Jun-10-2020

Microservice performance Monitoring (Datadog or AppDynamics)

Snehil Maheshwari



# Why and What?

You got a project, and you got all sorts of plans, design is ready, development has already begun, and developers started asking for environment to deploy their first draft version, but hang on, let’s take a step back, when you talk about an environment, what does it mean? As an SRE you cannot just deliver a box with JRE(or any other thing to run your application) and a webserver installed to run application, you need to follow a consistent guideline (e.g. SRE handbook) to provide an environment where not only your application is going to run but it’s going to stay happily, and also you need a way to keep a keen eye all the time, to check if it is doing ok, and that’s where APM (Application performance Monitoring) comes into the play.

In Today’s world of complex applications, you need a helping hand of various tools to provide a very observant eye for your application, either you can develop of your own like Netflix, or you can rely on tools which are available in market. But again, are you sure what you looking for? Means before searching on google “*best tool for application performance monitoring*” and buy an enterprise version, you may want to consider few things,

* Is software/tool in subject is going to provide you an edge on both kind of aspect of monitoring, infrastructure and application. (I am considering database also an application only)
* is it database you are monitoring, or is it a web application exposed to outer world.
* What is the kind the application we are talking about here, is it a monolithic application or a very small microservice deployed in a multitenant container-based solution?

You need to think about all of these factors, because when we talk about public cloud, this becomes very crucial, now you are not only monitoring performance of your application but also it will help you to optimize the cost, because one aspect of monitoring is also scaling your application on the basis of results or alerts you receive, let me explain little further on this.

There are three two major factors that you monitor in terms of infrastructure monitoring.

* Memory utilization
* CPU utilization

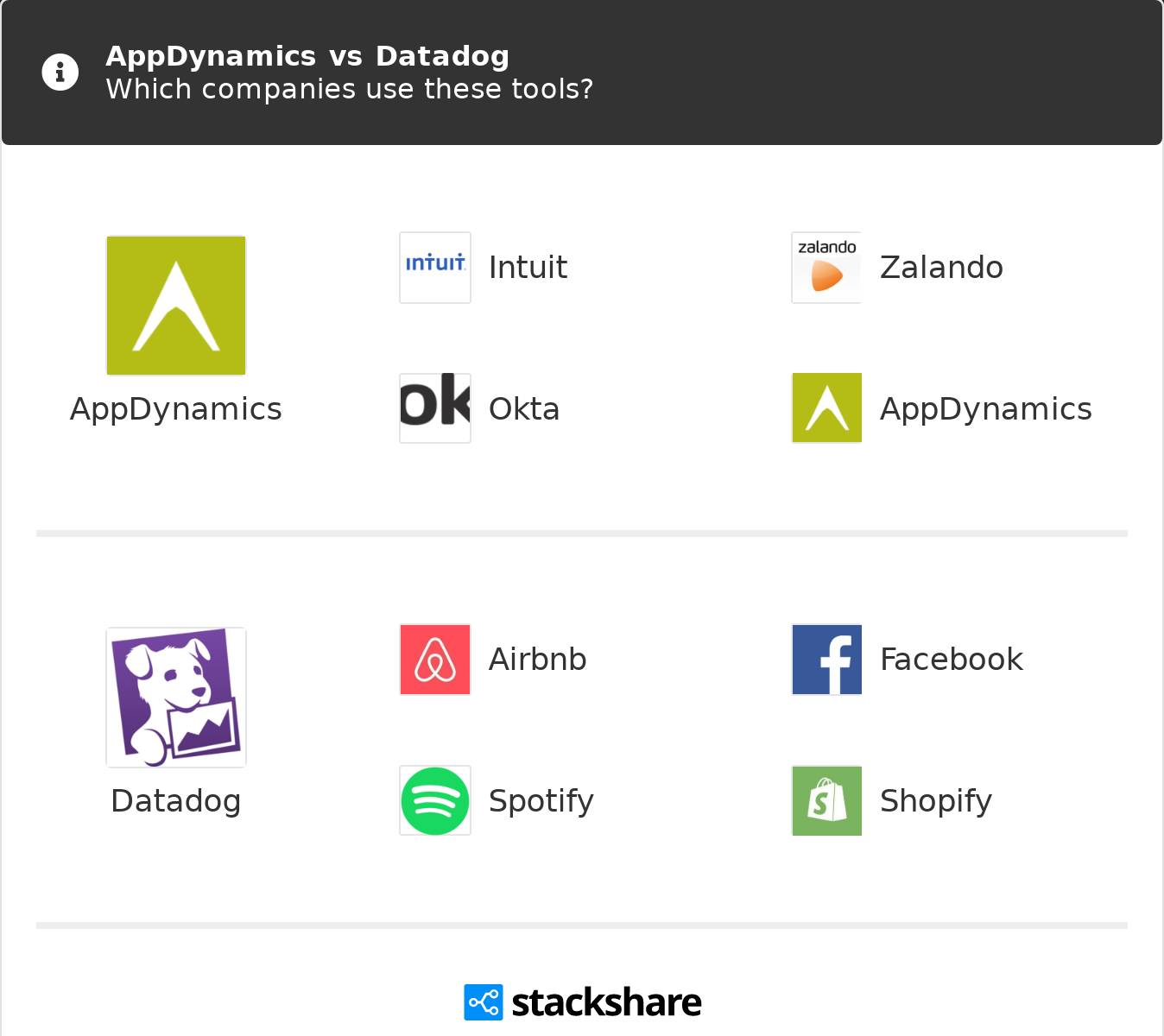
And there are the same factors based on which you scale your application, so if your APM tool is not providing proper result regarding these two, your scaling is not going to work properly, which is directly going to affect resource utilization on a bigger scale, and when we talk about public cloud providers, that’s what you are going get charged on.

And if you are talking about only a simple monolithic application deployed on a web container, it’s still a pretty easy job for you and also for a APM tool, but what if you are using a container-based application or microservice where processes play with [Linux namespace](http://ifeanyi.co/posts/linux-namespaces-part-1/) to provide isolation from host, how you can monitor something when it’s just a partial picture or in some case not a clear one?

My name is Snehil Maheshwari, I am an SRE/DevOps engineer for a financial domain enterprise and in next few parts of this paper I will try to compare two major APM tools for container-based solution, AppDynamics and Datadog. we will try to analyze why some tech giant went with one specific solution.

## Datadog Vs AppDynamics

Both are very well-known tools for APM, and both have completely different way of monitoring performance of application,



before we jump into the details how these application monitors performance of application, I want to give you a fare idea about the market coverage of both the tools.

stackshare did a pretty good job in comparison of market share [here](https://stackshare.io/stackups/appdynamics-vs-datadog), both the tools hit the market with in 2 years of gap. AppDynamics came in lot of discussion recently after it got acquired by Cisco back in March 2017

As I said earlier Both the tools have a very different approach of monitoring the performance of application. AppDynamics works on transaction flow to monitor how application is performing, while Datadog takes a same approach as many APM tools in market and works of metrics, so if you are looking for an APM tool which needs to provide you an insight of transaction flow at a very granular level, AppDynamics is your answer for that.

AppDynamics provides higher level of intelligence when it comes to event detections in transaction flow, but at the same time it has other component (machine agent) which will check resource utilization of your server, for which you need to have an agent running on your server same as Datadog, for process level monitoring AppDynamics doesn’t run any addition process, it’s very light operation of post to send feed to AppDynamics controller at real time, while Datadog does require an agent to be running on server to collect data from streams like logs.

Now if you remember earlier we were talking about infrastructure monitoring with respect to resource utilization in terms of CPU and Memory, Let’s emphasize on this a little bit more. As you already know AppDynamics closely follow transactional flow rather metrics, so obviously it is very tightly coupled with process, which in this case is container, now let’s see how container works in brief.

Any container basically uses different namespaces to generate isolation (Linux namespace not CM tool’s namespace), so processes can see only its own filesystem (using mount), its own process Ids (using PID), its own resources (using cgroups), these all are virtual isolation, underneath you still have same VM/machine.

I wanted you to have clear picture of containers, because in next few point I am going to talk about what are things which are still pose as challenges for AppDynamics, when it comes to a clustered container-based environment.

No correlation:

now consider one scenario, where you have multiple microservices running on same machine in different containers, and now you want to monitor all of them with AppDynamics, there is no easy way for you to get a full picture of node where all these container are running, however at the same place Datadog just need one agent running and on the basis of service call it will group them together and give a proper functional architecture of closely related services, and also it provides health of every single container, so while AppDynamics still lives with the isolation of namespaces, Datadog finds a way to couple them to provide node level view.

Multidimensional Data:

Just assume a scenario, where you are just looking for simple cpu utilization in and hour stats, pretty simple for any APM tool, now let’s add another dimension there memory, still doable, but what is going to happen when you will try to analyze whole cluster with all available dimensions, only wall time data (events time across whole transaction cycle) won’t cut it for you, you need a good visualization for that, that’s where AppDynamics lacks a lot, at the same time Datadog does a pretty good job with its tagging and also integrated apps which can feed their data directly to Datadog metrics.

Too much efforts to setup:

If you have only few microservice running in single container, and only few different end points, it’s good to have something like AppDynamics where auto detection of end points makes your life easier, but when it comes to container-based microservice environment, where every microservice is completely different from each other in terms of language, functionality etc., where you want to have a collaborated big picture, AppDynamics is not the solution for you, until unless you have dedicated resources which are just doing the work of instrumentation. But having a separate resource just for single piece defies the whole concept of DevOps.

While Datadog is a traditional metrics-based solution for APM with a highly active open source community, even it can also provide an end-user view without you going through any difficult setup beforehand, Datadog also uses network flow map, which provides a very clear picture about the flow of service.

Updates:

As I mentioned earlier AppDynamics is normally a part of your image as it’s tightly coupled with the process. (though you can design in such a way where every process in cluster will refer to a single app, but it’s a very complicated thing to maintain, when it comes to identify issues.), so if you want to make an upgrade to your AppDynamics major version, you must update all the images and deploy it again, which at the same time Datadog just needs to updated as master update, without a disruption to your service in anyway.

Memory based monitoring:

Honestly this is not something I am going to put in the name of AppDynamics only, memory utilization based monitoring is always a challenge for container-based applications, so AppDynamics is no different, as I have already mentioned earlier container plays with namespace only to have isolation, a simple misconfiguration of process argument can cause a false alarm because of high memory utilization, as container is just doing its job of utilizing all available free resources.

This is again not a problem with a metrics driven APM tool like Datadog, because instated of looking into individual processes, it works on unify logging and resource utilization of server based.

## To conclude it...

Datadog has a very clear advantage when it comes to work with clustered applications with more than 400 plus integration points, for example you can integrate it directly with your AWS account and it will automatically detect all the instances, DBs running with in it. Integration with lots of open source products and a very active open source community clearly gives an edge to Datadog when it comes to get a big picture of clusters application, which provide a very good operational ability to Datadog.

## Reference:

* <https://www.datadoghq.com/>
* <https://en.wikipedia.org/wiki/AppDynamics>
* <https://en.wikipedia.org/wiki/Datadog>
* <https://www.getapp.com/>
* <https://docs.appdynamics.com/>
* <https://stackshare.io/>
* <https://enlyft.com/>
* <https://www.gartner.com/>