# ПР №1 РАЗРАБОТКА КЛИЕНТ-СЕРВЕРНЫХ ПРИЛОЖЕНИЙ

Прежде всего продемонстрируем, как удалить пакеты старых версий Docker с именами docker, docker.io или docker-engine (содержимое директории /var/lib/docker/, включая образы, контейнеры, тома будет сохранено). Для этого в терминале ВМ выполним команду:

*$ sudo apt-get remove docker docker-engine docker.io containerd runc*

Если на созданной ВМ Docker ранее не устанавливалась, получаем ожидаемый вывод, а именно

*«Невозможно найти пакет docker-engine»*

Далее рассмотрим процесс установки, используя официальный репозиторий Docker. Это самый простой способ. Выполняем следующие команды:

1. Обновляем индексацию пакетов

*$ sudo apt-get update*

2. Устанавливаем пакеты, позволяющие менеджеру пакетов apt взаимодействовать

с репозиторием по протоколу HTTPS

*$ sudo apt-get install \*

*apt-transport-https \*

*ca-certificates \*

*curl \*

*gnupg-agent \*

*software-properties-common*

3. Добавляем официальный GPG–ключ репозитория Docker

*$ curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo apt-key add -*

Далее необходимо проверить что получен следующий ключ

*9DC8 5822 9FC7 DD38 854A E2D8 8D81 803C 0EBF CD88*

для чего введем команду поиска по последним 8 символам

*$ sudo apt-key fingerprint 0EBFCD88*

В результате получается вывод, представленный на Рисунке 1.

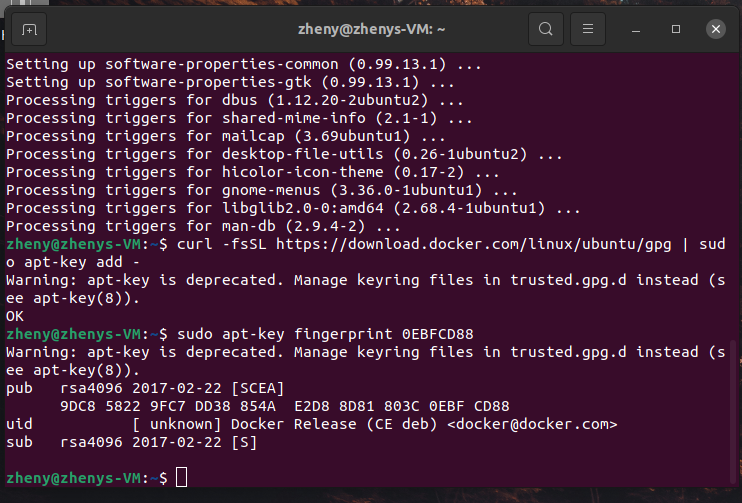


Рисунок 1 – Вывод команды проверки ключа

Далее добавляем репозиторий stable, Рисунок 2.

*$ sudo add-apt-repository \*

*"deb [arch=amd64] https://download.docker.com/linux/ubuntu \*

*$(lsb\_release -cs) \*

*stable"*

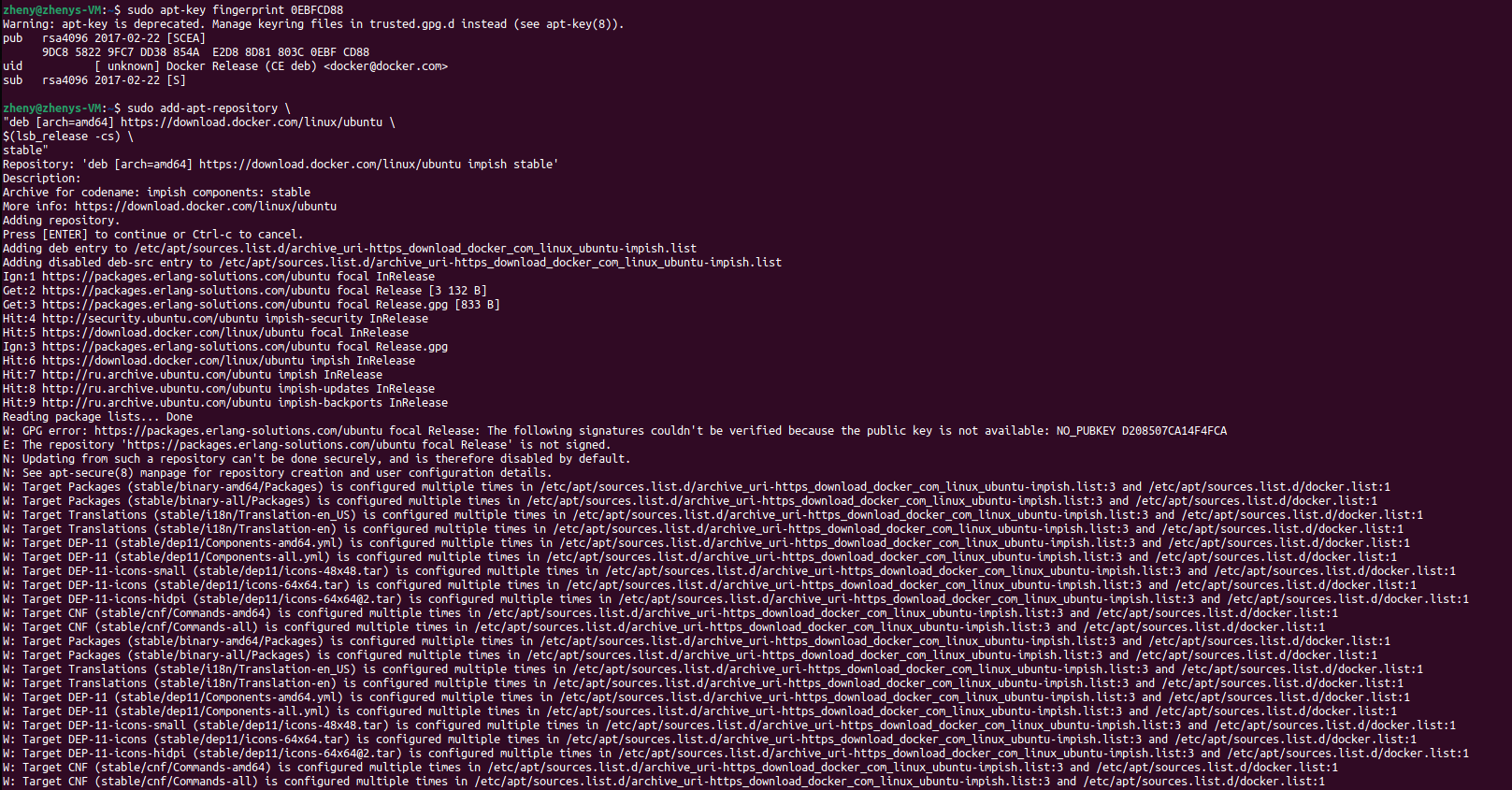


Рисунок 2 – Вывод команды добавления репозитория

После этого можно перейти к установке Docker Engine – Community, для этого:

1. Обновляем индексацию пакетов

*$ sudo apt-get update*

2. Устанавливаем последнюю версию docker-ce (собственно, Docker Engine – Community), docker-ce-cli (интерфейс клиента) и containerd.io (исполняемая среда для запуска контейнеров).

$ *sudo apt-get install docker-ce docker-ce-cli containerd.io*

После, необходимо убедиться, что Docker Engine – Community установлен корректно и работоспособен. Следующая команда загружает тестовый образ в контейнер и запускает его на выполнение.

*$ sudo docker run hello-world*

Пример результата выполнения этой команды приведен на Рисунке 3. Он показывает характерную последовательность из 4 шагов.

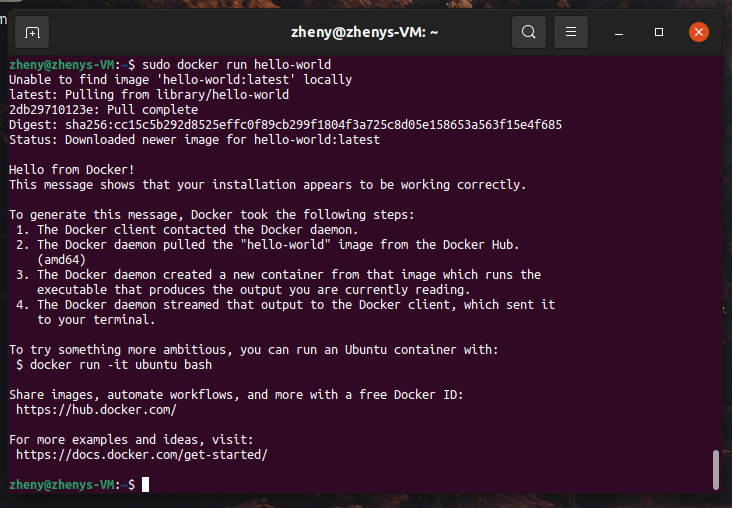


Рисунок 3 – Проверка работоспособности Docker Engine – Community

Далее устанавливаем средство docker-compose и проверим установленную версию. Сначала по ссылке определяем номер последней версии. К примеру установим версию 1.29.2.

[*https://github.com/docker/compose/releases*](https://github.com/docker/compose/releases)

Это значение будет фигурировать в последующей команде, Рисунок 4.

*$ sudo curl -L*

*"https://github.com/docker/compose/releases/download/****1.29.2****/docker-compose-$(uname -s)-$(uname -m)" -o /usr/local/bin/docker-compose*

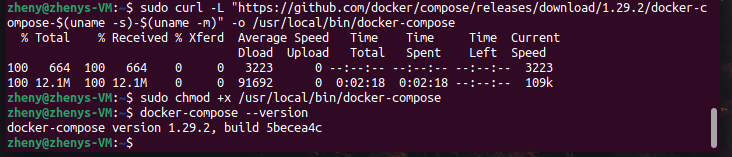


Рисунок 4 – Вывод установки и проверка установленной версии Docker-compose

В процессе установки была создана группа docker, но пользователи в нее не добавлены. Поэтому для ввода любой docker команды необходимо использовать sudo с последующим вводом пароля, т.к. Docker предполагает взаимодействие только с привилегированным пользователем root.

Осуществим вывод команды docker run hello-world без sudo после перезагрузки ВМ, чтобы убедиться, что все работает корректно, пример представлен на Рисунке 5. Docker первоначально не смог найти локальный образ hello-world, поэтому он загрузил образ из Docker Hub, который является репозиторием по умолчанию. После того как образ был загружен, Docker создал контейнер из образа, а приложение внутри контейнера было исполнено, отобразив сообщение.



Рисунок 5 – Вывод команды docker run hello-world без sudo после перезагрузки ВМ

**Дополнительно задание:**

Добавить в группу docker непривилегированного пользователя.