### 

**World’s Smartest Paperweight  
(Docker)**

Version 1.0

July 25, 2017

**2805 North Dallas Parkway**

**Suite 240**

**Plano, TX 75093**

**(214) 445-3500**

# World’s Smartest Paperweight (Docker)

## Product Overview

The Docker WSP is a stack of raspberry pi nodes that are used to demonstrate Both FDS’s proficiency with Docker and the capabilities of the Docker Platform.

|  |  |
| --- | --- |
| **NODES** | **CONTAINERS** |
|  |  |
| Node 3: Worker  Hostname: pi-worker3  IP Address: <not set to static> | Flair Page \* |
| Node 3: Worker  Hostname: pi-worker3  IP Address: <not set to static> | Flair Page \* |
| Node 3: Worker  Hostname: pi-worker3  IP Address: <not set to static> | Flair Page \* |
| Node 0: Manager  Hostname: pi-manager  IP Address: <not set to static> | Flair Page \*  DWSP UI  HAProxy  Mongo DB  NodeJS  NodeJS |

## Manager Node set up (Pi 3 Model B)

1. Download Raspbian Jessie image and flash image to SD card: [Source](https://www.raspberrypi.org/downloads/raspbian/)
2. Insert SD card with Raspbian image into Pi and boot.
3. Click Raspberry Start button and in preferences make the following changes
   1. Change hostname to “pi-manager” (reboot)
   2. Change Password to dockerpi
   3. Enable SSH and Serial connection
   4. Change Locale – Keyboard – Timezone (reboot)
4. Connect to Wi-Fi
5. Identify IP address or set static IP address
6. Install Docker. Open terminal and execute following commands:

# update and reboot

$ sudo apt-get update

$ sudo apt-get upgrade

$ sudo shutdown –r now

# Install Docker

$ curl –sSL https://get.docker.com | sh

# Add pi user to docker group

$ sudo usermod –aG docker pi

# restart to apply usermod changes

$ sudo shutdown –r now

# As pi user, check that Docker is working with the

$ Docker info

# Docker Installation complete

## Worker Node Set up

1. Flash an image of Raspbian Jessie Lite to an SD (Secure Digital) card
2. Using the raspbi-config interface make the following changes: Hostname, Enable SSH, and Serial Connection

$ sudo raspi-config

* Change hostname: “pi-workerX” where X is the worker number
* Change password: “dockerpi”
* Enable SSH and Serial connection
  + Select “Interfacing options” 🡪 “SSH” 🡪 “yes”
  + Select “Interfacing options” 🡪“Serial” 🡪 “yes”
* Change Locale | Keyboard | Timezone:
  + Select: “Localization Options” 🡪 “Change local” 🡪 “en\_US.UTF-8 UTF-8” 🡪 “C.UTF-8” 🡪 OK
  + Select: “Localization Options” 🡪 “Change Timezone” 🡪 “US” 🡪 “Central”
  + Select: “Localization Options” 🡪 “Change Keyboard Layout 🡪 “Generic 101-key PC” 🡪”Other” 🡪”English (US) 🡪”English (US)” 🡪 “default” 🡪 “No compose key”
* Reboot

1. Connect to Wi-Fi: edit /etc/wpa\_supplicant/wpa\_supplicant.conf by adding the following at the bottom of the file

network={

ssid=”wifi-name”

psk=”password”

}

1. Identify IP address “$ ifconfig”
2. Install Docker with the following commands:

# update and reboot

$ sudo apt-get update -y

$ sudo apt-get upgrade -y

$ sudo shutdown –r now

# Install Docker

$ curl –sSL https://get.docker.com | sh

# Add pi user to docker group

$ sudo usermod –aG docker pi

# restart to apply usermod changes

$ sudo shutdown –r now

# As pi user, check that Docker is working with the

$ Docker info

# Docker Installation complete

## Swarm set up

Create Swarm:

1. On the manager node

$ docker swarm init --advertise-addr <ip address>

# Expected Output format:

Swarm initialized: current node (dxn1zf6l61qsb1josjja83ngz) is now a manager.

To add a worker to this swarm, run the following command:

docker swarm join \

--token SWMTKN-1-49nj1cmql0jkz5s954yi3oex3nedyz0fb0xx14ie39trti4wxv-8vxv8rssmk743ojnwacrr2e7c \

192.168.99.100:2377

To add a manager to this swarm, run 'docker swarm join-token manager' and follow the instructions.

1. To add a worker node copy and paste command from the output of the init command into the terminal of a worker node.
2. View nodes in swarm (on manager node)

$ docker node ls

## Configure, Build, Run, and Use HAProxy Load Balancer

….

………

## Backup a Node: Clone – Resize – Restore

Cloning can be tedious and time consuming (Even more so if you do it incorrectly and have to repeat the process. These instructions are meant to guide one through the process of cloning and image resizing that image and restoring that image on another SD card. Insert the micro SD to be cloned into a Linux computer using the SD card port. After plugging in the card on the desktop you will see the card added as two icons. For each click icon Right click and select “Unmount Volume”.

1. Determine which disk to clone using: sudo fdisk -l
2. Note: it is critical that you identify the correct card: to confirm that you have chosen the correct card check the size and partitions types.
3. After Identifying the correct disk clone image:

$ sudo dd if=</dev/<name of disk> of=~/Desktop/example.img status=progress

1. Eject SD card safely
2. Install pishrink if you do not already have it.

$ wget <https://raw.githubusercontent.com/Drewsif/PiShrink/master/pishrink.sh>

1. Enable write permission for pishrink.sh

$ chmod +x ./pishrink.sh

1. Shrink image

$ sudo ./pishrink.sh ./example.img

1. Insert the SD card you want to restore the image to and unmount the disk by right clicking and selecting “Unmount Volume”.
2. Determine what disk you are going to restore the image using: sudo fdisk –l
3. Restore the image

$ sudo dd if=~/Desktop/example.img of=/dev/<name of disk> status=progress

1. Remount the SD card and from the cmd line navigate to edit the <path to Sd card>/etc/fstab with sudo nano <path to Sd card>/etc/fstab In this file change all occurrences of ‘ro’ to ‘rw’. Save changes.
2. Unmount and eject disk.
3. Put micro SD into test pi and Login pi /raspberry
4. Edit the /etc/fstab file back to how it was originally: change ‘rw’ to ‘ro’. Save changes. Put in ro mode. Safely shut down.

## Shell in a box Configuration

On the router node:

$ sudo apt-chache search shellinabox

$ sudo apt-get install openssl shellinabox

# Edit the config

$ sudo nano /etc/default/shellinabox

# CHANGE PORT

# TCP port that shellinaboxd’s webserver listens on

PORT=6175

# Start shell in a box

$ sudo service shellinabox start

# check that its running

$ sudo netstat –nap | grep shellinabox

## Versions:

* Raspbian Jessie With Desktop 2017-07-05 – Node(s): Manager, Worker,

## Resource Research Reference

Resources:

* [Raspbian Download](https://www.raspberrypi.org/downloads/raspbian/)

Guides / Reference :

* [Getting Started: Prakhar Srivastav](https://prakhar.me/docker-curriculum/)
* [Create a Proxy or Load Balancer](https://docs.docker.com/docker-cloud/apps/load-balance-hello-world/)
* [Getting started with swarm mode](https://docs.docker.com/engine/swarm/swarm-tutorial/)
* [Let Docker Swarm all over you Raspberry Pi Cluster](https://blog.hypriot.com/post/let-docker-swarm-all-over-your-raspberry-pi-cluster/)
* [Docker install walk-through](https://www.youtube.com/watch?v=HsrwrBjkdrU&index=1&list=PLl8N10sn2p33CVisUwEFJEGpJLOlE3OMU&t=328s)
* [HAProxy Documentation](https://cbonte.github.io/haproxy-dconv/1.8/intro.html#2)
* [Digital Ocean: set up HTTP Load Balancing](https://www.digitalocean.com/community/tutorials/how-to-use-haproxy-to-set-up-http-load-balancing-on-an-ubuntu-vps#testing-load-balancing-and-failover)
* [Enable SSH and Serial for Raspbian lite](https://www.youtube.com/watch?v=NoDySHJ23J8)
* [How to set up Wifi on Raspbian Jessie Lite](https://thepihut.com/blogs/raspberry-pi-tutorials/83502916-how-to-setup-wifi-on-raspbian-jessie-lite)
* [How to push a Docker image](https://docs.docker.com/docker-cloud/builds/push-images/)
* [ngOnChange function (Angular 4)](https://angular.io/guide/lifecycle-hooks)
* [web terminal](https://www.npmjs.com/package/web-terminal)
* [Markdown component](https://www.npmjs.com/package/angular2-markdown)
* [What is docker?](https://www.docker.com/what-docker)
* [What are the 5 key benefits of Docker?](https://dzone.com/articles/5-key-benefits-docker-ci)
* [Top Ten Benefits You Will Get By Using Docker](https://apiumhub.com/tech-blog-barcelona/top-benefits-using-docker/)
* [Docker Swarm Mode Walkthrough](https://www.youtube.com/watch?v=KC4Ad1DS8xU&t=176s)

FDS Docker Images:

* brennansaul/flairehello1
* brennansaul/flairhaproxy
* brennansaul/rpi-flairhaproxy

Referenced and Base Docker Images:

* resin/raspberry-pi-alpine-node
* resin/rpi-raspbian
* arm32v7/haproxy

## Document Versions

|  |  |  |
| --- | --- | --- |
| Version | Date | Changes |
| 1.0 | 7/26/17 | Started Documentation |
|  |  |  |
|  |  |  |