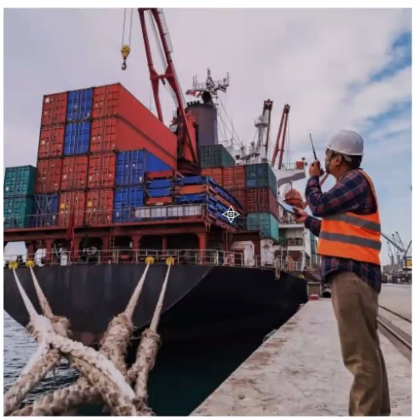
What is Docker? 
"DOCKER" refers to several things. This includes an open-source community project 
which started in 2013; tools from the open-source project; Docker Inc., the company that 
is the primary supporter of that project; and the tools that the company formally supports. 
Docker as a •Companf 
Docker as a -Product' 
Docker as a •Platform 
Docker as a 
"CLI Toor 
Docker as a -Computer Program" 
docker 

Docker liman işçisi, konteynır ları yönetiyor:



Bilgisayar alanında da konteynır çalıştırma engine idir docker.

Konteynır nedir?

Арр 
DEPS 
PYTHON 
BUNTU 

Developer python da geliştireceği bir uygulama çalıştıracak. Neye ihtiyaç var, bir bilgisyar ve bir işletim sistemine ihtiyaç var.

Daha sonra python a ihtiyaç var indirdi. Kütüphaneleri indirdik, dependencyleri de indirdi. Uygulama artık çalışıyor.

Şimdi tester lara gidecek ve onlar test edecek ancak uygulama çalışmıyor.

Ok 
oe 

Bu sorun yıllarca yaşanmış bir sorundur. Sistem ve version farklılıklarından kaynaklı bir çok uyuşmazlık sorunu oluşuyor.

Tester da ubuntu yok, sanal makine kurdu. Kütüphaneleri indir dediniz, indirdi. Ancak yine de çalışmadı uygulama. Kütüphaneler farklı versiyon da indirilmiş, onlar düzeltildi vs. Uzun uğraşlar..

Ya da müşterilere kendi bilgisayarınızı verme durumunuz yok.

Bir diğer sorun da; çalışan bir uygulama bir ay sonra çalışmıyor, bir bakıyorlar işletim sistemine güncelleme gelmiş. Oysa uygulama önceki sürüme göre yapılmıştı.

İşte konteynır bu sorunu çözüyor. Herhangi bir programı çalıştırmak için ihtiyaç olan ne varsa konteynır bunları içerir, işletim sistemi dahil.

APP 
LIB 
DEPS 
Containerized Application 
PYTHON 
BUNTIJ 

Bu uygulamalara da containerized application denir.

Ancak işletim sistemleri yer kaplar. Örneğin ubuntu 2 GB, paketlenmez, gönderilmez.

İşletim sistemi içinde ne var? Kernel, media player, browser, calculator vb.

Oysa bizim bunlara, kernele de ihtiyacımız yok. UYgulamanın çalışması için hangi kaynaklara ihtiyaç varsa sadece onu alacaz.

LIB 
App 
p YTHON 
Container 
Operating System 
Just required programs for application 
(without kernel) 

Peki kontenırı kim çalıştıracak? O program da docker; konteynır engine:

A pp 
LIB 
DEPS 
Containerized Application 
Container Engine 
Kernel 
CLARUSWAY 
p VT HON 
BUNTU 
DOCKER 
FULL OS 

İşletim sistemi içinde kernel ihtiyacımız yok ancak işletim sisteminin çalışabilmesi için kernel a ihtiyaç var.

lJBlJNTlJ 
OFTWARE 
FEDORA 
SOFTWARE 
CENTOS 
SOFTWARE 
DEBIAN 
SOFTWARE 
LINlJX KERNEL 
DONANIM 

Linux kernel üzerinde bildiğimiz üzere bir çok işletim sistemi var. Hepsinin ortak özelliği aynı çekirdeği kullanmalarıdır. Linux Kernel, host bilgisayarın kernel ı kullanılıyor. Docker da sadece ubuntu nun software kısmını alıyor.

A container is a standard unit of software that packages up code and all its 
dependencies so the application runs quickly and reliably from one computing 
environment to another. 
Containerized Applications 
Docker 
Host Operating System 
Infrastructure 

Docker kurulduktan sonra istediğimiz kadar uygulama yani konteynır çalıştırabiliriz.

Docker bu sayede bir bilgisayarda çalışan uygulamanın başka bilgisayarda çalışmaması sorununu çözdü.

Başka hangi sorunu çözdü?

Docker dan önce de konteynır vardı, docker ise açık kaynak kodlu yapılınca, bir anda kitlelere yayıldı ve çok hızlı gelişti. Alternatifleri var ve artıyor ancak docker ı neredeyse herkes kullanır.

Docker'dan önce:

Bare Metal 
os 
АРР 
= 1.5 + 1.5 GB RAM 
= 0.5 СИ + 0.5 ИМ 
АРР 1 
os 
АРР 2 
os 
АРР З 
os 
18 СРИ 
18 GB RAM 
АРР 4 
os 
АРР 5 
os 
АРР 6 
os 
HARDWARE= 5 CPU + 5 GB RAM 

Bir şirketin toplam 6 programa ihtiyacı var. Programlar da 0.5 CPU ve 0.5 RAM yiyor, İşletim sistemi de 1.5 CPU ve 1.5 RAM yiyor. Yani bir programı kurmak için 2 GB ihtiyaç var.

Ne yapıyorlardı? Her program için ayrı bir server kullanılıyordu. Ve bunlar 5 GB lık serverlar, dolayısıyla her server da 3 er GB kaynak israfı oluyordu.

Peki aynı server a neden birden fazla uygulama konmuyordu? Birincisi güvenlik. İkincisi ise farklı iki uygulama aynı serverda olursa, biri bozulunca diğerini de bozar. Bozulan bir uygulama daha fazla CPU yemeye başladı ve server ı çökertti, diğer uygulama da gitti. Bu nedenle farklı serverlarda olur.

Test yapılacak, test için yeniden server alımı yapılacak.

Ya da işletim sistemlerinin tek tek güncellenmesi, güvenlik önlemlerin güncellenmesi vs.

Daha sonra sanal makine yani virtualisation çıktı:

Virtualisation 
os 
= 1.5 + 1.56B RAM 
APP -0.5 CPU* 0.5 RAM 
APP 1 
APP 2 
APP 3 
3 CPU 
3 GB RAM 
APP 4 
APP 5 
APP 6 
Hypervisor 
HARDWARE: 5 CPU + 5 GB RAM 
CLARUSWAYO 
Hypervisor 
Bare Metal 
Hypervisor 

Sanallaştırma ne demek? Hypervisor sanallaştırma programı; bu program işletim sistemini kandırıyor, 2.5-2.5 bölüyor ve Hypervisor üzerine birden fazla işletim sistemi kurulabiliyor. Geriye ne kaldı? Kaynak israfı ciddi ölçüde engellenmiş oldu.

Çökerse bölünmüş kısmın bir tarafı çöker, diğer tarafa zarar vermez.

Test için de bir uygulamayı başka bir sanal makineye sıkıştırıp yer açıp test yapılır.

Ancak burada da şu dezavantaj var: sadece bir uygulama için işletim sisteminin tamamını kullanıyoruz.

Container 
os 
АРР 
= 1.5 + 1.5 GB ИМ 
=0.5 CPU+ 0.5 ИМ 
АРР 4 
АРР 1 
АРР 5 
АРР 2 
Docker 
os 
АРР 6 
АРР З 
HARDWARE= 5 сри + 5 GB 

Docker da ise bütün uygulamaları farklı konteynırlar içinde çalıştırabiliyoruz. Izolasyon sağlanıyor ve bir uygulama diğerini etkilemiyor.

Ancak sanal makinenin yeri ayrıdır, konteynırın yeri ayrıdır. Konteynırı amazon da kaldırdığımız instance ile aynı amaçla kullanılmaz. Konteynır ın sanal makine ile çalışma mantığı uyumludur ancak aynı şekilde çalışmaz. Konteynır dediğimiz bir program içindir. Birden fazla program olmaz. Konteynır da sorun mu var? Konteynırı düzeltmeye çalışmazsınız, silersiniz yeni konteynır oluşturursunuz. Ancak instance için bunu söyleyemeyiz.

'0 • ontainer? 
Milk 
ARUSWAYO 

Süte mi ihtiyacınız var? Çiftlikle uğraşmazsınız, gider üst alır içer ve atarsınız.

Konteynıra bağlanılabilir, sanal makine gibi kullanılabilir ancak konsepti bu değil, böyle kullanılmaz.

Bare Metal, on-premise demek. Kendi kullandığımız makinemiz.

Unlike virtual machines where hypervisor divides physical hardware into parts, 
Containers are like normal operating system processes. 
Operating 
Mach'* 
em 
Docke r 
Host Operating System 
Infrastructure 
Virul 
Operating 
Hypervisor 
Infrastructure 

Hypervisor donanım üzerine kuruluyor, işletim sistemini kandırıyor. Üzerine fazla işletim sistemleri kuruluyor. Cloud providerlar da bize sanal makineleri bu şekilde kullandırıyor.

Docker ise OS üzerine kuruluyor.

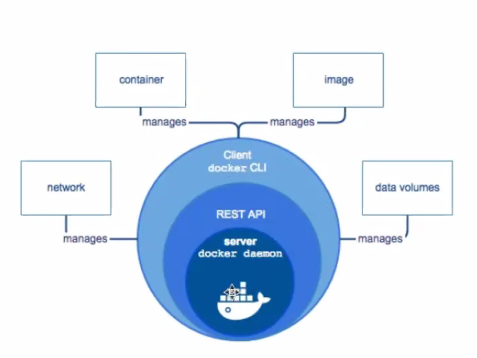
Virtual Machine 
Containers 

Docker vs. VMS 
Docker 
All containers share the same kernel of the host 
Containers instantiate in seconds 
Containers are lightweight (KBs/MBs) 
Less resource usage 
Can run many Docker containers on a laptop. 
Virtual Machines 
Each VM runs its own OS 
Boots uptime is in minutes 
VMS are ot few GBs 
More resource usage 
Cannot run more than a couple of VMS on an 
average laptop 
VM vs. Docker 
Startup 
.ARUSWAYO 

Docker ve SM farkları:

1. Bütün konteynırlar aynı kernel ı kullanır. SM de ise her SM kendi işletim sistemini ve kendi kernel ını kullanılr.
2. Konteynırlar saniyeler içinde ayağa kalkar. SM ler ise dakikalar sürüyor.
3. Konteynırlar çok hafiftir KB/MB ile ölçülür. SM iler ise GB larla.
4. Daha az kaynak kullanır, daha fazla kaynak kullanır.
5. Bir laptopta birçok docker konteynır kurulabilir. SM ise bir laptop ta en fazla birkaç tane kurulabilir.

Mimarisi:



Docker daemon programları çalıştıran kısmı

Docker CLI bizim kullandığımız ara yüz.

REST API ise CLI ile daemon arasındaki ilişkiyi kurar.

Images and Containers 
C] An image is a read-only template with instructions for creating a Docker container. 
O A container is a runnable instance of an image. 
Docker Image 
Docker Container 
Irnage (AM') 
Amazon EC2 

AWS de ami ler var ve onlardan bir tane instance ayağa kaldırıyoruz ve instance oluyor. Aynı şekilde Docker Image lerden de konteynır oluşturuyoruz.

Yani dosya haline image çalışan haline konteynır deniyor.

Dockerfile kullanımı basittir ve bununla docker image oluşturulur.

Env*av'wnt 
Oocker imaW 
Container 
Oocker Ccmtainer 

File yazıp docker a veriyoruz o image i oluşturuyor. Dockerhub da docker imagelerin saklandığı yerdir, docker ın resmi registry sidir.

Q ubuntul 
Explore 
Repositories 
Organizations 
Filters 
Products 
C) Images 
C] Extensions 
C) Plugins 
Trusted Content 
C] Q Docker Official Image O 
Verified Publisher O 
C] @ Sponsored OSS O 
Operating Systems 
Linux 
Windows 
Architectures 
C] ARM 
1 • 25 of 10,000 results for ubuntu. 
ubuntu @ DOCKER OFFICIAL IMAGE 
Updated 1 3 days ago 
Help 
• 010K* 
Ubuntu is a Debian-based Linux operating system based on free software 
Linux PowerPC 64LE IBMZ 386 riscv64 186-64 ARM ARM 64 
websphere-liberty @ DOCKEROFHCIAL 
Updated 8 days ago 
± IOM. • 0293 
WebSphere Liberty multi-architecture images based on Ubuntu 18.04 
Linux PowerPC 64LE IBMZ 386 06-64 
open-liberty Q IMAGE 
Updated 8 days ago 
10M* 
• 059 
upgrade 
Best Match 
pulls: 
Last week 
Leann_more 
Pulls: 12.369 
Last week 
Pulls: 8040 
Last week 

AWS, Google ya da diğer şirketlerin de kendi registry leri var.

Docker run komutuyla image lar registry den çekilip bilgisayarımızda çalıştırıyoruz.

Şimdi ec2 instance kaldırdık ve Hands on daki kodlarla docker ı yükledik. Ancak docker komutları sudo ile çalışıyor. ec2-user ı docker grubuna ekliyoruz bu sayede sudo komutunu kullanmaya gerek kalmıyor:

sudo usermod -a -G docker ec2-user

Sonrasında da

newgrp docker

Komutunu giriyoruz.

Docker version komutuyla verisonumuzu görebiliyoruz artık:

Version : 
20.10.17 
API version: 
1.41 (minimum version 1.12) 
Go version: 
gol.18.6 
Git commit: 
a89bU2 
Built: 
Sat Dec 3 2022 
OS/Arch: 
Iinux/ amd64 
Experimental : 
false 
containerd : 
Version : 
1.6.8 
GitCommit : 
9cd3357b7fd7218e4aec3eae239db1f68a5a6ec6 
runc : 
Version : 
1.1.4 
GitCommit : 
5fd4c4d144137e991c4acebb2146ab1483a97925 
docker-init: 
Version : 
ø.19.ø 
GitCommit : 
de4ØadØ 
[ec2-user@docker-instance 

Her bir CLI in kullanma mantığı var. docker --help komuutuyla docker komutlarını inceleyebiliriz.

docker run -i -t ubuntu bash komutunu girdiğimizde docker ubuntu image i dockerhub dan çeker ve indirir.

docker run --help ile bu komutu inceleyebiliriz ancak çok uzun flag leri var.

docker run hello-world docker ın çalışıp çalışmadığını anlamak için kullanılan bir image dır.

image lokalde varsa direk çalıştırır. yoksa registry den çeker çalıştırır.

ec2-user@docker-instance# docker run hello-world 
Unable to find image 'hello-world: latest' locally 
latest: Pulling from library/hello-world 
2db2971Ø123e: Pull complete 
Digest: sha256:63421b18c1443a9a85139225293fae7541fb4Øb7832d9deff8Øb6a9a75ce36øa 
Status: Downloaded newer image for hello-world: latest 
Hello from Docker! 
This message shows that your installation appears to be working correctly. 

ikinci kez aynı komutu girdik bu kez dire çalıştırdı çünkü artık lokalde var:

ec2-user@docker-instance# docker run hello-world 
Hello from Docker! 
This message shows that your installation appears to be working correctly. 
To 
1. 
2. 
generate this message, Docker took the following steps: 
The Docker client contacted the Docker daemon . 
The Docker daemon pulled the "hello-world" image from the Docker Hub. 

docker run -i -t ubuntu bash komutu; -i ben demek, -t terminal demek. bu komutu girdiğimizde bizi ubuntu komutu içerisine alacak:

ubuntu içerisinde apt-get update komutunu giriyoruz.

root@a7fabØ51dafc : /# whoami 
root 
root@a 7fabØ51dafc : /# 

whoami dedik ve nerede olduğumuzu anladık; root tayız.

şimdi cd komutu yapıp root un home klasörüne gidiyoruz ve touch myfile.txt diyerek dosya oluşturduk.

touch myfile.txt 
Is 
myfile. txt 
exit' 

daha sonra exit dedik.

ve tekrar docker run -it ubuntu bash komutu girelim:

docker run 
-it ubuntu bash 
root@fbf822c584b2:/# cd 
Is 

cd / ls dediğimizde home klasörüne gittik ve boş olduğunu gördük. dosyamız yok. neden?

çünkü her seferinde yeni bir konteynır ayağa kaldırıyor. şu an bizde iki konteynır var

her run komutu yeni bir konteynır ayağa kaldırır.

docker ps komutu konteynırlerı listeler.

ec2-user@docker-instance# docker ps 
Usage: docker ps [OPTIONS] 
List containers 
Options : 
- -help 
--all 
- -filter filter 
- -format string 
--1ast int 
-1, 
- -latest 
-no-trunc 
q, --quiet 
, --slze 
Show all containers (default shows just running) 
Filter output based on conditions provided 
Pretty-print containers using a Go template 
Show n last created containers (includes all states) (default 
Show the latest created container (includes all states) 
Don't truncate output 
Only display container IDs 
Display total file sizes 
-1) 

docker ps -a dersek bütün konteynırları göreceğiz:

CONTAINER ID 
fbf822c584b2 
telet 
a7fabØ51dafc 
th 
44ba13efc3b9 
1 cohen 
39e82b7db2d6 
binson 
55282eebc7d3 
g_gauss 
docker 
COMMAND 
"bash" 
"bash" 
" / hello" 
" / hello" 
"bash" 
ps 
IMAGE 
ubuntu 
ubuntu 
hello-world 
hello-world 
ubuntu 
-a 
CREATED 
3 minutes 
8 minutes 
9 minutes 
ago 
ago 
ago 
STATUS 
Exited 
Exited 
Exited 
Exited 
Exited 
PORTS 
(0) 
(0) 
(0) 
(0) 
37 seconds ago 
3 minutes ago 
9 minutes ago 
11 minutes ago 
11 minutes ago 
12 minutes ago 
(130) 11 minutes ago 
WIES 
boring_cha 
goofy_yon a 
xenodochia 
magical_ro 
interestin 

her konteynır bir uygulama içindir. uygulama çalışıyorsa konteynır çalışır, uygulama çalışmıyorsa konteynır stop eder. bu nedenle uygulama için sürekli çalışmalıdır.

biz konteynır dan exit ile çıkarsak stop eder:

ec2-user@docker-instance# docker ps 
CONTAINER ID IMAGE 
COBWIAND CREATED 
ec2-use 
STATUS 
PORTS 
WIES 

docker ps yaptığımızda aktif konteynır yok.

ps process kavramından geliyor. process işlem demektir. Bir programın running instance ıdır. Nasıl ki ami nin çalışan ec2 instance ıdır. linux dünyasında ya da diğer işletim sistemlerinde de bir programın çalışır haline process denir.

örneğin linux ta her komut bir programdır sleep komutunu çalıştıralım :

[II 5837 
ec2-user@ip 
PID TTY 
3749 pts/9 
4485 pts/9 
5837 ptS/9 
5868 ptS/9 
ec2-user@ip 
[21 5907 
ec2-usertip 
PID TTY 
3749 pts/9 
4485 pts/9 
5837 pts,'9 
5907 pts/9 
5924 ptS/9 
ec2-usertip 
[31 5979 
PID TTY 
3749 pts/9 
4485 pts,'9 
5837 pts/9 
5907 ptS/9 
5979 ptS/9 
5989 pts/9 
ec2-user@ip 
steep 60 & 
-172-31-63-46:-S ps 
TIME 
basti 
bash 
sleep 
-172-31-63-46:-$ sleep 90 & 
-172-31-63-46:-S ps 
TIME cro 
bash 
bash 
sleep 
sleep 
-172-31-63-46:-$ sleep 60 & 
ps 
TIME cro 
bash 
bash 
sleep 
sleep 
sleep 
ps 
1 
-172-31-63-46:-s 

çalıştırıp ps dediğimizde sleep programının çalıştığını gözlemliyoruz. bir program çalıştığı zaman OS için bu process olur.

o halde docker ps dediğimizde bize çalışan process leri gönderiyor.

demek ki her konteynır da bir processtir OS için.

docker run ubuntu ls ya da whoami dediğimizde de ubuntu için bu komutlarını çalıştıracak:

ec2-user@ip-172-31-63-46:-$ docker run ubuntu Is 
bin 
boot 
dev 
etc 
home 
Lib 
lib32 
lib64 
libX32 
media 
opt 
p roc 
root 
run 
sbin 
sys 
usr 
var 
docker run ubuntu xoami 
root 

ec2-user@ip-172-31-63-46:-$ docker run ubuntu steep 19 
o (ec2-user@ip 
(ec2-user@ip 
CONTAINER ID 
(ec2-user@ip 
CONTAINER ID 
ORTS 
1546be678488 
sad 
(ec2-user@ip¯ 
CONTAINER ID 
[ec2 - user@ip 
-172-31-63 
-172-31-63 
IMAGE 
-172-31-63 
IMAGE 
ubuntu 
knuth 
-172-31-63 
IMAGE 
-172-31-63- 
-46 —IS newgrp docker 
-46 —IS docker ps 
CREATED STATUS 
-46 -IS docker ps 
CREATED 
"sleep 
6 seconds ago 
-46 -IS docker ps 
CCM•I.AND CREATED STATUS 
46 
PORTS 
STATUS 
NN4ES 
Up 5 seconds 
PORTS 

sol terminalde docker run ubuntu sleep 10 dedik, ubuntu da bir program çalışıyor şu anda. sağ terminalde docker ps dedik ve ubuntu konteynırı gösterdi. 10 sn bitince docker ps dediğimizde artık çalışan bir konteynır olmadığı için liste boş döndü. yani konteynır bir program çalıştırmıyorsa stop eder.

çalıştırdığımız her konteynır a docker bir id verir.

CONTAINER ID 
d0f2c20b01ad 
1546be678488 
e59a2797daf2 
800e7f10a8ab 
141f8f98b878 
bff30b9c753 
elb4043e02ao 
ec3d62ef2d92 
22a326a4b4df 
5918eab6bba4 
WAGE 
ubuntu 
ubuntu 
ubuntu 
ubuntu 
docker ps -a 
cot"AND 
"sleep 30" 
"sleep 19" 
"whoami" 
CREATED 
2 minutes ago 
3 minutes ago 
3 minutes ago 
4 minutes ago 
STATUS 
Exited 
Exi ted 
Exited 
Exited 
Exit ed 
Exited 
Exi ted 
Exi ted 
Exited 
Exited 
About a minute 
2 minutes ago 
3 minutes ago 
4 minutes ago 
ago 
ubuntu •bash 
ubuntu 
ubuntu 
ubuntu 
hello-world 
hell o -world 
"bash" 
"'hello" 
1 
15 minutes 
16 minutes 
19 minutes 
25 minutes 
26 minutes 
28 minutes 
ago 
ago 
ago 
ago 
ago 
ago 
(e) 
(0) 
(0 
(8) 
12 
16 
17 
20 
26 
28 
minutes 
minutes 
minutes 
minutes 
minutes 
minutes 
ago 
ago 
ago 
ago 
ago 
ago 
el oquent _ændel 
sad knuth 
epic _perlman 
stupefied_noyce 
compassionate hoover 
sleepy_mccartfiy 
priceless_bhabha 
recursing_pascal 
optimistic_edison 
dazz ling_neumann 

üçüncü sütun en son çalışan komutu gösterir. created ne zaman run dediğimiz, status durumu,

docker start - -help 
usage: docker start (OPTIONS] CONTAINER [CONTAINER... I 
Start one or more stopped containers 
Opt ions : 
- -attach 
- -detach-keys string 
i, - -interactive 
Attach STDOUT/STDERR and forward signals 
Override the key sequence for detaching a container 
Attach container's STDIN 
1 

docker start komutu bir ya da daha fazla stop eden konteynırı çalıştırıyor. bunun için konteynır id ile yazarız.

ec2-user@ip-172 31 63 
- - -46: -$ docker 
01b4043€02ao 
docker 
CONTAINER ID IRAGE 
CORUND 
01b4043emae ubuntu 
"bash" 
ecz-user@ip-172-31-63-46 $ 
start elb4043002ao 
ps 
CREATED 
23 minutes ago 
STATUS 
Up 16 seconds 
PORTS 
NAMES 
priceless_bhabha 

01b4043002a0 id li konteynırı start ettik ve ps le çalıştığını gördük. ancak yukarıdaki listeden komut sütununda sadece bash olanları çalıştırır.

docker start yazdıktan sonra uniq olana kadar id nin ilk birkaç elemanını girmemiz de yeterli olur (docker start 01b gibi)

docker ps -a dediğimizde:

ecz-user@ip-172-31-63 
CONTAINER ID WAGE- 
141f8f98b878 ubuntu 
el b4043eD2ao ubuntu 
ec2-user@ip-172-31 63 
-46:—$ docker ps 
CCH•IAND 
"bash" 
"bash" 
CREATED 
21 minutes ago 
25 minutes ago 
STATUS 
up 56 seconds 
Up 2 minutes 
CONTAINER ID 
d0f2c20b01ad 
1546be678488 
esga2797daf2 
80ee7f1øaaab 
141f8f98b878 
bff3c3b9c753 
01b4043Ø02ae 
ec3d62ef2d92 
22a326a4b4df 
5918eab6bba4 
IRAGE 
ubuntu 
ubuntu 
ubuntu 
ubuntu 
ubuntu 
ubuntu 
ubuntu 
ubuntu 
hello-world 
hello-world 
docker ps -a 
cot"AND 
"whoami" 
"Is" 
"bash" 
" /bin/bash" 
"bash" 
"bash" 
" 'hello" 
'hello" 
1 
CREATED 
8 minutes ago 
9 minutes ago 
9 minutes ago 
STATUS 
Exited (G) 
Exited (e) 
Exited (e) 
Exited (e) 
Up About a 
Exited (e) 
NMES 
compass ionate_hoover 
p r bhabha 
PORTS 
7 minutes ago 
8 minutes ago 
9 minutes ago 
10 
21 
22 
25 
31 
32 
34 
minutes 
minutes 
mi nutes 
minutes 
minutes 
minutes 
minutes 
ago 
ago 
ago 
ago 
ago 
ago 
ago 
16 minutes 
minute 
16 seconds 
Up 2 minutes 
Exited (G) 26 minutes 
Exited (e) 32 minutes 
Exited (G) 34 minutes 
ago 
ago 
ago 
ago 
ago 
UANES 
eloquent_mendel 
sad knuth 
epic_perlnan 
stupefied_noyce 
com.passionate_hoover 
sleepy_mccarthy 
priceless_bhabha 
recursing_pascal 
optimistic_edison 
dazz ling_neumann 
1 

artık bazılarının status u up, bunlar az önce start dediğim konteynırlar. En sonda da isimler görüyoruz. bunlar insanların anlayabileceği şekilde konteynırlara isim veriliyor, esprili isimlerle. ancak biz kendimiz de isim verebiliriz ancak uniq olmalı.

docker start ı isimle de kullanabiliriz.

docker 
eloquent_mendel 
ecz-user@ip-172-31-63 
CONTAINER ID 
dOf2c20b01ad ubuntu 
141f8f98b878 ubuntu 
01b4043002ae ubuntu 
ec2-user@ip-172-31-63-46:- 
start eloquent_mendei 
-46:-S docker 
•sleep 30ie 
"bash" 
" bash " 
ps 
CREATED 
le minutes 
23 minutes 
28 minutes 
ago 
ago 
ago 
STATUS 
Up 2 seconde 
Up 3 minutes 
up 5 minutes 
NME-S 
el oquent_mende 
compassionate_hoover 
p r iceless_bhabha 

biz nasıl isim verebiliriz?

docker run --name clarusway -it ubuntu bash komutuyla isim verebiliriz. burada isim clarusway

konteynır ın ismi değiştirilimez, yenisini kaldırırsın. kullanım mantığı böyledir.

stop etmeden konteynırın içinden shift+p+q ile çıkılır. ancak vs code u kapatma kısa yoludur. kullanırsanız vs code dan çıkar. vs code kendi kısayollarını önceliğe alıyor.

yeni terminal açıp öyle de devam edebiliriz.

konteynır ları nasıl siliyoruz docker --help e baktığımızda rm komutu görüyoruz:

docker rm <dockerid> girmemiz gerekiyor. fbf8 i siliyorum:

fbf822c584b2 
telet 
a7fabØ51dafc 
th 
44ba13efc3b9 
1 cohen 
39e82b7db2d6 
binson 
55282eebc7d3 
g_gauss 
fbf8 
CONTAINER ID 
s 
4a8cbeac3c96 
ann1 
a7fabØ51dafc 
Y Jonath 
44ba13efc3b9 
dochial cohen 
39e82b7db2d6 
cal robinson 
55282eebc7d3 
resting_gauss 
ubuntu 
ubuntu 
hello-world 
hello-world 
ubuntu 
IMAGE 
ubuntu 
ubuntu 
hello-world 
hello-world 
ubuntu 
"bash" 
"bash" 
" / hello" 
" / hello" 
"bash" 
48 minutes ago 
53 minutes ago 
54 minutes ago 
56 minutes ago 
57 minutes ago 
docker rm fbf8 
docker ps -a 
COI"AND 
"bash" 
"bash" 
" / hello" 
" / hello" 
"bash" 
$1 
CREATED 
5 minutes ago 
58 minutes ago 
About an hour i 
About an hour i 
About an hour i 
ec2-user@docker-instance# & : 

çalışan bir konteynırı sileceksek -f flagi eklememiz gerekiyor:

ec2-user@ip-172-31-63 
-46:-$ docker 
clarusway 
ecz-user@ip-172-31-63 
-46:-$ docker 
CONTAINER ID IMAGE 
CO"4AND 
df5d3d789b3e ubuntu 
"bash" 
141f8f98b878 ubuntu 
" tTsh" 
01b4043e02ao ubuntu 
"bash" 
ec2-user@ip-172-31-63-46:- 
rm -f clarusway 
ps 
CREATED 
3 minutes ago 
49 minutes ago 
54 minutes ago 
STATUS 
Up 3 minutes 
Up 29 minutes 
Up 31 minutes 
NANES 
romant ic benz 
compassionate hoover 
p r iceless_bhabha 

docker run

docker ps

docker rm komutları %95 ihtiyacımızı görür. docker tek başına kullanılmaz pek, kubernetes ile birlikte çalışır.

docker conatiner ls ile de çalışan konteynır ları görebiliyoruz. docker ilk başladığında komut yapısı daha basitti. bir anda çok hızlı büyüyünce bir çok obje geldi ve docker üçlü komut yapısına geçti.

docker conatiner ls de bu nedenle gelişti. best practice de budur.

docker image ls le de image ları görürüz:

ec2-user@ip 
REPOSITORY 
ubuntu 
hello-world 
ec2-user@ip 
REPOSITORY 
ubuntu 
hel to-world 
ec2-use@p 
-172-31-63 
TAG 
NAGE ID 
latest 
latest 
TAG 
latest 
latest 
-172-31-63-46 $ 
-46: -$ docker image Is 
CREATED 
74f2314aØ3de 2 weeks ago 
feb5d9fea6a5 17 months ago 
-172-31-63-46:-$ docker images 
IPAGE ID 
CREATED 
74f2314ae3de 2 web ago 
feb5d9fea6a5 17 months ago 
SIZE 
77.8Æ 
13.3kB 
SIZE 
77.8MB 
13.3kB 

docker images de bunu verir.

tüm stop etmiş konteynır ları nasıl sileriz? docker conatiner --help ile bakalım

port 
prune 
rename 
List port nappings 
Rermve all stopped 
Reråme a container 
or a specific 
containers 

prune komutu ile siliniyormuş.

docker container prune

ec2-user@docker-instance# docker container prune 
WARNING! This will remove all stopped containers . 
Are you sure you want to continue? [y/N] y 
Deleted Containers: 
4a8cbeac3c%e38ddØ7Øb487316ef26267176973a9c5bd3ff94e57b54b37eb68 
a7fabØ51dafcf1a5ebb9b6a4765ccØ389b4føcd32øøac2466e1e6d9db2fee4a7 
44ba13efc3b9e614246967a7øefd44føuøab47e93fdf69e5fc631b54Ø5daa42 
39e82b7db2d65eØ1c39dd5825d498baØdc336eØ47f488b5Ø2136b2Øbd2Ø3b734 
55282eebc7d3dc6b6baf771dØ37675fb69d2bØ359b87a3abØ8b7d39d197eac45 
Total reclaimed space: 42.28MB 
ec2-use 
cker-instance# & : 

stop olmuş bütün konteynır ları sildi.

çalışan konteynırı silmek için önce stop edip sonra rm ile silebiliriz.

docker container Is - help 
Usage: docker container Is (OPTIONS) 
List containers 
Aliases: 
Is. ps. list 
q, - -quiet 
s, - -size 
f, - -filter fitter 
--fornat string 
n, --last int 
l, - -latest 
- -no-t runc 
Show alt containers (default shows just running) 
Filter output based on conditions provided 
Pretty-print containers using a Go template 
Show n last created containers (includes all states) (default 
Show the latest created container (includes all states) 
-1) 
Don't truncate output 
Only display container IDs 
Display total file sizes 
ec2-user@ip-172 31 
- -63-46: —$ docker ps •aq 
141f8f98b878 
elb4043e02ao 
docker rm -f "docker 
-aq)l 

burada da bir trick komut var.

şimdi ubuntu image ile clarus konteynır oluşturduk ve vi editör açmaya çalıştık ama sonuç:

docker run --naæ clarus 
touch myfile 
bin boot dev etc home lib Iib32 Iib64 Iibx32 
vi myfite 
bash: command not found 
1 
-it ubuntu bash 
media mnt myfile 

vi editör de yok. Neden? Çünkü image lar mümkün olduğu kadar küçük olmalıdır. Ne yapacağız? önce apt-get update sonra da sudo apt install vim ile vim editörünü yükleyebiliriz.

inspect ile de konteynır hakkında tüm bilgileri alabiliriz. konteynır ismi ya da id si ile kullanılır.

docker container inspect <konteynırismi>

ubuntu 
latest 
hello-world latest 
ec2-user@ip-172-31-63 
"RepoTags": ( 
74f2314ae3de 2 weeks ago 
feb5d9fea6a5 
17 months ago 
docker image inspect 74 
77.8MB 
13.3kB 
Old' : "sha256: 74f2314a03de34aoa2d552b805411fc9553a02ea71c1291b815b2f6 
"ubuntu: latest" 
igests": I 
"Parent": 
"Conment- . 
"Created": , 
"Container": "298f60554671ae2f5bf43b9892526aaa221e8e93c9cee1ca68ef65f. 
"ContainerConfig": { 
"Hostname": "298f6€554671", 
"Domainname": 
"User": "" 

docker image inspect <continerid> inspect komutu image ile de çalışır.

image leri de şu şekilde siliyoruz:

ec2-user@ip 
REPOSITORY 
ubun tu 
hello-world 
ec2-user@ip 
-172-31-63 
TAG 
latest 
latest 
-172-31-63 
-46: -$ docker image Is 
CREATED 
IMAGE ID 
74f2314a03de 2 weeks ago 
feb5d9fea6a5 17 months ago 
-46: -$ docker image m feb 
SIZE 
77 .81% 
13.3kB 
Untagged: hello-world:latest 
Untagged: hello 
- world($ha256 : fb4€ 
Deleted: sha256. 
• feb5d9f ea€a5e960€aaggse879d862b825965ba48dee5acaab5 
Deleted: sha256 
:e07ee1baac5fae6a26f3ecabfe54a36d3Qf96afda318fe€a9 
docker ill 

eğer image içinde konteynır çalışıyorsa o image i silmez.

şimdi örneğin nginx yüklerken yüklemeyi beklemek istemiyorum, promptumu almak istiyorum.

docker run -d nginx ile sağlıyoruz ( -d flagini ekledik)

docker run -d nginx 
38312281d31d314dc384814fb3a098dfb78f7501ee6212bce29bc7f6c1934655 
ec2-user@ip-172-31-63 
CONTAINER ID MAGE 
38312281d31d nginx 
ec2-user@ip-172-31-63 
38 
ec2-user@ip-i72-31-63 
-46:-$ docker ps 
æ•VAND 
'docker -entrypoint. 
docker rm -f 38 
1 
docker ps 
CREATED 
8 seconds ago 
STATU 
up 7 

silerken docker rm -f id ile silebilyioruz nginx i.

ec2-user@ip-172-31-63-46:-$ docker container run •d -p 80:8e nginx 
5eaee55763f54537d821d78379c239b56221b1e6fd1123e8dcgaca847dfbc753 

bu komutla da nginx direk hazır. instance ın public ip sini girdik ve gördük:

A Notsecure 10026.53.116 
ClaruswayLLC... O Instances I EC... 
O CitHub docker Docker Hub O Online Course... 
Welcome to nginx! 
SDLC 
Excalidraw 
If you see this page, the nginx web server is successfully installed and 
working. Further configuration is required. 
For online documentation and support please refer to nginorg. 
Commercial support is available at ngin&GQm. 
Thank you for using nginx. 

peki biz bunun içine girip nginx i değiştirebilir miyiz. docker exec ile gireriz:

docker exec <id> pwd gibi

ec2-usereip-172-31-63-46. 
• —$ docker exec 
bin 
boot 
docker-entrypoint. d 
docker-ent rypoint. sh 
etc 
home 
Lib 
I ib64 
media 
opt 
proc 
root 
run 
sbin 
SYS 
us r 
var 
ec2-user@ip-172-31-63-46:-$ docker exec 
docker•exee 
wed Mar 15 UTC 2023 
ec2-use 
5ea Is 
Sea pwd 
5ea dar 

docker exec -it 5e bash 
bin boot dev docker•entrypoint.d docker•entrypoint.: 
bash: ps: con•nand not found 
1 
root@5eaee55763f5 : 

bu komutla da içine girdik bash terminaline ve baktık ps komutu bile yok, o kadar sadeleştirilmiş image ler.

root@5eaee55763f5:/usr/share/nginx/html# 
SOX. html index. html 
root@5eaee55763f5 : /usr/share/nqinx/html# 
r/ share/ nginx/html# 
hi 
root@5eaee55763f5 : [us r/ share/nginx/html# 
root@5eaee55763f5:—# cat /etc/os-release 
Is 
echo "hi" > index-html 
cat index. html 
cd 
PRETTY GW/Linux 11 (buttseye)" 
GW,'Linux" 
VERSION ID-"11" 
(bullseye)" 
seye 
ID—debian 
" https:/.4%w•w.debian.org/" 
"https://w„w.debian.org/support 
BUG REPORT URL-"https://bugs.debian.org/" 
1 

bu nginx in olduğu konteynıra apachi kurabilir miyiz? kurabiliriz ancak kurulmaz. konteynır böyle kullanılmaz. apachi lazımsa ayrı bir konteynır kurulur ve çalıştırılır.

Ramazan - Instructor 
5. Her ne kadar container içerisine birden fazla 
uygulama yüklenebilse de kullanım için yugun 
değildir. Her container bir uygulama için 
oluşturulur. (edited) 

Arkadaşların özet notları:

docker run -it ubuntu bash # docker ayağa kaldırır ubuntu cihazda bash açar.

docker ps # process statusu aktif olan containerları gösterir.

docker ps -a # process statusu ne olursa olsun tüm containerları gösterir.

docker start Containe ID # inaktif id li containerları tekrar run eder.

docker start name # inaktif name li containerları tekrar run eder.

docker rm container ID veya name # inaktif containerlari siler

docker rm -f container ID veya name # statusu ne olursa olsun containeri siler.

docker container ls # best practise listeleme yöntemi -> docker ps -a ile aynı

docker container prune # tüm stop etmiş tüm containerları siler.

docker rm -f $(docker ps -aq) # tüm containerları linux komutları ile siler

docker container inspect mosmant # containerın tüm infolarını gösterir.

docker image inspect 74f2314a03de ubuntu image infosunu gösterir. i = d ubuntu cont idsi

docker rmi 74f2314a03de # ubuntu image siler. içinde container varsa image silinmez.

docker image ls # image leri listeler.

docker run nginx # nginx imageli container kurar.

docker run -d nginx # detach modda yükler. termianli ele geçirmez.

docker container run -d -p 80:80 nginx # detach modda açar 80 portu açar ve çalışturır. public ip yazar çalıştığını görürürz.

docker exec container ID <command>#container içinde komutu çalıştırır. pwd yazıp dir. görürür

docker exec -it containerID bash # container içine gireriz.

1.Her docker run yeni container demek. 2

. container bir uygulama içindir ve hep çalışmalıdır.

3. Her container bir process'dir.

4. imagelar olabildiğince küçük olmalıdır.

5. Her ne kadar container içerisine 1 den fazla uygulama yüklenebilse de kullanım için uygun değildir.Her container 1 uygulama için oluşturulmalıdır.

sudo usermod -a -G docker ec2-user

newgrp docker

# docker grubuna ekler.