

عنوان: نصب داکر

درس رایانش ابری

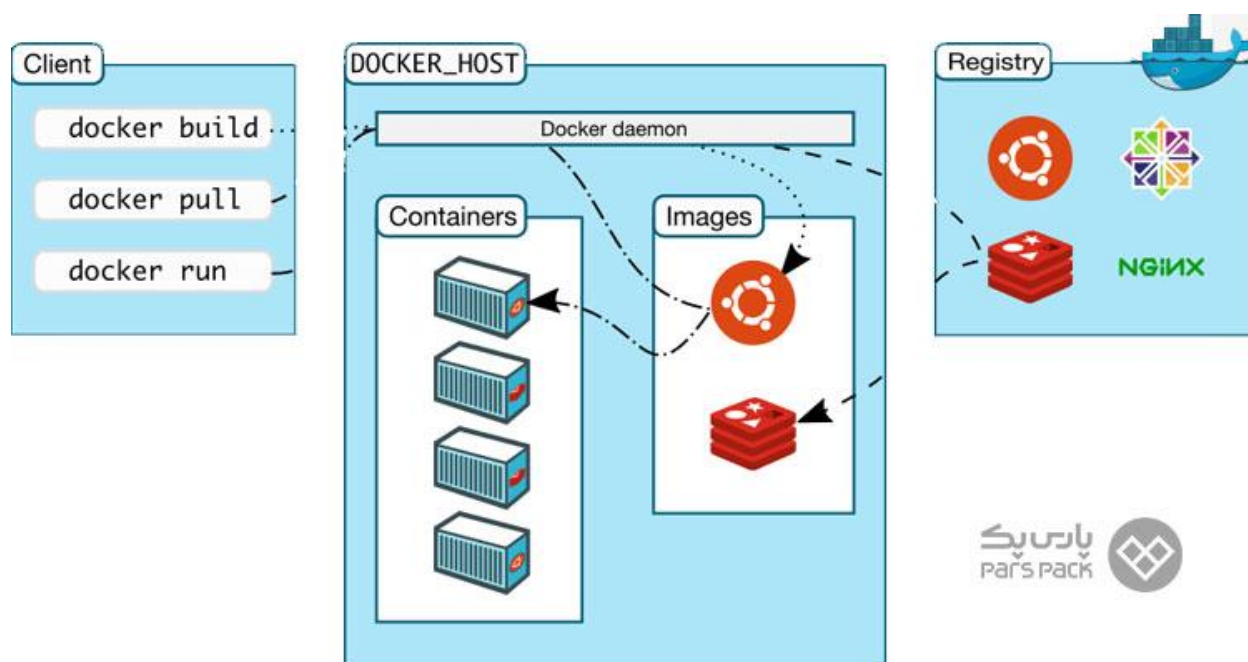
استاد راهنما: دکتر هدی طاهری

تهیه کننده: متین کاهی

نصب داکر

خب در ابتدا با نصب داکر (داکر اینجین) همراه شما هستیم تا بتوانیم پروژه ای که مینویسیم داکرایز کنیم اما قبل از اینکه نصب داکر رو شروع کنیم بیاید یک تعریف کلی از داکر داشته باشیم:

داکر چیست؟ داکر Docker برنامه‌ای رایانه‌ای و ابزاری است که ایجاد، توسعه و اجرای اپلیکیشن‌ها را با استفاده از کانتینر (container) آسان می‌کند. کانتینر به توسعه‌دهندگان این امکان را می‌دهد که تمام پیش‌نیازها و نیازمندی‌های برنامه خود را برای استفاده و اجرا جمع‌آوری کنند؛ مانند کتابخانه‌ها (Libraries) و زیرساخت‌های لازم.



همانطور که در تصویر مشاهده میکنید داکر به صورت کانتینر است که دستوراتی مانند **build, pull** دارد.

حال که تعریفی از داکر را متوجه شدیم بیاید به نصب داکر در **ubuntu** و توزیع های لینوکس بپردازیم.

در ادامه شاهد آموزش نصب داکر خواهیم بود

البته این نکته را بگویم داکر را میشود در ویندوز هم نصب کرد ولی از انجایی که در لینوکس معمول تر است ما آموزش نصب داکر در لینوکس را میدهیم

نصب داکر: در وب سایت رسمی داکر آموزش کاملی از مراحل نصب داکر چه در ویندوز و چه در لینوکس و چه در دیگر سیستم عامل ها داده شده است که میتوانید از لینک زیر مشاهده کنید:

<https://docs.docker.com/engine/install>

خب همانطور که در این وبسایت مشاهده میکنیم

داکر برخی پیش نیاز هایی دارد که باید قبل از نصب داکر نصب شود تا بتواند به خوبی کار کند

پیش نیاز هایی مثل, curl, ca-certificates

برای توزیع اوبونتو برای نصب پیش نیاز های داکر به دو شیوه میشود عمل کرد که یکیش برای ورژن های قدیمی است و یکیش برای ورژن های جدید اوبونتو

برای ورژن های جدید:

```
#Add Docker's official GPG key:
```

```
sudo apt-get update
```

```
sudo apt-get install ca-certificates curl
```

```
sudo install -m 0755 -d /etc/apt/keyrings
```

```
sudo curl -fsSL https://download.docker.com/linux/ubuntu/gpg -o /etc/apt/keyrings/docker.asc
```

```
sudo chmod a+r /etc/apt/keyrings/docker.asc
```

```
#Add the repository to Apt sources:
```

```
echo\
```

```
" deb [arch=$(dpkg --print-architecture) signed-by=/etc/apt/keyrings/docker.asc] https://download.docker.com/linux/ubuntu\
```

```
$ (. /etc/os-release && echo "$VERSION_CODENAME") stable\ | "
```

```
sudo tee /etc/apt/sources.list.d/docker.list > /dev/null
```

```
sudo apt-get update
```

برای ورژن های قدیمی از کد `bash script` زیر استفاده میکنیم:

```
for pkg in docker.io docker-doc docker-compose docker-compose-v2 podman-docker containerd runc; do sudo apt-get remove $pkg; done
```

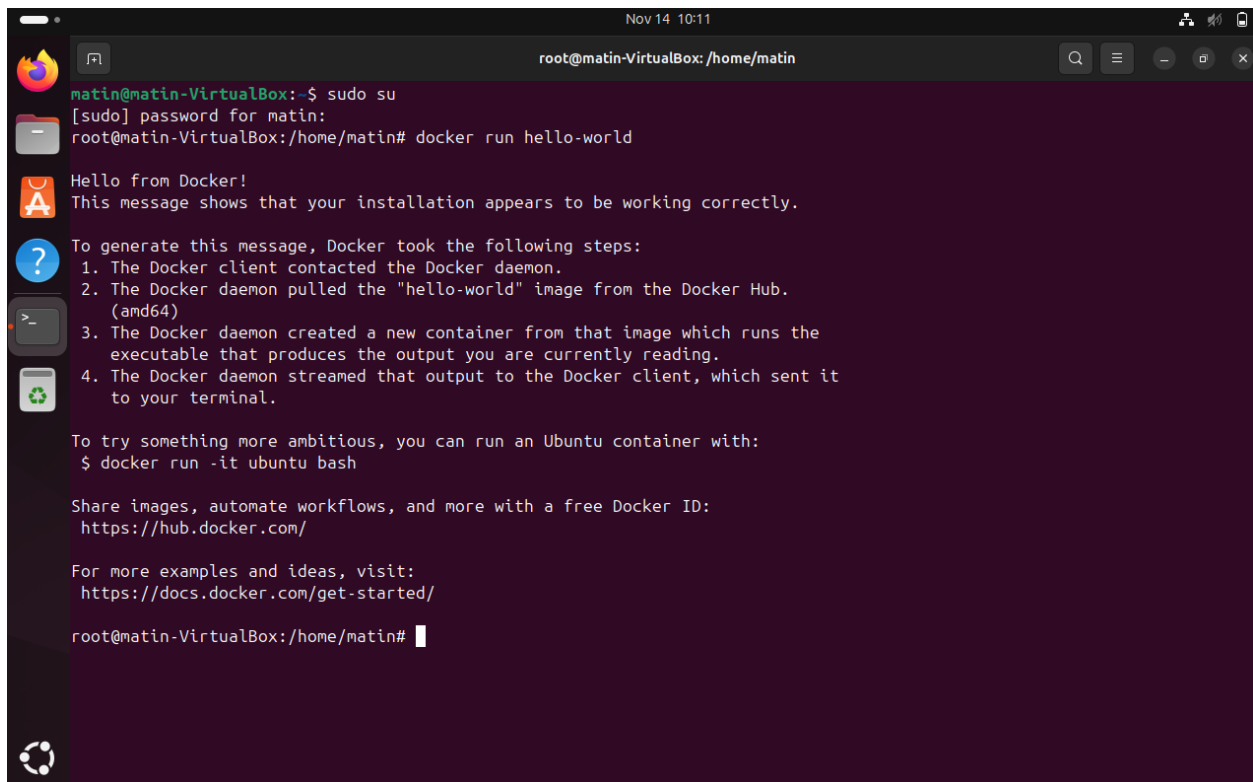
پس از نصب پیش نیاز ها نوبت به نصب داکر است که با استفاده از دستور زیر اقدام به نصب داکر میکنیم

```
sudo apt-get install docker-ce docker-ce-cli containerd.io docker-buildx-plugin docker-compose-plugin
```

بعد ازین برای اینکه بفهمیم نصب با موفقیت انجام شده کانتینر `hello-world` را صدا میکنیم

```
sudo docker run hello-world
```

که خروجی آن به شکل زیر است.

A screenshot of a terminal window titled "Nov 14 10:11" and "root@matin-VirtualBox: /home/matin". The terminal shows the command "sudo su" being executed, followed by "docker run hello-world". The output displays a "Hello from Docker!" message and a confirmation that the installation is working correctly. It then lists the steps Docker took to generate the message: 1. The Docker client contacted the Docker daemon. 2. The Docker daemon pulled the "hello-world" image from the Docker Hub. (amd64) 3. The Docker daemon created a new container from that image which runs the executable that produces the output you are currently reading. 4. The Docker daemon streamed that output to the Docker client, which sent it to your terminal. Below this, it suggests running an Ubuntu container with the command "\$ docker run -it ubuntu bash". It also provides links for sharing images and more examples. The terminal ends with the prompt "root@matin-VirtualBox: /home/matin#".

```
Nov 14 10:11
root@matin-VirtualBox: /home/matin
matin@matin-VirtualBox:~$ sudo su
[sudo] password for matin:
root@matin-VirtualBox: /home/matin# docker run hello-world

Hello from Docker!
This message shows that your installation appears to be working correctly.

To generate this message, Docker took the following steps:
1. The Docker client contacted the Docker daemon.
2. The Docker daemon pulled the "hello-world" image from the Docker Hub.
   (amd64)
3. The Docker daemon created a new container from that image which runs the
   executable that produces the output you are currently reading.
4. The Docker daemon streamed that output to the Docker client, which sent it
   to your terminal.

To try something more ambitious, you can run an Ubuntu container with:
$ docker run -it ubuntu bash

Share images, automate workflows, and more with a free Docker ID:
https://hub.docker.com/

For more examples and ideas, visit:
https://docs.docker.com/get-started/

root@matin-VirtualBox: /home/matin#
```

ممنون از دیدن شما.