветки прописано в переменной окружения пайплайна;
 - Custom branch**, если нужно указать ветку вручную, в этом случае в поле **Branch name** введите ее название.

8. В раскрывающемся списке **Scan settings type** выберите:

- **PT AI UI-defined settings**, если сканирование настроено в веб-интерфейсе PT AI Enterprise Edition.
- **JSON-defined settings**, если вы хотите прописать конфигурационный файл и файл политики безопасности на этапе сборки.

Примечание. Параметр **BranchName**, указанный в конфигурационном файле при выборе **JSON-defined settings**, имеет более низкий приоритет по сравнению с параметром, указанным в **Branch at PT AI Server**, и будет проигнорирован.

9. Если вы выбрали **PT AI UI-defined settings**, в поле **Project name** введите имя проекта, созданного и настроенного в веб-интерфейсе PT AI Enterprise Edition.

The screenshot shows a configuration window for PT AI vulnerability analysis. The window has a title bar with a hamburger menu, the title 'PT AI vulnerability analysis', a help icon, and a close button. The main content area is divided into several sections:

- Branch at PT AI Server:** A dropdown menu currently showing 'From pipeline environment'.
- Scan settings type:** A dropdown menu currently showing 'PT AI UI-defined settings'.
- Project name:** A text input field containing 'Java App 01'.
- Server config:** A dropdown menu currently showing 'Global scope defined PT AI server config', and a 'Configuration name' dropdown menu below it.
- Work mode:** A dropdown menu currently showing 'Synchronous mode'.
- AST results postprocessing:** A section containing a button labeled 'Add AST results postprocessing task'.

Рисунок 13. Настройка проектной задачи, если сканирование настроено в веб-интерфейсе PT AI Enterprise Edition

10. Если вы выбрали **JSON-defined settings**, в полях **Scan settings** и **Policy** введите текст конфигурационного файла и файла политики (необязательно) в формате JSON.

PT AI vulnerability analysis ?

Scan settings type

JSON-defined settings

Scan settings ?

```
{
  "$schema": "https://json.schemastore.org/aiproj-1.3.json",
  "Version": "1.3",
  "ProjectName": "Java App 01",
  "ProgrammingLanguages": {

```

Success, project name is Java App 01, language is ObjectiveC

Policy ?

```
[
  {
    "CountToActualize": 1,
    "Scopes": [

```

Success, policy contains 1 rules

Server config

Global scope defined PT AI server config

Configuration name

ptai-presentation

Рисунок 14. Настройка проектной задачи, если сканирование настраивается на этапе сборки

11. В раскрывающемся списке **Server config** выберите:
 - **Global scope defined PT AI server config** для использования глобальных параметров плагина, заданных при настройке подключения плагина к PT AI Enterprise Server.
 - **Task scope defined PT AI server config** для использования параметров плагина, заданных для конкретного проекта. Например, если вы хотите настроить подключение проекта к другому PT AI Enterprise Server.
12. Если вы выбрали **Global scope defined PT AI server config**, в поле **Configuration name** выберите название подключения, заданное в поле при настройке подключения плагина к PT AI Enterprise Server.
13. Если вы выбрали **Task scope defined PT AI server config**, настройте параметры подключения плагина для конкретного проекта аналогично параметрам подключения плагина к PT AI Enterprise Server.
14. В раскрывающемся списке **Work mode** выберите:
 - **Asynchronous mode** для сканирования в асинхронном режиме. В этом режиме Jenkins запустит проверку кода на наличие уязвимостей и продолжит сборку, не дожидаясь завершения проверки.
 - **Synchronous mode** для сканирования в синхронном режиме. В этом режиме Jenkins запустит проверку кода на наличие уязвимостей и будет ожидать завершения проверки.
15. Если вы выбрали **Synchronous mode**:

- Нажмите кнопку **Add AST results postprocessing task** и в раскрывшемся меню выберите пункты **Processing: check AST errors and warnings** и **Processing: check AST policy assessment result**.
 - В блоке **Processing: check AST errors and warnings** в раскрывающемся списке **Set build step execution status** выберите пункт **Change build execution result to "Failed"**, чтобы Jenkins останавливал сборку при возникновении ошибок, не связанных с результатами проверки кода на наличие уязвимостей (например, ошибки в конфигурации, отсутствующая или недействительная лицензия).
 - В блоке **Processing: check AST policy assessment result** в раскрывающемся списке **Set build step execution status** выберите пункт **Change build execution result to "Failed"**, чтобы Jenkins останавливал сборку при несоответствии сканируемого приложения заданной политике безопасности.
16. Задайте параметры для формирования отчета об уязвимостях (только для синхронного режима):
- В поле **File name** введите название отчета.
 - В поле **Templates** введите название шаблона отчета. Список возможных значений представлен в веб-интерфейсе PT AI Enterprise Edition в разделе **Шаблоны отчетов**.
 - Установите флажок **Add data flow diagram**, если вы хотите добавить в отчет диаграмму потока данных.
 - Установите флажок **Add vulnerability summary**, если вы хотите добавить в отчет справочную информацию об уязвимостях.
 - Если требуется, в поле **Vulnerability filter** введите JSON-описание фильтров, необходимых для [формирования отчета \(см. раздел 8.1.9\)](#).

Примечание. Вы также можете указать параметры для формирования отчета с помощью JSON-описания по кнопке **Export AST data as** и выбрав **JSON-defined reports**. Синтаксис JSON-описания приведен в разделе «[Файл описания отчетов \(см. раздел 8.1.9\)](#)».

AST results postprocessing

Reporting: save AST results report file

File name ?

tier1.html

Template ?

Scan results report

☒ Add data flow diagram

☐ Add vulnerability summary

Advanced ^

Vulnerability filter ?

```
{
  "issueLevels": [
    "MEDIUM",
    "HIGH"
  ],
  "scanMode": "FROMENTRYPOINT",
  "confirmationStatuses": [
    "UNDEFINED",
    "NONE"
  ]
}
```

Add AST results postprocessing task ▾

Рисунок 15. Настройка выпуска отчетов

AST results postprocessing

Reporting: generate JSON-defined reports

JSON settings ?

```
{
  "report": [
    {
      "fileName": "tier1.html",
      "includeDfd": true,
      "includeGlossary": true,
      "template": "Отчет по результатам сканирования",
      "filters": {
        "issueLevels": [
          "MEDIUM",
          "HIGH"
        ],
        "scanMode": "FROMENTRYPOINT",
        "confirmationStatuses": [
          "UNDEFINED",
          "NONE"
        ]
      }
    }
  ]
}
```

Add AST results postprocessing task ▾

Рисунок 16. Настройка отчетов в JSON-описании

17. В поле **Files to analyze** задайте возможность включения и (или) исключения определенных файлов из сканирования по маске, используя синтаксис [Apache Ant](#).
18. В поле **Exclude files** задайте перечень исключаемых из сканирования файлов и папок, используя синтаксис [Apache Ant](#). Исключенные таким способом файлы и каталоги не попадут в PT AI Enterprise Server.

19. Установите флажок **Use default excludes**, чтобы использовать стандартный список файлов и папок, которые будут исключены из сканирования. Список исключений вы можете посмотреть по кнопке  справа от параметра.

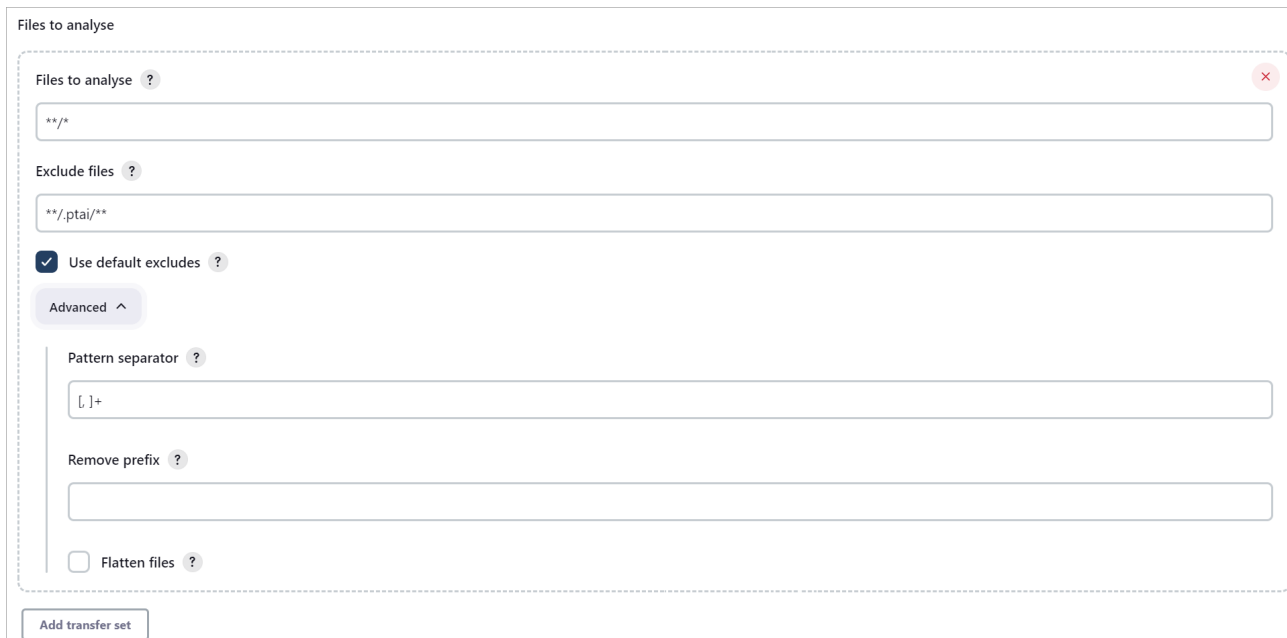


Рисунок 17. Настройка перечня передаваемых на сканирование файлов и папок

20. Нажмите кнопку **Advanced** для настройки дополнительных параметров:

- В поле **Pattern separator** задайте разделитель для полей **Files to analyze** и **Exclude files**.
- В поле **Remove prefix** при необходимости укажите начало пути к файлам, которое будет исключено при передаче в PT AI Enterprise Server. Например, если анализируемый код находится в папке `target/deployment/images/**/`, то при исключении `target/deployment` в PT AI Enterprise Server будет передано содержимое папки **deployment** без структуры `target/deployment`.
- Установите флажок **Flatten files**, чтобы передать все файлы без имен папок, если файлы вашего проекта имеют уникальные имена.

21. Установите флажок **Force full AST** для выполнения полного сканирования. Если флажок не установлен, будет запущено инкрементальное сканирование.
22. Установите флажок **Verbose log output** для включения расширенного протоколирования действий плагина.
23. Нажмите кнопку **Save**.
24. Запустите сборку проекта.

В процессе сборки запустится проверка кода на наличие уязвимостей через плагин. Сформированные отчеты сохранятся в разделе **Workspace** в каталоге **.ptai**.

[Back to Dashboard](#)
[Status](#)
[Changes](#)
[Workspace](#)
[Build Now](#)
[Configure](#)
[Delete Project](#)
[PT AI scan statistics](#)
[Rename](#)

Workspace of Sample Java App

.ptai /

ai_full_report.html	Fri Oct 15 17:18:47 MSK 2021
ai_report.json	Fri Oct 15 17:18:46 MSK 2021
blackbox.html	Fri Oct 15 17:18:49 MSK 2021
new.html	Fri Oct 15 17:18:49 MSK 2021
rest.url	Fri Oct 15 17:15:24 MSK 2021
tier1.html	Fri Oct 15 17:18:48 MSK 2021
tier2.html	Fri Oct 15 17:18:48 MSK 2021
tier3.html	Fri Oct 15 17:18:49 MSK 2021
vulnerable components.html	Fri Oct 15 17:18:49 MSK 2021

(all files in zip)

Рисунок 18. Отчеты о результатах проверки кода

Также вы можете посмотреть статистику по сборке, перейдя в результаты сборки и выбрав раздел **PT AI scan results**.

[Back to Project](#)
[Status](#)
[Changes](#)
[Console Output](#)
[View as plain text](#)
[Edit Build Information](#)
[Delete build '#2'](#)
[Git Build Data](#)
[PT AI scan results](#)
[Previous Build](#)

PT AI scan result statistic for this build

Scan settings

Project name	Java App 01
Web site address	http://localhost/123/
Programming language	JAVA
Work mode	Synchronous mode

Scan start date / time	15.10.21 17:15:15
Scan duration	0:01:43

Completion status	AST success
AST policy	AST policy violated

PT AI server URL	https://ai-presentation.ptsecurity.ru
PT AI server version	3.6.4.2843
PT AI agent version	3.6.4.1437

Breakdown of vulnerabilities

By severity (excluding rejected)

Severity	Count
High	1
Medium	2
Low	55
Potential	80

Рисунок 19. Статистика по выбранной сборке

В проекте также появится раздел **PT AI scan statistics** со статистикой по последним сборкам проекта.

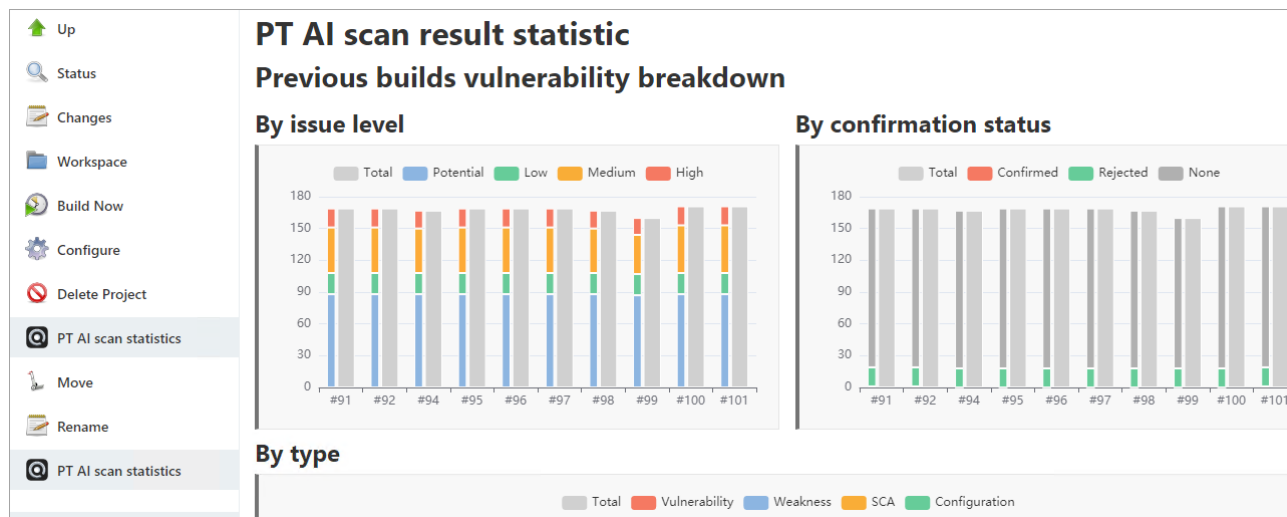


Рисунок 20. Статистика по последним сборкам проекта

8.1.6.3. Создание Groovy-кода для запуска плагина

- Чтобы настроить анализ кода с помощью Groovy-скриптов:
 1. Перейдите в **Pipeline Syntax** по адресу: `<Jenkins URL>/pipeline-syntax`.
 2. В раскрывающемся списке **Sample Step** выберите **ptaiUiSast: PT AI vulnerability analysis**.

Steps

Sample Step

ptaiAst: PT AI vulnerability analysis

ptaiAst ?

Branch at PT AI Server

Custom branch

Branch name ?

main

Scan label ?

Scan settings type

PT AI UI-defined settings

Project name ?

Java App 01

Server config

Global scope defined PT AI server config

Configuration name

ptai-scan

Рисунок 21. Выбор шага запуска плагина в **Pipeline Syntax**

3. Укажите параметры для анализа кода в соответствии с инструкцией из раздела «[Настройка проектной задачи \(см. раздел 8.1.6.2\)](#)», начиная с шага 7.
4. Нажмите кнопку **Generate Pipeline Script** для формирования команды вызова плагина в Groovy-скриптах.

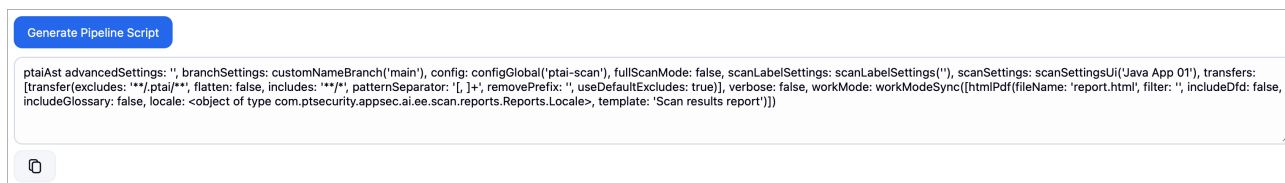


Рисунок 22. Генерация кода запуска плагина в **Pipeline Syntax**

5. Вставьте полученный код в ваш pipeline для запуска задачи анализа кода.

8.1.7. Интеграция с TeamCity

Для подключения анализа кода к сборке в TeamCity используется плагин `AI.TeamCity.Plugin`. Он устанавливается и настраивается на сервере TeamCity, что позволяет не настраивать каждый агент сборки TeamCity по отдельности.

После установки и настройки плагина вы можете добавить новый шаг сборки для сканирования кода с помощью PT AI Enterprise Edition, указав параметры сканирования, режим работы (синхронный или асинхронный), перечень файлов для анализа и требования к выпускаемым отчетам.

В синхронном режиме вы можете получать отчеты в виде артефактов сборки и скачивать их из TeamCity. Если выпуск отчетов не задан или используется асинхронный режим, вы можете просматривать результаты сканирования в веб-интерфейсе PT AI Enterprise Edition.

В этом разделе

[Настройка плагина \(см. раздел 8.1.7.1\)](#)

[Настройка проектной задачи \(см. раздел 8.1.7.2\)](#)

8.1.7.1. Настройка плагина

Настройка `AI.TeamCity.Plugin` плагина состоит из следующих шагов:

1. Получение плагина для вашей версии TeamCity в Positive Technologies или сборка плагина самостоятельно [по инструкции \(см. раздел 8.1.5\)](#).
2. Установка плагина в TeamCity.
3. Настройка подключения плагина к PT AI Enterprise Server.

Установка плагина в TeamCity

► Чтобы установить плагин в TeamCity:

1. Войдите в TeamCity под учетной записью администратора системы.
2. В меню администрирования выберите раздел **Plugins List**.
3. Нажмите кнопку **Upload plugin zip** и выберите файл с плагином.

Плагин установлен.

Настройка подключения плагина к PT AI Enterprise Server

► Чтобы настроить подключение плагина к PT AI Enterprise Server:

1. В меню администрирования выберите раздел **Integrations** → **PT AI**.

PT AI

PT AI server

PT AI server URL *

https://ai.ptsecurity.ru

PT AI server URL, for example, https://ptai.domain.org:443

PT AI client token *

.....

PT AI REST API client access token. Generate token using PT AI administrative web interface and place it in this field.

PT AI server CA certificates

Trust these CA certificates:

```
-----BEGIN CERTIFICATE-----
MIIDWzCCAkOgAwIBAgIQVJB6yIn11rhDZ7EvwuMwtzANBgkqhkiG9w
MRYwFAYDVQQLEA1BSS5FbnR1cnByaXN1MRIwEAYDVQQDDA1sb2NhbnR1

```

PEM-encoded PT AI server CA certificate chain. You may keep this field empty if accept non-trusted SSL certificates mode used or PT AI server certificate signed by certificate from your JRE truststore.

Accept non-trusted SSL certificates

☐

This option explicitly allows plugin to perform "insecure" SSL connections with PT AI server. All SSL connections are attempted to be made secure by using the CA certificate bundle from JRE truststore or from explicitly defined certificate chain PEM file. This makes all connections considered "insecure" fail unless this option is used.

Save

Test PT AI server connection

Рисунок 23. Настройка плагина

2. В поле **PT AI server URL** укажите адрес PT AI Enterprise Server.
3. В поле **PT AI client token** введите токен доступа для плагинов CI/CD.

4. В поле **PT AI server CA certificates** укажите цепочку SSL-сертификатов удостоверяющего центра, выпустившего SSL-сертификат для PT AI Enterprise Server, или сам сертификат PT AI Enterprise Server (при использовании самоподписанного сертификата) в кодировке Base64 в формате PEM.

Примечание. Если иерархия сертификатов, используемых для выпуска сертификата PT AI Enterprise Server, содержит Root CA и Intermediate CA как показано на рисунке ниже, необходимо выгрузить эти сертификаты в кодировке Base64, затем в текстовом редакторе соединить их в цепочку в обратном порядке: Intermediate → Root, после чего скопировать и вставить полученную цепочку ключей в поле **PT AI server CA certificates**.

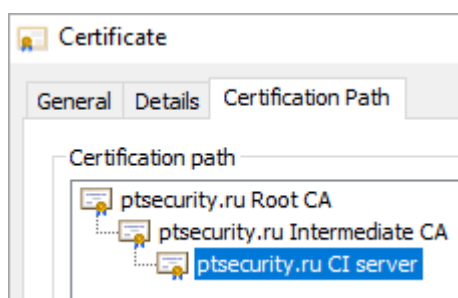


Рисунок 24. Иерархия сертификатов

5. Проверьте соединение с PT AI Enterprise Server по кнопке **Test PT AI server connection**.
6. Если вы не указали цепочку SSL-сертификатов, или она не прошла проверку, установите флажок **Accept non-trusted SSL certificates** для игнорирования проблем с сертификатом.

Примечание. При использовании этой опции сохраняется возможность установки защищенного соединения с PT AI Enterprise Server, но не выполняется проверка принадлежности сертификата, предъявленного сервером в ходе установки SSL-соединения, этому серверу. Поэтому не рекомендуется использовать эту опцию при развертывании в рабочей среде.

7. Нажмите кнопку **Save**.

Плагин настроен. Вы можете перейти к настройке проектной задачи.

8.1.7.2. Настройка проектной задачи

Шаг анализа кода встраивается в существующий процесс сборки. Поскольку эти процессы могут быть устроены по-разному, в инструкции рассматривается пример создания нового проекта с минимальной конфигурацией для проверки кода в PT AI Enterprise Edition.

Если вы работаете с существующим процессом сборки и хотите изменить его, начните выполнение инструкции с шага 7.

Проектная задача создается отдельно для каждого проекта сканирования.

► Чтобы создать проектную задачу:

1. В меню **Administration** нажмите кнопку **Create project**.
2. Выберите **From a repository URL**.
3. В поле **Repository URL** введите адрес хранилища исходного кода сканируемого приложения и нажмите кнопку **Proceed**.

Parent project: * <Root project>

Repository URL: * https://github.com/ptssdl/App01
A VCS repository URL. Supported formats: http(s)://, svn://, git://, etc. as well as URLs in Maven format. ⓘ

Username:
Provide username if access to repository requires authentication.

Password:
Provide password if access to repository requires authentication.

Proceed

Рисунок 25. Создание проекта

4. В раскрывающемся списке **Parent project** выберите название проекта и нажмите кнопку **Proceed**.
5. На открывшейся странице перейдите по ссылке **configure build steps manually**.
6. Добавьте шаги по сборке, юнит-тестированию и загрузке зависимостей, чтобы перед запуском шага анализа кода у вас был гарантированно собранный код и используемые им сторонние компоненты.

Build Steps
In this section you can configure the sequence of build steps to be executed. Each build step is represented by a build runner and provides integration with a specific build or test tool. ⓘ

+ Add build step Reorder build steps Auto-detect build steps

Build Step	Parameters Description
1. build and test	Maven Path to POM: pom.xml Goals: clean install Execute: If all previous steps finished successfully
2. download dependencies	Maven Path to POM: pom.xml Goals: dependency:copy-dependencies -DoutputDirectory="\$WORKSPACE\libs" Execute: If all previous steps finished successfully

Рисунок 26. Добавление шагов в сборку

7. Добавьте в сборку шаг с анализом кода, выбрав в раскрывающемся списке **Runner type** шаг **PT AI**.
8. В раскрывающемся списке **PT AI server connection** выберите:

- **Global scope defined PT AI server config** для использования глобальных параметров плагина, заданных при настройке подключения плагина к PT AI Enterprise Server.
 - **Task scope defined PT AI server config** для использования параметров плагина, заданных для конкретного проекта. Например, если вы хотите настроить подключение проекта к другому PT AI Enterprise Server.
9. Если вы выбрали **Task scope defined PT AI server config**, настройте параметры подключения плагина для конкретного проекта аналогично параметрам подключения плагина к PT AI Enterprise Server.
10. В раскрывающемся списке **Branch at PT AI Server** выберите:
- **From pipeline environment**, если название сканируемой ветки прописано в переменной окружения пайплайна;
 - **Custom branch**, если нужно указать ветку вручную, в этом случае в поле **Branch name** введите ее название.
11. В раскрывающемся списке **Scan settings type** выберите:
- **PT AI UI-defined settings**, если проект сканирование уже настроен в веб-интерфейсе PT AI Enterprise Edition.
 - **JSON-defined settings**, если вы хотите прописать конфигурационный файл и файл политики безопасности на этапе сборки.
- Примечание.** Параметр **BranchName**, указанный в конфигурационном файле при выборе **JSON-defined settings**, имеет более низкий приоритет по сравнению с параметром, указанным в **Branch at PT AI Server**, и будет проигнорирован.
12. Если вы выбрали **PT AI UI-defined settings**, в поле **Project name** введите имя проекта, созданного и настроенного в веб-интерфейсе PT AI Enterprise Edition.

New Build Step: PT AI | ▾

Runner type: PT AI ▾
PT AI AST scan

Step name: PT AI Scan
Optional, specify to distinguish this build step from other steps.

Execute step: 🔗 If all previous steps finished successfully ▾
Specify the step execution policy.

PT AI server connection settings

PT AI server connection Global scope defined PT AI server config ▾
Choose how PT AI server connection settings are defined

General AST settings

Scan settings type PT AI viewer-defined settings ▾
Choose how AST settings are defined

Project name * Java App 01 📄
Project name as it defined in PT AI Viewer UI

AST work mode

Work mode Synchronous mode ▾
Work mode parameter defines CI behaviour after AST job is being started.

Рисунок 27. Настройка проектной задачи, если сканирование настроено в веб-интерфейсе PT AI Enterprise Edition

- Если вы выбрали **JSON-defined settings**, в полях **Scan settings** и **Policy** введите текст конфигурационного файла и файла политики (необязательно) в формате JSON.

PT AI server connection settings	
PT AI server connection	Global scope defined PT AI server conf ▼ Choose how PT AI server connection settings are defined
General AST settings	
Branch at PT AI Server	From pipeline environment ▼ Choose how PT AI branch name are defined
Scan settings type	JSON-defined settings ▼ Choose how AST settings are defined
Scans settings *	Edit JSON settings: <pre>{ "\$schema": "https://json.schemastore.org/aiproj-1.7.json", "Version": "1.7", "ProjectName": "Java App 01", "ProgrammingLanguages": ["Java"] }</pre> Scan settings in JSON format
Policy	[] Project SAST policy in JSON format. If this parameter is empty then SAST policy will be downloaded from PT AI EE server. If you need to scan project without policy use [] value
AST work mode	
Work mode	Synchronous mode ▼ Work mode parameter defines CI behaviour after AST job is being started.

Рисунок 28. Настройка проектной задачи, если сканирование настроено на агенте сборки

14. В раскрывающемся списке **Work mode** выберите:

- **Asynchronous mode** для сканирования в асинхронном режиме. В этом случае TeamCity запустит проверку кода на наличие уязвимостей и продолжит сборку, не дожидаясь завершения проверки.
- **Synchronous mode** для сканирования в синхронном режиме. В этом случае TeamCity запустит проверку кода на наличие уязвимостей и будет ожидать завершения проверки.

15. Если вы выбрали **Synchronous mode**:

- Установите флажок **Fail step if SAST failed**, чтобы TeamCity останавливал сборку при несоответствии сканируемого приложения заданной политике безопасности.
- Установите флажок **Fail step if SAST unstable**, чтобы TeamCity останавливал сборку при возникновении ошибок, не связанных с результатами проверки кода на наличие уязвимостей (например, ошибки в конфигурации, отсутствующая или недействительная лицензия).

16. Задайте параметры для формирования отчета об уязвимостях (только для синхронного режима):

- Установите флажок в зависимости от необходимого формата отчета.
- В поле **File name** введите имя отчета.
- В поле **Template** введите название шаблона отчета. Список возможных значений представлен в веб-интерфейсе PT AI Enterprise Edition в разделе **Шаблоны отчетов**.
- Если требуется, в поле **Vulnerability filter** введите JSON-описание фильтров, необходимых [для формирования отчета \(см. раздел 8.1.9\)](#).

Build step fail conditions	☒ Fail step if SAST failed Mark build step as failed if AST policy assessment failed
	☐ Fail step if SAST unstable Mark build step as failed if AST policy assessment success but there were some minor warnings reported
Reporting: save AST results report file	☒ PT AI will generate human-readable report. You will need to define file name, report template name and optional set of JSON filters.
File name *	tier1.html File name where generated report will be saved to.
Template *	Scan results report Template name that is used for report generation.
Vulnerability filter	<pre>{ "issueLevels": ["MEDIUM", "HIGH"], "scanMode": "FROMENTRYPOINT", }</pre> JSON-defined report generation filter that allows to define what vulnerabilities are to be included into report.
Export: save raw AST results to JSON file	☐ PT AI will export raw AST results data in JSON format. You will need to define file name.
Export: save AST results as SARIF report	☐ PT AI will export AST results data in Static Analysis Results Interchange Format (SARIF) Version 2.1 file. You will need to define file name.
Export: save AST results as SonarQube GILF report	☐ PT AI will export AST results data in SonarQube Generic Issue Import Format (GILF) file. You will need to define file name.

Рисунок 29. Настройка одного отчета

- Если вы хотите сформировать несколько отчетов с помощью JSON-описания, установите флажок **JSON-defined reports**. Синтаксис JSON-описания приведен в разделе «[Файл описания отчетов \(см. раздел 8.1.9\)](#)».

HTML or PDF report	☐ PT AI will generate human-readable report in HTML or PDF formats. You will need to define file name, report template name, format, locale and optional set of JSON filters.
XML or JSON data file	☐ PT AI will export AST results data in JSON or XML formats. You will need to define file name, format, locale and optional set of JSON filters.
Raw AST data export	☐ PT AI will export raw AST results data in JSON format. You will need to define file name.
JSON-defined reports	☒ Advanced reporting settings that allow to use JSON-defined set of reports to be generated.
JSON settings *	<pre> { "report" : [{ "fileName" : "ai_full_report.html", "locale" : "EN", "format" : "HTML", "template" : "Scan results report" }, { "fileName" : "tier1.html", }] } </pre> JSON definition of reports to be generated and filters that are applied to scan results prior to report generation or data export

Рисунок 30. Настройка нескольких отчетов

- В поле **Files to analyze** задайте возможность включения и (или) исключения определенных файлов из сканирования по маске, используя синтаксис [Apache Ant](#).
- В поле **Exclude files** задайте перечень исключаемых из сканирования файлов и папок, используя синтаксис [Apache Ant](#). Исключенные таким способом файлы и каталоги не попадут в PT AI Enterprise Server.

Scan scope	
Files to analyse *	**/* Files to scan for vulnerabilities. The string is a comma separated list of includes for an Ant fileset eg. '**/*.jar' (see Patterns in the Ant manual). The base directory for this fileset is the workspace
Exclude files	.ptai/**,report.*,target/** Exclude files from the Transfer set. The string is a comma separated list of excludes for an Ant fileset eg. '**/*.log, **/*.tmp, .git/' (see Patterns in the Ant manual)

Рисунок 31. Настройка перечня передаваемых на сканирование файлов и папок

- Нажмите кнопку **Show advanced options** для настройки дополнительных параметров:
 - В раскрывающемся списке **Execute step** выберите в каких случаях запускать анализ кода.

- Установите флажок **Execute full AST** для выполнения полного сканирования. Если флажок не установлен, будет запущено инкрементальное сканирование.
- Установите флажок **Verbose log output** для включения расширенного протоколирования действий плагина.
- В поле **Remove prefix** при необходимости укажите начало пути к файлам, которое будет исключено при передаче в PT AI Enterprise Server. Например, если анализируемый код находится в папке `target/deployment/images/**/`, то при исключении `target/deployment` в PT AI Enterprise Server будет передано содержимое папки **deployment** без структуры `target/deployment`.
- В поле **Pattern separator** задайте разделитель для полей **Files to analyze** и **Exclude files**.
- Установите флажок **Use default excludes**, чтобы использовать стандартный список исключений файлов и папок.
- Установите флажок **Flatten files**, чтобы передать все файлы без имен папок, если файлы вашего проекта имеют уникальные имена.

20. Нажмите кнопку **Check AST settings** для проверки введенных параметров.

21. Нажмите кнопку **Save**.

22. Запустите сборку проекта.

По завершении сканирования на странице сборки проекта отобразится код возврата и результат сборки.

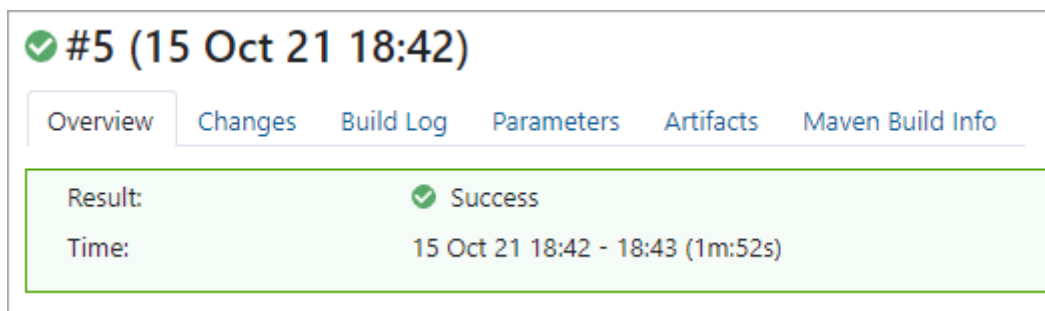


Рисунок 32. Результат сборки

Также на вкладке **Artifacts** в дереве папок **.ptai** вы можете просмотреть отчеты о сканировании.



Рисунок 33. Отчеты о сканировании

8.1.8. Интеграция с другими CI-системами

Для анализа кода в других CI-системах используется кросс-платформенный плагин `AI.Cli.Plugin`. Он устанавливается на узел, где выполняется сборка исходного кода. Для выполнения проверки кода нужно запустить исполняемый файл плагина в оболочке командной строки и передать ему требуемые параметры.

В параметрах плагина вы можете указать параметры сканирования, режим работы (синхронный или асинхронный), перечень файлов для анализа и требования к выпускаемым отчетам.

В синхронном режиме вы можете получать отчеты в точке запуска плагина. Если используется асинхронный режим, вы можете просмотреть результаты сканирования в веб-интерфейсе PT AI Enterprise Edition или сгенерировать отчеты отдельной командой плагина.

Дополнительно в плагине реализована возможность удаления проектов, что может быть использовано для разовых сканирований.

В этом разделе

[Установка плагина \(см. раздел 8.1.8.1\)](#)

[Подготовка команды запуска сканирования \(см. раздел 8.1.8.2\)](#)

[Дополнительные возможности плагина \(см. раздел 8.1.8.3\)](#)

8.1.8.1. Установка плагина

► Чтобы установить плагин AI.Cli.Plugin:

1. Получите плагин в Positive Technologies или соберите его самостоятельно [по инструкции \(см. раздел 8.1.5\)](#).
2. Скопируйте плагин на узел, где он будет запущен.
3. Установите JDK версии 8 или выше.

Примечание. Для запуска плагина можно использовать любую версию JDK, начиная с 8, но для сборки требуется JDK из [инструкции по сборке \(см. раздел 8.1.5\)](#).

4. Обеспечьте безопасное соединение с PT AI Enterprise Server одним из перечисленных способов:
 - Импортируйте цепочку SSL-сертификатов удостоверяющего центра, выпустившего SSL-сертификат для PT AI Enterprise Server, или сам сертификат PT AI Enterprise Server (при использовании самоподписанного сертификата) в хранилище сертификатов Java. Вам потребуются сертификаты в кодировке Base64 (PEM-формат) и утилита keytool (входит в пакет Java JDK).

Примечание. Если иерархия сертификатов, используемых для выпуска сертификата PT AI Enterprise Server, содержит Root CA и Intermediate CA как показано на рисунке ниже, необходимо экспортировать оба этих сертификата в кодировке Base64.

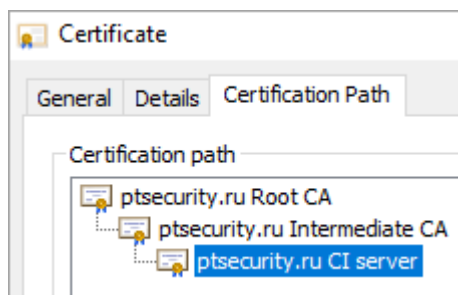


Рисунок 34. Иерархия сертификатов

Импортируйте сертификаты с использованием keytool (пример команды для ОС семейства Unix):

```
keytool -importcert -keystore jdk1.8\jre\lib\security\cacerts -storepass "changeit" -alias AIRootCA -file RootCA.pem -noprompt
keytool -importcert -keystore jdk1.8\jre\lib\security\cacerts -storepass "changeit" -alias AIIntermediateCA -file IntermediateCA.pem -noprompt
```

- Используйте параметр плагина `-truststore` для указания пути к PEM-файлу с цепочкой сертификатов сервера. Аналогично примеру выше вам потребуется экспортировать сертификаты в кодировке Base64, затем в текстовом редакторе соединить их в цепочку в обратном порядке: Intermediate → Root и сохранить в файл. Параметр нужно использовать для каждой команды запуска плагина.

- Используйте параметр плагина `--insecure` для работы без указания сертификата (не рекомендуется при развертывании в рабочей среде). При использовании этого параметра сохраняется возможность установки защищенного соединения с PT AI Enterprise Server, но не выполняется проверка принадлежности сертификата, предъявленного сервером в ходе установки SSL-соединения, этому серверу. Параметр нужно использовать для каждой команды запуска плагина.
5. Проверьте соединение с PT AI Enterprise Server, выполнив команду:


```
java -jar ptai-cli-plugin.jar check-server --url=https://<Адрес PT AI Enterprise Server>
--token=<Ваш токен>
```

8.1.8.2. Подготовка команды запуска сканирования

- Чтобы подготовить команду запуска сканирования:

1. Укажите полный путь к плагину в вызове Java:


```
java -jar ptai-cli-plugin.jar
```
2. Если требуется, укажите название сканируемой ветки в параметре `--branch-name`.
Будет запущено сканирование указанной ветки, если ветка не существует, то она будет создана.
3. Определите способ настройки сканирования одним из параметров:
 - `ui-ast`, если проект сканирования создан в веб-интерфейсе PT AI Enterprise Edition;
 - `json-ast`, если при запуске плагина вы хотите указать конфигурационный файл и файл политики безопасности.
4. Если вы ввели `ui-ast`, в параметре `--project` укажите имя проекта, созданного в PT AI Enterprise Edition.
5. Если вы ввели `json-ast`, в параметрах `--settings-json` и `--policy-json` укажите путь к конфигурационному файлу и файлу политики безопасности (необязательно) в формате JSON.
6. Укажите путь к каталогу с исходным кодом в параметре `--input`.
7. Определите режим сканирования:
 - `--async` для сканирования в асинхронном режиме. В этом случае будет запущена проверка кода на наличие уязвимостей, и сборка продолжится, не дожидаясь завершения проверки.
 - по умолчанию для сканирования в синхронном режиме. В этом случае будет запущена проверка кода на наличие уязвимостей, и сборка не продолжится до тех пор, пока проверка не завершится.
8. Укажите требования к отчету об уязвимостях (только для синхронного режима):
 - Для формирования одного отчета укажите параметры из таблицы ниже.

- Для формирования нескольких отчетов укажите требования в JSON-файле и задайте путь к нему в параметре `--report-json`. Синтаксис файла приведен в разделе «[Файл описания отчетов \(см. раздел 8.1.9\)](#)».
9. Укажите параметры подключения к PT AI Enterprise Server:
 - `--url` — адрес PT AI Enterprise Server.
 - `--token` — токен доступа для плагинов CI/CD.
 10. Укажите условия неуспешного сканирования (только для синхронного режима):
 - Добавьте параметр `--fail-if-failed`, чтобы плагин завершал работу с ошибкой при несоответствии сканируемого приложения заданной политике безопасности.
 - Добавьте параметр `--fail-if-unstable`, чтобы плагин завершал работу с ошибкой при возникновении ошибок, не связанных с результатами проверки кода на наличие уязвимостей (например, ошибки в конфигурации, отсутствующая или недействительная лицензия).
 11. Укажите каталог для сохранения отчетов по результатам сканирования в параметре `--output` (по умолчанию используется каталог **.ptai**).

Примечание. Во всех режимах работы плагин создает файл `rest.url` со ссылкой на результаты сканирования в формате JSON. Для открытия этой ссылки необходимо использовать токен доступа, который можно получить через API.
 12. С помощью параметров `--includes` и `--excludes` добавьте или исключите определенные файлы из сканирования по маске. Используется синтаксис [Apache Ant](#).
 13. Добавьте параметр `--verbose` для включения расширенного протоколирования действий плагина.
 14. Выполните получившуюся команду.

Если все параметры указаны корректно, отобразится уведомление о постановке сканирования в очередь, как показано на рисунке ниже.

```
demo@ptaisrv MINGW64 ~/Desktop
$ java -jar ptai-cli-plugin.jar ui-ast --input="C:\Users\demo\Desktop\app01" --project="Java App 01" --report-json=ai-reports.json --fail-if-failed --excludes=**/test/**,**/assets/** --url=https://ptaisrv.domain.org --token=oXaxuJpE+UuZ+JHRPaEP3KRLcKtjZ5yq
PT AI project ID is b94d7d79-26d7-4861-9c89-b4be7dc7feb9
Folder to collect files from is C:\Users\demo\Desktop\app01
Sources will be zipped to C:\Users\demo\AppData\Local\Temp\2\ptai.3412682704789256382.zip
Scan enqueued, PT AI AST result ID is 1b27c42c-905a-4fde-8e0d-f66a940ecf26
Existing file rest.url will be overwritten
AST result REST API URL: https://ptaisrv.domain.org/api/Projects/b94d7d79-26d7-4861-9c89-b4be7dc7feb9/scanResults/1b27c42c-905a-4fde-8e0d-f66a940ecf26
Existing file ptai.url will be overwritten
AST result PT AI URL: ptai://navigation/show?project=b94d7d79-26d7-4861-9c89-b4be7dc7feb9&result=1b27c42c-905a-4fde-8e0d-f66a940ecf26
Scan started
Initialize -> RestoreScanData 0%
Initialize -> RestoreScanData 1%
Initialize -> RestoreScanData 2%
Initialize -> RestoreScanData 3%
Initialize -> RestoreScanData 4%
Initialize -> RestoreScanData 5%
Initialize -> RestoreScanData 6%
Initialize -> RestoreScanData 7%
Initialize -> RestoreScanData 8%
```

Рисунок 35. Запуск плагина

Примеры команд запуска плагина:

- Проект создан в веб-интерфейсе PT AI Enterprise Edition, синхронный блокирующий запуск с генерацией отчетов:

```
java -jar ptai-cli-plugin.jar ui-ast --input="/home/folder" --project="ProjectName" --report-json=ai-reports.json --fail-if-failed --excludes=**/test/**,**/assets/** --url=https://ptaisrv.domain.org --token=yourtoken
```

- Проект создается на основе файла конфигурации, асинхронный запуск:

```
java -jar ptai-cli-plugin.jar json-ast --input="/home/folder" --settings-json=ai-settings.json --policy-json=ai-policy.json --async --excludes=**/test/**,**/assets/** --url=https://ptaisrv.domain.org --token=yourtoken
```

Таблица 26. Параметры для создания отчета

Формат отчета	Параметр	Возможные значение
HTML	--report-template	Список возможных значений представлен в веб-интерфейсе PT AI Enterprise Edition в разделе Шаблоны отчетов
	--report-file	Имя файла
	--report-include-dfd	True или false
	--report-include-glossary	True или false
SARIF	--sarif-report-file	Имя файла
SonarGiif	--giif-report-file	Имя файла
Необработанные данные в формате JSON	--raw-data-file	Имя файла

8.1.8.3. Дополнительные возможности плагина

Плагин AI.Cli.Plugin поддерживает дополнительные возможности работы с PT AI Enterprise Server.

Отображение списка доступных шаблонов отчетов

- Чтобы получить список шаблонов отчетов от PT AI Enterprise Server:

1. Укажите полный путь к плагину в вызове Java и параметр для получения шаблонов `list-report-templates`:

```
java -jar ptai-cli-plugin.jar list-report-templates
```

2. Укажите язык шаблонов с помощью параметра `--locale` (RU или EN).
3. Укажите параметры для подключения к PT AI Enterprise Server:
 - `--url` — адрес PT AI Enterprise Server;
 - `--token` — токен доступа для плагинов CI/CD.

Пример команды на получение списка шаблонов отчетов:

```
java -jar ptai-cli-plugin.jar list-report-templates --locale=RU --url=https://ptaisrv.domain.org --token=yourtoken
```

Создание отчетов по результатам сканирования

- Чтобы создать отчет по результатам сканирования в асинхронном режиме:

1. Укажите полный путь к плагину в вызове Java и параметр для создания отчетов `generate-report`:

```
java -jar ptai-cli-plugin.jar generate-report
```
2. Укажите идентификатор проекта (`--project-id`) или его имя (`--project-name`).

Примечание. Плагин генерирует файл `rest.url`, который содержит идентификатор проекта для автоматической подстановки этого параметра в скриптах.

3. Если требуется создать отчет о сканировании конкретной ветки, укажите ее в параметре `--branch-name`.
4. Укажите идентификатор результата сканирования (`--scan-result-id`).

Примечание. Вы можете скопировать его из вывода плагина при постановке задачи сканирования в очередь, либо из файла `rest.url`. Если параметр не указан, будет выгружен результат последнего сканирования указанного проекта.

5. Укажите параметры для создания отчета, описанные на шаге 7 раздела «Подготовка команды запуска плагина (см. раздел 8.1.8.2)».
6. Укажите параметры для подключения к PT AI Enterprise Server:
 - `--url` — адрес PT AI Enterprise Server;
 - `--token` — токен доступа для плагинов CI/CD.

Пример команды на создание отчетов:

```
java -jar ptai-cli-plugin.jar generate-report --project-id=projectUUID --scan-result-id=resultUUID --report-json=ai-report.json --url=https://ptaisrv.domain.org --token=yourtoken
```

Удаление проекта в PT AI Enterprise Server

► Чтобы удалить один или несколько проектов:

1. Укажите полный путь к плагину в вызове Java и параметр для удаления `delete-project`:
`java -jar ptai-cli-plugin.jar delete-project`
2. Укажите один из параметров:
 - `--project-id` — идентификатор проекта;
 - `--project-name` — название проекта;
 - `--project-name-regexp` — регулярное выражение для поиска имен.
3. Если требуется, введите `--yes` для автоматического подтверждения всех запросов на удаление проектов.

Примечание. Рекомендуется использовать этот параметр только в неинтерактивном режиме.

4. Укажите параметры подключения к PT AI Enterprise Server:
 - `--url` — адрес PT AI Enterprise Server;
 - `--user` — имя пользователя;
 - `--password` — пароль пользователя.

Пример команды на удаление нескольких проектов:

```
java -jar ptai-cli-plugin.jar delete-project --project-name-regexp=test.* --  
user=administrator --password=P@ssw0rd --url=https://ptaisrv.domain.org
```

8.1.9. Файл описания отчетов

Плагины PT AI Enterprise Edition позволяют создавать отчеты в разных форматах по завершении анализа кода в синхронном режиме. Требования к отчетам описываются в конфигурационном файле в формате JSON.

Конфигурационный файл состоит из четырех секций, ни одна из секций не является обязательной и может быть пропущена:

- `report: [...]` — отчет в форматах HTML;
- `sarif: [...]` — отчет в формате SARIF;
- `sonarGiif: [...]` — отчет для SonarQube;
- `raw: [...]` — необработанные данные в формате JSON.

Таблица 27. Описание секции report

Параметр	Описание
fileName	Название файла с отчетом
includeDfd	Добавить в отчет диаграмму потоков данных (по умолчанию true)
includeGlossary	Добавить в отчет справочную информацию об уязвимостях (по умолчанию true)
template	Шаблон отчета (см. веб-интерфейс PT AI Enterprise Edition)
filters	Фильтры (см. таблицу ниже)

Таблица 28. Описание секций sarif, sonarGiif и raw

Параметр	Описание
fileName	Название файла с отчетом
filters	Фильтры (см. таблицу ниже)

Таблица 29. Фильтры

Параметр	Описание
issueLevel	Уровень опасности уязвимости: — ALL — все уровни (по умолчанию); — LOW — низкий; — MEDIUM — средний; — HIGH — высокий; — POTENTIAL — потенциальные уязвимости
issueLevels	Уровень опасности уязвимости, несколько значений
exploitationCondition	Способ эксплуатации уязвимости: — ALL — все (по умолчанию); — NOCONDITION — без условия; — UNDERCONDITION — с условием
scanMode	Режим сканирования: — ALL — все (по умолчанию); — FROMENTRYPOINT — от точек входа; — FROMPUBLICPROTECTED — от методов public и protected; — FROMOTHER — другие режимы

Параметр	Описание
scanModes	Режим сканирования, несколько значений
suppressStatus	Исключенные уязвимости: — EXCEPTSUPPRESSED — неисключенные (по умолчанию); — SUPPRESSED — исключенные; — ALL — все
confirmationStatus	Статус подтверждения: — UNDEFINED — без статуса (по умолчанию); — APPROVED — подтвержденные (по умолчанию); — AUTOAPPROVED — автоматически подтвержденные (по умолчанию); — DISCARDED — опровергнутые; — ALL — все
confirmationStatuses	Статус подтверждения, несколько значений
sourceType	Метод анализа: — ALL — все (по умолчанию); — STATIC — анализ исходного кода; — BLACKBOX — черный ящик
actualStatus	Актуальность: — ALL — любой статус (по умолчанию); — ISNEW — новые; — NOTISNEW — из предыдущих сканирований
hideSecondOrder	Уязвимости второго порядка: true или false
hideSuspected	Подозрения на уязвимость: true или false
byBestPlaceToFix	Без места исправления: true или false

Пример файла с описанием отчетов в разных форматах:

```
{
  "report": [ {
    "fileName": "report.ru.html",
    "includeDfd": true,
    "includeGlossary": true,
    "template": "Отчет PCI DSS 3.2",
    "filters": {
      "issueLevel": "ALL",
```

```

        "exploitationCondition": "ALL",
        "scanMode": "ALL",
        "suppressStatus": "ALL",
        "confirmationStatus": "ALL",
        "sourceType": "ALL"
    }
} ],
"sarif": [ {
    "fileName": "report.sarif",
    "filters": {
        "issueLevel": "HIGH"
    }
} ],
"sonarGiif": [ {
    "fileName": "sonarGiif.json",
    "filters": {
        "issueLevel": "HIGH"
    }
} ],
"raw": [ {
    "fileName": "raw.json",
    "filters": {
        "issueLevel": "HIGH"
    }
} ]
}

```

Пример файла с описанием отчетов об уровне опасности, новых уязвимостях, уязвимых компонентах и уязвимостях от ядра BlackBox в формате HTML:

```

{
  "report": [ {
    "fileName": "tier1.html",
    "includeDfd": true,
    "includeGlossary": true,
    "template": "Отчет по результатам сканирования",
    "filters": {
      "issueLevels": ["MEDIUM", "HIGH"],
      "scanMode": "FROMENTRYPOINT",
      "confirmationStatuses": ["UNDEFINED", "NONE", "APPROVED", "AUTOAPPROVED"]
    }
  }, {
    "fileName": "tier2.html",
    "includeDfd": true,
    "includeGlossary": true,
    "template": "Отчет по результатам сканирования",
    "filters": {

```

```

        "issueLevels": ["MEDIUM", "HIGH"],
        "scanMode": "FROMPUBLICPROTECTED",
        "confirmationStatuses": ["UNDEFINED", "NONE", "APPROVED", "AUTOAPPROVED"]
    }
}, {
    "fileName": "tier3.html",
    "includeDfd": true,
    "includeGlossary": true,
    "template": "Отчет по результатам сканирования",
    "filters": {
        "issueLevels": ["LOW", "POTENTIAL"],
        "confirmationStatuses": ["UNDEFINED", "NONE", "APPROVED", "AUTOAPPROVED"],
        "sourceType": "STATIC"
    }
}, {
    "fileName": "new.html",
    "includeDfd": true,
    "includeGlossary": true,
    "template": "Отчет по результатам сканирования",
    "filters": {
        "actualStatus": "ISNEW",
        "confirmationStatuses": ["UNDEFINED", "NONE", "APPROVED", "AUTOAPPROVED"]
    }
}, {
    "fileName": "vulnerable components.html",
    "includeDfd": true,
    "includeGlossary": true,
    "template": "Отчет по результатам сканирования",
    "filters": {
        "issueLevels": ["MEDIUM", "HIGH"],
        "confirmationStatuses": ["UNDEFINED", "NONE", "APPROVED", "AUTOAPPROVED"],
        "scanMode": "FROMOTHER",
        "sourceType": "STATIC"
    }
}, {
    "fileName": "blackbox.html",
    "includeDfd": true,
    "includeGlossary": true,
    "template": "Отчет по результатам сканирования",
    "filters": {
        "confirmationStatuses": ["UNDEFINED", "NONE", "APPROVED", "AUTOAPPROVED"],
        "sourceType": "BLACKBOX"
    }
} ]
}

```

8.2. Интеграция PT AI Enterprise Edition со средами разработки

Для интеграции PT AI Enterprise Edition со средами разработки предусмотрены два плагина PT Application Inspector: для IntelliJ IDEA и Visual Studio Code.

Вы можете использовать плагин для локального сканирования исходного кода и работы с найденными уязвимостями в привычной среде разработки или синхронизировать проекты в IDE и в PT AI Enterprise Server, что позволит нескольким людям взаимодействовать с одним и тем же исходным кодом из разных интерфейсов. Например, разработчик запустит сканирование и присвоит найденным уязвимостям статус в своей локальной IDE, а менеджер безопасности на основании присвоенных разработчиком статусов выпустит отчет в веб-интерфейсе PT AI Enterprise Edition.

В этом разделе

[Интеграция с IntelliJ IDEA \(см. раздел 8.2.1\)](#)

[Интеграция с Visual Studio Code \(см. раздел 8.2.2\)](#)

8.2.1. Интеграция с IntelliJ IDEA

Для осуществления интеграции PT AI Enterprise Edition с IntelliJ IDEA необходимо предварительно установить плагин и анализатор кода PT Application Inspector, а затем подключить плагин к PT AI Enterprise Server.

С помощью плагина вы можете загружать исходный код и отправлять на сканирование проект из IntelliJ IDEA в PT AI Enterprise Server, а также синхронизировать результаты сканирований.

В этом разделе

[Установка плагина и анализатора кода \(см. раздел 8.2.1.1\)](#)

[Подключение плагина к PT AI Enterprise Server \(см. раздел 8.2.1.2\)](#)

[Синхронизация проектов \(см. раздел 8.2.1.3\)](#)

[Запуск сканирования в PT AI Enterprise Server \(см. раздел 8.2.1.4\)](#)

[Отправка исходного кода в PT AI Enterprise Server \(см. раздел 8.2.1.5\)](#)

[Синхронизация результатов сканирования \(см. раздел 8.2.1.6\)](#)

8.2.1.1. Установка плагина и анализатора кода

► Чтобы установить плагин:

1. В главном меню IntelliJ IDEA выберите **File** → **Settings** → **Plugins**.
2. Выберите вкладку **Marketplace** и в строке поиска введите **PT Application Inspector**.
3. Нажмите **Install**.

► Чтобы установить анализатор кода:

1. Откройте папку с исходным кодом проекта.
2. Активируйте плагин по кнопке **Enable** во всплывающем уведомлении.
3. Иницилируйте загрузку анализатора по кнопке **Download analyzer** во всплывающем уведомлении.
4. Дождитесь окончания загрузки и установки.

8.2.1.2. Подключение плагина к PT AI Enterprise Server

► Чтобы подключить плагин к PT AI Enterprise Server:

1. В главном меню выберите **Tools** → **PT Application Inspector** → **Connect to PT AI Server**.
2. Введите URL PT AI Enterprise Server.
3. Нажмите **Connect**.

Откроется страница входа в веб-интерфейс PT AI Enterprise Edition.

4. Выполните вход через настроенную у вас SSO-систему.

Откроется страница **Успешный вход**.

5. Вернитесь в IntelliJ IDEA.

В интерфейсе IntelliJ IDEA отобразится уведомление о подключении плагина к PT AI Enterprise Server.

Примечание. Если подключение не произошло, вернитесь на страницу **Успешный вход** и выполните приведенную там инструкцию.

8.2.1.3. Синхронизация проектов

После настройки подключения плагина к PT AI Enterprise Server вы можете настроить синхронизацию между локальным проектом в IntelliJ IDEA и проектом в PT AI Enterprise Server. Она необходима для параллельной работы пользователей с проектами в IntelliJ IDEA и в веб-интерфейсе PT AI Enterprise Edition.

► Чтобы подключить локальный проект к удаленному в PT AI Enterprise Server:

1. В IntelliJ IDEA в меню **Tools** → **PT Application Inspector** выберите **Manage Projects**.
2. Выберите проект из PT AI Enterprise Server, к которому нужно подключиться, и нажмите **Map**.
3. Выберите ветку в проекте в PT AI Enterprise Server и нажмите **Confirm**.
4. В окне **Project Synchronization** выберите вариант синхронизации статусов найденных уязвимостей и результатов сканирования. Для использования данных из PT AI Enterprise Server нужно выбрать вариант **Server**, для сохранения локальных статусов уязвимостей и параметров проекта — вариант **Local**.
5. Нажмите **Synchronize**.

Проекты синхронизируются, а в нижней части окна IntelliJ IDEA отобразится кнопка с названием основной ветки. По кнопке вы можете переключиться на другую ветку.

Также в плагине реализована функциональность отслеживания изменений в локальном Git-репозитории. При создании новой ветки вам будет предложено сопоставить ее с веткой на сервере, нажав соответствующую кнопку в уведомлении.

8.2.1.4. Запуск сканирования в PT AI Enterprise Server

После синхронизации проектов вы можете опраивать локальный проект на сканирование в PT AI Enterprise Server с сохранением результатов на сервере или без сохранения. В первом случае результат будет сохранен в подключенном проекте в PT AI Enterprise Edition. Во втором случае в PT AI Enterprise Edition будет создан временный проект, который удалится сразу после получения результатов сканирования в плагине.

► Чтобы запустить сканирование в PT AI Enterprise Server:

1. В IntelliJ IDEA в панели **PT Application Inspector** нажмите .
2. Выберите вариант запуска сканирования:
 - **With Results Saved on Server**. Для выполнения операции вам необходимо обладать полномочиями на создание проекта и управление сканированиями в PT AI Enterprise Edition.
 - **Without Saving Results on Server**. Для выполнения операции вам необходимо обладать полномочием на сканирование из плагинов IDE без сохранения результатов на сервере.

Проект будет отправлен в PT AI Enterprise Server, по завершении сканирования найденные уязвимости загрузятся в IntelliJ IDE.

8.2.1.5. Отправка исходного кода в PT AI Enterprise Server

Если вы хотите отложить запуск сканирования проекта в PT AI Enterprise Server или в текущий момент у вас нет свободных агентов сканирования, вы можете отправить исходный код локального проекта на сервер без запуска сканирования. Код будет загружен в подключенный проект, а запустить его сканирование можно будет позже в интерфейсе PT AI Enterprise Edition.

- ▶ Чтобы отправить исходный код в PT AI Enterprise Server,
в IntelliJ IDEA в меню **Tools** → **PT Application Inspector** выберите **Send Source Code to PT AI Server**.

8.2.1.6. Синхронизация результатов сканирования

При параллельной работе пользователей с проектом в IntelliJ IDEA и в интерфейсе PT AI Enterprise Edition необходимо периодически синхронизировать результаты сканирований, чтобы поддерживать их актуальность в обеих средах. Вы можете как отправлять результаты локального сканирования в PT AI Enterprise Server, так и скачивать результаты удаленного сканирования из PT AI Enterprise Server.

- ▶ Чтобы синхронизировать результаты сканирования,
в IntelliJ IDEA в меню **Tools** → **PT Application Inspector** выберите вариант синхронизации:
 - **Send Scan Results to PT AI Server;**
 - **Fetch Scan Results from PT AI Server.**В зависимости от выбранного варианта синхронизации отобразится соответствующее уведомление.

8.2.2. Интеграция с Visual Studio Code

Для осуществления интеграции PT AI Enterprise Edition с Visual Studio Code необходимо предварительно установить плагин и анализатор кода PT Application Inspector, а затем подключить плагин к PT AI Enterprise Server.

С помощью плагина вы можете запускать загружать исходный код и отправлять проект из Visual Studio Code на сканирование в PT AI Enterprise Server, а также синхронизировать результаты сканирований.

В этом разделе

[Установка плагина и анализатора кода \(см. раздел 8.2.2.1\)](#)

[Подключение плагина к PT AI Enterprise Server \(см. раздел 8.2.2.2\)](#)

[Синхронизация проектов \(см. раздел 8.2.2.3\)](#)

[Запуск сканирования в PT AI Enterprise Server \(см. раздел 8.2.2.4\)](#)

[Отправка исходного кода в PT AI Enterprise Server \(см. раздел 8.2.2.5\)](#)

[Синхронизация результатов сканирования \(см. раздел 8.2.2.6\)](#)

8.2.2.1. Установка плагина и анализатора кода

Вы можете загрузить плагин с сайта marketplace.visualstudio.com или из VSIX-файла, предоставляемого по запросу.

► Чтобы установить плагин:

1. В Visual Studio Code выполните команду:
`Extensions: Install from VSIX`
2. Выберите файл плагина на своем компьютере и нажмите **Install**.

► Чтобы установить анализатор кода:

1. Откройте папку с исходным кодом проекта.
2. Активируйте плагин по кнопке **Enable** во всплывающем уведомлении.
3. Запустите загрузку анализатора по кнопке **Download analyzer** во всплывающем уведомлении или с помощью команды `PT Application Inspector: Download analyzer`.
4. Дождитесь окончания загрузки и установки.

8.2.2.2. Подключение плагина к PT AI Enterprise Server

► Чтобы подключить плагин к PT AI Enterprise Server:

1. В Visual Studio Code выполните одно из следующих действий:
 - введите команду `PT Application Inspector: Connect to PT AI Server`;
 - нажмите **Connect to PT AI Server** в блоке **PT AI Server**.
2. Введите URL PT AI Enterprise Server и нажмите Enter.
Откроется страница входа в веб-интерфейс PT AI Enterprise Edition.
3. Выполните вход через настроенную у вас SSO-систему.
Откроется страница **Успешный вход**, а затем вы будете перенаправлены в Visual Studio Code.

Примечание. Если перенаправление не произошло, выполните инструкцию на странице **Успешный вход**.

8.2.2.3. Синхронизация проектов

После настройки подключения плагина к PT AI Enterprise Server вы можете настроить синхронизацию между локальным проектом в Visual Studio Code и проектом в PT AI Enterprise Server, она необходима для параллельной работы пользователей с проектами в Visual Studio Code и в веб-интерфейсе PT AI Enterprise Edition.

► Чтобы подключить локальный проект к удаленному в PT AI Enterprise Server:

1. В Visual Studio Code выполните одно из следующих действий:
 - введите команду `PT Application Inspector: Manage Projects`;
 - нажмите **Manage Projects** в блоке **PT AI SERVER**.
2. Выберите проект из PT AI Enterprise Server, к которому нужно подключиться.
3. Выберите `PT AI: Map PT AI Project`.
4. Выберите ветку в проекте в PT AI Enterprise Server.
5. Выберите вариант синхронизации статусов найденных уязвимостей. Для использования данных из PT AI Enterprise Server нужно выбрать вариант **Server**, для сохранения локальных статусов уязвимостей — вариант **Local**.
6. Выберите вариант синхронизации результатов сканирования аналогично статусам.

Проекты синхронизируются, а в строке состояния Visual Studio Code отобразится основная ветка. Переключиться на другую ветку вы можете по кнопке **Switch branch** в блоке **PT AI SERVER** или командой `PT Application Inspector: Change PT AI Server Project Branch`.

Также в плагине реализована функциональность отслеживания изменений в локальном Git-репозитории. При создании новой ветки вам будет предложено сопоставить ее с веткой на сервере, нажав соответствующую кнопку в уведомлении.

8.2.2.4. Запуск сканирования в PT AI Enterprise Server

После синхронизации проектов вы можете опрашивать локальный проект на сканирование в PT AI Enterprise Server с сохранением результатов на сервере или без сохранения. В первом случае результат будет сохранен в подключенном проекте в PT AI Enterprise Edition. Во втором случае в PT AI Enterprise Edition будет создан временный проект, который удалится сразу после получения результатов сканирования в плагине.

► Чтобы запустить сканирование в PT AI Enterprise Server,

в Visual Studio Code выполните одну из команд:

`PT Application Inspector: Scan Remotely and Save Results on Server`

Для выполнения операции вам необходимо обладать полномочиями на создание проекта и управление сканированиями в PT AI Enterprise Edition.

PT Application Inspector: Scan Remotely Without Saving Results on Server

Для выполнения операции вам необходимо обладать полномочием на сканирование из плагинов IDE без сохранения результатов на сервере.

Проект будет отправлен на сканирование в PT AI Enterprise Server, по завершении сканирования найденные уязвимости загрузятся в Visual Studio Code.

8.2.2.5. Отправка исходного кода в PT AI Enterprise Server

Если вы хотите отложить запуск сканирования проекта в PT AI Enterprise Server или в текущий момент у вас нет свободных агентов сканирования, вы можете отправить исходный код локального проекта на сервер без запуска сканирования. Код будет загружен в подключенный проект, а запустить его сканирование можно будет позже в интерфейсе PT AI Enterprise Edition.

- ▶ Чтобы отправить исходный код в PT AI Enterprise Server,

в Visual Studio Code выполните команду:

PT Application Inspector: Push source code to PT AI Server

8.2.2.6. Синхронизация результатов сканирования

При параллельной работе пользователей с проектом в Visual Studio Code и в интерфейсе PT AI Enterprise Edition необходимо периодически синхронизировать результаты сканирований, чтобы поддерживать их актуальность в обеих средах. Вы можете как отправлять результаты локального сканирования в PT AI Enterprise Server, так и скачивать результаты удаленного сканирования из PT AI Enterprise Server.

- ▶ Чтобы синхронизировать результаты сканирования,

в Visual Studio Code выполните нужную команду:

PT Application Inspector: Pull Scan Results from PT AI Server

или

PT Application Inspector: Push Scan Results to PT AI Server

В зависимости от выбранного варианта синхронизации отобразится соответствующее уведомление.

8.3. Интеграция с системой отслеживания ошибок Jira

Для интеграции PT AI Enterprise Edition с Jira вы можете использовать существующий в Jira проект или создать новый. В этот проект будут поступать задачи на исправление уязвимостей от PT AI Enterprise Edition.

В проекте в Jira необходимо создать пользовательское поле, в котором будет отображаться ссылка на уязвимость в PT AI Enterprise Edition. Ссылку PT AI Enterprise Edition создает автоматически при создании задачи в карточке уязвимости.

► Чтобы создать пользовательское поле в Jira:

1. В интерфейсе Jira выберите существующий проект или создайте новый.
2. Откройте страницу **Настройки проекта**.
3. В левой части страницы выберите раздел **Поля**.
4. По значку  в центральной части страницы откройте страницу **Администрирование**.
5. В левой части страницы выберите раздел **Настраиваемые поля**.
6. По кнопке **Добавить пользовательское поле** откройте окно добавления поля.
7. Выберите из списка типов полей **URL**.
8. Нажмите **Далее**.
9. В окне **Настройка URL** в поле **Имя** введите Application Inspector Link.

Примечание. Убедитесь, что вы правильно ввели текст Application Inspector Link. В противном случае ссылка на уязвимость в PT AI Enterprise Edition не будет отображаться в Jira.

10. Если необходимо, добавьте описание поля.
11. Нажмите **Создать**.
12. На странице **Администрирование** в разделе **Экраны** настройте отображение созданного поля на необходимых экранах (например, на экранах задач и ошибок).
13. Нажмите **Обновить**.

9. Использование REST API

Для расширения возможностей взаимодействия с продуктом, например, создания и настройки проектов, а также запуска сканирований предусмотрена интеграция через REST API. Для работы с REST API PT AI Enterprise Edition вы можете использовать устанавливаемый вместе с PT AI Enterprise Server сервис Swagger UI. Он доступен по адресу `https://<Адрес PT AI Enterprise Server>/swagger` и поддерживает текущую и предыдущую версии API.

В этом разделе

[Аутентификация по REST API \(см. раздел 9.1\)](#)

[Запуск сканирования проекта \(см. раздел 9.2\)](#)

[Перенос результатов сканирования между серверами PT AI Enterprise Edition \(см. раздел 9.3\)](#)

[Интеграция с сервисом уведомлений \(см. раздел 9.4\)](#)

9.1. Аутентификация по REST API

► Чтобы выполнить аутентификацию по REST API:

1. [Создайте и сохраните токен доступа для легкого агента и плагинов CI/CD \(см. раздел 7.5.1\).](#)
2. Выполните запрос `GET /auth/signin`, передав токен, полученный на предыдущем шаге, в заголовке `Access-Token`:

```
GET <Корневой URL API>/auth/signin
Headers:
  Access-Token: <initialAccessToken>
```

где:

`<Корневой URL API>` — `https://<Адрес или доменное имя сервера, на котором установлен PT AI Enterprise Edition>/api/<Номер версии API>`.

В ответ на успешный запрос сервис вернет токен доступа `accessToken` и токен обновления доступа `refreshToken`.

3. Для всех последующих запросов используйте токен доступа в заголовке `Authorization` в формате:

```
Authorization: Bearer <accessToken>
```

4. По истечении времени жизни токена, когда в ответ на запросы будет возвращаться ошибка 401 с заголовком `WWW-Authenticate`, обновите токен с помощью запроса:

```
GET <Корневой URL API>/auth/refreshToken
```

передав в заголовке `Authorization` токен обновления доступа:

```
Authorization: Bearer <refreshToken>
```

В ответ на успешный запрос сервис вернет новый токен доступа `accessToken`.

Примечание. Если на шаге 1 создан токен с ограниченным сроком действия, то время жизни токена обновления доступа также будет ограничено. В таком случае при получении ошибки 401 с заголовком WWW-Authenticate необходимо заново создать токен доступа (шаг 1) и запросить новые accessToken и refreshToken (шаг 2).

9.2. Запуск сканирования проекта

► Чтобы запустить сканирование проекта через REST API с выбором ветки:

1. Выполните запрос на создание проекта:

- С указанием полного набора параметров:

POST <Корневой URL API>/api/projects/extended

- С указанием минимального набора параметров:

POST <Корневой URL API>/api/projects/base

2. Выполните запрос на создание ветки:

- С загрузкой нескольких файлов:

POST <Корневой URL API>/api/store/project/{projectId}/branches

- С загрузкой архива:

POST <Корневой URL API>/api/store/project/{projectId}/branches/archive

3. Выполните запрос на запуск сканирования:

- Основной ветки:

POST <Корневой URL API>/api/scans/{projectId}/start

- Указанной ветки:

POST <Корневой URL API>/api/scans/branches/{branchId}/start

4. Если требуется, выполните запрос на обновление исходного кода проекта:

- Для основной ветки:

POST <Корневой URL API>/api/store/{projectId}/sources

- Для указанной ветки:

POST <Корневой URL API>/api/store/{projectId}/branches/{branchId}/sources

9.3. Перенос результатов сканирования между серверами PT AI Enterprise Edition

Начиная с версии 5.1.0, если PT AI Enterprise Edition развернут в двух изолированных контурах (например, в открытом контуре выполняется разработка приложения и анализ кода с помощью PT AI Enterprise Edition, а в закрытом — постобработка полученных результатов анализа), вы можете экспортировать из открытого контура в закрытый следующие данные:

- найденные уязвимости;
- результаты разбора уязвимостей (статусы, комментарии, добавление в избранные);
- некоторые параметры сканирования.

Перенос осуществляется с помощью запросов к REST API и имеет следующие ограничения:

- в рамках одного экспорта/импорта через архив можно перенести до 100 результатов сканирований (как полных, так и инкрементальных) одной ветки проекта;
- во избежание расхождения в форматах данных на обоих серверах должна быть установлена одинаковая версия PT AI Enterprise Edition, за исключением патчей;
- в рамках проекта не экспортируются:
 - объект сканирования (исходный код);
 - параметры черного ящика;
 - ссылки на задачи в Jira;
 - информация о пользователях;
 - правила анализа.

► Чтобы выполнить перенос данных между серверами PT AI Enterprise Edition:

1. В PT AI Enterprise Server-получателе данных создайте проект с веткой, в которую будут импортированы результаты сканирования, и загрузите в нее исходный код.
2. Завершите все сканирования проекта в PT AI Enterprise Server-отправителе данных.

Примечание. В процессе экспорта/импорта запуск сканирования на обоих серверах недоступен.

3. Выполните запрос на экспорт архива с проектом, а также результатами сканирования и разбора уязвимостей:

```
GET <Корневой URL API>/api/branches/{branchId}/exportTriage
```

4. Выполните запрос на импорт полученных данных в созданную на первом шаге ветку:

```
POST <Корневой URL API>/api/branches/{branchId}/importTriage
```

В ответ на запрос будет возвращен отчет, содержащий информацию о перенесенных данных. В интерфейсе PT AI Enterprise Edition в панели **История сканирования** будет отображаться информация о выполненном импорте.

9.4. Интеграция с сервисом уведомлений

Сервис уведомлений в PT AI Enterprise Edition предназначен для обмена сообщениями между PT AI Enterprise Server и клиентами в реальном времени по протоколу WebSocket.

Для интеграции с сервисом уведомлений необходимо установить клиентскую библиотеку ASP.NET Core SignalR версии не ниже 8.0.7. Клиентские библиотеки для некоторых языков доступны на сайте microsoft.com.

Общий сценарий настройки сервиса выглядит следующим образом:

1. Получение токена доступа.
2. Подключение к PT AI Enterprise Server.
3. Подписка на системное сообщение NeedRefreshToken.
4. Подписка на сообщения, необходимые для работы клиентской логики.

Получение токена доступа

Сервис уведомлений проверяет права клиента на обмен сообщениями с помощью токена доступа (JWT), который передается при подключении к серверу. Чтобы получить токен, необходимо выполнить соответствующий запрос к REST API PT AI Enterprise Edition.

Примечание. Для работы с REST API PT AI Enterprise Edition предусмотрен сервис Swagger UI, устанавливаемый вместе с продуктом. Он доступен по адресу `https://<Адрес PT AI Enterprise Server>/swagger`.

В корректно настроенных официальных клиентах SignalR параметр с токеном доступа заполняется автоматически. Если вы работаете через сокеты напрямую, то токен нужно передавать в строке запроса `access_token` или в заголовке запроса в поле `Authorization`.

Подключение к PT AI Enterprise Server

Подключение к сервису уведомлений осуществляется по адресу: `<URI PT AI Enterprise Server>/notifyApi/notifications?clientId=GUID` для идентификации клиента. Параметр `clientId` обязательный, в качестве значения рекомендуется использовать генерируемое на клиенте случайное значение GUID.

Сразу после подключения необходимо подписаться на сообщение NeedRefreshToken. Это сообщение будет приходить от сервера в случае, если токен истек и его нужно обновить, выполнив соответствующий запрос к REST API PT AI Enterprise Edition и заменив токен в параметрах подключения к PT AI Enterprise Server.

Примечание. Из-за возможных разрывов соединения рекомендуется предварительно настроить параметры автоматического подключения к PT AI Enterprise Server и установить повторное соединение.

Подписка на сообщения

PT AI Enterprise Server отправляет сообщения клиентам только по подписке. При отправке сообщений дополнительно учитываются права пользователя в соответствии с токеном доступа. Пользователь получит сообщение, только если у него есть соответствующие права.

Для включения подписки необходимо отправить на сервер [сообщение SubscribeOnNotification \(см. раздел 9.4.1\)](#). При подписке можно указать идентификаторы сущностей, сообщения от которых вы хотите получать, например идентификаторы конкретных проектов. Для фильтрации сообщений по идентификаторам на сервере предусмотрены соответствующие правила подписки.

Входящие и исходящие сообщения имеют формат JSON. У каждого сообщения есть уникальный идентификатор GUID.

В этом разделе

[Исходящие сообщения от клиентов \(см. раздел 9.4.1\)](#)

[Входящие сообщения: проекты \(см. раздел 9.4.2\)](#)

[Входящие сообщения: результаты сканирований \(см. раздел 9.4.3\)](#)

[Входящие сообщения: аутентификация \(см. раздел 9.4.4\)](#)

[Входящие сообщения: полномочия \(см. раздел 9.4.5\)](#)

[Входящие сообщения: параметры сервера \(см. раздел 9.4.6\)](#)

[Входящие сообщения: планировщик сканирований \(см. раздел 9.4.7\)](#)

[Входящие сообщения: отчеты \(см. раздел 9.4.8\)](#)

[Входящие сообщения: результат отправки тестового сообщения \(см. раздел 9.4.9\)](#)

9.4.1. Исходящие сообщения от клиентов

PT AI Enterprise Server отправляет сообщения клиентам только по подписке. Для включения подписки необходимо отправить на сервер сообщение `SubscribeOnNotification`, указав идентификаторы сущностей, сообщения от которых вы хотите получать.

SubscribeOnNotification

Подписка на получение сообщений определенного типа.

Формат данных:

```
{
  "notificationTypeName": "string, тип сообщений, на который нужно подписаться",
  "ids": "GUID[], идентификаторы сущностей, по которым нужно получать сообщения",
  "createdDate": "DateTime, дата создания подписки"
}
```

UnsubscribeOnNotification

Отписка от получения сообщений.

Формат данных:

```
{
  "notificationTypeName": "string, тип сообщений, от которого нужно отписаться",
  "ids": "GUID[], идентификаторы сущностей, по которым больше не нужно получать сообщения",
  "createdDate": "DateTime, дата создания подписки"
}
```

9.4.2. Входящие сообщения: проекты

Раздел содержит список входящих сообщений в области проектов.

Создан новый проект

Тип уведомления: ProjectCreated

Фильтрация: нет

Формат данных:

```
{
  "projectId": "GUID, идентификатор проекта"
}
```

Название проекта изменено

Тип уведомления: ProjectNameUpdated

Фильтрация: по ProjectId

Формат данных:

```
{
  "projectId": "GUID, идентификатор проекта",
  "name": "string, название проекта"
}
```

Проект удален

Тип уведомления: ProjectDeleted

Фильтрация: по ProjectId

Формат данных:

```
{
  "projectId": "GUID, идентификатор проекта"
}
```

Параметры проекта изменены

Тип уведомления: ProjectSettingsChanged

Фильтрация: по ProjectId

Формат данных:

```
{
  "projectId": "GUID, идентификатор проекта",
  "settingsId": "GUID, идентификатор параметров проекта",
  "settingsHashCode": "integer, хеш-сумма новых параметров"
}
```

Проект обновлен

Тип уведомления: ProjectUpdated

Фильтрация: по ProjectId

Формат данных:

```
{
  "projectId": "GUID, идентификатор проекта"
}
```

Параметры политик безопасности изменены

Тип уведомления: PoliciesRulesChanged

Фильтрация: по ProjectId

Формат данных:

```
{
  "projectId": "GUID, идентификатор проекта",
  "rulesHashcode": "integer, хеш-сумма правил"
}
```

Параметры интеграции с Jira изменены

Тип уведомления: IssueTrackerIntegrationChanged

Фильтрация: по ProjectId

Формат данных:

```
{
  "projectId": "GUID, идентификатор проекта",
  "settingsId": "GUID, идентификатор параметров проекта",
  "settingsHashCode": "integer, хеш-сумма правил"
}
```

Добавлено подключение к трекеру задач

Тип уведомления: IssueTrackerConnectionAdded

Фильтрация: нет

Формат данных:

```
{
  "connectionId": "GUID, идентификатор подключения"
}
```

Подключение к трекеру задач удалено

Тип уведомления: IssueTrackerConnectionDeleted

Фильтрация: нет

Формат данных:

```
{
  "connectionId": "GUID, идентификатор подключения"
}
```

Подключение к трекеру задач изменено

Тип уведомления: IssueTrackerConnectionChanged

Фильтрация: нет

Формат данных:

```
{
  "connectionId": "GUID, идентификатор подключения"
}
```

Файлы с исходным кодом проекта удалены

Тип уведомления: ProjectSourceDeleted

Фильтрация: по ProjectId

Формат данных:

```
{
  "projectId": "GUID, идентификатор проекта"
}
```

Файлы с исходным кодом проекта обновляются

Тип уведомления: ProjectSourceUpdating

Фильтрация: по ProjectId

Формат данных:

```
{
  "projectId": "GUID, идентификатор проекта",
  "progress": {
    "stage": "string, название стадии",
    "substage": "string, название промежуточной стадии",
    "value": "integer, прогресс обновления"
  }
}
```

Файлы с исходным кодом проекта обновлены

Тип уведомления: ProjectSourceUpdated

Фильтрация: по ProjectId

Формат данных:

```
{
  "projectId": "GUID, идентификатор проекта"
}
```

Ошибка в процессе работы с исходными файлами проекта

Тип уведомления: ProjectSourceError

Фильтрация: по ProjectId

Формат данных:

```
{
  "projectId": "GUID, идентификатор проекта",
  "error": "см. раздел ApiErrorModel в документации по REST API"
}
```

История проекта изменена

Тип уведомления: ProjectHistoryChanged

Фильтрация: по ProjectId

Формат данных:

```
{
  "projectId": "GUID, идентификатор проекта",
  "changeHistoryIds": "long[], идентификаторы записей в истории"
}
```

9.4.3. Входящие сообщения: результаты сканирований

Раздел содержит список входящих сообщений в области результатов сканирований.

Результат сканирования удален

Тип уведомления: ScanResultRemoved

Фильтрация: по ProjectId

Формат данных:

```
{
  "projectId": "GUID, идентификатор проекта",
  "scanResultId": "GUID, идентификатор результата сканирования",
  "deleteGroup": "boolean, удалить группу сканирования"
}
```

Сканирование добавлено в очередь

Тип уведомления: ScanEnqueued

Фильтрация: по ScanresultId

Формат данных:

```
{
  "projectId": "GUID, идентификатор проекта",
  "scanResultId": "GUID, идентификатор результата сканирования"
}
```

Сканирование удалено из очереди

Тип уведомления: ScanDequeued

Фильтрация: по ScanresultId

Формат данных:

```
{
  "projectId": "GUID, идентификатор проекта",
  "scanResultId": "GUID, идентификатор результата сканирования"
}
```

Сканирование запущено

Тип уведомления: ScanStarted

Фильтрация: по ScanresultId

Формат данных:

```
{
  "projectId": "GUID, идентификатор проекта",
  "scanResultId": "GUID, идентификатор результата сканирования"
}
```

Сканирование выполняется

Тип уведомления: ScanProgress

Фильтрация: по ScanresultId

Формат данных:

```
{
  "projectId": "GUID, идентификатор проекта",
  "scanResultId": "GUID, идентификатор результата сканирования",
  "progress": "прогресс сканирования, см. раздел ScanProgressModel в документации по REST API",
  "statistic": "статистика сканирования, см. раздел ScanStatisticModel в документации по REST API"
}
```

Сканирование завершено

Тип уведомления: ScanCompleted

Фильтрация: по ScanresultId

Формат данных:

```
{
  "projectId": "GUID, идентификатор проекта",
  "scanResultId": "GUID, идентификатор результата сканирования",
  "stage": "enum, стадия сканирования, см. раздел Stage в документации по REST API"
}
```

Этап сканирования изменился

Тип уведомления: ScanDelta

Фильтрация: по ScanresultId

Формат данных:

```
{
  "projectId": "GUID, идентификатор проекта",
  "scanResultId": "GUID, идентификатор результата сканирования",
  "statistic": "статистика сканирования, см. раздел ScanStatisticModel в документации по REST API"
}
```

Проверка запуска инкрементального сканирования

Тип уведомления: IncrementalScanCheck

Фильтрация: по ScanresultId

Формат данных:

```
{
  "id": "GUID, идентификатор проекта",
  "scanResultId": "GUID, идентификатор результата сканирования",
  "previousScanResultId": "GUID, идентификатор предыдущего результата сканирования",
  "isCheckSuccess": "boolean, результат подготовки инкрементального сканирования",
  "projectId": "GUID, идентификатор проекта"
}
```

Изменены метаданные уязвимостей

Тип уведомления: IssueMetadata

Фильтрация: нет

Формат данных:

```
{
  "issueMetadata": "см. раздел IssueBaseMetadata в документации по REST API"
}
```

Изменено описание уязвимостей

Тип уведомления: IssueDescription

Фильтрация: нет

Формат данных:

```
{
  "issueDescription": "см. раздел IssueDescriptionModelBase в документации по REST API"
}
```

Изменен список найденных при сканировании уязвимостей

Тип уведомления: IssuesUpdated

Фильтрация: по ScanresultId

Формат данных:

```
{
  "projectId": "GUID, идентификатор проекта",
  "scanResultId": "GUID, идентификатор результата сканирования",
  "issues": "IssueUpdatedInfoModel[], информация об обновленных уязвимостях, см. ниже"
}
IssueUpdatedInfoModel
{
  "id": "GUID, идентификатор уязвимости",
  "isFavorite": "boolean, является ли уязвимость избранной",
  "approvalState": "enum, статус подтверждения, см. раздел IssueApprovalState в документации по REST API",
  "exploitUrl": "string, URL эксплойта автопроверки",
  "exploit": "string, эксплойт автопроверки"
}
```

Для уязвимости добавлена ссылка на задачу в Jira

Тип уведомления: TicketLinkedToIssue

Фильтрация: по ScanresultId

Формат данных:

```
{
  "projectId": "GUID, идентификатор проекта",
  "scanResultId": "GUID, идентификатор результата сканирования",
  "issueId": "GUID, идентификатор уязвимости"
}
```

Для уязвимости удалена ссылка на задачу в Jira

Тип уведомления: TicketUnlinkedFromIssue

Фильтрация: по ScanresultId

Формат данных:

```
{
  "projectId": "GUID, идентификатор проекта",
  "scanResultId": "GUID, идентификатор результата сканирования",
  "issueId": "GUID, идентификатор уязвимости"
}
```

Началась массовая смена статусов уязвимостей

Тип уведомления: StartChangingIssuesStates

Фильтрация: по ScanresultId

Формат данных:

```
{
  "projectId": "GUID, идентификатор проекта",
  "scanResultId": "GUID, идентификатор результата сканирования"
}
```

Закончилась массовая смена статусов уязвимостей

Тип уведомления: FinishChangingIssuesStates

Фильтрация: по ScanresultId

Формат данных:

```
{
  "projectId": "GUID, идентификатор проекта",
  "scanResultId": "GUID, идентификатор результата сканирования"
}
```

9.4.4. Входящие сообщения: аутентификация

Раздел содержит список входящих сообщений в области аутентификации пользователей.

Токен доступа некорректен или истек. Необходимо обновить токен

Тип уведомления: NeedRefreshToken

Фильтрация: нет

Формат данных: нет

SSO-провайдер добавлен или обновлен

Тип уведомления: AuthProviderChanged

Фильтрация: нет

Формат данных:

```
{
  "providerId": "GUID, идентификатор SSO-провайдера",
  "isDisabled": "boolean, флаг добавления/отключения SSO-провайдера"
}
```

SSO-провайдер удален

Тип уведомления: AuthProviderDeleted

Фильтрация: нет

Формат данных:

```
{  
  "providerId": "GUID, идентификатор SSO-провайдера"  
}
```

Добавлен новый пользователь

Тип уведомления: UserCreated

Фильтрация: нет

Формат данных:

```
{  
  "userId": "GUID, идентификатор пользователя"  
}
```

Пользователь удален

Тип уведомления: UserDeleted

Фильтрация: по UserId

Формат данных:

```
{  
  "userId": "GUID, идентификатор пользователя"  
}
```

Пользователь заблокирован

Тип уведомления: UserBlocked

Фильтрация: по UserId

Формат данных:

```
{  
  "userId": "GUID, идентификатор пользователя"  
}
```

Пользователь разблокирован

Тип уведомления: UserUnblocked

Фильтрация: по UserId

Формат данных:

```
{
  "userId": "GUID, идентификатор пользователя"
}
```

9.4.5. Входящие сообщения: полномочия

Раздел содержит список входящих сообщений в области полномочий пользователей.

Разрешения для проектного пользователя изменены

Тип уведомления: ProjectMembersChangePermission

Фильтрация: по ProjectId

Формат данных:

```
{
  "projectId": "guid, идентификатор проекта на котором поменялись разрешения",
  "userId": "guid, идентификатор пользователя"
}
```

Разрешения для глобального пользователя изменены

Тип уведомления: GlobalMembersChangePermission

Фильтрация: нет

Формат данных:

```
{
  "userId": "GUID, идентификатор пользователя"
}
```

Роль создана

Тип уведомления: RoleCreated

Фильтрация: ManagePermissions

Формат данных:

```
{
  "roleId": "string, идентификатор роли"
}
```

Роль обновлена

Тип уведомления: RoleUpdated

Фильтрация: ManagePermissions

Формат данных:

```
{
  "roleId": "string, идентификатор роли"
}
```

Роль удалена

Тип уведомления: RoleDeleted

Фильтрация: ManagePermissions

Формат данных:

```
{
  "roleId": "string, идентификатор роли"
}
```

9.4.6. Входящие сообщения: параметры сервера

Раздел содержит список входящих сообщений в области параметров сервера.

Серверный сертификат скоро истечет

Тип уведомления: ServerCertWillExpired

Фильтрация: нет

Формат данных:

```
{
  "friendlyName": "string, имя сертификата",
  "thumbprint": "string, отпечаток сертификата",
  "expiredDate": "dateTime, срок действия сертификата"
}
```

Серверный сертификат изменен

Тип уведомления: ServerCertChanged

Фильтрация: нет

Формат данных:

```
{
  "certificateInfo": "см. раздел CertificateInfoModel в документации по REST API"
}
```

Лицензия изменена

Тип уведомления: LicenseChanged

Фильтрация: нет

Формат данных:

```
{
  "licenseData": "см. раздел EnterpriseLicenseModel в документации по REST API"
}
```

Лицензия скоро истечет

Тип уведомления: LicenseWillExpire

Фильтрация: нет

Формат данных:

```
{
  "licenseData": "см. раздел EnterpriseLicenseModel в документации по REST API",
  "licenseExpirationLevel": "enum, уровень важности. Возможные значения: Information
| Warning | Error"
}
```

9.4.7. Входящие сообщения: планировщик сканирований

Раздел содержит список входящих сообщений в области планировщика сканирований

Агент сканирования добавлен

Тип уведомления: ScanAgentRegistered

Фильтрация: нет

Формат данных:

```
{
  "scanAgentId": "GUID, идентификатор агента"
}
```

Агент сканирования удален

Тип уведомления: ScanAgentDeleted

Фильтрация: нет

Формат данных:

```
{
  "scanAgentId": "GUID, идентификатор агента"
}
```

Статус агента сканирования изменен

Тип уведомления: ScanAgentStatusChanged

Фильтрация: нет

Формат данных:

```
{
  "scanAgentId": "GUID, идентификатор агента",
  "statusType": "enum, возможные значения см. в разделе ScanAgentStatusType в документации по REST API",
  "projectName": "string, название сканируемого проекта",
  "progress": "прогресс сканирования, см. раздел ScanProgressModel в документации по REST API"
}
```

Превышено максимальное число агентов, доступных по текущей лицензии

Тип уведомления: ScanAgentLimitExceeded

Фильтрация: нет

Формат данных:

```
{
  "scanAgentName": "string, имя подключающегося агента",
  "limit": "integer, максимально допустимое количество агентов"
}
```

9.4.8. Входящие сообщения: отчеты

Раздел содержит список входящих сообщений в области отчетов.

Выпуск отчета

Тип уведомления: ReportGenerateProgress

Фильтрация: по SessionId

Формат данных:

```
{
  "sessionId": "GUID, идентификатор сессии выпуска отчета",
  "projectId": "GUID, идентификатор проекта",
  "progressMessage": "string, текст стадии выпуска",
  "progressValue": "float, прогресс выпуска отчета в процентах"
}
```

Не удалось выпустить отчет

Тип уведомления: ReportGenerateError

Фильтрация: по SessionId

Формат данных:

```
{
  "sessionId": "GUID, идентификатор сессии выпуска отчета",
  "projectId": "GUID, идентификатор проекта",
  "errorMessage": "string, текст ошибки"
}
```

Отчет выпущен

Тип уведомления: ReportGenerateFinished

Фильтрация: по SessionId

Формат данных:

```
{
  "sessionId": "GUID, идентификатор сессии выпуска отчета",
  "projectId": "GUID, идентификатор проекта"
}
```

9.4.9. Входящие сообщения: результат отправки тестового сообщения

Раздел содержит описание тестового сообщения для его отправки по почте.

Результат отправки тестового сообщения

Тип уведомления: TestMailSendingResult

Фильтрация: по TestMailId

Формат данных:

```
{
  "testMailId": "GUID, идентификатор тестового сообщения",
  "initiatorId": "GUID, идентификатор пользователя, который отправил тестовое сообщение",
  "success": "boolean, прошла ли отправка успешно",
  "errorMessage": "string, текст ошибки при неуспешной отправке"
}
```

10. Решение проблем

В этом разделе приводятся инструкции по диагностике, устранению ошибок и решению проблем, возникающих при работе с PT AI Enterprise Edition.

В некоторых случаях, когда вы работаете над устранением неисправности или взаимодействуете с технической поддержкой, может потребоваться предоставить журналы событий от модулей системы. Архив с файлами журнала можно выгрузить, воспользовавшись [соответствующей инструкцией \(см. раздел 7.8\)](#).

В этом разделе

[Нехватка оперативной памяти \(см. раздел 10.1\)](#)

[Ошибка миграции данных при большом объеме БД PostgreSQL \(см. раздел 10.2\)](#)

[Ошибка «wait-for-healthy-container.sh ai-casgate» \(см. раздел 10.3\)](#)

[Остановка сканирования из-за повреждения БД ядра Dependency-check \(см. раздел 10.4\)](#)

[Сброс пароля пользователя root \(см. раздел 10.5\)](#)

[Проблема подключения к веб-интерфейсу при использовании реверс-прокси \(см. раздел 10.6\)](#)

[Ошибка при обновлении до версии 5.2.0 вследствие замены внутренних сертификатов \(см. раздел 10.7\)](#)

[Не удастся запустить инкрементальное сканирование \(см. раздел 10.8\)](#)

[Проблема обновления токена доступа к GitLab \(см. раздел 10.9\)](#)

[Проблема доступа к внутреннему репозиторию \(см. раздел 10.10\)](#)

См. также

[Выгрузка файлов журнала событий \(см. раздел 7.8\)](#)

10.1. Нехватка оперативной памяти

Проблема

При сканировании проектов в PT AI Enterprise Agent для Linux отображаются ошибки «Insufficient memory».

Возможные причины

Агенту выделено недостаточно оперативной памяти для сканирования или ее пространство используется на 100%.

Решение

► Чтобы решить проблему:

1. Увеличьте время сканирования, запустив его с дополнительным параметром `--scan-threads-count X`, где значение `X` должно быть меньше половины количества ядер ЦПУ на узле с PT AI Enterprise Agent.

2. Очистите кэш PT AI Enterprise Agent:

```
linux sync && echo 3 > /proc/sys/vm/drop_caches
```

или

```
sudo sysctl -w vm.drop_caches=3
```

10.2. Ошибка миграции данных при большом объеме БД PostgreSQL

Проблема

При обновлении PT AI Enterprise Edition до версии ниже 4.11.0 возникает ошибка миграции данных, и сбрасываются параметры LDAP-подключения.

Возможные причины

Ошибка миграции может возникать из-за большого объема внешней БД PostgreSQL, подключенной к PT AI Enterprise Server. В версии 4.11.0 процесс миграции был оптимизирован, и проблема устранена.

Решение

Проблема решается в несколько этапов:

- очистка БД с использованием команды `VACUUM`;
- запуск обновления PT AI Enterprise Edition;
- создание нового LDAP-подключения, если при обновлении сбросились его параметры.

► Чтобы очистить БД:

1. Создайте резервную копию БД для возможности восстановления работоспособности PT AI Enterprise Edition:

- Создайте дампы баз данных внутри Docker-контейнера:


```
sudo docker exec -it ai-postgres pg_dump -U postgres --encoding=utf-8 --format=custom --compress=9 --file=/tmp/ai_csi_docker.dump ai_csi_docker
```
- Скопируйте дампы из Docker-контейнера в каталог ./:


```
sudo docker cp ai-postgres:/tmp/ai_csi_docker.dump ./
```
- 2. Отключите все компоненты PT AI Enterprise Edition, кроме компонента с БД:


```
docker stop $(docker container ls -aq --no-trunc -f label=com.ptsecurity.deployment) &&
docker start ai-postgres
```
- 3. Выполните VACUUM:


```
docker exec -it -u "$(id -u application-inspector):$(id -g application-inspector)" ai-postgres vacuumdb -U postgres -j 4 -z -v ai_csi_docker
```
- 4. Если очистка завершилась успешно, запустите все компоненты PT AI Enterprise Edition:


```
docker start $(docker container ls -aq --no-trunc -f label=com.ptsecurity.deployment)
```
- 5. Если выполнение VACUUM завершилось с ошибкой нехватки ОЗУ:
 - Посмотрите и сохраните пароль от БД:


```
docker inspect ai-postgres | grep POSTGRES_PASSWORD=
```
 - Остановите компонент с БД:


```
docker stop ai-postgres
```
 - Запустите временную копию БД с увеличенным значением ОЗУ (например, до 2 ГБ), и паролем от БД из переменной "POSTGRES_PASSWORD":


```
docker run --rm -it --shm-size=2gb -u "$(id -u application-inspector):$(id -g application-inspector)" -e 'PGDATA=/var/lib/postgresql/data' -e 'POSTGRES_PASSWORD=<Пароль от БД, полученный ранее>' -v '/opt/ptai/_data/volumes/postgres/config/postgresql.conf:/etc/postgresql/postgresql.conf:ro' -v '/opt/ptai/_data/volumes/postgres/data:/var/lib/postgresql/data:rw' -v '/opt/ptai/_data/volumes/postgres/logs:/var/log/postgresql:rw' --name ai-postgres-temp postgres:14.2 postgres -c config_file=/etc/postgresql/postgresql.conf
```
 - Повторно запустите выполнение VACUUM во временном Docker-контейнере:


```
docker exec -it -u "$(id -u application-inspector):$(id -g application-inspector)" ai-postgres-temp vacuumdb -U postgres -j 4 -z -v ai_csi_docker
```
 - Остановите временную копию компонента БД:


```
docker stop ai-postgres-temp
```
- 6. Запустите все компоненты PT AI Enterprise Edition:


```
docker start $(docker container ls -aq --no-trunc -f label=com.ptsecurity.deployment)
```

После очистки БД вы можете запустить обновление. Если процесс обновления проходит слишком долго или по-прежнему завершается с ошибками, запросите у службы технической поддержки патч-версию PT AI Enterprise Edition с необходимыми исправлениями и повторите обновление.

Если по завершении обновления не запускается Docker-контейнер auth и невозможно войти в веб-интерфейс PT AI Enterprise Edition, необходимо заново создать LDAP-подключение с сохранением учетных записей пользователей и их действий.

Примечание. Вы можете убедиться, что параметры LDAP-подключения сбросились, выполнив скрипт: `docker exec -u root -ti ai-postgres psql -U postgres -P pager=off -c 'select * from "AuthProviderEntity" a full join "AuthProviderSettingsEntity" s on a."SettingsId" = s."Id";' ai_csi_docker &> result.txt`. Непустой вывод будет означать, что необходимо заново создать подключение.

► Чтобы пересоздать LDAP-подключение:

1. Получите пароль от БД:
`docker inspect ai-postgres | grep POSTGRES_PASSWORD=`
2. Выполните вход в Docker-контейнер ai-postgres:
`docker exec -ti ai-postgres bash`
3. Подключитесь к БД в Docker-контейнере ai-postgres, указав пароль, полученный на шаге 1:
`psql -U postgres -h ai-postgres -d ai_csi_docker`
4. Выполните SQL-скрипт:
`alter table "UserEntity" drop constraint "UserEntityAuthProviderEntity_FK";
truncate "AuthProviderEntity" cascade;`
5. Завершите сеанс в БД командой `exit`.
6. Выполните выход из Docker-контейнера ai-postgres командой `exit`.
7. Перезапустите Docker-контейнер auth:
`docker restart auth`
8. Выполните вход в веб-интерфейс PT AI Enterprise Edition под учетной записью root и создайте LDAP-подключение заново.
9. Выполните вход в Docker-контейнер ai-postgres и подключитесь к БД ai_csi_docker, используя алгоритм, описанный на шагах 1 — 3.
10. Выполните SQL-скрипт:
`update "UserEntity"
set "AuthProviderId" = (select "Id" from "AuthProviderEntity" limit 1)
where "AuthProviderId" is not null;
alter table "UserEntity"
add constraint "UserEntityAuthProviderEntity_FK" foreign key ("AuthProviderId")
references "AuthProviderEntity" ("Id");`
11. Завершите сеанс в БД командой `exit`.
12. Выполните выход из Docker-контейнера ai-postgres командой `exit`.
13. Перезапустите Docker-контейнер auth:
`docker restart auth`

10.3. Ошибка «wait-for-healthy-container.sh ai-casgate»

Проблема

При установке, обновлении или перезапуске PT AI Enterprise Server возникает ошибка:

```
Error running command '../tools/wait-for-healthy-container.sh ai-casgate
300': exit status 1. Output: Wait for container 'ai-casgate' to be healthy
for max 300 seconds...
Timeout exceeded. Health status returned: unhealthy
```

Проблема актуальна для версий 4.8.x и 4.9.x.

Возможные причины

Недоступен Docker-контейнер ai-casgate.

Решение

► Чтобы решить проблему:

1. Распакуйте установочные файлы PT AI Enterprise Server в текущий каталог:
`./applicationinspector.<Версия>.run --noexec --target ./`
2. Откройте файл `ptai/tf/modules/infrastructure/casgate.tf` и замените строку:
`test = ["CMD-SHELL", "netstat -lp | grep LISTEN | grep -q 1/server"] # check if casgate server listens`
на
`test = ["CMD-SHELL", "netstat -ln | grep -q LISTEN"] # check if casgate server listens`
3. Сохраните изменения и запустите скрипт `prepare.sh`.

10.4. Остановка сканирования из-за повреждения БД ядра Dependency-check

Проблема

Ядро Dependency-check многократно пытается подключиться к БД H2, но в результате завершает работу с ошибкой, что приводит к остановке сканирования.

Возможные причины

Повреждение БД ядра Dependency-check.

Решение

► Чтобы решить проблему:

1. Удалите содержимое папки `C:/Users/Administrator/AppData/Local/AI Enterprise/Agent/Database/`.
2. Запустите сканирование повторно.

10.5. Сброс пароля пользователя root

Проблема

Не удастся восстановить пароль пользователя root до значения по умолчанию.

Способы устранения этой проблемы различаются в зависимости от версии PT AI Enterprise Edition.

Возможные причины

Проблема возникает в том случае, если вы не знаете или забыли текущий пароль пользователя root.

Решение

► Чтобы решить проблему в версии ниже 4.8.0:

1. Получите пароль от БД PostgreSQL:
`docker inspect ai-postgres | grep POSTGRES_PASSWORD=`
2. Выполните вход в Docker-контейнер ai-postgres:
`docker exec -ti ai-postgres bash`
3. В Docker-контейнере ai-postgres подключитесь к БД:
`psql -U postgres -h ai-postgres -d ai_csi_docker`
4. Укажите пароль, полученный на шаге 1.
5. Выполните скрипт:

```
delete from "AccessTokenEntity" where "UserCreatorId" in (select "Id" from "UserEntity"
where "Login"='root');
delete from "GlobalMemberEntity" where "UserId" in (select "Id" from "UserEntity" where
"Login"='root');
delete from "PmRuleEntity" where "ModifiedByUserId" in (select "Id" from "UserEntity"
where "Login"='root');
delete from "ProjectChangeHistoryEntity" where "UserId" in (select "Id" from "UserEntity"
where "Login"='root');
delete from "ProjectMemberEntity" where "UserId" in (select "Id" from "UserEntity" where
"Login"='root');
```

```
delete from "ReportTemplateEntity" where "UserCreatorId" in (select "Id" from "UserEntity"
where "Login"='root');
delete from "ScanQueueEntity" where "UserId" in (select "Id" from "UserEntity" where
"Login"='root');
delete from "UploadSessionEntity" where "UserId" in (select "Id" from "UserEntity" where
"Login"='root');
delete from "UserReportParametersEntity" where "UserId" in (select "Id" from "UserEntity"
where "Login"='root');
delete from "UserEntity" where "Login"='root';
delete from "UserProjectSettingsEntity" where "UserId" in (select "Id" from "UserEntity"
where "Login"='root');
delete from "UserEntity" where "Login"='root';
```

6. Завершите сеанс в БД командой `exit`.
7. Выполните выход из Docker-контейнера командой `exit`.
8. Перезапустите Docker-контейнер:
`auth - docker restart auth`
9. Выполните вход в интерфейс PT AI Enterprise Edition с учетными данными по умолчанию (логин — `root`, пароль — `root`).

► Чтобы решить проблему в версии 4.8.0 и выше:

1. В PT AI Enterprise Server создайте файл скрипта `change_password.sh`:
`touch change_password.sh`
2. Вставьте содержимое скрипта в указанный файл, заменив строку <Введите новый пароль> на пароль, содержащий заглавные и строчные буквы, а также специальные символы:

```
password=$(env | grep CASGATE_ADMIN_PASSWORD | cut -d=' ' -f2)
client_id=$(
curl -s --location 'http://localhost:8000/api/get-application?id=admin%2Fapp-built-in' \
| grep -oE '"clientId": "\w+"' | cut -d':' -f2 | sed -e 's/^"' -e 's/"$'
)
access_token=$(
curl -s --location --request POST \
--data-urlencode "grant_type=password" \
--data-urlencode "client_id=$client_id" \
--data-urlencode "username=admin" \
--data-urlencode "password=$password" \
'http://localhost:8000/api/login/oauth/access_token' \
| grep -oE '"access_token": "[^"]+?"' | cut -d':' -f2 | sed -e 's/^"' -e 's/"$'
)
curl -s --location --request POST 'http://localhost:8000/api/set-password' \
--header "Authorization: Bearer $access_token" \
--form 'userOwner="pt"' \
--form 'userName="root"' \
```

```
--form 'oldPassword=""' \
--form 'newPassword="Введите новый пароль"'
```

3. Запустите скрипт:

```
docker exec ai-casgate sh -c "$(cat change_password.sh)"
```

10.6. Проблема подключения к веб-интерфейсу при использовании реверс-прокси

Проблема

При подключении к веб-интерфейсу PT AI Enterprise Edition через реверс-прокси отображается ошибка «URL not in a list of allowed endpoints».

Возможные причины

Реверс-прокси меняет исходный адрес запроса. PT AI Enterprise Edition не распознает этот адрес, потому что он не указан в списке разрешенных в конфигурации.

Решение

- Чтобы решить проблему:

1. Откройте конфигурационный файл `ptai.ini`:
`nano /etc/ptai/ptai.ini`
2. Раскомментируйте параметр `AllowedRedirectEndpoints` и укажите один или несколько адресов реверс-прокси в нужном виде:
`AllowedRedirectEndpoints=<IP-адрес>, <IP-адрес>:<Порт>, <FQDN> или <FQDN>:<Порт>`
3. Измените конфигурацию PT AI Enterprise Server:
`sudo /opt/ptai/latest/bin/ptaictl reconf`
4. Перезапустите PT AI Enterprise Server:
`sudo /opt/ptai/latest/bin/ptaictl restart`

После изменения параметра `AllowedRedirectEndpoints` доступ с указанных адресов будет открыт.

10.7. Ошибка при обновлении до версии 5.2.0 вследствие замены внутренних сертификатов

Проблема

Не удастся обновить PT AI Enterprise Edition до версии 5.2.0.

Возможные причины

В предыдущей версии внутренние сертификаты были заменены с помощью скрипта, предоставленного службой технической поддержки.

Решение

► Чтобы решить проблему:

1. Остановите все Docker-контейнеры PT AI Enterprise Edition:
`/opt/ptai/latest/bin/ptaictl stop`
2. Замените каталоги `state` и `certs` на резервные копии, созданные при предыдущей замене сертификатов:
`cp -rf state/ /opt/ptai/_data/`
`cp -rf certs/ /opt/ptai/_data/volumes/`
3. Запустите обновление:
`sudo sh applicationinspector.<Версия PT AI Enterprise Server>.run`

10.8. Не удастся запустить инкрементальное сканирование

Проблема

По завершении первого полного сканирования не удастся запустить инкрементальное сканирование проекта.

Возможные причины

В системе существует хотя бы одно из ограничений, при которых запуск инкрементального сканирования невозможен:

- указаны параметры для запуска полного сканирования (например, `fullScanMode` или `Force full AST` в параметрах плагина Jenkins);
- предыдущее сканирование завершилось с ошибками;
- изменены параметры проекта;
- сканируется другая ветка проекта.

Решение

Перед запуском инкрементального сканирования убедитесь, что все вышеуказанные ограничения устранены.

10.9. Проблема обновления токена доступа к GitLab

Проблема

В нескольких проектах, использующих VCS (GitLab) в качестве источника исходного кода, применяется один и тот же устаревший или скомпрометированный токен доступа к GitLab. Его замену приходится осуществлять вручную в каждом отдельном проекте.

Возможные причины

Не настроено массовое обновление токена доступа к GitLab во всех VCS-проектах.

Решение

Предварительные требования:

- токен доступа к REST API PT AI Enterprise Edition;
- URL API PT AI Enterprise Edition;
- новый токен доступа к GitLab.

► Чтобы обновить токен доступа к GitLab:

1. Выполните запрос на получение списка всех проектов с информацией о типе источника исходного кода.

Пример запроса в формате cURL:

```
curl -k -X GET "<Корневой URL API>/api/projects"?limit=-1&offset=-1\"  
-H "Authorization: Bearer <accessToken>"
```

где:

- `limit=-1` — получить все проекты (без ограничения);
- `offset=-1` — показать полный список без постраничного деления.

В ответ на запрос будет возвращен массив объектов `ProjectModel` с полями: `id` — идентификатор проекта, `name` — название проекта, `settingsId` — идентификатор параметров проекта.

2. Определите тип источника кода для каждого проекта.

Пример запроса в формате cURL:

```
curl -k -X GET "<Корневой URL API>/api/projects/{projectId}/settings" \  
-H "Authorization: Bearer <accessToken>"
```

В ответ на запрос будет возвращен объект `ProjectSettingsUpdateModel`. Для проектов, использующих VCS в качестве источника исходного кода, будет присутствовать поле `vcsConnectionSettings`.

3. Обновите токен доступа к GitLab для VCS-проекта.

Пример запроса в формате cURL:

```
curl -k -X PUT "<Корневой URL API>/api/sourceControl/{projectId}/credentials" \
  -H "Authorization: Bearer <accessToken>" \
  -H "Content-Type: application/json" \
  -d "{
    \"sourceControlCredentialsType\": \"Password\",
    \"login\": \"<gitlabUsername>\",
    \"password\": \"<newGitlabToken>\"
  }"
```

где:

- `sourceControlCredentialsType` — тип учетных данных (для токенов GitLab — `"Password"`);
- `login` — имя пользователя GitLab;
- `password` — новый токен доступа к GitLab.

10.10. Проблема доступа к внутреннему репозиторию

Проблема

PT AI Enterprise Agent, установленный в закрытом контуре, не может загрузить зависимости для Java-проектов при обращении к внутреннему репозиторию Nexus. В журнале `dependencies_downloading.log` присутствует ошибка «The following artifacts could not be resolved...». Из-за этого анализатор не находит часть уязвимостей.

Возможные причины

В контейнере, в котором работает PT AI Enterprise Agent, отсутствует сертификат доверия к внутреннему репозиторию Nexus.

Решение

► Чтобы решить проблему:

1. Получите хранилище сертификатов (файл с расширением `.jks` или `.p12`), которое используется в вашем внутреннем репозитории.
2. Скопируйте файл на узел с PT AI Enterprise Agent.
3. Внесите изменения в скрипт установки PT AI Enterprise Agent `/opt/ptai-agent/_latest/bin/ptai-agent.sh` сразу после строки `exec docker run \`:

Формат JKS:

```
-v "<Путь к хранилищу на узле>:/opt/security/nexus-truststore.jks" \
-e MAVEN_OPTS="-Djavax.net.ssl.trustStore=/opt/security/nexus-truststore.jks
-Djavax.net.ssl.trustStorePassword=<Пароль от хранилища, не в открытом виде>"
```

Формат PKCS12:

```
-v "<Путь к хранилищу на узле>/nexus-truststore.p12:/opt/security/nexus-truststore.p12" \  
-e MAVEN_OPTS="-Djavax.net.ssl.trustStore=/opt/security/nexus-truststore.p12 \  
-Djavax.net.ssl.trustStorePassword=<Пароль от хранилища, не в открытом виде> \  
-Djavax.net.ssl.trustStoreType=PKCS12" \  

```

Примечание. Скрипт установки PT AI Enterprise Agent необходимо обновлять таким образом при каждом обновлении агента.

11. О технической поддержке

Гарантийная поддержка доступна для всех владельцев действующих лицензий на PT AI Enterprise Edition в течение периода предоставления обновлений и включает в себя следующий набор услуг.

Предоставление доступа к обновленным версиям продукта

Обновления продукта выпускаются в порядке и в сроки, определяемые Positive Technologies. Positive Technologies предоставляет обновленные версии продукта в течение оплаченного периода получения обновлений, указанного в бланке лицензии приобретенного продукта Positive Technologies.

Positive Technologies не производит установку обновлений продукта в рамках гарантийной поддержки и не несет ответственности за инциденты, возникшие в связи с некорректной или несвоевременной установкой обновлений продукта.

Восстановление работоспособности продукта

Специалист Positive Technologies проводит диагностику заявленного сбоя и предоставляет рекомендации для восстановления работоспособности продукта. Восстановление работоспособности может заключаться в выдаче рекомендаций по установке продукта заново с потенциальной потерей накопленных данных либо в восстановлении версии продукта из доступной резервной копии (резервное копирование должно быть настроено заблаговременно). Positive Technologies не несет ответственности за потерю данных в случае неверно настроенного резервного копирования.

Примечание. Помощь оказывается при условии, что конечным пользователем продукта соблюдены все аппаратные, программные и иные требования и ограничения, описанные в документации к продукту.

Устранение ошибок и дефектов в работе продукта в рамках выпуска обновленных версий

Если в результате диагностики обнаружен дефект или ошибка в работе продукта, Positive Technologies обязуется включить исправление в ближайшие возможные обновления продукта (с учетом релизного цикла продукта, приоритета дефекта или ошибки, сложности требуемых изменений, а также экономической целесообразности исправления). Сроки выпуска обновлений продукта остаются на усмотрение Positive Technologies.

Рассмотрение предложений по доработке продукта

При обращении с предложением по доработке продукта необходимо подробно описать цель такой доработки. Вы можете поделиться рекомендациями по улучшению продукта и оптимизации его функциональности. Positive Technologies обязуется рассмотреть все предложения, однако не принимает на себя обязательств по реализации каких-либо

доработок. Если Positive Technologies принимает решение о доработке продукта, то способы и сроки ее реализации остаются на усмотрение Positive Technologies. Заявки принимаются [на портале технической поддержки](#).

Портал технической поддержки

[На портале технической поддержки](#) вы можете круглосуточно создавать и обновлять заявки и читать новости продуктов.

Для получения доступа к portalу технической поддержки нужно создать учетную запись, используя адрес электронной почты в официальном домене вашей организации. Вы можете указать другие адреса электронной почты в качестве дополнительных. Добавьте в профиль название вашей организации и контактный телефон — так нам будет проще с вами связаться.

Поддержка на портале предоставляется на русском и английском языках.

Условия предоставления гарантийной поддержки

Для получения гарантийной поддержки необходимо оставить заявку [на портале технической поддержки](#) и предоставить следующие данные:

- номер лицензии на использование продукта;
- файлы журналов и наборы диагностических данных, которые требуются для анализа;
- снимки экрана.

Positive Technologies не берет на себя обязательств по оказанию услуг гарантийной поддержки в случае вашего отказа предоставить запрашиваемую информацию или отказа от внедрения предоставленных рекомендаций.

Если заказчик не предоставляет необходимую информацию по прошествии 20 календарных дней с момента ее запроса, специалист технической поддержки имеет право закрыть заявку. Оказание услуг может быть возобновлено по вашей инициативе при условии предоставления запрошенной ранее информации.

Услуги по гарантийной поддержке продукта не включают в себя услуги по переустановке продукта, решению проблем с операционной системой, инфраструктурой заказчика или сторонним программным обеспечением.

Заявки могут направлять уполномоченные сотрудники заказчика, владеющего продуктом на законном основании (включая наличие действующей лицензии). Специалисты заказчика должны иметь достаточно знаний и навыков для сбора и предоставления диагностической информации, необходимой для решения заявленной проблемы.

Услуги по гарантийной поддержке оказываются в отношении поддерживаемых версий продукта. Информация о поддерживаемых версиях содержится в «Политике поддержки версий программного обеспечения» и (или) иных информационных материалах, размещенных на официальном веб-сайте Positive Technologies.

Время реакции и приоритизация заявок

При получении заявки ей присваивается регистрационный номер, специалист службы технической поддержки классифицирует ее (присваивает тип и уровень значимости) и выполняет дальнейшие шаги по ее обработке.

Время реакции рассчитывается с момента получения заявки до первичного ответа специалиста технической поддержки с уведомлением о взятии заявки в работу. Время реакции зависит от уровня значимости заявки. Специалист службы технической поддержки оставляет за собой право переопределить уровень значимости в соответствии с приведенными ниже критериями. Указанные сроки являются целевыми, но возможны отклонения от них по объективным причинам.

Таблица 30. Время реакции на заявку

Уровень значимости заявки	Критерии значимости заявки	Время реакции на заявку
Критический	Аварийные сбои, приводящие к невозможности штатной работы продукта (исключая первоначальную установку) либо оказывающие критически значимое влияние на бизнес заказчика	До 4 часов
Высокий	Сбои, проявляющиеся в любых условиях эксплуатации продукта и оказывающие значительное влияние на бизнес заказчика	До 8 часов
Средний	Сбои, проявляющиеся в специфических условиях эксплуатации продукта либо не оказывающие значительного влияния на бизнес заказчика	До 8 часов
Низкий	Вопросы информационного характера либо сбои, не влияющие на эксплуатацию продукта	До 8 часов

Указанные часы относятся к рабочему времени (рабочие дни с 9:00 до 18:00 UTC+3) специалистов технической поддержки. Под рабочим днем понимается любой день за исключением субботы, воскресенья и дней, являющихся официальными нерабочими днями в Российской Федерации.

Обработка и закрытие заявок

По мере рассмотрения заявки и выполнения необходимых действий специалист службы поддержки сообщает вам:

- о ходе диагностики по заявленной проблеме и ее результатах;
- планах выпуска обновленной версии продукта (если требуется для устранения проблемы).

Если по итогам обработки заявки необходимо внести изменения в продукт, Positive Technologies включает работы по исправлению в ближайшее возможное плановое обновление продукта (с учетом релизного цикла продукта, приоритета дефекта или ошибки, сложности требуемых изменений, а также экономической целесообразности исправления). Сроки выпуска обновлений продукта остаются на усмотрение Positive Technologies.

Специалист закрывает заявку, если:

- предоставлены решение или возможность обойти проблему, не влияющие на критически важную функциональность продукта (по усмотрению Positive Technologies);
- диагностирован дефект продукта, собрана техническая информация о дефекте и условиях его воспроизведения, исправление дефекта запланировано к выходу в рамках планового обновления продукта;
- заявленная проблема вызвана программным обеспечением или оборудованием сторонних производителей, на которые не распространяются обязательства Positive Technologies;
- заявленная проблема классифицирована специалистами Positive Technologies как неподдерживаемая.

Примечание. Если продукт приобретался вместе с аппаратной платформой в виде программно-аппаратного комплекса (ПАК), решение заявок, связанных с ограничением или отсутствием работоспособности аппаратных компонентов, происходит согласно условиям и срокам, указанным в гарантийных обязательствах (гарантийных талонах) на такие аппаратные компоненты.



Positive Technologies — один из лидеров в области результативной кибербезопасности. Компания является ведущим разработчиком продуктов, решений и сервисов, позволяющих выявлять и предотвращать кибератаки до того, как они причинят неприемлемый ущерб бизнесу и целым отраслям экономики. Positive Technologies — первая и единственная компания из сферы кибербезопасности на Московской бирже (MOEX: POSI). Количество акционеров превышает 220 тысяч.