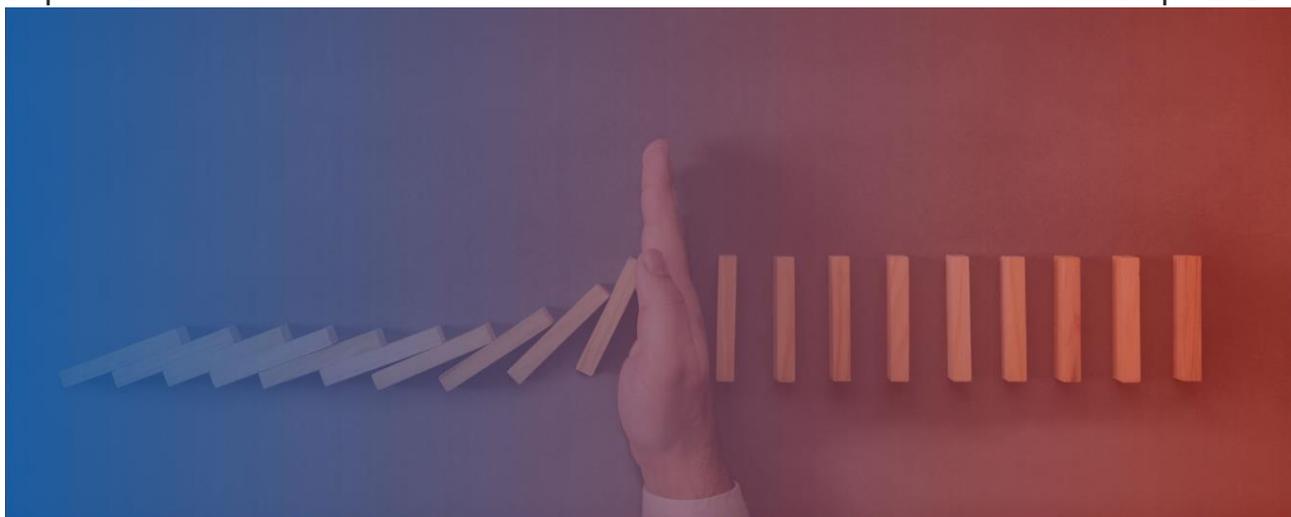


ООО «Дельта Солюшнс»

Руководство по эксплуатации *«Портал предоставления ресурсов Delta Cloud»*

Версия 1.0

Сентябрь 2023



Оглавление

1.	Введение	5
1.1.	Назначение Портала предоставления ресурсов	5
1.2.	Область применения Портала предоставления ресурсов.....	5
2.	Физическая и программная архитектура Портала предоставления ресурсов.....	6
2.1.	Структура Портала предоставления ресурсов.....	6
2.2.	Технологический стек	7
2.3.	Балансировщик запросов внешних систем	7
3.	Развертывание Портала предоставления ресурсов.....	8
3.1.	Подготовка к установке.....	8
3.2.	Процесс установки программы	17
4.	Схемы бизнес процессов системы	21
4.1.	Схема бизнес процесса	21
5.	Логирование системы.....	23
5.1.	Расположение log файлов и механизм логирования	23
6.	Ответы на часто задаваемые вопросы по эксплуатации системы.....	25

Лист изменений

Дата	Версия	Автор	Описание
24.11.2022	0.1	Стрельников Дмитрий	Создание
07.12.2022	0.2	Стрельников Дмитрий	Проработка комментариев
26.09.2023	1.0	Стрельников Дмитрий	Обновление

Список сокращений

KVM	Kernel-based Virtual Machine
ГИ	Графический интерфейс
ВМ	Виртуальная машина

1. Введение

1.1. Назначение Портала предоставления ресурсов

Портал предоставления ресурсов разработан для повышения удобства выделения ресурсов и мониторинга частной облачной инфраструктуры.

1.2. Область применения Портала предоставления ресурсов

Портал предоставления ресурсов предоставляет следующие возможности по управлению облачной инфраструктурой:

1. Формирование периодических отчетов о использовании ресурсов оборудованием;
2. Формирование периодических отчетов о заявках;
3. Создание виртуальных машин исходя из данных заявок;
4. Конфигурация созданных виртуальных машин;
5. Удаление заведенных в системе виртуальных машин;
6. Мониторинг использования зарезервированных ресурсов облачной инфраструктуры;
7. Заведение заявок на выделение ресурсов инфраструктуры под виртуальные машины;
8. Настраиваемое графическое отображение (дашборды) необходимых для отслеживания показателей облачной ИТ-инфраструктуры;
9. Ролевая модель с гибкой настройкой привилегий доступа.

2. Физическая и программная архитектура Портала предоставления ресурсов

2.1. Структура Портала предоставления ресурсов

В состав Портала предоставления ресурсов входят следующие подсистемы:

1. Подсистема ведения заявок на выделение ресурсов облачной инфраструктуры;
2. Подсистема управления ресурсами облачной инфраструктуры для создания, конфигурации и удаления виртуальных машин;
3. Подсистема мониторинга используемых ресурсов;
4. Подсистема отчетности;
5. Подсистема графического отображения информационных показателей и метрик;
6. Подсистема хранения данных:
 - 1) Подсистема хранения данных осуществляет хранение оперативных данных системы, данных для формирования отчетов, данных об аудите транзакций;
 - 2) Подсистема обеспечивает резервное копирование и сохранение данных встроенным функционалом поставляемого окружения;
 - 3) Подсистема обеспечивает хранение служебных сервисов, служебных данных, данных виртуальных машин.
7. Подсистема передачи данных, которая обеспечивает возможность передачи трафика данных внутри Портала ресурсов, а также для передачи данных во внешние информационные системы;
8. Ядро:
 - 1) consul;
 - 2) domain;
 - 3) factory;
 - 4) notifier;
 - 5) objectifier;
 - 6) tagger.
9. Подсистема Cloud (В составе Портала предоставления ресурсов), включающая следующие модули:
 - 1) cask – модуль сущностей системы;

- 2) porter;
 - 3) steward;
 - 4) ui – модуль графического интерфейса.
10. Балансировщик запросов внешних систем.

2.2. Технологический стек

В качестве технологических платформ баз данных используются СУБД с разными моделями структур данных:

1. С реляционной моделью:
 - ClickHouse.
2. С нереляционной моделью:
 - MongoDB.

В качестве облачного хранилища используется MinIO.

В качестве среды выполнения программного кода используется Deno.

Взаимодействие портала ресурсов с внешними системами осуществляется посредством использования архитектурного стиля REST с использованием стандартных возможностей протокола прикладного уровня HTTP (HyperText Transfer Protocol).

2.3. Балансировщик запросов внешних систем

В программном решении портала предоставления ресурсов предусмотрен программный балансировщик запросов от внешних систем, предотвращающий отказ одного из серверов развернутого решения. Данный компонент реализуется в виде системы обмена сообщениями NATS.

3. Развертывание Портала предоставления ресурсов

3.1. Подготовка к установке

Перед процедурами установки портала предоставления ресурсов, необходимо провести подготовку, включающую в себя следующие этапы:

1. Установка подходящей операционной системы;
2. Установка docker;
3. Установка docker compose plugin;
4. Установка обязательных переменных сред;
5. Установка необязательных переменных сред (опционально).

3.1.1 Установка подходящей операционной системы

Для развертывания ПО требуется соответствие следующей архитектуре: AMD64.

В качестве операционной системы необходимо соответствие следующим требованиям: ОС Linux.

Ядро операционной системы можно скачать по ссылке: <https://kernel.org/>.

По различным вопросам, касаемо ядра ОС – официальная документация: <https://docs.kernel.org/>.

Официальный сайт разработчиков: <https://www.linux.com/>.

Wiki по ОС: <https://www.wiki.kernel.org/>.

3.1.2 Установка Docker

Для работы портала предоставления ресурсов необходимо наличие docker. Для установки docker, необходимо перейти на официальный сайт: <https://www.docker.com/>. Далее необходимо выбрать подходящую операционную систему для установки (в данном случае LINUX). Для перехода на страницу загрузки дистрибутива под Linux-системы, необходимо нажать правой кнопкой мыши по иконке с названием Linux.



Рисунок 2. Окно загрузки Docker

В случае проблем с заходом на официальный сайт Docker, имеется возможность зайти напрямую на страницу загрузки под ОС Linux: <https://docs.docker.com/desktop/install/linux-install/>.

3.1.2.1 Требования к системе

Для успешной установки Docker Desktop, ваш хост Linux должен соответствовать следующим общим требованиям:

1. Поддержка 64-битного ядра и процессора для виртуализации.
2. Поддержка виртуализации KVM. Следуйте инструкциям поддержки виртуализации KVM, чтобы проверить, включены ли модули ядра KVM и как предоставить доступ к устройству kvm.
3. **QEMU должен быть версии 5.2 или новее.** Рекомендуется обновление до последней доступной версии.

4. Система инициализации systemd. Более подробно о system можно узнать по следующей ссылке:

[https://wiki.archlinux.org/title/Systemd_\(%D0%A0%D1%83%D1%81%D1%81%D0%BA%D0%B8%D0%B9\)](https://wiki.archlinux.org/title/Systemd_(%D0%A0%D1%83%D1%81%D1%81%D0%BA%D0%B8%D0%B9)).

5. Gnome или среда рабочего стола KDE.

Для многих дистрибутивов Linux среда Gnome не поддерживает значки в трее. Чтобы добавить поддержку значков в трее, вам необходимо установить расширение Gnome. Например, индикатор приложения .

6. Не менее 4 ГБ оперативной памяти.

7. Включите настройку сопоставления идентификаторов в пространствах имен пользователей. Инструкция по включению настройки сопоставления идентификаторов в пространствах имен находится по адресу: <https://docs.docker.com/desktop/faqs/linuxfaqs/#how-do-i-enable-file-sharing>.

Примечание:

Docker не поддерживает запуск Docker Desktop во вложенных сценариях виртуализации. Рекомендуется проверить поддержку запуска Docker Desktop для Linux в вашем дистрибутиве.

Docker предоставляет .deb пакеты .rpm из следующих дистрибутивов и архитектур Linux, представленных в таблице 1.

Таблица 1. Наличие .deb и .rpm пакетов в дистрибутивах платформ

Платформа	x86_64/AMD64
Ubuntu	Да
Debian	Да
Fedora	Да

Ссылки в таблице кликабельны. По нажатию на название платформы, произойдет открытие ссылки на загрузку .deb/ .rpm пакета, соответствующего выбранному дистрибутиву.

Docker Desktop для Linux запускает виртуальную машину (VM), поэтому требуется поддержка KVM.

3.1.2.2 Установка KVM

Модуль KVM должен загружаться автоматически, если хост поддерживает виртуализацию. Если это не произошло, вам необходимо загрузить модуль вручную. Для этого запустите:

```
$ modprobe kvm
```

Примечание:

В зависимости от процессора хост-машины должен быть загружен соответствующий модуль:

```
$ modprobe kvm_intel # Intel processors
```

```
$ modprobe kvm_amd # AMD processors
```

Если приведенные выше команды не работают, вы можете посмотреть диагностику, выполнив:

```
$ kvm-ok
```

Чтобы проверить, включены ли модули KVM, запустите:

```
$ lsmod | grep kvm
```

Установка прав пользователя KVM

Чтобы проверить право собственности на `/dev/kvm`, запустите:

```
$ ls -al /dev/kvm
```

Добавьте своего пользователя в группу `kvm`, чтобы получить доступ к устройству `kvm`:

```
$ sudo usermod -aG kvm $USER
```

Выйдите из системы и войдите снова, чтобы состояние вашей УЗ в группе было обновлено.

Общие шаги установки

Для просмотра документации по Docker, необходимо перейти по ссылке: <https://docs.docker.com/>.

Примечание:

Убедитесь, что ваша система с Linux соответствует системным требованиям, изложенным выше, и выполнены все предварительные требования для конкретного дистрибутива.

1. Загрузите правильный пакет для вашего дистрибутива Linux и установите его с помощью соответствующего менеджера пакетов. (Для Debian, Fedora и Ubuntu данные пакеты представлены в таблице 1 (ссылки кликабельны)). Для Archlinux пакет можно скачать по следующей ссылке: <https://docs.docker.com/desktop/install/archlinux/>.

2. Откройте меню «**Приложения**» в Gnome/KDE Desktop и найдите **Docker Desktop**.

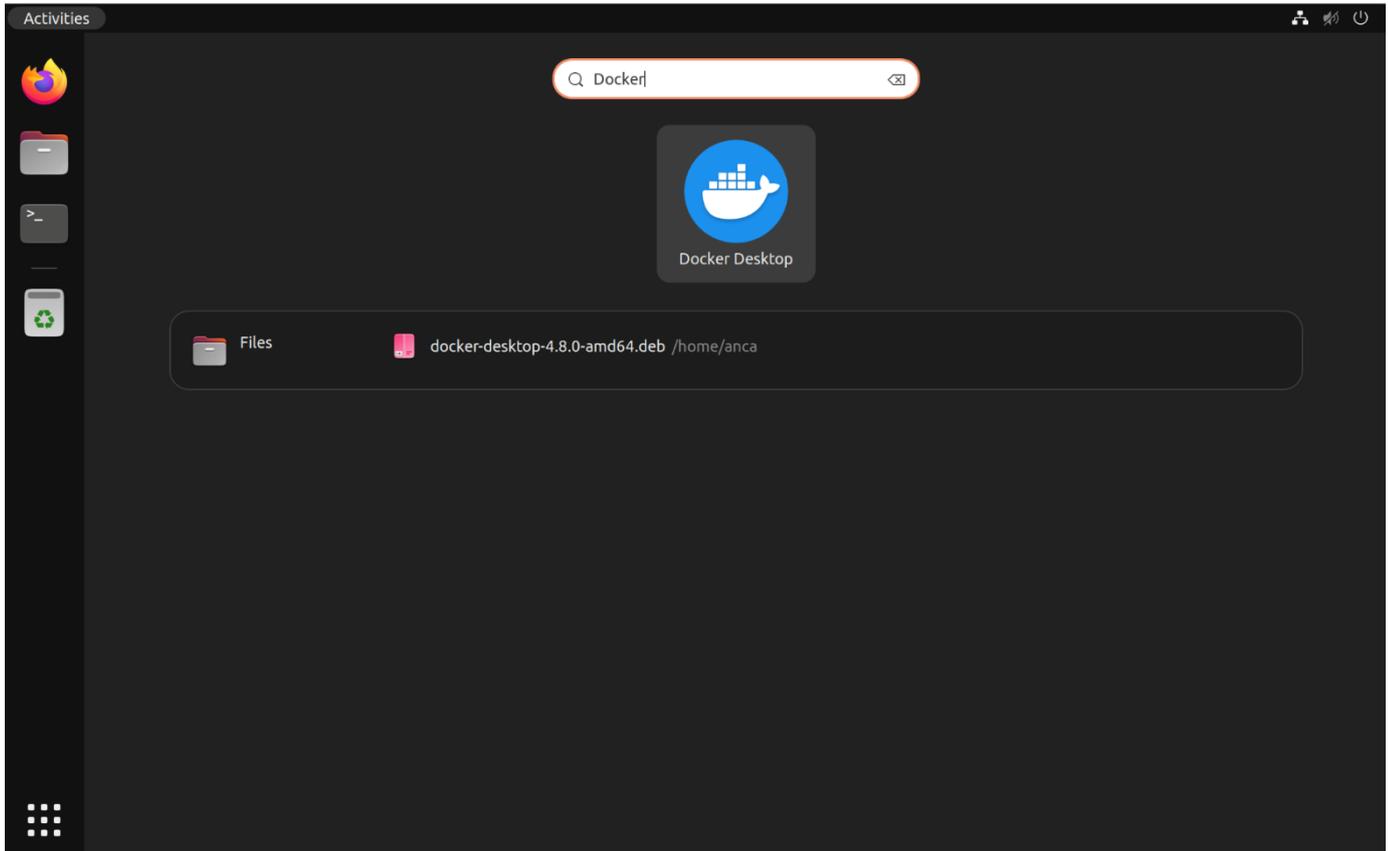


Рисунок 3. Поиск Docker Desktop в GNU Linux

3. Нажмите на Docker Desktop, чтобы запустить Docker. В меню Docker появится окно «Соглашения об обслуживании подписки Docker».

4. Нажмите «Принять», чтобы продолжить. Docker Desktop запускается после того, как вы принимаете условия. Если вы не согласитесь с условиями использования и не нажмете «Принять», приложение Docker Desktop закроется, и вы больше не сможете запускать Docker Desktop на своем компьютере. Вы можете принять условия позже, открыв Docker Desktop.

При возникновении проблем с установкой, вы можете посетить <https://docs.docker.com/desktop/troubleshoot/overview/>, где описаны ответы на распространенные проблемы, обходные пути, способы запуска и отправки диагностики, а также отправка проблем.

Общие рекомендации по работе с Docker приведены по ссылке: <https://docs.docker.com/get-started/>.

Инструкции по резервному копированию и восстановлению данных находятся по адресу: <https://docs.docker.com/desktop/backup-and-restore/>.

3.1.3 Установка docker compose plugin

При установке Docker Desktop, происходит автоматическая установка всех необходимых плагинов, включая Compose.

Чтобы проверить, установлен ли у вас Compose, откройте «About Docker Desktop» в меню Docker.

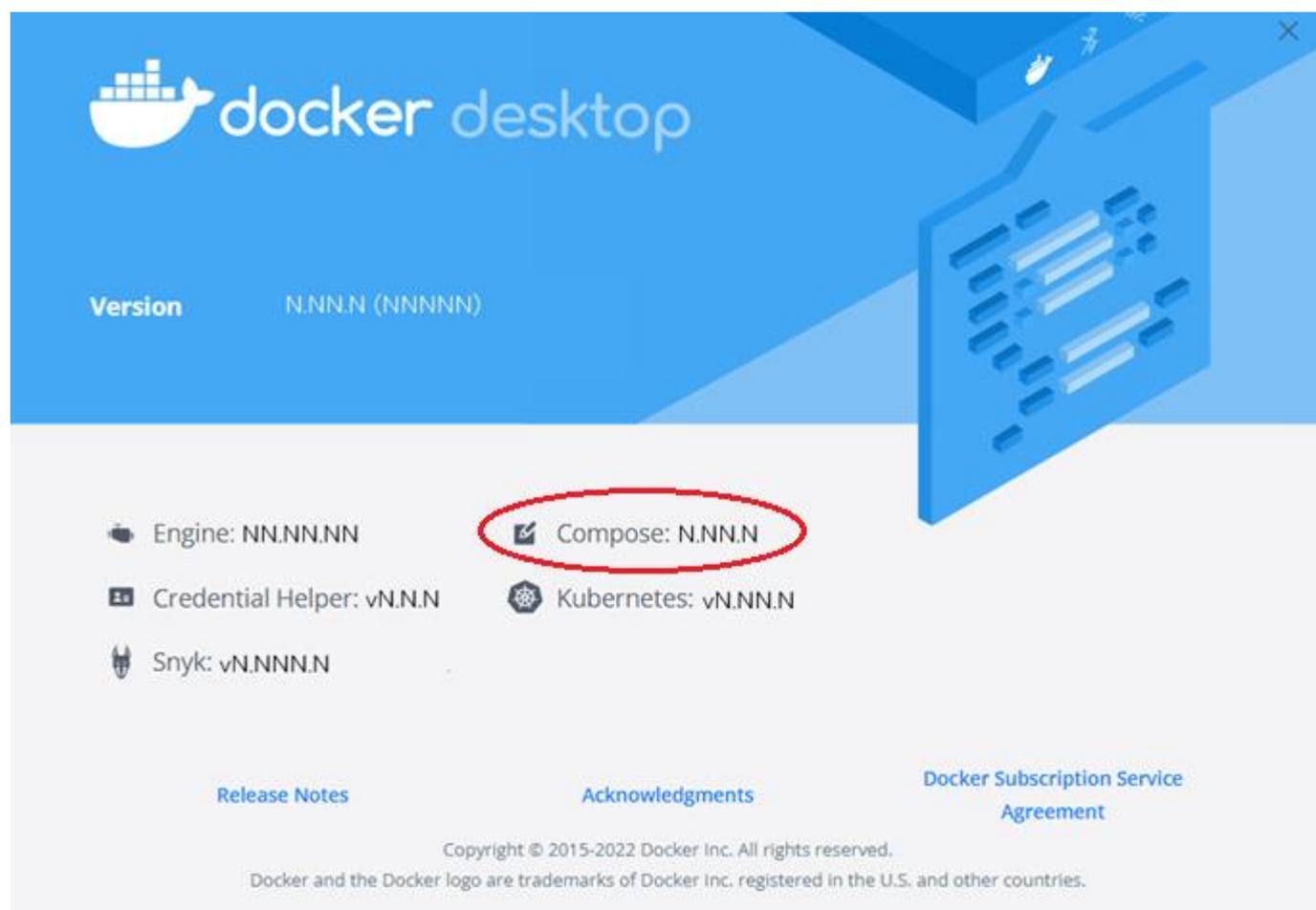


Рисунок 4. Окно About Docker Desktop

Если у вас еще не установлен Docker Desktop, установите данное ПО, следуя по шагам из пункта 2.1.2 (Установка docker).

3.1.3.1 Установка плагина compose при установленном Docker

Если у вас есть установка Docker в Linux, в которую вы хотите добавить плагин Compose, вы можете выбрать один из следующих вариантов:

- Установить с помощью репозитория
- Установить плагин вручную

Установка с помощью репозитория

Примечание:

Для установки плагина с помощью репозитория, у вас должен быть установлен Docker Engine и Docker CLI.

1) Настройте репозиторий. Инструкции для конкретного дистрибутива находятся по следующим ссылкам (кликабельно):

Ubuntu | CentOS | Debian | Fedora | RHEL | SLES.

2) Обновите индекс пакета и установите последнюю версию плагина Docker Compose:

- Ubuntu, Debian:

```
$ sudo apt-get update
$ sudo apt-get install docker-compose-plugin
```

- Основанные на RPM дистрибутивы:

```
$ sudo yum update
$ sudo yum install docker-compose-plugin
```

3) Убедитесь, что плагин Docker Compose установлен корректно, путем проверки версии следующей командой:

```
$ docker compose version
```

После выполнения команды должно появиться:

Docker Compose version vN.N.N ,

Где vN.N.N – текущая последняя версия плагина.

Установка плагина вручную

Примечание:

Этот вариант требует, чтобы вы управляли обновлениями вручную. Мы рекомендуем настроить репозиторий Docker для упрощения обслуживания.

1. Чтобы загрузить и установить плагин Compose CLI, запустите:

```
$ DOCKER_CONFIG=${DOCKER_CONFIG:-$HOME/.docker}
$ mkdir -p $DOCKER_CONFIG/cli-plugins
$ curl -SL
https://github.com/docker/compose/releases/download/v2.11.2/docker-
compose-linux-x86_64 -o $DOCKER_CONFIG/cli-plugins/docker-compose
```

После ввода, данная команда загружает последний выпуск Docker Compose (из репозитория

выпусков Compose) и устанавливает Compose для активного пользователя в \$HOME каталоге.

Для установки:

- Docker Compose для всех пользователей вашей системы, замените `~/.docker/cli-plugins` на `/usr/local/lib/docker/cli-plugins`.
- Другой версии Compose, замените v2.11.2 (последняя на момент написания настоящего документа) ее версией Compose, которую вы хотите использовать.

2. Примените исполняемые разрешения к двоичному файлу:

```
$ chmod +x $DOCKER_CONFIG/cli-plugins/docker-compose
```

или, если вы решили установить Compose для всех пользователей:

```
$ sudo chmod +x /usr/local/lib/docker/cli-plugins/docker-compose
```

3. Протестируйте установку, путем ввода следующих команд:

```
$ docker compose version
```

После выполнения команды должно появиться:

```
Docker Compose version vN.N.N,
```

Где vN.N.N – текущая последняя версия плагина.

Альтернативная установка (Автономный плагин Compose Standalone)

Примечание:

Обратите внимание, что в автономном режиме Compose используется синтаксис составления тире вместо стандартного синтаксиса текущего (пробела).

1. Чтобы скачать и установить Compose Standalone, запустите:

```
$ curl -SL  
https://github.com/docker/compose/releases/download/v2.11.2/docker-  
compose-linux-x86_64 -o /usr/local/bin/docker-compose
```

2. Примените разрешения на выполнение к автономному двоичному файлу в целевом пути для установки.

3. Проверьте корректность установки Compose с помощью команды `docker-compose`.

Обновление плагина Compose

Если у вас уже установлен плагин compose, и требуется его обновить, то необходимо выполнить следующие команды, исходя из вашего дистрибутива:

- Ubuntu, Debian:

```
$ sudo apt-get update
```

```
$ sudo apt-get install docker-compose-plugin
```

- Основанные на RPM дистрибутивы:

```
$ sudo yum update
```

```
$ sudo yum install docker-compose-plugin
```

3.1.4 Переменные среды

Обязательные:

- LICENSE – строка лицензии для работы системы в неограниченном режиме;
- JETSTREAM_DATA_DIR – директория для данных JetStream (NATS);
- DB_DATA_DIR – директория для данных базы данных (MongoDB);
- TSDB_DATA_DIR – директория для БД временных рядов (ClickHouse);
- S3_DATA_DIR – директория для S3-совместимого облачного хранилища (MinIO).

Оptionальные:

- DB_ROOT_USERNAME (root) – переменная среда имени пользователя БД (рут);
- DB_ROOT_PASSWORD (root) – переменная среда пароля пользователя БД (рут);
- TSDB_ROOT_USERNAME (root) – переменная среда имени пользователя БД временных рядов(рут);
- TSDB_ROOT_PASSWORD (root) – переменная среда пароля пользователя БД временных рядов(рут);
- S3_ROOT_USERNAME (root) – переменная среда имени пользователя облачного хранилища (рут);
- S3_ROOT_PASSWORD (root1234) – переменная среда пароля пользователя облачного хранилища (рут);
- KERBEROS_ACTIVE – необходимо установить "1" для включения Kerberos авторизации;
- KERBEROS_SERVER_NAME – имя сервера который будет использоваться для доступа к KDC;
- KERBEROS_CONF_B64 – Kerberos конфиг (krb5.conf) для сервера, кодировка base64;
- KERBEROS_KEYTAB_B64 – Kerberos keytab (krb5.keytab) для сервера, кодировка base64;
- REPLICAS_OVERLOAD – установите "1" для запрета репликации служб по умолчанию.

3.2. Процесс установки программы

Для установки портала предоставления ресурсов необходимо, чтобы хост, на котором будет разворачиваться ПО, соответствовала всем требованиям, описанным в пункте 2.1 (Подготовка к установке), включая установленное дополнительное ПО, а также следующим:

Таблица 2. Требования к хосту

	Минимальные требования	Рекомендуемые требования
Операционная система	Любой linux с поддержкой x64 архитектуры, поддерживаемый нужными версиями docker-окружения	
Docker engine	v20.10.0	Последняя версия
Docker compose	v20.10.0	Последняя версия
CPU	6 vCPU	12 vCPU или больше
RAM	6 GB	12 GB или больше
HDD	10 GB	20 GB или больше

1) Загрузите артефакты (пакеты) релизной версии ПО (в случае онлайн установки). В случае оффлайн установки, вам необходим готовый установочный пакет ПО (архив). Ссылка на онлайн скачивание пакета релизной версии ПО или сам пакет ПО высылаются по заявке через портал <https://deltasolutions.ru/contacts?target=software-request>.

Архив релизной версии ПО в обоих случаях должен содержать следующие файлы:

- `images.tar` – архивные образы для docker;
- `compose.yml` – compose конфигурация;
- `version` – файл, содержащий строку с версией.

2) Скопируйте `images.tar` и `compose.yml` в любую пустую папку на целевом хосте.

3) В выбранной папке, выполните следующие команды:

Загрузите образы контейнеров Портала предоставления ресурсов:

```
$ docker load --input images.tar
```

Примечание:

Если ваш сервер не подключен к Интернету, вам также придется загружать внешние (не связанные напрямую с порталом предоставления ресурсов) зависимости вручную с помощью `docker load` команды.

Обновите конфигурацию compose:

```
$ docker compose -p cloud up -d --remove-orphans
```

3.2.1 Конфигурация портов

1. Порты инструментария:
 - 31000 — примитивный просмотрщик логов и метрик контейнера.
2. Сервисные порты:
 - 32000 – точка входа NATS TCP;
 - 32010 — первичная точка входа в БД (MongoDB);
 - 32020 — точка входа HTTP API базы данных временных рядов (ClickHouse);
 - 32030 — точка входа HTTP API для S3-совместимого хранилища (MinIO);
 - 32031 — S3-совместимая административная панель хранилища.
3. Основные порты проекта:
 - 33000 — точка входа HTTP API;
 - 33010 – основной пользовательский интерфейс (GUI).

3.2.2 Настройка аутентификации с помощью Kerberos

Перед настройкой аутентификации с помощью Kerberos, должны быть настроены опциональные переменные среды Kerberos, указанные в пункте 2.1.4 (Переменные среды).

Чтобы включить аутентификацию на базе Kerberos, необходимо:

1. Добавьте целевой хост-компьютер в центр распределения ключей (KDC).
2. Установите соответствующую переменную среды, чтобы включить аутентификацию Kerberos (см. 2.1.4 Переменные среды).
3. Установите соответствующую переменную среды для имени сервера Kerberos, которое будет использоваться при доступе к KDC изнутри контейнера. Эта переменная должна содержать имя целевого хоста. Скорее всего, он должен быть в следующем формате:
`HTTP@<hostname>`.
4. Создайте файл конфигурации Kerberos, закодируйте его в кодировке base64 и установите соответствующую переменную среды.

Примечание :

Обратите внимание, что KDC должен быть доступен изнутри контейнеров, поэтому использование имен хостов, указанных только в `/etc/hosts` целевом хосте, запрещено. Тем не

менее, вам нужно либо использовать имена хостов, полученные через DHCP, либо использовать IP-адрес.

Чтобы проверить доступность KDC, вы можете выполнить выполнение в `core-domain` контейнере.

5. Сгенерируйте файл keytab Kerberos для целевого хоста, закодируйте его в кодировке base64 и установите соответствующую переменную среды.

Примечание :

В Windows нет встроенных средств для просмотра содержимого keytab файла. Но если, у вас а компьютере установлена версия Java JRE, вы можете воспользоваться утилитой klist.exe, которая входит в комплект java. Обратите внимание, что обычно keytab-файл содержит несколько непечатаемых символов, и их копирование из консоли может привести к различным ошибкам.

Рекомендация:

Рекомендуется выполнять кодировку base64 непосредственно на сервере KDC с помощью `cat data.keytab | base64` команды — не забудьте удалить переносы строк.

3.2.3 Инициализация пользователей

Чтобы не создавать пользователей по умолчанию вручную, вы можете использовать возможности, предоставляемые NATS для вызова скрипта инициализации пользователей. Чтобы выполнить скрипт инициализации пользователей, вам необходимо иметь клиент, например NATS CLI. Если у вас уже имеется данный инструментарий, вы сможете сделать специальный запрос на выполнение скрипта инициализации пользователей:

```
nats -s $NATS_URL request SCRIPTER.REQUEST.INIT_USERS ""
```

— где `$NATS_URL` это URL адрес сервера NATS. Ознакомьтесь с разделом «Открытые порты» для получения дополнительной информации о его местонахождении.

Если у вас не установлен NATS CLI, то вы можете установить корректный .deb файл на вашу рабочую машину по этой ссылке: <https://github.com/nats-io/natscli/releases>.

Если ваша рабочая машина имеет процессор от компании Intel, то вам необходима версия nats-X.Y.Z-amd64.deb (где X.Y.Z — номер версии релиза NATS). Затем, вам необходимо выполнить следующую команду, для установки данного файла:

```
sudo dpkg -i nats-X.Y.Z-amd64.deb
```

Также вы можете воспользоваться `yay` менеджером пакетов:

```
yay natscli
```

В случае возникновения вопросов по установке или работе с NATS CLI, вы можете ознакомиться с базовой информацией о NATS по ссылке: https://docs.nats.io/using-nats/nats-tools/nats_cli.

4. Схемы бизнес процессов системы

4.1. Схема бизнес процесса

Схема бизнес-процесса, реализуемая в рамках проекта, представлена на рисунке 5.

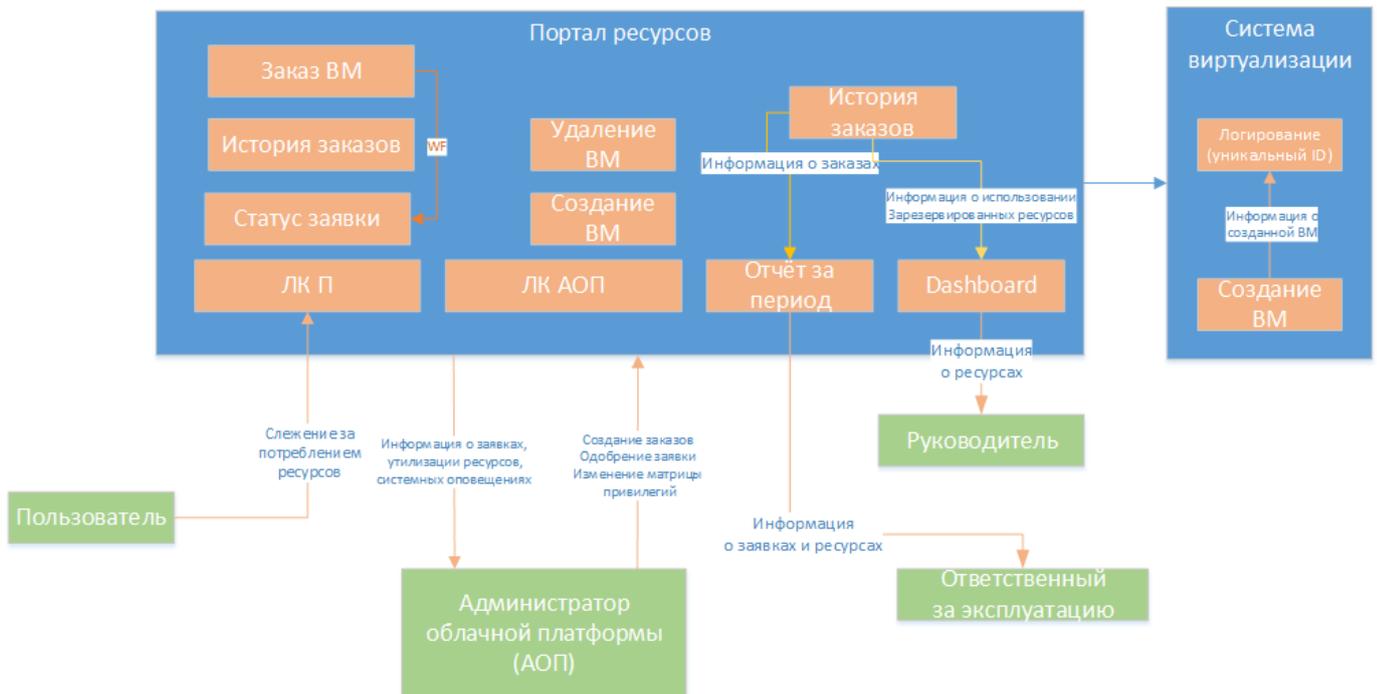


Рисунок 5. Схема процесса в рамках первой части проекта

В рамках бизнес-процесса реализовано:

1. Возможность авторизации (учетным данным пользователя или по SSO) как под учетной записью пользователя или администратора;
2. Создание виртуальных машин с резервированием ресурсов облачной инфраструктуры (посредством интеграции с Системой виртуализации);
3. Конфигурация созданных виртуальным машин (посредством интеграции с Системой виртуализации);
4. Удаление созданных виртуальных машин (посредством интеграции с Системой виртуализации);
5. Получение информационных системных оповещений;
6. Создание и редактирование заказов на выделение ресурсов для VM администратором облачной платформы;

7. Мониторинг выделенных/используемых ресурсов и статусов заказов;
8. Просмотр информации о использовании зарезервированных ресурсов облачной инфраструктуры;
9. Выгрузка отчетов за период по заявкам;
10. Создание, изменение и удаление учетных записей пользователей;
11. Настройка ролевой политики для учетных записей.

5. Логирование системы

5.1. Расположение log файлов и механизм логирования

Гибкий модуль логирования обеспечивает предоставление логов в различных форматах. Для пользователей предусмотрены настройки логируемых событий. По умолчанию в логи записываются различные системные состояния и события.

Для получения логов работы системы, можно использовать стандартный функционал Docker.

Получение журналов контейнера производится с помощью команды:

```
$ docker logs [OPTIONS] CONTAINER
```

Где [OPTIONS] возможны следующие:

Таблица 1. Опции

Опция	Описание
<code>--details</code>	Показать дополнительные сведения, указанные в журналах
<code>--follow, -f</code>	Следить за выводом журнала
<code>--since</code>	Показать журналы с отметкой времени (например, 2013-01-02T13:23:37Z) или относительной (например, 42 минуты за 42 минуты)
<code>--tail, -n</code>	Количество строк, отображаемых с конца журналов. (По умолчанию – все)
<code>--timestamps, -t</code>	Показать временные метки
<code>--until</code>	Показывать журналы перед отметкой времени (например, 2013-01-02T13:23:37Z) или относительной (например, 42 минуты за 42 минуты)

Команда `docker logs` пакетно извлекает журналы, существующие на момент выполнения.

Примечание:

Эта команда работает только для контейнеров, которые запускаются с помощью драйвера ведения журнала `json-file` или `.journald`

Команда `docker logs --follow` продолжит потоковую передачу новых выходных данных службы `STDOUT` и `STDERR`.

Передача отрицательного числа или нецелого числа `--tail` недействительна, и в этом случае устанавливается значение `all`.

Команда `docker logs --timestamps` добавит временную метку `RFC3339Nano`,

например 2014-09-16T06:17:46.000000000Z, к каждой записи журнала. Чтобы убедиться, что временные метки выровнены, наносекундная часть временной метки при необходимости будет дополнена нулем.

Команда `docker logs --details` добавит дополнительные атрибуты, такие как переменные среды и метки, предоставленные `--log-opt` при создании контейнера.

Параметр `--since` показывает только журналы контейнера, созданные после указанной даты. Вы можете указать дату в формате RFC 3339, метку времени UNIX или строку продолжительности Go (например `1m30s`, `3h`). Помимо формата даты RFC3339, вы также можете использовать RFC3339Nano, `2006-01-02T15:04:05`, `2006-01-02T15:04:05.999999999`, `2006-01-02Z07:00` и `2006-01-02`. Местный часовой пояс на клиенте будет использоваться, если вы не укажете Z смещение `+00:00` часового пояса в конце метки времени. При предоставлении меток времени Unix введите секунды `[.nanoseconds]`, где секунды — это количество секунд, прошедших с 1 января 1970 года (полночь UTC/GMT), не считая високосных секунд (также известных как эпоха Unix или время Unix), и необязательный параметр. Поле `nanoseconds` представляет собой долю секунды длиной не более девяти цифр. Вы можете комбинировать `--since` вариант с одним или обоими из `--follow` или `--tail` вариантов.

6. Ответы на часто задаваемые вопросы по эксплуатации системы

1. Ошибка при попытке авторизации на портале предоставления ресурсов

В том случае, если возникает ошибка при попытке авторизации в системе, необходимо проверить возможность доступа учетной записи к portalу предоставления ресурсов, связавшись с администратором портала.

2. Ошибка проверки номера при создании заявки на портале предоставления ресурсов

В системе предусмотрена валидация ввода по номеру заявки при заполнении поля «Номер заявки». Номер заявки представлен в виде «RP*****», где RP- наименование заявки, а *- цифры номера заявки. В данный момент на поле «Номер заявки» присутствует валидация на 7 цифр после RP. Если системой показывается ошибка валидации, проверьте вводимые данные и количество цифр номера заявки.

3. Какой алгоритм просмотра статуса выполненных задач/заявок?

Каждая задача имеет статус, который находится слева от задачи в виде иконки соответствующего цвета. В данный момент существуют 3 статуса: в процессе (иконка часов желтым цветом), завершена  и ошибка . Каждый статус присваивается задаче в зависимости от этапа, который проходит задача. Общий статус по заявке складывается из статусов задач. Если все задачи в рамках заявки имеют статус «Завершена», то заявка автоматически приобретает статус «Завершена». Если хотя бы одна задача имеет статус «В процессе» или «Ошибка», заявка будет находиться в статусе «В процессе».

4. Какой алгоритм заведения задачи в уже открытой заявке

Чтобы завести задачу в открытой заявке необходимо перейти в необходимую задачу на вкладке «Заявки», перейти в пункт «Создать задачу», а далее выбрать необходимую операцию и создать задачу. Более подробное описание алгоритма создания задачи в заявке можно найти в документации администратора «Delta Cloud Руководство администратора» в пункте 5.2.1 «Создание и изменение заявки».

5. Как система информирует пользователя о том, что задача в заявке не выполнена?

Понять текущее состояние задачи поможет статус задачи. Каждая задача имеет несколько статусов. Если задача в заявке не выполнена, ее статус будет иметь иконку , что будет обозначать, что задача не была завершена успешно. В случае неудачного завершения задачи, необходимо

посмотреть на причину ошибки в заявке и устранить ее, после чего попробовать заново создать новую задачу.