



OCTOPUS

## Развертывание и запуск Октопус в Oracle OS

# Развертывание и запуск Октопус в Oracle OS

## Подготовка системы

Перед установкой Октопус выполните следующие команды для обновления системы и установки необходимых зависимостей:

```
sudo dnf install -y yum-utils device-mapper-persistent-data
```

## Podman

Podman и Docker — конкурирующие контейнерные движки, которые могут конфликтовать между собой при совместной установке в одной системе.

Следует проверить, установлен ли podman, выполнив команду:

```
podman --version
```

Если Podman установлен, вы увидите его версию, например, `podman version 3.4.4`

Удалить его можно, применив команду:

```
sudo dnf remove -y podman buildah
```

Если Podman не установлен, система вернет сообщение: `podman: command not found`

## Установка Docker

Добавьте официальный репозиторий Docker:

```
sudo yum-config-manager --add-repo https://download.docker.com/linux/centos/docker-ce.repo
```

Установите Docker:

```
sudo dnf install -y docker-ce docker-ce-cli containerd.io sudo systemctl enable --now docker
```

Проверьте версию Docker:

```
sudo docker --version
```

## Установка Docker Compose v2

Создайте директорию для плагинов Docker:

```
mkdir -p $DOCKER_CONFIG/cli-plugins
```

Скачайте бинарный файл Docker Compose:

```
curl -SL https://github.com/docker/compose/releases/latest/download/docker-compose-linux-x86_64 -o $DOCKER_CONFIG/cli-plugins/docker-compose
```

Сделайте файл исполняемым:

```
chmod +x $DOCKER_CONFIG/cli-plugins/docker-compose
```

Проверьте установленную версию:

```
docker compose version
```

## Настройка пользователя

Добавьте авторизованного пользователя системы (по умолчанию **octopus**) в группу **docker**:

```
sudo usermod -aG docker $USER
```

Авторизуйтесь в приватном реестре (запросите логин/пароль у представителей Октопус):

```
docker login registry.usetech.ru -u <username> -p <password>
```

После добавления пользователя в группу "docker" выполните команду:

```
cd /usr/bin/ && sudo ln -s /home/octopus/octopus/scripts/update_begin.sh update
```

### Настройка переменных окружения

Создайте файл для переменных окружения:

```
sudo vim /etc/profile.d/octopus_variable.sh
```

Добавьте в файл следующие строки (подставьте в качестве значения переменной PRIVATETOKEN актуальный токен):

```
export OCTOPUS_WORKDIR=~/.octopus
export PRIVATETOKEN=<private_token>
```

Примените изменения:

```
source /etc/profile.d/octopus_variable.sh
```

## Развертывание Октопус

---

Перейдите в домашнюю директорию:

```
cd ~
```

Загрузите архив с конфигурациями (запросите архив octopus.deployment-<version>.zip у представителей Octopus) и распакуйте его:

```
unzip octopus.deployment-<version>.zip -d octopus
```

Отредактируйте файл .env :

```
vim ~/.octopus/.env
```

Измените строку:

```
OCTOPUS_REPOSITORY="registry.usetech.ru/octopus/octopus/"
```

### Запуск Октопус

Загрузите образы:

```
docker compose pull
```

Запустите контейнеры в фоновом режиме:

```
docker compose up -d
```

Проверьте статус контейнеров:

```
docker ps
```

### Доступ к веб-интерфейсу

После запуска контейнеров, Октопус будет доступен по IP-адресу сервера (для версий Октопус < 1.30 порт **80** ,для версий 1.30 и старше порт **443** )

Используйте логин/пароль, предоставленные представителями Октопус.

## Развертывание Октопус оффлайн

---

Перейдите в домашнюю директорию Octopus:

```
cd ~/.octopus
```

Запросите архив у представителей Octopus и разархивируйте его, выполнив команду:

```
unzip octopus-update-<version>.zip
```

Пример команды:

```
octopus-update-1.50-current.zip
```

Переместите конфигурацию:

```
mv octopus.deployment/ octopus/
```

Импортируйте образы:

```
./octopus/scripts/import-images.sh octopus.images/
```

Запустите Октопус:

```
cd octopus && docker compose up -d
```

## Оффлайн обновление Октопус

Загрузите архивы с конфигурациями и образами (запросите архив у представителей Октопус)

Выполните команду обновления:

```
update --file=octopus-update-<version>.zip
```

```
unknown option --file
[octopus@localhost ~]$ update --file=octopus-update-1.50-current.zip
API =
UI =
file = octopus-update-1.50-current.zip
offline update
octopus-update-1.50-current.zip
Version for update: 1.50
Archive: octopus-update-1.50-current.zip
  creating: octopus.deployment/
 inflating: octopus.deployment/docker-compose-diag.yml
 inflating: octopus.deployment/docker-compose-update.yml
 inflating: octopus.deployment/.env
 inflating: octopus.deployment/docker-compose-cluster.yml
 inflating: octopus.deployment/docker-compose.yml
  creating: octopus.deployment/config/
  creating: octopus.deployment/config/templates/
 inflating: octopus.deployment/config/templates/yc/yc_templates.csv
  creating: octopus.deployment/config/keycloak-init/
 inflating: octopus.deployment/config/keycloak-init/realm.json
  creating: octopus.deployment/config/postgres/
  creating: octopus.deployment/config/postgres/dumps/
```