



# Overview of Infrastructure Visibility

## On this page:

- [Server Visibility Metrics](#)
- [Basic Machine Metrics](#)
- [Infrastructure Visibility Strategies](#)

## Related pages:

- [Infrastructure Visibility](#)
- [Standalone Machine Agent](#)
- [Server Visibility](#)

Many times the root cause of application issues is most obvious by looking at application, server, and machine metrics that measure infrastructure utilization. For example, the following infrastructure issues can slow down your application:

- Too much time spent in garbage collection of temporary objects (application metric)
- Inefficient processes that result in high CPU utilization (server metric)
- Excessively high rates of reads/writes on a specific disk or partition (hardware metric)

[Infrastructure Visibility](#) enables you to isolate, identify, and troubleshoot these types of issues. Infrastructure Visibility is based on a machine agent that runs with an app server agent on the same machine. These two agents provide multi-layer monitoring as follows:

1. The app server agent collects metrics about applications and identifies applications, tiers, and nodes with slow transactions, stalled transactions, and other application-performance issues.
2. The machine agent collects metrics at two levels:
  - a. Server Visibility metrics about local processes, services, and resource utilization.
  - b. Basic machine metrics about disks, memory, CPU, and network interfaces.

This multi-layer monitoring enables you to find possible correlations between application issues and service, process, hardware, network, or other issues on the machine.

	This Agent Monitors...	Example Metrics
App Agent 	apps app servers JVMs CLR's	Slow Transactions Stalled Transactions Response Times Wait Times Block Times Error %
Server Visibility	processes services caching swapping paging queueing	Hardware/Software Interrupts Virtual memory/swapping Process faults CPU/disk/memory utilization by process
Machine Agent 	Hardware/OS	
	disks volumes partitions memory CPU network interfaces	CPU busy times Memory utilization Disk reads/writes JVM crashes

## Server Visibility Metrics

[Server Visibility](#) monitors local processes, services, and resource utilization. You can use these metrics to identify time windows when problematic application performance correlates with problematic server performance on one or more nodes.

Server Visibility is an add-on module to the [Standalone Machine Agent](#). With Server Visibility enabled, the machine agent provides the following additional functionality:

- Extended hardware metrics such as machine availability, disk/CPU/virtual-memory utilization, and process page faults
- Monitor internal or external HTTP and HTTPS services
- Support for grouping servers so you can apply health rules to specific server groups
- Support for defining alerts that trigger when certain conditions are met or exceeded based on monitored server hardware metrics

## Basic Machine Metrics

The Standalone Machine Agent collects basic hardware metrics from the server's OS. This agent provides the following functionality:

- Basic hardware metrics from the server's OS such as CPU and memory utilization, throughput on network interfaces, and disk and network I/O
- Support for creating extensions to generate custom metrics
- Support for running remediation scripts to automate your runbook procedures. You can optionally configure the remediation action to require human approval before the script is started.
- JVM Crash Guard for monitoring JVM crashes and optionally running remediation scripts

## Java and .NET Infrastructure Monitoring

Infrastructure Visibility uses different agents to monitor Java and .NET environments:

- The Java Agent collects metrics for business applications and JVMs. The Standalone Machine Agent collects Server Visibility and hardware/OS metrics.
- The .NET Agent collects metrics for business applications and instrumented CLR. The .NET Agent includes a .NET Machine Agent that collects IIS and hardware/OS metrics (see [Monitor Windows Hardware Resources](#)). The Standalone Machine Agent collects Server Visibility metrics.

Java Monitoring		.NET Mon		
Java Agent		Applications JVMs	.NET Agent	
Standalone Machine Agent		Server Visibility	Standalone Machine Agent	
		Hardware/OS	.NET Machine Agent	

## Infrastructure Visibility Strategies

You can use the following strategies to find infrastructure issues that affect application performance:

- Transaction snapshots for slow or stalled transactions – You can use snapshots to correlate infrastructure metrics for the specific node so that you can identify the root cause of slow or stalled transactions.
- Metric correlation – One example workflow is to open the Node Dashboard for a mission-critical server with a machine agent installed and then to cross-compare data in the following tabs:
  - JVM (application performance)
  - JMX (server performance)
  - Server (hardware resource consumption)
- Health rules – You can configure health rules on metrics such as garbage collection time, connection pool contention, or CPU usage to catch issues early in the cycle before there is an impact on your business transactions.
- Infrastructure rules, policies, and alerts – You can
  - Create health rules on metrics such as garbage collection time, connection pool contention, or CPU usage to catch issues early in the cycle before there is an impact on your business transactions.
  - Define [policies](#) that trigger [actions](#) (such send an email, start diagnostics, or perform a thread dump) when Infrastructure metrics report a critical level.
  - You can configure alerts for JVM and CLR crashes respectively using [JVM Crash Guard](#) and the [.NET Machine Agent](#).
  - Configure the agent to [run scripts](#) in response to critical events (for example, restart an application or JVM in response to a crash).

With the right monitoring strategy in place, you can be alerted to problems and fix them before user transactions are affected.