



Docker Cluster Deployment Guide

Version 1.8.0

Contents

Copyright Notice	3
Document Revision History	4
Preparations	5
Network	6
Port Usage Outside of Cluster Group	6
Port Usage Inside of Cluster Group	6
System Requirements	7
Supported Platforms	7
Cluster Size	7
Persistent Storage	7
Browsers	7
Cluster Setup	8
Individual Node DNS Entries	8
Load Balancing	8
Round-Robin DNS	8
Hardware (Websocket Enabled)	8
SSL Certificates	8
Docker	9
Docker Volumes	9
Import Beam Image	10
First Beam Node	11
IP Address	12
Additional Beam Nodes	13

Copyright Notice

No part of this document may be reproduced or transmitted in any form or by any means, electronic or mechanical, for any purpose, without express written permission. Under the law, reproducing includes translating into another language or format.

The software is protected by United States copyright laws and international treaty provision. Therefore, you must treat the software like any other copyrighted material (e.g. a book or sound recording).

Document Revision History

June 13th, 2018

- Initial release of documentation.

Deployment

Preparations

To set up Beam, you must have:

- Beam Docker image
- Supported Docker infrastructure
- IP addresses of cluster nodes
- Nginx compatible SSL certificate and SSL certificate key

Network

Port Usage Outside of Cluster Group

Protocol	Port	Direction	Purpose
HTTPS	443	Inbound	Beam API
HTTPS	443	Outbound	VCC API

Port Usage Inside of Cluster Group

Protocol	Port	Direction	Purpose
TCP	4369	Inbound/Outbound	Time series database (epmd)
TCP	8087	Inbound/Outbound	Time series database (client)
TCP	8099	Inbound/Outbound	Time series database (handoff)
TCP	6000-7999	Inbound/Outbound	Time series database (peer)
TCP	26379	Inbound/Outbound	Cluster messaging
HTTPS	443	Inbound/Outbound	Beam API
SSH	22	Inbound/Outbound	Cluster administration

System Requirements

Supported Platforms

- Docker

Cluster Size

The recommended size of a Beam cluster is 5 nodes on 5 distinct physical hosts.

Persistent Storage

Docker containers do not hold onto data on upgrades, It's recommended to use persistent storage such as Docker volumes to save time series data

Browsers

The Beam interface is supported on the latest versions of Firefox, Internet Explorer, Edge, Chrome, and Safari.

Cluster Setup

Clusters are headless and all nodes are functionally identical.

Individual Node DNS Entries

Individual nodes do not require distinct DNS entries but can be assigned one for administrative convenience.

Load Balancing

Nodes do not require session affinity and utilize long-lived websocket connections.

The cluster can operate in two load balancing configurations:

Round-Robin DNS

All node IP addresses are assigned to a single DNS entry.

Hardware (Websocket Enabled)

Nodes can be used with hardware load balancers such as those available from Cisco or F5 for fault-tolerance. Hardware load balancers **must be configured for use with websockets**. Refer to your load balancer's documentation for instructions on enabling websockets.

SSL Certificates

All cluster nodes share a single SSL certificate and certificate key to communicate with external services.

The SSL certificate and certificate key should be Nginx compatible. See - http://nginx.org/en/docs/http/configuring_https_servers.html - for more information.

Docker

Beam docker image requires persistent storage to store the time series data and

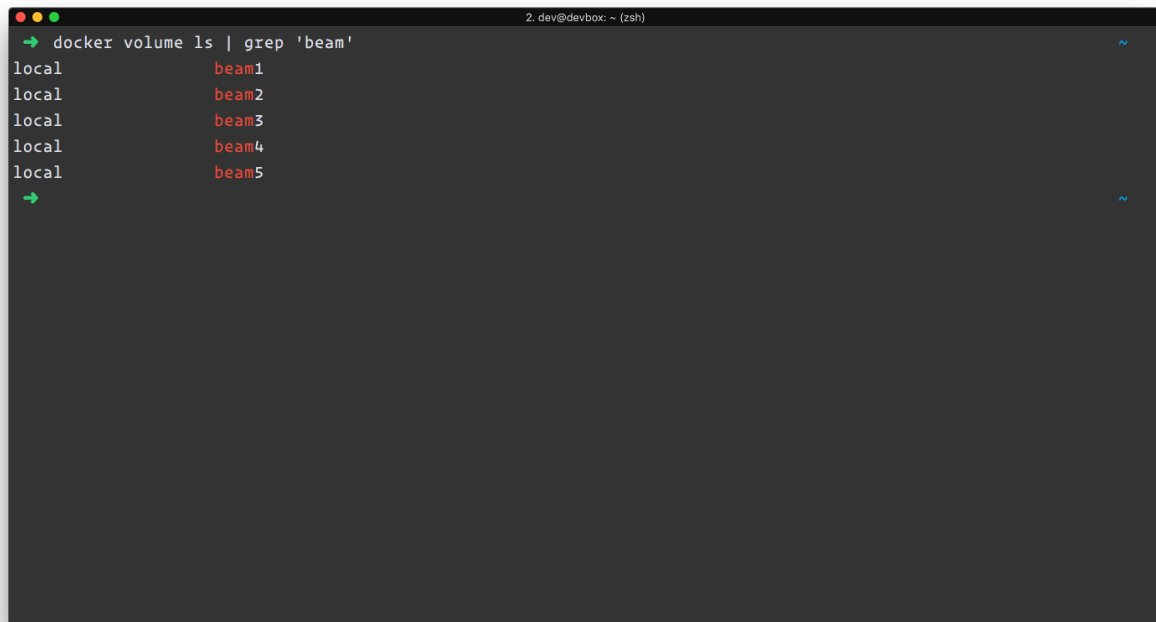
Docker Volumes

Each beam instance require a unique docker volume, so for a cluster for 5 beam nodes 5 docker volumes are required. Run the following command 5 times by substituting **x** by 1 through 5.

```
docker volume create beamx
```

Verify if all the volumes are created by running the following command

```
docker volume ls
```



```
2. dev@devbox: ~ (zsh)
→ docker volume ls | grep 'beam'
local          beam1
local          beam2
local          beam3
local          beam4
local          beam5
→
```

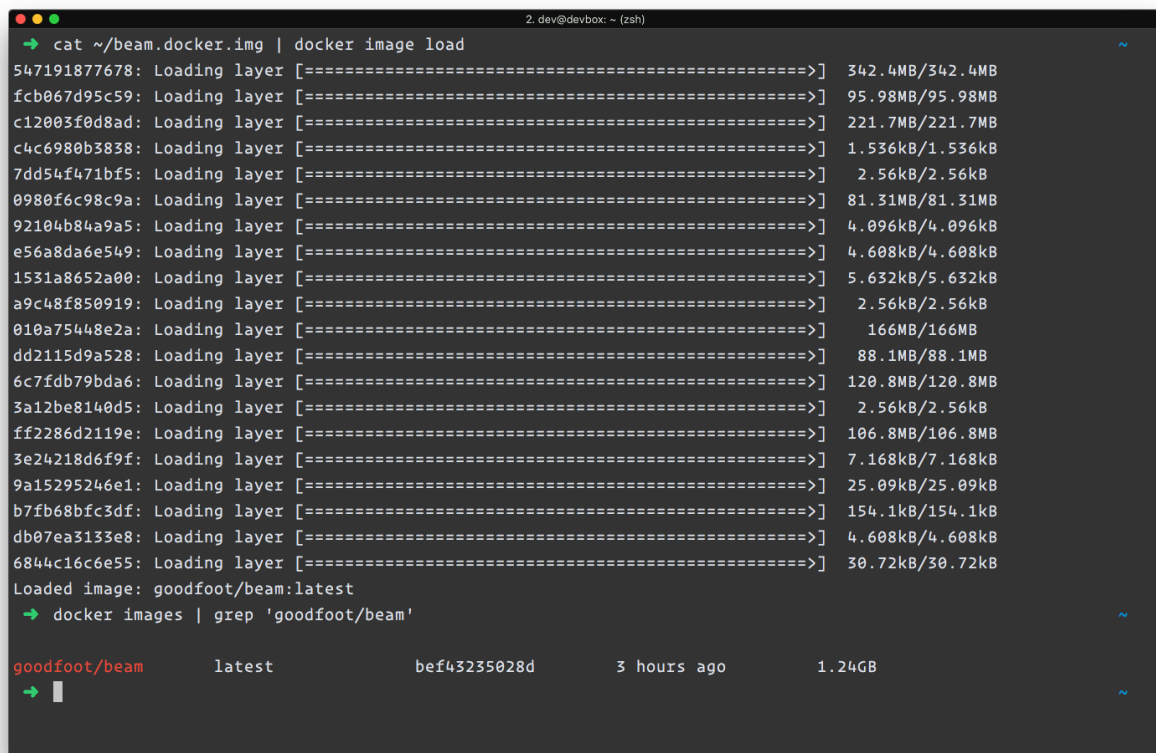
Import Beam Image

Run the following command to import Beam docker image

```
cat /path/to/beam.docker.img | docker image load
```

Verify if the import was successful by listing docker images

```
docker images | grep 'goodfoot/beam'
```



```
2. dev@devbox: ~ (zsh)
→ cat ~/beam.docker.img | docker image load
547191877678: Loading layer [====>] 342.4MB/342.4MB
fcb067d95c59: Loading layer [====>] 95.98MB/95.98MB
c12003f0d8ad: Loading layer [====>] 221.7MB/221.7MB
c4c6980b3838: Loading layer [====>] 1.536kB/1.536kB
7dd54f471bf5: Loading layer [====>] 2.56kB/2.56kB
0980f6c98c9a: Loading layer [====>] 81.31MB/81.31MB
92104b84a9a5: Loading layer [====>] 4.096kB/4.096kB
e56a8da6e549: Loading layer [====>] 4.608kB/4.608kB
1531a8652a00: Loading layer [====>] 5.632kB/5.632kB
a9c48f850919: Loading layer [====>] 2.56kB/2.56kB
010a75448e2a: Loading layer [====>] 166MB/166MB
dd2115d9a528: Loading layer [====>] 88.1MB/88.1MB
6c7fdb79bda6: Loading layer [====>] 120.8MB/120.8MB
3a12be8140d5: Loading layer [====>] 2.56kB/2.56kB
ff2286d2119e: Loading layer [====>] 106.8MB/106.8MB
3e24218d6f9f: Loading layer [====>] 7.168kB/7.168kB
9a15295246e1: Loading layer [====>] 25.09kB/25.09kB
b7fb68bfc3df: Loading layer [====>] 154.1kB/154.1kB
db07ea3133e8: Loading layer [====>] 4.608kB/4.608kB
6844c16c6e55: Loading layer [====>] 30.72kB/30.72kB
Loaded image: goodfoot/beam:latest
→ docker images | grep 'goodfoot/beam'

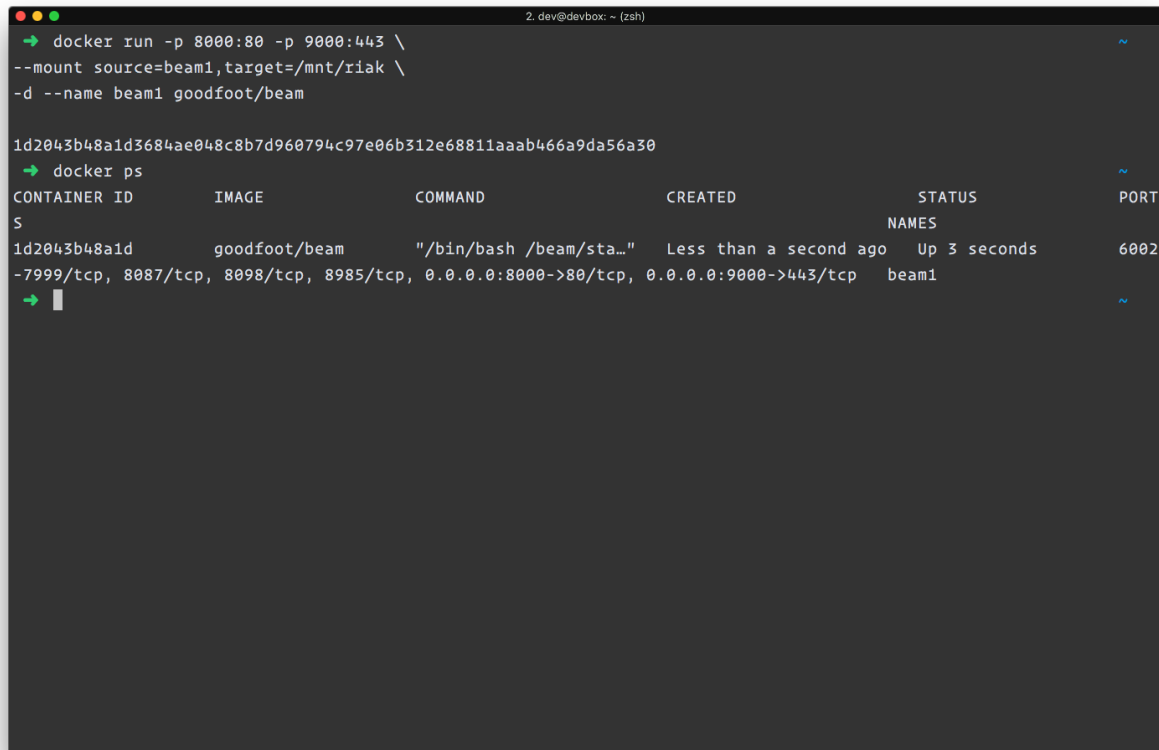
goodfoot/beam      latest                bef43235028d         3 hours ago        1.24GB
→
```

First Beam Node

Each container should have port 80, 443 publicly available so Beam Admin portal can connect to connect it and each beam node should be able to reach others.

Run the following command to start a beam container with port 80, 443 mapped out to host ports.

```
docker run -p 8000:80 -p 9000:443 \
--mount source=beam1,target=/mnt/riak \
-d --name beam1 goodfoot/beam
```



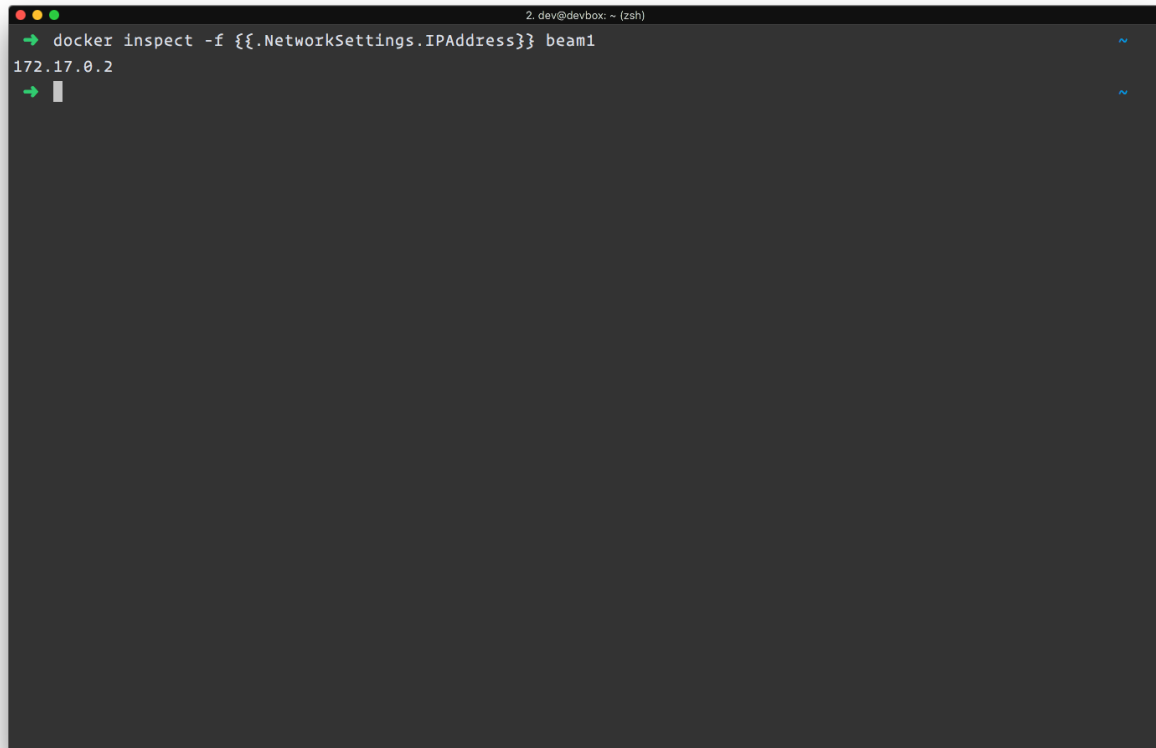
```
2. dev@devbox: ~ (zsh)
→ docker run -p 8000:80 -p 9000:443 \
--mount source=beam1,target=/mnt/riak \
-d --name beam1 goodfoot/beam

1d2043b48a1d3684ae048c8b7d960794c97e06b312e68811aabb466a9da56a30
→ docker ps
CONTAINER ID        IMAGE               COMMAND             CREATED            STATUS              PORTS
1d2043b48a1d        goodfoot/beam      "/bin/bash /beam/sta..." Less than a second ago Up 3 seconds        6002
-7999/tcp, 8087/tcp, 8098/tcp, 8985/tcp, 0.0.0.0:8000->80/tcp, 0.0.0.0:9000->443/tcp   beam1
```

IP Address

Beam containers IP Address can be extracted by running the following command, For example if the name of the first beam node is **beam1** then by running the following command returns the IP Address as **172.17.0.2**.

```
docker inspect -f {{.NetworkSettings.IPAddress}} beam1
```

A terminal window with a dark background and light gray text. The title bar at the top shows three colored window control buttons (red, yellow, green) on the left and the text '2. dev@devbox: ~ (zsh)' on the right. The terminal content shows a green prompt character followed by the command 'docker inspect -f {{.NetworkSettings.IPAddress}} beam1'. The output '172.17.0.2' is displayed on the next line. A second green prompt character is visible on the line below the output, followed by a small white cursor bar.

```
2. dev@devbox: ~ (zsh)  
→ docker inspect -f {{.NetworkSettings.IPAddress}} beam1  
172.17.0.2  
→
```

Additional Beam Nodes

In order to create additional beam nodes, run the following command after substituting **x** with the number of the node ranging from 2 to 5.

HTTP_PORT: Open port number on host machine i.e 8000, 8001, 8002, 8003, 8004

HTTPS_PORT: Open port number on host machine i.e 9000, 9001, 9002, 9003, 9004

```
docker run -p HTTP_PORT:80 -p HTTPS_PORT:443 \  
--mount source=beamx,target=/mnt/riak \  
-d --name beamx goodfoot/beam
```

For example to create four additional beam nodes, the commands are

```
; Second beam node  
docker run -p 8001:80 -p 9001:443 \  
--mount source=beam2,target=/mnt/riak \  
-d --name beam2 goodfoot/beam
```

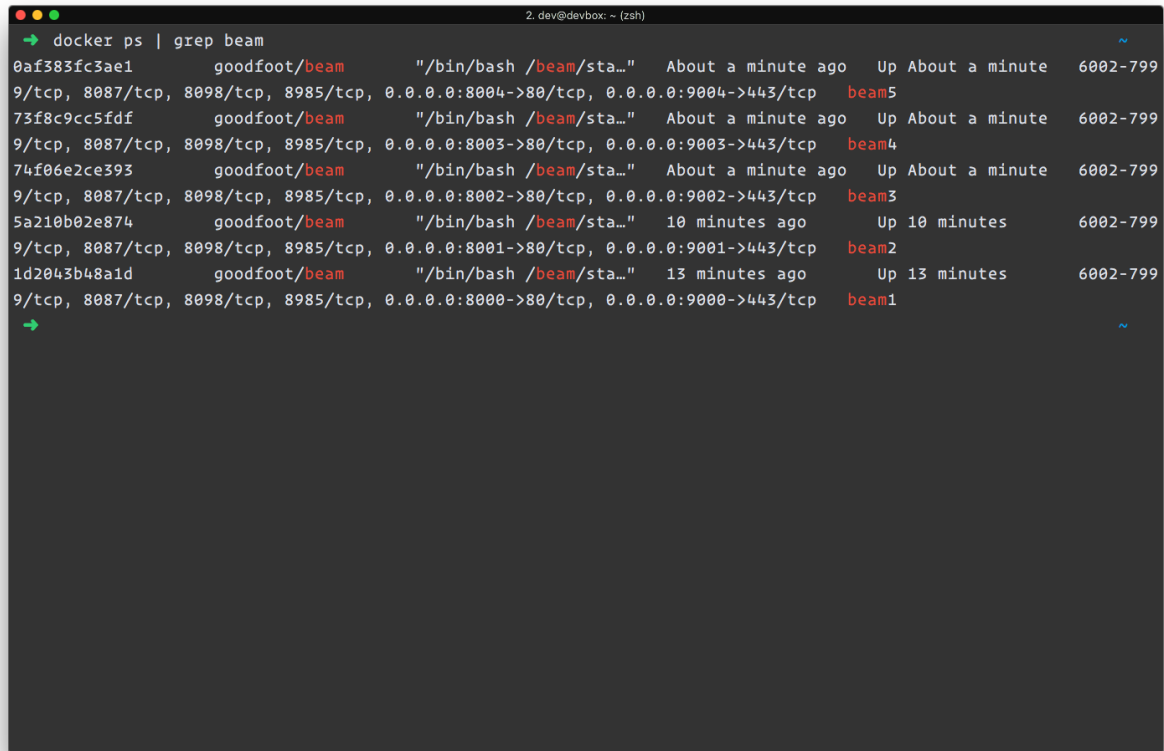
```
; Third beam node  
docker run -p 8002:80 -p 9002:443 \  
--mount source=beam3,target=/mnt/riak \  
-d --name beam3 goodfoot/beam
```

```
; Fourth beam node  
docker run -p 8003:80 -p 9003:443 \  
--mount source=beam4,target=/mnt/riak \  
-d --name beam4 goodfoot/beam
```

```
; Fifth beam node  
docker run -p 8004:80 -p 9004:443 \  
--mount source=beam5,target=/mnt/riak \  
-d --name beam5 goodfoot/beam
```

Check if the all the five nodes are created by running the following command

```
docker ps | grep beam
```



```
2. dev@devbox: ~ (zsh)
→ docker ps | grep beam
0af383fc3ae1      goodfoot/beam    "/bin/bash /beam/sta..." About a minute ago Up About a minute 6002-799
9/tcp, 8087/tcp, 8098/tcp, 8985/tcp, 0.0.0.0:8004->80/tcp, 0.0.0.0:9004->443/tcp beam5
73f8c9cc5fdf      goodfoot/beam    "/bin/bash /beam/sta..." About a minute ago Up About a minute 6002-799
9/tcp, 8087/tcp, 8098/tcp, 8985/tcp, 0.0.0.0:8003->80/tcp, 0.0.0.0:9003->443/tcp beam4
74f06e2ce393      goodfoot/beam    "/bin/bash /beam/sta..." About a minute ago Up About a minute 6002-799
9/tcp, 8087/tcp, 8098/tcp, 8985/tcp, 0.0.0.0:8002->80/tcp, 0.0.0.0:9002->443/tcp beam3
5a210b02e874      goodfoot/beam    "/bin/bash /beam/sta..." 10 minutes ago Up 10 minutes 6002-799
9/tcp, 8087/tcp, 8098/tcp, 8985/tcp, 0.0.0.0:8001->80/tcp, 0.0.0.0:9001->443/tcp beam2
1d2043b48a1d      goodfoot/beam    "/bin/bash /beam/sta..." 13 minutes ago Up 13 minutes 6002-799
9/tcp, 8087/tcp, 8098/tcp, 8985/tcp, 0.0.0.0:8000->80/tcp, 0.0.0.0:9000->443/tcp beam1
→
```

Open browser and load the docker host IP address on port 8000 to open Beam Page.

For example if the docker host IP address is example.com, then <http://example.com:8000> should load Beam page from **beam1** node. Similarly opening <http://example.com:8001> loads Beam page from **beam2** node.

