

Инструкция по установке POLYCORE

Содержание

Требования к системе	3
Серверная часть	3
Локальная сеть	3
Установка версии	4
Установка и настройка Docker	4
Установка компонентов на один сервер	4
Вспомогательный функционал	5
Порядок обновления системы	5
Резервные копии	6
Создание резервной копии	6
Восстановление из резервной копии	6

Требования к системе

Серверная часть

Минимальные требования к серверному оборудованию следующие:

- 8 vCPU (2.8 ГГц+)
- 12GB RAM
- 70GB HDD

Из которых:

Сервис app:

- 2 vCPU
- 4GB RAM
- 10GB HDD

Сервис grid:

- 4 vCPU
- 4GB RAM
- 30GB HDD

Внутренняя база данных системы (redis, postgres):

- 2 vCPU
- 4GB RAM
- 30GB HDD

Ориентировочная формула для подсчета конфигурации в зависимости от количества пользователей: дополнительно к минимальным системным требованиям необходимо RAM 256-512MB CPU 0.1 vCPU в среднем на каждого пользователя. Конечная конфигурация уточняется в каждом случае отдельно.

Операционная система: Astra Linux Special Edition 1.6 (Воронеж) или аналог.

Права пользователя, разворачивающего приложение: user - non-root with sudo privileges.

Дополнительные требования к установленным приложениям: Docker версии 20.10.0 и до 25, Docker-compose версия 1.29 и выше (для Docker с версии 26 необходимо использовать плагин Docker-compose).

Локальная сеть

Все компоненты платформы должны находиться в одной подсети или должна обеспечиваться прозрачная маршрутизация. Не рекомендуется использовать NAT. В рамках ознакомления рекомендуется отключить брандмауэры. Внутри локальной сети между всеми компонентами не должно быть ограничений по передаче данных. Для доступа из внешней сети достаточно открыть порт, используемый POLYCORE (порт задается при установке). При использовании системы с установленными антивирусами или комплексными системами защиты необходимо обеспечить свободную работу, сетевую активность и взаимодействие компонентов.

Установка версии

Установка и настройка Docker

1. Установить Docker в соответствии с инструкцией:
<https://docs.docker.com/install/linux/docker-ce/ubuntu/>.
Примечание - Рекомендуемая версия 25.0.5.
2. Установить Docker Compose в соответствии с инструкцией:
<https://docs.docker.com/compose/install/>.
Примечание - Рекомендуемая версия 1.29.2.

Для Docker с версии 26 необходимо использовать плагин Docker-compose.

Установка компонентов на один сервер

- Создать директорию для файлов образов

```
cd ~ && mkdir install
```

- Скопировать из полученного дистрибутива в созданную директорию архивы образов (где <version> - версия приложения):
 - r5_polycore_grid:<version>.tar
 - r5_polycore_app:<version>.tar
 - r5_redis:7-alpine.tar
 - r5_postgres13:1.0.tar
- Выполнить команды:

```
docker load -i ~/install/r5_polycore_grid:<version>.tar
docker load -i ~/install/r5_polycore_app:<version>.tar
docker load -i ~/install/r5_redis:6-alpine.tar
docker load -i ~/install/r5_postgres13:1.0.tar
```

- Создать директории сервиса

```
cd ~ && mkdir polycore && cd polycore
```

- Скопировать из полученного дистрибутива в созданную директорию компоненты:
 - docker-compose_prod.yml.tpl
 - .env.tpl
 - prod.nginx.conf.tpl
 - coolwsd.xml.tpl
- Настроить компоненты:
 - переименовать `docker-compose_prod.yml.tpl` в `docker-compose.yml`
 - заменить в `docker-compose.yml` все переменные вида `{{...}}` на актуальные значения
 - переименовать `.env.tpl` в `.env`
 - заменить в `.env` все значения, представленные как `{{...}}`, на актуальные
 - переименовать `prod.nginx.conf.tpl` в `nginx.conf`
 - заменить в `nginx.conf` все значения, представленные как `{{...}}`, на актуальные
 - переименовать `coolwsd.xml.tpl` в `coolwsd.xml`
 - заменить в `coolwsd.xml` все значения, представленные как `{{...}}`, на актуальные
 - создать директорию файлов БД, указанную в `docker-compose.yml` в качестве тома

- запустить сервис базы данных, предварительно указав желаемый пароль вместо `{{ sa_password }}` и полный путь для тома база данных бота на хосте вместо `{{ /polycore/database/volume/path }}` (тот же, что был задан в `docker-compose.yml`)
- если сервис не должен использовать https:
 - убрать из `docker-compose.yml` секреты
 - убрать из `nginx.conf` сертификаты и параметр `ssl`
- выполнить

```
docker run -it --rm -d --name polycore-db -v {{
/polycore/database/volume/path }}:/var/lib/postgresql/data -e
POSTGRES_DB=appdb -e POSTGRES_USER=sa -e POSTGRES_PASSWORD={{sa_password}} -e
PGDATA=/var/lib/postgresql/data
registry.polyanalitika.ru/r5/r5_postgres13:1.0
```

- выполнить скрипт, предварительно заменив в нем `{{ polycore_password }}` на соответствующий пароль из `.env`

```
docker exec -it --user postgres polycore-db psql -U sa -d appdb -c "create
user polycore with password '{{ polycore_password }}';create schema
polycore;alter schema polycore owner to polycore;alter user polycore set
search_path to polycore;revoke create on schema public from polycore;"
```

- остановить контейнер

```
docker stop polycore-db
```

- выполнить миграции структуры базы данных и инициализирующих данных, указав email и пароль администратора сервиса `sa`

```
docker compose run --rm -e POLYCORE_SA_EMAIL=sa@example.com -e
POLYCORE_SA_PASSWORD=changeme app flask --app polycore.app.py db upgrade
```

- запустить сервис командой `docker compose up -d`

Вспомогательный функционал

С зашифровать строку (полученное значение следует использовать с префиксом `fernet:`)

```
docker exec -i polycore-app flask encrypt --text=somepassword
```

Порядок обновления системы

При обновлении необходимо обеспечить наличие образов приложения новой версии в системе, аналогично как при [установке](#).

Ниже указан общий порядок обновления:

1. убедиться, что сервис остановлен
2. обновить версии образов в файле `docker-compose_prod.yml`
3. запустить сервис
4. в случае необходимости (уточняется в описании к конкретной версии), выполнить обновление системных таблиц: `docker exec -i polycore-app flask db upgrade`

Перед обновлением рекомендуется создать резервную копию БД.

Резервные копии

Создание резервной копии

Для создания резервной копии БД, выполнить

```
docker stop polycore-app && \  
mkdir -p backups && \  
docker exec -t polycore-db pg_dump -U sa -d appdb > backups/backup_$(date  
+%Y-%m-%d).sql && \  
docker start polycore-app
```

Восстановление из резервной копии

Для восстановления из резервной копии БД, выполнить

```
docker stop polycore-app && \  
cat backups/backup_<BACKUP DATE>.sql | docker exec -i polycore-db psql -U sa  
-d appdb && \  
docker exec -i polycore-app flask db upgrade && \  
docker start polycore-app
```