Primo RPA 1.24.6

Orchestrator

Руководство системного администратора по развертыванию

Оглавление

[Введение 4](#_Toc164098448)

[1. Требования к персоналу 7](#_Toc164098449)

[2. Комплект поставки 8](#_Toc164098450)

[3. Аппаратные требования 12](#_Toc164098451)

[4. Установка компонентов Оркестратора 14](#_Toc164098452)

[5. Варианты развертывания сервера приложений Оркестратора для ОС Windows 2016 Server 15](#_Toc164098453)

[6. Атрибуты безопасности 16](#_Toc164098454)

[7. Контроль целостности конфигурационных файлов 18](#_Toc164098455)

[8. Интеграция с AD 19](#_Toc164098456)

[9. Мультитенантность 25](#_Toc164098457)

[9.1. Мультитенантная AD-авторизация 27](#_Toc164098458)

[10. Первоначальная настройка Оркестратора 28](#_Toc164098459)

[10.1. Получение лицензий 28](#_Toc164098460)

[10.2. Установка лицензий 29](#_Toc164098461)

[10.3. Публикация дистрибутива 29](#_Toc164098462)

[10.4. Настройка и регистрация машин Роботов 30](#_Toc164098463)

[10.5. Устранение проблем 31](#_Toc164098464)

[11. Интеграция с внешними системами 32](#_Toc164098465)

[12. Тонкая настройка 33](#_Toc164098466)

[12.1. Внешняя поддержка RDP-сессий 33](#_Toc164098467)

[12.2. Множественные производственные календари 34](#_Toc164098468)

[12.3. Общие папки 34](#_Toc164098469)

[12.4. Блокировка робота агентом 35](#_Toc164098470)

[12.5. Настройка очереди проектов 35](#_Toc164098471)

[12.6. Таймаут, после которого робот считается в состоянии «Не доступен» 37](#_Toc164098472)

[12.7. Открытие/отключение/разлогинивание RDP-сессий 37](#_Toc164098473)

[12.8. Настройка папки для дампа секций журналов Роботов и Оркестратора 39](#_Toc164098474)

[12.9. Настройка стратегии очереди проектов для тенанта 39](#_Toc164098475)

[12.10. Настройка трансляции RDP-сессии 40](#_Toc164098476)

[12.11. Настройка параметров очереди обмена данными 42](#_Toc164098477)

[12.12. Настройка параметров оповещения 44](#_Toc164098478)

[12.13. Настройка очистки старых запусков 45](#_Toc164098479)

[12.14. Ограничение версии Студии 46](#_Toc164098480)

[12.15. Ограничение потока событий от триггеров 47](#_Toc164098481)

[12.16. Автоматическое временное замедление очереди проектов при пиках нагрузки на машинах роботов 51](#_Toc164098482)

[12.17. Отключение тенанта по умолчанию и блокировка встроенных не технических учетных записей 52](#_Toc164098483)

[12.18. Мерцающие RDP-сессий (виджет на главной странице) 54](#_Toc164098484)

[12.19. Проблемы с кодировкой при использовании кириллицы в названиях файлов ZIP-архива проекта 55](#_Toc164098485)

[12.20. Интеграция с CyberArk 55](#_Toc164098486)

[13. Схема взаимодействия Оркестратора и Робота 56](#_Toc164098487)

# Введение

«Primo RPA Orchestrator» (далее Оркестратор) предназначен для развертывания в локальной сети организации.

Локальная сеть может быть построена на основе AD[[1]](#footnote-1). В этом случае появляется возможность интеграции Оркестратора с AD организации для использования SSO[[2]](#footnote-2). Настройка интеграции с AD описана в разделе 8.

Если организация состоит из обособленных подразделений/филиалов, Оркестратор может функционировать в мультитенантном варианте. В этом случае подразделению/филиалу назначается тенант, который можно рассматривать как относительно независимый виртуальный экземпляр Оркестратора. Настройка мультитенантности описана в разделе 9.

Рисунок – Компоненты Оркестратора, Студия и Роботы

Компоненты Оркестратора и их связи между собой, с Роботами и внешними сервисами[[3]](#footnote-3) приведены ниже (рисунок 1):

Оркестратор содержит подсистемы/компоненты:

1. Серверы БД (PostgreSQL 13[[4]](#footnote-4)):
   1. Основная БД с настройками Оркестратора – Main DB (ltools[[5]](#footnote-5)).
   2. БД с пользователями и правами – Identity DB (ltoolsidentity).
   3. БД с лицензиями – License DB (ltoolslicense).
   4. БД с логом событий – Logs DB (ltoolslogs).
   5. БД NuGet-сервера – NuGet DB (ltoolsnuget).
2. Серверы приложений:
   1. WebApi – REST веб-API[[6]](#footnote-6).
   2. NuGet-сервер.
   3. RDP2 – служба для поддержки активных RDP-сессий (рисунок 2). Служба также может использоваться для трансляции активных RDP-сессий в UI.
   4. MachineInfo – служба определения параметров оборудования для работы с лицензиями.
   5. Front – веб-сервер для отдачи статического контента (UI администрирования в браузере, SPA) и реверс-прокси для WebApi, RobotLogs, RDP2[[7]](#footnote-7).
   6. States – служба вычисления системных состояний.
   7. Notification – служба нотификаций на e-mail.
   8. RobotLogs – служба приема логов от Роботов (частично от Оркестратора).
   9. LogEventsWebhook – служба интеграции логов посредством вэб-хуков[[8]](#footnote-8).
   10. RabbitMQ – брокер очередей сообщений.
3. Agent Оркестратора. Agent устанавливается на машине Робота как служба Windows и используется для управления Роботом[[9]](#footnote-9) и машиной Робота (логически является автономным продолжением Оркестратора на машине Робота). Если одна машина Робота делится между несколькими тенантами, для каждого тенанта устанавливается отдельный Agent на своем порту (5002, 5003, 5004, …).
4. Программа для шифрования паролей в конфигурационных файлах.

БД обязательно требуется развернуть на отдельной машине. Также, требуется Logs DB развернуть на отдельной машине от всех остальных БД, так как эта БД наиболее нагружена, настроить секционирование. Логи желательно сбрасывать через Logstash в ElasticSearch (на отдельном сервере) и периодически удалять из БД.

Робот – приложение, которое посредством Оркестратора разворачивается на специальным образом настроенной машине Робота и выполняет RPA-проект, который формируется заранее в Студии[[10]](#footnote-10).

На одной машине может работать несколько Роботов. Все машины Роботов должны быть настроены одинаково (версии ОС могут отличаться) и на каждой машине Робота должен быть установлен и настроен Agent. Машин Роботов может быть много.

Указанные порты (рисунок 1), далее используются при настройке конфигурационных файлов компонентов Оркестратора, машин Роботов и открытия портов на файерволе (в том числе аппаратном в сети организации).

Перед установкой Оркестратора нужна (желательно[[11]](#footnote-11)) чистая машина с последними обновлениями. И на неё должна быть скопирована папка с комплектом поставки (см. раздел 2). Это может быть любая папка, для определенности, пусть будет папка C:\Install[[12]](#footnote-12).

Для развертывания компонентов под OC Windows требуется PowerShell 7.1.3+.

Для выполнения команд и скриптов cmd и PowerShell в ОС Windows должны запускаться из-под Администратора. В OC \*nix под root (sudo).

После того, как комплект поставки скопирован на сервер, где будет развернут Оркестратор, требуется выполнить шаги раздела 4 настоящего руководства.

В комплект поставки включена внешняя аналитическая система Grafana, которая технически не является компонентом Оркестратора. Ее стоит рассматривать как сторонне средство для получения/визуализации аналитики по работе Оркестратора (таблица 1 №№ п/п 29 – 32, таблица 2 № п/п 36). Может быть заменена на любое аналогичное средство.

Оркестратор может выполнять почтовую рассылку о событиях, для этого он должен быть настроен для подключения к SMTP-серверу Организации.

Оркестратор может выполнять чтение писем для срабатывания триггеров заданий, для этого он должен быть настроен для подключения к POP3/IMAP-серверам Организации или MS Exchange Server.

# Требования к персоналу

Документ предназначен для системных администраторов, имеющих навыки:

1. Обязательно – уверенный пользователь ОС:
   1. Владеть инструментами ОС для навигации по папкам и файлам.
   2. Создавать и редактировать текстовые файлы в стандартном для ОС редакторе.
   3. Копировать файлы, копировать текст из файлов, сохранять файлы.
   4. Работать с архиватором zip.
   5. Знать что такое bat-файлы и уметь их создавать[[13]](#footnote-13).
2. Обязательно – опыт администрирования ОС:
   1. Запускать команды ОС и программы из-под Администратора/root.
   2. Работать в cmd[[14]](#footnote-14).
   3. Работать в PowerShell[[15]](#footnote-15).
   4. Иметь навыки администрирования AD, DNS, RDS [[16]](#footnote-16).
   5. Иметь навыки установки и администрирования служб.
   6. Иметь навыки администрирования пользователей и удаленного доступа.
3. Обязательно – минимальный опыт администрирования БД PostgreSQL/MS SQL Server[[17]](#footnote-17):
   1. Устанавливать PostgreSQL/MS SQL Server на сервер и производить базовую настройку.
   2. Запускать sql-скрипты.
4. Желательно – опыт администрирования БД:
   1. Секционирование таблиц[[18]](#footnote-18)
   2. Сбор метрик
   3. Конфигурирование под высокую нагрузку

# Комплект поставки

Оркестратор поставляется в комплектации (таблицы 1, 2):

Таблица 1 – Дистрибутивы[[19]](#footnote-19)

| №  п/п | Наименование файла | Размер файла (Кб)[[20]](#footnote-20) | Описание | Входит в поставку[[21]](#footnote-21) | Примечание |
| --- | --- | --- | --- | --- | --- |
|  | WebApi.zip | 78001 | WebApi | + | Для развертывания как службы Windows |
|  | WebApi-IIS.zip | 35316 | + | Для развертывания под IIS |
|  | WebApi-linux.zip | 76987 | + | Включает файл службы |
|  | States.zip | 41122 | States | + |  |
|  | States-linux.zip | 40544 | + | Включает файл службы |
|  | Notifications.zip | 66071 | Notifications | + |  |
|  | Notifications-linux.zip | 64709 | + | Включает файл службы |
|  | nginx-1.21.1.zip | 1711 | Nginx | + | Включает конфигурационный файл nginx (nginx.conf) и файлы самоподписанного SSL-сертификата для https (cert1.crt, cert1.rsa) |
|  | nginx-linux.zip | 785 | + |
|  | UI.zip | 2819 | Файлы UI Оркестратора в браузере, SPA | + |  |
|  | dotnet-hosting-7.0.11-win.exe | 108141 | Для хостинга NET 7-приложения под IIS | + |  |
|  | rewrite\_amd64\_en-US.msi | 5936 | Модули IIS, обеспечивающие функциональность реверс-прокси | + |  |
|  | requestRouter\_amd64.msi | 2440 | + |  |
|  | postgresql-13.4-1-windows.zip | 280372 | PostgreSQL | + | Включает скрипты начальной настройки БД ltoolslicense[[22]](#footnote-22) |
|  | postgresql-13-linux.zip | 7626 | + |
|  | pg\_ms.sh | 1 | + | Для кластера Postgres под Linux |
|  | MSSQL.zip | 1097 | MS SQL SERVER 2019 | + | Включает скрипты начальной настройки БД ltoolslicense |
|  | timescaledb-postgresql-13\_2.4.1-windows-amd64.zip | 4048 | TimescaleDB | + |  |
|  | timescaledb-postgresql-13-linux.zip | 3400 | + |  |
|  | rabbitmq.zip | 124132 | RabbitMQ server | + | Включает Erlang |
|  | rabbitmq-linux.zip | 41264 | + |
|  | Agent.zip | 47239 | Agent | + |  |
|  | Agent-linux.zip | 46507 | + |  |
|  | PowerShell-7.1.3-win-x64.msi | 97288 | PowerShell 7.1.3 | + |  |
|  | PrimoWorker.ps1 | 5 | PowerShell-скрипт для автоматизации настройки машины Робота и развертывания Агента | + |  |
|  | ChromeStandaloneSetup64.exe | 82509 | Браузер Google Chrome | + |  |
|  | PasswordEncryptor.zip | 35180 | Программа для шифрования паролей в конфигурационных файлах | + |  |
|  | grafana-8.0.6.windows-amd64.msi | 54972 | Внешняя аналитическая система Grafana | + | Не является компонентом Оркестратора |
|  | Роботы-1627543691525.json | 3 | Пример отчета в Grafana | + |  |
|  | v\_AllWorked-postgres.sql | 2 | View в БД для примера отчета в Grafana | + | PostgreSQL |
|  | v\_AllWorked-mssql.sql | 2 | + | MS SQL Server |
|  | dotnet7-centos.zip | 217288 | Пакеты NET7 для CentOS | + |  |
|  | dotnet7-ubuntu.zip | 142277 | Пакеты NET 7 для Ubuntu | + |  |
|  | syncthing-linux-amd64-v1.18.2.tar.gz | 9464 | Программа синхронизации папок | + | Для синхронизации папок с дистрибутивами робота и дампами Журнала |
|  | syncthing.service | 0 | + |
|  | syncthing-windows-amd64-v1.18.2.zip | 9582 | + |
|  | RDP-Disconnector.xml | 3 | Windows Task для восстановления сеанса после отключения RDP | + | Используются вместе |
|  | restore\_console.bat | 0 | Скрипт для перенаправления RDP-сессии в консоль | + |
|  | web.config | 4 | Должен использоваться при развертывании UI под IIS | + |  |
|  | MachineInfo.zip | 46743 | Служба определения параметров оборудования для работы лицензий | + |  |
|  | MachineInfo-linux.zip | 47494 | + |  |
|  | LogEventsWebhook.zip | 46369 | Служба Webhooks на события Оркестратора | + |  |
|  | LogEventsWebhook-linux.zip | 45086 | + |  |
|  | haproxy.zip | 1885 | HAProxy | + | Только для Linux. Включает конфигурационный файл, файлы сертификата |
|  | pg\_cron\_13-1.3.0-1.rhel8.x86\_64.rpm | 96 | Расширение PostgreSQL для выполнения заданий по расписанию для Linux | + |  |
|  | AgentInstaller.zip | 301868 | Инсталлятор для машины робота | + | Включает автоматическую регистрацию машины робота в Оркестраторе |
|  | RDP2.zip | 56931 | Служба поддержки активных RDP-сессий | + |  |
|  | RDP2-linux.zip | 52356 | + |  |
|  | RobotLogs.zip | 67289 | Служба приема логов от роботов | + |  |
|  | RobotLogs-linux.zip | 66423 | + |  |
|  | scr2.ps1 | 0 | Скрипт открытия теневой RDP-сессии | + |  |
|  | logstash.7z | 633659 | logstash | + |  |
|  | NuGet2.zip | 60268 | NuGet-сервер | + |  |
|  | NuGet2-linux.zip | 58959 | + |  |
|  | reset-ExchangeQueueStatistics-postgres.sql | 2 | Пересчет статистики по очередям обмена данными | + |  |
|  | reset-ExchangeQueueStatistics-mssql.sql | 2 | + |  |
|  | ИТОГО: | 3,07 Гб |  | 56 шт. |  |

Таблица 2 – Документация[[23]](#footnote-23)

| №  п/п | Путь | Наименование файла | Размер файла (Кб) | Входит в поставку |
| --- | --- | --- | --- | --- |
|  |  | Primo RPA 1.24.6 - Руководство пользователя.docx | 40819 | + |
|  |  | Развертывание Primo RPA 1.24.6 - Руководство администратора.docx | 1941 | + |
|  | Linux | Руководство по предварительной настройке машины Оркестратора под  CentOS 8.docx | 17 | + |
|  | Linux | Руководство по настройке машины Робота под Linux.docx | 502 | + |
|  | Linux | Руководство по установке Nginx под CentOS 8.docx | 73 | + |
|  | Linux | Руководство по установке Notifications под CentOS 8.docx | 69 | + |
|  | Linux | Руководство по установке PostgreSQL под CentOS 8.docx | 19 | + |
|  | Linux | Руководство по установке RabbitMQ под CentOS 8.docx | 105 | + |
|  | Linux | Руководство по установке States под CentOS 8.docx | 14 | + |
|  | Linux | Руководство по установке UI под CentOS 8.docx | 17 | + |
|  | Linux | Руководство по установке WebApi под CentOS 8.docx | 45 | + |
|  | Linux | Руководство по установке и настройке RDP2 под Astra Linux 1.7.docx | 140 | + |
|  | Linux | Руководство по установке и настройке RDP2 под CentOS 8.docx | 120 | + |
|  | Linux | Руководство по установке MachineInfo под CentOS 8.docx | 48 | + |
|  | Linux | Руководство по установке LogEventsWebhook под CentOS 8.docx | 111 | + |
|  | Linux | Руководство по открытию Swagger в Nginx под CentOS 8.docx | 159 | + |
|  | Linux | Руководство по установке RobotLogs под CentOS 8.docx | 19 | + |
|  | Linux | Руководство по установке NuGet2 под CentOS 8.docx | 69 | + |
|  | Linux/HA | Руководство по настройке Keepalived для Nginx на CentOS 8.docx | 23 | + |
|  | Linux/HA | Руководство по настройке кластера PostgreSQL на основе repmgr на CentOS 8.docx | 234 | + |
|  | Linux/HA | Руководство по развертыванию кластера RabbitMQ на CentOS 8.5.docx | 89 | + |
|  | Linux/HA | Руководство по развертыванию HAProxy на CentOS 8.5.docx | 128 | + |
|  | Windows | Руководство по настройке машины Робота под Windows 2016 Server.docx | 1728 | + |
|  | Windows | Руководство по предварительной настройке машины Оркестратора под Windows 2016 Server.docx | 19 | + |
|  | Windows | Руководство по установке Nginx под Windows 2016 Server.docx | 137 | + |
|  | Windows | Руководство по установке Notifications под Windows 2016 Server.docx | 71 | + |
|  | Windows | Руководство по установке PostgreSQL под Windows 2016 Server.docx | 358 | + |
|  | Windows | Руководство по установке MSSQL под Windows 2016 Server.docx | 685 | + |
|  | Windows | Руководство по установке PowerShell 7.1.3 под Windows.docx | 179 | + |
|  | Windows | Руководство по установке RabbitMQ под Windows 2016 Server.docx | 117 | + |
|  | Windows | Руководство по установке States под Windows 2016 Server.docx | 56 | + |
|  | Windows | Руководство по установке UI на nginx под Windows 2016 Server.docx | 17 | + |
|  | Windows | Руководство по установке WebApi и UI на IIS под Windows 2016 Server.docx | 3481 | + |
|  | Windows | Руководство по установке WebApi как службы под Windows 2016 Server.docx | 244 | + |
|  | Windows | Руководство по установке и настройке AD для SSO.docx | 2201 | + |
|  | Windows | Руководство по установке и настройке Grafana под Windows 2016 Server.docx | 3015 | + |
|  | Windows | Руководство по установке RDP2 под Windows 2016 Server.docx | 188 | + |
|  | Windows | Руководство по настройке теневого подключения к сессии робота.docx | 1952 | + |
|  | Windows | Руководство по установке MachineInfo как службы под Windows 2016 Server.docx | 111 | + |
|  | Windows | Руководство по установке LogEventsWebhook как службы под Windows 2016 Server.docx | 214 | + |
|  | Windows | Руководство по открытию Swagger в Nginx под Windows 2016 Server.docx | 119 | + |
|  | Windows | Руководство по открытию Swagger в IIS под Windows 2016 Server.docx | 119 | + |
|  | Windows | Руководство по установке RobotLogs как службы под Windows 2016 Server.docx | 712 | + |
|  | Windows | Руководство по установке NuGet2 под Windows 2016 Server.docx | 67 | + |
|  |  | Руководство по развертыванию фермы WebApi за nginx.docx | 284 | + |
|  |  | Спецификация WebApi на прием событий Оркестратора.docx | 34 | + |
|  |  | Установка и настройка Logstash.pdf | 242 | + |
|  |  | Руководство по настройке работы сервисов оркестратора с RabbitMQ через SSL.docx | 2618 | + |
|  |  | Секционирование существующих таблиц с журналом Робота и Оркестратора для PostgreSQL.docx | 135 | + |
|  |  | Секционирование существующих таблиц с журналом Робота и Оркестратора для SQLServer.docx | 18 | + |
|  |  | Фиксированное секционирование существующих таблиц с журналом Робота и Оркестратора для SQLServer.docx | 17 | + |
|  |  | Руководство по настройке PostgreSQL для работы через SSL.docx | 1450 | + |
|  |  | Руководство по настройке хранения секретов служб в отдельной БД.docx | 45 | + |
|  |  | ИТОГО: | 26,09 MбMбMбMбMб | 53 шт. |

Полный комплект поставки содержит 56 + 53 = 109[[24]](#footnote-24) файлов и весит ~3,5 Гб

Возможные варианты развертывания компонентов Оркестратора из дистрибутивов приведены в (3-х мерной) таблице 3. Варианты, для которых поддержка кластера не гарантирована производителем, выделены «\*».

Таблица 3 – Варианты развертывания компонентов Оркестратора

|  |  | MSSQL | PostgreSQL | WebApi | States | Notifications | Front | RobotLogs | RabbitMQ | RDP2 | MachineInfo | LogEventsWebhook | NuGet |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 |
| 1 | Windows | + | \* | + | + | + | + | + | + | + | + | + | + |
| 2 | Linux |  | + | + | + | + | + | + | + | + | + | + | + |

|  | Отдельная служба | nginx | IIS |
| --- | --- | --- | --- |
|  | 1 | 2 | 3 |
| 1 3 | + |  | + |
| 2 3 | + |  |  |
| 1 6 |  | \* (консольное приложение в автозагрузке)[[25]](#footnote-25) | + |
| 2 6 | + | + |  |
| 1 7 | + |  |  |
| 2 7 | + |  |  |
| 1 10 | + |  |  |
| 2 10 | + |  |  |
| 1 11 | + |  |  |
| 2 11 | + |  |  |
| 1 12 | + |  |  |
| 2 12 | + |  |  |

Для windows-серверов рекомендуется: 1 1 и 1 3 3, 1 6 3.

Для linux-серверов рекомендуется[[26]](#footnote-26): 2 2.

# Аппаратные требования

Аппаратные требования к Оркестратору[[27]](#footnote-27) для управления до 250[[28]](#footnote-28) Роботов одновременно приведены в таблице 4:

Таблица 4 – Аппаратные требования к Оркестратору

|  |  |  |  |
| --- | --- | --- | --- |
| №  п/п | Параметр | Требование | Примечание |
|  | CPU | 16 ядер |  |
|  | RAM | 16 Гб (рекомендуемая 32 Гб) |  |
|  | HDD | 1000 Гб (OS + Data) | Если RabbitMQ на отдельном сервере[[29]](#footnote-29), HDD рекомендуется 60 Гб |

У БД с логами ожидается быстрый рост объема. Ежедневный прирост – индивидуальная величина для разных пользователей (зависит от количества логов, которые создают работающие роботы) – может быть оценен по результатам эксплуатации. И на его основе может быть оценен более точно объем HDD и определена периодичность ротации/очистки логов. Величина 1000 Гб взята как усредненная оценка для комфортной работы.

Оценка объема HDD может быть произведена следующим образом:

Пусть 10 роботов параллельно работают 24 часа в сутки. Каждый обращается в Оркестратор 10 раз в секунду. Суммарно – 100 запросов в секунду. Пусть за один запрос Роботом отдается 0,1 Кб. Ежесуточный прирост = 100 \* (24 \* 60 \* 60) \* 0,1 = 864000 Кб = ~1 Гб.

Если используется поддержка RDP-сессий, надо исходить из оценки: один инстанс сервиса поддержки RDP[[30]](#footnote-30) на ~20 сессий. Это должна быть отдельная машина с не менее CPU 4[[31]](#footnote-31) ядрами и RAM 16 Гб. Или эти ресурсы должны быть включены в сервер Оркестратора. На поддержку одной RDP-сессию расходуется ~100 Мб[[32]](#footnote-32).

Аппаратные требования к машине робота приведены в таблице 5:

Таблица 5 – Аппаратные требования к машине робота

|  |  |  |  |
| --- | --- | --- | --- |
| №  п/п | Параметр | Требование | |
| Windows | Linux |
|  | CPU | 8 ядер | 8 ядер |
|  | RAM | 8 Гб | 8 Гб |
|  | HDD | 250 Гб (OS + Data) | 250 Гб (OS + Data) |

Аппаратные требования к машине робота зависят от количества одновременно работающих роботов и задач, которые выполняют роботы. Рекомендуется по 1 ядру CPU на каждого робота.

Обязательно требуется установить последние обновления ОС.

Если в качестве СУБД используется PostgreSQL, требуется:

1. Настроить использование pgbouncer
2. В конфигурационном файле настроить:
   1. work\_mem 1024 Мб
   2. effective\_cache\_size ~50-75% RAM
   3. shared\_buffers ~25-35% RAM

# Установка компонентов Оркестратора

Установка производится в последовательности, определенной в таблице 6. Все шаги полностью самодостаточны, содержат установку необходимых переменных окружения, открытия портов на файерволе ОС, настройку прав и т.п.

Таблица 6 – Последовательность установки компонентов

|  |  |  |  |
| --- | --- | --- | --- |
| №  п/п | Наименование | Подсистема | Примечание |
|  | Предварительная настройка | Оркестратор |  |
|  | Установка PostgreSQL (+pgbouncer) или MS SQL SERVER |  |
|  | Установка RabbitMQ | Если установлен ранее, требуется удалить все очереди. Может быть установлен для соединения по SSL (таблица 2, № п/п 52) |
|  | Установка WebApi и MachineInfo | Включает описанное ниже шифрование паролей в конфигурационных файлах. При установке на линукс chmod назначить в соответствии с Таблица 2, № 3, п. 6 |
|  | Установка Notifications |
|  | Установка NuGet |
|  | Установка States |
|  | Установка RDP2 |
|  | Установка RobotLogs |
|  | Установка Front |  |
|  | Настройка машины Робота | Робот |  |
|  | Установка Agent на машине Робота |  |

Изначально можно пропустить установку №№ п/п 5 – 7 и выполнить установку этих компонентов позже. Остальные никакие шаги не могут быть пропущены. Как минимум, требуется сверка ранее настроенного в организации окружения с документацией, идущей в комплекте поставки.

Установка LogEventsWebhook может быть произведена позже, когда будет разработан и развернут интеграционный шлюз Заказчика (см. главу 11).

# Варианты развертывания сервера приложений Оркестратора для ОС Windows 2016 Server

Имеется 2 варианта развертывания WebApi и Front (таблица 7) для ОС Windows 2016 Server:

Таблица 7 – Варианты развертывания WebApi и Front для ОС Windows 2016 Server

|  |  |  |  |
| --- | --- | --- | --- |
| №  п/п | Вариант развертывания WebApi и Front | Особенности SSO | Ссылка на руководства из таблицы 2 |
|  | WebApi и Front работают под IIS | Сервер может быть включен в AD или можно использовать keytab-файл | № п/п 33 |
|  | WebApi – служба Windows, Front – nginx | Только keytab-файл | №№ п/п 25, 32, 34 |

keytab-файл – файл, полученный в результате команды ktpass при регистрации Front сервиса в AD. Путь к полученному keytab-файлу настраивается в конфигурационном файле WebApi в секции ActiveDirectory в KerberosKeytabPath (рисунок 2):

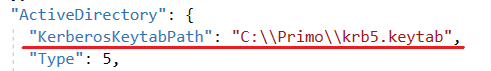


Рисунок – Путь до keytab-файла

При варианте «WebApi и Front работают под IIS» в конфигурационном файле WebApi UseIISIntegration должна быть установлена в true (рисунок 3):

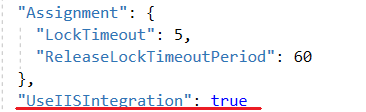


Рисунок – Настройка для варианта развертывания «WebApi и Front работают под IIS»

При варианте «WebApi – служба Windows, Front – nginx» UseIISIntegration должна быть установлена в false.

При развертывании под Linux WebApi и nginx устанавливаются как демоны Linux. SSO возможно только с использованием keytab-файла.

# Атрибуты безопасности

Конфигурационные файлы поставляются с зашифрованными паролями. Используется 4 предустановленных пароля (таблица 8):

Таблица 8 – Предустановленные пароли

|  |  |  |  |
| --- | --- | --- | --- |
| №  п/п | Открытый пароль | Зашифрованный пароль | Назначение |
|  | Qwe123!@# | JLWIyl1xZNDVVx8tcVllOg== | ActiveDirectory, SslCert, RabbitMQ, ApiKey, пользователь admin оркестратора |
|  | postgres | 49EqQ30zfcQTWxEGYE/mSw== | БД: ltools, ltoolsidentity, ltoolslicense, ltoolslogs, ltoolsnuget, ltoolssecrets  (PostgeSQL) |
|  | 123!@#Qwe123!@#Qwe123 | 8TQ18jklcX3CPIsG6cVeb4iClxAxrqSabembVRUnvXQ= | Email, с которого происходит рассылка |
|  | sa | OrrNv0YQhLhnAwG+CuUPKA== | БД: ltools, ltoolsidentity, ltoolslicense, ltoolslogs, ltoolsnuget, ltoolssecrets  (MS SQL SERVER) |

Если требуется сменить предустановленный пароль, например, у пользователя БД меняется пароль, то его требуется сначала зашифровать при помощи программы   
Primo.Orchestrator.PasswordEncryptor.exe[[33]](#footnote-33) (рисунок 4):

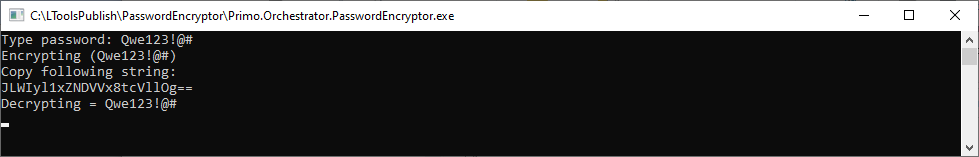


Рисунок – Шифрование паролей при помощи программы Primo.Orchestrator.PasswordEncryptor.exe

Зашифрованные пароли из окна терминала нужно будет скопировать (зашифрованный пароль находится под строкой «Copy following string:») в конфигурационные файлы appsettings.ProdWin.json[[34]](#footnote-34) и appsettings.Worker.json.

Соль для алгоритма шифрования паролей находится в одной из обфусцированных dll Оркестратора, поэтому шифрование доступно только с помощью Primo.Orchestrator.PasswordEncryptor.exe.

Продолжительность сессии (времени жизни токена) в минутах настраивается в appsettings.ProdWin.json в секции Security:Jwt (рисунок 5):

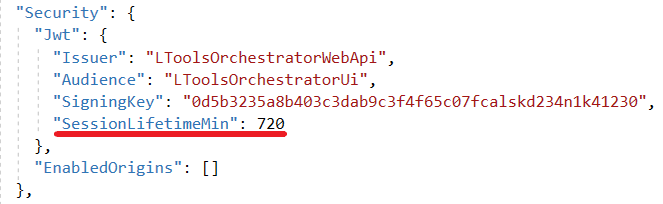


Рисунок – Настройка времени жизни токена

Сложность пароля (длина, обязательные символы и т.п.) настраивается в appsettings.ProdWin.json в секции Security:Password (рисунок 6):

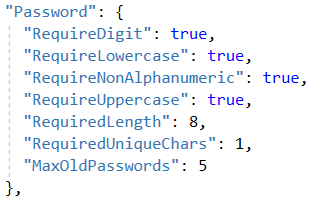


Рисунок – Настройка сложности пароля и истории старых паролей

За максимальное количество старых паролей в истории (чтобы пользователь не мог использовать недавние пароли при смене пароля) отвечает параметр MaxOldPasswords.

В секции Security:Lockout настраиваются параметры (время блокировки и максимальное количество неудачных попыток) временной блокировки пользователя при неуспешных попытках ввода пароля (рисунок 7):

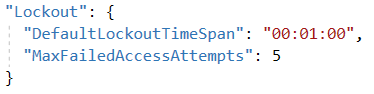


Рисунок – Настройка параметров временной блокировки пользователя при неуспешных попытках ввода пароля

В БД пароли для машин роботов хранятся в зашифрованном виде. Если по какой-то причине пароль не расшифровывается (если его неверно исправить в БД вручную) при работе приложения в БД сохранится случайная зашифрованная строка. Работать такой пароль не будет, его надо будет через UI Оркестратора поменять на правильный.

# Контроль целостности конфигурационных файлов

Служба Primo.Orchestrator.WebApi проверяет целостность своих конфигурационных файлов каждый раз при запуске. Если с прошлого запуска файл изменился, в Оркестраторе фиксируется событие типа «Инцидент безопасности» с названием «Конфигурационный файл изменен»   
(рисунок 8):

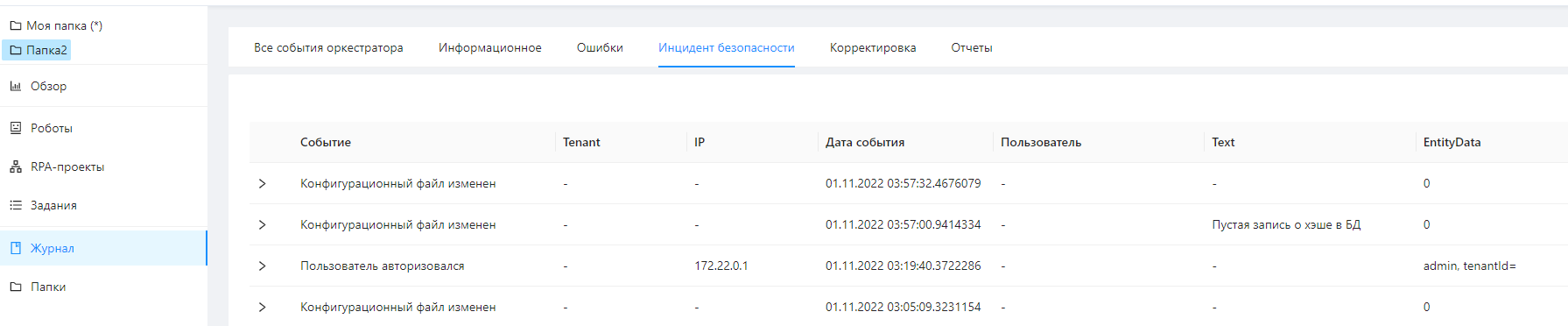


Рисунок – Журнал Оркестратора с событиями «Конфигурационный файл изменен»

Просмотреть прошлое содержание конфигурационных файлов можно, скачав их по ссылке «Просмотреть прошлый конфигурационный файл» (рисунок 9):

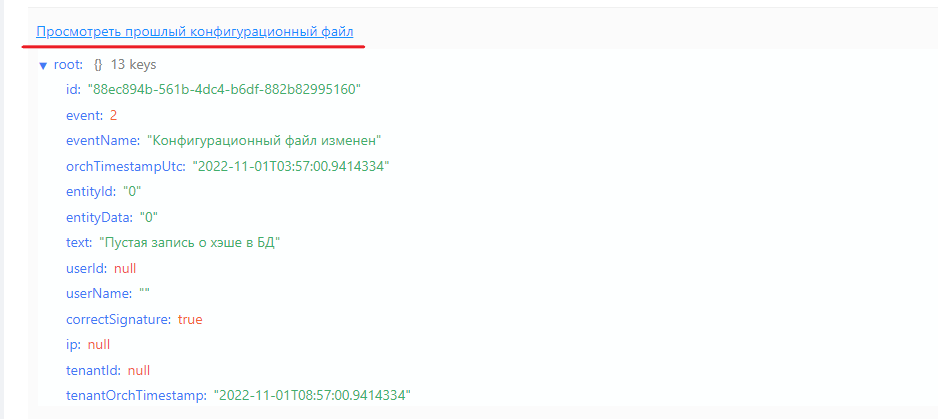


Рисунок – Ссылка для скачивания прошлого конфигурационного файла

Файлы скачиваются как один текстовый файл (\*.txt). Логически файлы appsettings.json и appsettings.ProdWin[[35]](#footnote-35).json разделяет строка разделитель <\*SPLIT\*> внутри этого скачанного файла. Дальше, сторонними средствами, например, Notepad++, можно сравнить файлы – скачанные и существующие конфигурационные файлы – на различие.

# Интеграция с AD

Интеграция с AD осуществляется следующим образом:

1. Front Сервис Оркестратора регистрируется в AD и DNS (см. таблица 2, № п/п 35).

2. В зависимости от ОС и варианта развертывания:

Для Windows 2016 Server и варианта развертывания «WebApi и Front работают под IIS»   
(см. таблица 7, № п/п 1) сервер с IIS может быть включен в AD. Тогда в IIS для узла Primo.WebApi должна быть разрешена аутентификация Windows (рисунок 10), а пул приложений этого узла должен работать под доменной SPN учетной записью, полученной при регистрации сервиса в AD   
(рисунок 11):

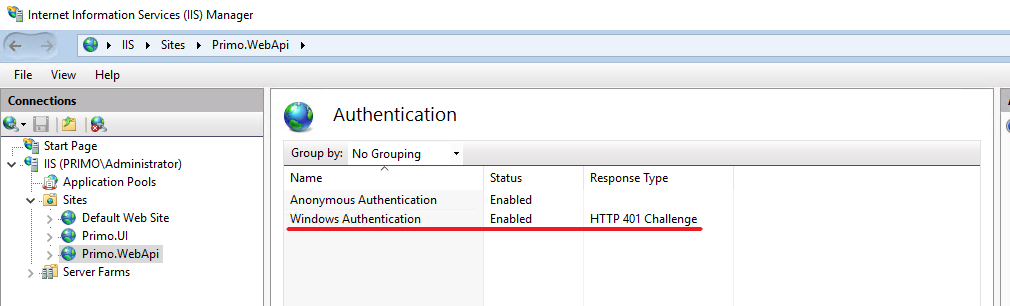


Рисунок – Разрешение аутентификация Windows для узлов в IIS

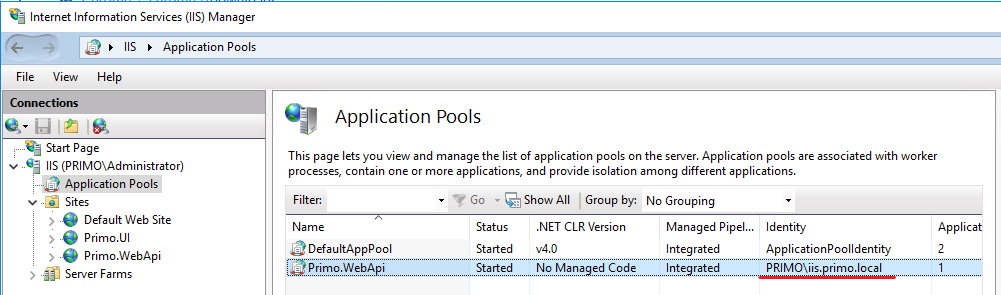


Рисунок – Пользователь, под которым работает пул приложений Primo.WebApi

Для ОС Linux, или варианта развертывания для Windows 2016 Server «WebApi – служба Windows, Front – nginx» (см. таблица 7, № п/п 2) или сервер с IIS не включен в AD – только на основе keytab-файла.

3. В конфигурационном файле WebApi в секции ActiveDirectory нужно прописать настройки для каждого AD, чтобы в UI Оркестратора роль Оркестратора могла быть ассоциирована с группой AD (рисунок 12):



Рисунок – Настройки AD

Для AD-групп в GroupsFilter может быть произведена более точная фильтрация[[36]](#footnote-36), например, по названиям групп:

Путем явного перечисления через ИЛИ:

(&(ObjectClass=group)(|(cn=primo)(cn=another)))

С использованием регулярного выражения:

(&(ObjectClass=group)(cn=prim\*))

Если фильтрация не используется, может возникнуть ошибка при запросе сильно большого количества групп.

Если не используется одновременно несколько AD, лишний AD должен быть удален.

Наименование AD в конфигурационном файле WebApi рекомендуется выбирать в соответствии со значениями DC в StartPoint, например, для primo1.orch DC=primo1, DC=orch.

В таблице 9 приведено описание параметров для настройки AD, которые администратор может менять:

Таблица 9 – Описание параметров для настройки AD

| №  п/п | Наименование параметра | Назначение | Примечание |
| --- | --- | --- | --- |
|  | Type | Настройка формы авторизации. Возможные значения 1 (авторизация только в Оркестраторе) и 5 (авторизация в Оркестраторе или SSO) |  |
|  | primo1.orch, primo2.orch и т.д. | Точные названия AD, без сокращения, со всеми поддоменами. Должны быть заменены | Рекомендуется использовать префиксы наименований, чтобы отфильтровать в GroupsFilter |
|  | Host | Адрес AD |  |
|  | UseSsl | Используется LDAPS |  |
|  | AcceptUntrustedCertificate | При использовании LDAPS допустимость невалидного SSL-сертификата |  |
|  | AdminUserName | Имя учетной записи, имеющей права на просмотр каталога AD |  |
|  | AdminPassword | Зашифрованный пароль | См. раздел 6 |
|  | StartPoint | Уровень, с которого просматривается каталог. Параметры DC должны соответствовать названию AD. Параметр CN не меняется |  |
|  | Tenants | Массив идентификаторов тенантов, которые относятся к AD | См. раздел 8 |
|  | TrustedDomains | Массив доменов (в нижнем регистре), из которых пользователь может авторизоваться в домене (при настроенных доверительных отношениях между доменами), в котором зарегистрирован сервис оркестратора |  |

Рабочие станции, на которых запускается UI Оркестратора, должны быть включены в AD.

Если возникают проблемы при настройке SSO, требуется проверить настройку по шагам:

Проверка, что группа AD пользователя, который авторизуется в Оркестраторе по SSO, привязана к роли Оркестратора в БД: <имя роли> - имя роли Оркестратора, к которой должна быть привязана группа AD <имя группы>

Шаг 1:

На рабочей станции, включенной в AD (через cmd):

# whoami /groups

Среди групп должна быть (определяется по SID) группа AD, которая привязана к роли оркестратора в БД (см. Шаг 3)

Шаг 2:

На сервере контроллера AD (через PowerShell):

# Get-ADGroup -Identity <имя группы>

SID группы AD должен быть привязан к роли Оркестратора в БД (см. Шаг 3) и должен содержаться среди SID на Шаге 1

Шаг 3:

В БД ltoolsidentity:

select r."Name",

r."TenantId",

ad."Ad",

ad."AdGroups"

from "Roles" r

inner join "RoleAds" ad on ad."RoleId" = r."Id"

where r."Name" = '<имя роли>'

Запрос вернет привязку по SID к <имя роли> групп AD.

Поле RoleAds.Ad в БД ltoolsidentity, название AD в конфиге WebApi (секция ActiveDirectory:MultyForest), и название AD, которое приходит из AD при авторизации, должны совпадать. Если они отличаются, необходимо привести их в соответствие и перезапустить службу Primo.Orchestrator.WebApi.

Шаг 4 (не обязательный)[[37]](#footnote-37):

Получаем при помощи запроса в PowerShell из AD все группы, предназначенные для использования в Оркестраторе (рисунок 13):

# Get-ADGroup -Filter 'Name -like "prim\*"' | select Name, SID | foreach { "('" + $\_.Name + "', '" + $\_.SID + "')," }

Например, используя префикс prim (может использоваться какая-то другая фильтрация).

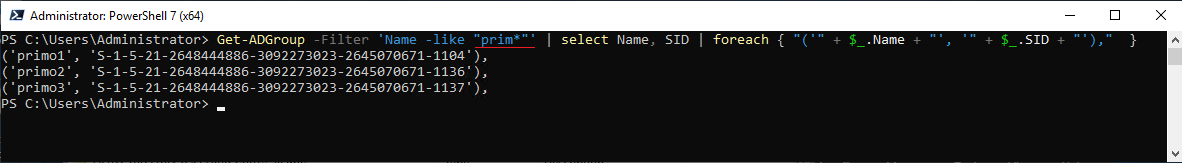


Рисунок – Запрос AD-групп, предназначенных для использования в Оркестраторе

Копируем полученный результат в SQL-запрос к БД ltoolsidentity (рисунок 14):

WITH t1 AS

(SELECT \*

FROM (VALUES

('primo1', 'S-1-5-21-2648444886-3092273023-2645070671-1104'),

('primo2', 'S-1-5-21-2648444886-3092273023-2645070671-1136'),

('primo3', 'S-1-5-21-2648444886-3092273023-2645070671-1137')) AS q ("Name", "SID"))

SELECT t2.\*, t1."Name"

FROM

(SELECT r."Name",

r."Description",

r."TenantId",

ra."Ad",

ra."SID"

FROM "Roles" r

CROSS JOIN LATERAL

(SELECT "Ad", "RoleId", "AdGroups",

unnest(string\_to\_array("AdGroups", '|')) "SID"

FROM "RoleAds") ra

WHERE ra."RoleId" = r."Id") t2

LEFT JOIN t1 ON t2."SID" = t1."SID"

WHERE "Ad" = 's1.primo1.orch'

Здесь вместо s1.primo1.orch нужно подставить название своего домена.

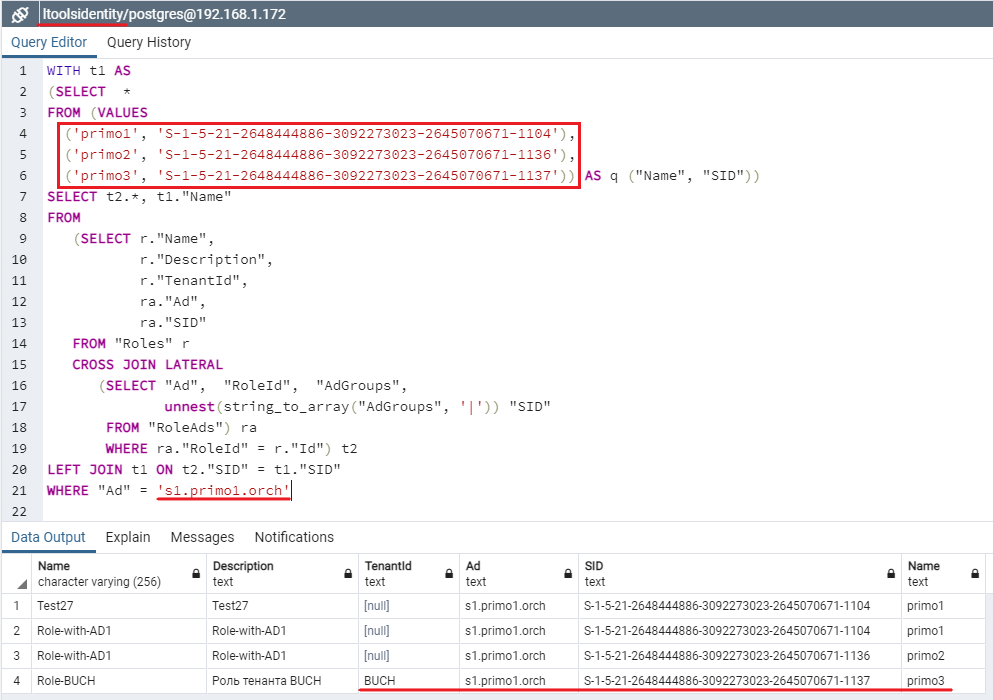


Рисунок – Результат запроса принадлежности AD-групп ролям/тенантам

По результату запроса видно, что AD-группа primo3 относится к тенанту BUCH. Это значит, что Оркестратор не даст привязать AD-группу primo3 к роли другого тенанта, так как AD-группа может быть привязана только к ролям одного тенанта[[38]](#footnote-38).

Если привязка правильная, имеет смысл переименовать AD-группу primo3 в primo3-BUCH, чтобы в UI оркестратора была видна принадлежность AD-группы к тенанту (через роль).

Если привязка не правильная, и AD-группа primo3 должна использоваться в другом тенанте, сначала нужно отвязать её от ролей тенанта BUCH.

# Мультитенантность

Мультитенантность настраивается в конфигурационном файле WebApi в секции Tenants (рисунок 15):



Рисунок – Настройка мультитенантности

В таблице 10 приведено описание параметров для настройки мультитенантности:

Таблица 10 – Описание параметров для настройки мультитенантности

| №  п/п | Наименование параметра | Назначение | Примечание |
| --- | --- | --- | --- |
|  | FromAppsettings | Всегда должно быть true | Зарезервировано для новых версий Оркестратора |
|  | TimeOffset | Смещение времени в часах от стандартного времени для отображения в журнале Робота и Оркестратора для дефолтного тенанта |  |
|  | IncomingEmail | Параметры ящика для получения почты для дефолтного тенанта |  |
|  | Agent | Параметры обращения к агенту |  |
|  | Items | Массив тенантов |  |
|  | Items[].Id | Идентификатор тенанта | Также используется при конфигурировании агента на машине робота |
|  | Items[].Name | Наименование тенанта |  |
|  | Items[].TimeOffset | То же что и № п/п 2, только для не дефолного тенанта |  |
|  | Items[].IncomingEmail | То же что и № п/п 3, только для не дефолного тенанта |  |
|  | Items[].Agent | То же что и № п/п 4, только для не дефолного тенанта | За счет этой настройки можно на одной машине робота развернуть несколько агентов для разных тенантов. Каждый агент будет слушать входящие запросы на своем порту |
|  | IncomingEmail.UserName | Пользователь – полное имя почтового ящика. Например, primo.rpa@mail.ru |  |
|  | IncomingEmail.Password | Пароль к почтовому ящику | Пароль шифруется при помощи программы LTools.Orchestrator.PasswordEncriptor.exe |
|  | IncomingEmail.Pop3 | Адрес Pop3-сервера. Например, pop.mail.ru | Если используется Imap не заполняются |
|  | IncomingEmail.Pop3Port | Порт Pop3-сервера. Например, 995 |
|  | IncomingEmail. RequireAuthenticate | Требуется ли аутентификация |  |
|  | IncomingEmail.UseSsl | Используется ли SSL/TLS/STARTTLS |  |
|  | IncomingEmail. UseStandartNotSSLPort | Используется стандартный порт[[39]](#footnote-39). В этом случае игнорируется флаг UseSsl |  |
|  | IncomingEmail.Imap | Адрес Imap-сервера. Например, imap.mail.ru | Если используется Pop3 не заполняются |
|  | IncomingEmail.ImapPort | Порт Imap-сервера. Например, 993 |
|  | IncomingEmail.ImapFolder | Папка с письмами. Например, INBOX |

Если используется SSO, то привязка тенантов к AD также настраивается в секции ActiveDirectory (рисунок 16):



Рисунок – Привязка тенантов к AD

В массиве Tenants перечисляются Id тенантов, которые привязаны к AD. Пустая строка – это Id дефолтного тенанта.

## Мультитенантная AD-авторизация

Если допускается использование одной доменной учетной записи в нескольких тенантах (мультитенантная AD-авторизация), необходимо в секции ActiveDirectory поставить параметру MultyTenantsGroup значение true (рисунок 17):

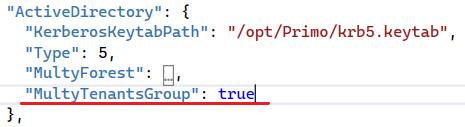


Рисунок – Разрешение мультитенантной AD-авторизации

В этом случае пользователь для разрешения неоднозначности выбора тенанта по роли Оркестратора должен будет явно указывать тенант при AD-авторизации.

# Первоначальная настройка Оркестратора

После развертывания Оркестратора он еще не готов к работе – требуется получение и установка лицензий[[40]](#footnote-40) на Оркестратор и Роботов и публикация дистрибутивов Робота[[41]](#footnote-41).

## Получение лицензий

Лицензия приобретается у вендора на основе запроса лицензии через интерфейс Оркестратора – верхнее правое меню, иконка «Шестеренка», меню «Лицензии» (рисунок 18), кнопка «Запрос на лицензию» и форма создания запроса (рисунок 19):

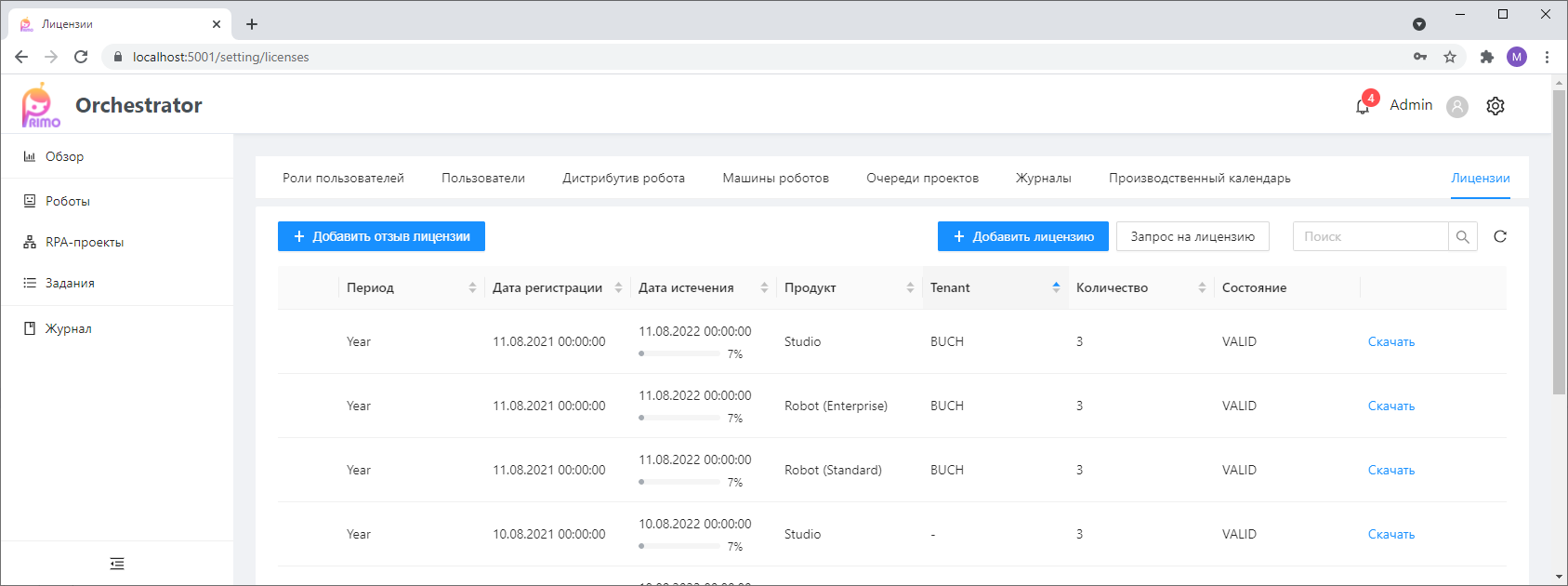


Рисунок – Все лицензии

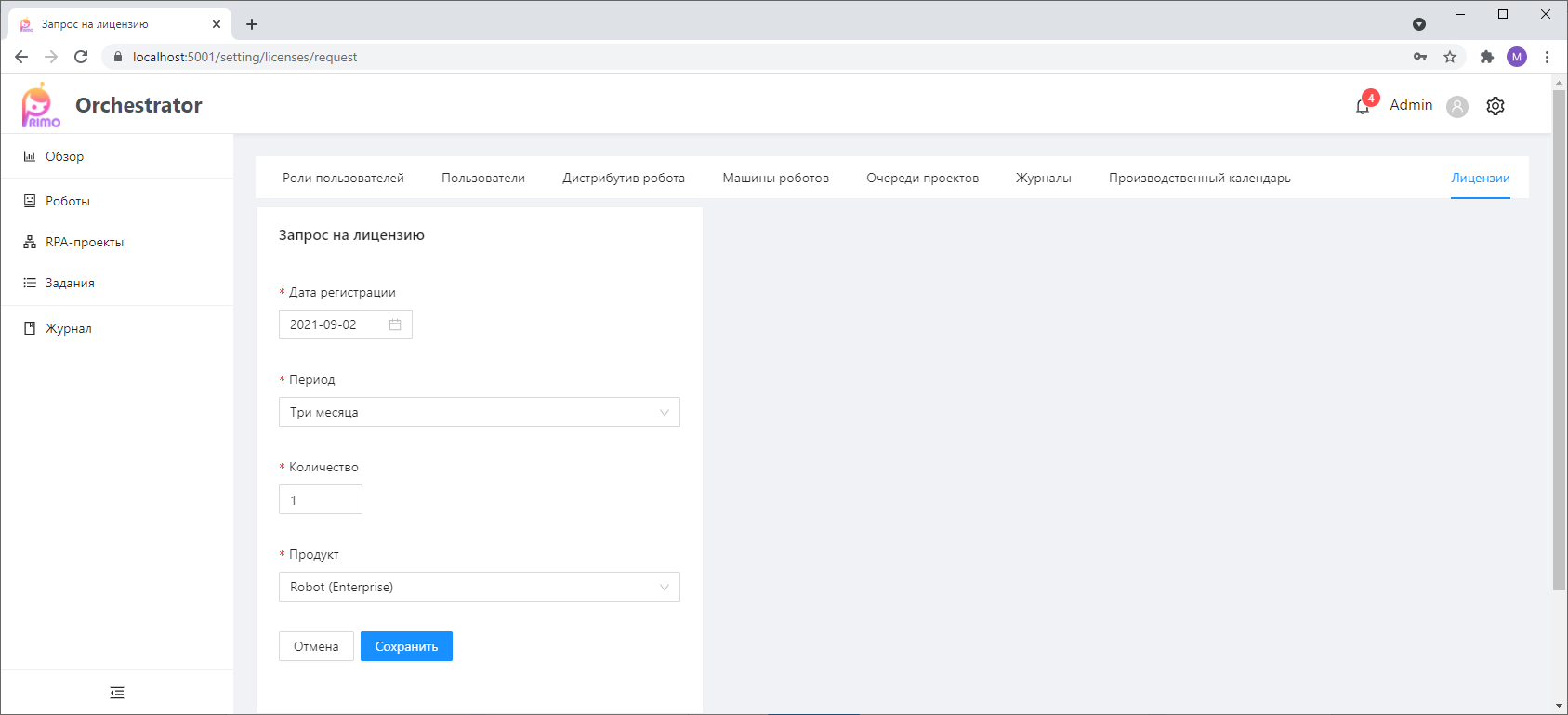


Рисунок – Форма создания запроса на лицензию

Запрос на лицензию необходимо сохранить в текстовый файл (например, robot.txt) и отправить Вендору. В ответ вендор отправит файл лицензии с расширением .license (например, robot.license).

## Установка лицензий

После получения файлов лицензий \*.license их необходимо добавить в Оркестратор по кнопке «Добавить лицензию» (рисунок 20):

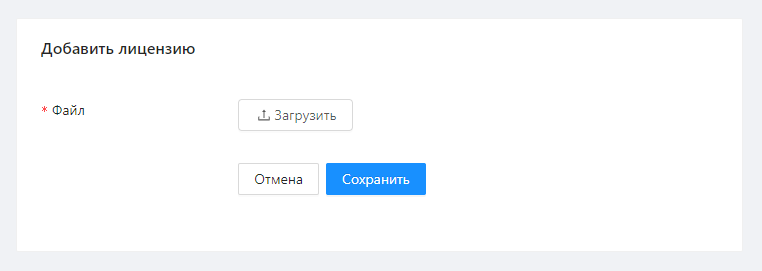


Рисунок – Форма добавления лицензии

и убедиться в их валидности и что не истек срок (рисунок 21):

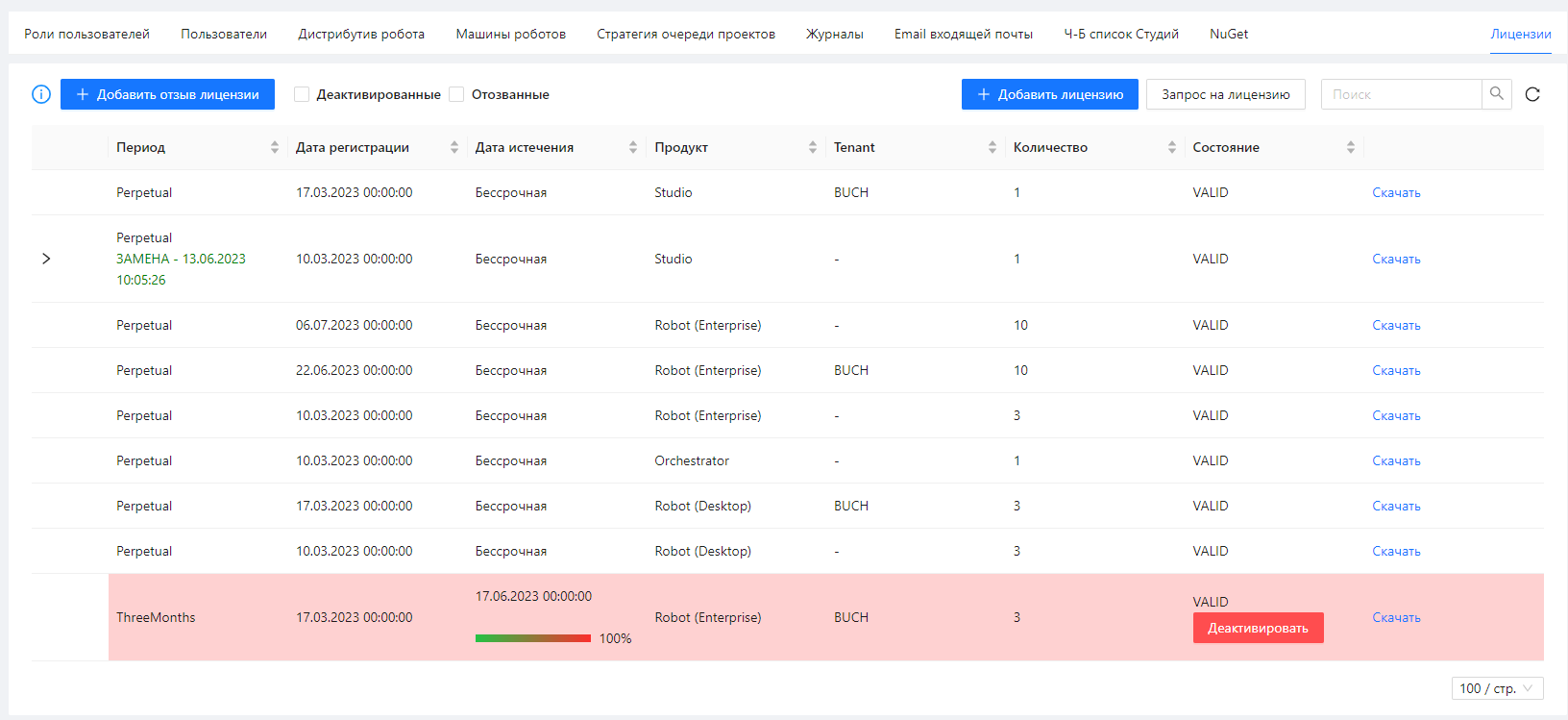


Рисунок – Отображение валидности и даты истечения лицензий

Просроченные или не валидные лицензии можно деактивировать, чтобы они не мешали в общем списке и не засорялся лог WebApi.

## Публикация дистрибутива

Через интерфейс Оркестратора нужно закачать файлы дистрибутивов Робота[[42]](#footnote-42)   
(рисунок 22):

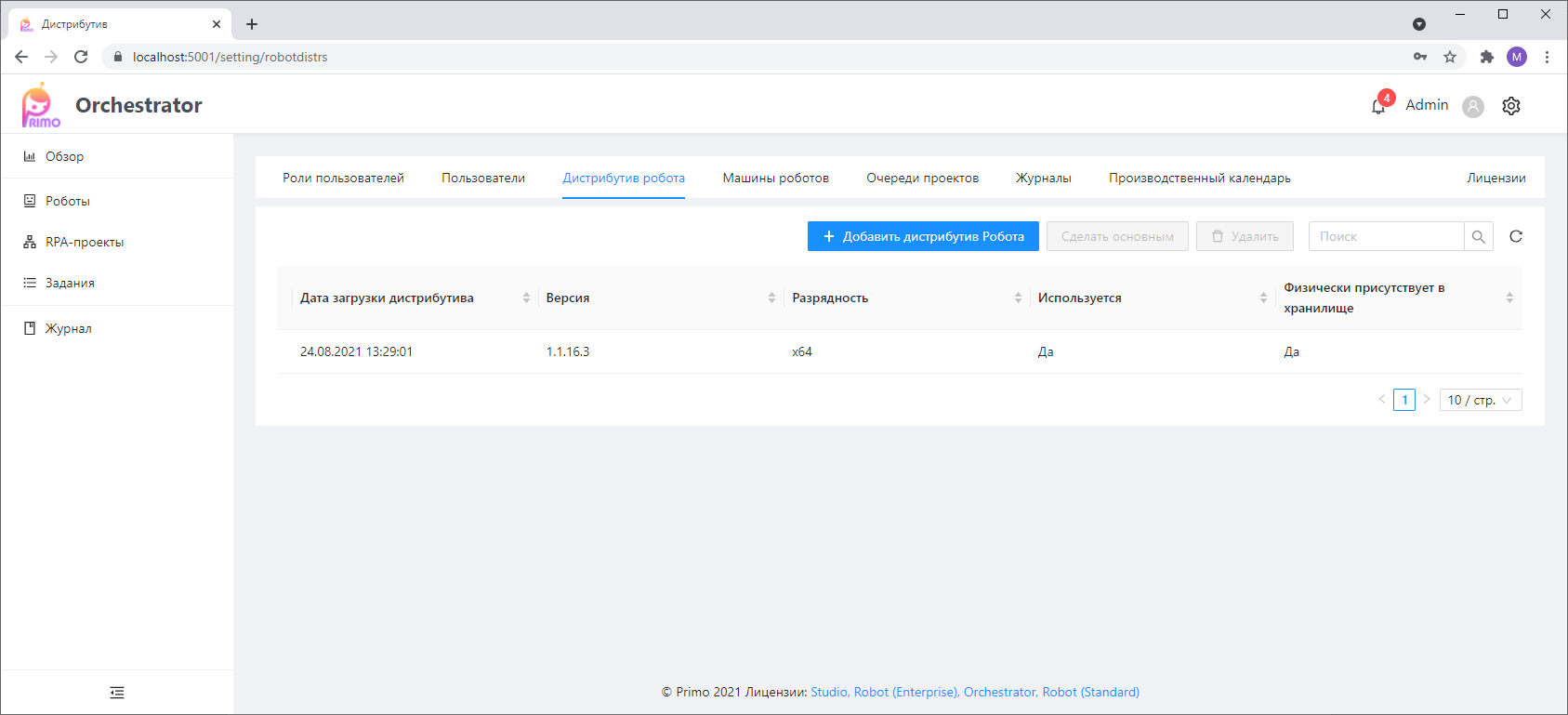


Рисунок – Дистрибутивы робота

## Настройка и регистрация машин Роботов

Машина Робота настраивается на основе «Руководство по настройке машины Робота.docx» из комплекта поставки. Перед её регистрацией в Оркестраторе машина Робота должна быть доступна из Оркестратора.

Через интерфейс Оркестратора регистрируем машины Роботов и убеждаемся в их доступности (рисунок 23):

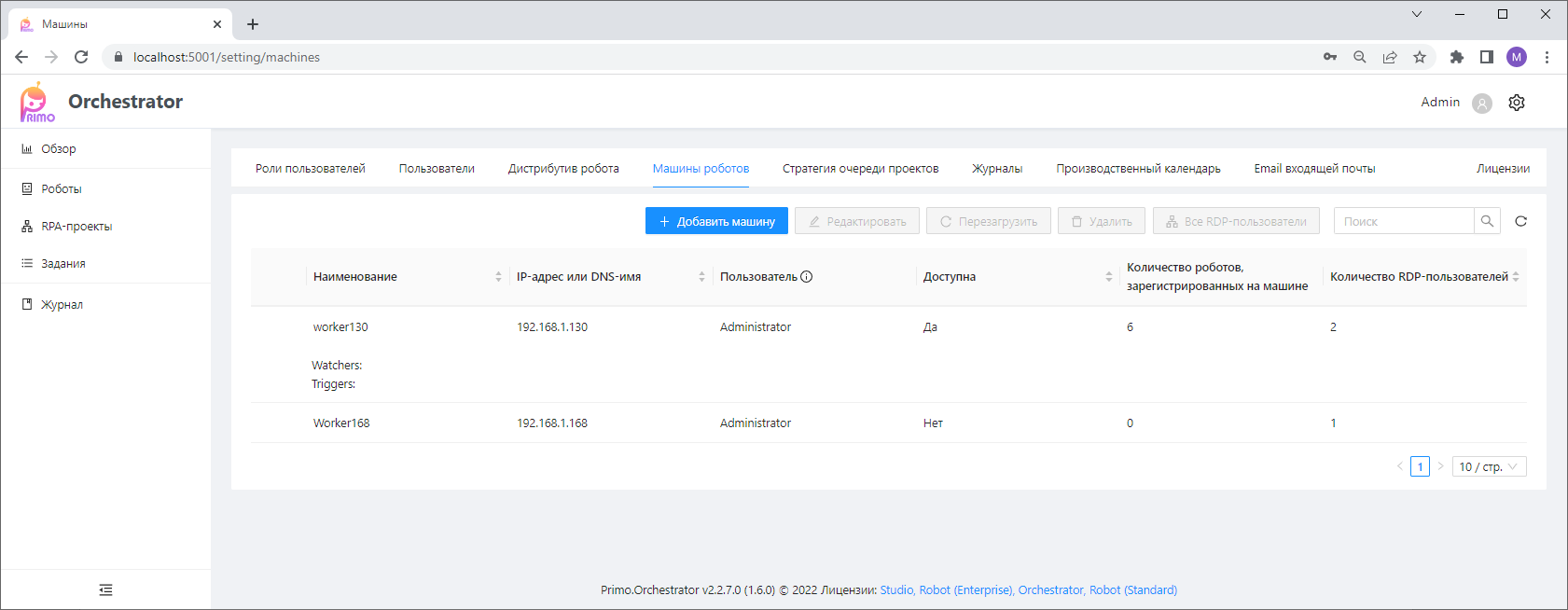


Рисунок – Зарегистрированные в Оркестраторе машины роботов

Пользователь – это пользователь (локальный или доменный, не обязательно Administrator), от имени которого на машине Робота Агент в процессе работы создает задание в Task Scheduler по запуску робота (Рисунок 24):

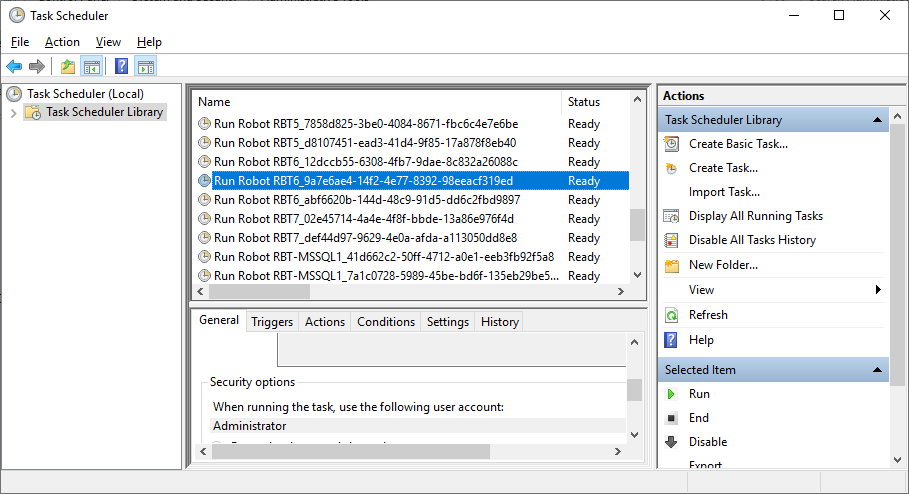


Рисунок – Задания в Task Scheduler по запуску робота

Теперь Оркестратор готов к работе: можно развертывать Роботов на настроенных машинах Роботов и управлять выполнением RPA-проектов Роботами.

## Устранение проблем

Если возникли проблемы при развертывании/запуске роботов, проверить по списку:

1. Секция Оркестратора «RobotDeployment: OrchBaseUrl» – адрес Оркестратора, который должен знать Робот. Симметрично – секция Агента «Orchestrator: BaseUrl» – эти адреса должны совпадать.
2. Порт в секции Агента «Kestrel: EndPoints: ExternalHttps: Url» – на этом порту будет слушать служба Агента. Симметрично – секция Оркестратора «Agent: Port» для дефолтного Агента и настройки дополнительных Агентов для поддержки RDP-сессий.
3. Секции Оркестратора «RobotDeployment: PortMin» и «RobotDeployment: PortMax» – диапазон портов, используемый для выдачи порта Роботу, на порту из этого диапазона запушенный Робот будет слушать.
4. WebApi должен быть доступен с машины робота. Проверить – непосредственно в браузере на машине робота запустить Оркестратор.
5. Агент должен быть доступен с машины WebApi. Проверить – https://{IP}:5002/api/Worker. Можно тоже непосредственно в браузере, Ctrl+Shift+i, вкладка Сеть, вернет http-статус 204.
6. Сетевая связность, файервол, открытые порты, правила фильтрации (обратиться к сетевым администраторам).

# Интеграция с внешними системами

Для интеграции с внешними системами имеется возможность использовать Webhooks на события Оркестратора (рисунок 25):



Рисунок – Интеграция с внешней системой

Для этого требуется:

Развернуть службу LogEventsWebhook (см. таблица 2, №№ п/п 15, 41), настроить её на получение событий из RabbitMQ и обращение к WebApi внешней системы.

Разработать и развернуть WebApi (интеграционный шлюз) с 2-мя end-point для приема событий от Webhooks. Этот WebApi Заказчик разрабатывает самостоятельно в соответствии со спецификацией (см. таблица 2, № п/п 47).

Разрешить Webhooks на события Оркестратора в конфигурационном файле   
(Integration: EventWebhook:Enabled) оркестратора (рисунок 26):

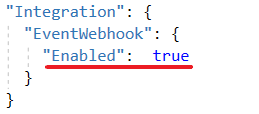


Рисунок – Разрешение Webhooks на события Оркестратора

После настройки интеграции проверить её работу можно, сопоставляя записи журнала Оркестратора с записями в хранилище, в котором сохраняются события от Webhooks.

# Тонкая настройка

## Внешняя поддержка RDP-сессий

Когда роботу для работы требуется внешняя поддержка RDP-сессии, перед запуском робота Оркестратор запрашивает открытие сессии в сервисе RDP. Процесс открытия RDP-сессии занимает некоторое время. Параметры открытия RDP-сессии настраиваются в секции   
RobotStart (рисунок 27):

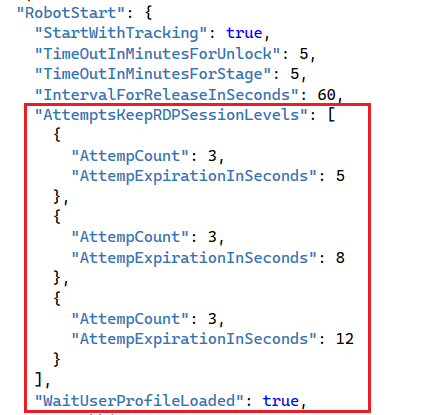


Рисунок – Параметры открытия RDP-сессии

В таблице 11 приведено описание параметров открытия RDP-сессии:

Таблица 11 – Описание параметров открытия RDP-сессии

| №  п/п | Наименование параметра | Назначение | Примечание |
| --- | --- | --- | --- |
|  | AttemptsKeepRDPSessionLevels | Массив пар параметров AttempCount и AttempExpirationInSeconds – уровней стратегии открытия RDP-сессии. Каждый последующий уровень должен иметь большее значение AttempExpirationInSeconds, чем предыдущий | Попытки открыть начинаются с уровня 0 (первый элемент массива), при исчерпании количества попыток происходит переход на следующий уровень, пока уровни не закончатся.  Не рекомендуется использовать больше 3 уровней |
|  | AttempCount | Количество попыток открытия RDP-сессии в рамках уровня |  |
|  | AttempExpirationInSeconds | Время задержки (сек) между попытками | Не гарантировано, при наличии многих RDP-пользователей может отличаться от указанного в настройке |
|  | WaitUserProfileLoaded | Ожидать загрузки профиля пользователя | RDP-сессия когда открывается Оркестратором, проверяется опросом машины робота загрузился ли профиль RDP-пользователя. Пока не закончатся попытки AttempCount или не загрузится профиль |

Общая рекомендация относительно расположения службы RDP2 – желательно на отдельной машине. На машине робота её можно развертывать если она обслуживает ТОЛЬКО пользовательский сессии на этой машине (автосессии).

## Множественные производственные календари

Включение функции настраиваются в секции ProductionCalendar (рисунок 28):

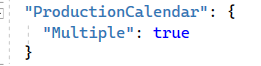


Рисунок – Включение/отключение множественных производственных календарей

## Общие папки[[43]](#footnote-43)

Включение функции настраиваются в секции Folders (рисунок 29):

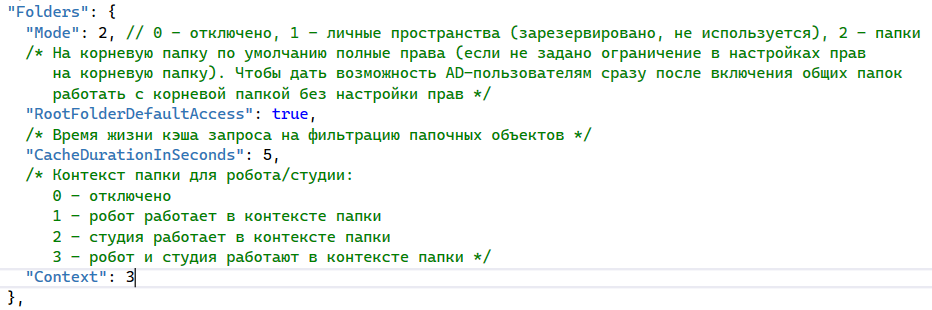


Рисунок – Включение/отключение общих папок

Для отключения использовать «Mode: 0» (по умолчанию выключено).

Параметр «RootFolderDefaultAccess: true» означает, что коневая папка будет доступна всем пользователям, в том числе AD-пользователям, без ограничения прав, если права на корневую папку для пользователя не настроены явно.

## Блокировка робота агентом

Оркестратор опрашивает роботов с некоторой периодичность на предмет их состояния. Такая схема может приводить к накладкам при запуске одного и того же робота при его интенсивном использовании, когда робот не успевает ответить на запрос состояния, или находится в процессе закрытия. Чтобы это исключить, используется блокировка робота Агентом.

Суть её сводится к дополнительному запросу Оркестратора к Агенту на установку блокировки робота на основе файла. Файл блокировки после завершения работы робот удаляет за собой. Если файл блокировки при установке блокировки не удален, или имеется активный процесс робота, блокировка будет отклонена.

Блокировка робота Агентом работает с определенной версии дистрибутива робота, в которой такая функция реализована. На более ранние версии это не распространяется.

Папка, в которую записываются файлы блокировки Роботов на машине робота, настраивается в конфигурационном файле Агента (рисунок 30):

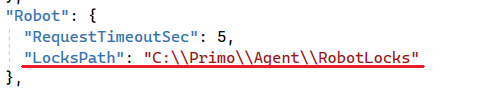


Рисунок – Папка, в которую записываются файлы блокировки Роботов на машине робота

RDP-пользователям (локальным или доменным) необходимо выдать полные права на эту папку (рисунок 31):

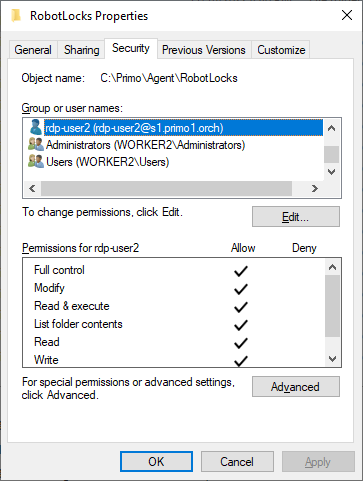


Рисунок – Права на папку с файлами блокировки Роботов Агентом

## Настройка очереди проектов

По умолчанию используется (рекомендуемый вариант) очередь проектов на основе RabbitMQ. Такая очередь работает по принципу:

1. Для выполнения проекта Оркестратор ставит проект в очередь проектов.
2. При выходе проекта из очереди Оркестратор ищет свободного подходящего робота:
   1. Если находит – робот запускается с проектом.
   2. Если не находит – проект отправляется в зависимости от своего приоритета (1 – 3) в одну из очередей ожидания. По истечении времени ожидания (несколько секунд) проект выходит из очереди ожидания и снова отправляется в очередь проектов   
      (см. п.1).

Таким образом, очередь представляет циклический буфер, который по мере добавления в очередь новых проектов и отсутствии роботов будет увеличиваться.

Настройка времени задержки в очереди ожидания осуществляется в секции RabbitMQ (рисунок 32):

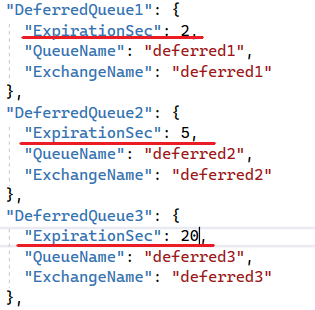


Рисунок – Настройка очередей ожидания

Чем выше эти значения, тем дольше проект будет находиться в очереди ожидания, и выше вероятность, что пока проект будет там находиться, другой проект успеет занять освободившегося робота, изменив, возможно, ожидаемый порядок выполнения проектов.

Чем меньше эти значения, тем выше нагрузка на систему – будет происходить постоянный поиск подходящего робота, хотя, это будет, возможно, заведомо не нужно для таких временных промежутков, если проекты выполняются достаточно долго.

Такое поведение очереди можно изменить на очередь проектов с гарантированным порядком запуска проектов (проектам должны быть назначены приоритеты) в секции RobotStart (рисунок 33):

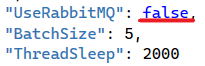


Рисунок – Переключение на очередь с гарантированным порядком

В этом случае сначала будут обрабатываться все элементы в очереди, которые **уже** не получили свободных роботов. И только потом новые элементы. Количество новых элементов, обрабатываемых за одну итерацию опроса очереди, задается параметром BatchSize. Время (мсек) между итерациями опроса задается параметром ThreadSleep.

Параметр ProjectQueue:CleaningDepth позволяет настроить периодичность автоматического удаления из очереди проектов записей, старше (дней) чем текущее время.

## Таймаут, после которого робот считается в состоянии «Не доступен»

Таймаут (секунд), после которого робот считается в состоянии «Не доступен», задается в секции «Robot» в параметре RunStatusTimeout (рисунок 34):

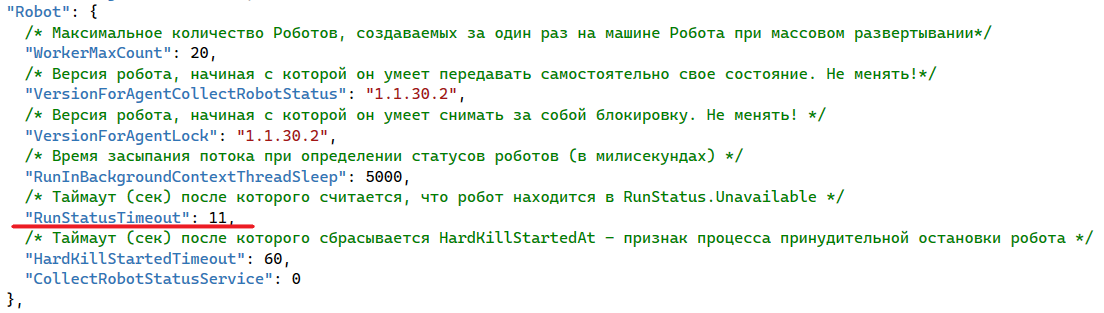


Рисунок – Таймаут, после которого робот считается в состоянии «Не доступен»

Остальные параметры в этой секции менять запрещено без согласования с вендором.

## Открытие/отключение/разлогинивание RDP-сессий

Параметры открытия/отключения/разлогинивания RDP-сессий находятся в секции RDP (рисунок 35):



Рисунок – Параметры открытия/отключения/разлогинивания RDP-сессий

В таблице 12 приведено описание параметров:

Таблица 12 – Описание параметров открытия/отключения/разлогинивания RDP-сессии

| №  п/п | Наименование параметра | Назначение | Примечание |
| --- | --- | --- | --- |
|  | SessionsReleaseThreadSleep | Время засыпания потока отключения RDP-сессий (мсек) |  |
|  | EnabledLogOffDisconnectedRDPUsers | Разлогинивать отключенных RDP-пользователей |  |
|  | LogOffDisconnectedRDPUsersThreadSleep | Время засыпания потока разлогинивания отключенных RDP-пользователей (мсек) |  |
|  | LostConnectTimeout | Время (сек) с последней даты обновления состояния, после которого соединение считается потерянным |  |
|  | LogOffDisconnectedRDPUsersAsync | Асинхронное разлогинивание RDP-пользователей |  |
|  | MaxAttemptStartLogOff | Максимальное кол-во попыток отправки команды разлогона в Агент | Если попытки исчерпаны, оркестратор не будет пытаться сделать разлогон, сессия останется висеть отключенной |
|  | MaxAttemptLogOff | Максимальное кол-во попыток разлогона. На основе ответа от Агента |
|  | SessionsReleaseDelay | Задержка релиза сессии (мсек). Сессия может очень быстро понадобиться другому роботу, чтобы её не пересоздавать заново | Только для ExclusiveSessionsRelease = false |
|  | ExclusiveSessionsRelease | Если поднят этот флаг, робот, когда освобождает сессию, не смотрит на отсутствие релиза сессии у других роботов. Иначе сессия не будет отключена, если хотя бы один робот не сообщил об этом, или его проект не требует закрытия сессии | Должно использоваться, когда только один робот закрывает сессию, чтобы не ломать работу остальных роботов в этой сессии |
|  | RDPSessionsReleaseService | Системные параметры, в сл. версиях будут удалены | |
|  | LogOffDisconnectedRDPUsersService |

Если RDP-лицензий ограниченно, нет запаса, а роботы в одной сессии не запускаются сразу друг за другом (например, по триггеру завершения), рекомендуется SessionsReleaseDelay=0.

Если при условии ограничения на RDP-лицензии все же роботы в одной сессии запускаются сразу друг за другом, рекомендуется ExclusiveSessionsRelease=true.

## Настройка папки для дампа секций журналов Роботов и Оркестратора[[44]](#footnote-44)

Папка с дампами секций журналов Роботов и Оркестратора задается в секции LogsDump (рисунок 36):

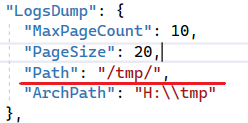


Рисунок – Папка с дампами секций журналов Роботов и Оркестратора

Эта папка должна находиться на сервере БД.

Создавать дампы можно через UI оркестратора для управления секциями журналов (рисунок 37):

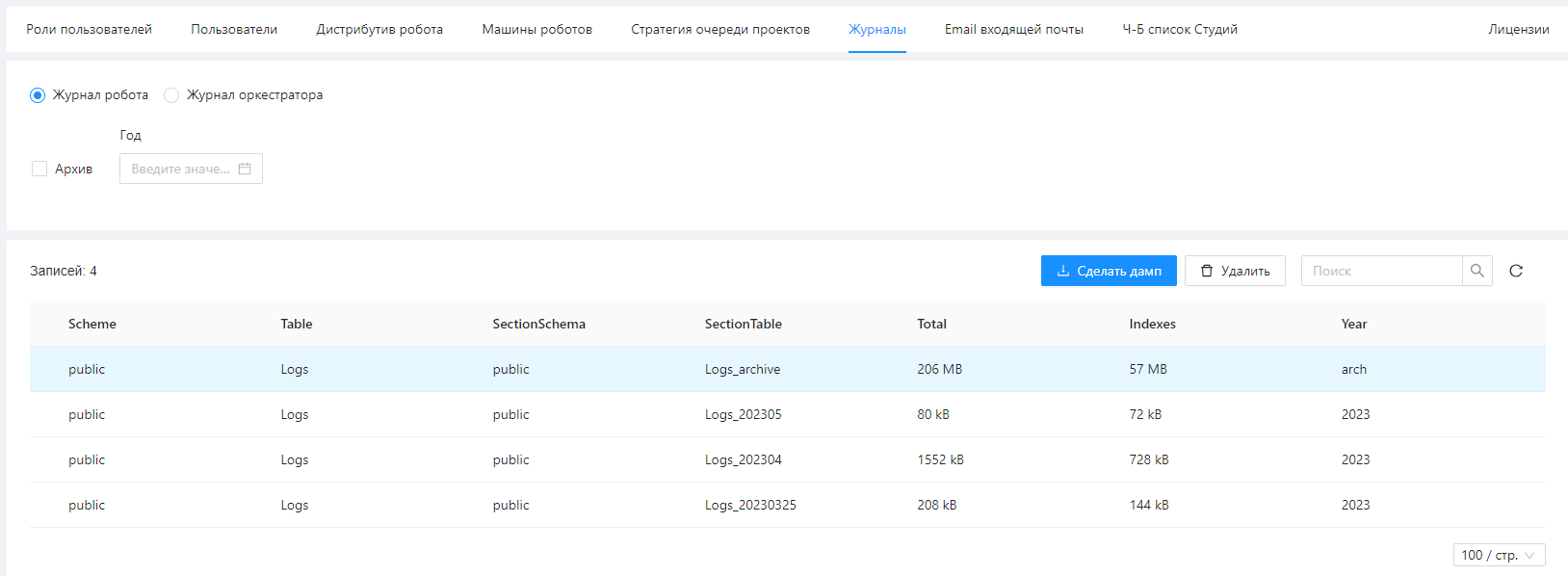


Рисунок – UI оркестратора для управления секциями журналов

Удалять секции можно там же, чтобы чистить старые данные. Например, сначала сделать дамп секции, и потом её удалить. Сформированные дампы не доступны через UI Оркестратора.

UI оркестратора для управления секциями журналов предназначено только для использования администратором.

## Настройка стратегии очереди проектов для тенанта

Глобальная (она же для дефолтного тенанта) стратегия очереди проектов (рисунок 38) может быть заменена в конфигурационном файле в секции настройки тенанта (

рисунок 39):

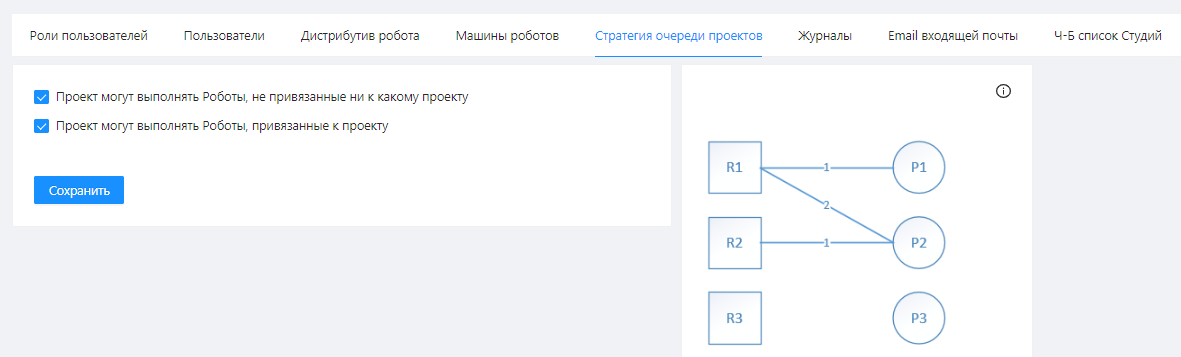


Рисунок – UI оркестратора для настройки стратегии очереди проектов (глобальной)

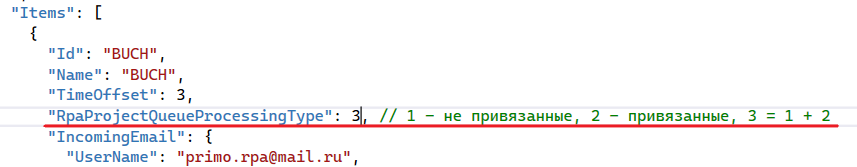


Рисунок – Настройка стратегии очереди проектов для тенанта

Если для тенанта не задана настройка стратегии очереди проектов, то используется глобальная (для дефолтного тенанта).

## Настройка трансляции RDP-сессии

Параметры трансляции RDP-сессий находятся в секции RDP (рисунок 40):



Рисунок – Параметры трансляции RDP-сессий

Таблица 13 – Описание параметров трансляции RDP-сессии

| №  п/п | Наименование параметра | Назначение | Примечание |
| --- | --- | --- | --- |
|  | RdpBaseUrl | URL (конечная точка nginx) одной службы RDP2 – для случая, когда не задан AddressFilter в конфиге службы RDP2 | Начинается с внешнего Url Оркестратора. Например, тут это https://192.168.0.19:44392 |
|  | RdpBaseUrls | Сопоставление URL (конечной точки nginx) служб RDP2 – для случая, когда задан AddressFilter в конфигах служб RDP2. Ключом является внутренний IP сервера с RDP2, значением – внешний URL, который проксируется в nginx во внутренний IP. Внешний Url склеивается из базового внешнего Url оркестратора и идентификатора для проксирования. | В конфиге службы RDP2 должен быть прописан IP сервера с RDP2 (параметр Host). Если он не будет прописан, то определится (возможно, не правильно) из http-запроса RDP2 к Оркестратору |

Параметры RdpBaseUrl и RdpBaseUrls должны соотноситься с настройкой nginx и параметром Host конфига каждого экземпляра (на отдельной машине) службы RDP2:

Вариант 1 – Используется один экземпляр RDP2 для всех машин роботов (рисунок 41):



Рисунок – Конфигурация nginx для одного экземпляра RDP2

Если используется IIS, настройка осуществляется аналогично в Web.config узла Primo.UI (рисунок 42)[[45]](#footnote-45):

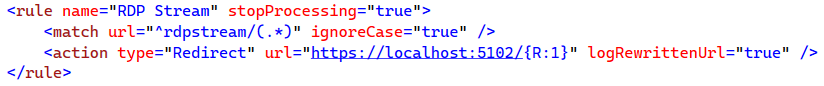


Рисунок – Конфигурация IIS для одного экземпляра RDP2

В конфиге WebApi используется только параметр RdpBaseUrl. Параметр RdpBaseUrls не используется. В конфиге RDP2 параметр Host можно не задавать.

Вариант 2 – Используется несколько экземпляров RDP2, множество машин роботов поделено (без пересечения!!!) между экземплярами за счет параметра AddressFilter (рисунок 43):



Рисунок – Конфигурация для нескольких экземпляра RDP2

В конфиге WebApi используется только параметр RdpBaseUrls. Параметр RdpBaseUrl не используется. В конфиге RDP2 параметр Host (внутренний IP сервера, рисунок 44) желательно задать, так как определение IP машины с RDP2 из http-запроса может работать не правильно (зависит от настройки сети).

Для IIS настраивается аналогично в Web.config узла Primo.UI.



Рисунок – Параметр Host конфига службы RDP2 с внутренним IP сервера

В секции EnabledOrigins конфига каждого экземпляра службы RDP2 должен быть прописан внешний Url Оркестратора (рисунок 45)!!!

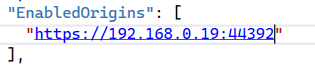


Рисунок – Секция EnabledOrigins конфига службы RDP2 с внешним Url Оркестратора

## Настройка параметров очереди обмена данными

Описание параметров настройки очереди обмена данными (секция ExchangeQueue) приведено в таблицах 14 и 15:

Таблица 14 – Параметры сбора статистики

| №  п/п | Наименование параметра | Назначение | Примечание |
| --- | --- | --- | --- |
|  | StatisticAvgWindow | Ширина окна сбора средних (сек) |  |
|  | StatisticAvgWindowPreCount | Запас создания окон | Внутренний системный параметр для оптимизации. Не рекомендуется менять |
|  | StatisticAvgLastWindowsCount | Количество последних окон, которое отдается в запросе |  |
|  | StatisticRecalcInterval | Интервал (мсек) пересчета статистики |  |
|  | StatisticBatchSize | Размер пачки, вычитываемой за раз из очереди событий статистики |  |

События, связанные со статистикой очередей обмена данными (круговая диаграмма в таблице с очередями), регистрируются в реальном времени во внутренней очереди Оркестратора. Обработка этой очереди происходит с периодичностью StatisticRecalcInterval. За одну итерацию из очереди считывается пачка событий размером StatisticBatchSize, на основе которой пересчитывается статистика.

Среднее время обработки элемента очереди (разность дат события прочтения элемента и последнего события[[46]](#footnote-46) завершения обработки элемента) считается по окну StatisticAvgWindow, например, раз в 2 минуты. Окна создаются с запасом StatisticAvgWindowPreCount при заполнении очередного множества окон. Запрос для построения графика в UI отдаёт последние заполненные окна в размере StatisticAvgLastWindowsCount.

Таблица 15 – Параметры оптимизации чтения из очереди по FIFO

| №  п/п | Наименование параметра | Назначение | Примечание |
| --- | --- | --- | --- |
|  | FIFORetry | Количество ретраев при извлечении по FIFO. Если задан в настройке очереди, аналогичный параметр берется из БД для очереди |  |
|  | PrefetchDepth | Глубина предварительной выборки в PrefetchBuffer. Не рекомендуется выше 500. Если 0, PrefetchBuffer не используется (рисунок 46) | Предварительная выборка используется в скоростном режиме. PrefetchBuffer – временный буфер, в который вычитывается сразу несколько элементов очереди из БД, и последующие обращения идут в этот буфер |
|  | PrefetchBufferResetInterval | Интервал (мсек) сброса PrefetchBuffer | Происходит синхронизация вычитанных из PrefetchBuffer элементов с БД |
|  | SpeedModeInterval | Интервал (мсек) между 2-мя последовательными запросами к чтению из очереди по FIFO, определяющий переключение в скоростной режим | PrefetchBuffer используется в скоростном режиме |



Рисунок – Иллюстрация работы PrefetchBuffer

В момент запроса t1 очередь еще работает в обычном режиме, запрос обслуживается напрямую в БД.

В момент запроса t2, так как t2 – t1 < SpeedModeInterval, очередь переходит в скоростной режим, и формируется PrefetchBuffer. Запрос обслуживается этим буфером. Операций в этот момент происходит больше, так как требуется сформировать PrefetchBuffer, но они соизмеримы по времени с обслуживанием запроса t1.

В моменты t3 и t4 очередь уже работает в скоростном режиме и запросы обслуживаются сразу из PrefetchBuffer.

Синхронизация прочитанных в PrefetchBuffer элементов с БД происходит в фоне с периодичностью PrefetchBufferResetInterval. Эта синхронизация выполняется массово (не одиночные запросы к БД) через внутреннюю очередь.

Условие проверки уникальности натурального ключа (глобального или локального) элемента очереди после его удаления задается булевыми параметрами DeletedNaturalKeysIsGlobalUniqueRequired и DeletedNaturalKeysIsUniqueRequired. Если true – удаленный элемент не участвует в проверке уникальности.

## Настройка параметров оповещения

Настройка параметров оповещения производится в файле appsettings.json службы States (таблица 16):

Таблица 16 – Параметры оповещения

| №  п/п | Тип оповещения | Секция конфигурационного файла | Параметры |
| --- | --- | --- | --- |
|  | Об истечении срока действия пароля | UserExpired | NotificationInterval - Интервал оповещения (минут) - через какое время будет повторная отправка. По умолчанию 1440.  StartBefore - За какое время (минут)до истечения срока начнется оповещение. По умолчанию 1440. |
|  | Об истечении срока действия лицензии | LicenseExpired | NotificationInterval - Интервал оповещения (часов) - через какое время будет повторная отправка. По умолчанию 24.  DaysStartBefore - За какое время (дней) до истечения срока начнется оповещение. По умолчанию 15. |

## Настройка очистки старых запусков

Очистка старых завершенных запусков осуществляется (если включена) как удаление запусков из оперативной БД ltools и перенос их вместе со значениями аргументов проекта на запуске в БД с логами ltoolslogs. Завершенным считается запуск, у которого зафиксировано одно из: завершение работы робота, принудительное завершение работы робота, освобождение робота по таймауту, ошибка при запуске.

Описание параметров настройки очистки старых запусков (секция RpaProjectLaunch) приведено в таблице 17:

Таблица 17 – Параметров настройки очистки старых запусков

| №  п/п | Наименование параметра | Назначение | Примечание |
| --- | --- | --- | --- |
|  | LogsThreadSleep | Время (сек) засыпания потока сброса завершенных старых запусков в БД с логами. Рекомендуемое значение 300  (5 минут) | Чем чаще будет просыпаться поток, тем более оперативно старые завершенные запуски будут переноситься в БД с логами, меньшими пачками. Но, сильно частая работа потока может создать повышенную нагрузку на сервер |
|  | MaxBatch | Максимальный размер пачки за один раз передаваемый в БД с логами.  Рекомендуемое значение 500 | Чем больше размер пачки, тем быстрее (в сочетании с параметром LogsThreadSleep) старые запуски будут переноситься в БД с логами. Но, сильно большие пачки могут создать повышенную нагрузку на сервер |
|  | Period | Время (мин[[47]](#footnote-47)) после которого завершенный запуск считается старым.  Рекомендуемое значение примерно 80640 (4 недели) | Рекомендуется выбирать в зависимости от интенсивности запусков: чем чаще запуски, тем меньшее значение может использоваться, чтобы в оперативной БД не копилось слишком много запусков |
|  | EnableLogsTransfer | Разрешен автоматический перенос старых запусков в БД с логами.  Не рекомендуется выключать, так как ведет к росту размера оперативной БД |  |

Пример расчета значений параметров настройки очистки старых запусков:

Частота запусков – 1000 в сутки. Целевой объем запусков в оперативной БД – за последние 2 недели.

Чтобы обеспечить постоянный объем оперативной БД частота переноса запусков в БД с логами должна равняться частоте запусков, то есть 1000 в сутки. Period = 20160 обеспечит актуальности 2 недели. Чтобы переносить ежесуточно 1000 записей о запусках, не нагружая пиками серверы и одновременно не нагружая их холостой работой, выберем MaxBatch = 100. Это значит поток переноса должен проснуться не менее 10 раз в сутки, то есть   
LogsThreadSleep = 135 \* 60 = 8100.

Рекомендуемые значения параметров в таблице 17 выбраны исходя из примерно 10-ти кратного запаса.

## Ограничение версии Студии

Для снижения риска при использовании устаревших версий Студии получать ошибки из-за несовместимости версий Студии и Оркестратора имеется возможность ограничить минимальную версию Студии.

Описание параметров ограничения версии Студии (секция StudioVersionChecker) приведено в таблице 18:

Таблица 18 – Параметры ограничения версии Студии

| №  п/п | Наименование параметра | Назначение | Примечание |
| --- | --- | --- | --- |
|  | MinimumStudioVersion | Минимальная версия Студии (строка). Если не задана/пустая строка – ограничение версии Студии не применяется. Формат версии 1.23.8.4 |  |
|  | StudioVersionExceptionBehaviorType | Тип поведения Студии, если версия не проходит проверку (передаем в Студию при её попытке получить лицензию у Оркестратора):   * 0 – Закрываем Студию без вывода пользователю какого-либо сообщения * 1 – Закрываем Студию, перед этим выводя пользователю сообщение об ошибке * 2 – Студию не закрываем, только выводим пользователю сообщение об ошибке |  |

## Ограничение потока событий от триггеров

Для снижения/устранения риска отказа в обслуживании при обработке потока событий от триггеров имеется возможность ограничить исходный поток событий – преобразовать его при помощи промежуточной очереди в поток с меньшей и предсказуемой интенсивностью (при условии не превышения интегральной характеристики интенсивности исходного потока, см. ниже).

Описание параметров ограничения потока событий от триггеров (секция Assignment) приведено в таблице 19:

Таблица 19 – Параметры ограничения потока событий от триггеров

| №  п/п | Наименование параметра | Назначение | Примечание |
| --- | --- | --- | --- |
|  | ChainLimiterTriggerEventTime | Использовать ограничение для потока событий триггеров (мсек).  Если 0 – ограничение не используется. Также ограничение не используется, если все одновременно параметры ChainLimiterTriggerEventEmail, ChainLimiterTriggerEventSchedule и ChainLimiterTriggerEventFromRobot имеют значение false.  Рекомендуемое значение 500 |  |
|  | MaxTriggerQueueLength | Максимальный размер промежуточной очереди, при превышении которого события будут теряться – фактически означает отказ в обслуживании |  |
|  | ChainLimiterTriggerEventEmail | Использовать ограничение для потока событий триггеров почты |  |
|  | ChainLimiterTriggerEventSchedule | Использовать ограничение для потока событий триггеров расписаний |  |
|  | ChainLimiterTriggerEventFromRobot | Использовать ограничение для потока событий триггеров запуска из роботов |  |
|  | MaxLastEventsInterval | Максимальный интервал (минут) для отображения последних событий уменьшения/увеличения/переполнения внутренней очереди на главной странице |  |

Рассмотрим подробнее, как работает данное ограничение, и как стоит выбирать его параметры. Пусть на отрезке времени [t0, t9] (рисунок 47, a) на каждом Δt[[48]](#footnote-48) интенсивность потока событий I = Imax. Будем считать интегральной характеристикой потока событий S сумму всех прямоугольников с основанием Δt и высотой I. В частности, рисунок 47 a) показывает, что система работает на пределе своих возможностей (Imax, Smax), но, с нагрузкой справляется, отказа в обслуживании[[49]](#footnote-49) не происходит.



Рисунок – Интегральная характеристика потока событий

Рисунок 47 б) показывает, что на отрезке [t3, t6] будет отказ в обслуживании, так как превышено Imax на этом отрезке, и, как следствие, превышена интегральная характеристика интенсивности потока (так как на остальных Δt I = Imax).

Рисунок 47 в) также показывает, что на отрезке [t3, t6] будет отказ в обслуживании, так как превышено Imax на этом отрезке. Но, максимальная интегральная характеристикаинтенсивности потока не превышена. В этом случае возможно перераспределение нагрузки.

Если система принимает и обрабатывает поток событий по простой схеме (рисунок 48, а), перераспределение нагрузки не используется. Это соответствует значению параметра ChainLimiterTriggerEventTime = 0. Даже без превышения интегральной характеристики интенсивности потока, на пиках система не справится, будет отказ в обслуживании. Контролировать интенсивность для каждого отдельно взятого Δt, которая считается внешним фактором[[50]](#footnote-50), система принципиально не может.



Рисунок – Схемы распределения нагрузки по принятию и обработке потока событий триггеров

При включенном распределении нагрузки (рисунок 48, б), соответствует значению параметра ChainLimiterTriggerEventTime > 0, пики нагрузки примет промежуточная очередь. Преобразованный поток событий по своему построению[[51]](#footnote-51) на любом Δt будет иметь I ≤ Imax. Таким образом, очередь сгладит пики, и весь поток, возможно, с некоторой задержкой, будет обработан.

Если же максимальная интегральная характеристика интенсивности исходного потока превышена, схема с промежуточной очередью не даст результата – очередь переполнится, события будут теряться[[52]](#footnote-52), что также равносильно отказу в обслуживании.

Для числовой оценки интегральной характеристики интенсивности потока можно использовать лог событий оркестратора – таблица OrchEvent, одно из трех значений поля Event:

* AssignmentIncreaseTriggerNativeEventBus = 9016 – внутренняя очередь увеличивается;
* AssignmentDecreaseTriggerNativeEventBus = 9017 – внутренняя очередь уменьшается;
* AssignmentOverflowTriggerNativeEventBus = 9018 – внутренняя очередь переполнена;

и поле EntityId (для событий увеличения/уменьшения внутренней очереди). Эти события могут быть записаны не чаще, чем с интервалом ChainLimiterTriggerEventTime, так как их наступление проверяется при каждом просыпании потока сбора событий из промежуточной очереди.

Поле EntityId показывает, как меняется процент от MaxTriggerQueueLength (фиксированные значения: 1 – 25%, 2 – 50%, 3 – 75%[[53]](#footnote-53)) использования промежуточной очереди. Событие наступает при переходе вверх/вниз одного из 3-х переходов: 25%, 50%, 75%. Так как промежуточная очередь принимает на себя исходный поток событий, по динамике процента её использования во времени можно оценить интегральную характеристику интенсивности этого потока.

На основе этой оценки можно выбрать приемлемые значения параметров ChainLimiterTriggerEventTime и MaxTriggerQueueLength. А также выработать рекомендации по снижению интенсивности исходного потока событий за счет уменьшения частоты срабатывания триггеров (перенастройка параметров заданий).

Рассмотрим пример оценки интегральной характеристики интенсивности исходного потока на основе лога событий Оркестратора (таблица 20)[[54]](#footnote-54). Пусть система работает с параметрами ChainLimiterTriggerEventTime=500 и MaxTriggerQueueLength=20[[55]](#footnote-55). В логе не зафиксировано событие переполнения внутренней очереди – иначе это бы означало отказ в обслуживании, и детальный анализ лога на предмет интенсивности тут уже был не нужен.

Анализ проведем на основе временного окна в 20 сек – выберем события, которые попали в это окно. Пусть это будут 4 события[[56]](#footnote-56):

SELECT "OrchTimestampUtc",

"Event",

"EntityId"

FROM "OrchEvents"

WHERE "Event" in (9016, 9017, 9018)

AND "OrchTimestampUtc" BETWEEN '2023-11-10 11:00:00' AND '2023-11-10 12:00:00'

AND "NodeId" = 0

ORDER BY "OrchTimestampUtc" ASC;

Таблица 20 – Лог событий увеличения/уменьшения процента использования внутренней очереди событий от триггеров

| №  п/п | OrchTimestampUtc | Event[[57]](#footnote-57) | EntityId[[58]](#footnote-58) |
| --- | --- | --- | --- |
|  | 00:00:00:000 | ↑ | 25% |
|  | 00:00:04:000 | ↓ | 25% |
|  | 00:00:06:000 | ↑ | 25% |
|  | 00:00:10:000 | ↓ | 25% |

Так как Δt = ChainLimiterTriggerEventTime = 500, тогда таблица 20 преобразуется в таблицу 21 (нумерация строк с 0) – интенсивность исходного потока событий:

Таблица 21 – Интенсивность исходного потока событий от триггеров

| №  п/п | OrchTimestampUtc1 | Event | EntityId | Si |
| --- | --- | --- | --- | --- |
|  | 00:00:00:000 | ↑ | 25% | > 5 |
|  | 00:00:00:500 |  |  |  |
|  | 00:00:01:000 |  |  |  |
|  | 00:00:01:500 |  |  |  |
|  | 00:00:02:000 |  |  |  |
|  | 00:00:02:500 |  |  |  |
|  | 00:00:03:000 |  |  |  |
|  | 00:00:03:500 |  |  |  |
|  | 00:00:04:000 | ↓ | 25% | < 5 |
|  | 00:00:04:500 |  |  |  |
|  | 00:00:05:000 |  |  |  |
|  | 00:00:05:500 |  |  |  |
|  | 00:00:06:000 | ↑ | 25% | > 5 |
|  | 00:00:06:500 |  |  |  |
|  | 00:00:07:000 |  |  |  |
|  | 00:00:07:500 |  |  |  |
|  | 00:00:08:000 |  |  |  |
|  | 00:00:08:500 |  |  |  |
|  | 00:00:09:000 |  |  |  |
|  | 00:00:09:500 |  |  |  |
|  | 00:00:10:000 | ↓ | 25% | < 5 |

Очередь разбирается с интенсивностью 1 сообщение в 500 мсек. За одно обращение из очереди читается не более одного элемента. По таблице 21 видно, что:

После 1-го роста очереди на 25% (№п/п 0 – 7), что значит более 5 (но менее 10) элементов в очереди, до следующего уменьшения из очереди читается 8 элементов. Так как событие уменьшения очереди зафиксировано после чтения 8 элементов, значит очередь в этот промежуток принимала события.

После 1-го уменьшения очереди на 25% (№п/п 8 – 11), что значит менее 5 элементов в очереди, до следующего роста из очереди читается 4 элемента. Те элементы, которые были в очереди на момент фиксации её уменьшения на этом промежутке вычитаны полностью. Но, так сразу следом идет увеличение очереди, это опять говорит о том, что очередь в этот промежуток принимала события.

После 2-го роста очереди на 25% (№п/п 12 – 19), что значит более 5 элементов в очереди, до следующего уменьшения из очереди читается 8 элементов. Так как событие уменьшения очереди зафиксировано после чтения 8 элементов, значит очередь и в этот промежуток принимала события.

После 2-го уменьшения очереди на 25% (№п/п 20) таблица заканчивается, но, очередь в этот момент не пустая.

Таким образом, имеем 2 25%-ных пика интенсивности, которые приняла на себя очередь. В целом интенсивность исходного потока не является для системы критической, но обработка событий идет с небольшой (примерно 5 элементов) задержкой. В данном случае, если у системы есть запас мощности, можно уменьшить ChainLimiterTriggerEventTime, и события будут извлекаться на обработку чаще, внутренняя очередь будет расти медленнее, возможно, её рост даже не будет зафиксирован в логе. Увеличивать MaxTriggerQueueLength не имеет смысла, так как пики не высокие – 25%.

## Автоматическое временное замедление очереди проектов при пиках нагрузки на машинах роботов

Для снижения/устранения риска отказа в обслуживании при запуске робота на машине робота с проектом, который вышел из очереди проектов, по причине перегруженности машины роботов, можно использовать автоматическое замедление очереди проектов – проект продолжит ждать в очереди проектов, пока нагрузка на машине робота не спадет до приемлемой.

Описание параметров, влияющих на автоматическое временное замедление очереди проектов при пиках нагрузки на машинах роботов (секции Worker и ProjectQueue), приведено в таблицах 22 и 23:

Таблица 22 – Параметры сбора метрик производительности с машин роботов (секция Worker)

| №  п/п | Наименование параметра | Назначение | Примечание |
| --- | --- | --- | --- |
|  | CollectBackgroundPerformance | Собирать метрики производительности машин в фоне, одновременно с проверкой их доступности. Если включена, может использоваться автоматическое временное замедление очереди проектов при пиках нагрузки на машинах роботов.  Только для  CollectWorkerStatusService = 1 |  |
|  | PerformanceDeltaPrcnt | Шаг разбиения шкалы 0 - 100 на интервалы, например, (0, 25, 50, 75, 100) |  |
|  | PerformanceDelta100Prcnt | Величина (абсолютное значение процента), на которую метрики должны отличаться от 100%, чтобы был зафиксирован факт перегрузки машины |  |

Таблица 23 – Параметры автоматического временного замедления очереди проектов при пиках нагрузки на машинах роботов (секция ProjectQueue)

| №  п/п | Наименование параметра | Назначение | Примечание |
| --- | --- | --- | --- |
|  | WorkerPerformanceHighDuration | Продолжительность (сек) нахождения метрик машины робота в диапазоне высокой нагрузки, при которой проект должен снова уйти в очередь проектов.  Если равно 0 – замедление очереди не используется | Если равно 0, метрики производительности с машин роботов все равно собираются, если этот сбор включен |
|  | WorkerPerformanceHighCPUOnly | Для принятия решения о том, что машина робота перегружена, использовать только CPU% |  |
|  | WorkerPerformanceHighResolver | Тип определения высокой нагрузки:   * 0 – Фиксированное значение, например, 95%. Не связано с Worker.PerformanceDeltaPrcnt * 1 – 100% – Worker.PerformanceDeltaPrcnt | Если шкала разбивается на большие интервалы (см. таблица 22, №п/п 2) или не кратные 100, то попадание нагрузки в последний интервал может быть грубой оценкой или не оправдано завышенной. Тогда можно эту оценку указать явно в AbsoluteWorkerPerformanceHigh и использовать тип 1. |
|  | AbsoluteWorkerPerformanceHigh | Абсолютное значение нагрузки, которое считается высоким.  Только для  WorkerPerformanceHighResolver = 0 |  |

## Отключение тенанта по умолчанию и блокировка встроенных не технических учетных записей

Для удовлетворения возможных требований в части информационной безопасности Заказчика имеется возможность отключить тенант по умолчанию (параметр Tenants.DisabledDefaultTenant: true в конфигурационном файле WebApi, рисунок 49):

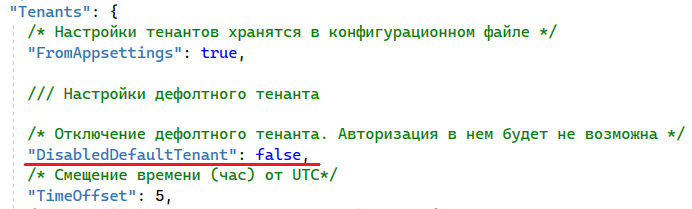


Рисунок – Отключение тенанта по умолчанию

Отключение тенанта по умолчанию означает не возможность авторизации в этом тенанте.

Для блокировки встроенных не технических учетных записей (studio, admin и superadmin) в БД ltoolsidentity нужно выполнить запрос:

UPDATE "Users"

SET "Disabled" = True[[59]](#footnote-59)

WHERE "Id" IN ('a18be9c0-aa67-4af8-bd17-00bd9344e575',

'a18be9c0-aa65-4af8-bd17-00bd9344e575',

'00000000-0000-0000-0000-100000000000');

После отключении тенанта по умолчанию все-равно остается возможность зайти под специально настроенным AD-пользователем со встроенной ролью Administrator в этот тенант для выполнения системных настроек Оркестратора (управление дистрибутивами робота, лицензиями и т.д.). Для этого необходимо выполнить следующие шаги (таблица 24):

Таблица 24 – Шаги по настройке специального AD-пользователя для выполнения системных настроек Оркестратора

| №  п/п | Шаг | Примечание |
| --- | --- | --- |
|  | Разрешить использование тенанта по умолчанию  Tenants.DisabledDefaultTenant: false | Требуется перезагрузка службы WebApi |
|  | Разблокировать, если заблокирована, встроенную учетную запись admin ('a18be9c0-aa65-4af8-bd17-00bd9344e575'), см. запрос выше |  |
|  | Авторизоваться под admin |  |
|  | В AD создать группу для специального пользователя, например primo-serving |  |
|  | Связать группу primo-serving с ролью Administrator | Далее любой AD-пользователь, который входит в группу primo-serving, будет являться специальным |
|  | Выйти из системы |  |
|  | Заблокировать встроенную учетную запись admin |  |
|  | Запретить использование тенанта по умолчанию  Tenants.DisabledDefaultTenant: true | Требуется перезагрузка службы WebApi |

## Мерцающие RDP-сессий (виджет на главной странице)

Параметры определения мерцающих RDP-сессий (при наложение учетных записей RDP-пользователей или ограничении количества RDP-пользователей на машине робота) находятся в секции RDP (рисунок 50):



Рисунок – Параметры определения мерцающих RDP-сессий

В таблице 25 приведено описание параметров:

Таблица 25 – Описание параметров определения мерцающих RDP-сессий

| №  п/п | Наименование параметра | Назначение | Примечание |
| --- | --- | --- | --- |
|  | MaxConnectedFlickersCount | Максимальное значение счетчика постоянных переключений, после которого устанавливается признак, что RDP-сессия в состоянии мерцания (значение по умолчанию – 3) |  |
|  | ConnectedFlickersInterval | Интервал (мсек) переключения состояния RDP-сессии, который является кандидатом на мерцание (значение по умолчанию – 10000) |  |
|  | ResetConnectedFlickersIntervalFactor | Коэффициент умножения ConnectedFlickersInterval. Если в течение полученного интервала не происходило переключения, сбрасывается счетчик переключений и флаг наличия мерцания (значение по умолчанию – 3) |  |

## Проблемы с кодировкой при использовании кириллицы в названиях файлов ZIP-архива проекта[[60]](#footnote-60)

При возникновении проблем с кодировкой при использовании кириллицы в названиях файлов ZIP-архива проекта кодировку нужно в конфигурационном файле Агента на машине робота задать принудительно (рисунок 51):

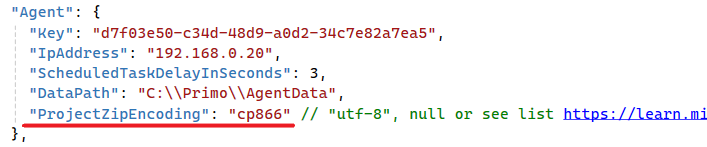


Рисунок – Параметр ProjectZipEncoding для принудительного задания кодировки ZIP-архива

При ProjectZipEncoding = null кодировка определится автоматически на основе окружения.

## Интеграция с CyberArk

Интеграция с CyberArk (рисунок 52) используется для возможности хранить логин/пароль в ассетах типа Credentials отдельно от Оркестратора в CyberArk, настраивается в конфигурационном файле WebApi:



Рисунок – Параметры подключения к API CyberArk

LogonType = 0 зарезервировано, не меняется. BaseUrl – url CyberArk. UserName/Password – логин пароль для авторизации в API CyberArk (должны предоставляться администратором CyberArk). Password должен быть зашифрован утилитой для шифрования паролей LTools.Orchestrator.PasswordEncryptor.

# Схема взаимодействия Оркестратора и Робота

Свои запросы Оркестратор напрямую отправляет только своим Агентам на машинах робота (рисунок 53):



Рисунок – Сетевое взаимодействие Оркестратора с Роботами

От своих Агентов и запущенных Роботов Оркестратор получает:

1. От Агентов – обратную связь и запросы на скачивание проектов/дистрибутивов.
2. От запущенных Роботов:
   1. Запросы на получение лицензии.
   2. Телеметрию.
   3. Логи.
   4. Запросы для работы с очередями и ресурсами, мягкий останов.

Работа по запуску и контролю работы Windows-робота выполняется опосредованно (рисунок 54):

1. Оркестратор получает сигнал «Запустить Робота» (ручной запуск или по заданию).
2. Оркестратор открывает RDP-сессию под учетной записью Робота, дожидается загрузки профиля пользователя.
3. Оркестратор делегирует запуск Робота своему Агенту. Агент отправляет в Оркестратор трекинг своей работы по запуску Робота (обратная связь).
4. Агент скачивает из Оркестратора архив RPA-проекта.
5. Агент готовит файл Run.bat, через который робот запускается с параметрами.
6. Агент последним шагом в потоке работы по запуску Робота создает/меняет Windows Task с учетной записью Робота[[61]](#footnote-61).
7. Windows Task запускается по триггеру средствами ОС.
8. Windows Task непосредственно запускает Робота через Run.bat.
9. Запущенный Робот запрашивает в Оркестраторе лицензию.
10. Запущенный робот в процессе своей работы отправляет в Оркестратор телеметрию и логи. Также разные запросы по работе с очередями и ресурсами, мягкий останов.



Рисунок – Работа по запуску и контролю работы Windows-робота



Рисунок – Работа по запуску и контролю работы Linux-робота

Работа по запуску и контролю работы Linux-робота выполняется немного иначе, чем для Windows-робота (без участия RDP2), но, по похожей схеме (рисунок 55):

1. Оркестратор получает сигнал «Запустить Робота» (ручной запуск или по заданию).
2. Оркестратор делегирует запуск Робота своему Агенту. Агент отправляет в Оркестратор трекинг своей работы по запуску Робота (обратная связь).
3. Агент скачивает из Оркестратора архив RPA-проекта.
4. Агент готовит файл robotTask.sh, через который робот запускается с параметрами.
5. Агент открывает виртуальную графическую сессию
6. Агент непосредственно запускает Робота через robotTask.sh.
7. Запущенный Робот запрашивает в Оркестраторе лицензию.
8. Запущенный робот в процессе своей работы отправляет в Оркестратор телеметрию и логи. Также разные запросы по работе с очередями и ресурсами, мягкий останов.



Рисунок – Работа по развертыванию Робота

Работа развертыванию Робота (одинаково для Windows/Linux) выполняется опосредованно (рисунок 56):

1. Оркестратор получает сигнал «Развернуть робота».
2. Оркестратор делегирует развертывание Робота своему Агенту. Агент отправляет в Оркестратор трекинг своей работы по развертыванию Робота (обратная связь).
3. Агент скачивает из Оркестратора дистрибутив Робота.
4. Агент распаковывает дистрибутив робота в папку робота.
5. Агент готовит вносит изменения в конфигурационный файл робота Primo.Robot.exe.config.
6. Агент осуществляет (через вызовы WinApi + cmd) резервирование Url для робота и привязывает к нему SSL-сертификат[[62]](#footnote-62).

Развернутый робот готов к запуску.

1. Службы Active Directory [↑](#footnote-ref-1)
2. Single Sign-On – технология единого входа [↑](#footnote-ref-2)
3. Распределение по машинам серверной части может отличаться в зависимости от комплекта поставки и/или принятых в организации решений по развертыванию Системы. Некоторые сервисы могут быть не показаны на рисунке. [↑](#footnote-ref-3)
4. Оркестратор также поддерживает работу с MS SQL SERVER 2016+ [↑](#footnote-ref-4)
5. Здесь и далее – физическое название БД, используется в конфигурационных файлах [↑](#footnote-ref-5)
6. Служба Windows или IIS [↑](#footnote-ref-6)
7. Nginx или IIS [↑](#footnote-ref-7)
8. Заказчик самостоятельно в соответствии со спецификацией разрабатывает интеграционный шлюз. [↑](#footnote-ref-8)
9. Не входит в комплект поставки, скачивается отдельно по предоставляемому вендором адресу. [↑](#footnote-ref-9)
10. Не входит в комплект поставки [↑](#footnote-ref-10)
11. Чтобы не возникло конфликтов занятых портов, файлов и т.п. [↑](#footnote-ref-11)
12. Полный путь зависит от ОС [↑](#footnote-ref-12)
13. Для Windows [↑](#footnote-ref-13)
14. Для Windows [↑](#footnote-ref-14)
15. Для Windows [↑](#footnote-ref-15)
16. Для Windows. В зависимости от варианта развертывания могут потребоваться навыки   
    администрирования IIS [↑](#footnote-ref-16)
17. В зависимости от СУБД Оркестратора [↑](#footnote-ref-17)
18. Комплект содержит инструкции по секционированию. В том числе для передачи журналов робота в ElasticSearch для последующего долговременного хранения и аналитики по историческим данным   
    (таблица 2, № п/п 47). [↑](#footnote-ref-18)
19. Папка Distr с подпапками Windows и Linux [↑](#footnote-ref-19)
20. Размер именно файла, можно просмотреть в свойствах файла. Размер на диске может на единицу быть больше [↑](#footnote-ref-20)
21. Зависит от конкретной поставки [↑](#footnote-ref-21)
22. В зависимости от поставки может не включать дистрибутив (пакеты или установочные файлы) PostgreSQL [↑](#footnote-ref-22)
23. Папка Docs [↑](#footnote-ref-23)
24. Возможны дополнительные файлы дистрибутивов и документации [↑](#footnote-ref-24)
25. Не рекомендуется использовать, так как у nginx под Windows отсутствует режим работы в качестве службы [↑](#footnote-ref-25)
26. Иначе не получится полноценно использовать pgbouncer. Не рекомендуется использовать PostgreSQL под Windows [↑](#footnote-ref-26)
27. Обобщенные требования ко всем компонентам бэкэнда Системы [↑](#footnote-ref-27)
28. Обобщенное требование. В зависимости от выполняемого проекта/настроек логирования Робот может создавать разную нагрузку на Оркестратор [↑](#footnote-ref-28)
29. Рекомендуется размещать на отдельном линуксовом сервере [↑](#footnote-ref-29)
30. Служба RDP2, идет в комплекте поставки [↑](#footnote-ref-30)
31. Если используется стрим RDP-сессии в UI Оркестратора, рекомендуется не менее 8 ядер [↑](#footnote-ref-31)
32. Без стрима RDP-сессии в UI Оркестратора [↑](#footnote-ref-32)
33. Архив PasswordEncriptor.zip из комплекта поставки [↑](#footnote-ref-33)
34. Или appsettings.ProdLinux.json, если поставка для Linux [↑](#footnote-ref-34)
35. Под Линукс – appsettings.ProdLinux.json [↑](#footnote-ref-35)
36. <https://tools.ietf.org/rfcmarkup?rfc=2254> [↑](#footnote-ref-36)
37. После того, как интеграция с AD настроена [↑](#footnote-ref-37)
38. Если не разрешена мультитенантная AD-авторизация (см. п. 9.1) [↑](#footnote-ref-38)
39. <https://github.com/jstedfast/MailKit/blob/master/FAQ.md#SslHandshakeException> [↑](#footnote-ref-39)
40. Не идут в комплекте поставки [↑](#footnote-ref-40)
41. Дистрибутивы Робота идут в комплекте поставки [↑](#footnote-ref-41)
42. Поставляются отдельно [↑](#footnote-ref-42)
43. Рекомендуется включать с версии Студии, которая поддерживает работу с папками [↑](#footnote-ref-43)
44. Функция реализована пока только для PostgeSQL [↑](#footnote-ref-44)
45. Это правило должно идти первым или выше правила для прокси /api/ [↑](#footnote-ref-45)
46. В настоящей реализации такое событие всегда одно [↑](#footnote-ref-46)
47. Маленькие значения рекомендуется задавать только для тестирования режима [↑](#footnote-ref-47)
48. Для определенности считаем их всех равными [↑](#footnote-ref-48)
49. Различные таймауты сетевых подключений, отсутствие свободных подключений к БД и т.п. [↑](#footnote-ref-49)
50. Хотя этот поток событий система генерирует сама, делает это она на основе настроек пользователя, которые в этой части не подконтрольны. [↑](#footnote-ref-50)
51. Из промежуточной очереди в обработку забирается не более одного элемента через фиксированный промежуток времени ChainLimiterTriggerEventTime. [↑](#footnote-ref-51)
52. Или обрабатываться с неприемлемой задержкой, если параметр MaxTriggerQueueLength выбран достаточно большим. [↑](#footnote-ref-52)
53. Особенность реализации [↑](#footnote-ref-53)
54. Виджет «Интенсивность потока событий от триггеров» на главной странице [↑](#footnote-ref-54)
55. Для демонстрации, реально это количество должно быть порядка нескольких тысяч [↑](#footnote-ref-55)
56. В случае кластера WebApi – для каждого NodeId [↑](#footnote-ref-56)
57. Стрелками показано увеличение/уменьшение очереди. [↑](#footnote-ref-57)
58. Сразу расшифровка в виде процента. [↑](#footnote-ref-58)
59. Для разблокировки - False [↑](#footnote-ref-59)
60. <https://learn.microsoft.com/en-us/dotnet/api/system.text.encoding.getencodings?view=net-8.0> [↑](#footnote-ref-60)
61. Только для Windows, для Linux непосредственный запуск [↑](#footnote-ref-61)
62. Для версий Windows-роботов, которые поднимали свои конечные точки. Не используется в Linux [↑](#footnote-ref-62)