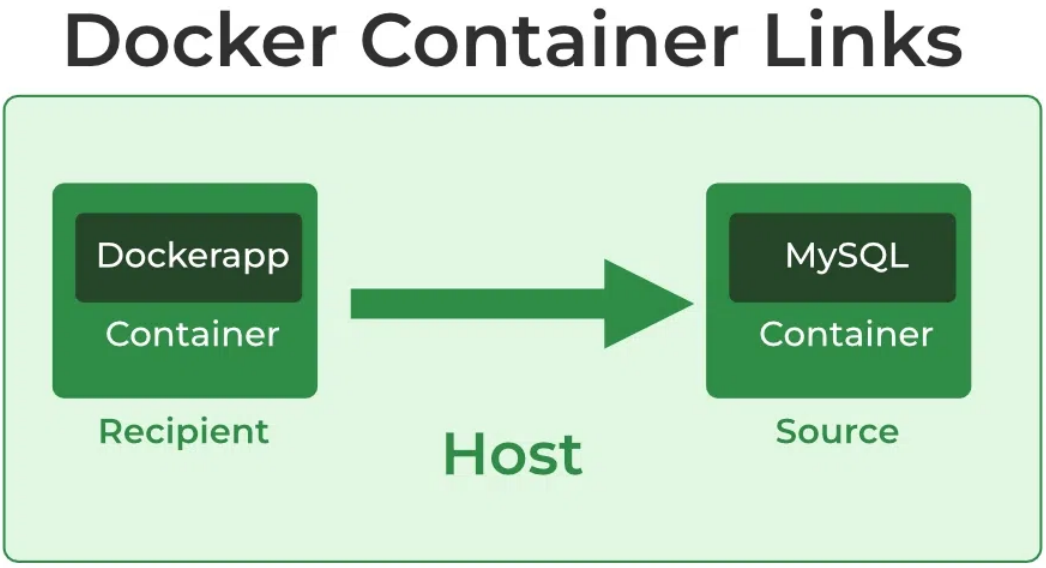
Docker – Container Linking

Last Updated : 27 Apr, 2023

Docker is a set of platforms as a service (PaaS) products that use the Operating system level visualization to deliver software in packages called containers.There are times during the development of our application when we need two containers to be able to communicate with each other. It might be possible that the services of both containers are dependent on each other. This can be done with the help of **Container Linking.**

Previously the containers were used by using the "–link" flag but that has now become deprecated and is considered a legacy command.



Connect with the Linking System

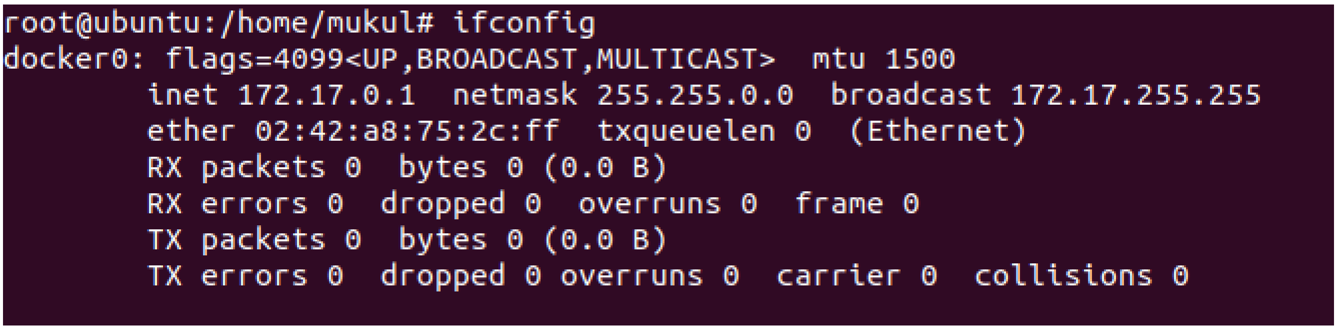
There are two ways of linking the containers

- The default way
- User-defined way

To understand the formation of a custom network between two containers we need to understand how docker assigns the network automatically.

The Default Way

Once we install docker and create a container a default bridged network is assigned to docker, by the name of Docker0. The IP is in the range of 172.17.0.0/16 (where 172.17.0.1 is assigned to the interface)



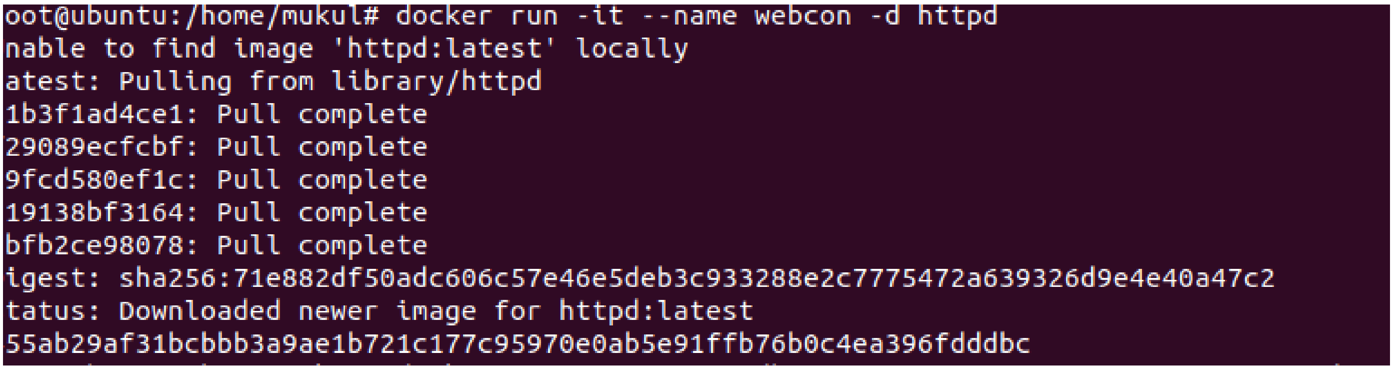
```
root@ubuntu:/home/mukul# ifconfig
docker0: flags=4099<UP,BROADCAST,MULTICAST>  mtu 1500
        inet 172.17.0.1  netmask 255.255.0.0  broadcast 172.17.255.255
        ether 02:42:a8:75:2c:ff  txqueuelen 0  (Ethernet)
        RX packets 0  bytes 0 (0.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 0  bytes 0 (0.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

Now the containers that we will create will get their IPs in the range of 172.17.0.2/16.

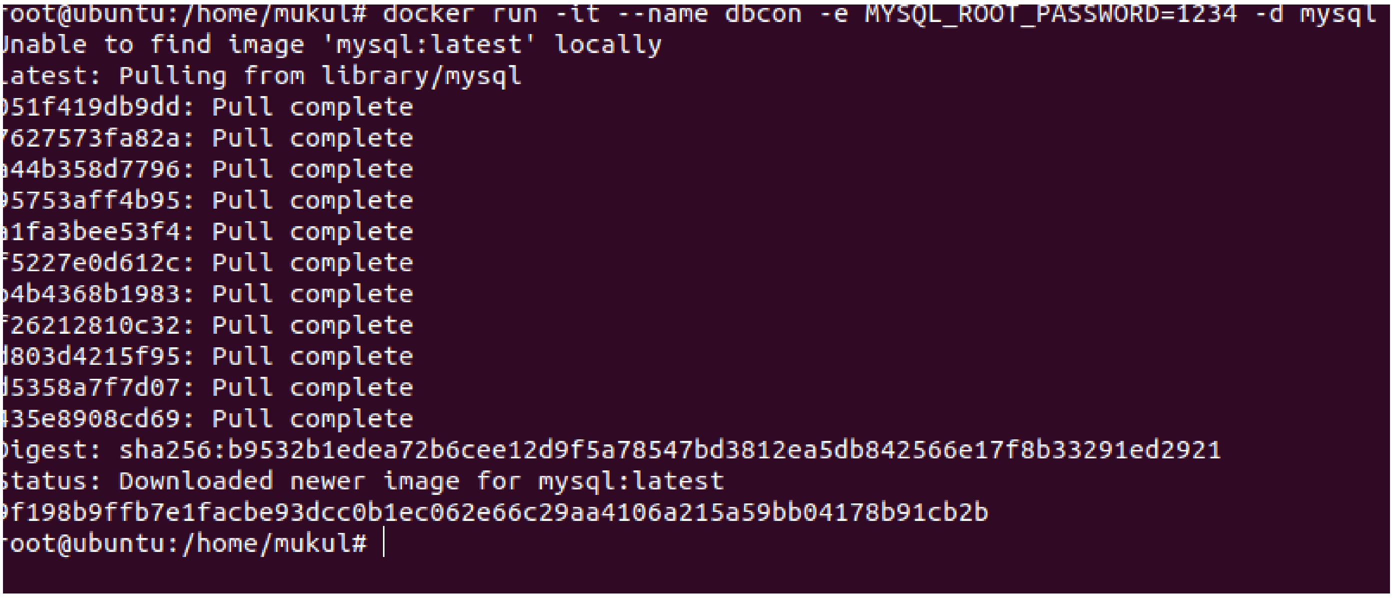
Step 1: Create two new containers, webcon, and dbcon

```
$ docker run -it --name webcon -d httpd
$ docker run -it --name dbcon  -e MYSQL_ROOT_PASSWORD=1234 -d mysql
```

You can use any image, we'll be using **MySQL** and HTTPD images in our case.



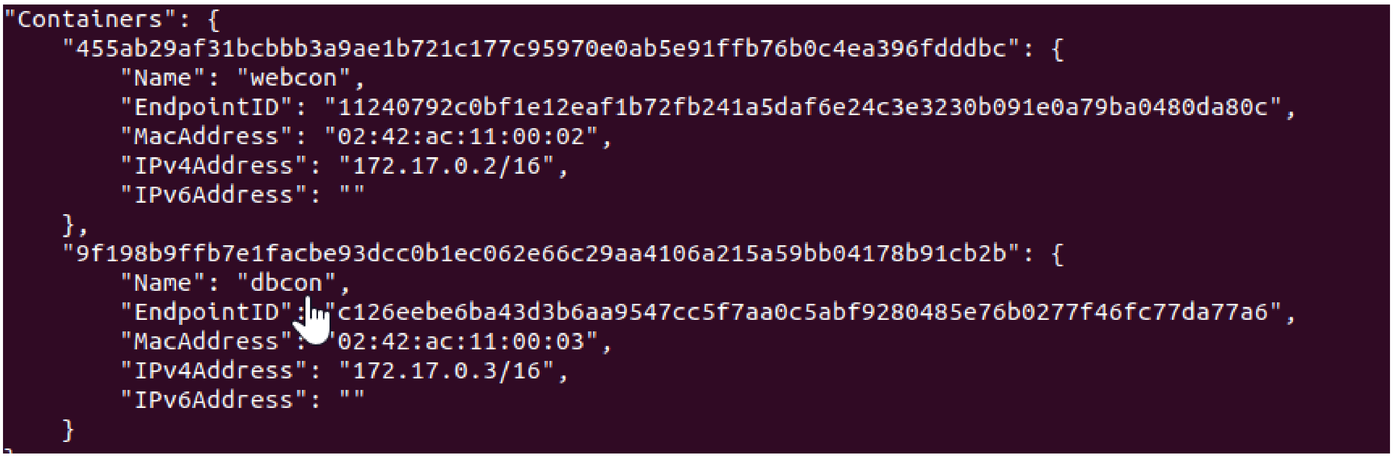
```
oot@ubuntu:/home/mukul# docker run -it --name webcon -d httpd
nable to find image 'httpd:latest' locally
atest: Pulling from library/httpd
1b3f1ad4ce1: Pull complete
29089ecfcbf: Pull complete
9fcd580ef1c: Pull complete
19138bf3164: Pull complete
bfb2ce98078: Pull complete
igest: sha256:71e882df50adc606c57e46e5deb3c933288e2c7775472a639326d9e4e40a47c2
tatus: Downloaded newer image for httpd:latest
55ab29af31bcbbb3a9ae1b721c177c95970e0ab5e91ffb76b0c4ea396fdddbc
```



```
oot@ubuntu:/home/mukul# docker run -it --name dbcon -e MYSQL_ROOT_PASSWORD=1234 -d mysql
nable to find image 'mysql:latest' locally
atest: Pulling from library/mysql
51f419db9dd: Pull complete
627573fa82a: Pull complete
44b358d7796: Pull complete
5753aff4b95: Pull complete
1fa3bee53f4: Pull complete
5227e0d612c: Pull complete
4b4368b1983: Pull complete
26212810c32: Pull complete
803d4215f95: Pull complete
5358a7f7d07: Pull complete
35e8908cd69: Pull complete
igest: sha256:b9532b1edea72b6cee12d9f5a78547bd3812ea5db842566e17f8b33291ed2921
tatus: Downloaded newer image for mysql:latest
f198b9ffb7e1facbe93dcc0b1ec062e66c29aa4106a215a59bb04178b91cb2b
oot@ubuntu:/home/mukul#
```

Step 2: Check the IPs of the new containers.

```
$ docker network inspect bridge
```

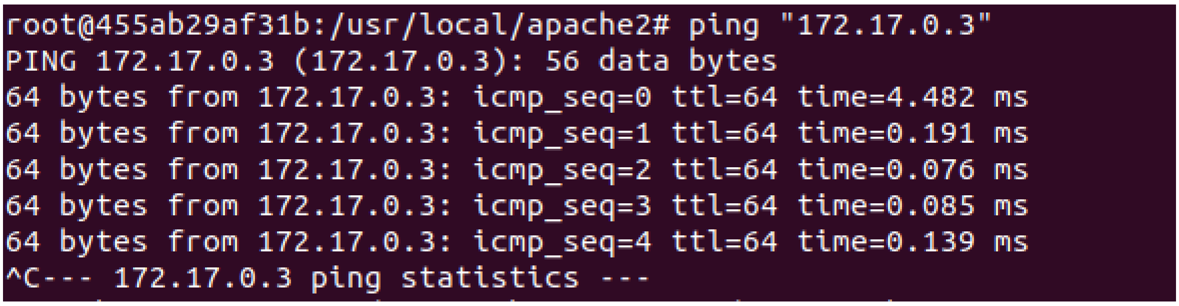


```
"Containers": {
    "455ab29af31bcbbb3a9ae1b721c177c95970e0ab5e91ffb76b0c4ea396fdddbc": {
        "Name": "webcon",
        "EndpointID": "11240792c0bf1e12eaf1b72fb241a5daf6e24c3e3230b091e0a79ba0480da80c",
        "MacAddress": "02:42:ac:11:00:02",
        "IPv4Address": "172.17.0.2/16",
        "IPv6Address": ""
    },
    "9f198b9ffb7e1facbe93dcc0b1ec062e66c29aa4106a215a59bb04178b91cb2b": {
        "Name": "dbcon",
        "EndpointID": "c126eebe6ba43d3b6aa9547cc5f7aa0c5abf9280485e76b0277f46fc77da77a6",
        "MacAddress": "02:42:ac:11:00:03",
        "IPv4Address": "172.17.0.3/16",
        "IPv6Address": ""
    }
```

With the help of these IPs, the docker host establishes a connection with the containers.

Step 3: Get inside the webcon container and try to ping the dbcon container, if you get a response back this means that the default connection is established.

```
$ docker container exec -it webcon /bin/bash
(to get into the webcon container)
$ ping "172.17.0.3"
(ping the dbcon container)
```



```
root@455ab29af31b:/usr/local/apache2# ping "172.17.0.3"
PING 172.17.0.3 (172.17.0.3): 56 data bytes
64 bytes from 172.17.0.3: icmp_seq=0 ttl=64 time=4.482 ms
64 bytes from 172.17.0.3: icmp_seq=1 ttl=64 time=0.191 ms
64 bytes from 172.17.0.3: icmp_seq=2 ttl=64 time=0.076 ms
64 bytes from 172.17.0.3: icmp_seq=3 ttl=64 time=0.085 ms
64 bytes from 172.17.0.3: icmp_seq=4 ttl=64 time=0.139 ms
^C--- 172.17.0.3 ping statistics ---
```

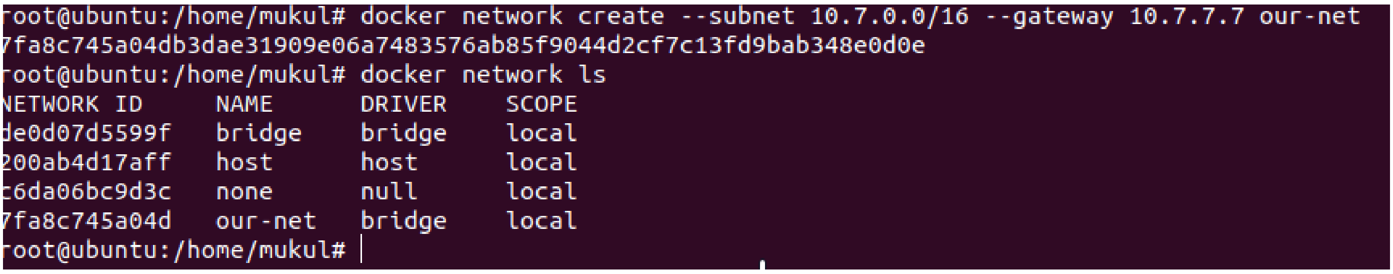
User-Defined Way

Step 1: Create a custom bridge network.

```
$ docker network create <bridge_name>
(This will create a bridge with custom subnet and gateway)
```

We can also give our own subnet and gateway.

```
$ docker network create --subnet <your_subnet>
 --gateway <Your_gateway> bridgename
```



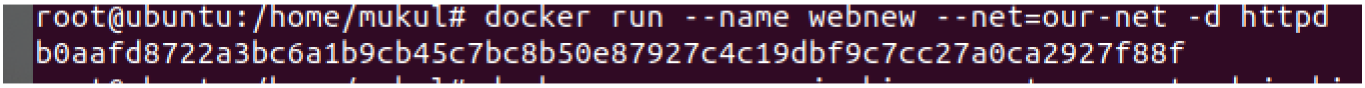
```
root@ubuntu:/home/mukul# docker network create --subnet 10.7.0.0/16 --gateway 10.7.7.7 our-net
7fa8c745a04db3dae31909e06a7483576ab85f9044d2cf7c13fd9bab348e0d0e
root@ubuntu:/home/mukul# docker network ls
NETWORK ID     NAME      DRIVER    SCOPE
de0d07d5599f   bridge    bridge    local
200ab4d17aff   host      host      local
c6da06bc9d3c   none      null      local
7fa8c745a04d   our-net   bridge    local
root@ubuntu:/home/mukul# 
```

Step 2: Verify if your network has been created or not.

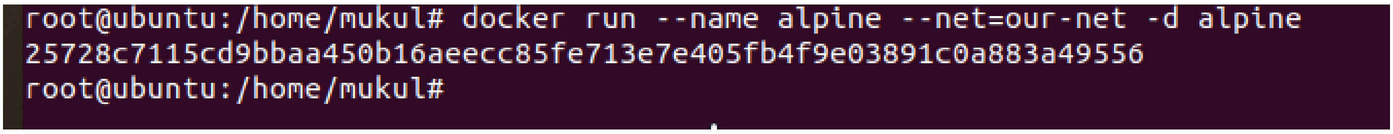
```
$ docker network ls
```

Step 3: Associate or link the two containers on the network that you just created by using the “–net” flag.

```
$ docker run --name <container_name>
 --net=<custom_net>
-d <image_name>
```



```
root@ubuntu:/home/mukul# docker run --name webnew --net=our-net -d httpd
b0aafd8722a3bc6a1b9cb45c7bc8b50e87927c4c19dbf9c7cc27a0ca2927f88f
```



```
root@ubuntu:/home/mukul# docker run --name alpine --net=our-net -d alpine
25728c7115cd9bbaa450b16aeecc85fe713e7e405fb4f9e03891c0a883a49556
root@ubuntu:/home/mukul#
```

We have used httpd and Alpine images for our containers.

Step 4: Get inside the webnew container(IP- 10.7.0.10) and ping the alpine container(IP- 10.7.0.2)

```
$ docker exec -it webnew /bin/bash
$ ping "10.7.0.2" (inside the webnew container)
```

If you start receiving the packets from the Alpine container then you have successfully established a connection between both containers using your own OUR-NET network. So this is how you can create your own custom bridged network which allows you to establish a connection between your container.

The Importance of Naming

Docker mainly depends upon the names of the containers we can see in the above example whenever you create a new container the name gets created automatically we can also name our container is will us in two different ways.

- By giving the container a name, we can keep track of the type of program that is executed inside of it, such as a web application or a database.
- If a web application wants to communicate with DB servers, for instance, it can act as a barrier between the two containers like a connection link.

We can name our container with the help of the command shown below

```
(--name) docker run -d -P --name <name/imagename/tag>
```

Environment Variables

If suppose the developer mentioned some –env (Environmental variables) in the source code by which we can connect to the database server, for example, Username and password then while creating the container we set the username and password as shown in the below command.

```
docker run -d --name <name>  -e USERNAME=<***> -e  PASSWORD=<***>  --
network <****>
```

We can set the above-mentioned env variables to the database container by using the following command.

```
docker run -d -p <port> --name <name>  -e HOSTNAME=<***> -e
USERNAME=<***> -e  PASSWORD=<***> --network <***>
```

Updating the /etc/hosts file

Docker adds a host entry for the source container to the /etc/hosts apart from the environmental variables we provide the command to link two containers is mentioned below.

```
docker run -t -i --rm --link <Mention Entries>
```

To check the list of entries that have been mentioned in the /etc/hosts file we can use the below command.

```
cat  /etc/hosts
```

Comment

More info

Next Article

How to Manage Docker Containers?

Similar Reads

What is Docker?

Docker is a set of Platforms as a service (PaaS) products that use Operating system-level virtualization to deliver software in packages called containers. Containers are isolated from one another and bundle the...

15+ min read

Introduction to Docker

Docker Installation

Docker Commands

Docker Images

Docker Containers

Containerization using Docker

Docker is the containerization platform that is used to package your application and all its dependencies together in the form of containers to make sure that your application works seamlessly in any...

🕒 9 min read

Virtualisation with Docker Containers

In a software-driven world where omnipresence and ease of deployment with minimum overheads are the major requirements, the cloud promptly takes its place in every picture. Containers are creating thei...

🕒 9 min read

Docker - Docker Container for Node.js

Node.js is an open-source, asynchronous event-driven JavaScript runtime that is used to run JavaScript applications. It is widely used for traditional websites and as API servers. At the same time, a Docker...

🕒 12 min read

Remove All Containers and Images in Docker

Pre-requisite: Docker In Docker , if we have exited a container without stopping it, we then need to stop them manually as it has not stopped on exit. Similarly, for images, we need to delete them from top to...

🕒 4 min read

How to Push a Container Image to a Docker Repository?

In this article we will look into how you can push a container image to a Docker Repo. We're going to use Docker Hub as a container registry, that we're going to push our Docker image to. Follow the belo...

🕒 2 min read

Docker - Container Linking

Docker is a set of platforms as a service (PaaS) products that use the Operating system level visualization to deliver software in packages called containers.There are times during the development...

🕓 4 min read

How to Manage Docker Containers?

Before virtualization, the management of web servers and web applications was tedious and much less effective. Thanks to virtualization, this task has been made much easier. This was followed by...

🕓 13 min read

Mounting a Volume Inside Docker Container

When you are working on a micro-service architecture using Docker containers, you create multiple Docker containers to create and test different components of your application. Now, some of those...

🕓 10 min read

Difference between Docker Image and Container

Pre-requisite: Docker Docker builds images and runs containers by using the docker engine on the host machine. Docker containers consist of all the dependencies and software needed to run an application i...

🕓 5 min read

Difference between Virtual Machines and Containers

Virtual machines and Containers are two ways of deploying multiple, isolated services on a single platform. Virtual Machine:It runs on top of an emulating software called the hypervisor which sits...

🕓 2 min read

How to Install Linux Packages Inside a Docker Container?

Once you understand how to pull base Docker Images from the Docker registry, you can now simply pull OS distributions such as Ubuntu, CentOS, etc directly from the Docker hub. However, the OS Image tha...

🕓 2 min read

Copying Files to and from Docker Containers

While working on a Docker project, you might require copying files to and from Docker Containers and your Local Machine. Once you have built the Docker Image with a particular Docker build context,...

🕓 9 min read

How to Run MongoDB as a Docker Container?

MongoDB is an open-source document-oriented database designed to store a large scale of data and allows you to work with that data very efficiently. It is categorized under the NoSQL (Not only SQL)...

🕓 4 min read

Docker - Docker Container for Node.js

Node.js is an open-source, asynchronous event-driven JavaScript runtime that is used to run JavaScript applications. It is widely used for traditional websites and as API servers. At the same time, a Docker...

12 min read

Docker - Container for NGINX

Docker is an open-source platform that enables developers to easily develop, ship, and run applications. It packages an application along with its dependencies in an isolated virtual container which usually...

11 min read

How to Provide the Static IP to a Docker Container?

Docker is an open-source project that makes it easier to create, deploy and run applications. It provides a lightweight environment to run your applications.It is a tool that makes an isolated environment inside...

2 min read

Docker Compose

Docker Swarm

Docker Networking

Docker Registry

Article Tags : DevOps Docker Docker Container