The Bond Between NodeJS and DevOps is now Getting Stronger

Have you noticed the surprising interlinking of **NodeJS** with **DevOps**? Have you found that NodeJS is at the core of most of the DevOps tools? We know that DevOps is bringing together the Developers and the Operations team. We know that this is shortening the entire process cycle from the development to the launch phase. But do you know why among all the available tools and platforms DevOps, the automation tools in particular, rely on NodeJS? Be it a normal monitoring tool or a deployment tool again, NodeJS is preferred. We also have news that NodeJS is going to get more DevOps friendly.

NodeJS is getting better day after day:

With each new release, each new upgrade NodeJS is getting better and is continuing to make developers happy. This is one of the reasons behind the explosive growth and adoption of NodeJS. NodeJS and NodeJS based solutions are always considered reliable options for testing and automation of testing. For example, NightwatchJS is an end-to-end solution for testing web applications. Though there are other JavaScript based platforms, **NodeJs** is one of the easiest choices for creating real time applications. Also it could be used both for server side and client side applications. The ability of the platform to cater to scalable solutions is something that makes it a favorite for enterprise applications.
We also use NodeJS for infrastructure maintenance mainly because it can easily handle even lengthy tasks without imposing a heavy load on the server. The footprint of NodeJS is also very small and it continues to be a preferred choice for event-based applications.
With the increase in emphasis for the incorporation of **DevOps in software development cycles**, it is a great time to understand how learning **DevOps for NodeJS development** and using NodeJS for developing DevOps tools would prove beneficial. There are in fact numerous courses that focus on DevOps and NodeJS. If you are a **NodeJs developer**, learning DevOps would be a great push in your career. In fact there is a steeply rising curve in the IT domain for those who are adept at these two major IT trends. And if you know how to apply one along with the other, the opportunities expand even further.

Dreadnot for deployment in DevOps:

NodeJS is slowly breaking free of its dependencies over the other languages. Using this platform to build user-facing applications is not new. But the recent trend is the use of this tool

to build deployment automation tools, like Dreadnot. Better deployment would enhance DevOps culture. Dreadnot assists continuous deployment which significantly shortens process times making it easier to add features and resolve errors after deployment as well. Possible errors during the deployment phase could also be reduced. Test-driven deployment cycles that are quick would help developers concentrate on making amendments and testers in finding bugs well ahead. DevOps is spoken about far and wide but still, the number of tools that actually favor this culture is pretty less and Dreadnot is one of the first tools that really helps implement DevOps. The current scenario is that DevOps teams are more accustomed to Ruby than NodeJS and other **JavaScript based platforms**.

Better business with DevOps and NodeJS

Agile and other methodologies are working on improving the overall processes. But corrections made at the higher level by correcting the overall practices would cause a more long term result. And this is precisely what DevOps does to the enterprises. And in the middle of all this, if a reliable future ready platform like NodeJS is used by the developers then the outcome would be much better than the perception. One is a tool for **development of web applications** and the other is a culture to be implemented at workplaces but they are growing hand in hand and this is why we say that the bond is getting stronger.

Incorporating one change at a time would be a good way to start instead of taking a long leap. NodeJS has been noticed mainly because comes with a non-blocking IO and as it is even driven and delivers a good user experience, it is an easy choice to develop DevOps aiding tools. The killer combination delivers reliable deployments, better automation, quicker

development cycles, and a proper streamlining of the processes. As **NodeJS** comes with the option to proxy as a web server, the **DevOps** team would be able to save all the time taken to set up a web server. And not to forget the use of the same JavaScript tool at both client and server end applications.

If **DevOps** can make workplaces more employee-friendly by strengthening the culture, NodeJS can help make apps that are user friendly. Given that there is a huge crowd of businesses now using both of these, you would be part of something big if you join the league right away. Better business outcomes can be promised as these both are greatly compatible and supportive of each other. There is a lot of scope now for improvement here and smaller businesses trying to delve into the DevOps way might find it a bit challenging. But remember that **NodeJS** could be learnt easily by developers who have worked with Java but embedding DevOps in the IT domain might be a slow but a steady process. And the impact as well as the time taken to experience the impact also might differ from one business to another.