

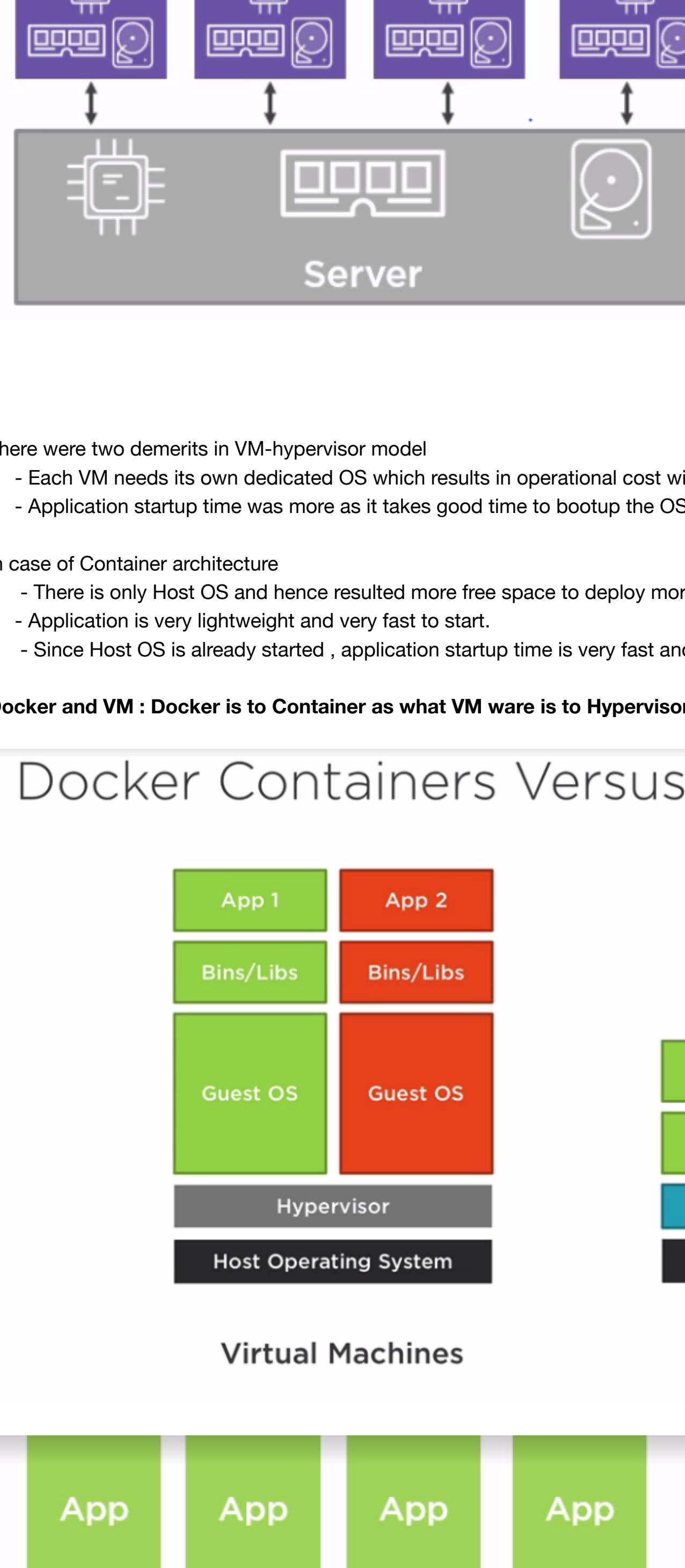
Docker Fundamental

Docker concepts is the way to remove the performance and resourcing bottlenecks from the VM concepts.

VM / Hypervisor

Hypervisor allows multiple applications to run on single server. Each application run in the dedicated VM instance. Each VM instance has its allotted CPU, RAM, disk and other resourcing details shared with the server. But each such VM instance also has its own Operating system. And that's the biggest bottleneck in this architecture.

Each OS demands dedicated CPU, RAM, disk space. Also they may incur license cost and Admin cost[security patching, Antivirus management...]. These all factors are required just to manage OS and has no direct benefits towards deployed application and thus is a wastage of business money.



There were two demerits in VM-hypervisor model

- Each VM needs its own dedicated OS which results in operational cost without adding any value to the business
- Application startup time was more as it takes good time to bootup the OS.

In case of Container architecture

- There is only Host OS and hence resulted more free space to deploy more applications.
- Application is very lightweight and very fast to start.
- Since Host OS is already started, application startup time is very fast and leads us to do smooth scalability factor.

Docker and VM : Docker is to Container as what VM ware is to Hypervisor

Docker Containers Versus Virtual Machines



Docker

- It simplifies and provides the consistent way of building, shipping and running the code in different env.

Docker benefits

- setup development env for the team very quickly- Once image is provided, these can be run anywhere to get the container of live running application.

- eliminate APP conflicts, technical stack version conflicts or version conflicts

- provides env consistency between different env like dev, staging, qa or prod- coz it moves codes in a isolated and fully managed or packaged containers. These containers have all the application dependencies defined and existed.

- ship software faster,

Docker Fundamentals :

Images and Container

Images are the blueprints or the definitions of the technical stack, application and all of its dependencies. It is used to create container which contain the live code and acts as a run time package for the application. .

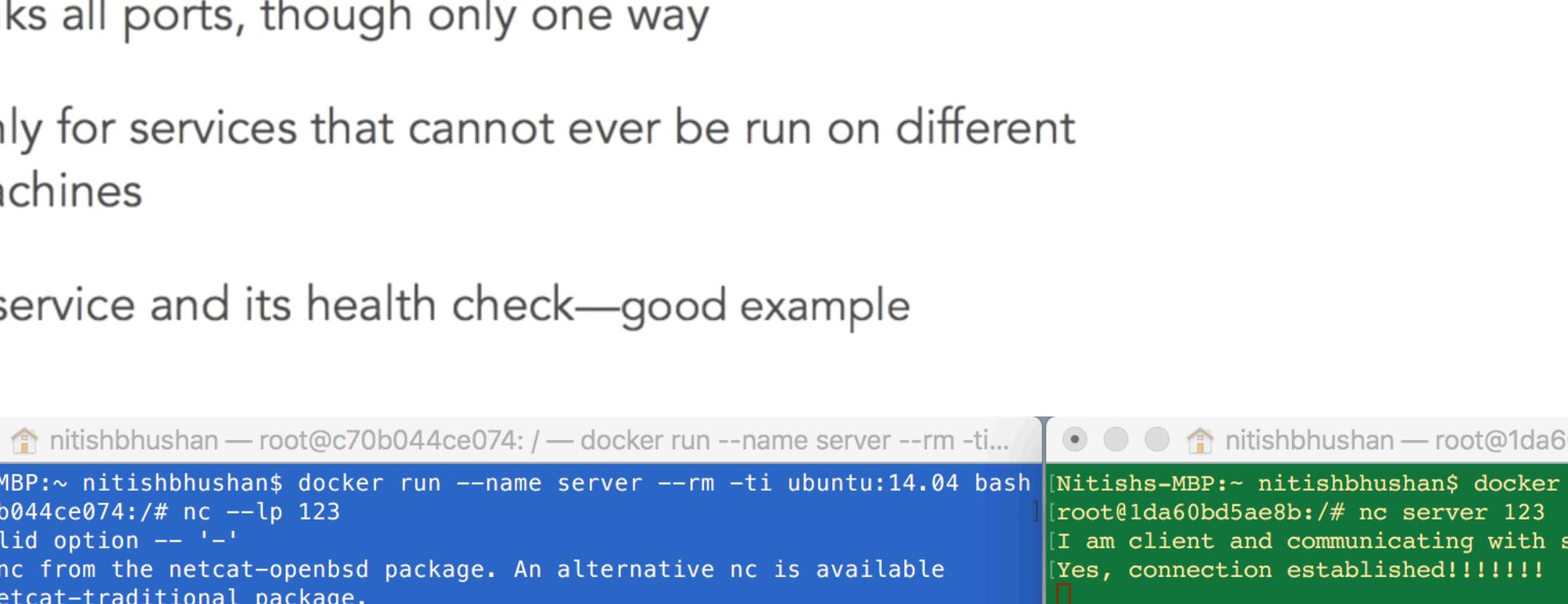
In the OS concept, Images act as a Class and Container is its objects.

As per the official definition :

- Image are the READ-ONLY template of layered filesystem[as per the application needs] that is used to create Docker container

- Container are isolated and secured env created from the Images that can be started, stopped, deleted, moved and run.

The Docker Flow



Docker run would take the image and gives back the container

Docker commit <container id> <new image name> would take the container and gives back the image.

Docker images tagging

docker commit <container id> <repo> -- default tag -> latest'

docker commit <container id> <repo> <tag> -> tag would be -> tag1'

```
Nitish-MBP:~ nitishbhushans$ docker ps -l
CONTAINER ID        IMAGE               COMMAND                  CREATED             STATUS              PORTS
 NAMES
605885eac5c7        "hello"
        awesome_stonebraker
Nitish-MBP:~ nitishbhushans$ docker commit 605885eac5c7 my_commit:v0.0
sha256:c399ac1734f40862b1e03c470b4356258de7c39496f7cc73e24510dc8a
Nitish-MBP:~ nitishbhushans$ docker images
REPOSITORY          TAG      IMAGE ID            CREATED             SIZE
<none>              <none>   c5399ac1734f   3 seconds ago   1.84KB
my_commit           v0.0     4fe693be3ce89   About a minute ago  1.84KB
<none>              <none>   cf4f04fffc289   2 days ago       195MB
friendlyhello       latest   cf4f04fffc289   2 days ago       195MB
ubuntu              latest   2d690327ab2e   3 weeks ago      122MB
python              2.7-slim  8b88106072d7   3 weeks ago      184MB
nginx               latest   de19345146b9   3 weeks ago      188MB
nodejs              latest   053b3031f6c3   3 weeks ago      1.84KB
javaee/angular2-java-hello-world latest   11e4d501f6c3   7 months ago     1.37GB
Nitish-MBP:~ nitishbhushans$ docker commit 605885eac5c7 my_commit:v1.0
sha256:4a0e301a0091db7000c533eab64f42abb7d9da4cbcb1915edc743ad7b8
Nitish-MBP:~ nitishbhushans$ docker images
REPOSITORY          TAG      IMAGE ID            CREATED             SIZE
my_commit           v1.0     9d6dd0447d62   2 seconds ago   1.84KB
my_commit           <none>   c5399ac1734f   9 days ago       1.84KB
18 seconds ago
<none>              <none>   4fe693be3ce89   About a minute ago  1.84KB
friendlyhello       latest   cf4f04fffc289   2 days ago       195MB
ubuntu              latest   2d690327ab2e   3 weeks ago      122MB
python              2.7-slim  8b88106072d7   3 weeks ago      184MB
ubuntu              latest   de19345146b9   3 weeks ago      188MB
nodejs              latest   053b3031f6c3   3 weeks ago      1.84KB
javaee/angular2-java-hello-world latest   11e4d501f6c3   7 months ago     1.37GB
Nitish-MBP:~ nitishbhushans$ docker commit 605885eac5c7 my_commit:v1.1
sha256:24a0e301a0091db7000c533eab64f42abb7d9da4cbcb1915edc743ad7b8
Nitish-MBP:~ nitishbhushans$ docker images
REPOSITORY          TAG      IMAGE ID            CREATED             SIZE
my_commit           v1.1     9d6dd0447d62   2 seconds ago   1.84KB
my_commit           <none>   c5399ac1734f   9 days ago       1.84KB
18 seconds ago
<none>              <none>   4fe693be3ce89   About a minute ago  1.84KB
friendlyhello       latest   cf4f04fffc289   2 days ago       195MB
ubuntu              latest   2d690327ab2e   3 weeks ago      122MB
python              2.7-slim  8b88106072d7   3 weeks ago      184MB
ubuntu              latest   de19345146b9   3 weeks ago      188MB
nodejs              latest   053b3031f6c3   3 weeks ago      1.84KB
javaee/angular2-java-hello-world latest   11e4d501f6c3   7 months ago     1.37GB
```

Docker Network:

Docker provides private network to the container and there are many different private network. We can group container to one private network to bring the logical grouping between containers.

We can also orchestrate the container interconnection and can define the container communication through 'expose' ports and 'linking' containers.

Linking between container

Approach 1 : link two container through Host machine

Connecting between Containers



From the client container, docker would link to 'server', which ideally means that at the time of 'client' container startup process, Docker would insert the 'server' details in the etc/hosts. This details also includes 'server' IP address and hence 'client' knows about the 'server'.

Limitation - Since docker would map the 'server' IP address in the 'client' container, this mode of inter container is useful only if services are running with same host machine. otherwise, once container restart, IP would change and container communication would drop.

Third and safest way to create inter container communication - Use the private network concept

step1: first create the private network

step2: link server and client container to this private network

```
nitish-MBP:~ nitishbhushans$ docker network create pri_network_1
Nitish-MBP:~ nitishbhushans$ docker run --name server --rm -ti ubuntu:14.04 bash
root@7db937a66bd:~# nc de931c72e81 1234
root@7db937a66bd:~# nc de931c72e81 1234
Hello, U there !!!!
```

```
nitish-MBP:~ nitishbhushans$ docker run --name client --rm -ti --link server:1234 ubuntu:14.04 bash
root@7db937a66bd:~# nc 172.17.0.2 1234
Hello, U r there !!!!
```

```
nitish-MBP:~ nitishbhushans$ docker run --name client --rm -ti --link server:1234 ubuntu:14.04 bash
root@7db937a66bd:~# nc 172.17.0.2 1234
Hello, U there !!!!
```

```
nitish-MBP:~ nitishbhushans$ docker run --name client --rm -ti --link server:1234 ubuntu:14.04 bash
root@7db937a66bd:~# nc 172.17.0.2 1234
Hello, U r there !!!!
```

```
nitish-MBP:~ nitishbhushans$ docker run --name client --rm -ti --link server:1234 ubuntu:14.04 bash
root@7db937a66bd:~# nc 172.17.0.2 1234
Hello, U r there !!!!
```

```
nitish-MBP:~ nitishbhushans$ docker run --name client --rm -ti --link server:1234 ubuntu:14.04 bash
root@7db937a66bd:~# nc 172.17.0.2 1234
Hello, U r there !!!!
```

```
nitish-MBP:~ nitishbhushans$ docker run --name client --rm -ti --link server:1234 ubuntu:14.04 bash
root@7db937a66bd:~# nc 172.17.0.2 1234
Hello, U r there !!!!
```

```
nitish-MBP:~ nitishbhushans$ docker run --name client --rm -ti --link server:1234 ubuntu:14.04 bash
root@7db937a66bd:~# nc 172.17.0.2 1234
Hello, U r there !!!!
```

```
nitish-MBP:~ nitishbhushans$ docker run --name client --rm -ti --link server:1234 ubuntu:14.04 bash
root@7db937a66bd:~# nc 172.17.0.2 1234
Hello, U r there !!!!
```

```
nitish-MBP:~ nitishbhushans$ docker run --name client --rm -ti --link server:1234 ubuntu:14.04 bash
root@7db937a66bd:~# nc 172.17.0.2 1234
Hello, U r there !!!!
```

```
nitish-MBP:~ nitishbhushans$ docker run --name client --rm -ti --link server:1234 ubuntu:14.04 bash
root@7db937a66bd:~# nc 172.17.0.2 1234
Hello, U r there !!!!
```

```
nitish-MBP:~ nitishbhushans$ docker run --name client --rm -ti --link server:1234 ubuntu:14.04 bash
root@7db937a66bd:~# nc 172.17.0.2 1234
Hello, U r there !!!!
```

```
nitish-MBP:~ nitishbhushans$ docker run --name client --rm -ti --link server:1234 ubuntu:14.04 bash
root@7db937a66bd:~# nc 172.17.0.2 1234
Hello, U r there !!!!
```

```
nitish-MBP:~ nitishbhushans$ docker run --name client --rm -ti --link server:1234 ubuntu:14.04 bash
root@7db937a66bd:~# nc 172.17.0.2 1234
Hello, U r there !!!!
```

```
nitish-MBP:~ nitishbhushans$ docker run --name client --rm -ti --link server:1234 ubuntu:14.04 bash
root@7db937a66bd:~# nc 172.17.0.2 1234
Hello, U r there !!!!
```

```
nitish-MBP:~ nitishbhushans$ docker run --name client --rm -ti --link server:1234 ubuntu:14.04 bash
root@7db937a66bd:~# nc 172.17.0.2 1234
Hello, U r there !!!!
```

```
nitish-MBP:~ nitishbhushans$ docker run --name client --rm -ti --link server:1234 ubuntu:14.04 bash
root@7db937a66bd:~# nc 172.17.0.2 1234
Hello, U r there !!!!
```

```
nitish-MBP:~ nitishbhushans$ docker run --name client --rm -ti --link server:1234 ubuntu:14.04 bash
root@7db937a66bd:~# nc 172.17.0.2 1234
Hello, U r there !!!!
```

```
nitish-MBP:~ nitishbhushans$ docker run --name client --rm -ti --link server:1234 ubuntu:14.04 bash
root@7db937a66bd:~# nc 172.17.0.2 1234
Hello, U r there !!!!
```

```
nitish-MBP:~ nitishbhushans$ docker run --name client --rm -ti --link server:1234 ubuntu:14.04 bash
root@7db937a66bd:~# nc 172.17.0.2 1234
Hello, U r there !!!!
```

```
nitish-MBP:~ nitishbhushans$ docker run --name client --rm -ti --link server:1234 ubuntu:14.04 bash
root@7db937a66bd:~# nc 172.17.0.2 1234
Hello, U r there !!!!
```

```
nitish-MBP:~ nitishbhushans$ docker run --name client --rm -ti --link server:1234 ubuntu:14.04 bash
root@7db937a66bd:~# nc 172.17.0.2 1234
Hello, U r there !!!!
```

```
nitish-MBP:~ nitishbhushans$ docker run --name client --rm -ti --link server:1234 ubuntu:14.04 bash
root@7db937a66bd:~# nc 172.17.0.2 1234
Hello, U r there !!!!
```

```
nitish-MBP:~ nitishbhushans$ docker run --name client --rm -ti --link server:1234 ubuntu:14.04 bash
root@7db937a66bd:~# nc 172.17.0.2 1234
Hello, U r there !!!!
```

```
nitish-MBP:~ nitishbhushans$ docker run --name client --rm -ti --link server:1234 ubuntu:14.04 bash
root@7db937a66bd:~# nc 172.17.0.2 1234
Hello, U r there !!!!
```

```
nitish-MBP:~ nitishbhushans$ docker run --name client --rm -ti --link server:1234 ubuntu:14.04 bash
root@7db937a66bd:~# nc 172.17.0.2 1234
Hello, U r there !!!!
```

```
nitish-MBP:~ nitishbhushans$ docker run --name client --rm -ti --link server:1234 ubuntu:14.04 bash
root@7db937a66bd:~# nc 172.17.0.2 1234
Hello, U r there !!!!
```

```
nitish-MBP:~ nitishbhushans$ docker run --name client --rm -ti --link server:1234 ubuntu:14.04 bash
root@7db937a66bd:~# nc 172.17.0.2 1234
Hello, U r there !!!!
```

```
nitish-MBP:~ nitishbhushans$ docker run --name client --rm -ti --link server:1234 ubuntu:14.04 bash
root@7db937a66bd:~# nc 172.17.0.2 1234
Hello, U r there !!!!
```

```
nitish-MBP:~ nitishbhushans$ docker run --name client --rm -ti --link server:1234 ubuntu:14.04 bash
root@7db937a66bd:~# nc 172.17.0.2 1234
Hello, U r there !!!!
```

```
nitish-MBP:~ nitishbhushans$ docker run --name client --rm -ti --link server:1234 ubuntu:14.04 bash
root@7db937a66bd:~# nc 172.17.0.2 1234
Hello, U r there !!!!
```

```
nitish-MBP:~ nitishbhushans$ docker run --name client --rm -ti --link server:1234 ubuntu:14.04 bash
root@7db937a66bd:~# nc 172.17.0.2 1234
Hello, U r there !!!!
```

```
nitish-MBP:~ nit
```