Fleet Management

A Brief Introduction

Raspberry Pi Ltd

2022-07-19: githash: 1f1902f-clean

Colophon

build-date: 2022-07-19
build-version: githash: 1f1902f-clean

Legal disclaimer notice

TECHNICAL AND RELIABILITY DATA FOR RASPBERRY PI PRODUCTS (INCLUDING DATASHEETS) AS MODIFIED FROM TIME TO TIME ("RESOURCES") ARE PROVIDED BY RASPBERRY PI LTD ("RPL") "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. TO THE MAXIMUM EXTENT PERMITTED BY APPLICABLE LAW IN NO EVENT SHALL RPL BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THE RESOURCES, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

RPL reserves the right to make any enhancements, improvements, corrections or any other modifications to the RESOURCES or any products described in them at any time and without further notice.

The RESOURCES are intended for skilled users with suitable levels of design knowledge. Users are solely responsible for their selection and use of the RESOURCES and any application of the products described in them. User agrees to indemnify and hold RPL harmless against all liabilities, costs, damages or other losses arising out of their use of the RESOURCES.

RPL grants users permission to use the RESOURCES solely in conjunction with the Raspberry Pi products. All other use of the RESOURCES is prohibited. No licence is granted to any other RPL or other third party intellectual property right.

HIGH RISK ACTIVITIES. Raspberry Pi products are not designed, manufactured or intended for use in hazardous environments requiring fail safe performance, such as in the operation of nuclear facilities, aircraft navigation or communication systems, air traffic control, weapons systems or safety-critical applications (including life support systems and other medical devices), in which the failure of the products could lead directly to death, personal injury or severe physical or environmental damage ("High Risk Activities"). RPL specifically disclaims any express or implied warranty of fitness for High Risk Activities and accepts no liability for use or inclusions of Raspberry Pi products in High Risk Activities.

Raspberry Pi products are provided subject to RPL's Standard Terms. RPL's provision of the RESOURCES does not expand or otherwise modify RPL's Standard Terms including but not limited to the disclaimers and warranties expressed in them.

Document version history

Release	Date	Description
1.0	1 Jan 2022	Initial release

Scope of document

This document applies to the following Raspberry Pi products:

Pi 0			Pi 1		Pi 2		Pi 3	Pi 4	Pi 400	CM1	CM3	CM4	Pico
0	W	H	A	B	A	B	B	All	All	All	All	All	All
*	*	*	*	*	*	*	*	*	*	*	*	*	

Introduction

As Raspberry Pi devices become more commonplace as part of a company's portfolio of devices, there becomes a need to manage these devices.

Raspberry Pi devices can cover a huge range of requirements in a company's network. For example, you may use the Raspberry Pi 400 as a desktop device, or a thin client; you may have Raspberry Pi 4 devices running kiosk displays; you may have other models attached to production line machinery to monitor health; you may even have devices running machinery directly. The sheer range of applications that the Raspberry Pi devices are suitable for means any one company may have hundreds, thousands, or even tens of thousands of devices in their portfolio. All these devices need to be maintained and, as numbers increase, manual checking becomes extremely inefficient. This is where fleet management software comes into play.

This management may encompass a number of different areas, some of which are:

- ensuring the right software is installed
- ensuring security patches are present
- monitoring device health

Raspberry Pi do not provide any sort of fleet management software, this is an area where there is a plethora of existing suppliers. This whitepaper attempts to show what these suppliers can do, and also provides a brief introduction to some that Raspberry Pi know work with Raspberry Pi devices. Please note that these are NOT recommendations; Raspberry Pi have not done any deep investigations into any of these products, simply ensured that Raspberry Pi devices are supported. It is down to the individual to ensure that any product or service matches their use case. Please also note that any suppliers listed are not the only suppliers of services that may be appropriate, i.e. the list is not exhaustive.

It is always worth remembering that at its core, the Raspberry Pi can be very simply regarded as "just another computer". It runs Linux and has an Arm processor, instead of Windows on x86 like many desktops, but those are simply details rather than any show-stopping difference. Getting into the mindset of it just being another device to manage, rather than specifically a Raspberry Pi, will reap great rewards.

Basic concepts

At the root of all fleet management systems is a requirement for access to the devices. When talking only about system or application updates, this is done in two ways, known as *push* or *pull*.

Updates via push or pull

Pull-based systems require a piece of software called an agent to be running on the device. At a specified time interval, this agent will communicate with a server to determine if anything needs to be done, and will *pull* data from that server to accomplish the requirements.

In a push-based system, the server communicates directly with the device and *pushes* instructions to it. No agent is required on the remote device.

Each of these techniques has advantages and disadvantages. The following is a list of some of the pros and cons.

Technique	Advantages	Disadvantage
Push	No agent required Server entirely in control of any update processes and timings Push can be done from any server	Server requires a list of the addresses of all devices to be managed Devices must be running a constantly open connection system such as OpenSSL, which may be a security risk Depending on the level of sophistication, all devices may get all data, which may be unnecessary
Pull	Devices are in control of their own destiny Devices only pull down what they need No standard ports open on the device (e.g. OpenSSL) Agent initiates a secure link with the server (e.g. a virtual private network) only when needed	Requires an agent running on the Raspberry Pi (Arm Linux) Requires a *server of record* that the agent communicates with

Different situations will require different solutions. It is up to the customer to determine the best procedures for their particular use case.

Monitoring

Monitoring of device health (e.g. disk usage, CPU usage) is possible in both push and pull systems.

In a push system, the server may connect to the remote device directly, via OpenSSL or similar, and run specific commands on the remote device to recover monitoring information.

In a pull system, the agent will collate monitoring information, and during its timed communications with the server may pass that information back.

In both cases, similar types of information are available. It is likely that push systems will have a faster response time for standard monitoring, but providing pull agents are run frequently (some systems check every 5 seconds, some every minute, some maybe once every 24 hours) this should not cause problems. Polling times are usually a user-modifiable parameter.

How to choose a management system

There are many systems available, ranging from free open-source systems where you need to set up servers in your internal network, to enterprise-level paid-for services that can run either in your own network or via a cloud service. In some case the enterprise services are based on open-source offerings, where the complexities of setup are delegated to the service provider.

Of course, if you as the customer are already running a management system, finding out whether that system supports Raspberry Pi devices would be the first thing to check. If they don't support Raspberry Pi devices, it's worth asking whether it is being considered! Some service providers may support Linux agents, but perhaps for Ubuntu running on the x86 platform, rather than the Raspberry Pi operating system (OS) on Arm. Usually recompiling existing agents for Raspberry Pi OS, either 32 or 64 bit, is a simple task. In some cases, agents are already available precompiled in the Raspberry Pi OS software repositories.

You can search the Raspberry Pi OS repositories using the `apt-cache` command. For example, the following command searches for the Puppet system in the repos. This command also works in other OSs such as Ubuntu.

```
apt-cache search puppet
```

This will result in a long list of Puppet packages that can be installed.

Know your requirements

When choosing a management system, it is very important to first have a list of the requirements. There are many questions that need to be asked, and Raspberry Pi cannot predict all that may be important to your organisation. Some you may wish to consider are:

- Do I already have an existing system in place?
- How many devices do I need to support?
- Is it only Raspberry Pi devices, or do I also need to support Windows, other Internet of Things (IoT) devices, Linux, etc.?
- Push or pull?
    - Do I want an agent running?
    - Am I happy with the security of a push system?
- Do I need monitoring?
- What security levels do I need?
- How much work do I want to do myself?
    - Purely open source, run your own server.
    - Buy in services.

Getting your requirements right at an early stage will prevent many problems later.

Some providers in the field

Firstly, these are not recommendations, only examples of systems that work with Raspberry Pi devices. This list is also far from exhaustive, only giving a few examples of the many companies and systems that exist.

Salt/Saltstack

> What is Salt? It's automation, infrastructure management, it's data-driven orchestration, remote execution, configuration management, and so much more.
>
> — Salt/SaltStack

An open-source or enterprise push system, Salt runs on Raspberry Pi with packages available in the repositories.

It uses a central repository to provision new servers and other information technology (IT) infrastructure, to make changes to existing ones, and to install software in IT environments, including physical and virtual servers, as well as the cloud. Salt is a command line tool, but highly featured, and as a consequence does require a high level of knowledge to run it successfully. It uses the SSH protocol for access security.

https://saltproject.io/

The Salt project has a specific page for working with Raspberry Pi devices.

JFrog

> Update, control, monitor and secure remote Linux & IoT devices, at scale, with the click of a button.
>
> — JFrog Connect

JFrog is an easy to install and use pull system, with a cloud-based interface, providing updates, system monitoring, alerts, remote logging, and other features.

https://jfrog.com/connect/

In testing it was very easy to add new Raspberry Pi devices to the JFrog dashboard. It was quick and easy to monitor the device, for example for disk space, CPU usage, etc. The ability to run terminals via an encrypted link to specific devices could be very useful. Email alerts are available when devices start, stop, or exceed particular parameters such as CPU use.

Progress Chef

> Automation Software for Continuous Delivery of Secure Applications and Infrastructure.
>
> — Progress Chef

A usually paid-for system, Progress Chef also has community support for the Raspberry Pi.

https://www.chef.io/

CFEngine

> Whether your IT infrastructure consists of private or public cloud servers, desktops or IoT devices, CFEngine provides powerful automation and unparalleled visibility to keep infrastructure secure & compliant.
>
> — CFEngine

CFEngine is available as open source (Community) or commercial (Enterprise) editions. Packages are available in the Raspberry Pi OS repositories.

The Community edition is free to use and licensed under the GNU General Public License (GPL), while Enterprise adds a comprehensive set of features and service-level agreement support under a commercial licence. See https://cfengine.com/community-vs-enterprise-comparison/ for the differences.

https://cfengine.com/

In many ways similar to Salt, CFEngine requires a good level of knowledge to implement in an effective way.

Puppet

> Puppet is the industry standard for IT automation.
>
> Modernize, manage and bring your hybrid infrastructure into compliance through Puppet's powerful continuous automation.
>
> — Puppet

Puppet is an open-source pull project, with many packages already in the Raspberry Pi OS repositories. It comes in community and enterprise editions. The community edition is open source, while the Enterprise edition (a paid-for option) provides more extensive features.

Puppet is scalable to tens, if not hundreds, of thousands of devices. It uses the concept of "desired state", checking a primary server at 30 minute intervals to ensure that devices are "in state", i.e. that they are in the state that they need to be, for example with correct software install version numbers, security fixes, etc. The server returns to the device a catalogue of information of what the desired state is, and the Puppet agent will then ensure that that state is reached.

NOTE

The Puppet website does not say that Raspberry Pi devices are supported. However, Debian Bullseye, on which the current Raspberry Pi OS is based, is supported, and the required agents are in the Raspberry Pi OS repositories already compiled for `aarch64` (the architecture used in the 64-bit Raspberry Pi OS).

https://puppet.com/

Ansible

> Red Hat® Ansible® Automation Platform is a foundation for building and operating automation across an organization. The platform includes all the tools needed to implement enterprise-wide automation.
>
> — Ansible

Ansible is a push automation system developed at Red Hat. Ansible applications are available in the Raspberry Pi OS repositories. Available in a Community edition (free) or from Red Hat as a supported paid-for system.

Ansible works by connecting to your nodes and pushing out small programs, called Ansible modules, to them. These programs are written to be resource models of the desired state of the system. Ansible then executes these modules (over SSH by default), and removes them when finished.

Your library of modules can reside on any machine, and there are no servers, daemons, or databases required. Typically you'll work with your favourite terminal program, a text editor, and probably a version control system to keep track of changes to your content.

https://www.ansible.com/

balenaCloud

> The container-based platform for deploying IoT fleets
>
> Comprehensive device deployment and management infrastructure, hosted by balena.
>
> — balenaCloud

Although it does have support for Raspberry Pi devices, you are limited to using the balena OS instead of an agent. This is an OS optimised to run Docker containers on embedded devices.

https://www.balena.io/cloud

Raspberry Pi