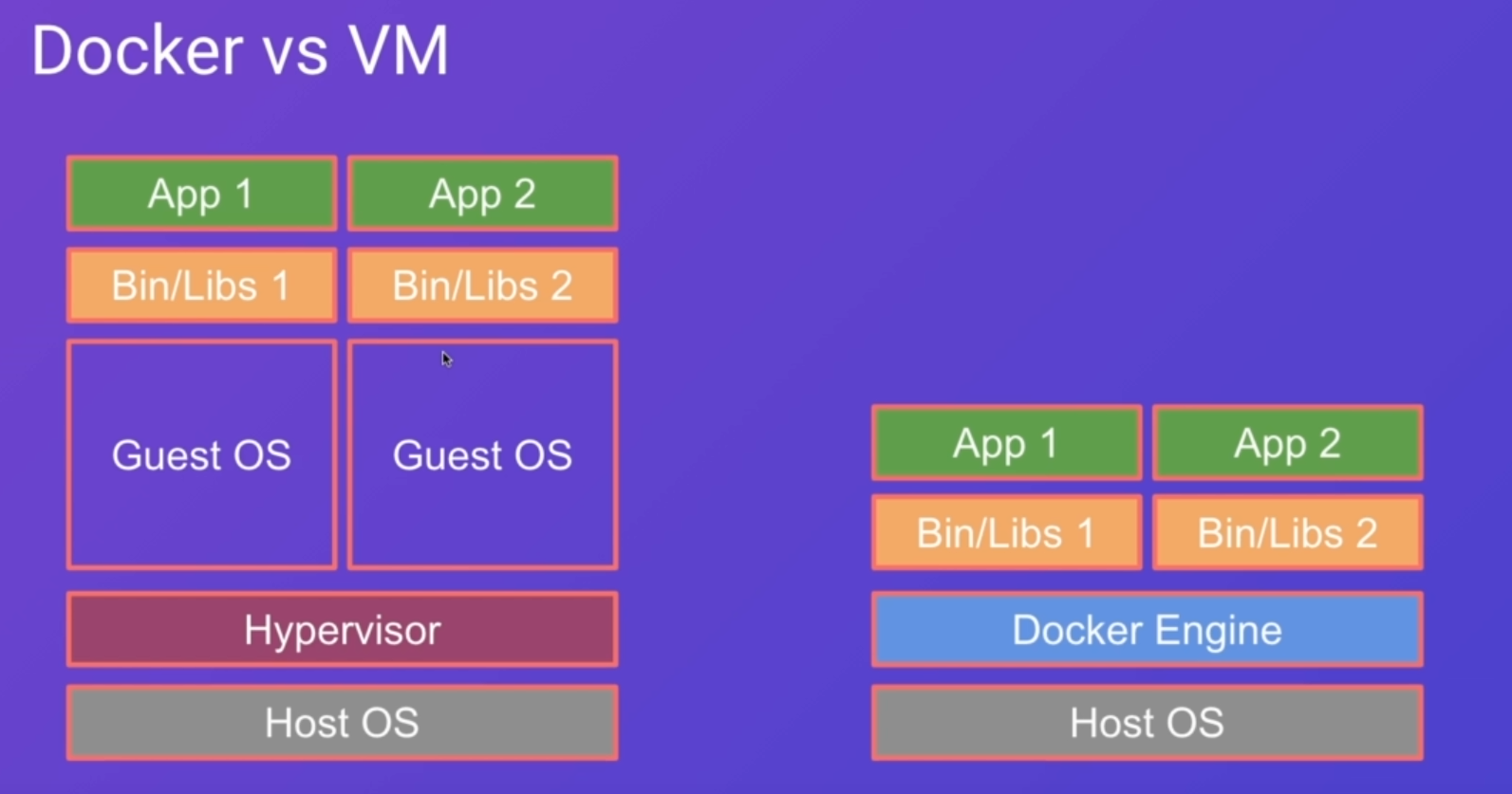
***DOCKER***



Сравнение с виртуальной машиной.

У нас есть хостовоя ОС на которой установлен Docker Engine или Hypervisor (гипервизор тоже может быть хостовой ОС).

Далее на Hypervisor-е запускаются гостевые ОС, которые внутри уже содержат бинарники и библиотеки необходимые для запуска приложения и само приложение. Если необходимо запустить два приложения, запускается дополнительная гостевая ОС, на которую уже размещают бинарники и библиотеки второго приложения и само второе приложение.

У Docker-а на хостовую ОС устанавливается Docker Engine (один) и на него устанавливаются библиотеки и приложения, при этом нет прослойки в виде гостевой ОС.

Docker не имеет ничего общего с виртуальной машиной (если только изолированность).

рассмотрим как работают процессы в линукс

есть процесс, в котором мы сейчас находимся и мы хотим запустить процесс на node(напр)что будет происходить, когда мы запустим процесс на node:

1. создастся новый процесс при этом скопировав предыдущий(дешёвая операция)был баш мы делаем форк и появляется второй баш, далее как только до него доходит очередь он запускается и подменяется необходимым процессом. это делается с помощью execv.
2. у нового процесса появится procesID и он будет функционировать в рамках этого процесса.

В данном случае процесс не изолирован, он может получать доступ к диску, может что-то удалить, а если другой процесс будет использовать ту же область диска, то он может удалить нужную ему информацию.

В линуксе предвесником изоляции была команда chroot

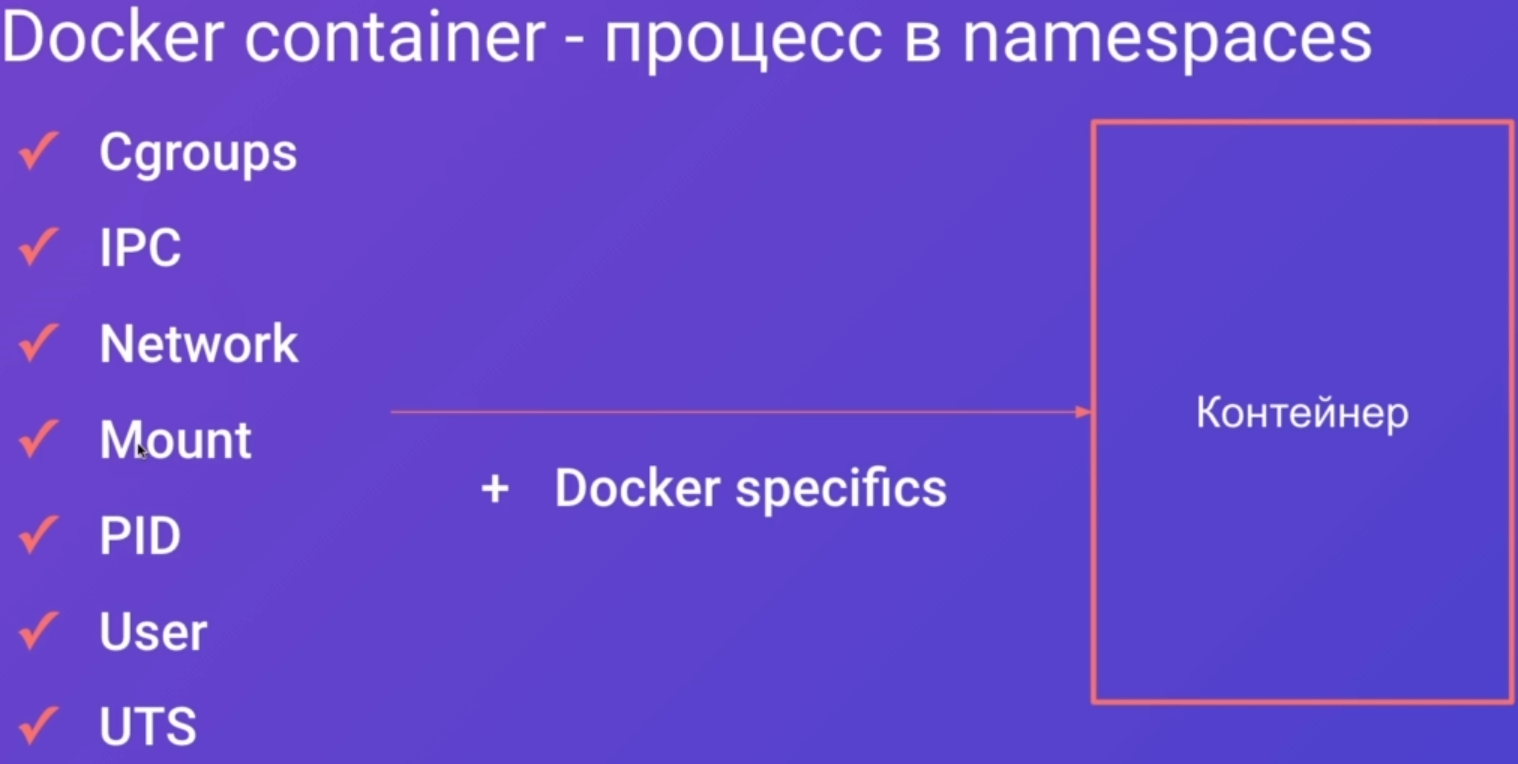
позволяет сменить корневую директорию

docker для изоляции использует неймспэйсы.

Но по аналогии можно сказать, что мы запустим процесс, а затим сменим ему рутовую директорию на к-л кастомную и он не будет видеть другие бинарники и бибилиотеки внутри своего изолированного пространства. В рамках этой изоляции мы можем скопировать другие бинарники и запускаться с ними в рамках этой изоляции не получая доступ к дисковому пространству.

что есть неймспэйсы?

когда запускается контейнер создается новый неймспейс. неймспейс имеет несколько наборов это



Cgroups управляет ограничениями по памяти и процессору для конкретного namespace.

IPC – коммуникация между процессами

Свой Network

Mount – говорит какие директории доступны какие нет.

processID который может повторяться с processID текущего Если мы запускаем на хосте processID должен быть уникален, если мы запускаем в namespace, то processID может быть свой

User может быть аналогичный и UTS

Кроме всего этого есть специфичные для docker обвязки

9:45