Community Feature! Robbie Ferguson of Category5 TV

Posted by Rose Todaro on April 4, 2018

Today we're featuring Robbie Ferguson, a Nagios community member, the winner of our case study contest and the founder of Category5 TV. We loved hearing about his creative Nagios projects, as well as his use of Nagios to help monitor the equipment needed to run the show.

I had the chance to ask him some questions about what it's like to start your own TV network, how he uses Nagios to help manage technological challenges and what other projects he has in the works.

Head over to Category5.tv to check out their roster of programming, including this great Raspberry Pi + Nagios tutorial. Additionally, if you have any stories of how you use Nagios, for work or for play, we'd love to hear about them. Email casestudies@nagios.com for a chance to have your story featured on our website.

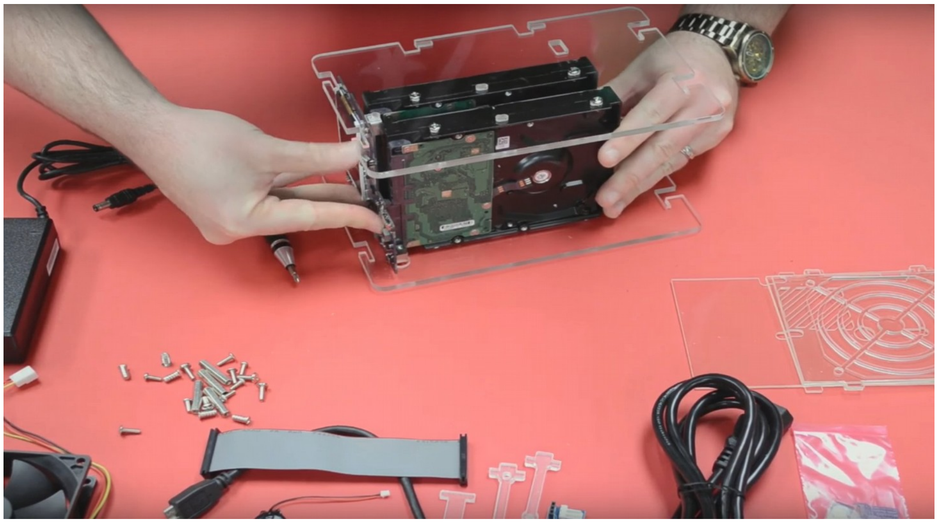
Can you tell me a bit about Category5 TV? How did you come to host a webcast technology tv show?

I founded Cat5 as a way to provide free tech support to users. I owned a computer

consulting business at that time and found myself spending a lot of time providing free support. People don't generally expect to be invoiced for calling you up and asking a few questions like "what should I look for in a printer purchase" – so I created Category5 Technology TV as a 2-fold solution: 1) it allowed me to focus my weekday efforts on generating revenue with my business and 2) it ensured those customers still could receive free support (as I genuinely care about taking care of even those who don't want to pay me for my effort).
Category5 Technology TV has grown into a popular weekly broadcast seen all around the world via smart TVs, Roku, Kodi, Plex, and even Cable TV in Southern California (and hopefully more areas in future).

What have been some of the challenges of running an independent TV station?

Here's a genius idea: create a thing that costs a lot of money … but don't charge anyone for it! Our biggest challenge has been to creatively build revenue streams that maintain the show at a level of quality that the viewers expect. We have a small studio (about 900 sq ft) that we rent, monthly service bills, plus the hardware expense of doing what we do. Part of this challenge means trying to keep our costs as low as we can. Nagios helps us do that not only on the bottom line, but also in proactively being able to monitor our assets and attend to issues before they become catastrophic.



Assembling an ODROID XU4 CloudShell 2 NAS Unit

How did you first come around to setting up and using Nagios?

It's really hard to know when the "first" setup occurred. I've used Nagios for many years for client sites, but my personal usage in our studio environment (which birthed NEMS Linux) began with two occurrences:

1) Our internet service at our studio is LTE-based and the provider blocks our ability to use remote control software (eg., VNC, RDP, TeamViewer). We needed to know that our studio was operational even when nobody was there since video rendering tasks can sometimes take days. At one point a Blender render was running that took nearly a week to process, and a Windows update decided to reboot the computer that was rendering. Due to timing, we didn't know this until a few days later, and had to restart the entire render process.

2) I'd had a hard drive crash in our main storage rig, and like so many other small operations that operate on a small budget, I wasn't keeping a close eye. There were no alerts from the server itself. By the time I'd finally logged into the interface it could have been weeks or even months. No problem: due to parity, we had single drive fault tolerance. But then during the rebuild to a replacement drive, the parity drive failed. We were very fortunate to keep good backups, so very little (if anything) was lost, but I realized had Nagios been in place, I would have known about the drive failure — possibly even before the drive failure.

I knew Nagios could be used to help notify me of these types of issues, so I built my own Nagios server on a Raspberry Pi (again, keeping to a very small budget).

A Nagios + Raspberry Pi Tutorial from Category5 TV

What are some of the ways Nagios has changed your work processes?

Response time to potential issues is very good thanks to Nagios. Receiving a notification when, for example, a drive is 80% capacity so I can swap it out with something larger before it hits that catastrophic 100% is why our uptime is so excellent. We operate with a very finite budget, and Nagios genuinely allows us to be constantly monitoring assets without actually, you know… constantly monitoring assets (which would cost a lot in either wages or time). I use the capacity example because it really happened to us. One of our broadcast systems which is crucial to our operation uses extremely fast, but extremely small capacity drives. We had it happen where one of the system drives wasn't cleaning up transient files and hit capacity, and the only alerts we had in place were from Uptime Robot. While Uptime Robot is in its own right a great service, it is reactive, not proactive like Nagios. That means, when our server went down we got alerts. But not before. We had no idea it was coming and were ill-prepared. It meant significant downtime and required repairing the filesystem and resulted in the loss of some logs. Since implementing Nagios in our network, we've never since had such an issue as the alerts always notify us well in advance of anything hitting hard physical limits.

What has your experience been tinkering with Nagios and Raspberry Pi? Have you implemented any cool projects using Raspberry Pi and Nagios together?

Oh yeah. My goal with NEMS Linux (Nagios on a Raspberry Pi) has been not just to install Debian and Nagios on a Pi and call it a day, no sir! Nagios scripting is onerous and complicated for novice sysadmins. So NEMS Linux brings together several open source projects to provide an almost entirely browser-based GUI experience. NConf was a great project in its day, but it died. NConf being a browser-based GUI to replace the need to edit Nagios script files manually. So being a dead project, I ported its GitHub page, made it so it works on modern systems (eg., PHP 7, MariaDB with MySQLi) and made that a core component of NEMS. Now, Nagios can be deployed, configured and monitored from within a web browser session. I've personally written an off-site backup system for it that makes it a couple clicks to restore a NEMS deployment to previous settings, and the codebase for NEMS Linux itself continues to grow. Also, with the growing popularity of NEMS Linux, I'm also porting it to more powerful devices suited to businesses. I'm nearing the completion of initial builds for both the CL100 by Logic Supply and Compulab's Fitlet2 which will each provide a strong network monitoring solution–powered by Nagios Core–to the budget-minded network admin.

Source: https://www.nagios.com/news/2018/04/community-feature-robbie-ferguson-of-category5-tv/