



کانتینرها روشی برای انتقال اپلیکیشن‌های قابل‌حمل ایجاد می‌کنند. یعنی همانند ماشین‌های مجازی (VM) می‌توان آن‌ها را از یک محیط به محیطی دیگر انتقال داد، بدون این‌که مشکلی در اجرای آن‌ها وجود داشته باشد. با این تفاوت که کانتینرها سربار زیاد و وابستگی‌های ماشین مجازی را به همراه ندارد و حجم آن نیز کاهش چشمگیری پیدا می‌کند. سرویس‌ها و اپلیکیشن‌ها توسط کانتینر بسته‌بندی شده و به راحتی بین محیط‌های فیزیکی، مجازی و ابری منتقل می‌شوند. داکر یک کانتینر است که توسط شرکت Docker Inc. ایجاد شده و مدیریت می‌شود. داکر، عملکردهای کانتینر محلی در لینوکس را برداشته و از طریق خط فرمان یا یک API، آن‌ها را برای کاربر نهایی فراهم می‌کند. بسیاری از کامپوننت‌های (اجزای) اپلیکیشن اکنون به صورت آماده در کانتینرهای داکر موجودند و به راحتی می‌توان پشته نرم‌افزار را به عنوان اجزای جدا شده از هم توسعه داد. در واقع، در شناسایی سازگار بودن بخش‌های خارجی و داخلی با یکدیگر به ما کمک می‌کند. داکر، انقلابی در چگونگی توسعه اپلیکیشن‌ها به وجود آورد. در این مقاله از مرحله به مرحله نصب داکر تا ساخت کانتینر داکر برای وب سرور Apache را توضیح داده‌ایم.

نصب داکر

ما از اوبونتو به عنوان بنیاد **ساخت داکر** استفاده می‌کنیم. اوبونتو تنها یک توزیع محبوب و فراگیر از لینوکس نیست، بلکه خود تیم داکر نیز از اوبونتو برای توسعه استفاده کرده و **داکر** از نسخه 12.08 روی سرورهای اوبونتو پشتیبانی می‌کند و برای سادگی از دستورات نصب اوبونتو 16.04 استفاده می‌کنیم.

آماده کردن اوبونتو برای داکر

اولین گام به دست آوردن ورژن مناسب کرنل و هیدر آن است:

```
sudo apt-get install --install-recommends linux-generic-hwe-16.04 $
```

این فرآیند کمی زمان‌بر است و پس از انجام ممکن است به ریست سیستم نیاز باشد:

```
sudo reboot $
```

یا این‌که به ارتقای دیگر پکیج‌ها نیاز باشد:

```
sudo apt-get update $
```

```
$ sudo apt-get upgrade
```

نصب داکر در اوبونتو

نصب داکر در توزیع‌های دیبیا، اوبونتو فدورا، Raspbian و CentOS با استفاده از روش اسکریپت Shell آسان

است و با مراجعه به وبسایت

get.docker.com می‌توانید آن را دانلود کنید. برای این کار باید از دستور Curl استفاده کنید. جدیدترین نسخه Curl

به این صورت دریافت می‌شود:

sudo apt-get install curl

پس از نصب Curl اسکریپت را نصب و اجرا کنید:

curl -s <https://get.docker.com> | sudo sh

پس از نصب اسکریپت متنی مشابه شکل 1 مشاهده خواهید کرد که جزئیاتی درباره نسخه **داکر** و اجزای سرور و کلاینت دارد.

```
+ sh -c curl -fsSL "https://download.docker.com/linux/ubuntu/gpg" | apt-key add -qq - >/dev/null
+ sh -c echo "deb [arch=amd64] https://download.docker.com/linux/ubuntu xenial edge" > /etc/apt/sources.list.d/docker.list
+ [ ubuntu = debian ]
+ sh -c apt-get update -qq >/dev/null
+ sh -c apt-get install -y -qq --no-install-recommends docker-ce >/dev/null
+ sh -c docker version
Client:
Version:      17.12.0-ce
API version:  1.35
Go version:   go1.9.2
Git commit:   c97c6d6
Built:        Wed Dec 27 20:11:19 2017
OS/Arch:     linux/amd64

Server:
Engine:
Version:      17.12.0-ce
API version:  1.35 (minimum version 1.12)
Go version:   go1.9.2
Git commit:   c97c6d6
Built:        Wed Dec 27 20:09:53 2017
OS/Arch:     linux/amd64
Experimental: false

If you would like to use Docker as a non-root user, you should now consider
adding your user to the "docker" group with something like:

    sudo usermod -aG docker your-user

Remember that you will have to log out and back in for this to take effect!

WARNING: Adding a user to the "docker" group will grant the ability to run
containers which can be used to obtain root privileges on the
docker host.
Refer to https://docs.docker.com/engine/security/security/#docker-daemon-attack-surface
for more information.
serdar@ubuntu:~$ _
```

.10000
00 000
00000000
Shell
باید درباره
اضافه
کردن
کاربران
Root
نشده به
نکته‌ای
اشاره
کرد. انجام
این کار
راحت
است اما
پیشنهاد
می‌شود
برای کار
با **داکر** و
دیگر
عملکردها،
کاربر

روت نشده ایجاد کنید.

در ادامه عملیات نصب، کانتینر پایه **داکر** را تست کنید. دستور زیر ایمج اوبونتوی **داکر** را دانلود و در مسیر مشخص شده اجرا می‌کند:

```
sudo docker run -i -t ubuntu /bin/bash $
```

در صورت موفقیت‌آمیز بودن Hostname را از طریق Command Prompt خواهید دید که به چیزی شبیه #/:Root@216b04387924 تغییر یافته است. برای خارج شدن از این قسمت و هر بخشی در Shell از دستور Exit استفاده کنید.

حال یک **داکر** روی سرور خود دارید و با دستور زیر می‌توانید اطلاعات پایه‌ای درباره آن کسب کنید:
sudo docker info \$

خروجی دستور Docker Info تعداد کانتینرها و ایمج‌ها را نشان می‌دهد. ممکن است این دستور خیلی طولانی باشد که در شکل 2 قسمتی از آن را مشاهده می‌کنید.

```

Native Overlay Diff: true
Logging Driver: json-file
Cgroup Driver: cgroupfs
Plugins:
Volume: local
Network: bridge host macvlan null overlay
Log: auslogs fluentd gcplogs gelf journald json-file logentries splunk syslog
Swarm: inactive
Runtimes: runc
Default Runtime: runc
Init Binary: docker-init
containerd version: 89623f28b87a6004d4b785663257362d1658a729
runc version: b2567b37d7b75eb4cf325b77297b140ea686ce8f
init version: 949e6fa
Security Options:
apparmor
seccomp
Profile: default
Kernel Version: 4.10.0-42-generic
Operating System: Ubuntu 16.04.3 LTS
OSType: linux
Architecture: x86_64
CPUs: 2
Total Memory: 1.951GiB
Name: ubuntu
ID: H023:MDAH:FF1B:TJ2Z:W3YG:BHK5:SU53:MK54:7XXE:0XJG:UMQE:GXET
Docker Root Dir: /var/lib/docker
Debug Mode (client): false
Debug Mode (server): false
Registry: https://index.docker.io/v1/
Labels:
Experimental: false
Insecure Registries:
127.0.0.0/8
Live Restore Enabled: false

[END]

```

.2 □□□

□□□□□

□□□□□

Docker

Info

□□□□

□□□

□□□□□□□

□□□□□□

□□□□

□□□ □□□

در این قسمت اگر فایروال UFW در اوبونتو در حال اجرا است باید دستورات دیگری هم لحاظ شوند که از آنها عبور می‌کنیم.

کار با ایمج‌های داکر و کانتینرهای داکر

کانتینرهای داکر نسبت به ماشین‌های مجازی کارایی بسیار بیشتری دارند. وقتی کانتینر یک پروسه را اجرا نمی‌کند، انگار که آن پروسه کاملاً از کار افتاده و در واقع از منابع سیستم استفاده نمی‌کند. با استفاده از دستور Docker Ps می‌توان کانتینرهای فعال و غیرفعال را مشاهده کرد:

This command will show ALL containers on the system #

\$ sudo docker ps -a

This will show only RUNNING containers

\$ sudo docker ps

تمام دستورات موجود را به سادگی با وارد کردن Docker مشاهده کنید.

مطلب پیشنهادی



پلتفرمی برای مدیریت کانتینرها
کوبرنتس در مقابل داکر، کدامیک بهتر هستند

کانتینرهای داکر در مقابل ایمج‌های داکر

کانتینرهای داکر از روی ایمج‌ها ساخته می‌شوند که به Shell‌های سیستم‌عامل نیاز دارند که شامل کتابخانه‌ها و باینری‌ها برای اجرای اپلیکیشن‌ها در کانتینر است. ایمج‌ها با برجسب‌ها نام‌گذاری می‌شوند، به خصوص متاداده که ذخیره‌سازی و استفاده از نسخه‌های مختلف ایمج را ساده می‌کند. در حالت عادی ممکن است یک ایمج چند برجسب مختلف داشته باشد:

ubuntu:16.04, ubuntu:xenial-20171201, ubuntu:xenial, ubuntu:latest

زمانی که عبارت Docker Pull Ubuntu وارد می‌شود، ایمج اوبونتوی پیش‌فرض درخواست می‌شود که ایمجی با برچسب latest (آخرین) است. به عبارت دیگر، این عبارت مساوی با عبارت Docker Pull Ubuntu:latest است.

جریان کاری کانتینر و ایمج داکر

زمانی که یک ایمج را به کار می‌گیرید، یک کانتینر زنده از آن ساخته‌اید و این کار با دستور Docker Run انجام شده است. هنگام افزودن نرم‌افزار به کانتینر و اعمال تغییرات در تنظیمات با کمک دستور Docker Commit می‌توانید یک ایمج جدید از آن تغییرات بسازید.

نکته مهم و قابل‌توجه این است که **داکر** تنها تغییرات را ذخیره می‌کند. زمانی که ایمج خودتان را می‌سازید، تغییرات ایجاد شده نسبت به ایمج پایه ذخیره می‌شود و ایمج جدید به ایمج اصلی پیوند می‌خورد؛ بنابراین ممکن است ایمجی داشته باشید که به صورت مجازی حجم 266 مگابایتی دارد اما تنها چند مگابایت فضا را روی دیسک اشغال می‌کند.

ساخت یک ایمج جدید از کانتینر

برای درک بهتر این قضیه که ایمج و کانتینر چگونه کار می‌کنند، اجازه دهید یک سرور وب آپاچی (Apache) راه‌اندازی کنیم.

شروع کار با یک کانتینر داکر جدید

در ابتدا باید یک کانتینر جدید بسازید که راه‌های کمی برای این کار وجود دارد. یک Root Shell در کانتینر جدید آغاز کنید:

```
sudo docker run -i -t --name apache_web ubuntu /bin/bash $
```

این دستور یک کانتینر جدید با شناسه یکتا و نام Apache_Web می‌سازد. دستور بعدی apt-get است که برای نصب وب‌سرور آپاچی استفاده می‌شود:

```
root@d7c8f02c3c8c:/# apt-get update
```

```
root@d7c8f02c3c8c:/# apt-get install apache2
```

خروجی عادی Apt-Get ظاهر خواهد شد و بسته Apache2 روی کانتینر جدید شما نصب شده است. پس از کامل شدن نصب، آپاچی را اجرا و Curl را نصب کرده و نصب را تست کنید:

```
root@d7c8f02c3c8c:/# service apache2 start
```

```
root@d7c8f02c3c8c:/# apt-get install curl
```

```
root@d7c8f02c3c8c:/# curl http://localhost
```

در ادامه آخرین دستور باید صفحه آپاچی پیش‌فرض را به صورت HTML خام در کنسول ببینید. این به آن معنا است که سرور آپاچی نصب شده و در کانتینر شما اجرا می‌شود.

ساخت یک اسکریپت استارت‌آپی برای کانتینر داکر

به یاد داشته باشید که **کانتینر داکر** تنها تا زمانی کار می‌کند که پروسه یا پروسه‌های آن فعال باشند. پس اگر پروسه‌ای که اجرا کرده‌اید، به پس‌زمینه منتقل شد، داکر کانتینر را متوقف خواهد کرد.

یک اسکریپت بسازید؛ smartapache.sh در مسیری که مشخص کرده‌ایم:

```
You might need to first install Nano inside the container #
```

```
root@d7c8f02c3c8c:/# apt-get install nano
```

```
root@d7c8f02c3c8c:/# nano /usr/local/sbin/startapache.sh
```

در فایل startapache.sh این خطوط را اضافه کنید:

```
bin/bash!#
```

```
./etc/apache2/envvars
```

```
/usr/sbin/apache2 -D FOREGROUND
```

تغییرات را اعمال و فایل را ذخیره کرده و سپس آن را اجرا کنید:

```
root@d7c8f02c3c8c:/# chmod +x /usr/local/sbin/startapache.sh
```

به کار گرفتن کانتینر برای ساخت یک ایمج داکر جدید

تغییرات اعمال شده را ذخیره کنید:

```
sudo docker commit apache_web local:apache_web $
```

این کار به عنوان یک ایمج جدید حکم تأیید ذخیره کانتینر و برگرداندن یک شناسه یکتا را دارد.

مطلب پیشنهادی



اجرا و ساخت چندین نرم افزار روی یک سیستم عامل داکر؛ نهنگی که به برنامه سازی سرعت می دهد!

پایه های شبکه ای داکر

حال ایمج خود را دارید و می توانید کانتینر خود را راه اندازی کنید. پس از نصب داکر سه شبکه مجازی ایجاد می شود که ممکن است توسط کانتینرهای داکر استفاده شوند:

Bridge: شبکه ای که کانتینرها به صورت پیش فرض به آن متصل می شوند. Bridge Network به کانتینرها اجازه می دهد، به صورت مستقیم با یکدیگر صحبت کنند، نه با سیستم Host. شبکه ای که به کانتینرها اجازه می دهد توسط host مستقیم دیده شوند.

None: شبکه ای خالی است و کانتینری که به این شبکه متصل شده چیزی جز خودش را نمی تواند ببیند.

زمانی که کانتینر را اجرا می کنید و نیاز است که با دیگر کانتینرها یا جهان بیرونی ارتباط برقرار کنید، باید به صورت دستی پورتها را از آن کانتینر به Host نگاشت کنید. در ادامه مثالی که ذکر کردیم، می توانید هنگام اجرای کانتینر، این کار را در خط فرمان انجام دهید:

```
sudo docker run -d -p 8080:80 --name apache local:apache_web /usr/local/sbin/startapache.sh $
```

سوئیچ -p برای نگاشت پورت است. در اینجا پورت 8080 به Host با پورت 80 نگاشت شد. -name به شما اجازه می دهد تا یک نام برای کانتینر انتخاب کنید و البته اختیاری است.

می توان وضعیت کانتینر و نگاشت TCP را با دستور Docker Ps مشاهده کرد. همچنین با دستور Docker Port جستجو در نگاشت شبکه ممکن خواهد بود:

```
sudo docker port apache 80 $  
0.0.0.0:8080
```

داکر برای مک و ویندوز

کانتینرهای داکر یک فناوری مشخص از لینوکس هستند و اما می توانید به راحتی آن را در سیستم عامل Mac یا ویندوز (با استفاده از مجازی سازی) به کار گیرید. شرکت داکر نسخه هایی را منتشر کرده که در محیط سیستم عاملها اجرا شوند. طریقه کار آن همانند چیزی است که بررسی کردیم و خط فرمانی مشابه با لینوکس وجود دارد. نصب آنها شبیه سایر اپلیکیشنها است و آنها از یک هایپروایزر استفاده می کنند؛ Xhyve برای مک و Hyper-V برای ویندوز.

البته نسخه دستکاپی از داکر وجود دارد که یک محیط گرافیکی برای مدیریت تعاملات در نظر گرفته است (شکل 3). برای مثال، می توانید مشخص کنید که چه درایوهای محلی برای کانتینرها در دسترس باشند، چقدر CPU یا حافظه در اختیار کانتینر قرار گیرد و رفتار ارتباطات شبکه ای چگونه باشد.

