CMPE 283: Virtualization Tec
Lab 1 - Cross Host Linking Docker Containers
Hitesh Padekar
(009273303)

Step 1: Pull image from the docker hub

Image for Node.JS app:

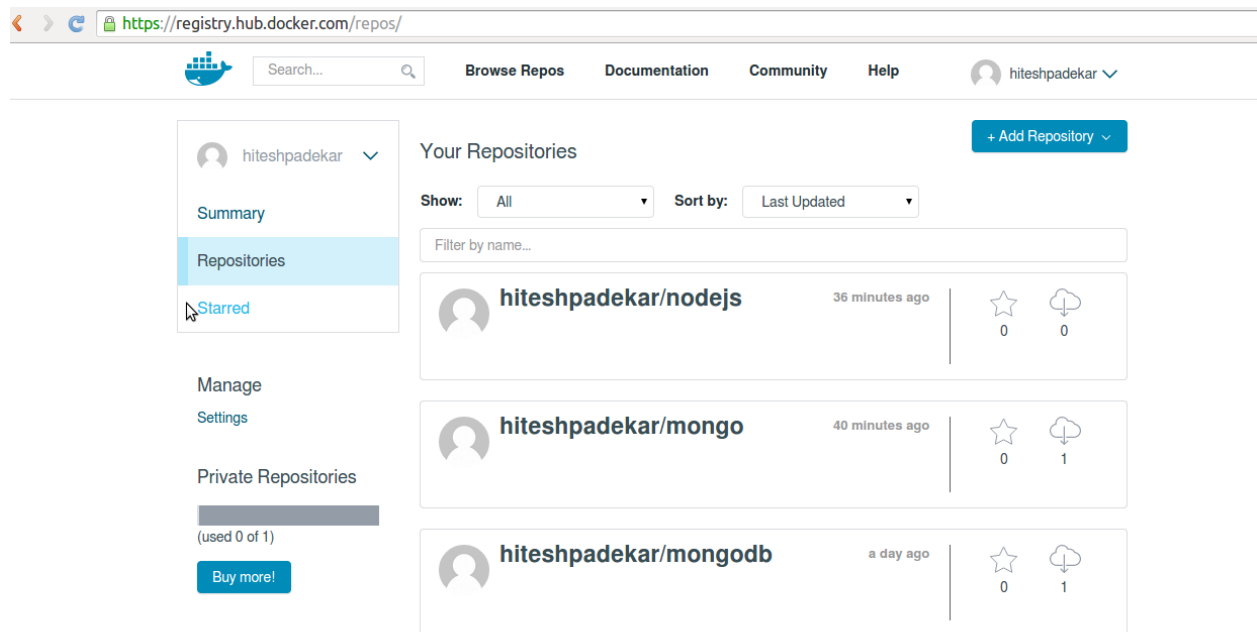website: https://registry.hub.docker.com/u/hiteshpadekar/nodejs/

to pull docker image use: docker pull hiteshpadekar/nodejs

Image for MongoDb:

website: https://registry.hub.docker.com/u/hiteshpadekar/mongo/

to pull docker image use: docker pull hiteshpadekar/mongo

Search...   Browse Repos   Documentation   Community   Help   hiteshpadekar

Updated 36 minutes ago

hiteshpadekar / nodejs

Pull this repository   docker pull hiteshpadekar/nodejs

No description set

⭐ 0   💬 0   ⬇ 0

Information	Tags

No description set

Properties

🕐 2015-02-15 20:28:09
hiteshpadekar

Settings

🔗 Description
⚓ Webhooks
👥 Collaborators
✉ Mark as unlisted

🔒 Make Private
🗑 Delete Repository

Comments ➕

No comments available, be the first to comment.

Repository for nodejs app

Search...   Browse Repos   Documentation   Community   Help   hiteshpadekar

Updated 6 hours ago

hiteshpadekar / mongo

Pull this repository   docker pull hiteshpadekar/mongo

No description set

⭐ 0   💬 0   ⬇ 0

Information	Tags

No description set

Properties

🕐 2015-02-15 14:06:31
hiteshpadekar

Settings

🔗 Description
⚓ Webhooks
👥 Collaborators
✉ Mark as unlisted

🔒 Make Private
🗑 Delete Repository

Comments ➕

No comments available, be the first to comment.

Repository for mongodb

To pull images use "docker pull" command


```
root@master:/home/hitesh/283/docker# docker images
REPOSITORY          TAG                IMAGE ID          CREATED           VIRTUAL SIZE
ubuntu              latest             5ba9dab47459      2 weeks ago       188.3 MB
root@master:/home/hitesh/283/docker# docker ps -a
CONTAINER ID        IMAGE              COMMAND           CREATED           STATUS            PORTS
        NAMES
root@master:/home/hitesh/283/docker#
```

Pull Node.JS app from the repository:


```
root@master:/home/hitesh/283/docker# docker pull hiteshpadekar/nodejs
Pulling repository hiteshpadekar/nodejs
ddff69e605db: Pulling dependent layers
511136ea3c5a: Download complete
27d47432a69b: Download complete
5f92234dcf1e: Download complete
51a9c7c1f8bb: Download complete
5ba9dab47459: Download complete
8d7598604c43: Downloading 2.111 MB/86.51 MB 22m3s
```

Pull mongodb from the repository:

```
root@master: /home/hitesh/283/docker

root@master: /home/hitesh/283/docker   ×   root@master: /home/hitesh/283/docker   ×   root@master: /home/hitesh/283/docker   ×

root@master:/home/hitesh/283/docker# docker pull hiteshpadekar/mongo
Pulling repository hiteshpadekar/mongo
f00d768bc174: Pulling dependent layers
```
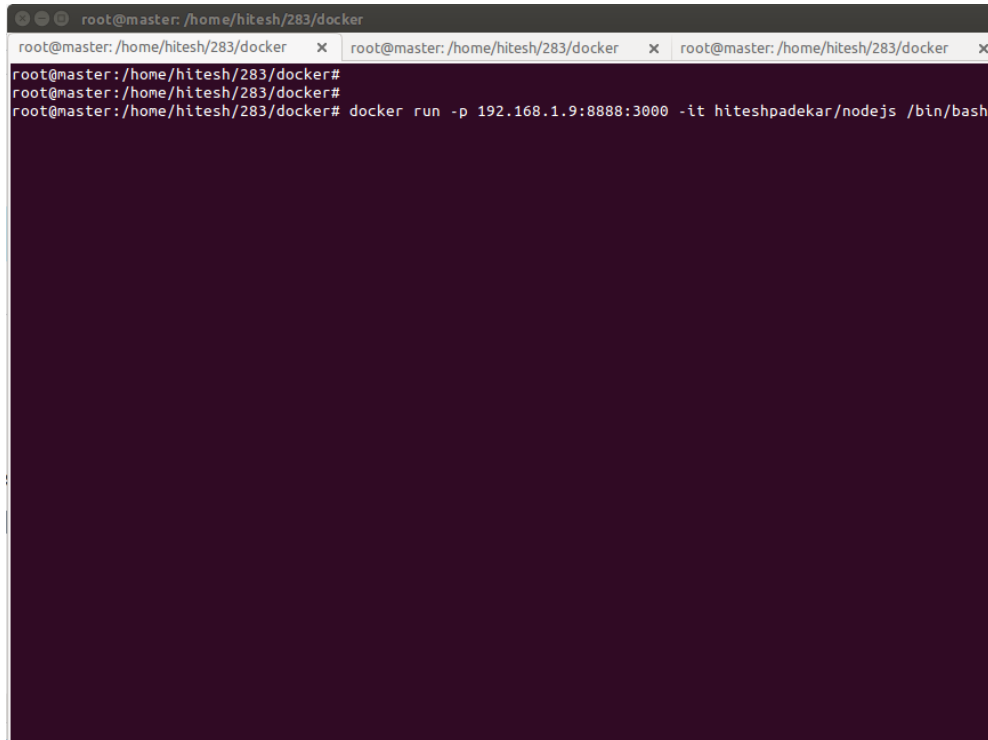
Pull Complete:

```
root@master: /home/hitesh/283/docker

root@master: /home/hitesh/283/docker   ×   root@master: /home/hitesh/283/docker   ×   root@master: /home/hitesh/283/docker   ×

root@master:/home/hitesh/283/docker# docker pull hiteshpadekar/mongo
Pulling repository hiteshpadekar/mongo
f00d768bc174: Download complete
511136ea3c5a: Download complete
27d47432a69b: Download complete
5f92234dcf1e: Download complete
51a9c7c1f8bb: Download complete
5ba9dab47459: Download complete
7b1fb386b959: Download complete
353e4a4578d4: Download complete
c85c83a1acdd: Download complete
e4089d40d54e: Download complete
148815be519f: Download complete
0d9b89c01bc2: Download complete
b427c5761656: Download complete
root@master:/home/hitesh/283/docker#
```

We can see the downloaded images by running "docker images" command:

```
root@master: /home/hitesh/283/docker

root@master: /home/hitesh/283/docker    ×    root@master: /home/hitesh/283/docker    ×    root@master: /home/hitesh/283/docker    ×

root@master:/home/hitesh/283/docker# docker pull hiteshpadekar/nodejs
Pulling repository hiteshpadekar/nodejs
ddff69e605db: Download complete
511136ea3c5a: Download complete
27d47432a69b: Download complete
5f92234dcf1e: Download complete
51a9c7c1f8bb: Download complete
5ba9dab47459: Download complete
8d7598604c43: Download complete
root@master:/home/hitesh/283/docker# docker images
REPOSITORY                TAG           IMAGE ID          CREATED              VIRTUAL SIZE
hiteshpadekar/mongo       latest        f00d768bc174      About an hour ago    5.382 GB
hiteshpadekar/nodejs      latest        ddff69e605db      About an hour ago    427.2 MB
ubuntu                    latest        5ba9dab47459      2 weeks ago          188.3 MB
root@master:/home/hitesh/283/docker#
```
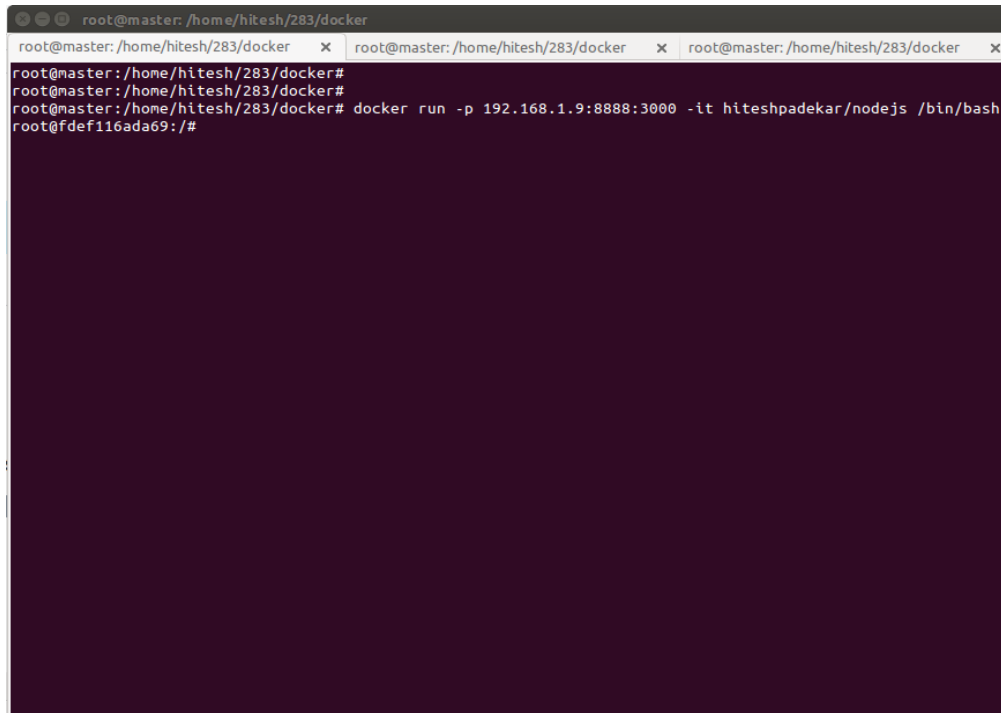
Step 2: Starting the containers for Node.js app and mongodb

For Node.js app container run following command:

After hitting Enter we will enter the node.js container:

Similarly for mongodb container enter following command and hit Enter key:

```
root@master: /home/hitesh/283/docker
root@master: /home/hitesh/283/docker  ×   root@master: /home/hitesh/283/docker  ×   root@master: /home/hitesh/283/docker  ×
root@master:/home/hitesh/283/docker# docker images
REPOSITORY                TAG           IMAGE ID          CREATED             VIRTUAL SIZE
hiteshpadekar/mongo       latest        f00d768bc174      About an hour ago   5.382 GB
hiteshpadekar/nodejs      latest        ddff69e605db      About an hour ago   427.2 MB
ubuntu                    latest        5ba9dab47459      2 weeks ago         188.3 MB
root@master:/home/hitesh/283/docker# docker run -p 192.168.1.9:27017:27017 -it hiteshpadekar/mongo /bin/bash
```

Step 3: Starting MongoDB inside mongodb container

Start "mongod" deamon inside the container

```
root@master: /home/hitesh/283/docker
root@master: /home/hitesh/283/docker  ×   root@master: /home/hitesh/283/docker  ×   root@master: /home/hitesh/283/docker  ×
root@master:/home/hitesh/283/docker# docker images
REPOSITORY              TAG             IMAGE ID            CREATED             VIRTUAL SIZE
hiteshpadekar/mongo     latest          f00d768bc174        About an hour ago   5.382 GB
hiteshpadekar/nodejs    latest          ddff69e605db        About an hour ago   427.2 MB
ubuntu                  latest          5ba9dab47459        2 weeks ago         188.3 MB
root@master:/home/hitesh/283/docker# docker run -p 192.168.1.9:27017:27017 -it hiteshpadekar/mongo /bin/bash
root@bc1eed7f4f2b:/# mongod &
[1] 8
root@bc1eed7f4f2b:/# mongod --help for help and startup options
2015-02-16T06:05:21.202+0000 [initandlisten] MongoDB starting : pid=8 port=27017 dbpath=/data/db 64-bit host
=bc1eed7f4f2b
2015-02-16T06:05:21.202+0000 [initandlisten] db version v2.6.7
2015-02-16T06:05:21.202+0000 [initandlisten] git version: a7d57ad27c382de82e9cb93bf983a80fd9ac9899
2015-02-16T06:05:21.202+0000 [initandlisten] build info: Linux build7.nj1.10gen.cc 2.6.32-431.3.1.el6.x86_64
 #1 SMP Fri Jan 3 21:39:27 UTC 2014 x86_64 BOOST_LIB_VERSION=1_49
2015-02-16T06:05:21.202+0000 [initandlisten] allocator: tcmalloc
2015-02-16T06:05:21.202+0000 [initandlisten] options: {}
2015-02-16T06:05:21.247+0000 [initandlisten] journal dir=/data/db/journal
2015-02-16T06:05:21.247+0000 [initandlisten] recover begin
2015-02-16T06:05:21.247+0000 [initandlisten] recover lsn: 319503
2015-02-16T06:05:21.247+0000 [initandlisten] recover /data/db/journal/j._0
2015-02-16T06:05:22.515+0000 [initandlisten] recover skipping application of section seq:0 < lsn:319503
2015-02-16T06:05:22.515+0000 [initandlisten] recover skipping application of section seq:141733 < lsn:319503
2015-02-16T06:05:22.670+0000 [initandlisten] recover cleaning up
2015-02-16T06:05:22.670+0000 [initandlisten] removeJournalFiles
2015-02-16T06:05:22.693+0000 [initandlisten] recover done
2015-02-16T06:05:22.693+0000 [initandlisten] preallocating a journal file /data/db/journal/prealloc.0
2015-02-16T06:05:25.295+0000 [initandlisten]          File Preallocator Progress: 482344960/1073741824    4
4%
2015-02-16T06:05:28.571+0000 [initandlisten]          File Preallocator Progress: 692060160/1073741824    6
4%
2015-02-16T06:05:31.078+0000 [initandlisten]          File Preallocator Progress: 933232640/1073741824    8
6%
2015-02-16T06:05:40.181+0000 [initandlisten] waiting for connections on port 27017
```
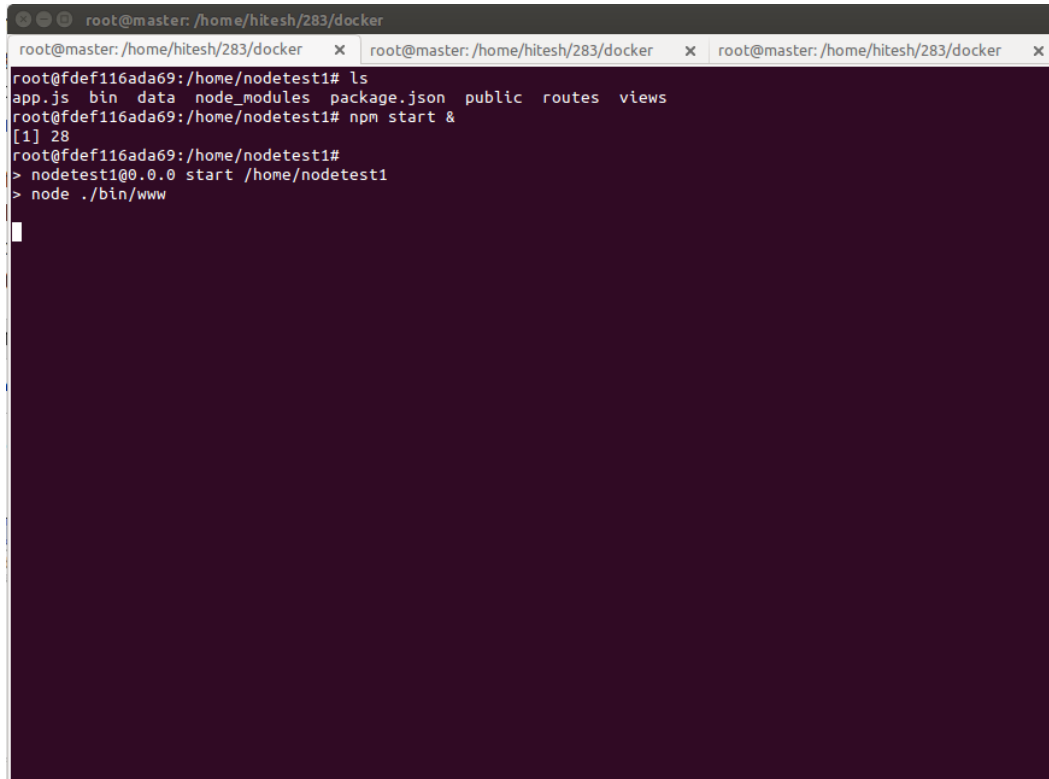
Start mongo client and see the documents in the mongodb database

```
root@master: /home/hitesh/283/docker
root@master: /home/hitesh/283/docker  ×   root@master: /home/hitesh/283/docker  ×   root@master: /home/hitesh/283/docker  ×
root@bc1eed7f4f2b:/# mongo
MongoDB shell version: 2.6.7
connecting to: test
2015-02-16T06:10:46.662+0000 [initandlisten] connection accepted from 127.0.0.1:55601 #2 (1 connection now o
pen)
> use nodetest1
switched to db nodetest1
> db.usercollection.find().pretty()
{
        "_id" : ObjectId("54e14a9c5dd5b41e7151dcc7"),
        "username" : "hitesh",
        "email" : "hitesh.padekar@gmail.com"
}
{
        "_id" : ObjectId("54e14b595dd5b41e7151dcc8"),
        "username" : "shruti",
        "email" : "shruti.rao@sjsu.edu"
}
{
        "_id" : ObjectId("54e14b595dd5b41e7151dcc9"),
        "username" : "vinit",
        "email" : "vinit.parekh@gmail.com"
}
>
```
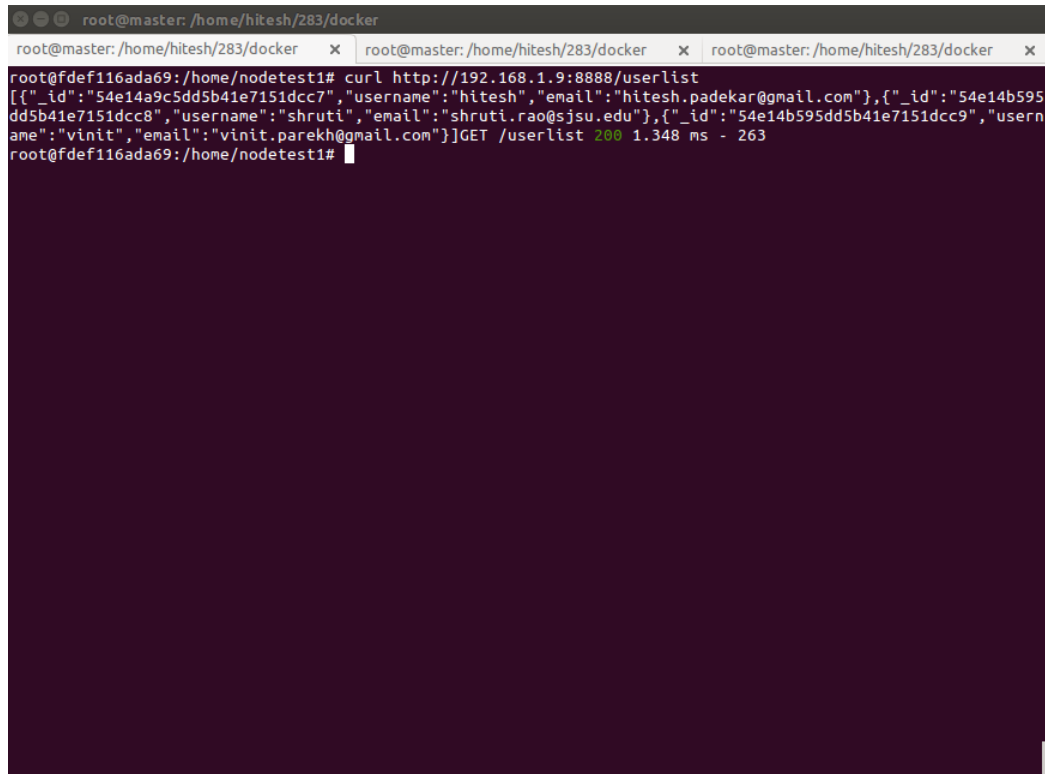
Step 4: Starting Node.js web app inside nodejs container

Start node.js web application inside the container by running "npm start"

Step 5: Web application running inside the node.js container and fetching data from mongodb running on other container.

Node.js web application running on node.js container can be verified using "curl" command

It can also be verified in the browser