



ПО «Раменка-Сервер»

Руководство пользователя (РП) по установке ВСУ
«Раменка-Сервер»

Москва, 2023

СОДЕРЖАНИЕ

1. Требования для установки ВСУ «Раменка-Сервер»	3
2. Инструкция по установке ВСУ «Раменка-Сервер»	4

1. Требования для установки ВСУ «Раменка-Сервер»

Аппаратные требования к серверу:

- процессор семейства Intel Core i7 8x (или более производительный) или аналогичный AMD. Количество ядер 4 или больше;
- оперативная память объёмом 8 Гб или больше;

Программные требования к серверу:

- ОС семейства Linux:
 - ОС Fedora 36 и выше;
 - ОС Debian 11 и выше;
 - ОС Ubuntu Jammy Jellyfish 22.04 и выше;
 - ОС Astra Linux 1.6 и выше;
 - ОС Red 7.3.2 и выше.
- Docker 20.10.21 или выше;
- Docker Compose 2.12.2 или выше;

ВСУ «Раменка-Сервер» корректно работает в среде следующих интернет-браузеров:

- Chrome 60 и выше;
- FireFox 60 и выше;
- Safari 10 и выше.

2. Инструкция по установке ВСУ «Раменка-Сервер»

ВСУ «Раменка-Сервер» – это серверное приложение, работающее на базе WEB-сервера и NGINX-сервера, запускаемое в ОС Linux в контейнерном окружении платформы Docker с использованием инструмента контейнеризации Docker Compose. (см. рис.1)

Далее описан порядок действий для запуска ВСУ.

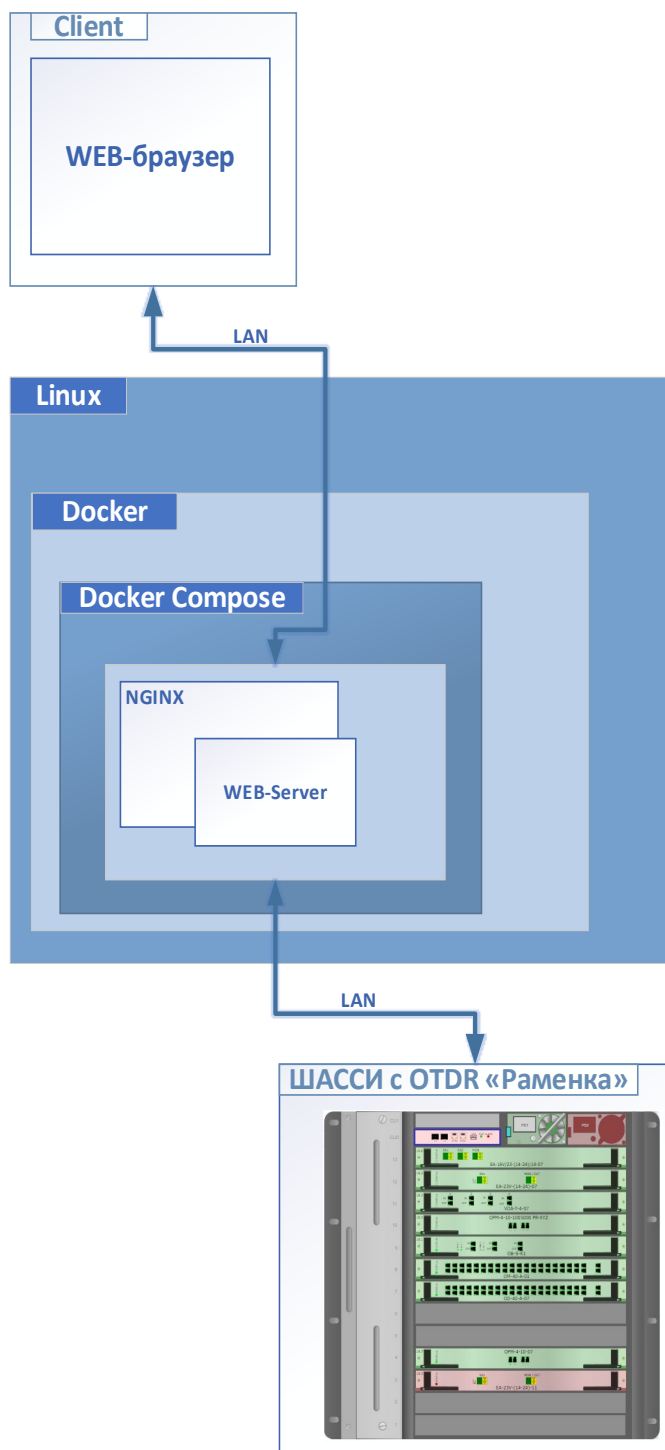


Рис.1 – Принцип работы ВСУ «Раменка-Сервер»

2.1

Установить Docker 20.10.21 или выше, установить Docker Compose 2.12.2 или выше.

Актуальная инструкция на английском языке

<https://docs.docker.com/desktop/install/linux-install/> ниже приведены шаги из этой инструкции.

2.2

Удалить прошлую версию Docker, если она была ранее установлена (здесь и далее, после знака "\$" написаны команды, которые необходимо ввести в терминал Linux, а после знака "#" написаны текстовые комментарии, поясняющие суть выполняемых операций)

```
$ sudo apt remove docker-desktop
```

```
# удаляет ранее установленный пакет
```

```
$ rm -r $HOME/.docker/desktop
```

```
# удаляет рекурсивно все файлы и папки внутри $HOME/.docker/desktop
```

```
$ sudo rm /usr/local/bin/com.docker.cli
```

```
# удаляет файл com.docker.cli
```

```
$ sudo apt purge docker-desktop
```

```
# удаляет ранее установленный пакет, включая конфигурационные файлы
```

2.3

Установить gnome-terminal (для не - Gnome систем)

```
$ sudo apt install gnome-terminal
```

```
# автоматически устанавливает gnome-terminal
```

2.4.1

Настроить репозиторий Docker (для возможности установки и последующего автоматического обновления)

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

```
# обновляет список доступных пакетов в репозитории
```

```
$ sudo apt-get install \
```

```
ca-certificates \
```

```
curl \
```

```
gnupg \
```

```
lsb-release
```

```
# устанавливаются распространенный набор сертификатов, программа консольных команд CURL, программа шифрования GnuPG, утилиту контроля соответствия стандарту LSB
```

2.4.2

Добавить официальный GPG ключ Docker (пример для ОС Debian и для ОС Ubuntu различные команды)

Для ОС Debian:

```
$ sudo mkdir -p /etc/apt/keyrings
# создание директории /etc/apt/keyrings
$ curl -fsSL https://download.docker.com/linux/debian/gpg | sudo gpg --dearmor -o /etc/apt/keyrings/docker.gpg
# скачать официальный Gpg ключ Docker и передать его в сформированный локальный файл ключа.
```

Для ОС Ubuntu:

```
$ sudo mkdir -p /etc/apt/keyrings
# создание директории /etc/apt/keyrings
$ curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo gpg --dearmor -o /etc/apt/keyrings/docker.gpg
# скачать официальный Gpg ключ Docker и передать его в сформированный локальный файл ключа.
```

2.4.3

Завершение настройки репозитория (пример для ОС Debian и ОС Ubuntu различные команды)

Для ОС Debian:

```
$ echo \
"deb [arch=$(dpkg --print-architecture) signed-by=/etc/apt/keyrings/docker.gpg]
https://download.docker.com/linux/debian \ $(lsb_release -cs) stable" | sudo tee
/etc/apt/sources.list.d/docker.list > /dev/null
# формируется способ сборки пакетов docker из репозитория, подключается ключ Gpg
```

Для ОС Ubuntu:

```
$ echo \
"deb [arch=$(dpkg --print-architecture) signed-by=/etc/apt/keyrings/docker.gpg]
https://download.docker.com/linux/ubuntu \ $(lsb_release -cs) stable" | sudo tee
/etc/apt/sources.list.d/docker.list > /dev/null
# формируется способ сборки пакетов docker из репозитория, подключается ключ Gpg
```

2.4.4

Для применения параметров обновленного репозитория

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

```
# обновляет список доступных пакетов в репозитории, подгружает произведенные ранее изменения
```

2.4.5

Для установки пакетов

```
$ sudo apt-get install docker-ce docker-ce-cli containerd.io docker-compose-plugin
```

```
# устанавливает пакеты containerd.io_<version>_<arch>.deb docker-ce_<version>_<arch>.deb, docker-ce-cli_<version>_<arch>.deb, docker-compose-plugin_<version>_<arch>.deb
```

2.4.6

Проверьте версии установленных Docker Compose и Docker

```
$ docker compose version
```

```
# запрашивает версию Docker Compose
```

```
$ docker --version
```

```
# запрашивает версию Docker
```

```
$ docker version
```

```
# запрашивает версии пакетов Docker
```

2.5

Скачать версию сборки ВСУ «Раменка-Сервер», содержащую архив server.tar, по предоставленной ссылке

2.6

Перейдите в терминале в каталог, где располагается server.tar, оттуда запустить команду на распаковку архива (например, если файл был скачен в папку Downloads)

```
$ cd $HOME/Downloads
```

```
# перейти в командной строке в директорию, где лежит server.tar
```

```
$ ls
```

```
# отобразить содержимое рабочей директории, убедиться, что там есть server.tar
```

```
$ tar -xf server.tar
```

```
# распакует архив внутрь рабочей папки
```

2.7

Из архива распакуются необходимые файлы для конфигурации сервера ВСУ.

В том числе файл `install.sh`, завершающий настройку сервера ВСУ, включая WEB-сервер и NGINX-сервер, запускаемые в Docker контейнере. Так же в архиве доступен для ознакомления файл `Readme.md`. Для запуска скрипта, находясь в каталоге, введите `./install.sh`, например

```
$ sudo ./install.sh
```

```
# запускает работу скрипта развертывания конфигурации сервера от имени суперпользователя на порту 80.
```

При необходимости можно указать другой номер порта для доступа к серверу в команде `./install.sh <порт>`, где в качестве параметра передается номер порта, например

```
$ sudo ./install.sh 8000
```

```
# запускает работу скрипта развертывания конфигурации сервера от имени суперпользователя на порту 8000.
```

2.8

Включите сервер ВСУ командой

```
$ sudo docker compose up -d
```

```
# создает виртуальный контейнер, запускает контейнер и содержимое
```

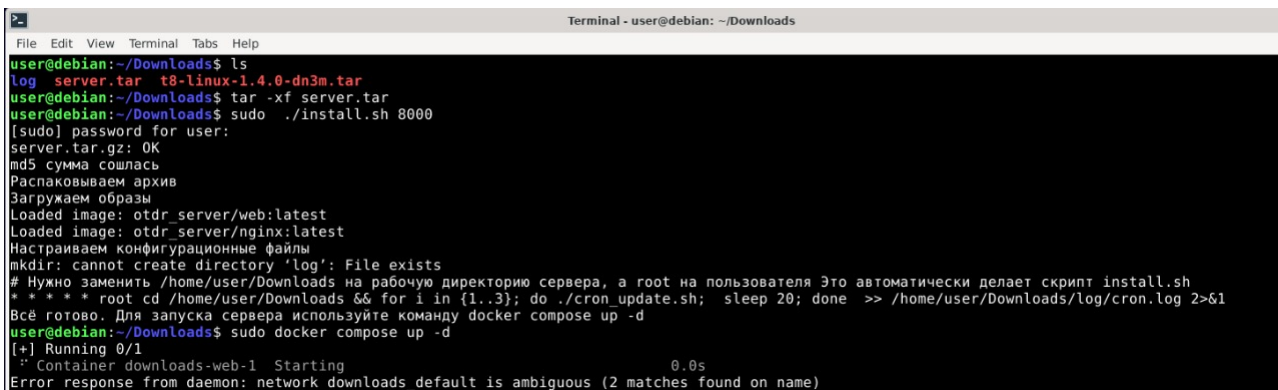
2.9

На этом настройка сервера и окружения закончена, сервер ВСУ после включения будет доступен из WEB-браузера по адресу <http://localhost> или <http://localhost:<порт>>, например, <http://localhost> или <http://localhost:8000> при развертывании сервера на порту 8000.

Примечания:

```
$ sudo docker compose down
```

останавливает старые контейнеры, если выпадает следующая ошибка



```
Terminal - user@debian: ~/Downloads
File Edit View Terminal Tabs Help
user@debian:~/Downloads$ ls
log server.tar t8-linux-1.4.0-dn3m.tar
user@debian:~/Downloads$ tar -xf server.tar
user@debian:~/Downloads$ sudo ./install.sh 8000
[sudo] password for user:
server.tar.gz: OK
md5 сумма сошлась
Распаковываем архив
Загружаем образы
Loaded image: otdr_server/web:latest
Loaded image: otdr_server/nginx:latest
Настраиваем конфигурационные файлы
mkdir: cannot create directory 'log': File exists
# Нужно заменить /home/user/Downloads на рабочую директорию сервера, а root на пользователя Это автоматически делает скрипт install.sh
* * * * * root cd /home/user/Downloads && for i in {1..3}; do ./cron_update.sh; sleep 20; done >> /home/user/Downloads/log/cron.log 2->&1
Всё готово. Для запуска сервера используйте команду docker compose up -d
user@debian:~/Downloads$ sudo docker compose up -d
[+] Running 0/1
 * Container downloads-web-1 Starting                                0.0s
Error response from daemon: network downloads_default is ambiguous (2 matches found on name)
```

```
$ sudo docker compose down -v
```

останавливает сервер и удаляет базу данных