Изображения, мы в наш локальный репозиторий скачали, мы его видим здесь, радуемся. Как теперь посмотреть их через Docker SDK?

На этот счет у нас есть команда ImageList всего с двумя аргументами, контекстом и ImageListOptions, включающий в себя ровно 2 поля. Поле All, то есть поле All нам покажет, стоит ли включать в себя, так называемый, Intermediate Image. То есть те Image-и, которые ускоряет Docker build, за счет кэширования внутри Docker-а определенных слоев. То есть эти слои по умолчанию не показываются. Показываются только финальные изображения. Вы можете включить их отображения с помощью вот этой команды.

И команда filters. Команду filters можно посмотреть в вывод команды…, в help-е команды Docker Images. Опять же, команда Docker Images – это та самая команда, которая является оберткой над вот этой вот…, над вот этой вот командой, Docker SDK. И в ней есть вот этот All как раз, показать все изображения. Digests – это форматирование, следовательно, атрибут Docker SDK, Docker и клиента. Есть команда filter, которая как раз делает нам, объясняет нам, что происходит. Если мы прокрутим чуть вниз, она показывает, что filtering – это, соответственно, позволяет нам как-то отфильтровать отображения.

Опять же, вот сейчас фильтры поддерживаются. То есть dangling – это изображение без репозиториев и тэгов. То есть, те изображения, которые вы не (НРЗ 01:38), не считаются, так сказать, болтающимися без дела, если даже они у вас используется. То есть, это такая терминология Docker-а. Есть label-ы, когда вы можете повесить label на изображении, по ним фильтровать. И можно отфильтровать before, since. То есть от определенного изображения или до определенного изображения, какие были build-ы.

Соответственно, я считаю, что фильтры – это не очень удачная находка, потому что они довольно громоздкие и сложные. Вот, например, здесь они реализованы, то есть, сами фильтры, сами по себе. Args – это вот такая вот структура. И, конечно, ей просто в таком чистом виде не очень удобно, поэтому у нее есть куча всяких настроечек для того, чтобы их передавать дальше в client. И вот таким образом ими можно пользоваться, что, в принципе, уже намного удобнее. И хотя бы позволяет вам как-то там фильтровать. Но пять же, я фильтрами использую крайне редко. Мне проще вывести список всех параметров, потом уже по ним сделать какой-то поиск.

Соответственно, они передаются в ImageList, и ImageList возвращает нам slice из Image summary. Содержащий все контейнеры, привязаные к этому изображению: дату создания и ее ID-шник. ID-шник – это, вот как раз этот – SSH. То есть, вот это вот Image ID, но длинный. Да? То есть, полноценный, hash. Parent ID, если вы собрали с какого-то, ну…, если она имеет какую-то зависимость. Соответственно, Digests репозиториев, tag-и репозиториев, размер. И в принципе, это все.

То есть, сейчас здесь я вывожу репозиторий, tag репозитория. Это будет, вот это вот название. И, то есть название, соединенное вместе с тегом. И image ID. То есть, если мы его запустим, мы увидим, как раз, что здесь у меня postgres: latest, alpine: latest, mongo: latest совпадает с тем, что здесь выведено, их размеры указаны и их как раз ID-шники. Если я поставлю debug, чтобы посмотреть на все поля, то, что я здесь вижу – я здесь вижу: контейнеры, контейнеров, вот, например, к изображению myserver. К вот этому у меня нету привязанных сейчас. Parent ID у него тоже нет, за отсутствием родителей. Есть дата создания, есть его ID-шник, есть его размер. И соответственно, этим я могу пользоваться для того, чтобы выводить и показывать мои изображения.