UnicChat

Руководство администратора

V01/02/2024

Настройка окружения

Это руководство содержит подробные инструкции по настройке окружения, необходимой для запуска UnicChat. Правильная настройка окружения обеспечивает оптимальную производительность, безопасность и функциональность UnicChat.

Подготовка

Прежде чем приступить к настройке окружения, убедитесь, что выполнены следующие предварительные условия:

Системные требования: Убедитесь, что ваша система соответствует минимальным требованиям к аппаратному и программному обеспечению UnicChat.

Выполните развертывание UnicChat: Убедитесь, что у вас установлен сервер UnicChat. Для развертывания можно использовать любой из следующих методов:

- Развертывание с помощью Docker и Docker Compose (рекомендуется)
- Развертывание с помощью любого из дополнительных методов развертывания

Необходимо базовое понимание работы с интерфейсами командной строки, управлением серверами и настройкой сети.

Настройка

1. Установка переменных окружения

Переменные окружения могут использоваться для настройки процесса развертывания сервера UnicChat или для настройки его использования. Настройка переменных окружения зависит от метода развертывания. В общем случае можно выполнить следующие шаги:

- Задайте переменные среды развертывания: В файле .env или compose.yml (для развертывания Docker) задайте значения для обязательных переменных, таких как ROOT_URL, PORT и MONGO_URL. Вы также можете задать значения для необязательных переменных в соответствии с вашими требованиями.
- Измените переменные среды развертывания: Измените необходимые переменные и разверните сервер UnicChat снова, чтобы изменения вступили в силу.
- Подробную информацию см. в разделе Переменные среды.
- Настройка переменных среды: Настройте переменные среды для управления настройками сервера UnicChat.

2. Настройка базы данных

UnicChat использует MongoDB. Настройте подключение к базе данных:

- URI строки подключения MongoDB: Настройте подключение UnicChat к экземпляру MongoDB с помощью URI строки подключения. Аутентификация осуществляется с помощью имени пользователя и пароля.
- Резервное копирование и восстановление: Выполните резервное копирование данных приложения сервера UnicChat.
- Репликация: Настройте набор реплик MongoDB для повышения доступности данных.

3. Подключение внешних сервисов

Подключите внешние сервисы, такие как электронная почта, аутентификация и хранение файлов:

- Электронная почта: Используйте инструмент Mailer для отправки электронных писем пользователям в рабочей области. Для Omnichannel используйте настройку Email Inboxes для создания и управления учетными записями электронной почты для каналов.
- Аутентификация: Настройте аутентификацию и авторизацию пользователей с помощью любого из доступных методов.
- Загрузка и хранение файлов: Настройте параметры хранения для загрузки и хранения файлов (например, локальная файловая система, хранилище S3 и т. д.).

4. Настройки безопасности

Повысьте безопасность вашего сервера UnicChat:

- Проверка подлинности с помощью SSL-сертификата: Добавьте уровень безопасности в сервер UnicChat, настроив веб-сервер Nginx.
- Настройка обратного прокси-сервера SSL: Внедрите обратные прокси-серверы для работы с SSL.
- Настройка брандмауэра: Настройте брандмауэр сервера, чтобы разрешить необходимый трафик и заблокировать нежелательный доступ.
- Регулярные обновления: Постоянно обновляйте UnicChat и его зависимости, чтобы устранить уязвимости в системе безопасности.

5. Масштабируемое развертывание и автоматизация

Добавляйте дополнительные ресурсы по мере роста пользователей, чтобы поддерживать оптимальную производительность системы:

- Развертывание микросервисов: Повышение гибкости и отказоустойчивости за счет развертывания UnicChat в виде отдельных компонентов для управления большими объемами пользователей и адаптации к динамичным бизнес-требованиям.
- Запуск нескольких экземпляров: Используйте имеющееся оборудование для запуска нескольких экземпляров приложения UnicChat на текущем хосте.

6. Мониторинг данных

Собирайте и визуализируйте метрики сервера UnicChat с помощью подключаемых сервисов Prometheus и Grafana. Настройка мониторинга позволяет получить такую информацию, как использование данных сервер UnicChat, подписки, детали запросов REST API и многое другое.

Устранение неполадок

Устранение распространенных проблем при настройке окружения:

- Проблемы с подключением: Убедитесь, что все службы правильно настроены и могут взаимодействовать друг с другом.
- Ошибки в переменных окружения: Дважды проверьте настройки переменных окружения на наличие опечаток или неправильных значений.

Правильная настройка окружения имеет решающее значение для стабильного и эффективного развертывания UnicChat. Выполнив эти шаги, вы сможете убедиться, что ваш сервер UnicChat правильно настроен, безопасен и готов к работе.

Установка переменных окружения

Настройка UnicChat с помощью переменных окружения

Переменные окружения могут быть заданы для того, чтобы повлиять на развертывание сервера или на параметры и конфигурацию сервера. Настройка переменных окружения зависит от выбранного метода развертывания. В этой теме мы рассмотрим обязательные и необязательные переменные окружения, которые помогут вам успешно установить сервера в соответствии с вашими потребностями.

При использовании наиболее распространенного метода развертывания с помощью Docker и Docker Compose эти переменные можно установить в разделе окружения services в файле compose.yml.

Обязательные переменные окружения

Имя	Описание	Комментарий
`ROOT_URL`	URL-адрес, на котором будет размещен ваш экземпляр UnicChat. Другими словами, URL, который вы будете вводить в браузере для доступа к UnicChat.	Формат: [протокол]://[домен или ip][: необязательный порт]/[необязательный путь].
`PORT`	Порт, к которому будет привязан ваш экземпляр UnicChat.	Если он обслуживается под FQDN (что рекомендуется), эта настройка не имеет большого значения, если настроенный порт не находится в зарезервированном диапазоне и не пересекается с другими сервисами.
`MONGO_URL`	Строка подключения к MongoDB	Прочитайте официальную документацию MongoDB, для лучшего понимания. Этот параметр зависит от

		выбранного вами метода развертывания.
MONGO_OPLOG_URL	Строка подключения MongoDB к локальной базе данных.	Аналогично вышеуказанному, за исключением того, что указывает непосредственно на локальную базу данных. Например, mongodb://localhost:27017/local

Дополнительные переменные окружения

Имя	Описание	Комментарий
BIND_IP	IP-адрес, к которому будет привязываться узел (или основной процесс UnicChat).	Если доступ будет осуществляться через домен, что рекомендуется, пользователям следует установить этот параметр на адрес loopback (т. е. 127.0.0.1) или любой частный IP-адрес узла, доступный через обратный прокси-сервер или балансировщик нагрузки. Это гарантирует, что экземпляр не будет доступен через любой IP, который не предназначен для этого.
ADMIN_USERNAME	Имя пользователя администратора.	Пользователь Администратор может быть создан автоматически при развертывании. Установите это значение на желаемое имя пользователя администратора. Обязательно, если для успешного создания учетной записи передается любая из других переменных ADMIN_*.
ADMIN_NAME	Имя администратора.	Обязательно, если для успешного создания учетной записи передается любая из других переменных ADMIN_*.

ADMIN_PASS	Пароль пользователя администратор (в открытом тексте).	Обязателен, если для успешного создания учетной записи передается любая из других переменных ADMIN_*.
ADMIN_EMAIL	Адрес электронной почты пользователя администратор.	Обязателен, если для успешного создания учетной записи передается любая из других переменных ADMIN_*.

Конфигурация MongoDB

Аутентификация в URI MongoDB

Соединение между UnicChat и экземпляром MongoDB осуществляется с помощью URI строки подключения MongoDB. Аутентификация MongoDB осуществляется с помощью имени пользователя и пароля.

Добавьте следующий фрагмент в ваш файл .env:

MONGO_URL=mongodb://[username:password@]host1[:port1][,...hostN[:portN]][/[defaultauthdb][?options]]

В контейнер передайте MONGO_URL и MONGO_OPLOG_URL с правильными значениями для подключения.

Ваш файл docker-compose.yml должен выглядеть следующим образом:

```
environment:
    - "MONGO_URL=mongodb://uctestuser:mymongopassword@mongo:27017/unicchat?authSource=admin"
    - "MONGO_OPLOG_URL=mongodb://uctestuser:mymongopassword@mongo:27017/local?authSource=admin"
```

Если вы используете docker run, это должно выглядеть следующим образом:

```
docker run \
-e "MONGO_URL=mongodb://uctestuser:mymongopassword@mongo:27017/unicchat?authSource=admin" \
-e "MONGO_OPLOG_URL=mongodb://uctestuser:mymongopassword@mongo:27017/local?authSource=admin" \
unicchat/unicchat:X.X.X
```

Роль для аутентификации MongoDB

Если вы используете аутентификацию MongoDB, вам также может понадобиться добавить пользователю роль ClusterMonitor. Роль clusterMonitor предоставляет пользователям доступ к инструментам мониторинга MongoDB только для чтения. Это обязательное условие для того, чтобы ваш экземпляр мог использовать потоки изменений. Потоки изменений позволяют рабочей области реагировать на изменения данных в реальном времени.

Выполните следующую команду, заменив имена пользователей на те, которые выбраны для ваших пользователей:

```
admin = db.getSiblingDB("admin");
admin.grantRolesToUser('OPLOGUSER',[{ role: "clusterMonitor", db: "admin" }])
```

admin.grantRolesToUser('UNICUSER',[{ role: "clusterMonitor", db: "admin" }])

Резервное копирование и восстановление MongoDB

Создание резервных копий данных рабочего пространства - очень важная практика. Эти резервные копии могут служить мерой безопасности, позволяющей восстановить данные в случае непредвиденных обстоятельств.

UnicChat использует MongoDB в качестве базы данных. С MongoDB у вас есть несколько встроенных вариантов резервного копирования в зависимости от метода развертывания.

Резервное копирование MongoDB с помощью mongodump

Мы рассмотрим, как создать резервную копию базы данных MongoDB с помощью mongodump. mongodump позволяет создавать резервные копии автономных, репликационных или шардированных кластерных развертываний.

Резервное копирование отдельного экземпляра MongoDB

Команда для резервного копирования простого автономного экземпляра MongoDB имеет следующий формат:

mongodump --uri="mongodb://<host URL/IP>:<Port>" [additional options]

- При запуске mongodump из командной строки без каких-либо опций предполагается, что база данных находится на localhost на порту 27017 без аутентификации.
- После завершения резервного копирования будет создан каталог /dump.

Резервное копирование удаленного экземпляра MongoDB

Резервное копирование удаленного экземпляра MongoDB можно выполнить с помощью следующей команды:

mongodump --uri="mongodb://<host URL/IP>:<Port>" [additional options]

Восстановление MongoDB с помощью mongorestore

После резервного копирования экземпляра вам может понадобиться восстановить данные. Это можно сделать с помощью mongorestore. mongorestore позволяет загрузить в экземпляр MongoDB данные либо из двоичного дампа базы данных, созданного mongodump, либо из стандартного ввода.

Синтаксис команды mongorestore следующий:

mongorestore <options> <connection-string> <directory or file to restore>

Простая команда ниже восстанавливает из каталога dump на локальный экземпляр mongodb, работающий на порту 27017:

mongorestore dump/

Вы можете восстановить удаленный экземпляр, выполнив следующую команду:

mongorestore --uri="mongodb://<host URL/IP>:<Port>" /dump

Поддерживаемые версии MongoDB

Требуемые версии MongoDB могут меняться в разных основных версиях. Очень важно выбрать правильную версию базы данных при развертывании, а также убедиться, что существующая установка использует поддерживаемую версию.

Рекомендуемая версия MongoDB

Если это уже существующее развертывание, то хорошей идеей будет находиться примерно в середине списка поддерживаемых версий. Мы знаем, как сложно обновлять наши базы данных, особенно если речь идет о большом развертывании.

Если вы собираетесь развернуть новую систему, мы рекомендуем выбрать последнюю поддерживаемую версию.

Конфигурация брандмауэра

Если вы используете firewalld и не используете обратный прокси, вам может потребоваться разрешить трафик на порт 3000:

sudo firewall-cmd --permanent --add-port=3000/tcp

sudo systemctl reload firewalld

Настройка обратного прокси-сервера SSL

UnicChat - это сервер приложений среднего уровня, и сам по себе он не поддерживает SSL. Однако UnicChat хорошо работает с несколькими промышленными, проверенными обратными прокси-серверами (см. nginx ниже, например), которые вы можете настроить для работы с SSL.

Примечание: При развертывании UnicChat вы должны установить параметр ROOT_URL на HTTPS-адрес без указания номера порта. Поэтому вместо ROOT_URL=http://localhost:3000 используйте что-то вроде https://your_hostname.com.

Примечание: При настройке обратного прокси перед сервером UnicChat вам необходимо настроить UnicChat на использование правильного clientAddress. Ограничитель скорости (и, возможно, другие функции) не будут работать должным образом, если этого не сделать. Установите переменную окружения HTTP_FORWARDED_COUNT на нужное количество прокси-серверов перед UnicChat. Если вы используете snap, то документация о том, как это сделать, находится здесь.

Работа за обратным прокси-сервером nginx SSL

Примечание: Эти инструкции были написаны для Ubuntu.

Запустите эту команду от имени пользователя root:

apt-get install nginx

Добавьте свой закрытый ключ в файл /etc/nginx/certificate.key

Заблокируйте права доступа: chmod 400 /etc/nginx/certificate.key

Добавьте свой сертификат в файл /etc/nginx/certificate.crt

Отредактируйте /etc/nginx/sites-enabled/default или, если вы используете nginx из docker /etc/nginx/conf.d/default.conf и убедитесь, что вы используете свое реальное имя хоста вместо примера имени хоста "your_hostname.com" ниже.

```
# Upstreams
upstream backend {
  server 127.0.0.1:3000;
}

# HTTPS Server
server {
  listen 443;
  server_name your_hostname.com;

  # You can increase the limit if your need to.
  client_max_body_size 200M;

  error_log /var/log/nginx/unicchat.access.log;

  ssl on;
  ssl_certificate /etc/nginx/certificate.crt;
  ssl_certificate_key /etc/nginx/certificate.key;
  ssl_protocols TLSv1 TLSv1.1 TLSv1.2; # don’t use SSLv3 ref: POODLE

  location / {
    proxy_pass http://backend;
    proxy_http_version 1.1;
    proxy_set_header Upgrade $http_upgrade;
    proxy_set_header Connection "upgrade";
    proxy_set_header Host $http_host;

    proxy_set_header X-Real-IP $remote_addr;
    proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
    proxy_set_header X-Forwarded-Proto https;
    proxy_set_header X-Nginx-Proxy true;

    proxy_redirect off;
  }
}
```

Перезапустите Nginx: service nginx restart

Настройка аутентификации SSL-сертификата клиента для UnicChat

Возможно, вам захочется добавить дополнительный уровень безопасности в свое приложение. После установки UnicChat, следуя нашему руководству по развертыванию с помощью Docker и Docker Compose, вот следующие шаги, которые необходимо выполнить:

Установите Nginx

```
sudo apt install -y nginx
```

Установите Certbot

Установите Certbot для управления SSL-сертификатами от LetsEncrypt

sudo snap install --classic certbot

sudo certbot –nginx

Вам будет предложено указать действующий адрес электронной почты и набор доменов.

Генерация сертификатов центра сертификации

Сгенерируйте ключ для вашего центра сертификации:

```
openssl genrsa -des3 -out ca.key 4096
```

Сгенерируйте сертификат для вашего центра сертификации:

```
openssl req -new -x509 -days 365 -key ca.key -out ca.crt
```

Выполните ту же команду, чтобы обновить сертификат. Чтобы запомнить выбранные вами параметры, выполните следующую команду:

openssl x509 -in ca.crt -noout -text

Переместить сертификат

Переместите сертификат в каталог /etc/ssl/private/client-cert-ca.crt.

Обновите конфигурацию Nginx

Добавьте сертификат, включите SSL-аутентификацию клиента и добавьте блок location.

```
ssl_client_certificate /etc/ssl/private/client-cert-ca.crt;
ssl_verify_client optional;

location / {
  if ($ssl_client_verify != SUCCESS) {
    return 403;
  }

   proxy_pass http://localhost:3000;
   proxy_http_version 1.1;
   proxy_set_header Upgrade $http_upgrade;
   proxy_set_header Connection "upgrade";
   proxy_set_header Host $http_host;
   proxy_set_header X-Real-IP $remote_addr;
   proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
   proxy_set_header X-Forwarded-Proto https;
   proxy_set_header X-Nginx-Proxy true;
   proxy_redirect off;
 }
```

Выпуск клиентских SSL-сертификатов для пользователей

При желании вы можете поручить пользователям выполнить большинство из этих шагов. Но ниже перечислены шаги, необходимые для создания сертификата, который будет использоваться для аутентификации клиента.

Генерирование ключа для пользователя

openssl genrsa -des3 -out user.key 4096

Создать CSR

```
openssl req -new -key user.key -out user.csr
```

Ответьте на все вопросы, обязательно указав свой адрес электронной почты и общее имя. CSR должен быть отправлен администратору (или вам, если вы делаете это от имени пользователя).

Подпишите CSR с помощью центра сертификации

В качестве администратора возьмите предоставленный вам или сгенерированный вами CSR, подпишите его и создайте действительный сертификат:

openssl x509 -req -days 365 -in user.csr -CA ca.crt -CAkey ca.key -set_serial 01 -out user.crt

При каждой подписи нужно увеличивать серийный номер. Когда срок действия сертификата истечет, новый CSR создавать не нужно; можно подписать тот же самый сертификат, что приведет к созданию нового сертификата, привязанного к этому открытому ключу.

Возвращение сертификата

Теперь подписанный сертификат (user.crt) может быть отправлен обратно пользователю вместе с сертификатом ЦС (ca.crt).

Для использования в браузерах и мобильных устройствах сгенерируйте pkcs #12, используя пользовательский сертификат и ключ:

openssl pkcs12 -export -out user.pfx -inkey user.key -in user.crt -certfile ca.crt