IBM Integration Bus
V10.0

IIB Docker Container Deployment

Part A – Get and Install Docker

Part B – Download, Start and test the IIB Docker Container

V1.4 February 2015

Table of Contents

Overview

Description

The steps in this lab will take you through set and getting started with IIB Docker containers

You will complete the following steps. If you are an existing Docker user you will be able to jump ahead of the initial instructions.

1. Part 1 – Get and install Docker
    a. Get a Docker ID
    b. Get and install Docker on the operating system of your choice
    c. Verify the Docker install and set up
2. Part 2 – Instantiate and verify the IIB runtime Docker container
    a. Obtain and instantiate the IIB runtime Docker container
    b. Verify the running container
    c. Verify the running IIB node in the container

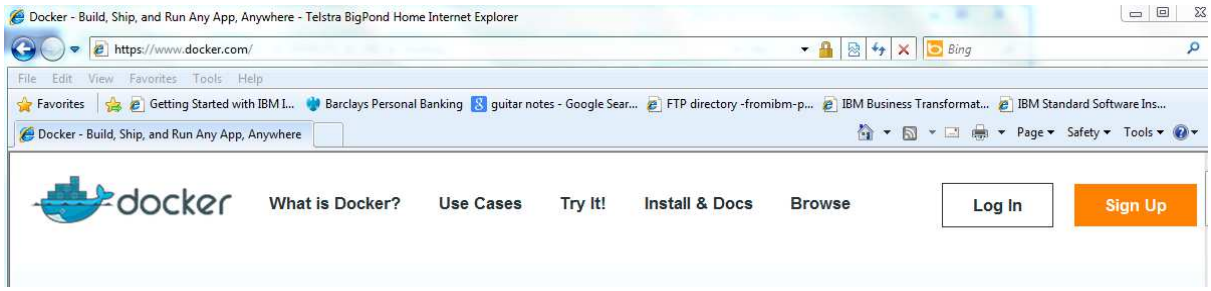
Pre-requisites

None

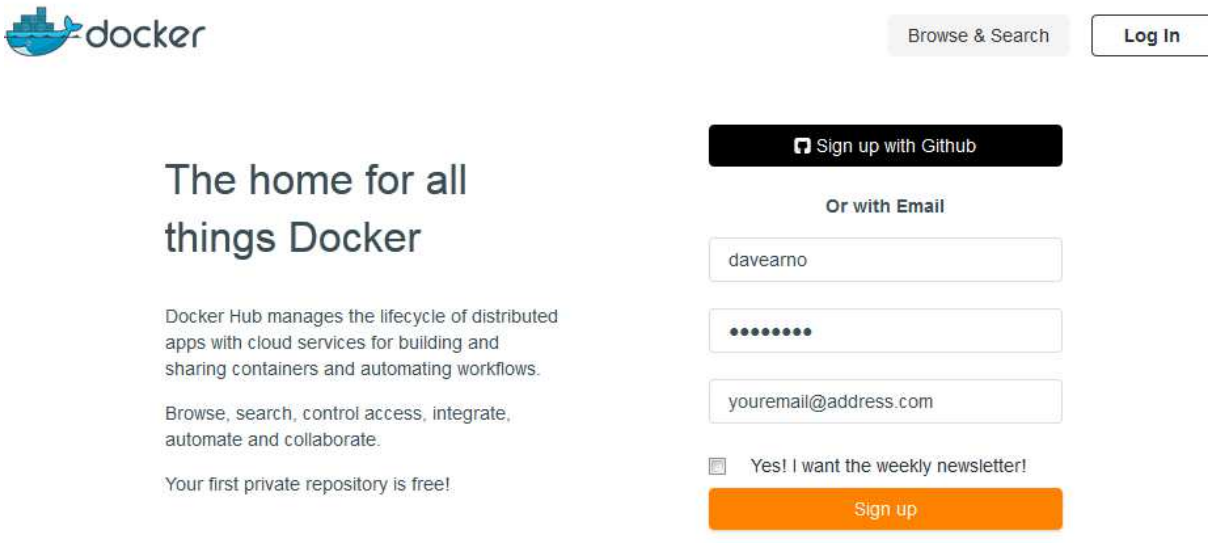
Part A – Get and Install

Get a Docker ID

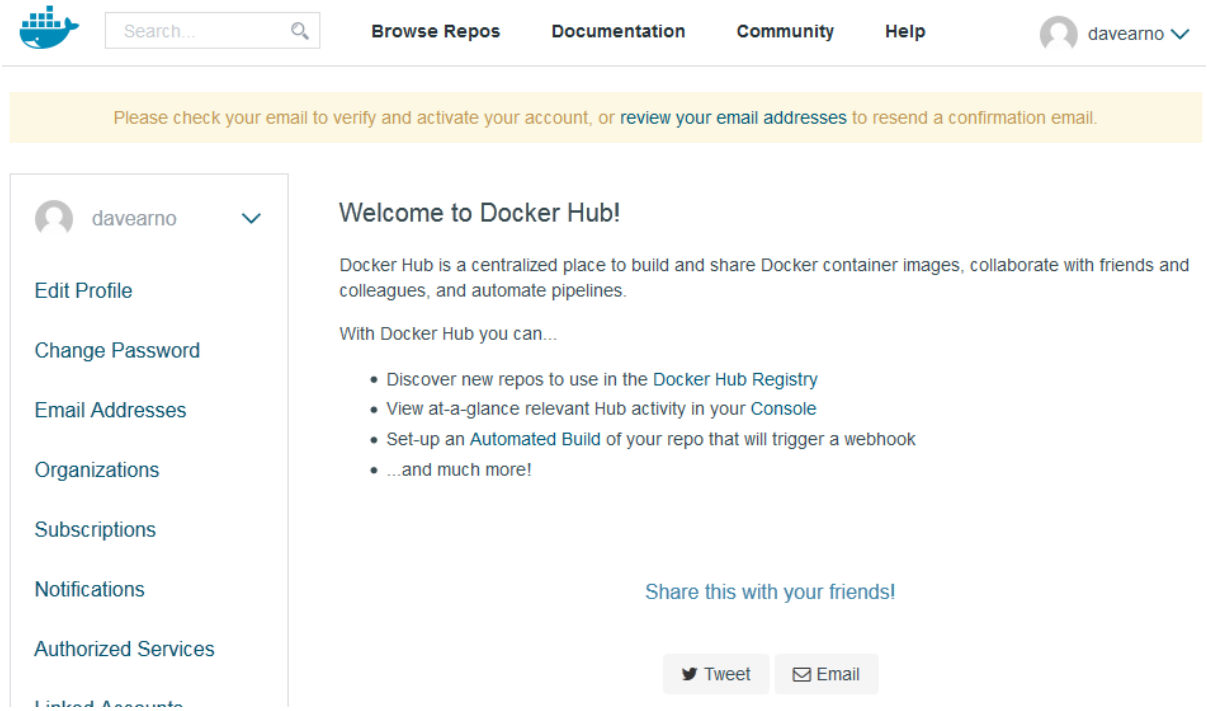
Go to the Docker website https://www.docker.com/



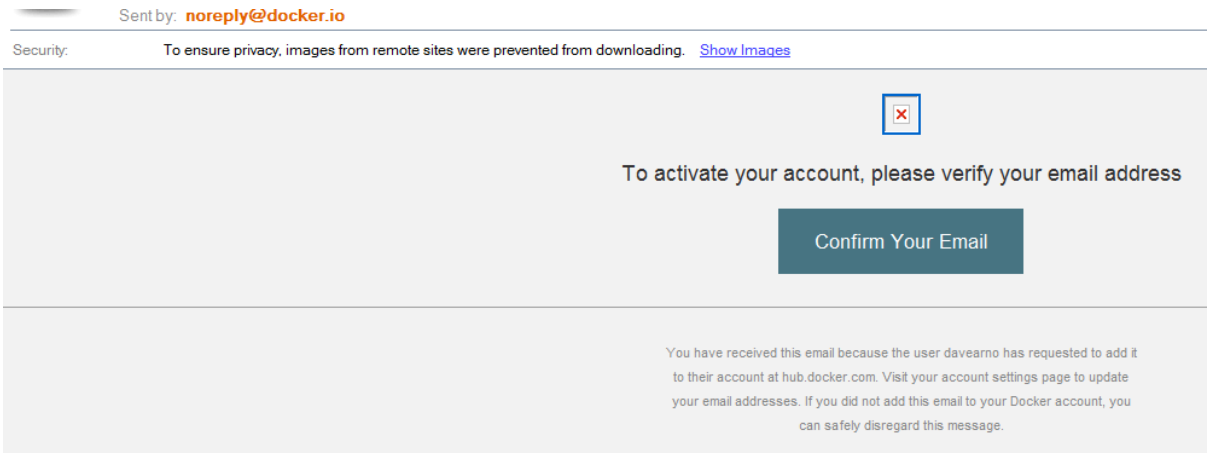
Select Sign Up



Choose a Docker ID name and password and supply and email address



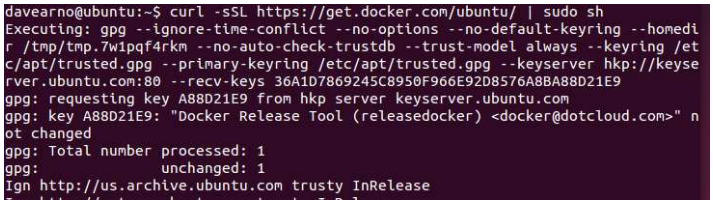
You will need to activate the Docker account via the confirmation email.



Hit the Confirm Your Email button

Get and install Docker – Option 1 – for Ubuntu via Curl

```
$ sudo apt-get install curl
$ curl -sSL https://get.docker.com/ubuntu/ | sudo sh
```

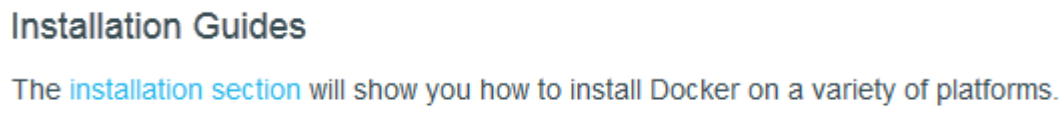


If you installed via option 1 skip forward to the "Verifying the Docker Install and Set Up" on page 7.

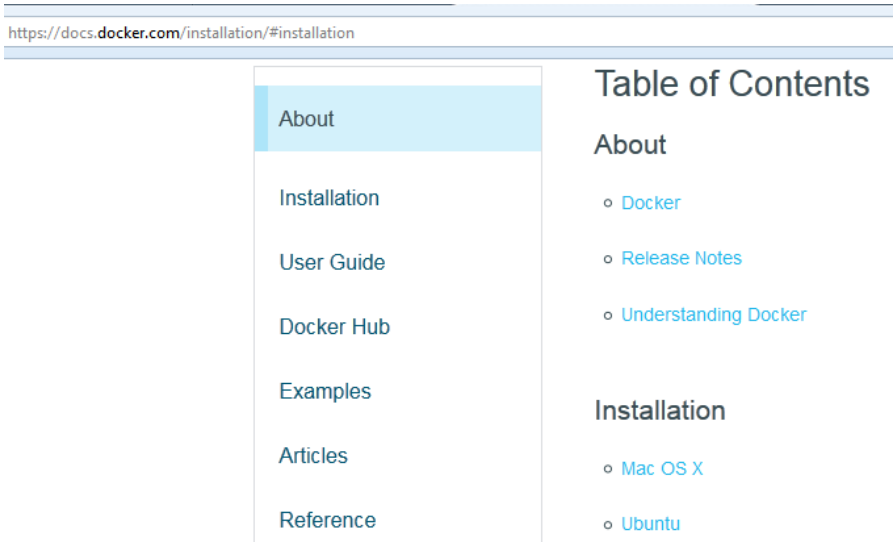
Get and install Docker – Option 2 via the Docker online instructions for all platforms

Return to Docker website https://www.docker.com/

Select Install & Docs



Page down to the Installation Guides and click on the installation section link



Select your target platform

We will use Ubuntu for example purposes.
Click on Ubuntu

Page down to the Docker maintained package installation and follow the instructions.

Docker-maintained Package Installation

If you'd like to try the latest version of Docker:

First, check that your APT system can deal with `https` URLs: the file `/usr/lib/apt/methods/https` should exist. If it doesn't, you need to install the package `apt-transport-https`.

```
[ -e /usr/lib/apt/methods/https ] || {
  apt-get update
  apt-get install apt-transport-https
}
```

Then, add the Docker repository key to your local keychain.

```
$ sudo apt-key adv --keyserver hkp://keyserver.ubuntu.com:80 --recv-keys 36A1D7869245C8950F966E92D8576A
8BA88D21E9
```

You can copy and paste the commands from below:

Check that the APT can deal with HTTPS urls

```
$ ls -lt /usr/lib/apt/methods/https
```

```
davearno@ubuntu:~$ ls -lt /usr/lib/apt/methods/https
-rwxr-xr-x 1 root root 76928 Apr 10  2014 /usr/lib/apt/methods/https
```

Add the Docker repository key to your local keychain.

```
$ sudo apt-key adv --keyserver hkp://keyserver.ubuntu.com:80 --recv-keys
```

```
36A1D7869245C8950F966E92D8576A8BA88D21E9
```

```
davearno@ubuntu:~$ sudo apt-key adv --keyserver hkp://keyserver.ubuntu.com:80 --
recv-keys 36A1D7869245C8950F966E92D8576A8BA88D21E9
[sudo] password for davearno:
Executing: gpg --ignore-time-conflict --no-options --no-default-keyring --homedi
r /tmp/tmp.jGSfoOLTta --no-auto-check-trustdb --trust-model always --keyring /et
c/apt/trusted.gpg --primary-keyring /etc/apt/trusted.gpg --keyserver hkp://keyse
rver.ubuntu.com:80 --recv-keys 36A1D7869245C8950F966E92D8576A8BA88D21E9
gpg: requesting key A88D21E9 from hkp server keyserver.ubuntu.com
gpg: key A88D21E9: public key "Docker Release Tool (releasedocker) <docker@dotcl
oud.com>" imported
gpg: Total number processed: 1
gpg:               imported: 1  (RSA: 1)
```

Add the Docker repository to your apt sources list, update and install the `lxc-docker` package.
You may receive a warning that the package isn't trusted. Answer yes to continue installation.

```
$sudo sh -c "echo deb https://get.docker.com/ubuntu docker main\
> /etc/apt/sources.list.d/docker.list"

$sudo apt-get update
```

```
Hit http://security.ubuntu.com trusty-security/universe Sources
Hit http://us.archive.ubuntu.com trusty-updates Release
Hit http://security.ubuntu.com trusty-security/multiverse Sources
Hit http://us.archive.ubuntu.com trusty-backports Release
Hit http://security.ubuntu.com trusty-security/main amd64 Packages
Hit http://us.archive.ubuntu.com trusty/main Sources
Hit http://security.ubuntu.com trusty-security/restricted amd64 Packages
Hit http://us.archive.ubuntu.com trusty/restricted Sources
Hit http://us.archive.ubuntu.com trusty/universe Sources
Hit http://security.ubuntu.com trusty-security/universe amd64 Packages
Hit http://us.archive.ubuntu.com trusty/multiverse Sources
Hit http://security.ubuntu.com trusty-security/multiverse amd64 Packages
Hit http://us.archive.ubuntu.com trusty/main amd64 Packages
Hit http://security.ubuntu.com trusty-security/main i386 Packages
Ign https://get.docker.com docker/main Translation-en_US
Ign https://get.docker.com docker/main Translation-en
Fetched 7,610 B in 2min 41s (47 B/s)
Reading package lists... Done
davearno@ubuntu:~$
```

Install Docker on your system:

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

```
davearno@ubuntu:~$ sudo apt-get install lxc-docker
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following extra packages will be installed:
  lxc-docker-1.5.0
The following packages will be REMOVED:
  docker.io
The following NEW packages will be installed:
  lxc-docker lxc-docker-1.5.0
0 upgraded, 2 newly installed, 1 to remove and 466 not upgraded.
Need to get 4,635 kB of archives.
After this operation, 9,555 kB disk space will be freed.
Do you want to continue? [Y/n] Y
Get:1 https://get.docker.com/ubuntu/ docker/main lxc-docker-1.5.0
,632 kB]
0% [1 lxc-docker-1.5.0 7,702 B]
```

Verify the Docker install and set up

To verify that everything has worked as expected. This step will pulldown and start a minimal ubuntu docker container. You can cancel it if the download looks time consuming.

```
$ sudo docker run --rm -i -t ubuntu /bin/bash
```

Enable tab-completion on your system:

```
$ source /etc/bash_completion.d/docker.io
```

Finally, add your user to the Docker group

```
$ sudo adduser username docker
```

```
davearno@ubuntu:~$ sudo adduser davearno docker
Adding user `davearno' to group `docker' ...
Adding user davearno to group docker
Done.
```

Check your Docker version and explore the docker commands

```
$ sudo docker –version
```

```
davearno@ubuntu:~$ docker --version
Docker version 1.5.0, build a8a31ef
```

Some useful commands for getting help

```
$ sudo docker --help
$ sudo docker command --help
```

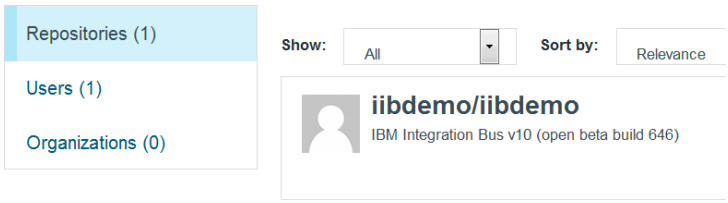
Part B – Instantiate and verify the Docker container

Obtain and instantiate the IIB runtime Docker container

Login to the Docker website https://hub.docker.com/ and search for the IIBDEMO image.

Type iibdemo and hit enter.

You should get the following result



Click on iibdemo/iibdemo to review information about the IIB Demo Docker image.

Pull down the IIBDEMO image from the Docker registry server

The iibdemo image was built with both IIB v10 Beta and IBM MQ for Developers built in. Although no MQ queue managers have been created in the image at this time. Therefore, the image is currently 1.9GB in size.

```
$sudo docker pull iibdemo/iibdemo
```

```
davearno@ubuntu:~$ sudo docker pull iibdemo/iibdemo
[sudo] password for davearno:
Pulling repository iibdemo/iibdemo
03949285bd78: Pulling image (latest) from iibdemo/iibdemo, endpoint: https://reg
03949285bd78: Downloading 4.319 MB/1.972 GB 2h22m54s
511136ea3c5a: Download complete
5b12ef8fd570: Download complete
dade6cb4530a: Download complete
cb1b6d0cd2ed: Download complete
```

```
davearno@ubuntu:~$ sudo docker pull iibdemo/iibdemo
[sudo] password for davearno:
Pulling repository iibdemo/iibdemo
03949285bd78: Download complete
511136ea3c5a: Download complete
5b12ef8fd570: Download complete
dade6cb4530a: Download complete
cb1b6d0cd2ed: Download complete
Status: Downloaded newer image for iibdemo/iibdemo:latest
davearno@ubuntu:~$
```

List the docker images

```
$ sudo docker images
```

```
davearno@ubuntu:~$ docker images
REPOSITORY          TAG                 IMAGE ID            CREATED             V
IRTUAL SIZE
iibdemo/iibdemo     latest              03949285bd78        19 hours ago        3
.087 GB
centos              latest              dade6cb4530a        11 days ago         2
10.1 MB
```

Creating and starting the Docker image as a container for the first time. On creation we are mapping the ports that the IIB runtime will use.

```
$ sudo docker run -di --user=iibadm --name=dev_esb01 -p 7080:7080 -p
1414:1414 -p 4414:4414 -p 6666:6666 -p 7800:7800 --hostname=dev_esb01 --
workdir=/home/iibadm iibdemo/iibdemo:latest /bin/bash -l
```

```
davearno@ubuntu:~$ docker run -di --user=iibadm --name=dev_esb01 -p 7080:7080 -p
1414:1414 -p 4414:4414 -p 6666:6666 -p 7800:7800 --hostname=dev_esb01 --workdir=/
home/iibadm iibdemo/iibdemo:latest /bin/bash -l
3a70e7900ed8dd3cab43ac52327c3bf5195f6bdc269c4f6e509f7fd7d977dc7e
davearno@ubuntu:~$
```

Start the IIB node in the container and list running containers

```
$ sudo docker exec -id dev_esb01 /bin/bash -lc 'iib start TESTNODE'
$ sudo docker ps
```

```
davearno@ubuntu:~$ docker exec -id dev_esb01 /bin/bash -lc 'iib start TESTNODE'
davearno@ubuntu:~$ docker ps
CONTAINER ID        IMAGE                    COMMAND             CREATED
    STATUS              PORTS                                                NAMES
3a70e7900ed8        iibdemo/iibdemo:latest   "/bin/bash -l"      2 minutes ago
    Up 2 minutes        0.0.0.0:1414->1414/tcp, 0.0.0.0:4414->4414/tcp, 0.0.0.0:6
666->6666/tcp, 0.0.0.0:7080->7080/tcp, 0.0.0.0:7800->7800/tcp   dev_esb01
```

Verify the running container and it's processes

```
$sudo docker top container_name
```

```
davearno@ubuntu:~$ docker top dev_esb01
UID                 PID                 PPID                C                   S
TIME                TTY                 TIME                CMD
1001                4203                1503                0                   1
6:22                ?                   00:00:00            /bin/bash -l
1001                4490                1                   0                   1
6:24                ?                   00:00:00            bipservice TESTNODE
1001                4495                4490                6                   1
6:24                ?                   00:00:14            bipbroker TESTNODE
1001                4551                4495                0                   1
6:24                ?                   00:00:00            bipMQTT -c /var/mqsi/c
omponents/TESTNODE/config/TESTNODE -p 11883
1001                4565                4495                3                   1
6:24                ?                   00:00:07            DataFlowEngine TESTNOD
E 6f7e050e-d070-434d-949b-e07237c71e13 service
```

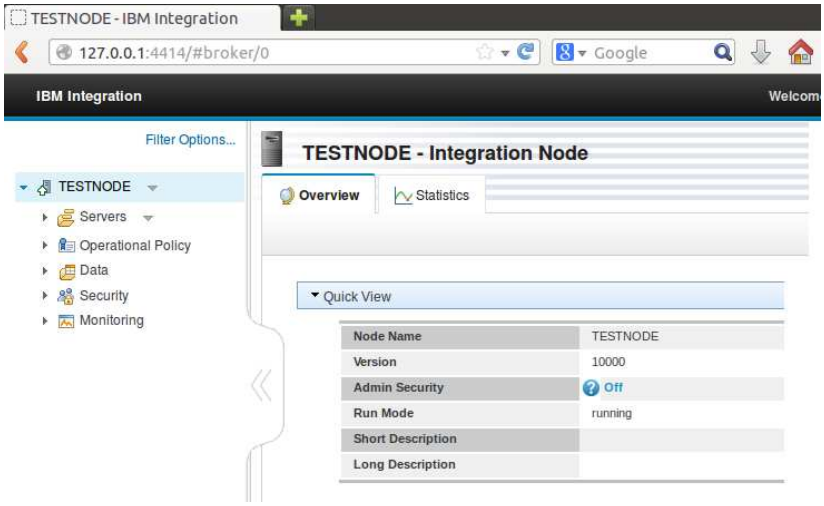
List the mapped ports for the container.

```
$sudo docker port container_name
```

```
davearno@ubuntu:~$ docker port dev_esb01
7800/tcp -> 0.0.0.0:7800
1414/tcp -> 0.0.0.0:1414
4414/tcp -> 0.0.0.0:4414
6666/tcp -> 0.0.0.0:6666
7080/tcp -> 0.0.0.0:7080
```

Connect the IIB Web GUI to the running IIB node in the container

Start your browser and enter the url http://127.0.0.1:4414



Verify the running IIB node in the container

There is a message flow already deployed and running on the IIB Test Node in the IIBDEMO container. The message flow is a simple HTTP echo flow.

Use curl or your favourite tool to send an HTTP post to the message flow.

```
$ echo '{"text": "Hello **world**!"}' | curl -d @-
http://127.0.0.1:7800/echo
```

```
davearno@ubuntu:~$ echo '{"text": "Hello **world**!"}' | curl -d @- http://127.0
.0.1:7800/echo
{"text": "Hello **world**!"}davearno@ubuntu:~$
```

Stopping the running IIB node and the container

For reference purposes here are the commands for starting and stopping the container.

```
$docker exec -id  dev_esb01  "/bin/bash -lc 'iib stop TESTNODE' "
```

```
$docker stop dev_esb01
```

```
davearno@ubuntu:~/Downloads/iib-10.0.646.0$ docker exec -id  dev_esb01  "/bin/ba
sh -lc 'iib stop TESTNODE' "
davearno@ubuntu:~/Downloads/iib-10.0.646.0$ docker stop dev_esb01
dev_esb01
davearno@ubuntu:~/Downloads/iib-10.0.646.0$
```

Re-starting IIB container and the IIB Node

```
$docker start dev_esb01
```

```
davearno@ubuntu:~/Downloads/iib-10.0.646.0$ docker start dev_esb01
dev_esb01
```

```
$docker exec -id dev_esb01 /bin/bash -lc 'iib start TESTNODE'
```

```
davearno@ubuntu:~/Downloads/iib-10.0.646.0$ sudo docker exec -id dev_esb01 /bin
bash -lc 'iib start TESTNODE'
[sudo] password for davearno:
davearno@ubuntu:~/Downloads/iib-10.0.646.0$
```