

15 Minutes Introduction to AppDynamics

KickStarter Series

Karun Subramanian



www.karunsubramanian.com

Contents

What is AppDynamics and how it can save the day?	3
The plumbing – How does it work?	4
How can I get started, really?	6
Configure your first Java Agent	7
Look inside your Java Application, like never before	12
Get alerted before things get worse	13
Monitor your hardware too	14
Where to go next?	15

What is AppDynamics and how it can save the day?

AppDynamics is a leading Application Performance Management (APM) product. It is a tool that monitors your Application Infrastructure and gives you code level visibility. It is supported for all major technologies (Java, .NET, PHP, Node.js, NOSQL etc) and can be installed either as on-premise or as SaaS (Software As a Service) solution.

A piece of software called **Agent** is installed in the Application to be monitored. The Agent collects the performance metrics and sends them to a Server process called **Controller**. Controller processes the metrics and presents them via Web Browser. A monitoring analyst can configure Alerts and generate reports using the Web Interface.

The agent constantly monitors the application. Since it uses byte code instrumentation technology, Agent has hooks to every line of code. This is how AppDynamics is able to provide code level visibility. Agents are available for most of the popular technologies.

Most of the monitoring capability is 'out of the box', including Alerts. Another neat 'out of the box' feature is the 'Application Flow Mapping'. AppDynamics discovers various subsystems and backends and draws them beautifully on the browser. At times, **you will be surprised** to discover about backends that you never thought your application was talking to. In addition AppDynamics 'learns' application behavior and automatically sets baselines, and alerts when the deviation from baseline is not normal (anomaly).

The image below shows a typical AppDynamics Controller UI.

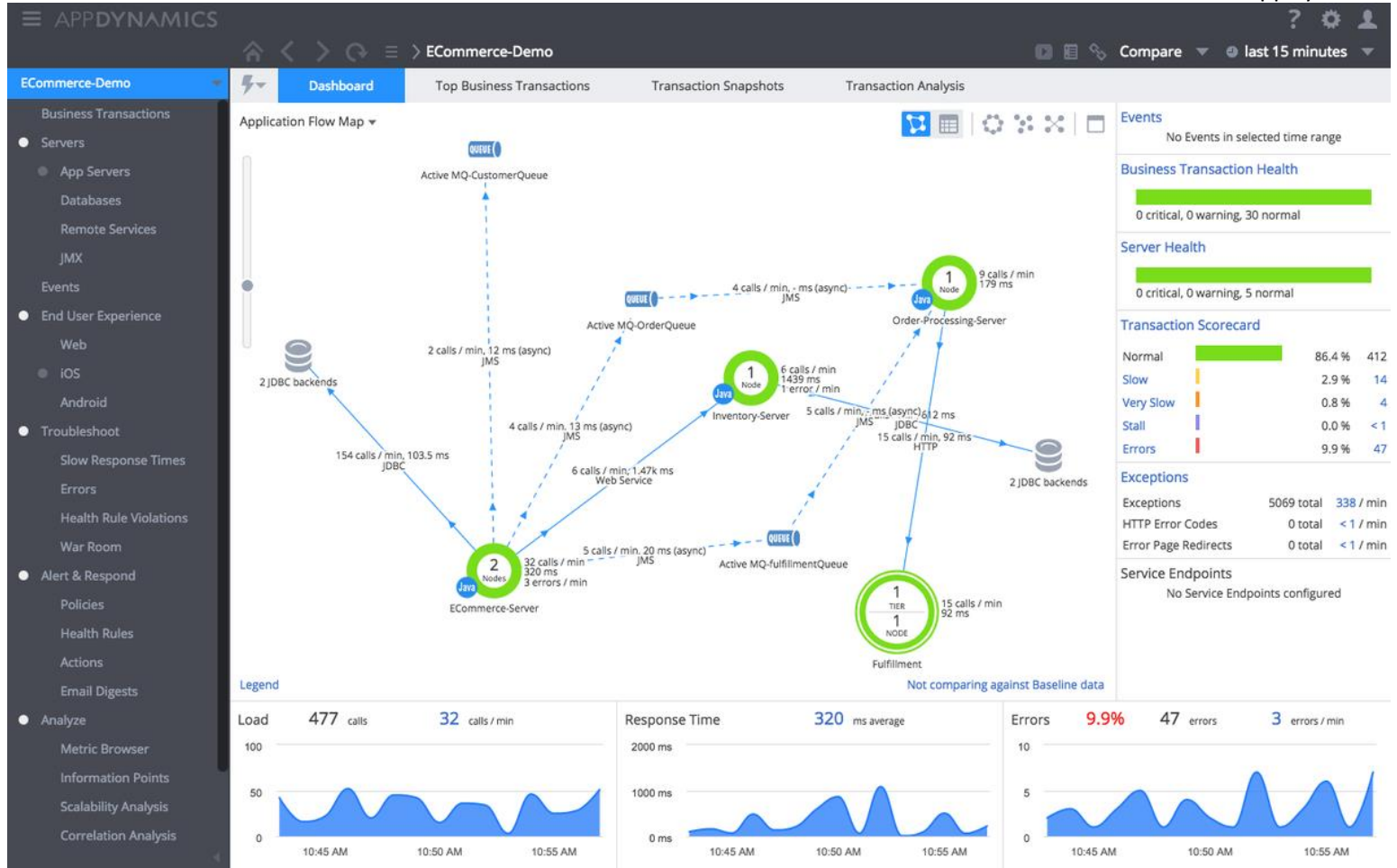


Image Source: www.appdynamics.com

AppDynamics employs the concept of 'Business Transaction' (BT). A BT represents a service provided by your application that is invoked by the user. All involved subsystems and backends for a given BT are tracked and reported by AppDynamics. BTs are automatically discovered but you can define your own BTs as well.

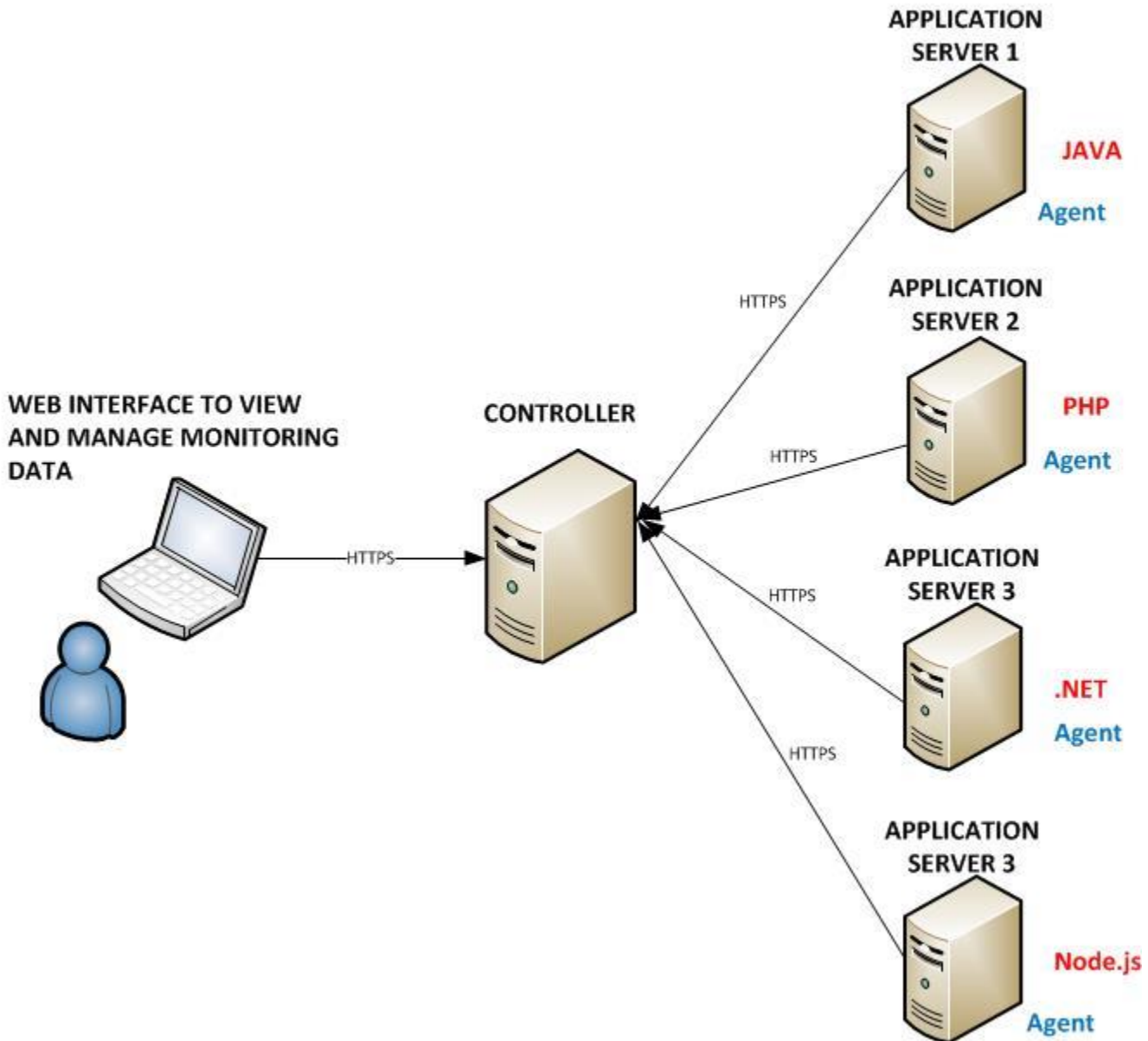
Using a special type of agent called 'Machine Agent', AppDynamics can monitor hardware too. Basic resource utilization such as CPU, Memory, Disk usage are monitored. Machine agents are available for most OS (Windows, Linux, Solaris etc).

AppDynamics lets you write your own extension using shell scripts or Java. It is a very powerful feature that enables you to monitor virtually anything. There are numerous extensions already developed by the community and available for download.

The plumbing - How does it work?

Image Source: AppDynamics.com

AppDynamics High Level Architecture



The controller is the engine of AppDynamics. It receives monitoring data from Agents installed on the Application servers (or Database Servers, Web Servers etc). The user accesses and manages the monitoring data by connecting to the Controller via web browser (usually at port 8090). It helps to understand the terminology used in AppDynamics to know how it works.

Node

A node is a monitored Server or JVM (or CLR). Typically the node corresponds to an individual Application Server (Physical or Virtual)

Tier

A tier is a logical grouping of nodes generally relating to a particular business function. Note that an agent cannot belong to more than one Tier

Business Application

A Business Application is the top level container in AppDynamics. It contains Tiers. Role Based Access Controls in the Controller UI are implemented at the Business Application Level

Backends

Backends are any external systems to the AppDynamics agent that are not instrumented. For example, if a .NET Application that is instrumented makes a call to a SQL Server Database Server, if the SQL Server is not instrumented with AppDynamics agent, it is considered a 'backend' in the .NET application. The response time for the backend calls are recorded (along with load, errors etc). However, deeper level metrics are possible only when the remote system is instrumented. AppDynamics automatically recognizes most of the popular backend systems.

How can I get started, really?

In order to get the taste of AppDynamics, you can get started for free. At a high level, here is what you have to do:

1. Identify a Server that can serve as AppDynamics Controller. Controller is available for the following platforms

Windows (32 and 64 bit)

Linux (32 and 64 bit)

2. Download and install AppDynamics Controller (<https://download.AppDynamics.com>). Installation is pretty straight forward.
3. Download and install one of the App agents (Agents are available for the following):
 - JVM (Oracle HotSpot, JRockit, IBM JVM)
 - .NET
 - PHP
 - Node.js
 - Python
 - WebServer on Linux (Apache)
 - DB Agent (DB2,Oracle,MySQL,SQL Server, PostgreSQL, Sybase IQ)
4. Configure the App agent; Restart the Application and start monitoring like never before (instruction on configuring a Java agent in the next section)

Note:

The controller listens on port 8090 and 8091. Make sure these ports are not blocked by Firewall.

Controller on Linux requires *libaio* installed. Also, the number of File Descriptors available should be at least 65635.

You will need to create two or three user accounts during the install. Remember all the user names and passwords (especially the root user for the Controller)

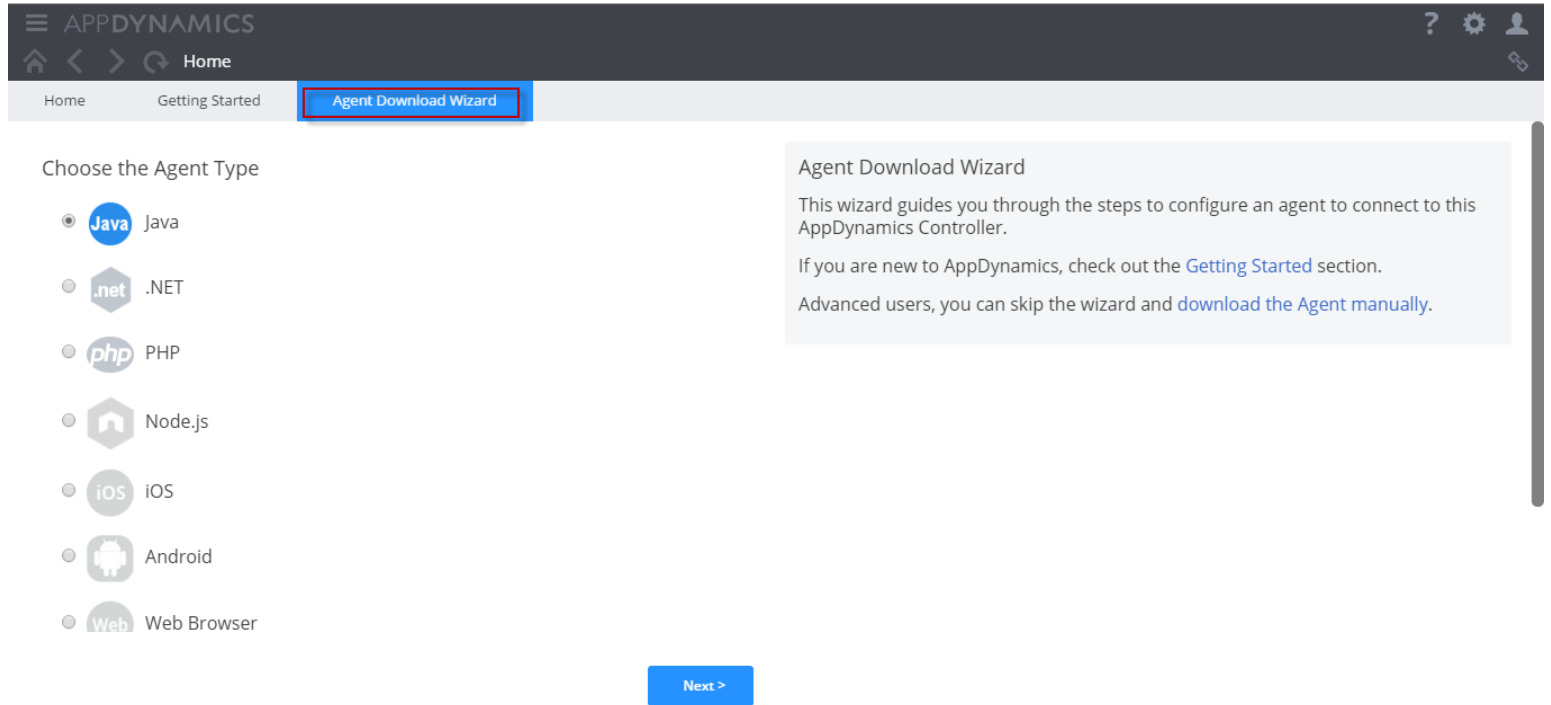
Instead of installing the Controller in your local environment, you can choose to use one of AppDynamics hosted Controller (SaaS based). I suggest on premise to start with so that you get the real taste of managing the Controller first hand.

Configure your first Java Agent

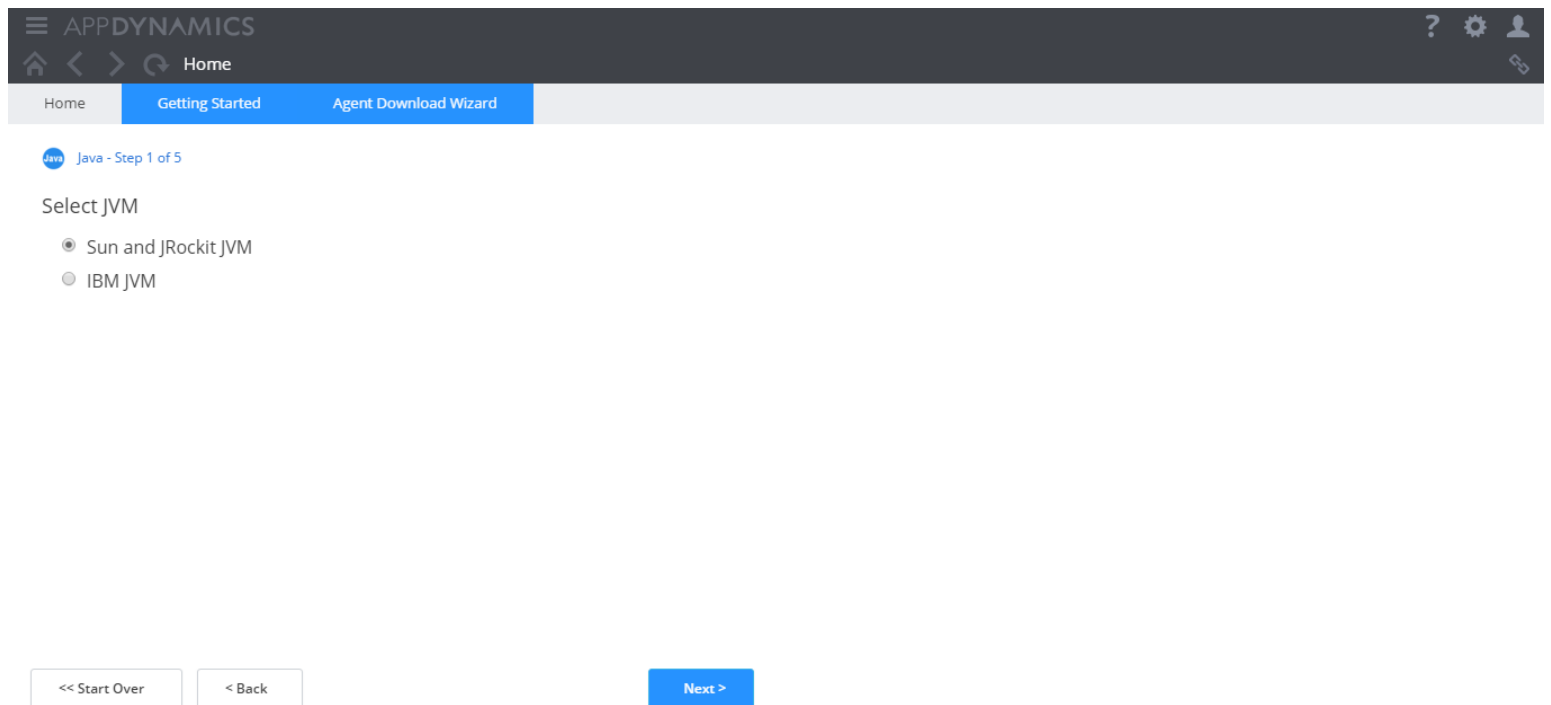
Installing and Configuring a Java agent is pretty straight forward and simple. You can do this two ways: Use the 'Agent download Wizard' from the controller UI (recommended if this is the first agent you are configuring), or simply download the agent software from <https://download.AppDynamics.com>.

Using the Agent download Wizard

From the Controller UI, click on the 'Agent Download Tab'. Choose 'Java' as the Agent Type



Select JVM



The controller Information is auto populated (this is the advantage of using the wizard)

The screenshot shows the 'Agent Download Wizard' interface for Java. The breadcrumb trail includes 'Home', 'Getting Started', and 'Agent Download Wizard'. The current step is 'Java - Step 2 of 5', titled 'Specify Controller URL'. The instruction is 'Set the controller host and port:'. A text input field contains 'http://myServer|' and a port input field contains '8090'. There is an unchecked checkbox for 'Use SSL'. A 'Next >' button is highlighted in blue. A help box on the right explains that the agent sends monitoring data to the Controller and provides a link to 'more details' for changing controller properties.

Choose or create a new Application

The screenshot shows the 'Agent Download Wizard' interface for Java. The breadcrumb trail includes 'Home', 'Getting Started', and 'Agent Download Wizard'. The current step is 'Java - Step 3 of 5', titled 'Which Application should the agent report to?'. There are two radio button options: 'Use an existing Application:' with a dropdown menu showing 'Select existing Application', and 'Create a new Application:' with a text input field containing 'AcmeShoppingPortal|'. The 'Create a new Application:' option is selected. A 'Next >' button is highlighted in blue. A help box on the right explains that a Business Application models all components or modules in an application environment and provides a link to 'Mapping Applications in AppDynamics'.

Choose or create Tier

Your agent is now ready for download

Once you download the Agent (zip or tar file), take the file to the Server you need to instrument and unzip/untar the file. Now all you have to do is configure your java application to load the agent when it

starts. The only required configuration is done at the java command line that starts the application.

Simply add the following parameter

```
-javaagent:AGENT_HOME/javaagent.jar
```

The Agent log file is at <AGENT_HOME>/logs/<Node name>/agent<date>.log

Using AppDynamics download site

Instead of using the wizard, you may choose to download the agent from AppDynamics website and manually configure the agent. You can download the agent from <https://download.AppDynamics.com>. Once you unzip/untar the agent on the server to be instrumented, all you have to do is update controller-info.xml and add the java command line parameter to the startup script of your application.

Configure controller-info.xml

This file is found at <AGENT_HOME>/conf/controller-info.xml. Here are the important parameters to be configured

```
<controller-host>myServer</controller-host>
```

```
<controller-port>8090</controller-port>
```

```
<application-name>AcmeShoppingPortal</application-name>
```

```
<tier-name>OrderFulfillment</tier-name>
```

```
<account-name>***</account-name>
```

```
<account-access-key>***</account-access-key>
```

If you don't know the account name and access key, check the license page at the controller UI (You can access the license page by click on the 'Settings' button and choosing 'License').

Add the following parameter to your Application startup script and restart your application.

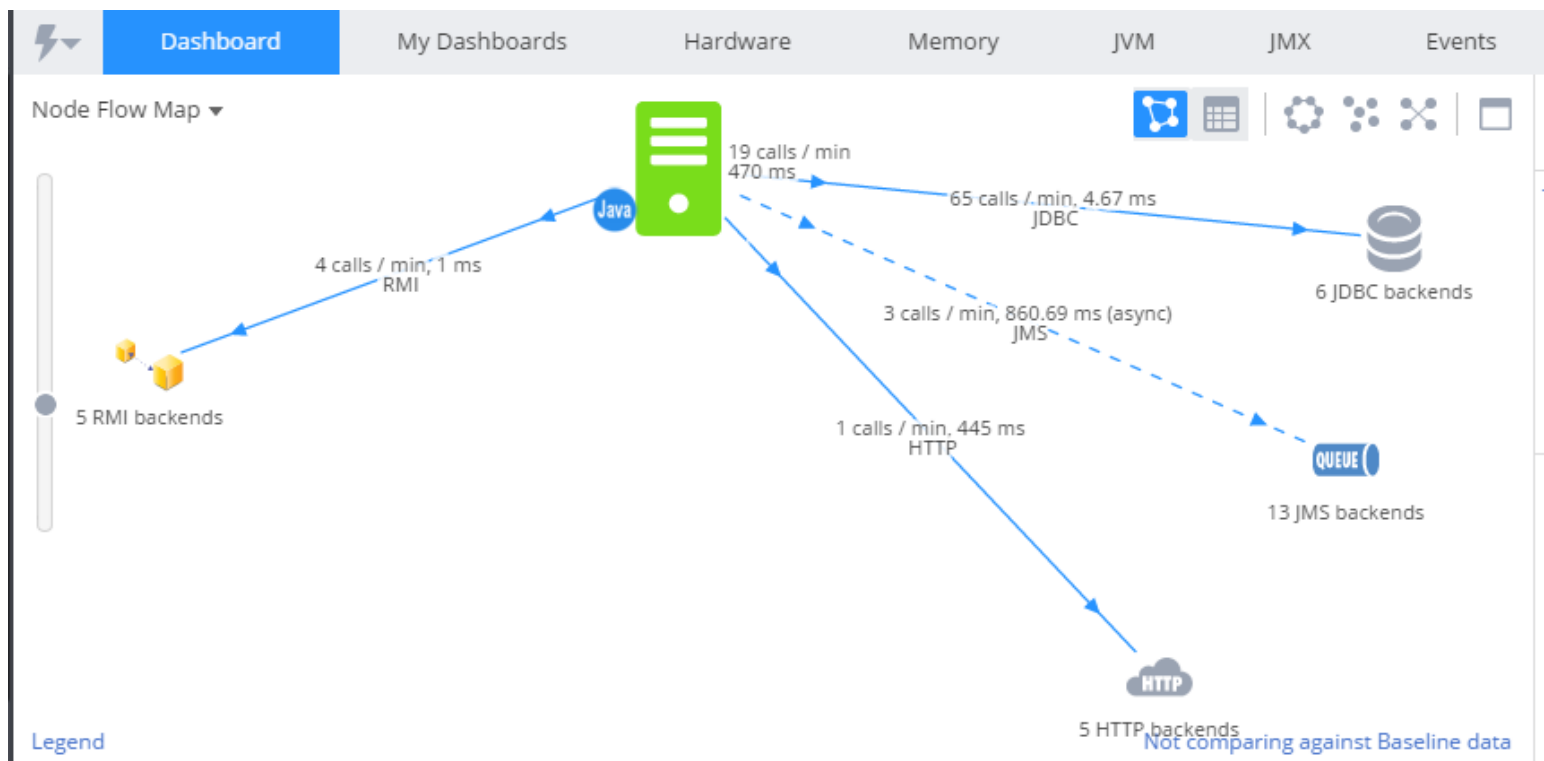
```
-javaagent:AGENT_HOME/javaagent.jar
```

The Agent log file is at <AGENT_HOME>/logs/<Node name>/agent<date>.log

Look inside your Java Application, like never before

As soon as you have your Java application reporting to Controller, you can learn a lot about your application. Some of the things you learn about your application may even be surprising to you. AppDynamics automatically creates a flow map of your application. It discovers backends and plots it for you.

Login to the Controller UI (typically <http://<controllerHost>:8080>). Navigate to your Application and Tier if required. A typical application flow may look like the following:



From this point, you can drill down to any of the components to get deeper level metrics.

AppDynamics automatically captures 'slow response times' and presents them in 'Slow response times' Dashboard from the home page of the Application. Slow response times include remote service calls and jdbc calls. In addition AppDynamics captures any application 'errors' such as a Java Exception or a HTTP error code and presents them in its own dashboard.

A tremendous feature of AppDynamics is the Transaction snapshot. With a Transaction snapshot, you can dismantle the transaction down to the code level and see the response time in various subsystems.

For example, you can clearly see if a transaction is slow because of a DB call or because of a remote web service call (or because of JVM time itself, which may indicate Garbage Collection or other Memory issues)

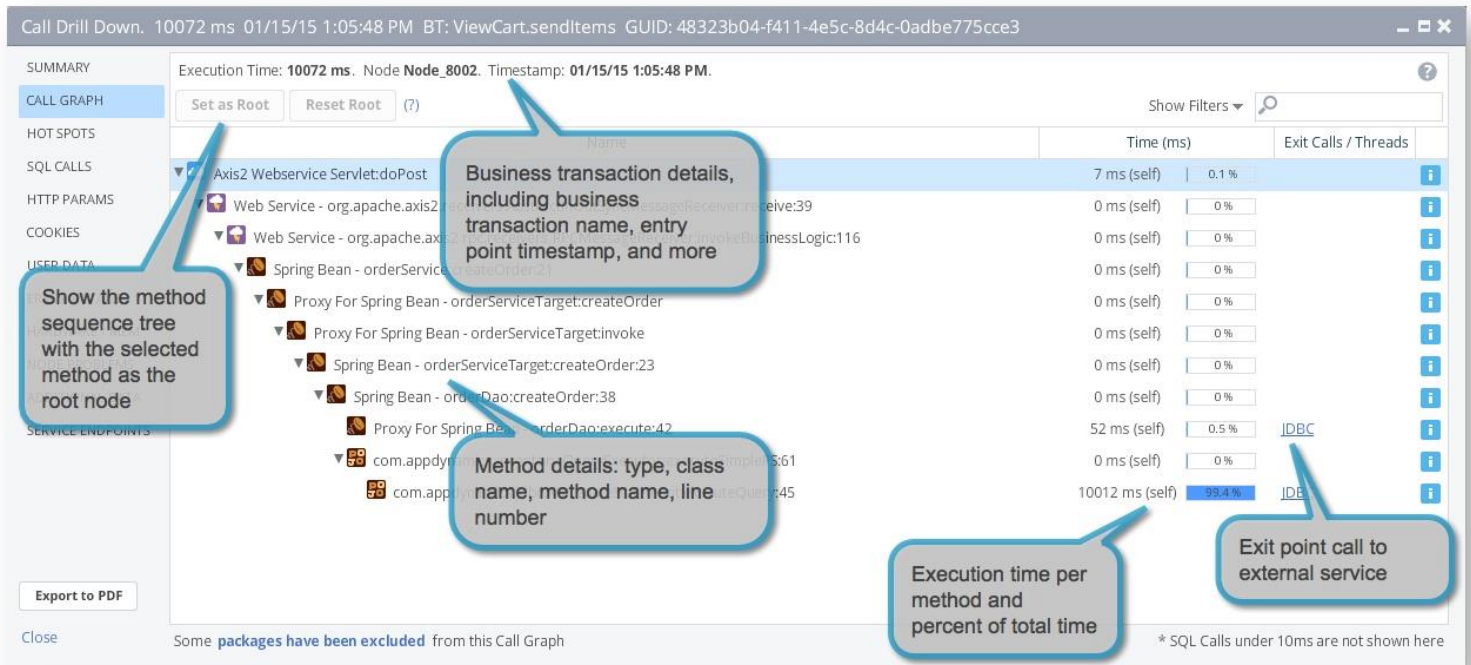


Image Source: www.appdynamics.com

Get alerted before things get worse

AppDynamics automatically learns the application behavior by determining the baseline value for any metric. When the current value deviates from baseline over a certain limit, it triggers an alert. The deviation is configurable.

AppDynamics implements alerting using two elements: Health rules and Policies. Health rule is where you will define the performance metric to monitor and the threshold to watch for. Policy tells AppDynamics what to do when Health rules are violated. The action could be emailing or paging someone (or a group), or even executing a program.

AppDynamics come prebuilt with several Default Health Rules. An example is JVM memory utilization (Heap). You can update the thresholds for these default health rules.

In addition you can create an Email Digest that includes a summary of events over a certain period. This can be useful item for senior management.

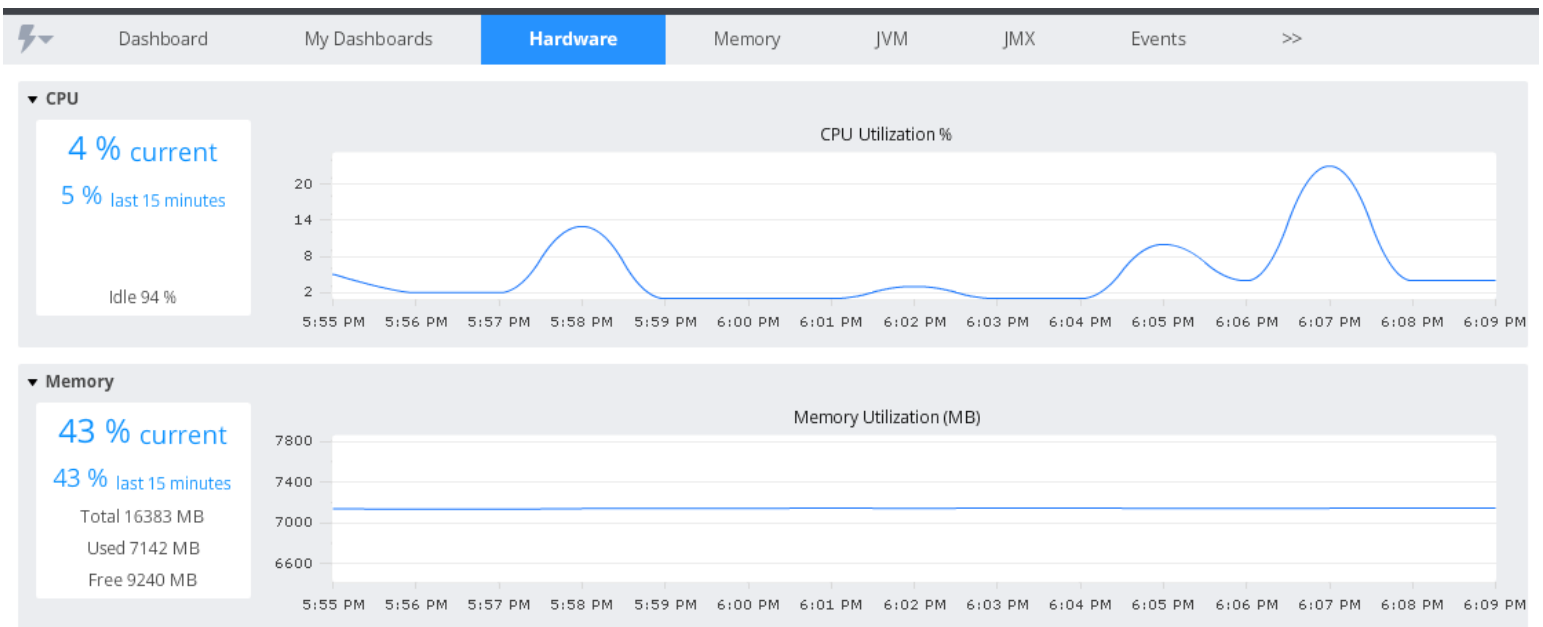
©Karun Subramanian



Image Source: www.appdynamics.com

Monitor your hardware too

With AppDynamics Machine Agent, you can monitor basic metrics such as CPU utilization, Memory Utilization etc for most of the popular Operating Systems.



Some default Metrics monitored:

CPU Metrics

Disks Metrics

Memory Metrics

Network Metrics

System Metrics

In addition to monitoring the hardware, Machine Agent lets you write your own monitoring programs and send the data to AppDynamics Controller. This is a very powerful feature as it virtually allows you to monitor anything. These programs are called extensions and there are tons of extensions already available for download at <https://www.appdynamics.com/community/exchange/>

Where to go next?

Congratulations on learning what AppDynamics can do for your organization. I have just covered the tip of iceberg. What you can do with AppDynamics is only up to you. You can take your monitoring to an entirely different level by leveraging all the features that AppDynamics offers.

I recommend downloading and trying out your own AppDynamics Controller. Install a Java Agent on one of your test environments and play with it for a while. Start at <https://download.appdynamics.com>. For documentation, start at <https://docs.appdynamics.com>. There is also a community at <http://www.appdynamics.com/community/>

AppDynamics offers tons of webinars. Check out the events calendar at <http://www.appdynamics.com/community/events/>

Happy Monitoring !!!