**ИНСТРУКЦИЯ ПО УСТАНОВКЕ**

**RT.CLUSTERMANAGER**

2025

1. Введение

## 1.1 Назначение документа

Данный документ содержит описание действий и условий необходимых для установки программного обеспечения **RT.ClusterManager**.

## 1.2 Термины, определения и сокращения

В настоящем документе использованы и определены следующие термины и сокращения:

| ***Термин/Сокращение*** | ***Определение*** |
| --- | --- |
| Ansible (software) | Система управления конфигурациями. Используется в работе **RT.ClusterManager** для запуска ansible-ролей |
| ansible (user) | (как пользователь) – фиксировано заданный (в данном документе) пользователь для корректной работы ansible-ролей из контейнера rtcm на хосты. При необходимости имя данного пользователя можно заменить везде на другое, но новый пользователь должен уметь подключаться к хостам посредством ключевой пары без пароля (в том числе при выполнении sudo). |
| CM-server | Главный (управляющий) виртуальный или физический сервер, на котором будет развёрнута сама Система, посредством запуска docker-контейнеров и доступен веб-интерфейс. |
| Docker | Платформа, которая упрощает процесс сборки, запуска, управления и распространения приложений с помощью механизма контейнеризации. |
| On-Premise | Модель локального развертывания программного обеспечения, которая предполагает использование инфраструктуры, принадлежащей непосредственно компании, — серверов, хранилищ данных и другого оборудования. В этом случае ПО находится под контролем организации, в отличие от использования облачной платформы, где обслуживание и управление ресурсами происходит на стороне провайдера. |
| Интрасеть | Внутренняя сеть организации или предприятия, обычно построечная на базе семейства протоколов TCP/IP и предоставляющая возможность абонентам обмениваться данными с одним или несколькими веб-серверами, на которых сосредоточены основные информационные ресурсы. В отличие от Internet такая сеть, как правило, обеспечивает повышенную надежность и имеет ограниченный доступ. |
| ОС | Операционная система. |
| Рабочая\_директория  ***/opt/rtcm*** | Фиксировано заданная рабочая директория, откуда будут производиться все необходимые действия. Для примера ***/opt/rtcm***, при необходимости можно изменить на другую (но использовать ее везде, в том числе и в конфигурационных файлах). Важно отметить, что конечная директория обязательно должна быть “***rtcm***” |
| Резолвить | (от английского resolve, resolving) Применяется для обозначения процедуры получения IP-адреса по имени (прямой резолвинг) или имени по IP-адрес (обратный). |
| Репозито́рий | Место, где хранятся и откуда поставляются установочные пакеты. Чаще всего в репозитории хранятся файлы, доступные для дальнейшего распространения по сети. |
| Система | Система «**RT.ClusterManager**». |
| СУБД | Система управления базами данных. |
| Хост | от англ. Host – «владелец, принимающий гостей») – любое устройство, предоставляющее сервисы формата «клиент-сервер» в режиме сервера по каким-либо интерфейсам и уникально определённое на этих интерфейсах. В более широком смысле под хостом могут понимать любой компьютер, подключённый к локальной или глобальной сети. |

## 1.3 Эксплуатационные требования

Вам понадобится пул виртуальных (можно и физических) машин в интрасети или хосты арендованные у хостинг-провайдера, находящиеся в одной подсети, имеющие статический MAC и IPv4 адреса (IPv6 hadoop не поддерживает), а именно:

* CM-server (Linux) – сервер управления Системой (**RT.ClusterManager**), а также локальный репозиторий для установки Системы;
* Master-node (Linux) – серверы управления slave-нодами. Необходимо 1-2 сервера (минимум 2 в отказоустойчивом режиме);
* Slave-node (Linux) – серверы хранения и обработки данных. Необходимо 2-3 сервера;
* Kerberos-server (Windows) – Domain Controller (DC) (MS Windows Server 2016) с развёрнутой службой каталогов Active Directory и настроенным доменом (можно тестовым).

### 1.3.1 Требования к ОС

Предполагается использование следующих дистрибутивов Linux:

* Centos 7,
* Redos 7.2/Redos7.3,
* RHEL 7,
* Ubuntu 20.04.

**Примечание:**Для продукта **RT.Streaming** на данный момент поддерживаются только операционные системы: Centos 7, RHEL 7

### 1.3.2 Минимальные требования к оборудованию

Минимальные требования к оборудованию для установки **RT.ClusterManager**:

* Центральный процессор: 4 ядра;
* Оперативная память: 8 Гб.;
* Жесткий диск: 120 Гб.

Требования к оборудованию для развертывания базовой конфигурации кластеров, создаваемых с помощью **RT.ClusterManager** представлены в [п.4.4.2 документа «RT.ClusterManager.Общее описание»](https://docs.data.rt.ru/%D0%94%D0%BE%D0%BA%D1%83%D0%BC%D0%B5%D0%BD%D1%82%D0%B0%D1%86%D0%B8%D1%8F/RT_ClusterManager/Vers_2/General_description_ver2#:~:text=4.4.2%20%D0%A2%D1%80%D0%B5%D0%B1%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D1%8F%20%D0%BA%20%D0%BE%D0%B1%D0%BE%D1%80%D1%83%D0%B4%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D1%8E).

### 1.3.3 Сетевые требования

Сетевые требования для установки **RT.ClusterManager** и корректной работы:

* Все хосты должны находиться в одной сети;
* Должен быть обеспечен прямой и обратный резолв;

**Важно:**Для корректной работы кластеров, управляемых **RT.ClusterManager**, необходимо использование DNS-сервера, обеспечивающего прямой и обратный резолв всех хостов кластера, в том числе хоста, на котором установлен **RT.ClusterManager**.

### 1.3.4 Чек-лист для проверки стенда перед инсталляцией

Перед выполнением инсталляции необходимо выполнить данные проверки для соответствия хостов предъявляемым к ним требованиям.

**1. Системные требования**

|  |  |  |
| --- | --- | --- |
| Достаточное количество ОЗУ | free -m, top | Убедиться, что доступно не менее 8Gb |
| Место на диске | lsblk, df -h | Убедиться, что имеется не менее 100Gb |

**2. Сетевые требования**

|  |  |  |
| --- | --- | --- |
| Проверка системного времени, даты, часового пояса | date, timedatectl | Время и часовой пояс в норме, а так же синхронизированы с AD, freeIPA (если используется Kerberos). Убедиться что включена синхронизация по протоколу NTP. |
| FQDN Имя: <server-name.domain.com> | hostname -f | Имя показывает имя сервера и домен |
| Короткое имя: <server-name> | hostname -s | Отображает только короткое имя - имя сервера |
| Только доменное имя: <domain.com> | hostname -d | Отображает только доменное имя |
| В hosts не прописано имя хоста, только localhost | cat /etc/hosts | нет упоминания названия сервера в файле hosts: 127.0.0.1 server.exemple.domain.ru |
| ipv6 отключен (актуально для Hadoop и Greenplum) | ip -6 a | В выводе команды нет упоминания inet6 |
| Резолв короткого и FQDN имени | ping <server> | Пакеты пересылаются, сетевая связанность есть |
| Обратный резолв имени | nslookup <server> | yum install bind-utils nslookup возвращает корректное имя нет строки \*\* server can't find <server\_name>: NXDOMAIN как альтернатива можно использовать утилиту dig |
| Проверка поискового домена и адреса DNS-сервера | cat /etc/resolv.conf | в выводе есть упоминание домена search <my\_domain> |
| iptables нет лишних правил | iptables -L | Все 3-и цепочки (INPUT, FORWARD, OUTPUT) имеют разрешительную политику по умолчанию (policy ACCEPT) |
| nftables нет лишних правил | nft list ruleset | Команда ничего не выдаёт в консоль |
| Проверка порта. Например ssh порта | nc -vz <server> 22 | yum install nc (если не установлен по умолчанию) Если порт открыт: Ncat: 0 bytes sent, 0 bytes received in 0.01 seconds. Если порт закрыт: Ncat: Connection refused. |
| Проверка работы порта на самом сервисе | ss -tulpn | grep 22 | Порт должен быть в состоянии LISTEN |
| Безпарольный доступ по ssh | ssh <server> | Подключение проходит нормально, по командам (ip a) и (hostname) мы понимаем что подключились к нужному хосту |
| Безпарольный доступ по sudo | sudo su | После ввода команды, пароль не запрашивается и команда whoami выдаёт: root |

**3. Требования файловой системы**

|  |  |  |
| --- | --- | --- |
| Дескрипторы больше 12 000 | ulimit -Sn, ulimit -Hn | Если в выводе команды 1024, то поднимаем лимиты с помощью команды: ulimit -n 12000 |

# 2. Подготовка к работе

**Примечание:**Предполагается, что рабочая директория проекта - **/opt/rtcm**, а директория локального репозитория -**/opt/localrepo**  
если планируется использование других директорий - учитывайте это при выполнении команд.

Установку **RT.ClusterManager** возможно произвести двумя способами:

* Через стандартные репозитории компании Ростелеком, доступные через интернет (см. описание [Nexus](https://docs.data.rt.ru/%D0%94%D0%BE%D0%BA%D1%83%D0%BC%D0%B5%D0%BD%D1%82%D0%B0%D1%86%D0%B8%D1%8F/RT_ClusterManager/Nexus));
* Посредством установки и настройки локального репозитория в интрасети Вашей компании.

Для доступа к архивам со всеми необходимыми компонентами **RT.ClusterManager** и других систем предлагаемых ПАО Ростелеком своим клиентам используется Личный Кабинет. Также отдельные компоненты можно скачать в репозиториях (см. описание [Nexus](https://docs.data.rt.ru/%D0%94%D0%BE%D0%BA%D1%83%D0%BC%D0%B5%D0%BD%D1%82%D0%B0%D1%86%D0%B8%D1%8F/RT_ClusterManager/Nexus)).

В случае использования метода локального репозитория, перед установкой **RT.ClusterManager** необходимо выполнить запуск и настройку локального репозитория (см. [п.2.2](https://docs.data.rt.ru/%D0%94%D0%BE%D0%BA%D1%83%D0%BC%D0%B5%D0%BD%D1%82%D0%B0%D1%86%D0%B8%D1%8F/RT_ClusterManager/Installation_RT_ClusterManager#:~:text=2.2%20%D0%9D%D0%B0%D1%81%D1%82%D1%80%D0%BE%D0%B9%D0%BA%D0%B0%20%D0%BB%D0%BE%D0%BA%D0%B0%D0%BB%D1%8C%D0%BD%D0%BE%D0%B3%D0%BE%20%D1%80%D0%B5%D0%BF%D0%BE%D0%B7%D0%B8%D1%82%D0%BE%D1%80%D0%B8%D1%8F%20(%D0%BF%D1%80%D0%B8%20%D0%BD%D0%B5%D0%BE%D0%B1%D1%85%D0%BE%D0%B4%D0%B8%D0%BC%D0%BE%D1%81%D1%82%D0%B8))). Локальный репозиторий предоставляет доступ внутри интрасети организации к архиву с инсталляционными файлами для серверов кластера (Master и Slave nodes) и сервера, на который будет устанавливаться **RT.ClusterManager** (CM-server).

## 2.1 Состав дистрибутивного архива

**1. RT.ClusterMahager**

| ***Тип файла*** | ***Название файла в архиве*** | ***Действие*** |
| --- | --- | --- |
| Архив с установочными пакетами для формирования локального rpm репозитория | /opt/localrepo/archives/rt.ca.repo\_rpm.tar.gz | Установка репозитория (см. [п. 2.2](https://docs.data.rt.ru/%D0%94%D0%BE%D0%BA%D1%83%D0%BC%D0%B5%D0%BD%D1%82%D0%B0%D1%86%D0%B8%D1%8F/RT_ClusterManager/Installation_RT_ClusterManager#:~:text=2.2%20%D0%9D%D0%B0%D1%81%D1%82%D1%80%D0%BE%D0%B9%D0%BA%D0%B0%20%D0%BB%D0%BE%D0%BA%D0%B0%D0%BB%D1%8C%D0%BD%D0%BE%D0%B3%D0%BE%20%D1%80%D0%B5%D0%BF%D0%BE%D0%B7%D0%B8%D1%82%D0%BE%D1%80%D0%B8%D1%8F%20(%D0%BF%D1%80%D0%B8%20%D0%BD%D0%B5%D0%BE%D0%B1%D1%85%D0%BE%D0%B4%D0%B8%D0%BC%D0%BE%D1%81%D1%82%D0%B8)) настоящего документа). |
| Архив с установочными пакетами для формирования локального deb репозитория | /opt/localrepo/archives/rt.ca.repo\_deb.tar.gz |
| Директория с установочными пакетами сервиса prometheus для формирования локального репозитория | /opt/localrepo/archives/rt.prometheus.repo |
| Docker образ **RT.ClusterManager** | /opt/localrepo/archives/rtcm/images\_docker/rtcm\_версия.tar.gz | Загрузка doсker-образов из архивов (см. [п. 3.2 шаг 2](https://docs.data.rt.ru/%D0%94%D0%BE%D0%BA%D1%83%D0%BC%D0%B5%D0%BD%D1%82%D0%B0%D1%86%D0%B8%D1%8F/RT_ClusterManager/Installation_RT_ClusterManager#:~:text=2.%20%D0%97%D0%B0%D0%B3%D1%80%D1%83%D0%B7%D0%BA%D0%B0%20%D0%BE%D0%B1%D1%80%D0%B0%D0%B7%D0%BE%D0%B2%20%D0%B2%D1%80%D1%83%D1%87%D0%BD%D1%83%D1%8E) настоящего документа). |
| Docker образ Prometheus | /opt/localrepo/archives/rtcm/images\_docker/prometheus\_версия.tar.gz |
| Docker образ Blackbox | /opt/localrepo/archives/rtcm/images\_docker/blackbox\_версия.tar.gz |
| Docker образ Postgres | /opt/localrepo/archives/rtcm/images\_docker/postgres\_версия.tar.gz |
| Скрипт для автоматизации загрузки docker-образов  load\_images | /opt/localrepo/archives/rtcm/images\_docker/load\_images.sh | Подгрузка образов командой ./load\_images.sh (см. [п. 3.2 шаг 1](https://docs.data.rt.ru/%D0%94%D0%BE%D0%BA%D1%83%D0%BC%D0%B5%D0%BD%D1%82%D0%B0%D1%86%D0%B8%D1%8F/RT_ClusterManager/Installation_RT_ClusterManager#:~:text=1.%20%D0%97%D0%B0%D0%BF%D1%83%D1%81%D0%BA%20%D1%81%D0%BA%D1%80%D0%B8%D0%BF%D1%82%D0%B0%20%D0%B0%D0%B2%D1%82%D0%BE%D0%BC%D0%B0%D1%82%D0%B8%D1%87%D0%B5%D1%81%D0%BA%D0%BE%D0%B9%20%D0%BF%D0%BE%D0%B4%D0%B3%D1%80%D1%83%D0%B7%D0%BA%D0%B8%20%D0%BE%D0%B1%D1%80%D0%B0%D0%B7%D0%BE%D0%B2:) настоящего документа). |
| Файл конфигурации сервиса Blackbox | /opt/localrepo/archives/blackbox/blackbox.yml | Редактирование файлов конфигурации |
| Файл конфигурации сервиса **RT.ClusterManager** | /opt/localrepo/archives/rtcm\_conf/rtcm.properties | Редактирование файлов конфигурации |
| Файл конфигурации сервиса prometheus для оповещений | /opt/localrepo/archives/config/alert.rules.yml | Редактирование файлов конфигурации |
| Файл конфигурации сервиса prometheus для сбора метрик | /opt/localrepo/archives/config/prometheus.yml | Редактирование файлов конфигурации |
| Скрипт для автоматизации операций  администрирования сервиса **RT.ClusterManager** | /opt/localrepo/archives/images/docker/topgun.sh | Запуск скрипта Параметры запуска описаны в [п. 6 документа “API для работы с RT.ClusterManager”.](https://docs.data.rt.ru/%D0%94%D0%BE%D0%BA%D1%83%D0%BC%D0%B5%D0%BD%D1%82%D0%B0%D1%86%D0%B8%D1%8F/RT_ClusterManager/rt_clustermanager_API#:~:text=6.%20%D0%98%D1%81%D0%BF%D0%BE%D0%BB%D1%8C%D0%B7%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5%20%D1%81%D0%BA%D1%80%D0%B8%D0%BF%D1%82%D0%B0%20topgun.sh%20%D0%B4%D0%BB%D1%8F%20%D1%83%D0%BF%D1%80%D0%B0%D0%B2%D0%BB%D0%B5%D0%BD%D0%B8%D1%8F%20RT.ClusterManager) |
| docker-compose.yaml  Файл описывающий параметры запускаемых контейнеров | /opt/localrepo/archives/rtcm/docker-compose.yaml | Конфигурирование и запуск (см. [п. 3.3](https://docs.data.rt.ru/%D0%94%D0%BE%D0%BA%D1%83%D0%BC%D0%B5%D0%BD%D1%82%D0%B0%D1%86%D0%B8%D1%8F/RT_ClusterManager/Installation_RT_ClusterManager#:~:text=3.3%20%D0%9F%D0%BE%D0%B4%D0%B3%D0%BE%D1%82%D0%BE%D0%B2%D0%BA%D0%B0%20%D0%BD%D0%B0%D1%81%D1%82%D1%80%D0%BE%D0%B5%D0%BA) и [п. 3.4](https://docs.data.rt.ru/%D0%94%D0%BE%D0%BA%D1%83%D0%BC%D0%B5%D0%BD%D1%82%D0%B0%D1%86%D0%B8%D1%8F/RT_ClusterManager/Installation_RT_ClusterManager#:~:text=3.4%20%D0%97%D0%B0%D0%BF%D1%83%D1%81%D0%BA/%D0%BF%D0%B5%D1%80%D0%B5%D0%B7%D0%B0%D0%BF%D1%83%D1%81%D0%BA%20%D0%BA%D0%BE%D0%BD%D1%82%D0%B5%D0%B9%D0%BD%D0%B5%D1%80%D0%B0) настоящего документа). |
| Плагины:  **RT.System** | /opt/localrepo/archives/RT.System-версия.tar.gz | Данные файлы в не распакованном виде подгружаются:  1. При создании кластера с помощью wizard (см. [п. 6 шаг 4, Руководства администратора](https://docs.data.rt.ru/%D0%94%D0%BE%D0%BA%D1%83%D0%BC%D0%B5%D0%BD%D1%82%D0%B0%D1%86%D0%B8%D1%8F/RT_ClusterManager/Vers_2/RT_ClusterManager_Admin_Guide_ver2#:~:text=4.%20%D0%A8%D0%B0%D0%B3%20%E2%80%9C%D0%A1%D0%BE%D0%B7%D0%B4%D0%B0%D0%BD%D0%B8%D0%B5%20%D0%BA%D0%BB%D0%B0%D1%81%D1%82%D0%B5%D1%80%D0%B0%E2%80%9D,%20%D0%BD%D0%B0%20%D1%8D%D1%82%D0%BE%D0%BC%20%D1%88%D0%B0%D0%B3%D0%B5%20%D0%BD%D0%B5%D0%BE%D0%B1%D1%85%))  2. В UI**RT.ClusterManager** при загрузке плагинов вручную (см. [п. 7.1 Руководства администратора](https://docs.data.rt.ru/%D0%94%D0%BE%D0%BA%D1%83%D0%BC%D0%B5%D0%BD%D1%82%D0%B0%D1%86%D0%B8%D1%8F/RT_ClusterManager/Vers_2/RT_ClusterManager_Admin_Guide_ver2#:~:text=7.1%20%D0%97%D0%B0%D0%B3%D1%80%D1%83%D0%B7%D0%BA%D0%B0%20%D0%BF%D0%BB%D0%B0%D0%B3%D0%B8%D0%BD%D0%BE%D0%B2)). |

**2. Плагин приложения на примере RT.DataLake**

| ***Тип файла*** | ***Название файла в архиве*** | ***Действие*** |
| --- | --- | --- |
| Архив с установочными пакетами для формирования локального репозитория | rt.datalake.repo.tar.gz | Установка репозитория см. [п. 2.2](https://docs.data.rt.ru/%D0%94%D0%BE%D0%BA%D1%83%D0%BC%D0%B5%D0%BD%D1%82%D0%B0%D1%86%D0%B8%D1%8F/RT_ClusterManager/Installation_RT_ClusterManager#:~:text=2.2%20%D0%9D%D0%B0%D1%81%D1%82%D1%80%D0%BE%D0%B9%D0%BA%D0%B0%20%D0%BB%D0%BE%D0%BA%D0%B0%D0%BB%D1%8C%D0%BD%D0%BE%D0%B3%D0%BE%20%D1%80%D0%B5%D0%BF%D0%BE%D0%B7%D0%B8%D1%82%D0%BE%D1%80%D0%B8%D1%8F%20(%D0%BF%D1%80%D0%B8%20%D0%BD%D0%B5%D0%BE%D0%B1%D1%85%D0%BE%D0%B4%D0%B8%D0%BC%D0%BE%D1%81%D1%82%D0%B8)) настоящего документа |
| Плагины:  **RT.DataLake** | RT.DataLake-версия.tar.gz | Данные файлы в не распакованном виде подгружаются:  1. При создании кластера с помощью wizard (см. [п. 6 шаг 4, Руководства администратора](https://docs.data.rt.ru/%D0%94%D0%BE%D0%BA%D1%83%D0%BC%D0%B5%D0%BD%D1%82%D0%B0%D1%86%D0%B8%D1%8F/RT_ClusterManager/Vers_2/RT_ClusterManager_Admin_Guide_ver2#:~:text=4.%20%D0%A8%D0%B0%D0%B3%20%E2%80%9C%D0%A1%D0%BE%D0%B7%D0%B4%D0%B0%D0%BD%D0%B8%D0%B5%20%D0%BA%D0%BB%D0%B0%D1%81%D1%82%D0%B5%D1%80%D0%B0%E2%80%9D,%20%D0%BD%D0%B0%20%D1%8D%D1%82%D0%BE%D0%BC%20%D1%88%D0%B0%D0%B3%D0%B5%20%D0%BD%D0%B5%D0%BE%D0%B1%D1%85%D0%BE%D0%B4%D0%B8%D0%BC%D0%BE))   2. В UI **RT.ClusterManager** при загрузке плагинов вручную (см. [п. 7.1 Руководства администратора](https://docs.data.rt.ru/%D0%94%D0%BE%D0%BA%D1%83%D0%BC%D0%B5%D0%BD%D1%82%D0%B0%D1%86%D0%B8%D1%8F/RT_ClusterManager/Vers_2/RT_ClusterManager_Admin_Guide_ver2#:~:text=7.1%20%D0%97%D0%B0%D0%B3%D1%80%D1%83%D0%B7%D0%BA%D0%B0%20%D0%BF%D0%BB%D0%B0%D0%B3%D0%B8%D0%BD%D0%BE%D0%B2)). |

**Примечание:**В именах файлов, в место “**версия”** должен присутствовать код версии данного файла который обычно состоит из нескольких частей разделённых точками.

**Примечание:**Состав системных плагинов и плагинов приложений зависит от комплекта поставки программного обеспечения. Плагины понадобятся при создании провайдера и кластеров в веб-интерфейсе **RT.ClusterManager**, на этапе подготовке к установке **RT.ClusterManager** они не нужны.

## 2.2 Настройка локального репозитория (при необходимости)

Обратите внимание, что начиная с шага 6 команды отличаются для разных ОС

### 2.2.1 Общие команды для любых ОС

Сервер, на который будет установлен локальный репозиторий должен быть доступен для Master-nodes, Slave-nodes и CM-server.

Для установки локального репозитория необходимо выполнить следующие операции:

1. Создаём директорию проектируемого локального репозитория и необходимые вложенные директории, например:

mkdir -p /opt/localrepo/archives

Copy

2. Заходим в директорию с будущими архивами локального репозитория:

cd /opt/localrepo/archives/

Copy

3. Скачиваем инсталляционные архивы на CM-server:

wget --user=пользователь --ask-password --no-check-certificate ссылка\_на\_архив\_Ростелеком

вводим пароль, предоставленный компанией Ростелеком

Copy

4. Распаковываем скаченный архив командой:

tar zxf RT.ClusterManager\_версия\*.tar.gz

tar zxf RT.DataLake\_версия\*.tar.gz

tar zxf RT.WareHouse\_версия\*.tar.gz -C /opt/localrepo/archives/RT.WareHouse/

tar zxf RT.WideStore\_версия\*.tar.gz -C /opt/localrepo/archives/RT.WideStore/

Copy

5.  Распаковываем архив с проектом в созданных директориях проектов **/opt/localrepo/archives/\*** :

tar -zxf название архива (например, RT.ClusterManager\_версия.tar.gz)

Copy

6. После распаковки удаляем изначальные архивы для сохранения места:

rm /opt/localrepo/archives/RT.ClusterManager\_версия.tar.gz

rm /opt/localrepo/archives/RT.DataLake\_версия.tar.gz

Copy

Далее команды в зависимости от ОС.

### 2.2.2 для CentOS, RedOS (для других ОС смотри ниже)

1. Создаем директорию **packages** для инсталляционных пакетов, переносим в нее архивы c названием **\*.repo** (локальный репозиторий rt.ca.repo находится в архиве **RT.ClusterManager**, остальные находятся в одноименных архивах продуктов из ЛК)

mkdir /opt/localrepo/packages

# общие репозитории - нужны всегда

mv /opt/localrepo/archives/rtcm/rt.ca.repo\_rpm.tar.gz /opt/localrepo/packages/

mv /opt/localrepo/archives/rtcm/rt.prometheus\_exporter.repo /opt/localrepo/packages/

# репозитории продуктов - зависят от поставки

mv /opt/localrepo/archives/RT.DataLake/rt.datalake.repo.tar.gz /opt/localrepo/packages/

mv /opt/localrepo/archives/RT.StreamingKafka/rt.streamingkafka.repo.tar.gz /opt/localrepo/packages/

mv /opt/localrepo/archives/RT.StreamingNiFi/rt.streamingnifi.repo.tar.gz /opt/localrepo/packages/

mv /opt/localrepo/archives/RT.WareHouse/rt.warehouse.repo.tar.gz /opt/localrepo/packages/

mv /opt/localrepo/archives/RT.Monitoring/rt.monitoring.repo.tar.gz /opt/localrepo/packages/

Copy

2. Распаковываем перемещенные архивы:

cd /opt/localrepo/packages

tar -xzvf rt.ca.repo\_rpm.tar.gz

tar -xzvf rt.datalake.repo.tar.gz

tar -xzvf rt.streaming.repo.tar.gz

# Репозиторий prometheus не заархивирован, поэтому нет необходимости в его распаковывании

Copy

3. Удаляем изначальные архивы локальных репозиториев для сохранения места:

rm rt.ca.repo\_rpm.tar.gz rt.datalake.repo.tar.gz rt.streaming.repo.tar.gz rt.prometheus\_exporter.repo.tar.gz

ls

Copy

4.  Внутри каждого репозитория удалим пакеты, не относящиеся к CentOS/RedOS для избегания коллизий

# для warehouse

cd rt.warehouse.repo

# java

rm -rf ./java-\*-alt0\*

# rt.wh

rm -rf ./rt.wh-\*.alt10.\*

cd ../

Copy

**Примечание:** В результате директория должна содержать репозитории для работы **RT.ClusterManager** - **rt.ca.repo** и **rt.prometheus\_exporter.repo**, а также репозитории самих сервисов, например, **rt.datalake.repo** или **rt.streaming.repo.**

5. Если на сервере не установлен ***artifactory/nexus/nginx/apache***, альтернативные хранилища репозиториев или web-серверы для создания локального репозитория, устанавливаем **nginx**:

sudo yum install nginx

Copy

Если пакетный менеджер **yum** не может найти пакет, возможно нужно установить **epel**или его аналог репозиторий командой:

yum install epel-release

Copy

6. Настраиваем nginx:

sudo vi /etc/nginx/conf.d/localrepo.conf

Copy

**Примечание:** Для работы с **vi** в режиме редактирования необходимо нажать клавишу **i**  
В нижней левой части экрана отобразится приглашение к вводу **--INSERT--**, затем, с помощью стрелок необходимо переместиться на нужную строку, изменить ее и после редактирования нажать клавишу **ESC**(для выхода из данного режима). Затем напечатать **:wq** для сохранения и выхода из файла.

Пример файла конфигурации:

**Примечание:** В примере необходимо изменить **имя\_сервера** на имя хоста (из вывода команды ‘**hostname -f**’ или его **ip**-адрес)

server {

listen 1337 default\_server; # порт, на котором будет отдаваться репозиторий

listen [::]:1337 default\_server; # порт, на котором будет отдаваться репозиторий

server\_name имя\_сервера; # ip адрес или hostname сервера

root /opt/localrepo; # путь до репозитория

# Load configuration files for the default server block.

include /etc/nginx/default.d/\*.conf;

# расположение, по которому будет доступен репозиторий

location / {

allow all;

sendfile on;

sendfile\_max\_chunk 1m;

autoindex on;

autoindex\_exact\_size off;

autoindex\_format html;

autoindex\_localtime on;

}

error\_page 404 /404.html;

location = /40x.html {

}

error\_page 500 502 503 504 /50x.html;

location = /50x.html {

}

}

Copy

7. Добавляем **nginx**в автозагрузку и последующий запуск:

service nginx enable

service nginx start

Copy

8. После переконфигурации nginx репозитории будут доступны по ссылкам:

http://полный\_адрес\_сервера\_с\_репозиторием:1337/packages/rt.ca.repo/

http://полный\_адрес\_сервера\_с\_репозиторием:1337/packages/rt.datalake.repo/

http://полный\_адрес\_сервера\_с\_репозиторием:1337/packages/rt.streaming.repo/

http://полный\_адрес\_сервера\_с\_репозиторием:1337/packages/rt.prometheus.repo/

9. В дальнейшем в web-интерфейсе **RT.ClusterManager** во вкладке **Настройки → repos**необходимо поменять репозитории на:

http://полный\_адрес\_машины\_с\_репозиторием:1337/packages/название\_репозиториев

([п. 8.2.1 шаг 4,  документа «RT.ClusterManager. Руководство администратора»](https://docs.data.rt.ru/%D0%94%D0%BE%D0%BA%D1%83%D0%BC%D0%B5%D0%BD%D1%82%D0%B0%D1%86%D0%B8%D1%8F/RT_ClusterManager/Vers_2/RT_ClusterManager_Admin_Guide_ver2#:~:text=4.%20%D0%92%20%D0%BF%D0%BE%D0%BB%D1%8F%D1%85%20%D1%84%D0%BE%D1%80%D0%BC%D1%8B,%20%D0%B4%D0%BB%D1%8F%20%D0%BF%D0%BE%D0%BB%D0%B5%D0%B9)).

### 2.2.3 для Altlinux 10

1. Создаем директорию **packages** для инсталляционных пакетов, переносим в нее архивы c названием **\*.repo** (локальный репозиторий rt.ca.repo находится в архиве **RT.ClusterManager**, остальные находятся в одноименных архивах продуктов из ЛК)

export REPO\_DIR=/opt/localrepo

export arch=x86\_64

export REPO\_NAME\_CA=rt.ca.repo

export REPO\_NAME\_WH=rt.warehouse.repo

mkdir -p $REPO\_DIR/$arch/base

# общие репозитории - нужны всегда

mv /opt/localrepo/archives/rtcm/rt.ca.repo.tar.gz $REPO\_DIR/$arch/

# пакеты прометея на самом деле простые бинари и не требуют лежать в общей структуре репозитория

mv /opt/localrepo/archives/rtcm/rt.prometheus.repo $REPO\_DIR/

# репозитории продуктов - зависят от поставки

mv /opt/localrepo/archives/RT.WareHouse/rt.warehouse.repo.tar.gz $REPO\_DIR/$arch/

Copy

2. Распаковываем перемещенные архивы и меняем имена директорий в соответствии со структурой репозитория:

cd $REPO\_DIR/$arch

tar -xzvf rt.ca.repo.tar.gz

tar -xzvf rt.warehouse.repo.tar.gz

# Репозиторий prometheus не заархивирован, поэтому нет необходимости в его распоковывании

mv ./rt.warehouse.repo ./RPMS.rt.warehouse.repo/

mv ./rt.ca.repo ./RPMS.rt.ca.repo/

Copy

3. Удаляем изначальные архивы локальных репозиториев для сохранения места:

rm rt.ca.repo.tar.gz rt.warehouse.repo.tar.gz

ls

Copy

4 Внутри каждого репозитория удалим пакеты, не относящиеся к **Altlinux 10** для избегания коллизий

# для warehouse

cd rt.warehouse.repo

# java

rm -rf ./java-\*.el7\*

# rt.wh

rm -rf ./rt.\*.el7.\*

rm -rf ./rt.\*.r73c.\*

# прочие пакеты

rm -rf ./\*.el7\*

cd ../

Copy

**Примечание:** В результате директория должна содержать репозитории для работы **RT.ClusterManager** - **rt.ca.repo** и **rt.prometheus.repo**, а также репозитории самих сервисов, например, **rt.warehouse.repo.**

5. Если на сервере не установлен ***artifactory/nexus/nginx/apache***, альтернативные хранилища репозиториев или web-серверы для создания локального репозитория, устанавливаем **nginx**:

sudo apt-get install nginx

Copy

6. Настраиваем **nginx**:

sudo vim /etc/nginx/sites-available.d/repo.conf

Copy

**Примечание:** Для работы с **vi** в режиме редактирования необходимо нажать клавишу **i**  
В нижней левой части экрана отобразится приглашение к вводу **--INSERT--**, затем, с помощью стрелок необходимо переместиться на нужную строку, изменить ее и после редактирования нажать клавишу **ESC**(для выхода из данного режима). Затем напечатать **:wq** для сохранения и выхода из файла.

Пример файла конфигурации:

**Примечание:** В примере необходимо изменить **имя\_сервера** на имя хоста (из вывода команды ‘**hostname -f**’ или его **ip**-адрес)

server {

listen 1337 default\_server; # порт, на котором будет отдаваться репозиторий

listen [::]:1337 default\_server; # порт, на котором будет отдаваться репозиторий

server\_name имя\_сервера; # ip адрес или hostname сервера

root /opt/localrepo; # путь до репозитория

# Load configuration files for the default server block.

include /etc/nginx/default.d/\*.conf;

# расположение, по которому будет доступен репозиторий

location / {

allow all;

sendfile on;

sendfile\_max\_chunk 1m;

autoindex on;

autoindex\_exact\_size off;

autoindex\_format html;

autoindex\_localtime on;

}

error\_page 404 /404.html;

location = /40x.html {

}

error\_page 500 502 503 504 /50x.html;

location = /50x.html {

}

}

Copy

7. Добавляем **nginx**в автозагрузку и последующий запуск:

sudo systemctl enable --now nginx

Copy

8. Генерируем метаданные для репозиториев

sudo apt-get install apt-repo-tools

genbasedir --bloat --progress --topdir=$REPO\_DIR $arch $REPO\_NAME\_CA

genbasedir --bloat --progress --topdir=$REPO\_DIR $arch $REPO\_NAME\_WH

Copy

9. Проверим работоспособность репозитория

sudo apt-repo add "rpm file:/opt/localrepo x86\_64 rt.ca.repo"

sudo apt-repo update

sudo apt-repo list

sudo apt-get install --download-only rt.ca

Copy

10. После переконфигурации **nginx**репозитории будут доступны по ссылкам:

http://полный\_адрес\_сервера\_с\_репозиторием:1337/x86\_64/RPMS.rt.ca.repo/

http://полный\_адрес\_сервера\_с\_репозиторием:1337/x86\_64/RPMS.rt.warehouse.repo/

http://полный\_адрес\_сервера\_с\_репозиторием:1337/rt.prometheus.repo/

Copy

11. В дальнейшем в web-интерфейсе **RT.ClusterManager** во вкладке **Настройки → repos**необходимо поменять репозитории на:

Для репозиториев, которые вам не нужны (например art-centos), их значение нужно очистить либо сделать равным ${

# для репозиториев ca и продуктов

rpm http://полный\_адрес\_машины\_с\_репозиторием:порт/ x86\_64 название\_репозиториев\_без\_RPMS

# для репозитория prometheus

http://полный\_адрес\_машины\_с\_репозиторием:порт

Copy

([п. 8.2.1 шаг 4,  документа «RT.ClusterManager. Руководство администратора»](https://docs.data.rt.ru/%D0%94%D0%BE%D0%BA%D1%83%D0%BC%D0%B5%D0%BD%D1%82%D0%B0%D1%86%D0%B8%D1%8F/RT_ClusterManager/Vers_2/RT_ClusterManager_Admin_Guide_ver2#:~:text=4.%20%D0%92%20%D0%BF%D0%BE%D0%BB%D1%8F%D1%85%20%D1%84%D0%BE%D1%80%D0%BC%D1%8B,%20%D0%B4%D0%BB%D1%8F%20%D0%BF%D0%BE%D0%BB%D0%B5%D0%B9)).

### 2.2.4 для Astra Linux

1. Создаем директорию packages для инсталляционных пакетов, переносим в нее архивы c названием ***\*.repo*** (локальный репозиторий rt.ca.repo находится в архиве **RT.ClusterManager**, остальные находятся в одноименных архивах продуктов из ЛК)

mkdir /opt/localrepo/packages

# общие репозитории - нужны всегда

mv /opt/localrepo/archives/rtcm/rt.ca.repo\_deb.tar.gz /opt/localrepo/packages/

mv /opt/localrepo/archives/rtcm/rt.prometheus\_exporter.repo /opt/localrepo/packages/

# репозитории продуктов - зависят от поставки

mv /opt/localrepo/archives/RT.DataLake/rt.datalake.repo.tar.gz /opt/localrepo/packages/

mv /opt/localrepo/archives/RT.Streaming/rt.streaming.repo.tar.gz /opt/localrepo/packages/

Copy

2. Распаковываем перемещенные архивы:

cd /opt/localrepo/packages

tar -xzvf rt.ca.repo\_deb.tar.gz

tar -xzvf rt.datalake.repo.tar.gz

tar -xzvf rt.streaming.repo.tar.gz

# Репозиторий prometheus не заархивирован, поэтому нет необходимости в его распаковывании

Copy

3. Удаляем изначальные архивы локальных репозиториев для сохранения места:

rm rt.ca.repo\_deb.tar.gz rt.datalake.repo.tar.gz rt.streaming.repo.tar.gz rt.prometheus\_exporter.repo.tar.gz

ls

Copy

**Примечание:** В результате директория должна содержать репозитории для работы **RT.ClusterManager** **- rt.ca.repo** и **rt.prometheus\_exporter.repo**, а также репозитории самих сервисов, например, **rt.datalake.repo** или **rt.streaming.repo**.

4. Если на сервере не установлен ***artifactory/nexus/nginx/apache***, альтернативные хранилища репозиториев или web-серверы для создания локального репозитория, устанавливаем **nginx**:

sudo apt install nginx

Copy

Если пакетный менеджер ***apt***не может найти пакет, возможно нужно установить его вручную по инструкции - <https://wiki.astralinux.ru/pages/viewpage.action?pageId=71832638>  
5. Настраиваем **nginx**:

sudo vi /etc/nginx/conf.d/localrepo.conf

Copy

**Примечание**: Для работы с vi в режиме редактирования необходимо нажать клавишу i

В нижней левой части экрана отобразится приглашение к вводу -~~INSERT~~-, затем, с помощью стрелок необходимо переместиться на нужную строку, изменить ее и после редактирования нажать клавишу ESC (для выхода из данного режима). Затем напечатать ***:wq*** для сохранения и выхода из файла.  
Пример файла конфигурации:

**Примечание:** В примере необходимо изменить **имя\_сервера** на имя хоста (из вывода команды ‘**hostname -f**’ или его ip-адрес)

server {

listen 1337 default\_server; # порт, на котором будет отдаваться репозиторий

listen [::]:1337 default\_server; # порт, на котором будет отдаваться репозиторий

server\_name имя\_сервера; # ip адрес или hostname сервера

root /opt/localrepo; # путь до репозитория

# Load configuration files for the default server block.

include /etc/nginx/default.d/\*.conf;

# расположение, по которому будет доступен репозиторий

location / {

allow all;

sendfile on;

sendfile\_max\_chunk 1m;

autoindex on;

autoindex\_exact\_size off;

autoindex\_format html;

autoindex\_localtime on;

}

error\_page 404 /404.html;

location = /40x.html {

}

error\_page 500 502 503 504 /50x.html;

location = /50x.html {

}

}

Copy

6. Добавляем **nginx**в автозагрузку и последующий запуск:

systemctl enable nginx

systemctl start nginx

Copy

7. После переконфигурации **nginx**репозитории будут доступны по ссылкам:

http://полный\_адрес\_сервера\_с\_репозиторием:1337/packages/rt.ca.repo/

http://полный\_адрес\_сервера\_с\_репозиторием:1337/packages/rt.datalake.repo/

http://полный\_адрес\_сервера\_с\_репозиторием:1337/packages/rt.streaming.repo/

http://полный\_адрес\_сервера\_с\_репозиторием:1337/packages/rt.prometheus.repo/

Copy

8. В дальнейшем в web-интерфейсе **RT.ClusterManager** во вкладке ***Настройки → repos*** необходимо поменять репозитории на:

# для репозитория art-general-os (обязательно нужно указать base репозиторий астры, потому что из него устанавливаются необходимые java-пакеты)

deb https://dl.astralinux.ru/astra/stable/1.7\_x86-64/repository-base/ 1.7\_x86-64 main contrib non-free

# для репозитория prometheus

http://полный\_адрес\_сервера\_с\_репозиторием:порт/rt.prometheus.repo/

# для остальных репозиториев

deb [trusted=true] http://полный\_адрес\_машины\_с\_репозиторием:порт/название\_репозиториев 1.7\_x86-64 main

Copy

## 2.3 Подготовка хостов

ВНИМАНИЕ: Если используется ReOS версии 7.3.5, то необходимо на всех серверах кластера установить пакет yum

Это касается случаев, когда RedOS 7.3.5 ставится с нуля и не относиться к случаям, когда было произведено обновление например с 7.3.4 на 7.3.5

dnf install yum -y

Copy

Необходимо чтобы хосты соответствовали требованиям  изложенным в [п. 1.3.4](https://docs.data.rt.ru/%D0%94%D0%BE%D0%BA%D1%83%D0%BC%D0%B5%D0%BD%D1%82%D0%B0%D1%86%D0%B8%D1%8F/RT_ClusterManager/Installation_RT_ClusterManager#:~:text=1.3.4).

Для подготовки хостов необходимо:

1. На всех серверах и контейнере ***rtcm***(о контейнерах ниже по тексту, а также позже в [п. 3](https://docs.data.rt.ru/%D0%94%D0%BE%D0%BA%D1%83%D0%BC%D0%B5%D0%BD%D1%82%D0%B0%D1%86%D0%B8%D1%8F/RT_ClusterManager/Installation_RT_ClusterManager#:~:text=3.%20%D0%A3%D1%81%D1%82%D0%B0%D0%BD%D0%BE%D0%B2%D0%BA%D0%B0%20RT.ClusterManager)) должны быть синхронизированы часовые пояса и время.

2. Переименовать все хосты в удобном порядке.

Пример hostname FQDN:

|  |  |
| --- | --- |
| **Назначение хоста** | ***hostname fqdn*** |
| CM-server (cm) | rt-cm.corp.local |
| Kerberos-server (dc) | rt-dc.corp.local |
| Master-node 1 (m1) | rt-m1.corp.local |
| Master-node 2 (m2) | rt-m2.corp.local |
| Slave-node 1 (s1) | rt-s1.corp.local |
| Slave-node 2 (s2) | rt-s2.corp.local |
| Slave-node 3 (s3) | rt-s3.corp.local |

Убедиться, что в FQDN нет упоминания о localhost командой:

hostname -f

Copy

3. Отключение IPv6  
CentOS 7, RedOS 7.3 (на RedOS 7.2 процедура может быть иной)

**Важно!** Если IPv6 используется в качестве доступа по SSH, то что бы не потерять доступ необходимо, в ***/etc/ssh/sshd\_config contains*** добавить

# vi /etc/ssh/sshd\_config

....

AddressFamily inet

....

Copy

Для включения **Xforwarding**необходимо перезапустить сервис:

systemctl restart sshd

Copy

Проверить включён ли IPv6 можно командой:

ip a | grep inet6

Copy

Для отключения, необходимо отредактировать конфигурационный файл:

vi /etc/sysctl.conf

Copy

Добавить строки:

net.ipv6.conf.all.disable\_ipv6 = 1

net.ipv6.conf.default.disable\_ipv6 = 1

Copy

(если требуется только на одном интерфейсе, то вместо **all**можно указать своё название сетевого интерфейса)

sysctl -p

Copy

Проверить, что IPv6 отключился можно командой (вывод будет пустой, не должен найти строки с упоминанием inet6)

ip a | grep inet6

Copy

4. На всех linux-хостах необходимо создать пользователя с паролем для корректной работы ansible-ролей (в нашем примере везде будет пользователь **ansible**).  
При необходимости к этой команде можно добавить нужные ключи.

**Важно: Если используется Astra Linux, то необходимо использовать команду написанную ниже, вместо команды 'useradd ansible'**

**(это создаст домашнюю директорию, добавит пользователя в группу astra-admin, для sudo и задаст оболочку по умолчанию bash вместо sh, которая идёт по умолчанию у вновь созданных)**

useradd -m -G astra-admin -s /bin/bash ansible

Copy

Для rpm-like Linux (RedOS, CentOS)

useradd ansible

Copy

Выполните команду:

passwd ansible

Copy

Задайте пароль для пользователя.

5. На CM-server (находясь под пользователем **ansible**) необходимо сгенерировать ключевую пару асинхронного шифрования  
Во всех параметрах, которые будет запрашивать команда - нажимаем Enter, ничего не вводя.

ssh-keygen -t rsa -b 2048 -m PEM

Copy

**Примечание:** Поддерживаются только ключи в первой строке которых указан формат **RSA**, не короче 2048b

6. Добавить запись открытой части ключа на хосты, и на CM-server:

**Примечание:** Необходимо выполнить команду на каждом хосте, в том числе CM-server

ssh-copy-id fqdn\_hostname

Copy

После этого на всех хостах в файле ***/home/ansible/.ssh/authorized\_keys*** появится запись открытой части ключа пользователя **ansible**со стороны CM-server.

7. На всех хостах разрешить ***sudo***без запроса пароля для пользователя **ansible**:

visudo # открываем конфигурационный файл “sudo” редактором vi

Copy

8. В конце файла находим по тексту строку:

root ALL=(ALL) ALL

Copy

9. Под ней добавляем строку:

ansible ALL=(ALL) NOPASSWD: ALL

Copy

10. Сохраняем изменения.

11. Предпочтительно использовать DNS интрасети, имеющий сетевую связность с подсетью контура и контроллером домена (в случае с kerberos), чтобы все машины кластера умели резолвить друг друга и DC.

**Важно:** Использование файла hosts вместо DNS допустимо только на тестовых кластерах. При использовании в продуктивном контуре для сохранения полной работоспособности рекомендуем использовать выделенный DNS-сервер, обеспечивающий прямой и обратный резолв.

  В противном случае в конец файла  ***/etc/hosts*** (не затирая существующие записи) добавить все машины кластера в следующем формате:

  Пример записей в файле **hosts**:

192.168.180.100 rt-cm.corp.local rt-cm # MyCluster RTCM docker-host server

192.168.180.101 rt-m1.corp.local rt-m1 # MyCluster m1

192.168.180.102 rt-m2.corp.local rt-m2 # MyCluster m2

192.168.180.103 rt-s1.corp.local rt-s1 # MyCluster s1

192.168.180.104 rt-s2.corp.local rt-s2 # MyCluster s2

192.168.180.105 rt-s3.corp.local rt-s3 # MyCluster s3

192.168.180.120 rt-dc.corp.local rt-dc DC # DC AD Kerberos principals

Copy

**Примечание:** Для правильной работы компонент необходимо чтобы в файле **hosts были**представлены как короткие (например, rt-cm), так и полные имена (например, rt-cm.corp.local) с указанием домена

12. После запуска контейнеров (см [п. 3.4 шаг 1](https://docs.data.rt.ru/%D0%94%D0%BE%D0%BA%D1%83%D0%BC%D0%B5%D0%BD%D1%82%D0%B0%D1%86%D0%B8%D1%8F/RT_ClusterManager/Installation_RT_ClusterManager#:~:text=1.%20%D0%98%D0%B7%20%D1%80%D0%B0%D0%B1%D0%BE%D1%87%D0%B5%D0%B9%20%D0%B4%D0%B8%D1%80%D0%B5%D0%BA%D1%82%D0%BE%D1%80%D0%B8%D0%B8%20%D0%B2%D1%8B%D0%BF%D0%BE%D0%BB%D0%BD%D0%B8%D1%82%D1%8C:)) необходимо скопировать ключевую пару пользователя **ansible**(id\_rsa, id\_rsa.pub)в необходимую директорию:

cp дир\_пользователя/.ssh/id\_rsa\* Рабочая\_директория/rtcm\_data/

Copy

  Например,

cp /home/ansible/.ssh/id\_\* /opt/rtcm/rtcm\_data/

Copy

13. Часовой пояс в контейнере **rtcm** должен соответствовать часовым поясам на других хостах кластера.

  Для изменения часового пояса в контейнере rtcm требуется внести необходимый часовой пояс в файле**docker-compose.yaml**в переменной **TZ**, находящейся в блокеservices → rtcm → environment

  Например, ***Europe/Moscow***.  
  
 Затем запустить контейнеры (см [п. 3.4 шаг 1](https://docs.data.rt.ru/%D0%94%D0%BE%D0%BA%D1%83%D0%BC%D0%B5%D0%BD%D1%82%D0%B0%D1%86%D0%B8%D1%8F/RT_ClusterManager/Installation_RT_ClusterManager#:~:text=1.%20%D0%98%D0%B7%20%D1%80%D0%B0%D0%B1%D0%BE%D1%87%D0%B5%D0%B9%20%D0%B4%D0%B8%D1%80%D0%B5%D0%BA%D1%82%D0%BE%D1%80%D0%B8%D0%B8%20%D0%B2%D1%8B%D0%BF%D0%BE%D0%BB%D0%BD%D0%B8%D1%82%D1%8C:))

### 2.3.1 Дополнительная настройка в ****Astra Linux**** для кластеров****RT.WereHouse****

**Примечание:** Для выполнения команд, на сервере должно быть установлено программное обеспечение для выполнения HTTP запросов, например, утилита командной строки **сurl**.

При использовании операционной системы **Astra Linux,** при использовании кластеров создаваемых с помощью плагина **RT.WereHouse** на всех хостах необходимо настроить***pam-kiosk*,** для чего:

Используя команду:

sudo pam-auth-update

Copy

В “*PAM profiles to enable*” выберите "Unix authentication":

pam-auth-update

debconf: не удалось инициализировать интерфейс: Dialog

debconf: (Ни одна из dialog-подобных программ не установлена, поэтому вы не можете использовать dialog-интерфейс. at /usr/share/perl5/Debconf/FrontEnd/Dialog.pm line 78.)

debconf: будет использован интерфейс: Readline

Настройка PAM

-------------

Через подключаемые модули аутентификации (PAM) указывается как нужно проводить аутентификацию, авторизацию и смену пароля в системе, а также можно

назначать запуск дополнительных действий при старте пользовательского сеанса.

Некоторые пакеты модулей PAM предоставляют профили, которые можно использовать для автоматического регулирования поведения всех использующих PAM программ

в системе. Выберите профили, которые нужно применить.

1. Cracklib password strength checking 5. Unix authentication

2. Kiosk2 user confinement 6. Register user sessions in the systemd control group hierarchy

3. pam tally 7. Create home directory on login

4. Last login inactivity days (Default 90 days) 8. ни один из предложенных выше

(Укажите цифры, соответствующие выбранным вариантам, разделяя их пробелами.)

Активируемые профили PAM: 5

Copy

Основные конфиги pam лежат в **/etc/pam.d/**

# 3. Установка RT.ClusterManager

**Примечание:** Установка **RT.ClusterManager** может быть выполнена в автоматическом режиме с помощью скрипта ***install-rtcm.sh****,*достаточно задать значения переменных в скрипте и запустить скрипт, как указано в [п. 5.2.1 документа “API для работы с RT.ClusterManager”](https://docs.data.rt.ru/%D0%94%D0%BE%D0%BA%D1%83%D0%BC%D0%B5%D0%BD%D1%82%D0%B0%D1%86%D0%B8%D1%8F/RT_ClusterManager/rt_clustermanager_API#:~:text=5.2.1%20%D0%97%D0%B0%D0%BF%D1%83%D1%81%D0%BA%20%D1%83%D1%81%D1%82%D0%B0%D0%BD%D0%BE%D0%B2%D0%BA%D0%B8%20RT.ClusterManager).

**Примечание:** Для управлением **RT.ClusterManager**, в том числе запуском, остановом, созданием резервной копии можно воспользоваться скриптом ***topgun.sh*** ([см. п. 6 документа “API для работы с RT.ClusterManager”](https://docs.data.rt.ru/%D0%94%D0%BE%D0%BA%D1%83%D0%BC%D0%B5%D0%BD%D1%82%D0%B0%D1%86%D0%B8%D1%8F/RT_ClusterManager/rt_clustermanager_API)).

Вне зависимости от типа репозитория перед началом выполнения всех работ необходимо убедиться, что в **Рабочую\_директорию** (в нашем случае «***/opt/rtcm***») скопированы (или созданы, если их нет в архиве) нужные директории и файлы из архивов репозитория:

1. blackbox/
2. config/
3. prometheus\_config/
4. prometheus\_data/
5. rtcm\_conf/
6. rtcm\_data/
7. rtcm\_db/
8. postgres\_data/
9. docker-compose.yaml

На все указанные директории с вложенными в них файлами должны быть предоставлены полные права:

chmod 777 -R blackbox

chmod 777 -R config

chmod 777 -R prometheus\_config

chmod 777 -R prometheus\_data

chmod 777 -R postgres\_data

chmod 777 -R rtcm\_conf

Copy

Данные команды необходимо выполнить, так как директории монтируются в контейнер, где файлы читает другой пользователь (пользователь контейнера).  
В противном случае необходимо создать вышеуказанного пользователя на машине с таким же UID как у пользователя в контейнере.

***Важно:****В файле****docker-compose.yaml****необходимо указать нужный часовой пояс*.

## 3.1 Установка и настройка докера CentOS/RHEL

Для установки Docker выполните команды на CM-server из **рабочей директории**:

1. Устанавливаем, запускаем и добавляем в автозапуск docker:

sudo yum install -y docker docker-compose

sudo systemctl start docker

sudo systemctl enable docker

Copy

2. Проверьте текущее состояние docker:

sudo systemctl status docker

Copy

В случае репозитория «из коробки» (скачивание пакетов через интернет) необходимо:

3. Создать файл **/etc/docker/daemon.json**:

{

"debug": true,

"registry-mirrors": [

"https://myrepo\_adress.ru",

"https://ip\_repo"

],

"insecure-registries" : ["myrepo\_adress.ru","ip\_repo"],

"experimental": false

}

Copy

***Примечание:****Если по дефолтному пути установки (****var/lib/docker****или****/opt/docker****) меньше 50 ГБ места, рекомендуется создать отдельный раздел для докера, например,****/docker****и добавить его использование в файл daemon.json "data-root":****"/docker****".*

4. Перезагрузите сервис:

sudo systemctl reload docker

Copy

## 3.2 Загрузка doсker-образов из архивов

Для загрузки docker-образов из архива можно воспользоваться двумя способами:

1. Запуск скрипта автоматической подгрузки образов:

./load\_images.sh

Copy

2. Загрузка образов вручную.  
Для этого на CM-server выполните следующие команды:

sudo docker load -i путь\_до\_дир\_с\_образами/образ\_докер-контейнера\_rtcm.tar.gz

sudo docker load -i путь\_до\_дир\_с\_образами/образ\_докер-контейнера\_postgres.tar.gz

sudo docker load -i путь\_до\_дир\_с\_образами/образ\_докер-контейнера\_prometheus.tar.gz

sudo docker load -i путь\_до\_дир\_с\_образами/образ\_докер-контейнера\_blackbox.tar.gz

Copy

## 3.3 Подготовка настроек

**Примечание:** На хосте c **RT.ClusterManager** убедитесь, что команда echo $HOSTNAME возвращает непустое значение - а именно DNS имя машины, которое резолвится с соседних хостов

Перед запуском **RT.ClusterManager** необходимо выполнить следующие действия:

1. Убедиться, что файл **docker-compose.yaml**находится в**рабочей директории CM-server.**
2. В файле **docker-compose.yaml** необходимо внести следующие изменения:

* В блоке **services→rtcm→environment**заполнить значения переменных \*\_REPO полными урлами репозиториев - эти строки в неизменном виде попадут в **.repo** файлы на узлах кластера.

     Если используется репозиторий РТК:

* При наличии в файле **docker-compose.yaml** переменных REPO\_USER, REPO\_PASS заполнить их данными учётной записи для подключения к репозиториям, а сами строки подключения к репозиториям заполнять без учётных данных

**Примечание:** Данные учетной записи должны быть указаны без кавычек

* При отсутствии этих переменных, указать учётные данные для подключения в самой строке подключения к репозиториям в формате https://USER:PASS@REPO\_URL (список см. ниже).

    Перечень переменных для заполнения:

* **Обязательные записи:**
  + PROMETHEUS\_REPO - путь до директории, в которой лежит нод экспортер прометея (если используется архив для оффлайн установки здесь указать путь до репозитория чекер агента).
  + CHECKERAGENT\_REPO - репозиторий rt.ca, пакеты поставляются вместе с CM.
  + CENTOS\_REPO - заполняется при наличии у заказчика собственного репозитория пакетов для centos, если его нет - оставить пустым.
* **Записи для отдельных продуктов**(строки неиспользуемых на данном кластере продуктов можно оставить исходными):
  + DATALAKE\_REPO
  + STREAMINGKAFKA\_REPO
  + STREAMINGNIFI\_REPO
  + WAREHOUSE\_REPO
  + WIDESTORE\_REPO

**ВНИМАНИЕ!** Если вы устанавливаете продукт на ОС Альт, то нужно разворачивать локальный репозиторий с пакетами по инструкции из раздела 2.2. этого документа (пакеты для репозитория поставляются в архиве продукта). Доступного в интернете репозитория РТК в формате Альт пока нет

Значения этих переменных при использовании репозитория РТК (логин и пароль указываются в отдельных переменных - см. выше):

| ***Имя переменной*** | ***Значение*** | ***Примечание*** |
| --- | --- | --- |
| PROMETHEUS\_REPO | https://repo.data.rt.ru/repository/rt.clustermanager/prometheus/ |  |
| CHECKERAGENT\_REPO | https://repo.data.rt.ru/repository/rt.checkeragent\_rpm\_stable/ |  |
| CENTOS\_REPO | https://mirror.yandex.ru/centos/7/os/x86\_64/  http://repo.red-soft.ru/redos/7.3c/x86\_64/os/ | можно использовать любую внешнюю или свою  Если установка производится на хосты с RedOS использовать второе значение |
| DATALAKE\_REPO | https://repo.data.rt.ru/repository/rt.datalake\_distr3.0.0\_rpm\_stable/ https://repo.data.rt.ru/repository/rt.datalake\_distr3.2.2\_rpm\_stable/ | ссылка зависит от версии плагина |
| STREAMINGKAFKA\_REPO | https://repo.data.rt.ru/repository/rt.streamingkafka\_rpm\_stable/  https://repo.data.rt.ru/repository/rt.streamingkafka\_redos7.3\_rpm\_stable/ | для установки на ОС CentOS  для установки на ОС RedOS |
| STREAMINGNIFI\_REPO | https://repo.data.rt.ru/repository/rt.streamingnifi\_rpm\_stable/  https://repo.data.rt.ru/repository/rt.streamingnifi\_redos7.3\_rpm\_stable/ | для установки на ОС CentOS  для установки на ОС RedOS |
| WAREHOUSE\_REPO | https://repo.data.rt.ru/repository/rt.warehouse\_rpm/v1/ | для установки 6й версии |
| WAREHOUSE\_REPO | https://repo.data.rt.ru/repository/rt.warehouse\_rpm/v4/ | для установки 5й версии |
| WIDESTORE\_REPO | https://repo.data.rt.ru/repository/rt.widestore\_rpm/v1/ |  |

Порт графического интерфейса **RT.ClusterManager** не рекомендуется менять. Но в случае, если всё-таки требуется изменить порт по умолчанию, его необходимо вписать в двух местах:

* в блоке **services→rtcm→ports:**

    - "8080:8080"     (только в левой части изменить на требуемый. В правой части - порт внутри контейнера, его изменять не следует).

* в блоке **services→rtcm→environment:**

    - CM\_HOST=server*\_*name     (изменить адрес хоста на тот, на котором запущен **RT.ClusterManager**).  
    - CM\_PORT=8080     (изменить адрес порта на необходимый).

* В блоке **services→rtcm→extra\_hosts:**

    - При отсутствии DNS серверов можно прописать имена и адреса всех хостов (по аналогии с файлом **hosts**, редактируемым выше) в блоке extra\_hosts.

Например:

* “rt-m1.local.domain:192.168.150.15”
* “rt-m2.local.domain:192.168.150.16”

### 3.4 Запуск/перезапуск контейнера

Для запуска контейнеров с **RT.ClusterManager** необходимо выполнить следующие действия:

1. Из **рабочей директории**выполнить:

sudo docker-compose -p rtcm up -d --remove-orphans

Copy

*Имя группы контейнеров «rtcm» должно присутствовать. В эту группу входит три контейнера (одноимённый rtcm, blackbox и prometheus).*

2. Проверьте состояние контейнеров:

sudo docker ps

Copy

  В случае если контейнеры запущены – можно зайти в web-интерфейс для проверки **RT.ClusterManager** (см. [п. 3.5](https://docs.data.rt.ru/%D0%94%D0%BE%D0%BA%D1%83%D0%BC%D0%B5%D0%BD%D1%82%D0%B0%D1%86%D0%B8%D1%8F/RT_ClusterManager/rt_clustermanager_Installation#:~:text=3.5%20%D0%97%D0%B0%D0%BF%D1%83%D1%81%D0%BA)).

3. Дополнительные команды для работы с **docker**:

  Для остановки всех контейнеров необходимо использовать команду:

sudo docker-compose -p rtcm stop

Copy

  Для удаления всех контейнеров необходимо использовать команду:

sudo docker-compose -p rtcm down

Copy

  Для просмотра загруженных образов контейнеров (в том числе их id) необходимо использовать команду:

sudo docker images

Copy

  Для удаления образов контейнеров необходимо использовать команду:

sudo docker image rm [id\_контейнера]

Copy

  Для просмотра сетей докера необходимо использовать команду:

sudo docker network ls

Copy

***Примечание****: В случае перезапуска контейнеров, далее в веб-интерфейсе Системы необходимо выйти из учётной записи и обновить страницу.*

## 3.5 Запуск web-интерфейса RT.ClusterManager

1. Сервис будет доступен с любого компьютера с установленным браузером и имеющим маршруты до CM-server по hostname (если резолвиться по имени) или ip адресу по необходимому порту (по умолчанию 8080. Или любой другой, который задали перед запуском контейнеров в файле **docker-compose.yaml**).

  Например:

http://CM:8080

Copy

http://192.168.150.150:8080

Copy

**Примечание:** На данном этапе рекомендуем сделать снимки (снапшоты) **всех виртуальных машин** кластера на которые будет устанавливаться программное обеспечение, в том числе **RT.ClusterManager**.  
Это позволит сократить время при тестировании различных вариаций установки программного обеспечения благодаря отсутствию необходимости проведения этапов предварительной настройки, описанных в данном документе.

**ВНИМАНИЕ!**

пп.2 , 3 актуальны только для версии 2.7.0 и выше,

на версиях ниже логин rtcm@rt.ru пароль test.

2. Для безопасной инициализации нужно отредактировать файл ***single-user-providers.xml***расположенный по адресу: ***/opt/app/rtcm/conf:***

<?xml version="1.0" encoding="UTF-8" standalone="yes"?>

<loginIdentityProviders>

<provider>

<identifier>single-user</identifier>

<class&gt;single-user.class&lt;/class&gt;

<property name="Rtcm user">rtcm@rt.ru</property>

<property name="Password">test0000</property>

<property name="Secret key">super-secret-key</property>

</provider>

</loginIdentityProviders>

Copy

   В файле вы можете изменить значения:

* **Rtcm user** - логин для первого входа ,
* **Password**- пароль для авторизации,
* **Secret key** - alias ключа в хранилище RT.CM для расшифровки пароля.

3.  Первый вход выполняется с учётными данными указанными в файле ***single-user-providers.xml***

4. В браузере должна отобразиться страница первого входа  в Систему, с предложением создать администратора **RT.ClusterManager**:

Первая авторизация в Системе, создание пользователя с ролью “Супер администратор”

Заполните поля на форме и нажмите кнопку “Создать”. В Системе будет создан пользователь с ролью “Супер администратор”.

В дальнейшем, проходя авторизацию под созданным пользователем можно будет создавать других пользователей с ролями соответствующими их функциональным обязанностям. Подробнее о создании пользователей смотрите [п. 9.2 в документе “RT.ClusterManager. Руководство администратора”.](https://docs.data.rt.ru/%D0%94%D0%BE%D0%BA%D1%83%D0%BC%D0%B5%D0%BD%D1%82%D0%B0%D1%86%D0%B8%D1%8F/RT_ClusterManager/Vers_2/RT_ClusterManager_Admin_Guide_ver2#:~:text=9.2%20%D0%A1%D0%BE%D0%B7%D0%B4%D0%B0%D0%BD%D0%B8%D0%B5%20%D0%BD%D0%BE%D0%B2%D0%BE%D0%B3%D0%BE%20%D0%BF%D0%BE%D0%BB%D1%8C%D0%B7%D0%BE%D0%B2%D0%B0%D1%82%D0%B5%D0%BB%D1%8F)

**Примечание:**Поля “Имя” и “Фамилия” могут содержать до 16 символов русского или английского алфавита, а также символы точки и тире.

**Примечание:** Пароль пользователя должен содержать не менее 8 символов, должен состоять из латинских букв,  чисел от 0 до 9 и следующих спецсимволов  «!@-\_+#».

# 4. Автоматическая установка RT.ClusterManager

Поддерживаемые ОС:

* **CentOS 7.9** (Примечание: docker-compose будет скачен из интернета),
* **RedOS 7.3c,**
* **RedOS 7.3.x,**
* **RedOS 8.0,**
* **Alt Linux 10p.**

Перед установкой желательно обновить ОС:

dnf update

Copy

Если нет понимания в какой директории расположить **RT.ClusterManager**, то в соответствии с FHS (Filesystem Hierarchy Standard) логично расположить новое ПО **rtcm в /opt**:

cd /opt

Copy

**Скачиваем архив с Nexus сайта:**

https://repo.data.rt.ru/#browse/browse:rt.clustermanager\_arch

Copy

Можно использовать wget, curl (с ключами -L и -o) или любой удобный вам способ.

Распаковываем архив:

tar xvf RT.ClusterManager\_\*.gz

Copy

Переходим в директорию:

cd rtcm

Copy

**Примечание:**Если требуется можно донастроить параметры в**docker-compose.yaml**, перед началом установки

Запускаем установку (он установит docker, docker-compose, запустит docker, подгрузит docker-образы, пошлёт сигнал запуска на topgun.sh):

./install-rtcm.sh install

Copy

После этого приблизительно через 20 сек (если диск HDD, а не SSD, то чуть дольше) можно проверять web-интерфейс:

http://<server>:8080

Copy

**Примечание:**Стоит отметить, что если требуется вносить правки в docker-compose.yaml, то необходимо будет останавливать и запускать RTCM.

./topgun.sh down

./topgun.sh up

Copy