Инструкция по установке сервера dcomms

Введение

Что такое dcomms

dcomms (decentralized communications) это преднастроенный набор Docker-контейнеров децентрализованных федеративных приложений для коммуникаций, файлов конфигурации и скриптов управления для быстрого и простого разворачивания этих приложений на собственных серверах с помощью технологии Docker Swarm.

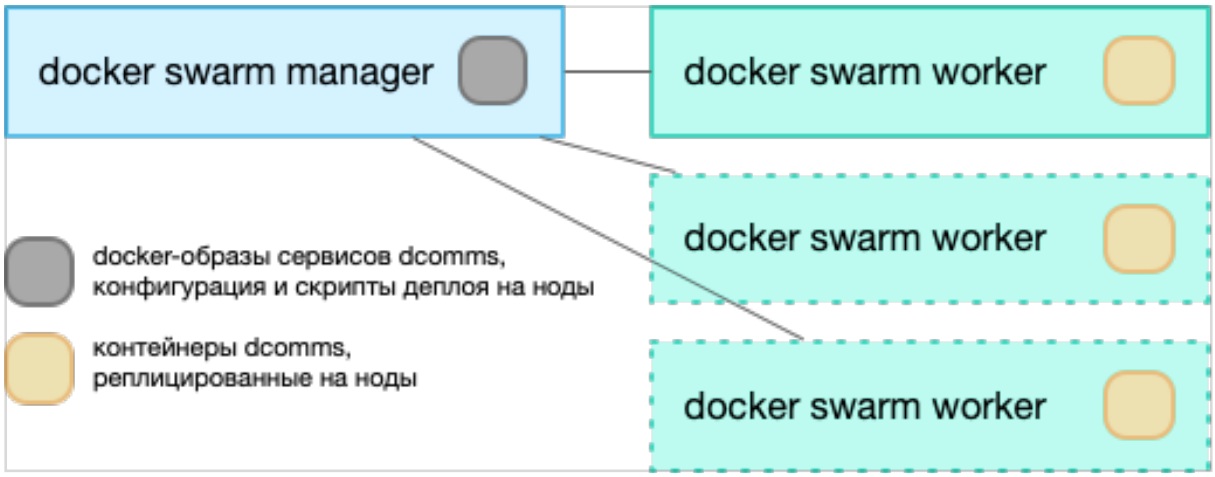
Децентрализованные приложения, входящие в набор dcomms, позволят вашим пользователям коммуницировать как при наличии подключения к Интернету, так и без него.

В качестве примера предлагаем посмотреть на сервис Chat v.3: https://moscow.chatv3.ru (названный так в честь Internet 3.0), построенный с помощью dcomms. Сервера Chat v.3 расположены в нескольких городах России, на каждом сервере идентичный набор Docker-контейнеров с децентрализованными приложениями. Также есть отдельный управляющий сервер, через который с помощью технологии Docker Swarm осуществляется деплой системы.

Используя репозиторий dcomms вы можете без проблем запустить сервис, аналогичный Chat v.3, внутри своей инфраструктуры и на собственном домене. С помощью Docker Swarm и инструментов dcomms вы сможете развернуть любое количество серверов и сабдоменов для вашей децентрализованной сети.

Как это работает

Для разворачивания собственного сервиса для децентрализованных коммуникаций используется система оркестровки Docker-контейнерами Docker Swarm. Сервер Docker Swarm с ролью manager предназначен для репликации dcomms-контейнеров на серверы с ролью worker, как изображено на схеме ниже.



Набор скриптов управления делает всю работу автоматически: вам необходимо только склонировать репозиторий dcomms к себе на сервер, внести правки в файлы конфигурации и указать доменное имя.
Для всех ресурсов будут автоматически выпущены Let's Encrypt TLS сертификаты, а доступ к ним будет предоставлен через веб-сервер Caddy.
После репликации контейнеров ваш собственный сервис для децентрализованной коммуникации будет сразу готов к работе.
Смотрите более подробно о том, какие приложения и компоненты включены в состав dcomms, а также пошаговую инструкцию по установке в следующих разделах.

Вы можете использовать несколько серверов dcomms для того, чтобы работоспособность была сохранена даже в том случае, если какие-то из серверов окажутся недоступными. Федеративные децентрализованные сервисы устроены таким образом, что пользователи, зарегистрированные на одном сервере, могут соединяться и коммуницировать с пользователями, зарегистрированными на другом сервере.

Что входит в состав dcomms

dcomms основывается на нереплицированных контейнерах, построенных на образах последних версий следующих приложений:

- CENO client, поставляемый censorship.no
- Synapse Docker, поставляемый matrix.org, ссылка на github
- Element, поставляемый vector-im, ссылка на github
- Caddy, поставляемый Caddy Docker Maintainers
- docker-mailadm, включающий в себя dovecot и postfix, поставляемый Delta Chat

Разворачивание своего сервера dcomms

Что для этого необходимо

- SSH root-доступ на 2 сервера: один для Docker Swarm с ролью manager, второй для роли worker для непосредственного запуска приложений. Минимальные системные требования для роли manager: 1 vCPU, 1GB RAM; для роли worker: 2 vCPU, 4GB RAM; ОС Linux 64 bit с ядром 3.10 или выше. Существует возможность запуска и на одном сервере, это описано в разделе "Установка dcomms на одном сервере".
- Доменное имя.
- Доступ к управлению A-записями и MX-записями вашего доменного имени.
- Публичный IP-адрес, если сервис публичный и доступ к нему должен быть возможен через Интернет.
- Компетенции по базовому администрированию Linux и Docker.

Установка

Установка и настройка с помощью docker swarm

Настройка доменов

Добавьте следующие A-записи для вашего доменного имени:

- `server1.example.com` ➜ IP адрес сервера с ролью worker. В качестве `server1` вы можете использовать любое имя, здесь это указано для примера, вместо `example.com` также имеется ввиду ваше доменное имя. В случае разворачивания сервиса на нескольких нодах в качестве `server1` можно использовать, например, название города или локации, в которой расположен соответствующий сервер.
- `matrix.server1.example.com` ➜ IP адрес сервера с ролью worker (для сервера Matrix).
- `chat.server1.example.com` ➜ IP адрес сервера с ролью worker (для веб-клиента Element чата Matrix).

Добавьте MX-запись для работы Delta Chat:

- `server1.example.com` ➜ `server1.example.com`

Установка и разворачивание сервиса dcomms

Подготовка сервера Docker Swarm manager

На сервере с ролью manager необходимо выполнить следующие шаги:

1. Установите Docker по официальному мануалу.
2. Выберите и создайте директорию, в которой будут храниться файлы dcomms и конфигурации. Например, `/usr/share/dcomms`:

```
mkdir -p /usr/share/dcomms
cd /usr/share/dcomms
```

2. Создайте пользователя и группу `eq`:

```
adduser --system eq
addgroup --system eq
adduser eq eq
```

3. Склонируйте или скачайте репозиторий dcomms:

```
git clone https://github.com/censorship-no/dcomms/
```

4. Выполните инициализацию режима Docker Swarm:

```
docker swarm init
```

После выполнения этой команды будет выведена строка подключения worker-нод к manager-ноде, например:

```
docker swarm join --token SWMTKN-1-
hbwgp1uosq68elf6fantcns4ivsmbusiepzftztetklggza0mn-0c8idote4jrf77s198hcd6uuf
X.X.X.X:2377
```

Сохраните эту команду, она понадобится позже.

Подготовка сервера Docker Swarm worker

На сервере с ролью worker необходимо выполнить следующие шаги:

1. Установите Docker по официальному мануалу.
2. Подключите worker-ноду к manager-ноде, используя сохраненную ранее команду `docker swarm join`:

```
docker swarm join --token SWMTKN-1-
hbwgp1uosq68elf6fantcns4ivsmbusiepzftztetklggza0mn-0c8idote4jrf77s198hcd6uuf
X.X.X.X:2377
```

3. Создайте директорию `/var/www/dcomms` (это необходимо для работы веб-сервера Caddy):

```
mkdir -p /var/www/dcomms
```

Шаги выше необходимо сделать для каждой worker-ноды в случае, если их несколько.

Конфигурация контейнеров и деплоя

Указание предназначения нод для приложений с помощью Labels

С помощью Docker Labels и использования нескольких worker-нод можно указать, какие ноды для каких приложений использовать. Для варианта по умолчанию выполните такую настройку:

```
docker node ls
```

Команда выведет все подключенные ноды. Скопируйте идентификатор worker-ноды и выполните следующие шаги:

```
docker node update --label-add=dwebstackrole=bridge <nodeid>
docker node update --label-add=dwebstackdomain=server1.example.com <nodeid>
```

Вместо `<nodeid>` подставьте скопированный идентификатор ноды, а вместо `server1.example.com` ваш сабдомен для сервиса dcomms.

Шаги выше необходимо сделать для каждой worker-ноды в случае, если их несколько.

Конфигурация приложений

В директории `/conf/*` хранятся заранее подготовленные файлы конфигурации приложений. В файле `/conf/element/config.json` укажите вашу временную зону в параметре `defaultCountryCode`.
Там же в `roomDirectory.servers` укажите все поддомены `chat.*` ваших worker-нод.
Это необходимо для правильной работы веб-клиента Element для чата Matrix.

Внесите другие изменения в конфигурации, если это необходимо.

Также вы можете указать конкретные версии образов в файле `docker-compose.yml`

вместо latest для большей стабильности сервиса в продакшне.

Деплой контейнеров и запуск сервиса

Подготовка окружения, загрузка и запуск контейнеров на worker-нодах (деплой) осуществляется на manager-ноде скриптом `provision.sh`. Для запуска деплоя выполните следующую команду для каждой worker-ноды соответственно:

```
DWEB_DOMAIN=server1.example.com ./provision.sh
```

Будьте внимательны: не деплойте ноды с одним и тем же DWEB_DOMAIN несколько раз, чтобы предотвратить перезапись данных Synapse.

В случае успеха веб-версия Element для чата Matrix будет доступна по адресу: https://chat.server1.example.com. Адрес для подключения других клиентов Matrix: https://matrix.server1.example.com.

Повторный деплой или перезапуск

Для отката и повторного деплоя или перезапуска всех сервисов dcomms на определенном сабдомене используйте скрипт `redeploy.sh` с manager-ноды:

```
DWEB_DOMAIN=server1.example.com ./redeploy.sh
```

Генерация QR-кода приглашения в Delta Chat

Для подключения к серверу Delta Chat в клиентском приложении необходимо просканировать QR-код. Для генерации QR-кода, связанного с вашим только что установленным сервером выполните следующие шаги:

1. На сервере worker просмотрите список запущенных контейнеров:

```
docker stats
```

Скопируйте идентификатор контейнера mailadm.

2. Запустите оболочку внутри контейнера mailadm:

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

Указав вместо `<container_id>` скопированный идентификатор.

3. Узнайте уникальный токен данного сервера:

```
cd /var/lib/mailadm
mailadm list-tokens
```

Команда выведет токен, например:

```
...
token = t1YWxCmX
...
```

Скопируйте его.

4. Сгенерируйте QR-код, передав в команду этот токен:

```
mailadm gen-qr t1YWxCmX
```

5. Скопируйте изображение с QR-кодом из контейнера на сервер:

```
exit
docker cp <container_id>:/var/lib/mailadm/dcaccount-server1.example.com-
t1YWxCmX ./
```

Вместо `<container_id>` используйте ваш идентификатор контейнера mailadm, а также подставьте имя файла QR-кода согласно вашим данным.

6. Разместите этот файл на веб-сервере или где-либо ещё для возможности его раздачи пользователям, чтобы они могли подключиться к вашему серверу Delta Chat.

Разворачивание веб-сайта сервиса

Проект dcomms также предлагает вам заготовку веб-сайта для продвижения вашего собственного сервиса dcomms. Он располагается в репозитории https://github.com/

censorship-no/chatv3-web/.

Вы можете разместить сайт на любом сервере, указав свои адреса нод с dcomms, либо разместить его прямо на каждой worker-ноде в директории `/var/www/dcomms`, тогда он будет доступен, соответственно, по адресу https://server1.example.com.

Установка dcomms на одном сервере

Для небольшой или экспериментальной инсталляции вы можете использовать один и тот же сервер для роли Docker Swarm manager и worker. Для этого используйте в качестве `<nodeid>` идентификатор manager-ноды.

Однако для обеспечения надежности сервиса и соответствия идеи децентрализации в полной мере мы рекомендуем разворачивать несколько серверов dcomms.

Настройка firewall

dcomms использует сеть нод Docker, поэтому мы рекомендуем запрещать доступ ко всем другим ненужным портам на уровне каждой ноды. Порты, которые должны быть открыты для корректной работы сервиса:

- CENO client: 28729/udp
- Caddy (webserver): 443/tcp, 80/tcp, 8448/tcp
- Delta Chat (postfix/dovecot): 587/tcp 143/tcp

Диагностика проблем

Для просмотра логов на сервере с ролью manager используйте следующие команды (для каждого приложения и worker-ноды, соответственно):

```
docker service logs dwebstack-server1_example_com_bridge -ft
docker service logs dwebstack-server1_example_com_caddy -ft
docker service logs dwebstack-server1_example_com_dovecot -ft
docker service logs dwebstack-server1_example_com_element -ft
docker service logs dwebstack-server1_example_com_mailadm -ft
docker service logs dwebstack-server1_example_com_postfix -ft
docker service logs dwebstack-server1_example_com_synapse -ft
```