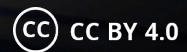




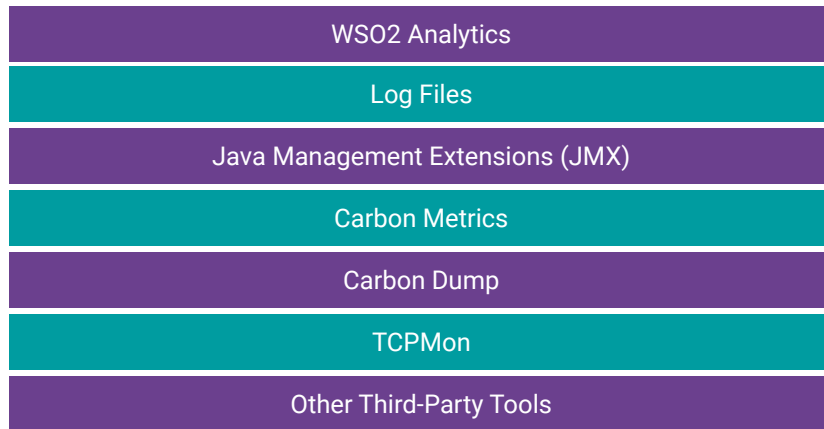
WSO2 Product Administration

WSO2 Training

Module 05 - Monitoring Carbon Products



Alternatives to Monitoring Carbon



Carbon products expose a multitude of features to allow them being monitored.

Memory and processing power usage information describe how memory and resources are allocated among the processes.

Java Thread information can be used to identify the active threads at particular point in time and what each thread is involved with.

These information is specifically useful when troubleshooting load related issues.

Tools that comply with Java Management Extensions (JMX) specification are commonly used in monitoring these information.

Another crucial factor is monitoring the errors logs on server logs, backend servers and on the client side. These logs help troubleshooting and isolating issues to a great extent.

Gathering and monitoring these operational data is important to ensure stability of a deployment.

WSO2 Analytics

Custom Dashboards for

WSO2 API Manager

WSO2 Identity Server

WSO2 Enterprise Integrator

All products can **push** business and **operation stats to WSO2 Analytics**

OOTB summarization queries to get information out of raw data

Can connect with **multiple protocols**



WSO2 Analytics is a solution for aggregating, analyzing and presenting business activity related information.

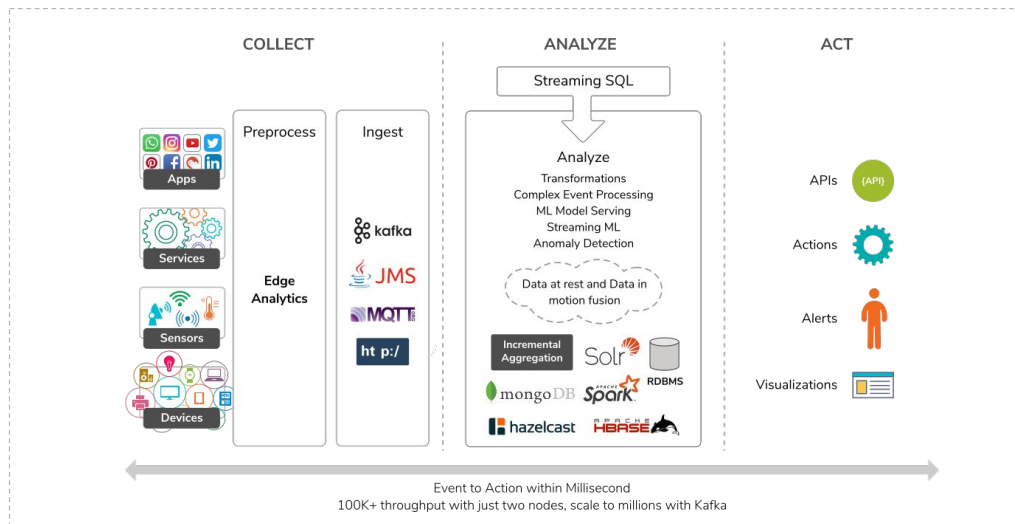
It enables real time visibility for complex deployments.

Any product can push statistics to WSO2 Analytics. WSO2 Analytics is capable of summarizing or storing the raw data depending on the requirement.

WSO2 Analytics is also packaged with out of the box summarization queries to summarize raw data that is published by servers on WSO2 Analytics.

It is also capable of communicating in several different protocols including HTTP, HTTPS.

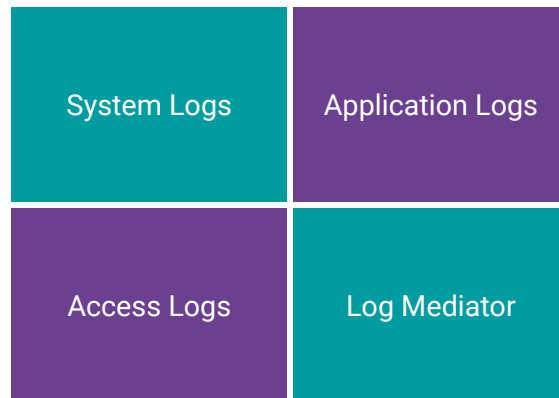
Analytics Server Architecture



WSO2 SP contains Siddhi as its core to collect, analyze and act on the incoming events.

<https://docs.wso2.com/display/SP440/Architecture>

Logging



<https://docs.wso2.com/display/admin44x/Monitoring+Logs>



Logging is one the most important and commonly used monitoring tool of a production server.

Logging enables identifying errors, security threads and usage patterns.WSO2 products use a log4j based logging mechanism based on the Apache Common Logging facade library.

Using these logs you can monitor server activities ,wire logs pertaining to incoming and outgoing messages of the server, HTTP access logs , and stack traces of various errors that occur during server runtime.

These logs are very helpful in debugging and isolating issues and identifying their root causes.

There are 3 main types of logs available on Carbon servers.

Application logs, log events invoked by web applications hosted on the carbon server and allow user to view logs of a selected web application.

System logs display log events pertaining to the entire system.

Access logs indicate service/REST API invocations and management console access related logs.

In addition to the above the log mediator can be engaged to monitor and log messages mediated within sequences.

Using Log Files

CARBON_HOME/repository/logs

Configuring products for logging
<PRODUCT_HOME>/repository/conf/log4j.properties

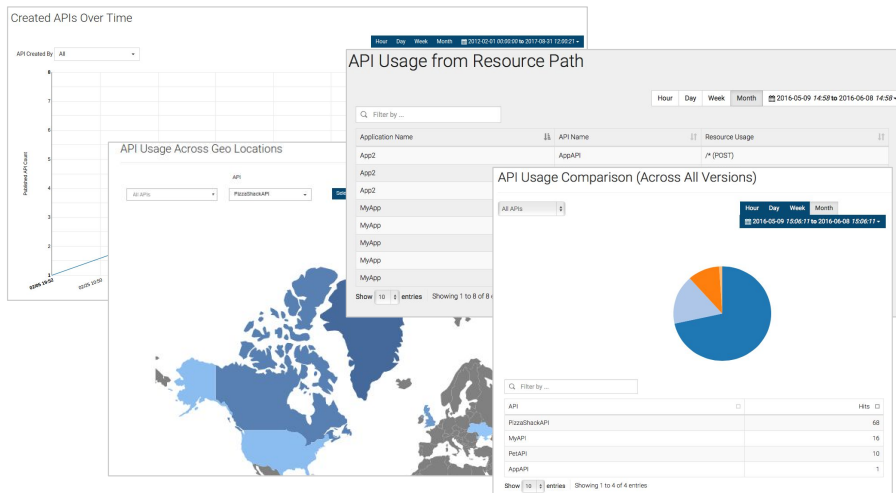
<https://docs.wso2.com/display/admin44x/Monitoring+Logs>



Carbon_HOME/repository/conf/log4j.properties file governs logging on the server, and it can be used to enable/disable various loggers based on the information that need to be captured in carbon logs.

Log files are located in CARBON_HOME/repository/logs/ directory and can be viewed on the command line terminal and the management console.

API M Analytics



WSO₂

Several examples of usage and performance statistics are given below:

Created APIs Over Time - given above

API Usage

API Last Access Times

Usage by Resource Path

Usage by Destination

API Usage Comparison

API Throttled Requests

Faulty Invocations

API Latency Time

API Usage Across Geo Locations

API Usage Across User Agent

App Throttled Requests

Applications Created Over Time

API Subscriptions

Developer Signups Over Time

Subscriptions Created Over Time

API Usage per Application

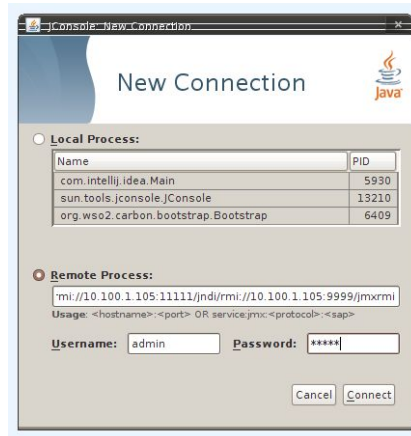
Top Users per Application

Resource Usage per Application

Faulty Invocations per Application

Availability of APIs

Monitoring Using JMX



<https://docs.wso2.com/display/ADMIN44x/JMX-Based+Monitoring>



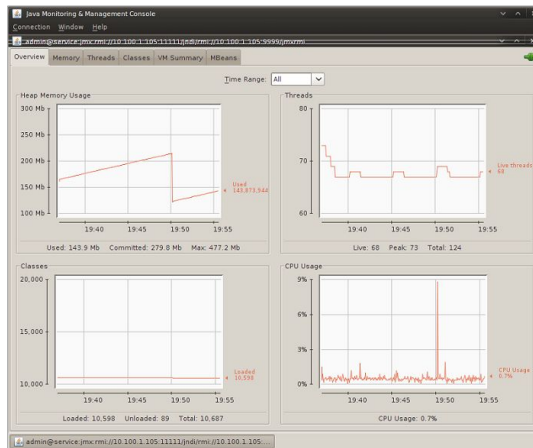
Java Management Extensions (JMX) allow implementing management interfaces for Java applications. WSO2 Servers expose several management resources in the form of JMX MBeans that can be accessed through JMX clients in order to help monitor and manage the server externally.

JMX clients such as Jconsole or VisualVM can be used to monitor server memory usage, threads and many more aspects.

JMX configurations are available in the `<PRODUCT_HOME>/repository/conf/etc/jmx.xml` and `<PRODUCT_HOME>/repository/conf/carbon.xml` files. These configurations can be altered when required.

You can connect to JMX clients by using the JMXServerManager JMX Service URL of the Carbon server.

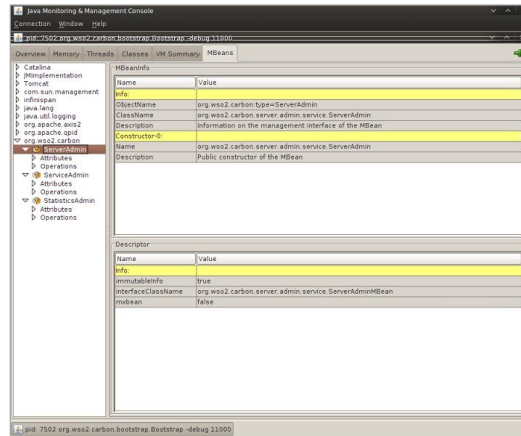
Monitoring Using JMX



<https://docs.wso2.com/display/ADMIN44x/JMX-Based+Monitoring>

The slide shows how memory usage, thread, classes and CPU usage graphs are depicted on Jconsole. You can monitor these information real-time.

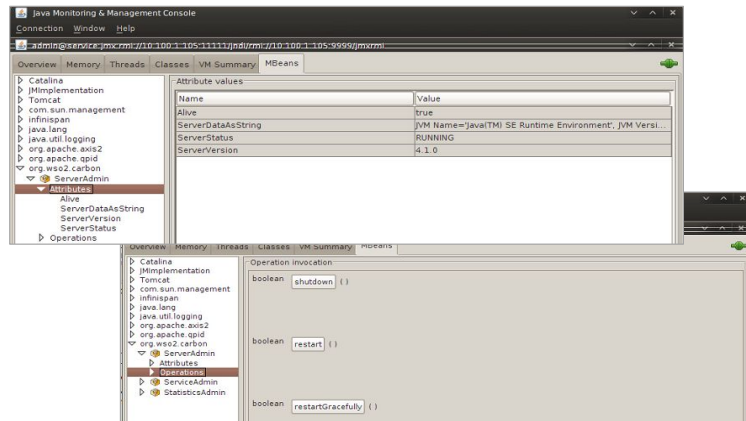
Monitoring Using JMX



<https://docs.wso2.com/display/ADMIN44x/JMX-Based+Monitoring>

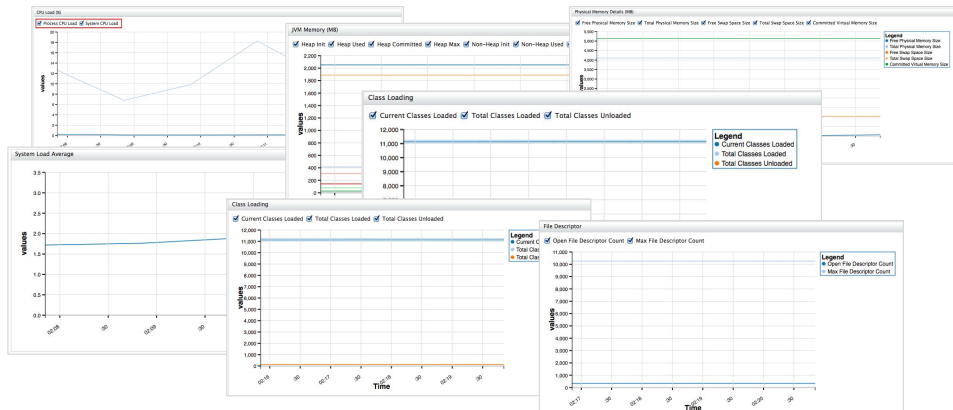
MBeans tab lists and allows to monitor and manage the server's Mbeans. All the platform MXBeans and their various operations and attributes are accessible via JConsole's MBeans tab.

Monitoring Using JMX



<https://docs.wso2.com/display/ADMIN44x/JMX-Based+Monitoring>

Carbon Metrics



<https://docs.wso2.com/display/ADMIN44x/Monitoring+with+WSO2+Carbon+Metrics>



WSO2 products based on Carbon 4.4.x Kernel versions are shipped with JVM Metrics which is a form of Java Metrics.

The Java Metrics library contains metrics that are used for monitoring, thereby allowing monitoring of statistics related to performance of Carbon Servers. These are exposed via the WSO2 Carbon Metrics API in WSO2 products.

Carbon Dump

```
sh carbondump.sh [-carbonHome path] [-pid of the carbon instance]
```

<https://docs.wso2.com/display/ADMIN44x/Capturing+System+Data+in+Error+Situations>

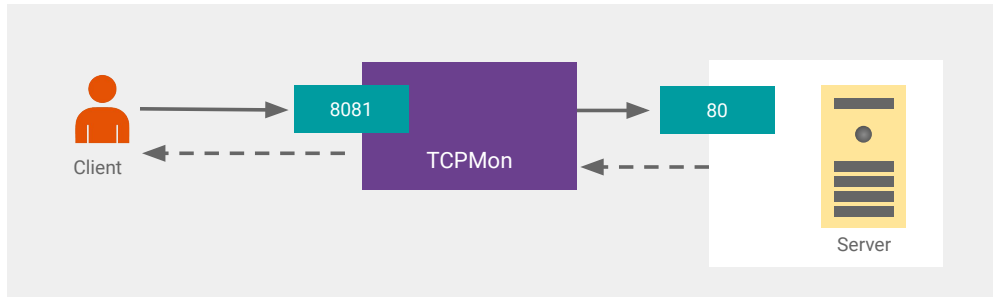


Carbondump is a tool that can be used to collect data from a carbon instance at runtime during the occurrence of an error.

This tool captures operating system information such as installed modules and running tasks, Memory information related to the Java process such as Objects waiting for finalization, Java heap summary, Garbage collection algorithms used, Carbon instance information such as log files, Database files, Thread Dumps and checksum values of files contained in the \$Carbon_home.

These information help analyze the server to determine root causes of issues. You can run this tool by providing the process ID of the running Carbon instance as given on the slide.

TCPMon Monitoring



<https://docs.wso2.com/display/ADMIN44x/Monitoring+TCP-Based+Messages>

TCPMon is a tool that can be used to intercept TCP based communication between client and backend servers.

This is a useful tool that helps debugging issues by allowing to monitor and verify the integrity of messages passing between client and server.

You can see 2 connections in the diagram in between client and server.

1. Request from client to Server.
2. Response from Server to client.

The request happens between the client and Port 80 of the server.

When TCPMon is placed in between the client and server, it intercepts the call by listening to the request at port 8081 and captures the request.

Then then this request is forwarded to the port 80 of the server.

The response coming from the backend server is captured by TCPMon at port 8081 and forwarded back to the client.

Other Third-Party Tools

- Nagios
- Dynatrace
- AppDynamics
- Prometheus



<https://medium.com/@dilanasanga/monitor-wso2-apps-jvm-usage-with-nagios-c12f748aa70c>

<https://medium.com/@godwinamila/configure-dynatrace-jmx-plugin-to-collect-stats-from-wso2-products-27b3f267b706>

<https://medium.com/@thejanrupasinghe/wso2-enterprise-integrator-performance-monitoring-with-appdynamics-141c5d65fd88>

<https://medium.com/@wso2tech/monitor-wso2-products-using-appdynamics-8faf72e83a7>

<https://medium.com/@lashan/monitoring-wso2-products-with-prometheus-4ace34759901>

THANK YOU

ws02.com

