

Инструкция по инсталляции

1. БАЗОВОЕ СИСТЕМНОЕ ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ

Системные программные средства, для которых обеспечивается эффективная работа:

- операционная система для серверов: Ubuntu Server 20.04 LTS
- операционные системы для рабочих станций - Windows 7, Windows 8, Windows 8.1, Windows 10.

На сервере необходимо следующее программное обеспечение:

- программная платформа для разработки, тестирования и развертывания приложений Docker 20.10.8;
- утилита для определения и совместного использования многоконтейнерных приложений Docker-Compose 1.29.1;
- утилита для передачи данных, указанных с помощью синтаксиса URL cURL 7.68.0;

На клиентских рабочих станциях необходимо следующее программное обеспечение:

- веб-браузер Google Chrome 90.0 и выше или Mozilla Firefox версии 89.0 или выше, или Yandex Browser версии 21.3 и выше.

2. ТЕХНИЧЕСКОЕ ОБЕСПЕЧЕНИЕ

Установленные для эксплуатации Программы технические средства (персональные компьютеры, принтеры, устройства резервного хранения данных, сетевые компоненты) должны быть совместимы между собой и поддерживать сетевой протокол TCP/IP.

Для работы Программы используется «IBM-совместимые» компьютеры с операционной системой Microsoft Windows, GNU/Linux.

Серверные компоненты Программы должны быть установлены на выделенном сервере, предназначенном исключительно для эксплуатации серверных компонент Программы.

Минимальные технические характеристики клиентских компьютеров:

- процессор 2GHz;
- память 2GB;
- свободное дисковое пространство 512MB (+ размер прикладных метаданных).

Минимальные технические характеристики серверного оборудования:

- двухпроцессорная система 2GHz;
- память 8GB;
- свободное дисковое пространство 10GB (+ пространство для размещения прикладных систем и баз данных).

Рекомендуемые технические характеристики клиентских компьютеров:

- процессор 2.4GHz;
- память 8GB;
- свободное дисковое пространство 1GB (+ размер прикладных метаданных).

Рекомендуемые технические характеристики серверного оборудования:

- четырехпроцессорная система 2,5 GHz;
- память 16GB – 32 GB;

□ свободное дисковое пространство 20GB (+ пространство для размещения прикладных систем и баз данных).

Установка должна производиться на операционную систему **Ubuntu Server 20.04 LTS**.
Возможны два варианта установки приложения: на отдельный сервер, в комплексе с другими приложениями на один сервер.

Установка на отдельный сервер

Перед установкой убедитесь, что в системе установлены следующие пакеты:

- **apt-transport-https**
- **ca-certificates**
- **curl**
- **gnupg**
- **lsb_release**
- **docker**
- **docker-compose**

Описание установки пакетов:

- <https://docs.docker.com/engine/install/ubuntu/>
- <https://docs.docker.com/compose/install/>

Остальные пакеты устанавливаются из стандартных репозиториев.

Процедура установки:

1. Получение актуальных версий распаковав архив **agents_network.zip** и загрузив образы командами:

```
docker load -i postgres.tar  
docker load -i nginx_latest.tar  
docker load -i agents_network.tar  
docker load -i redis_5_alpine.tar
```

2. Настройка конфигурационных файлов:

1. Создать файл `docker-compose.yml` со следующим содержимым:

```
version: "2.4"  
  
services:  
  web:  
    image: nginx:latest  
    container_name: nginx  
    depends_on:  
      - app  
    ports:  
      - "127.0.0.1:80:80/tcp"  
    networks:  
      - agents_network  
    volumes:  
      - ./default.conf:/etc/nginx/conf.d/default.conf  
    logging:  
      driver: "json-file"  
    options:  
      max-size: "10m"
```

```
max-file: "2"
```

```
app:  
  image: unicom/agents_network:latest  
  container_name: agents_network  
  networks:  
    - agents_network  
  environment:  
    - APP_NAME=agents_network  
    - APP_ENV=production  
    - REDIS_HOST=redis  
    - REDIS_PORT=6379  
    - DB_CONNECTION=pgsql  
    - DB_HOST=postgres  
    - DB_DATABASE=postgres  
    - DB_USERNAME=postgres  
  logging:  
    driver: "json-file"  
    options:  
      max-size: "10m"  
      max-file: "2"
```

```
db:  
  image: postgres:11.5-alpine  
  container_name: postgres  
  networks:  
    - agents_network  
  logging:  
    driver: "json-file"  
    options:  
      max-size: "10m"  
      max-file: "2"
```

```
redis:  
  image: redis:5-alpine  
  container_name: redis  
  networks:  
    - agents_network  
  logging:  
    driver: "json-file"  
    options:  
      max-size: "10m"  
      max-file: "2"
```

```
networks:  
  agents_network:  
    name: agents_network
```

2. В одной директории с файлом манифеста `docker-compose.yml` необходимо создать файл конфигурации `nginx` со следующим содержимым:

```
server {
```

```
listen 80;
server_name localhost;
server_tokens off;
client_body_buffer_size 4k;
client_body_timeout 30s;
client_header_buffer_size 1k;
client_header_timeout 30s;
client_max_body_size 2m;
keepalive_timeout 60s;
tcp_nodelay on;
tcp_nopush off;
```

```
location / {
    proxy_pass http://app:9002;
    proxy_buffer_size 4k;
    proxy_buffering on;
    proxy_buffers 8 8k;
    proxy_connect_timeout 30s;
    proxy_read_timeout 60s;
    proxy_headers_hash_max_size 512;
    proxy_http_version 1.1;
    proxy_request_buffering on;
}
}
```

Установка в комплексе с другими приложения на один сервер

Перед непосредственной установкой должны быть установлены остальные приложения комплекса и следующие пакеты:

- **apt-transport-https**
- **ca-certificates**
- **curl**
- **gnupg**
- **lsb_release**
- **docker**
- **docker-compose**

Процедура установки:

1. Получение актуальных версий распаковав архив **agents_network.zip** и загрузив образы командами:
docker load -i postgres.tar
docker load -i nginx_latest.tar
docker load -i agents_network.tar
docker load -i redis_5_alpine.tar

2. Настройка конфигурационных файлов:

1. Создать файл **docker-compose.yml** со следующим содержимым:

```
version: "2.4"
```

services:
web:
image: nginx:latest
container_name: nginx
depends_on:
- app
ports:
- "127.0.0.1:80:80/tcp"
networks:
- agents_network
volumes:
- ./default.conf:/etc/nginx/conf.d/default.conf
logging:
driver: "json-file"
options:
max-size: "10m"
max-file: "2"

app:
image: unicom/agents_network:latest
container_name: agents_network
networks:
- agents_network
environment:
- APP_NAME=agents_network
- APP_ENV=production
- REDIS_HOST=redis
- REDIS_PORT=6379
- DB_CONNECTION=pgsql
- DB_HOST=postgres
- DB_DATABASE=postgres
- DB_USERNAME=postgres
logging:
driver: "json-file"
options:
max-size: "10m"
max-file: "2"

db:
image: postgres:11.5-alpine
container_name: postgres
networks:
- agents_network
logging:
driver: "json-file"
options:
max-size: "10m"
max-file: "2"

redis:
image: redis:5-alpine

```
container_name: redis
networks:
  - agents_network
logging:
  driver: "json-file"
  options:
    max-size: "10m"
    max-file: "2"
```

```
networks:
  agents_network:
    name: agents_network
```

2. В одной директории с файлом манифеста docker-compose.yml необходимо создать файл конфигурации nginx со следующим содержимым:

```
server {
    listen 80;
    server_name localhost;
    server_tokens off;
    client_body_buffer_size 4k;
    client_body_timeout 30s;
    client_header_buffer_size 1k;
    client_header_timeout 30s;
    client_max_body_size 2m;
    keepalive_timeout 60s;
    tcp_nodelay on;
    tcp_nopush off;

    location / {
        proxy_pass http://app:9002;
        proxy_buffer_size 4k;
        proxy_buffering on;
        proxy_buffers 8 8k;
        proxy_connect_timeout 30s;
        proxy_read_timeout 60s;
        proxy_headers_hash_max_size 512;
        proxy_http_version 1.1;
        proxy_request_buffering on;
    }
}
```

Запуск приложений:

Запустить agents_network: `docker-compose -f docker-compose.yml up -d`