

# Top tools to manage cloud apps

By Susan Perschke | [Follow](#)

Network World | May 4, 2015 3:00 AM PT

Once moved to the cloud, applications become more challenging to manage. Cloud app performance management and performance monitoring tools help identify bottlenecks and other performance metrics.

Good tools can help determine if bottlenecks are isolated to the application itself or if there are system-wide issues with a particular provider. The most granular tools can even do a deep-dive into the app to see if individual processes, such as database queries, are running optimally.

For this review, we tested four commercial products, Exoprise CloudReady, AppNeta, ThousandEyes and Dynatrace. We focused mainly on ease of deployment, day-to-day management, overall features and cost. We did not focus much on network overhead, but this is certainly a consideration, especially where agents need to be deployed.

All four products are quite capable and deliver well beyond basic network and application monitoring features. Where they differ is in the granularity of the insight they provide and the focus of the solution.

**+ ALSO ON NETWORK WORLD: [4 open source monitoring tools that deserve a look](#) +**

Our Clear Choice Test winner, with some caveats, is AppNeta, due in no small part to its modular design that casts a very wide net. It also has a solid and up-to-date user interface that is easy to navigate and use. By providing so many features, AppNeta allows you to perform most monitoring tasks, from very simple response metrics to deep dives into every nut and bolt of a Web application.

The caveats are mainly cost and learning curve; in order to utilize all features you need to license several of the modules and also spend some time acquainting staff with the features. Not all organizations will have the time or inclination to allocate the resources needed to fully utilize the entire solution. However, the script generator that allows you to record steps as you go through a Web application can help automate the process to an even higher degree.

Our runner-up is Exoprise CloudReady, not necessarily for its total set of features, which are solid but not nearly as comprehensive as AppNeta, but for ease of use and cost. Exoprise CloudReady was by far the easiest solution to get up and running quickly. Knowing first-hand how little time many IT administrators have to allocate to one more task, we can appreciate that. The pricing structure is also very competitive and simple to understand.

ThousandEyes and Dynatrace share the 'second runner-up' spot as they both provide very good application monitoring. Dynatrace brings application insight on par with the AppNeta TraceView module, but it's a bit complicated to install and we think the interface could benefit from another review from the 'ergonomics' department. A big plus is that it can be run as a self-contained solution in-house with no need for interaction with third-party infrastructure.

ThousandEyes matches AppNeta in that it provides the ability to do simple monitoring while also allowing for pretty comprehensive application insight. By providing over 100 worldwide locations to test from, ThousandEyes gives administrators the ability to see how an application performs from every corner of the world. ThousandEyes also has a very cool feature that allows you to share live or snapshot data on the fly with others without the need for a separate product subscription.

Here are the individual reviews:

## **Exoprise CloudReady**

CloudReady monitors cloud apps and online services such as AWS, Azure, Office 365, email, Salesforce, DropBox and SharePoint. It does not require agents installed on the target, but a lightweight sensor agent is required if you wish to monitor from your own infrastructure. In addition to more traditional network path diagnostics, CloudReady uses synthetic transaction monitoring to emulate typical user transactions and also crowdsources aggregate data from all customers to deliver a better overview of common network infrastructure, allowing customers to pinpoint potential bottlenecks.

Exoprise is managed from a hosted Web interface or the Secure Management Shell that is installed locally; you can deploy sensors from either.

### **+ A LOOK BACK: 10 software as a service companies to watch +**

Getting started takes only a few minutes and a free 15-day trial is available. After providing some basic information we were ready to load the online dashboard, which was intuitive and uncluttered. The first step is to create one or more sensors. Sensors can be deployed as 'public' or 'private', the difference being that 'public' sensors run in the cloud on Exoprise infrastructure and 'private' sensors run on your own infrastructure.

We started with a basic public ping sensor, configured using a short wizard that allows you to enter five IP addresses or websites. We chose a mix of both. Next we selected the interval and identified which Exoprise server to ping from. The choices included West and East Coast options along with one location each in Australia, Japan and Europe. While we waited for our ping data to start collecting, we configured a 'private' sensor to monitor one of our SMTP servers.

The first step was to download and install the Secure Management Shell, which is essentially a browser-like application that runs locally. It has the same look and feel as the full website version, a feature we like. The Secure Management Shell creates a 'site' from which you can deploy sensors, meaning you could set up 'sites' throughout your organizations to monitor both public and private infrastructure from one or multiple locations.

The shell took only a minute to install and configure, and we were then ready to deploy our first ‘private’ sensor. We decided to check on the status of one of our hosted email accounts. As with the ‘public’ sensor, a wizard asks for the email credentials, how often to check and finally from where to check. The last step is different from a ‘public’ sensor in that you select one of your own infrastructure sites. This could be a desktop, server or VM.

Data collected by the sensors is displayed on a dashboard that can be accessed using either the shell or Web interface. The dashboard is easy to read with a small ‘widget’ display for each sensor that provides at-a-glance information for each. Items on the dashboard can be sorted by criteria such as status, location and type. For example, one sensor can display a graph for upload and download speeds for Azure, while another may display DNS lookup time for a particular DNS server. We found the ‘trend’ icons to be helpful as they show if something is not working properly by displaying color-coded condition icons, green orange and red.

The dashboard also displays alarm history for each sensor and you can click to drill down to more detail. The detail page provides context-sensitive information based on the type of sensor. This can include items such as network path performance, data point details and errors. Alarms can be configured for any sensor and the alarm options depend on the type of sensor. The alarm configuration interface is pretty granular and alarm notifications can also be sent via email. We found the pricing structure refreshingly simple; \$100 per month per sensor for up to 20 sensors; for sensors beyond the first 20, there is discount pricing available at \$90 per month for 21 to 40 sensors and so on. Annual pricing is also available at substantial discounts. For instance, 100 sensors paid monthly is \$80 per month, paid annually the price drops to \$50 per month.

## NETRESULTS

Product	CloudReady	AppNeta	Dynatrace	ThousandEyes
Pros	Easy to launch, ability to test from multiple locations, numerous pre-built sensors ready to use.	Feature-rich solution, TraceView provides solid application detail, excellent user interface.	Detailed application insights, self-contained, no need to rely on third-party infrastructure	Ability to test from over 100 locations world-wide, great sharing feature.
Cons	Lacks some of the granularity of the other contenders.	Multiple solutions requiring multiple licenses, complex pricing structure, steep learning curve.	Cumbersome to configure, navigation could be more intuitive.	Interface navigation could be more intuitive.

We liked the ease of installation and basic configuration of CloudReady. We were up and running in a few minutes. We also liked the many pre-configured sensors. CloudReady isn't as feature-rich as some of the other products reviewed, but what it does, it does well.

## AppNeta

AppNeta provides what the vendor calls ‘full stack’ application performance management. Four different modules (TraceView, AppView, PathView and FlowView) together provide a solution for performance monitoring of Web apps, SaaS and application networks, together with other network performance monitoring and analysis. Monitoring is performed from a Web interface hosted by

AppNeta and the sensors are run either from your own infrastructure or locations provided by AppNeta globally. There are several items that can be monitored without the need for agent deployment, but in order to use the TraceView module you need to deploy an agent to the server running the Web application.

**+ ALSO AppNeta: A compelling multi-site wide area and network performance management suite**

+

A free trial is available for some of the features of AppNeta, but in order to review the whole solution the vendor provided us a PathView appliance. The appliance is available in six models, from a small one that fits in your palm to large rack-mounted units. It is also available as a VMware virtual machine, but is not yet available for Oracle VirtualBox or Microsoft Hyper-V.

The appliance we tested was the mid-sized r-40, a 1U rack unit that comes pre-configured from AppNeta with customer-supplied information such as user name, password, IP addresses, etc. The appliance is essentially an agent that can be used to check either on-premises or in-cloud applications. Customers can deploy one or multiple appliances (virtual or physical) to achieve insight from multiple geographic locations. Once up and running, we accessed the appliance from a Web interface that provides additional configuration options. For most installations, the appliance does not need much hands-on beyond the initial configuration.

Primary management functions are performed from the PathView Cloud Web app, hosted by AppNeta. Once we logged in we were presented with a screen that showed our newly installed appliance and after giving it a location name we were ready for a test drive. Depending on your subscription level, the AppNeta Web application will display enabled features in a left-hand navigation panel. The dashboard provides a basic overview of the data collected by AppNeta with alerts, system status, number of appliances connected along with the number of licenses used and licenses available.

PathView, the first module, provides insight into the end-to-end performance of the path between two network points, such as access to an application from one location to another. A path is configured through a short wizard that guides you through the process. We set up a path from our r-40 appliance to one of our routers at a commercial data center offsite from our testing facility. This provided a hop-by-hop analysis of the path with information about latency, data loss, round trip times and the ability to drill down for further details for each hop and recommendations for resolving bottlenecks. The data is presented in several formats, but we especially liked the easy to read graphs for each category. Paths can be run on demand or scheduled to update at set intervals.

The next module, AppView, monitors HTTP performance as experienced by end users. AppView can be used from an appliance (AppView Enterprise) or from one of the many available appliances deployed by AppNeta throughout the world. First, we manually created a job that would measure a few different end-user aspects of a fairly large commercial website, such as overall response time and throughput.

AppView also provides a script generator that is an add-on to the Firefox browser. It essentially records your browsing behavior as you navigate through a website or Web application, and saves it as script you can use to repeat the same steps as a job automatically. We configured our script to navigate to a

site, click on a few links, log in using a user name and password, click a few more links and then logout. Once AppView monitors have been set up, they can be viewed from the dashboard, which displays a world map with pinpoints for each of the monitoring locations that can be clicked on to drill down for additional detail.

The TraceView module provides a detailed look at Web applications running technologies such as PHP, .Net, Java and Ruby, among others. This requires the installation of an agent on the application to be monitored. You can create a custom agent based on your environment by toggling options on/off and providing a download link. The install wizard provides a live view of the progress as your server(s) come online as you install the agents. After installing each agent we were able to view our application in the PathView Cloud management console. TraceView allows you to focus on all aspects of an application from the front-end to the back-end. For instance, we were able to see that one call to the database was consistently slow every time it was initiated, which pinpointed the bottleneck that we needed to address to improve performance.

The fourth module of the AppNeta solution, FlowView, works with the PathView appliance to analyze the type traffic flowing through the appliance. There are more than 1,200 categories that the traffic can be grouped into, such as gaming, business apps and social media. The results are displayed in a graphical interface that shows how the traffic is allocated.

# Top tools to manage cloud apps

By Susan Perschke | Follow

Network World | May 4, 2015 3:00 AM PT

*Page 2 of 2*

The vendor told us the pricing structure essentially breaks out along the lines of apps you built and apps you subscribe to. Even with this understanding, we found the pricing structure to be a bit too complex for our liking. For instance TraceView is available in Project, Startup and Enterprise versions, with the Project version being free and the other two at \$79 and \$119 per host per month respectively. AppView is available in three versions as well (Global, Office and Data Center), priced at \$39, \$338 and \$868 per month. Global is priced per monitor whereas the Office and Data Center versions include 10 and 40 monitors each. The latter two also include PathView at no extra cost, but licensed separately, these are priced at \$239 and \$729 per month for the Office and Data Center versions. Then, if you add FlowView to the mix you're looking at another \$338 or \$868 per month.

There is a lot to like about AppNeta, including the excellent user interface and flexibility to monitor from third-party worldwide locations as well as from your own infrastructure. In addition, AppNeta allows you to perform a simple check to see if a cloud app is responding or deep dive using TraceView to really get into the details of how the application is performing. The minus is that in order to utilize all of the features you have to license all of them and also get up to speed on learning the various features. Also some cloud providers may balk at installing the requisite agents needed to utilize the AppNeta TraceView features.

## **ThousandEyes**

ThousandEyes can be deployed either using enterprise or cloud agents, or a browser plugin to provide end-user visibility. ThousandEyes cloud agents are globally distributed, installed and managed by ThousandEyes at over 100 locations and are shared across multiple customers. In addition to monitoring on-premises applications, ThousandEyes can also monitor SaaS applications along with IaaS. In addition you can run your own agents from locations branch offices, a data center or a key customer's location.

Getting started takes only a few minutes and when launching the dashboard for the first time we were presented with several support options including a 'getting started' screencast and the option to chat with tech support (a favorite of ours as you don't need to call or send an email and wait for a reply). The initial dashboard is clean with a navigation bar at the top and a support pop-in menu on the right.

The terminology varies from vendor to vendor when it comes to cloud app monitoring tasks. ThousandEyes calls theirs tests and we started out by creating a few simple tests to check the availability and load time of a website and also to check the response to DNS queries directed to one of our DNS servers. The tests are created using a simple one-page configuration and they are grouped into four categories; Network, DNS, Web and VoIP. Each category has a handful of sub categories, but we found the configuration very self-explanatory and most network admins will be very familiar with the options available.

Once a test has been created, the collected data starts appearing in the dashboard along with other tests already configured. The dashboard displays some basic information for each test, such as status, and a graph showing the current trend of the data collected. From there, you can drill down to view more detail for a particular test. Once on the detail page, there is a drop-down of other tests that makes it easy to navigate to the detail view for all available tests.

One of the main features on the detail page is the graph from which you can see the trend for a given time period or zoom in to view information for each data point collected. Another is the world map that shows the current agents used for a test and from here you can click any location to see data from that location. For example we could see that the DNS resolution time for our server located in Denver is 48 ms from Newark, N.J., and 174 ms from Sydney, Australia.

We liked the ability to share live data with your team or third-party providers to resolve problems faster. This is done from a share link that can be configured a few different ways; you can send a link with snapshot data for a specific date range, a public link to live data view or export to a couple of different formats.

High level data is available for the past 90 days and detailed data available for 30, but you have the ability to take snapshots of events that can be saved indefinitely. ThousandEyes provides several ways to create custom on-screen reports that can be saved or just used for a one-time view of certain data.

As mentioned earlier, in addition to using existing ThousandEyes infrastructure, you can run your own enterprise agents from anywhere in your own network infrastructure. This can be accomplished through installing the agents on any common Linux flavor box or by downloading a virtual server available for VMware, VirtualBox or Hyper-V. We downloaded one of the virtual servers and it started right up with no configuration needed other than the network setting. The VM is managed from a Web interface and goes through a short wizard on first use. Upon configuration, the enterprise agent is available for use just like one of ThousandEyes cloud agents.

ThousandEyes makes it easy to get started with basic monitoring and provides a good mix of features for those seeking some additional granularity without too much complexity. One of our favorite features is the share feature and we also give a thumbs up to the more than 100 geographic locations

from which ThousandEyes can monitor application performance. One negative is that the pricing calculations can get complicated in a hurry and costs can add up rapidly when deploying multiple tests.

The pricing for the enterprise agents is straightforward; the Standard version is \$99 per month per agent and the Pro version is \$299 per month per agent. The difference between the Standard and Pro versions is the ability to monitor page load and Web transactions with the Pro version. There is also a free Lite version with the same features as the Standard version, but with some limitations.

## **Dynatrace**

Dynatrace provides network performance and application monitoring for platforms such as Java, .Net and PHP. They also have user experience management offerings, as well as synthetic monitoring that simulates Web application loads. There are several moving parts to Dynatrace, including an agent that needs to be installed on the server running the Web application and a 'collector' that processes events collected by the agents. The data is managed by a server running on a LAN or hosted in the cloud by Dynatrace, both of which provide a Web interface with several modules that allow insight into the application.

### **+ ALSO Nordstrom's use of APM lets it identify app performance issues in minutes vs. days ]**

To get started you log in to a Web dashboard hosted by Dynatrace. On first login, this provides you with some basic information about Dynatrace and a step-by-step guide on how to get started. The first step is to download the Dynatrace server, which is available in 32- and 64-bit for both Windows and Linux. The installation walks you through a simple wizard, which got us up and running on a Windows 64-bit server in just a few minutes. There are memory and CPU requirements depending on the number of transactions the server will handle. These range from a single-core, 3GB server for small operations to 24-core, 22GB server for an extra large installation. The extra large installation can handle 1,200 agents and over 3,000 transactions per second.

When the installation is complete, you are presented with a browser-like interface from which you select the type of application to monitor. Dynatrace provides quite a bit of granularity on how and what to monitor for a given application, both in terms of client-side technology such as Web, iOS and Android, and server-side, where you can specify the type of application server from IIS/.Net and Apache to Oracle, IBM and Java technology.

In order to monitor a Web application, an agent is required on the server where the application resides. There are several agents available for most common servers (Apache, IIS, NginX) and we downloaded one to work with our test Web server. With the agent up and running, we loaded the Dynatrace dashboard, which provides a map-like overview with nodes representing each part of the application infrastructure. For example, you will see requests from visitors broken into categories like desktop browser, mobile browser and mobile app.

There are nodes representing the front-end Web servers and nodes for the back-end functionality such as payment processing, database servers and business applications. Each node is color coded, with green representing no issues and red indicating problems. By mousing over each node we could see a



short summary of that node's status, for instance our Web server indicated a memory issue and by clicking to view the details we could see that there were page faults occurring occasionally under load.

Once into the detail view, you can pretty much evaluate each aspect of a Web application, from the first client request to the last response, and everything in between. One of our Web applications was making outside calls to load JQuery JS files and we could see that they were loading slowly; by moving these to our own server they loaded much faster on subsequent calls.

Dynatrace provides a very solid interface with a multitude of ways to view the data. You can view details about business metrics and performance changes over time as well pinpoint transaction hotpots and analyze browser performance. The graphs and data presentations are easy to read and what they call the 'Start Center' provides you with easy access to the various sections of the application.

In addition to monitoring your private cloud application, you can also deploy Dynatrace agents to monitor third-party hosting solutions like Amazon and Microsoft Azure. This gives you not only insight into how the application is performing, but also the layers that are added by the hosting provider, thus allowing you to also pinpoint potential bottlenecks not contained in the application.

Our review focused on the self-hosted server solution of Dynatrace, but the product is also available as a hosted version, which offers the same features. According to Dynatrace the cost is the same, except there is an extra setup fee associated with the hosted solution.

Dynatrace does not publish pricing on its website, nor was the company particularly open about its pricing in our discussions with them. However, when presented with a specific use case scenario (.Net application with about 8,000 to 10,000 unique visitors daily), they provided a ballpark cost of \$10,000 per year, with options for discounts for longer term contracts. It should be noted that for high volume sites you do not need to license your application's total traffic; instead you can elect to use a subset of total traffic. According to Dynatrace, several of its large clients elect to analyze only a fraction of total traffic.

Similar to the TraceView module of AppNeta, Dynatrace provides a comprehensive solution for those seeking more than just a cursory overview of a Web application's performance. It does require a bit more effort to install and configure than some of the other offerings, but in return you are rewarded with very detailed application information. On our server (quad-core, decent memory) we found the interface to be a bit slow, but this may speed up with additional resources. We also liked the self-contained nature of Dynatrace as the server runs on your infrastructure.

As previously noted, all four products were capable performers. Selecting a solution for your organization boils down to a few key decisions: how much detail do you need, how much overhead are you willing to incur to collect data, and what is your app monitoring budget. As expected, the Cadillac solution costs quite a bit more, and if you have more basic monitoring needs, a simpler solution may suffice.

*Perschke is a web and database developer with 15+ years of industry experience. You can reach her at [susan@arcseven.com](mailto:susan@arcseven.com).*

## **Susan Perschke**

Perschke is a web and database developer with 15+ years of industry experience. You can reach her at [susan@arcseven.com](mailto:susan@arcseven.com).

