Ansible Automation Guidelines

What is Ansible?

Ansible is an open source automation tool used to configure servers, install software and perform a wide variety of IT tasks automation from a central location.

Why Ansible?

Given the actual structural configuration of machines and their diversity, a central point of communication was needed in order to keep track of their health and usage, as well to automate the collection of data to do so. Ansible was chosen because it allows a connection to servers over the SSH protocol, which is the current main method of connection for all the servers, which makes it a “non-invasive” solution without the need of any software installation in production servers.

What is a playbook?

A playbook in Ansible’s environment represents a repeatable configuration management YAML file, that executes a task, several taks or multiple playbooks at once when invoked. It can be used to push out new configuration or confirm the configuration of remote systems.

Installation and Configuration

Ansible’s official documentation can be found here.

Automation of Information Gathering

Accessing manually every server to gather data is time consuming and such data must be updated frequently to be useful. The ability to collect and analyse this information fast, in a short spaced schedule, is vital to keep any server's infrastructure running and locate / forecast issues. This is critical when dealing with our devices, which are severely limited by memory / storage / energy consumption most of the time.

To solve that, 2 data collecting scripts where created:

- mem_usage.sh
- system_info.sh

The first collects basic usage data, i.e. CPU, memory and storage, while the second collects more in-depth information, i.e. uptime, available updates, kernel version, etc.

Given that these scripts only run locally, in order to automate the process of copying them out to every server and copy their output back, some Ansible playbooks were created as follows:

- add_ssh_keys.yaml

  This playbook copy my SSH public key, for accessing the servers without typing passwords, which increases the security level.

- system_info.yml

  This playbook makes a copy of the script *system_info.sh* to the servers, execute it and send back the output to a file inside the invoker machine (device where the running command was executed). Then, the data collected is merged and written into a report called **sytem_report**.

- mem_usage.yml

  This one makes exactly the same as the previous playbook, but it generates another report named **usage_report.csv** which can be converted into an Excel spreadsheet.

- update.yml

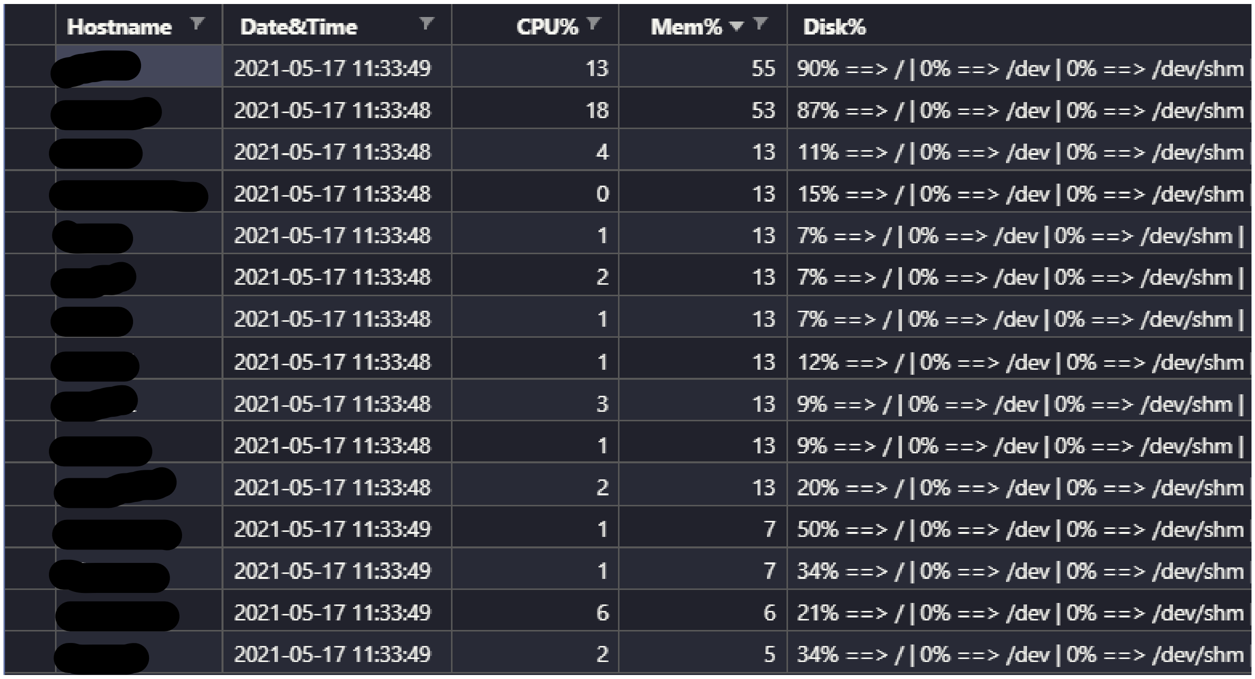
  This one updates the system's package repository and, after, updates the system with safe upgrades, i.e. security patches.

- get_general_info_and_update.yml

  This playbook just triggers all the *previous* playbooks at once.

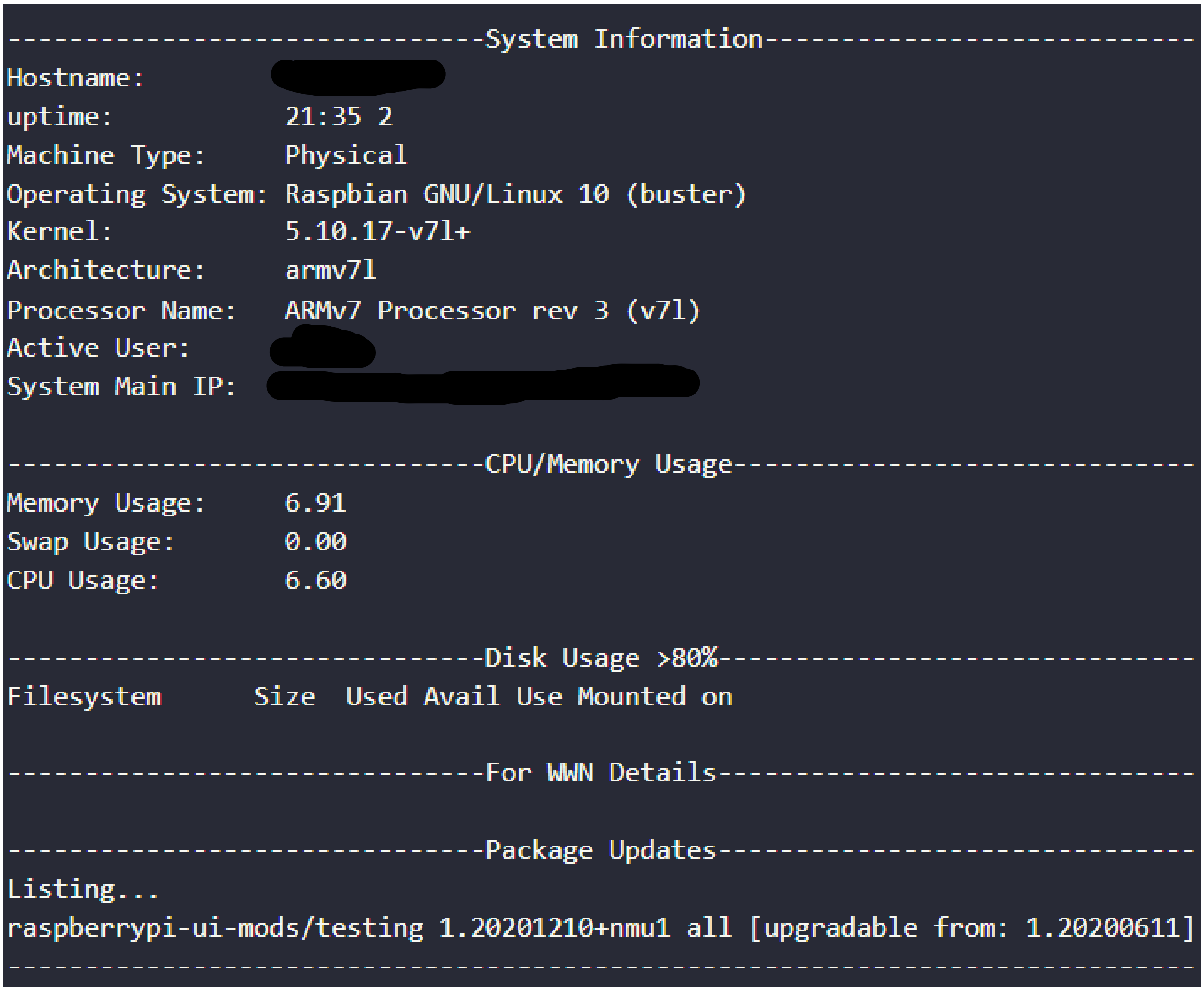
Reports

As above mentioned, there are 2 health reports generated, being a *csv* and a *text* files.



	Hostname	Date&Time	CPU%	Mem%	Disk%
	[redacted]	2021-05-17 11:33:49	13	55	90% ==> / | 0% ==> /dev | 0% ==> /dev/shm
	[redacted]	2021-05-17 11:33:48	18	53	87% ==> / | 0% ==> /dev | 0% ==> /dev/shm
	[redacted]	2021-05-17 11:33:48	4	13	11% ==> / | 0% ==> /dev | 0% ==> /dev/shm
	[redacted]	2021-05-17 11:33:48	0	13	15% ==> / | 0% ==> /dev | 0% ==> /dev/shm
	[redacted]	2021-05-17 11:33:48	1	13	7% ==> / | 0% ==> /dev | 0% ==> /dev/shm |
	[redacted]	2021-05-17 11:33:48	2	13	7% ==> / | 0% ==> /dev | 0% ==> /dev/shm |
	[redacted]	2021-05-17 11:33:48	1	13	7% ==> / | 0% ==> /dev | 0% ==> /dev/shm |
	[redacted]	2021-05-17 11:33:48	1	13	12% ==> / | 0% ==> /dev | 0% ==> /dev/shm
	[redacted]	2021-05-17 11:33:48	3	13	9% ==> / | 0% ==> /dev | 0% ==> /dev/shm |
	[redacted]	2021-05-17 11:33:48	1	13	9% ==> / | 0% ==> /dev | 0% ==> /dev/shm |
	[redacted]	2021-05-17 11:33:48	2	13	20% ==> / | 0% ==> /dev | 0% ==> /dev/shm
	[redacted]	2021-05-17 11:33:49	1	7	50% ==> / | 0% ==> /dev | 0% ==> /dev/shm
	[redacted]	2021-05-17 11:33:49	1	7	34% ==> / | 0% ==> /dev | 0% ==> /dev/shm
	[redacted]	2021-05-17 11:33:49	6	6	21% ==> / | 0% ==> /dev | 0% ==> /dev/shm
	[redacted]	2021-05-17 11:33:49	2	5	34% ==> / | 0% ==> /dev | 0% ==> /dev/shm

Fig. 1 - usage_report.csv



```
-------------------------------System Information----------------------------
Hostname:           [redacted]
uptime:             21:35 2
Machine Type:       Physical
Operating System:   Raspbian GNU/Linux 10 (buster)
Kernel:             5.10.17-v7l+
Architecture:       armv7l
Processor Name:     ARMv7 Processor rev 3 (v7l)
Active User:        [redacted]
System Main IP:     [redacted]

-------------------------------CPU/Memory Usage------------------------------
Memory Usage:       6.91
Swap Usage:         0.00
CPU Usage:          6.60

-------------------------------Disk Usage >80%-------------------------------
Filesystem      Size  Used Avail Use Mounted on

-------------------------------For WWN Details-------------------------------

-------------------------------Package Updates-------------------------------
Listing...
raspberrypi-ui-mods/testing 1.20201210+nmu1 all [upgradable from: 1.20200611]
-----------------------------------------------------------------------------
```

Fig. 2 - system_report

Playbooks Execution

Any previously mentioned playbook can be executed with the command, assuming that *your ssh public key* is added to the remote servers:

ansible-playbook -i /path/to/playbook_name.yml