

NEMS, or Nagios Enterprise Monitoring System, is a small linux distro developed for the Raspberry Pi. Its purpose is to provide a relatively turnkey, easy to implement, and flexible, method of providing monitoring services. As a distribution, it can be flashed to an SD card as an all-in-one package. As the NEMS website explains, "It has all the bells and whistles while being optimized for solid stability and fast performance."

But going further than just the software and distro, complete hardware packages exist as well (<https://amzn.to/2Z7DJHS>) to provide everything an administrator or engineer could need to get started. So why do most systems administrators install Nagios on a dedicated VM or larger server instead of using a single board computer? The project's website explains:

Nagios doesn't need a big fancy supercomputer to offer exceptional enterprise monitoring of network assets and resources, so our temptation is to re-purpose older servers to perform this reasonably lightweight task.

If you ask me, that's not only overkill, but the attempt to save money by reusing older hardware will actually cost more due to higher electricity usage vs. say, a tiny Raspberry Pi 3 Microcomputer... which ironically may in fact have more modern system specifications than that old beast of a server you've been using.

It can be a wee bit daunting to setup a Raspberry Pi Nagios server from scratch, and there were no projects I found which were actively maintained at a level adequate for professional use. So I decided to start a new project—called NEMS: Nagios Enterprise Monitoring Server.

This project uses freely available applications such as Debian Linux, Nagios Core and a variety of other goodies, and I too release a fully ready-to-use image for you to use within yours or your customers' network environments.



These are good points. While the tendency is to think of a larger server as being more capable and more reliable than a single board computer, the truth is that "larger and more powerful" might actually mean more frail, complex, and wasteful. Quite simply, not much exists on a Raspberry Pi3 that could fail, and the system is extremely efficient

due to its target audience. There are no spinning disks by default, all cooling is passive, power supplies are modular, memory is integrated, and everything is packaged into a commodity piece of hardware. Because Nagios does little more than send and receive small health checks through a set of open ports, requirements are much lower than that of a web-facing application server or more resource-hungry solution. If the Pi 3 ever does fail, another complete hardware board can either be acquired overnight,

or several boards can be kept on standby. Swapping the SD card to another board is usually all that is necessary to get back up and running.

So what does it take to get an instance of this software started, and what does the setup procedure look like?

To start, the requirements of NEMS Linux are very low. The only real catch to this distro's installation is that it can only run on single board computers with enough RAM for all of the required processes. The Raspberry Pi 2 and lower do not have the capabilities necessary to provide a satisfactory experience and might not be stable enough for deployment. If the target system has at least 1 gigabyte of memory, that should be sufficient to run the distro.

The first and most major step is to flash an SD card with the distribution. This download can be retrieved at nemslinux.com, with compatible images also available for ODROID XU4, RasPi, Asus Tinkerboard, and other single board computers. A larger SD card is recommended, and 32GB would be an absolute minimum for a project like this. My system uses a 128GB card, with all logs being sent to a local ELK stack to ease storage requirements.

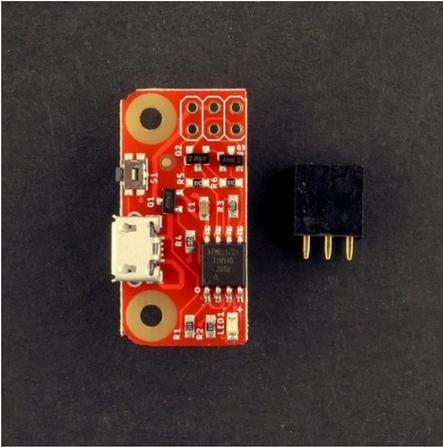


Once the OS is flashed and the board is booted, the admin will then navigate to `nems.local` in the board's browser. That's it. Configuration is easily done to the base server by navigating to the NEMS systems settings tool through the web UI. There, you can set basic details, such as the server name, background image, and Windows domain access credentials.

Under the reporting tab, NEMS includes Adagios, a modern Nagios dashboard with details about monitored systems and services. Also included are the NEMS mobile UI for cell phones and other small screens, and the NEMS TV dashboard, designed for larger-format screens. Each of these interfaces allows for a universally-accessible eagle-eye view of an infrastructure with little fuss in terms of access capabilities or configuration out of the box.

NEMS Linux also includes services such as cloud backup for configuration. sign-up can be done through the configuration menu as well, and once an API key is retrieved from the project's site, it can be entered there to enable those aspects of the distro.

Configuring health checks is similar to other Nagios or Icinga2 plugin configuration, and most host group information and contacts can even be ported from other Nagios installs. All config files are placed under `/etc/nems` on the filesystem. Plugins and system checks on most Debian and RedHat platforms will work just fine with a slight tweak of filesystem locations.



Stability is very good, but can be improved further by coupling a NEMS Linux/RasPi 3 board with a PiWatcher: <https://shop.omzlo.com/collections/other/products/piwatcher-the-best-watchdog-for-your-raspberry-pi>

This is a great little device that automatically reboots the Raspberry Pi 3 if it stops responding to pings for a preconfigured amount of time. That way, if the board crashes for any reason, automatic recovery will occur without an administrator intervening, or noticing that the interfaces and dashboards have become unavailable.

In all, setup and configuration is a breeze. If a small server room host monitoring solution is needed in an environment, consider setting up a single board computer with a distro like NEMS. From initial start to completion, the project can take as little as a few hours to have a highly-flexible device, capable of watching over a flock of servers. The configuration can be backed up on-site or to the cloud, and the system's intuitive UI makes pinpointing and troubleshooting issues easy.