



UNIVERSITY OF BRIGHTON

INTERIM PLANNING AND INVESTIGATION REPORT

# Generation of Raspbian images

*Adam Pietrzycki*  
14843569

supervised by  
Dr. Aidan DELANEY

November 29, 2016

## **Abstract**

As of the 8th September 2016, in a blog post by raspberry pi founder Eben Upton [1], mentions that the ten millionth raspberry pi has just been sold. The official operating system for these devices is called Raspbian, it is a port of Debian which is available as a standalone image or in a 'New Out Of the Box Software' package for beginners. 'NOOBS' is pre-installed on SD cards that can be bought from many retailers, before the images can be burned to the cards they need to be somehow generated. The current method of generating these images can be found on GitHub [2], it is a set of shell scripts which from the commits look to be predominantly maintained by a single developer. The problem with this is that if the developer decided to move on and depart from the project it might take some time before someone else understands the code well enough to be able to carry it on; this can be simply described as the 'Bus Factor.' [3] Since the release of the raspberry pi in February 2012, new tools have been developed and standards decided; so it might be a nice idea to freshen up the current method.

# Contents

<b>1</b>	<b>Introduction</b>	<b>2</b>
1.1	Aims . . . . .	2
1.2	Objectives . . . . .	2
1.2.1	Project . . . . .	2
1.2.2	Personal . . . . .	2
<b>2</b>	<b>Background Research</b>	<b>3</b>
2.1	Deployment Management Tools . . . . .	3
2.1.1	Chef . . . . .	3
2.1.2	Puppet . . . . .	3
2.1.3	Salt . . . . .	3
2.1.4	Ansible . . . . .	3
2.2	Virtualization . . . . .	3
2.2.1	ESXi . . . . .	3
2.2.2	VirtualBox . . . . .	3
2.2.3	Docker . . . . .	4
2.2.4	Vagrant . . . . .	4
2.2.5	QEMU . . . . .	4
2.3	Current generation method . . . . .	4
2.3.1	Analysis . . . . .	4
2.3.2	Potential improvements . . . . .	4
<b>3</b>	<b>Project Planning</b>	<b>5</b>
3.1	Methodologies . . . . .	5
3.2	Stakeholders . . . . .	5
3.3	Project schedule . . . . .	5
	<b>Bibliography</b>	<b>5</b>

# Chapter 1

## Introduction

### 1.1 Aims

The overall aim of the project is to show that the current code base which generates the Raspbian images can be re-written and re-designed to make use of open source tools and a 'human friendly data serialization standard,' [4] called YAML.

### 1.2 Objectives

The objectives can be separated into two different categories; project and personal. These will provide an image of the project structure and will help towards the success of completion.

#### 1.2.1 Project

- Write code that utilizes open source tools and language standards.
- Generate Raspbian images at a faster speed.
- Give options to users for easy image customization.
- Get feedback from the Raspberry Pi foundation.

#### 1.2.2 Personal

- Learn shell scripting.
- Learn a provision management tool.
- Understand Linux on a more complex level.
- Expand knowledge of virtualization.

# Chapter 2

## Background Research

### 2.1 Deployment Management Tools

There are many deployment management tools to choose from, all offer different benefits and are backed by various companies. A comparison of tools has been covered in detail at Openstack's Summit [5] of which are referenced in the sections below.

#### 2.1.1 Chef

Chef [6]

#### 2.1.2 Puppet

Puppet [7]

#### 2.1.3 Salt

Salt [8]

#### 2.1.4 Ansible

Ansible [9]

### 2.2 Virtualization

2 Quick description of virtualization

#### 2.2.1 ESXi

ESXi [10]

#### 2.2.2 VirtualBox

VirtualBox [11]

### **2.2.3 Docker**

Docker [12]

### **2.2.4 Vagrant**

Vagrant [13]

### **2.2.5 QEMU**

QEMU [14]

## **2.3 Current generation method**

Short description

### **2.3.1 Analysis**

Show research

### **2.3.2 Potential improvements**

Show research

# Chapter 3

## Project Planning

Standards and Quality check

### **3.1 Methodologies**

Short comparison of methodologies

### **3.2 Stakeholders**

List stakeholders and their interests

### **3.3 Project schedule**

Add a graph for stages

# Bibliography

- [1] Raspberry Pi. (2016). Ten millionth Raspberry Pi, and a new kit.  
<https://www.raspberrypi.org/blog/ten-millionth-raspberry-pi-new-kit/>
- [2] Raspberry Pi. (2016). RPi-Distro/pi-gen.  
<https://github.com/RPi-Distro/pi-gen>
- [3] Wikipedia. (2016). Bus Factor.  
[https://en.wikipedia.org/wiki/Bus\\_factor](https://en.wikipedia.org/wiki/Bus_factor)
- [4] YAML. (2009). %YAML 1.2.  
<http://yaml.org/>
- [5] Openstack. (2015). Chef vs. Puppet vs. Ansible vs. Salt - What's Best for Deploying and Managing OpenStack?.  
<https://www.openstack.org/summit/tokyo-2015/videos/presentation/chef-vs-puppet-vs-ansible-vs-salt-whats-best-for-deploying-and-managing-openstack>
- [6] Chef. (2016). Chef – Automate Your Infrastructure.  
<https://www.chef.io/chef/>
- [7] Puppet. (2016). Puppet - The shortest path to better software.  
<https://puppet.com/>
- [8] Glauser, R. (2016). SaltStack automation for CloudOps, ITOps & DevOps at scale.  
<https://saltstack.com/>
- [9] Red Hat. (2016). Ansible is Simple IT Automation.  
<https://www.ansible.com/>
- [10] VMWare. (2016). vSphere ESXi Bare-Metal Hypervisor.  
<http://www.vmware.com/products/esxi-and-esx.html>
- [11] Oracle. (2016). Oracle VM VirtualBox.  
<https://www.virtualbox.org/>
- [12] Docker. (2016). Docker.  
<https://www.docker.com/>
- [13] HashiCorp. (2016). Vagrant by HashiCorp.  
<https://www.vagrantup.com/>



- [14] QEMU. (2016). QEMU - Open source processor emulator.  
[http://wiki.qemu.org/Main\\_Page](http://wiki.qemu.org/Main_Page)