

# آنچه درباره داکر باید بدانید

تهیه کننده : نریمان مандی

4	فصل اول
4	آشنایی به زبان ساده
4	مشکلی که docker سعی میکند حل کند چیست؟
4	چگونه docker کمک میکند تا مشکل بالا را حل کنیم؟
5	اما این container ها رو باید از کجا پیدا کرد؟
5	چگونه باید اپلیکیشن خودمون رو به یک container اضافه کنیم؟
6	آیا من فقط به یک container برای کل اپلیکیشنم نیاز دارم؟
6	آیا مدیریت کردن چند تا container به صورت همزمان کار سختی نیست؟
6	ابزار هایی که بهتر است بشناسید
6	Kitematic
6	Docker Swarm
7	فصل دوم
7	ساختار Docker
7	ساختار Docker
8	Docker Image
8	Docker Container
8	Docker Registries
8	Docker Deamon
9	فصل سوم
9	نصب Docker CE (Community Edition)
9	نصب Repository
10	نصب Docker CE
11	فصل چهارم
11	نصب Geoserver به کمک docker
11	هدف از ساخت docker image برای جنوسرور چیست؟
11	در صورت تغییر در تنظیمات ، اضافه کردن لایه ها ، استایل ها ، تنظیمات cache نقشه و ... آیا هریار که container ما پاک شود و دوباره راه اندازی شود ، اطلاعات از بین میرود؟
12	مرحله اول
12	winsent/geoserver
12	نحوه ساخت یک Docker Image از روی Dockerfile
12	نحوه pull کردن یک image از docker hub
12	دیدن لیستی از همه Image های داکری که روی هاست قرار دارد
12	حذف docker image
13	ساخت یک container از روی image

13	.....-d
13	.....-p 8087:8080
13	.....-v
13	.....نمایش لیست تمامی container های فعال
13	.....نمایش لیست تمامی container های فعال و غیر فعال
13	.....حذف container
13	.....start و Stop کردن container ها
14	.....Stop کردن همه container ها به صورت دسته جمعی و حذف آنها
14	.....روش اول
14	.....روش دوم
14	.....دستور رفتن به داخل container
14	.....خروج از یک container بدون stop شدن
14	.....ساخت Docker Image از روی Container
14	.....ساخت tar فایل از image
15	.....لود کردن tarFile
15	.....استفاده از docker compose
15	.....بارگزاری دیتای پیشفرض (نقشه ،لایه و ...) در هنگام ساخت container

## فصل اول

### آشنایی به زبان ساده

اولین نکته که باید مورد توجه قرار بگیره اینه که داکر هم مثل بقیه تکنولوژی ها تا وقتی که ازش دوریم خیلی سخت به نظر میاد اما وقتی شروع به کار کردن و مطالعه میکنیم ، میبینیم که چقدر ابزار کاربردی و راحتی است که در ادامه سعی دارم تمام سوالاتی که در ابتدا برای خودم مطرح بود رو در اینجا مطرح کنم و بهشون پاسخ بدم و در نهایت هم به عنوان یک مثال کاربردی نحوه بردن جنوسرور روی داکر که برای لود نقشه ها ، در فرمت های مختلف در سیستم جی آی اسی استفاده میشه رو باهم انجام بدیم.

#### مشکلی که docker سعی میکند حل کند چیست ؟

مشخصا داکر مشکلات زیادی را حل میکند اما یکی از مهم ترین آن ها `improve consistency` (بهبود ثبات یا استقرار ) میباشد  
وقتی گروهی از افراد روی یک پروژه ای کار میکنند ، میتونه تفاوت های در سیستم عامل ها و تنظیمات وجود داشته باشه که میتونه باعث بروز یک سری از مشکلات بشه و همچنین ممکنه در بالا آوردن اپلیکیشن روی کامپیوتر خودمون و محیط `production` (عملیاتی ) هم تناقضاتی وجود داشته باشه .  
تمامی اینها شما رو از حل کردن مشکلات اصلی که در حقیقت `build` کردن پروژه تون هست دور میکنه.

#### چگونه docker کمک میکند تا مشکل بالا را حل کنیم؟

Docker یه چیزی داره به عنوان `container` که مانند ماشین های مجازی لینوکس هستند با این تفاوت که اونها خیلی کارآمد تر هستند و منابع کمتری مصرف میکنند و اپلیکیشن شما رو در یک محیط (`container`) ایزوله با محیطی پایدار اجرا میکنه.  
`Container` که اپلیکیشن شما رو داخل خودش داره متونه روی وینوز ، مک او اس و لینوکس اجرا بشه و همین قضیه مشکل دولوپر های مختلف با سیستم های عامل متفاوت را حل میکند.  
قابلیت `docker` برای ساحت و نگهداری محیطی استوار باعث `portable` شدن اپلیکیشن شما میشه.

## اما این container ها رو باید از کجا پیدا کرد ؟

Docker به جایی داره که شما میتونید خیلی از این container ها رو اونجا پیدا کنید و ازشون استفاده کنید به نام docker hub که در زیر آدرس اون رو قرار میدم . البته توجه داشته باشید که داکر به آی پی ایران سرویس نمیده و برای نصب کردن و استفاده از docker hub باید از وی پی ان استفاده کنید.

<https://hub.docker.com>

تا زمانی که شما container مورد نظر خودتون رو بسازید میتونید از اینجا موارد مشابه ای رو پیدا کنید و سرعت ساخت اپلیکیشنتون رو بالاتر ببرید.

نمونه هایی از container هایی که میتونید از داکر هاب دانلود و استفاده کنید را در زیر نمایش میدهیم

[Node.js](#) , [MySQL](#) , [Ubuntu](#)

## چگونه باید اپلیکیشن خودمون رو به یک container اضافه کنیم ؟

زمانی که شما container مورد نظر خودتون رو پیدا کردید قطعا میخواهید اون رو customize کنید و dependency های خودتون رو بهش اضافه کنید . اینجا جاییه که Dockerfile میاد سر کار.

Dockerfile به طور خلاصه توضیح میده که شما از داکر میخوانین چطور برای بار اول container اتون رو بالا بیاړه

به عنوان مثال یک نمونه از Dockerfile رو در زیر براتون قرار میدم.

```
# Use the official Node.js runtime as a base image
FROM node:alpine

# Set the directory of my web application to /app
WORKDIR /app

# Copy over my project's directory into the container /app folder
Add . /app

# Install all the dependencies for my web application
RUN yarn install

# Make the port 3000 accessible outside of Docker
EXPOSE 3000

# Execute the command yarn start
CMD ["yarn", "start"]
```

این Dockerfile هر وقت که یک instance ازش ساخته بشه اجرا میشه . هنوز برای ساخت یک container شما نیازمند docker image هستید . هر وقت شما یک Image بسازید متوانید از روی اون به هر تعدادی که دلتون میخواد container بسازید.

در حقیقت شما از داکر هاب ایمج های داکری آماده را دانلود میکنید که میتوانید به کمک آن ها هر تعداد container ای که لازم دارید را بسازید . همچنین میتوانید تنظیمات شخصی خود را بر روی کانتینری که ساخته اید اعمال کنید و دوباره از آن container یک Image جدید بسازید تا برای ساخت کانتینری هر بار نیازمند انجام تنظیمات قبلی نباشید و container های خودتون رو بر اساس Image جدیدی که ساخته اید ، تولید کنید.

## آیا من فقط به یک container برای کل اپلیکیشنم نیاز دارم ؟

وقتی قابلیت ساخت کانتینر های مختلف وجود داره یک best practice اینه که بخش های مختلف اپلیکیشنمون رو ایزوله کنیم به container های مختلف

برا مثال فرض کنید یه وب اپلیکیشن که به یک دیتابیس و یک وب سرور پایتون نیازمنده به دو تا container نیازمنده . یکیش برای سرور پایتون و یکیشم برای دیتابیس

شاید بپرسید چرا ؟ چون اداره اپلیکیشنتون رو ماژولار تر میکنه و اینکه وقتی در آینده اپلیکیشنتون رشد کنه یا گیر های ترافیکی داشته باشه شما میتونین چند تا instance از python server برای مدیریت ترافیک استفاده کنید.

## آیا مدیریت کردن چند تا container به صورت همزمان کار سختی نیست ؟

دقیقا اینجا جاییه که داکر به ابزار دیگه رو معرفی میکنه به عنوان docker compose . این ابزار به شما این امکان رو میده که میتونید کانفیگ container هایی که به هم مرتبط هستند رو در یک فایل yml مشخص کنید . یه فایل Docker-compose.yml به شکل زیر است

```
version: '2'
services:
  server:
    build: .
    ports:
      - "3000:3000"
    volumes:
      - ./app
  database:
    image: "mariadb:10.3"
    environment:
      - MYSQL_ROOT_PASSWORD=MyPassword
      - MYSQL_DATABASE=MyDatabase
    ports:
      - "3306:3306"
```

همین طور که در بالا مشاهده میکنید ما دو تا سرویس (container) داریم یکی به اسم server و یکی به اسم database که در اون متغیر ها و پورت ها و volume ها رو تعریف کردیم . وقتی فایل Docker-compose آماده میشه ، مدیریت گروهی از container ها که مربوط به یک اپلیکیشن هستند خیلی ساده میشه.

## ابزار هایی که بهتر است بشناسید

**Kitematic** : به شما این امکان رو میده تا کانتینر های خودتون رو در یک محیط ویژوالی مدیریت کنید . شاید برای اولش که یک مقدار با محیط CLI آشنا نشدید گزینه خوبی باشه

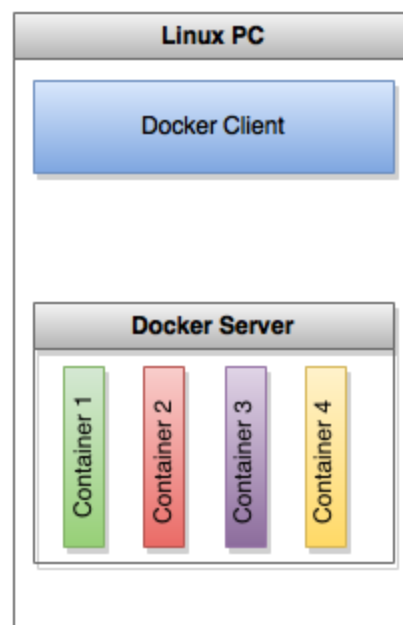
**Docker Swarm** : وقتی اپلیکیشن شما بزرگ میشه شما ممکنه بخواین که چند تا container داشته باشین تا traffic شما رو هندل کنند. این ابزار باعث میشه مدیریت گروهی از container ها (clusters) در یک زمان کار ساده تری بشه

## فصل دوم

# ساختار Docker

### ساختار Docker

Docker ساختاری server base داره و به زبان ساده این طور عمل میکنه که client دستورش رو به docker client میده و اون هم از طریق Docker Deamon(Dockerd) اجراش میکنه



Docker شامل سه کامپوننت هست

1. Image
2. Container
3. Registries

## Docker Image

بسته نرم افزاری هست که docker رو اجرا میکنه (اصولا container) ، image شامل تمامی فایل ها و library هایی است که برای اجرای نرم افزار به اون نیاز داریم و هر کسی میتونه image شخصی خودش رو از طریق Docker Client CLI بسازه .

## Docker Container

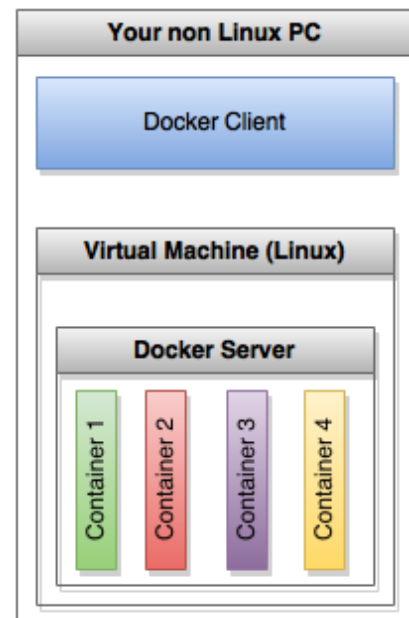
ها در حقیقت یک instance از image هستند و نکته اینجاست که آنها نسبت به host system کاملا ایزوله و مستقل از منابع نرم افزاری هستند.

## Docker Registries

یک پلتفرمی هست که میزبان image های کاربران میباشد. هر کسی میتونه image های مورد نظرش رو push یا download کنه و حتی میتونید برای خودتون یا سازمانتون یک private registry تولید کنید.

## Docker Deamon

از هسته لینوکس استفاده میکنه و شاید به شما این باور رو بده که همیشه روی OS های دیگه اجراش کرد اما به همین خاطر تیم docker ابزاری آماده کرده به عنوان Boot2Docker که شامل package های یک linux سبک هست که تمامی نیازمندهای Dockerd رو داره و به نحوی linuxVM داره در بالاترین سطح OS شما اجرا میشه تا Docker کارش رو انجام بده





# نصب Docker CE (Community Edition)

در این بخش ما طریقه [نصب داکر](#) بر روی سیستم عامل **ubuntu** را آموزش میدهیم توجه داشته باشید که با توجه به لینک نصب میتوانید نحوه نصب بر روی تمامی سیستم عامل ها را مشاهده کنید.

در اینجا ما نحوه نصب Docker CE از طریق repository را توضیح میدهیم

## نصب Repository

1. در ابتدا پکیج apt رو آپدیت کنید

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

2. نصب پکیج هایی که به شما اجازه میده از طریق HTTPS به repository دسترسی داشته باشد و ازش استفاده کنید

```
$ sudo apt-get install \
  apt-transport-https \
  ca-certificates \
  curl \
  software-properties-common
```

3. اضافه کردن GPG Key رسمی داکر

```
$ curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo apt-key
add -
```

4. از دستور زیر برای استفاده از [stable repositories](#) استفاده کنید

```
$ sudo add-apt-repository \
  "deb [arch=amd64] https://download.docker.com/linux/ubuntu \
  $(lsb_release -cs) \
  stable"
```

## نصب Docker CE

1. دوباره پکیج apt را آپدیت کنید

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

2. آخرین نسخه Docker CE را نصب کنید

```
$ sudo apt-get install docker-ce
```

دستور بالا خودش آخرین ورژن رو میگیره ولی اگر بخواهیم ورژن خاصی رو نصب کنیم هم میتونیم از روش زیر استفاده کنیم

```
$ apt-cache madison docker-ce
```

Result :

```
docker-ce | 18.03.0~ce-0~ubuntu | https://download.docker.com/linux/ubuntu  
xenial/stable amd64 Packages
```

با دستور بالا میتونیم ورژن های موجود docker رو ببینیم و با دستور زیر آن را نصب کنیم

```
sudo apt-get install docker-ce=<VERSION>
```

برای اینکه متوجه بشیم که Docker CE درست نصب شده ایمیج زیر را اجرا میکنیم

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

این دستور در ducker hub به دنبال ایمیج hello-world میگرده و در صورتی که در آنجا وجود داشته باشه آخرین ورژن اش رو دانلود میکنه

## فصل چهارم

### نصب Geoserver به کمک docker

نیاز به اجرای جئوسروور در محیط لینوکس و سختی نصب پلاگین هایی مانند gdal که برای لود عکس هایی با فرمت ecw استفاده میشود باعث شد تا برای اجرای جئوسروور ، از Docker کمک بگیریم

در ادامه سعی میکنیم کلیه دستورات docker و linux و همچنین توضیحات تکمیلی را در قالب نصب و ساخت ایمج جئوسروور دلخواهتان ، توضیح و بررسی کنیم

#### هدف از ساخت docker image برای جئوسروور چیست؟

قطعا همان طور که در ابتدای این مقاله گفته شد ، سادگی و قابل حمل بودن و سرعت deploy توسط docker باعث شد برای سهولت کار جئوسروور را تبدیل به یک docker image کنیم . برا مثال ما میخواستیم جئوسروورمان دارای دو پلاگین oracle و GDAL باشد که به صورت پیش فرض بر روی جئوسروور نصب نیستند و نصب آن ها بر روی سیستم عامل های مختلف دارای پیچیدگی های مختلفی میباشد اما با انجام کلیه این تنظیمات و ساخت یک docker image دیگر نیازی نیست که هر بار تنظیمات را انجام دهیم و با ساخت هر container از ایمج جئوسروور تنظیمات ما به صورت پیشفرض در جئوسروور اعمال میشوند.

در صورت تغییر در تنظیمات ، اضافه کردن لایه ها ، استایل ها ، تنظیمات cache نقشه و ... آیا هر بار که

container ما پاک شود و دوباره راه اندازی شود ، اطلاعات از بین میرود ؟

جواب منفی است ، در docker امکان volume کردن یا به زبان ساده تر امکان شنود کردن اتفاقاتی که در container در حال رخ دادن است وجود دارد و ما میتوانیم با استفاده از این قابلیت اتفاقات مورد نظرمون رو شنود کنیم و در host خودمون نگه داریم . به عنوان مثال در ایمج جئوسروور ، ما کلیه محتویات پوشه data\_dir را که شامل کلیه تنظیمات و تغییرات جئوسروور هست را از طریق volume (( که در ادامه توضیح خواهیم داد)) در هاست خود ذخیره میکنیم تا هر بار که یک instance جدید ساخته میشود آخرین تغییرات در جئوسروور وجود داشته باشد و نیازی به دوباره کاری برای بارگزاری اطلاعات جئوسروور نباشد.

## مرحله اول

اولین مسئله ای که باید به آن دقت کرد این است که قبل از تلاش برای ساختن یک ایمیج از صفر تا صد بهتر است که سری به [dockerhub](https://hub.docker.com/) بزنیم. چون در بیشتر مواقع مواردی مشابه به تنظیماتی که در نظر شما وجود دارد را میتوان از آنجا پیدا کرد با همین رویکرد ما توانستیم با جستجوی geoserver در ایمیج ها و بررسی امکانات آنها یکی را انتخاب کنیم تا به عنوان اولین قدم روی آن کار کرده تا در نهایت base image خود را آماده کنیم

### winsent/geoserver

ایمیج انتخابی ما که آدرسش در اسم ایمیج لینک شده است ، ایمیج winsent بود چون به صورت پیشفرض دارای پلاگین GDAL بود که البته برای این کار باید چند مرحله انجام میدادیم

با بررسی [repository](#) ایمیج و دانلود source آن متوجه شدیم برای اینکه بخواهیم ایمیجی با پلاگین از قبل آماده شده GDAL داشته باشیم باید از روی Dockerfile موجود در پوشه ورژن 2.12 و پوشه lbcw یک ایمیج تهیه کنیم.

در ابتدا برای اینکه اگر کاربر ادمین نباشید و نخواهید هر بار دستور sudo را قبل از دستورات docker بیاورید میتوانید از کد زیر در terminal استفاده کنید. بعد از enter کردن رمز عبور یوزر خود را تایپ کنید و enter کنید حالا شما به عنوان یوزر ادمین شناخته شده اید

```
$ sudo -i
```

### نحوه ساخت یک Docker Image از روی Dockerfile

```
$ docker build -t imageName .
```

توجه : در انتها بعد از اسم ایمیج باید یک space زده و [.] را بگذارید تا دستور عمل کند.

بعد از ساخته شدن image یکسری دستورات لازمه را مرور میکنیم

### نحوه pull کردن یک image از docker hub

```
$ docker pull image_name
```

دیدن لیستی از همه Image های داکری که روی هاست قرار دارد

```
$ docker images
```

### حذف docker image

وقتی دستور docker images رو اجرا میکنیم کل اطلاعات ایمیج شامل Id ، نام و ... رو به ما نشون میده

```
$ docker rmi <imageId>
```

## ساخت یک container از روی image

برای این منظور از دستور زیر استفاده میکنیم

```
$ docker run -d -p 8087:8080 -v path/to/host:/opt/geoserver/data_dir <imageName>
```

-d

یعنی container رو در background اجرا کنه

-p 8087:8080

یعنی پورت 8080 در container رو به پورت 8087 در هاست خودتون مپ میکنه . یعنی برای دسترسی بهش کافیه توی هاست خودتون از آدرس زیر استفاده کنید

127.0.0.1:8087/geoserver

-v

وظیفه volume کردن را بر عهده دارد یعنی محتویات داخل پوشه /opt/geoserver/data\_dir که در container قرار دارد را میریزه داخل مسیر سمت چپی که در هاست شماست.

## نمایش لیست تمامی container های فعال

```
$ docker ps
```

## نمایش لیست تمامی container های فعال و غیر فعال

```
$ docker ps -a
```

## حذف container

```
$ docker rm <containerName/Id>
```

## Stop و start کردن container ها

```
$ docker stop/start <container Name/Id>
```

## Stop کردن همه container ها به صورت دسته جمعی و حذف آنها

### روش اول

```
$ docker stop $(docker ps -a -q)
```

```
$ docker rm $(docker ps -a -q)
```

\$ docker ps -a -q این دستور id (-q) همه کانتینر ها (-a) را بر میگردونه

### روش دوم

```
$ docker rm -f $(docker ps -a -q)
```

-f به معنی force یا اجبار میباشد ، پس بدون نیاز به stop کردن خودش اول stop میکنه همه container ها را و بعد پاکشون میکنه

## دستور رفتن به داخل container

```
$ docker exec -it <containerName/Id> /bin/bash
```

## خروج از یک container بدون stop شدن

Ctrl + d

حالا که container ما ساخته شده میخوایم پلاگین های oracle را هم بهش اضافه کنیم . برای این منظور باید jar فایل های oracle و ojdbc را در مسیر زیر داخل container کپی کنیم

```
$ docker cp ojdbc.jar myContainer:/opt/geoserver/webapps/geoserver/WEB-INF/lib/
```

حالا که این کار را انجام دادیم پلاگین های ما اضافه میشوند و در حقیقت base image ما ساخته شده است به همین منظور از container موجود یک Image میسازیم تا همیشه از آن استفاده کنیم

## ساخت Docker Image از روی Container

```
$ docker commit <container_id> <container_name>:<tag_version>
```

مثال :

```
$ docker commit 2fg4805485ad baharan/geoserver:1.0.0
```

اگر tag\_version: رو نگذاریم خودش latest در نظر میگیره

برای نگهداری Image خودمون یک tar فایل ازش میسازیم تا هرجایی که میخوایم ازش استفاده کنیم اون رو لود کنیم

## ساخت tar فایل از image

```
$ docker save <image_name> yourImageName.tar
```

مثال:

```
$ docker save baharan/geoserver:1.1.0 bh-geoserver.tar
```

## لود کردن tarFile

حالا برای لود کردن این Image باید از دستور زیر استفاده کرد

```
$ docker load tarFileAddress
```

**توجه:** در volume ای که در هاست خود داریم باید پوشه ای به نام data بسازیم فایل های ecw را درون آن پوشه بارگذاری کنیم . این باگ پلاگین GDAL میباشد که حتما باید دیتا های رستری آن داخل پوشه data در مسیر geoserver\_data\_dir باشد

## استفاده از docker compose

برای راحتی کار میتوانیم برای ساخت container از فایل dockerCompose.yml استفاده کنیم

به عنوان مثال فایل dockerCompose جنوسرور

```
version: '2'
services:
  app:
    image: "baharan/geoserver:1.0.0"
    ports:
      - "8080:8080"
    volumes:
      - /home/nariman/Desktop/.../geoserverData:/opt/geoserver/data_dir
```

برای اجرا کردن از طریق docker compose کافیست به مسیری که فایل در اون قرار دارد بروید و دستور زیر را اجرا کنید

```
$ docker-compose -f dockerComposefile.yml up -d
```

## بارگذاری دیتای پیشفرض (نقشه، لایه و ...) در هنگام ساخت container

اگر دیتا ای دارید که میخواهید در هنگام بالاآوردن container خودتون درون جنوسرور لود شود باید به صورت زیر عمل کنیم .

همان طور که قبل تر توضیح دادیم volume باعث میشود تا تغییرات container بر روی هاست ذخیره شوند حالا اگر دیتا و کانفیگ هایی از قبل داریم باید آنها را درون پوشه `/home/nariman/Desktop/.../geoserverData` قرار دهیم چون به هنگام ساخت کانتینر این پوشه چک میشود و اگر دارای فایل و فولدر باشد ، آنها را به قسمت `opt/geoserver/data_dir` کپی میکند و وقتی geoserver اجرا میشود ما دیتاهای خودمون رو میبینیم و از این به بعد هم هر تغییری در تنظیمات geoserver انجام بدهیم در این بخش ذخیره میشود و به هنگام بالا آمدن مجدد مشکلی نخواهیم داشت