APPDYNAMICS

INSIGHT SERIES
Docker

Monitor applications and microservices in agile Docker environment

What is Docker?

Docker is an open container technology platform for building, shipping, and running distributed applications. It gives programmers, development teams, and operations engineers the common toolbox they need to take advantage of the distributed and networked nature of modern applications. Docker allows applications to run on any Linux platform, irrespective of what tools were used to build it, making it easy to distribute, test, and run software.

These characteristics make Docker an ideal platform for microservices, a software architecture style in which complex applications are composed of small, independent processes communicating with each other using language-agnostic APIs, often HTTP resource APIs.

How does Docker help Dev and Ops?

Docker helps developers by providing very portable packaging for their applications, so that they spend less time customizing the OS environment or ensuring dependencies and focus more on developing applications. Applications running in Docker containers are easy to modify, update, and scale without bothering neighboring containers. IT ops professionals find it easy to deploy. Efficient use of system resources by Docker allows IT ops professionals to deploy more applications on fewer hardware resources.

Challenges in Docker-based environments

The ease of Docker deployment, combined with the trend of microservices, provides lots of advantages. However, this also creates new application management challenges.

- Application complexity explodes in a microservice-based architecture, as applications can potentially depend on tens, hundreds, or even thousands of microservices and containers.
- Ensuring optimal application performance proactively is challenging, as the application performance is dependent on many distributed application components, microservices, and supporting Docker infrastructure.
- Root cause analysis for application performance issues becomes very complex. For example, an application’s response time may be impacted by high memory usage within a Docker container, but it will be very difficult to isolate the problem if you can’t drill down from the application to Docker container metrics.This can result in high mean-time-to-resolution (MTTR) and poor end-user experience.

> *“Docker makes developers’ lives easier. Among other benefits, Docker makes it incredibly easy to package and distribute software, the ‘thankless scut work’ that can consume 90 percent of enterprise IT budgets.”*
>
> *Al Gillen, IDC*

IDC Analyze the Future

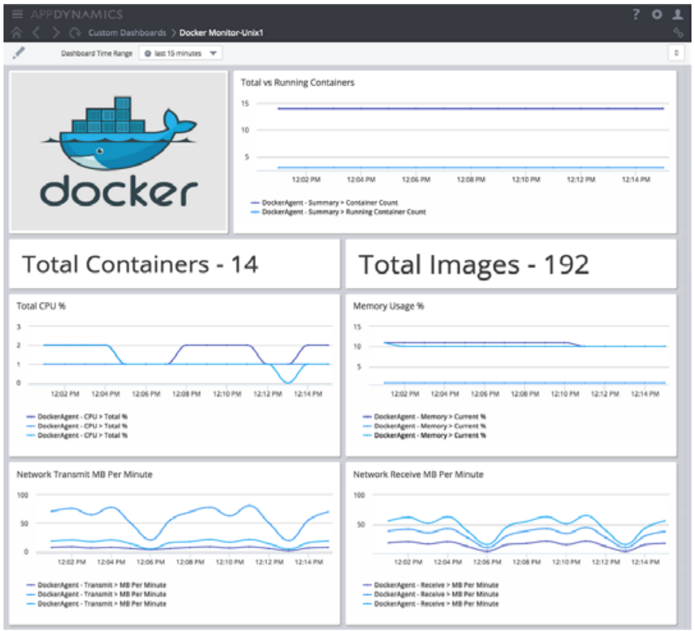
KEY FEATURES

- Monitor Docker metrics such as the total number of containers, running containers, images, CPU usage, memory usage, network traffic, etc.
- Correlate Docker metrics with metrics from the applications running in the containers.
- Dynamically baseline Docker metrics automatically and alert on deviation from the baseline performance.
- Customize health rules, policies, and actions around the Docker metrics.
- Utilize out-of-the-box custom dashboards for key Docker metrics.

The AppDynamics Docker monitoring solution

AppDynamics provides visibility into complex, distributed applications and business transactions composed of multiple, smaller, decoupled (micro) services deployed in a Docker environment. The **AppDynamics Docker Monitoring Extension** monitors and reports on various metrics, such as:

- the total number of containers
- running containers or images
- CPU usage
- memory usage
- network traffic, etc.



APPDYNAMICS

Custom Dashboards > Docker Monitor-Unix1

Dashboard Time Range: last 15 minutes

Total Containers - 14

Total Images - 192

docker
Total vs Running Containers
15
10
5
12:02 PM
12:04 PM
12:06 PM
12:08 PM
12:10 PM
12:12 PM
12:14 PM
DockerAgent - Summary > Container Count
DockerAgent - Summary > Running Container Count
Total CPU %
3
2
1
0
DockerAgent - CPU > Total %
DockerAgent - CPU > Total %
DockerAgent - CPU > Total %
Memory Usage %
15
10
5
DockerAgent - Memory > Current %
DockerAgent - Memory > Current %
DockerAgent - Memory > Current %
Network Transmit MB Per Minute
100
50
0
DockerAgent - Transmit > MB Per Minute
DockerAgent - Transmit > MB Per Minute
DockerAgent - Transmit > MB Per Minute
Network Receive MB Per Minute
100
50
DockerAgent - Receive > MB Per Minute
DockerAgent - Receive > MB Per Minute
DockerAgent - Receive > MB Per Minute

Figure: AppDynamics Docker Extension dashboard

The AppDynamics Docker Monitoring Extension gathers metrics from the Docker Remote API, using either Unix Socket or TCP, giving users a choice of data collection protocols.

The Docker metrics can now be correlated with metrics from applications running in the container. For example, the overall performance (calls per minute) of a Web server deployed in a Docker container can be correlated with Docker performance metrics (network transmit/receive, and CPU usage). As the number of calls per minute to the Web server increases, you can see that the network traffic and CPU usage increases as well.

The Docker monitoring extension also creates an out-of-the-box custom dashboard with key Docker metrics, as shown in the screenshot above. This out-of-the-box dashboard helps customers jump-start the monitoring of their Docker environment.

Customers can leverage all the core functionality of AppDynamics (e.g. dynamic baselining, health rules, policies, actions, etc.) for all the Docker metrics, while correlating them with the metrics already running in the Docker environment. In addition, AppDynamics runbook automation can help customers elastically scale the Dockerized environment as workloads grow.

KEY BENEFITS

- Automatically discover business transactions across applications and microservices deployed in Docker containers and show detailed application topography and dependencies.
- Identify problems before they impact customers and minimize false alarms with alerting based on dynamic baselines of fully correlated Docker and application metrics.
- Quickly resolve application performance issues by correlating application and Docker performance metrics and identifying root cause issues.
- Scale the Dockerized environment elastically as workloads grow with help of AppDynamics runbook automation.

Try it FREE at appdynamics.com