IBM Sametime on WAS Level

Introduction

Several kbase articles concerning monitoring Sametime are already available. The only missing part till now was, what IBM Sametime things you can check /monitor on the WebSphere Level itself.

This article should give you some sort of idea what you can do with GreenLight

Configuration

First of all please verify that you have the core setup foor WebSphere monitoring in place

for v3.2 appliance

WebSphere for GL v3.2 - setup

for v3.5 appliance

WebSphere for GL v3.5 - setup

1) Download Sametime Shell Script Sensors

IBM Sametime:

Download and Extract the files to */opt/panagenda/scripts/gl_sensor*

Download Sametime Sensors

2) Create Linux Shell Script Sensor - GreenLight UI

-create a *LinuxShellScript* Sensor

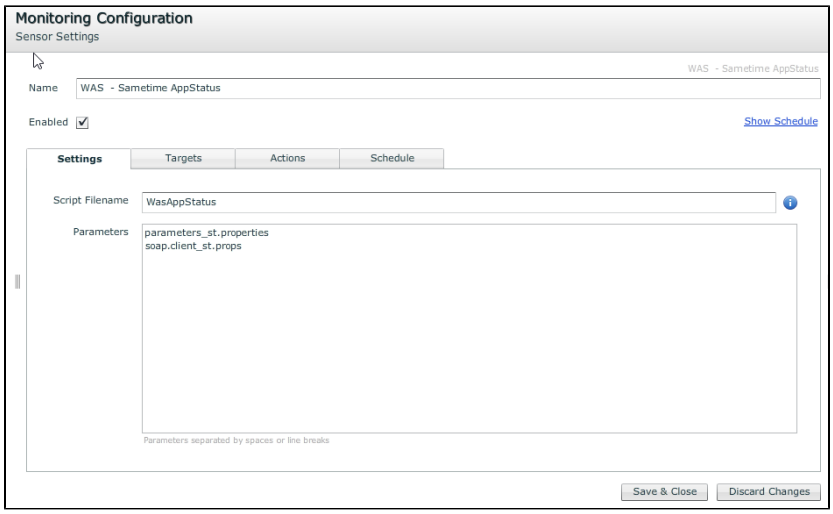
-Script Filename: WasAppStatus

This is one of the files which you have copied over to */opt/panagenda/scripts/gl_sensor* in Step1

-Parameters:

parameters_st.properties

soap.client_st.props



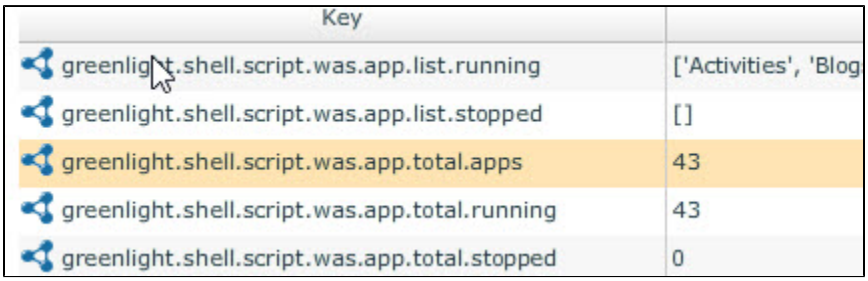
-Define on Targetlevel your Sametime FQDN (e.g. MeetingServer Url, ..)

-save/close

Output

You are going to get for ANY installed application within the WAS Cell the Status (running, stopped, not available)

example:

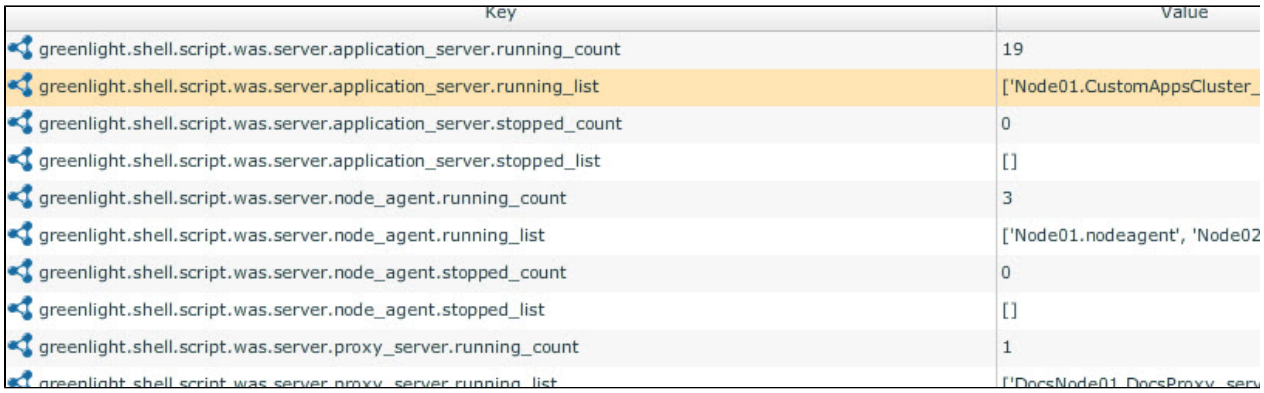


What other Scripts are available?

You can do now the same steps for the remaining Scripts

- *WasServerStatus.sh (was_serverstatus.sh)*

  Monitors each WAS server



Key	Value
greenlight.shell.script.was.server.application_server.running_count	19
greenlight.shell.script.was.server.application_server.running_list	['Node01.CustomAppsCluster_
greenlight.shell.script.was.server.application_server.stopped_count	0
greenlight.shell.script.was.server.application_server.stopped_list	[]
greenlight.shell.script.was.server.node_agent.running_count	3
greenlight.shell.script.was.server.node_agent.running_list	['Node01.nodeagent', 'Node02
greenlight.shell.script.was.server.node_agent.stopped_count	0
greenlight.shell.script.was.server.node_agent.stopped_list	[]
greenlight.shell.script.was.server.proxy_server.running_count	1
greenlight.shell.script.was.server.proxy_server.running_list	['DocsNode01.DocsProxy_serv

- WasDataSource.sh (was_datasource.sh)

  Monitors WAS Datasource Connectivity (e.g. is WAS able to connect to the DB2 Datasource of Connections?

- WebSrvStatus (was_websrvstatus.sh)

  Monitors all Webservers within the Cell (stopped, running)

- Messagebus.sh (was_messagebus.sh)

  Retrieves pending messagebus information from any queue!

- *WasJVMHeap.sh (was_jvmheap.sh)*

  Retrieves JVM Information such as Xmx, Xms, etc...

- *WasPMIConnectionpool.sh (was_pmi_connectionpool.sh)*

  Retrieves PMI Information of JVM processes and Webpools (ActiveCount, Poolsize,...)

- *WasPMIPerf.sh*

  Retrieves PMI Information of JVM Processes (Used Memory, Free Memory,. ...)

- *WasVariables.sh*

  Lists all WAS Variables