



Sonatype Nexus Repository Review From A Customer



From IT Central Station, the leading review site for enterprise technology solutions.

Review by a Real User

Verified by IT Central Station



Architect at a consultancy with 1,001-5,000 employees

Architec9c59

WHAT IS OUR PRIMARY USE CASE?

At the moment we use it as storage, as a repository, the proxy to internet repositories, and for internal storage of our binaries. But we are looking seriously into using it for compliance to policy, for open-source dependencies that may have security issues or contradictory license usage. If certain dependencies do not comply with our licensing policies, then we want to be able to identify them. We are very interested in it to ensure the traceability of our open-source dependencies, to make sure that we are not using dependencies that could cause problems in the future, that could cause intellectual-property issues with the rest of our software. I wouldn't stretch it as far as calling it open-source governance. It's more of a safety check, to make sure that we don't make any mistakes that could cause legal problems later.

HOW HAS IT HELPED MY ORGANIZATION?

It has really sanitized our development lifecycle because we know where people will store their software, and we have a clear naming convention for this. Before that, people would store software in various places, on network drives, on random computers, and it was not obvious where things would be.

WHAT IS MOST VALUABLE?

For us, the ability to do proxying and federations of repositories is very important. It gives us flexibility. We are among the largest physics research laboratory in the world. With thousands of developers, we need to have good solutions to federate our teams. Those are the perfect features for us, the ability to proxy and to federate. It also has very good enterprise integration, so we are able to integrate it with the rest of our infrastructure for authentication, for role management. That is very useful.

WHAT NEEDS IMPROVEMENT?

We feel that if the product could be configured more easily through configuration files, instead of API calls and databases. That would make it easier to integrate with other DevOps tools. This is one of the hurdles that we encountered when we tried to integrate Nexus 3 with our OpenShift installation. The need to manipulate a dedicated Nexus database, instead of being able to generate configuration files, was a bit problematic. The inclusion of repositories that are currently supported by the community would be helpful, if possible. In particular, I'm thinking of Debian repositories. Otherwise, we don't have any request for large features because it's already a well-featured product. Everything else is included already. We are quite happy with the feature set.



FOR HOW LONG HAVE I USED THE SOLUTION?

More than five years.

WHAT DO I THINK ABOUT THE STABILITY OF THE SOLUTION?

We are very satisfied with the reliability. With Nexus 2 we have been very happy. For Nexus 3, which we deployed on top of OpenShift, we've had some issues with the storage, but I strongly suspect this is not due to Nexus itself. It's mostly due to the infrastructure we used underneath. It could also be that the product is not tested and the quality is not verified on cloud infrastructures like OpenShift. To be honest, I haven't read about it. But for us, I believe the problem is due to the infrastructure, not so much the product. That being said, I feel that Nexus 3 is generally less stable than Nexus 2. It's more picky with, for instance, authentication. If you're not authenticated anymore, the user interface is not very friendly to you, and it tends to kick you out or lose your data. More could be done in that area. For instance, with Atlassian products, when you get kicked out due to authentication expiry, you still have a chance to recover what you were working on. Maybe the same could be looked into for Nexus 3.

WHAT DO I THINK ABOUT THE SCALABILITY OF THE SOLUTION?

If you deploy a single instance, it scales very well due to the storage options that it offers. I can't comment so much on the performance because we have never had more than 100 users using it. But to scale up, to build a federation of Nexus instances, like we do here - we have a big organization - I don't feel that it's very easy for the same reason I mentioned earlier: the lack of configuration through configuration files; that you have to call an API and manipulate the database to configure Nexus instances. I get the feeling from our admins that it's not easy to scale up. They don't feel confident instantiating tens of these or hundreds of these.

HOW ARE CUSTOMER SERVICE AND TECHNICAL SUPPORT?

Sonatype's technical support is excellent: Very fast response time, they're very knowledgeable. They know exactly what to advise us. They're very good.

IF YOU PREVIOUSLY USED A DIFFERENT SOLUTION, WHICH ONE DID YOU USE AND WHY DID YOU SWITCH?

In our team, we did not use another solution. Before adopting Nexus we used just static websites that we copied files to and indexed manually. There was no other solution before that. In other parts of the organization they used Artifactory. They're switching because Sonatype is just more manageable. It offers more types of repositories that may not be supported in Artifactory, and the search facilities are better in Nexus.

HOW WAS THE INITIAL SETUP?

The initial setup is simple if you are a small organization, if you're setting up one instance. If you want to make this reproducible, for instance, you want to make templates to be able to offer the ability for an entire community to deploy their own instances on demand, then it becomes much more complex. One of my colleagues had to deal with this, and he has written a paper about the problems that he encountered that he wouldn't have encountered with other products like GitLab or other commercial products. For a single instance, it's a perfectly easy and comfortable process. If you want to scale up it's more difficult. Our deployment took a year, but it was not due to technical problems. It was more like scheduling delays and organizational changes. In total, it took three or four weeks to get something working and be ready to offer it to the organization. In terms of an implementation strategy, we rely on OpenShift. It's a Red Hat product, and now I think it has been acquired by IBM. But it enables your community to request a new service on demand. Our strategy was to make a template out of Nexus 3 for OpenShift, so that we could then make it available to all our users and developers for their own purposes. Then we had to ensure that it integrated well with our enterprise resources, authentication servers, or single sign-on facility. That took some effort. After that, when we talk about strategy, OpenShift dictates, more or less, the strategy you have to adopt to make it available to your users. You're quite well-

guided by OpenShift to on what to do next. Now the template is available to all users. They can just go to the catalog of templates, click on Nexus 3, and they get an open-source instance.

WHAT ABOUT THE IMPLEMENTATION TEAM?

We had an internal team dealing with it.

WHAT WAS OUR ROI?

It's hard to quantify because we didn't have any processes before. We started from zero. We spend less time chasing software, and that translates into more productivity, but I don't have a KPI to share.

WHAT'S MY EXPERIENCE WITH PRICING, SETUP COST, AND LICENSING?

We are quite lucky because the founder of Sonatype visited us in 2008 and granted a permanent free license to us as an organization, because it's a laboratory that does a lot of research, and it's in the public domain. He felt it was a good gesture that they could extend to us. I can't comment on the pricing.

WHAT OTHER ADVICE DO I HAVE?

My advice would be to think about the scalability. I would really advise everybody to go for it, to go for having repositories in their organization, to make their software and even hardware development more organized. Go for it, adopt one. We are using the latest version of Nexus 2. We are in the process of rolling out Nexus 3 on a much wider scale. In terms of using the solution to manage binaries, build artifacts, and release candidates across the DevOps pipeline, I would say yes, we're doing so. We're not as well-developed in terms of DevOps as other organizations, but we are certainly working that direction, and we use it for the DevOps lifecycle. We use GitLab and we also use Jenkins CI, and with these two products we are able to very simply deploy to Nexus Pro and perform staging operations, to make sure that, before we release to production, we can ascertain the quality of our software. We don't necessarily automate the deployment to production now, it's still a manual process. But we are working very hard to complete the DevOps loop. In terms of open-source governance, we should be much more proactive. Nexus is definitely a tool that makes this possible in an organization - to be proactive about your open-source choices. It's less critical in our organization because all our research is open. We try to stay away from intellectual-property problems. But on occasion we still have to pay attention to that and Nexus, with its advanced governance features, would have been a real time-saver if we had it at the time, so we are looking into it now. The number of developers using it right now is about 200 to 300, but because we're rolling it out to the rest of the organization, making it a much more easy service to afford with OpenShift. The audience could be up to a couple of thousands of developers. In terms of deployment and maintenance, for a single instance, we need about 10 percent of a full-time person. There are moments where it's very quiet and others where we encounter a problem and we have to take some action. But it's a very lightweight solution compared to what other commercial products or open-source products could cost, in terms of maintenance. I'd give Nexus Repository a nine out of ten. A perfect ten would have better support for configuration, for templating on the cloud infrastructure.

[Read 8 reviews of Sonatype Nexus Repository](#)