

Paper 309-2010

The Art of Managing the SAS 9.2 Middle Tier

Saravana R. Chandran, SAS Institute Inc., Cary, NC

ABSTRACT

SAS 9.2 supports three Web application servers: Red Hat JBoss, Oracle WebLogic, and IBM WebSphere. Every application server deployment and configuration can vary, but the life cycle of the SAS 9.2 middle tier application in the Web application server environment is the same. The intent of this paper is to provide various best practices for the SAS 9.2 middle tier in the context of its life cycle in a Web application server environment. The paper provides information about deployment, configuration, tuning, administering, monitoring, troubleshooting, and migrating from one environment to another all in the context of the SAS 9.2 middle tier application. It will also discuss the various best practices to improve the operating efficiencies of the SAS 9.2 middle tier application in the Web application server environments.

INTRODUCTION

SAS 9.2 consist several Web applications, each one of them delivered as an EAR (Enterprise Archive) file that consists of one more WAR (Web Application Archive) files. SAS 9.2 supports three major Web application servers: Red Hat JBoss, Oracle WebLogic, and IBM WebSphere in both 32-and 64-bit architectures. A detailed SAS 9.2 support matrix on platforms and Web application servers can be found in the [SAS 9.2 Support for Web Application Servers and HTTP Servers](#)

The intent of this paper is to explore various areas of the SAS 9.2 middle tier life cycle in detail and provide insights that can improve the operational efficiencies on managing the SAS 9.2 middle tier. The paper explores these areas:

- configuration and management
- SAS 9.2 middle tier dependencies
- deployment topology
- tuning
- monitoring
- migration
- multicast
- clustering

CONFIGURATION AND MANAGEMENT

In order to install and configure SAS 9.2 you use the SAS Deployment Wizard. By default, the SAS Deployment Wizard will build the SAS 9.2 Web application, configure the Web application server that you intend to use, and automatically deploy the SAS 9.2 Web applications. As part of the SAS Deployment Wizard configuration process, it will create the SAS servers (SASServer), configure resources that are required for the SASServers (such as JDBC, JMS Server, JMS Queue, Topic and Connection Factories, and Mail) and deploy the SAS 9.2 Web applications to the respective servers.

By default, for EBI deployment, the SAS Deployment Wizard provides the choice to create either one or two servers. For large performance and scalability requirements you might require more servers to distribute the SAS 9.2 Web applications or to scale the SAS 9.2 Web applications into horizontal or vertical clusters. In both WebLogic and WebSphere, you can create servers using their Admin Console or their Jython Scripting (In WebSphere – wsadmin and WebLogic – WLST) mechanism. In JBoss, you configure the changes by manually editing and copying the configuration file.

SAS 9.2 CONFIG FILE STRUCTURE

The SAS 9.2 middle tier configuration directory contains several configuration files related to SAS Web applications. The following figure shows the directory structure of the SAS 9.2 Config directory.

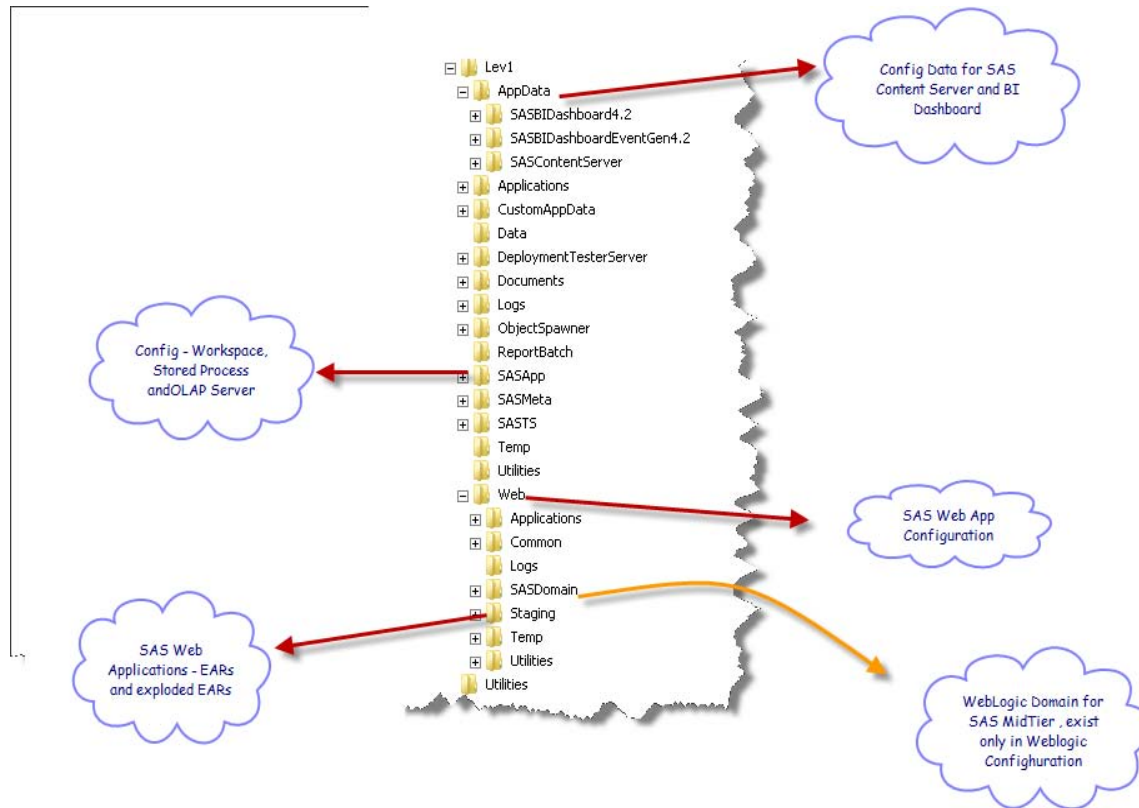


Figure 1. SAS 9.2 Config File Structure

The **AppData** directory contains configuration for both SAS BI Dashboard and SAS Content Server. The SAS Content Server repository and its configuration reside in the **SASContentServer** folder.

The **Web** directory contains all configurations for remote services, logging configurations, EAR files, and other utilities. The **Staging** directory contains the EAR files and exploded EAR files. The **Application** directory contains configuration for remote services. In a WebLogic configuration, the **SASDomain** directory contains domain configurations for SAS Domain.

In JBoss configurations, separate SASServer(s) are created under the **server** directory of the JBoss home directory. The **conf** directory under the SASServer(s) contains respective server configuration. The **deploy_sas** directory contains all the SAS 9.2 Web applications (EARs or exploded) as shown in the below figure.

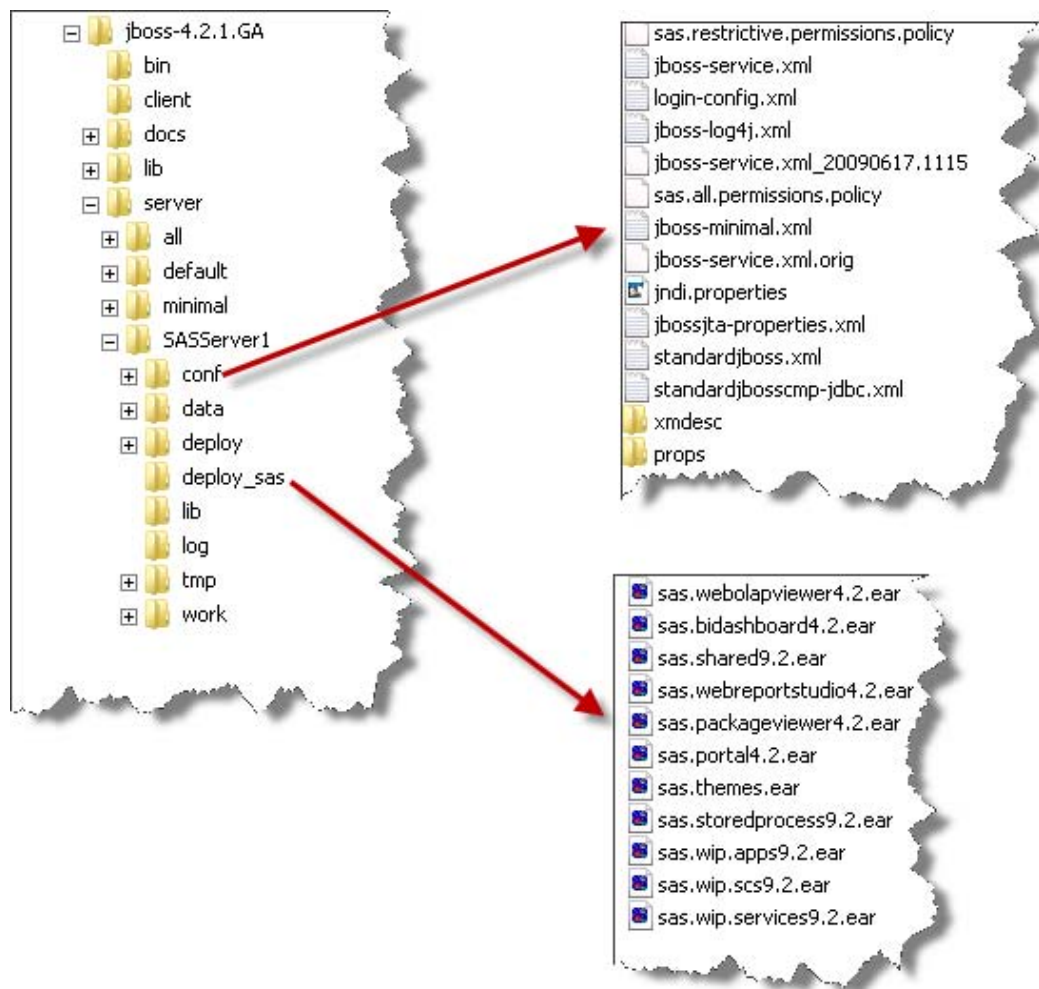


Figure 2. JBoss Config File Structure

In WebLogic, SAS 9.2 configuration creates **SASDomain** under the **Web** Directory of the SAS **Config** directory. SASDomain is the typical WebLogic Domain configuration with SAS 9.2 specific configured server(s) and customized WebLogic start / stop scripts.

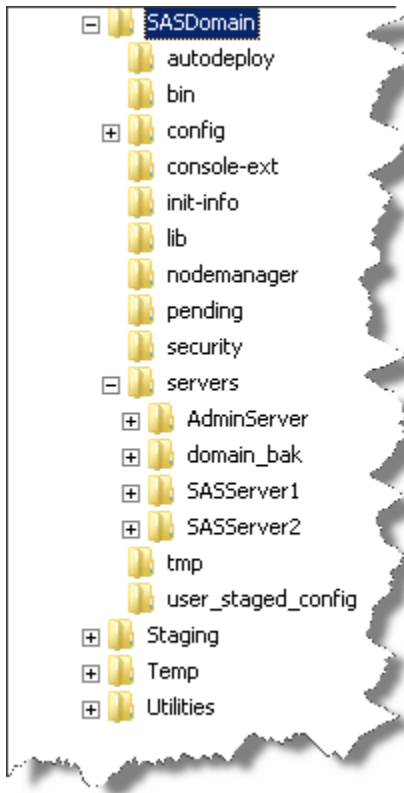


Figure 3. WebLogic SASDomain Config File Structure

In WebSphere the SAS 9.2 configuration creates a SAS 9.2 specific profile that includes dMgr and Server profiles and a server profile federated with dMgr. As part of the SAS Deployment Wizard installation and configuration, you can override the profile name and other options provided in the wizard.

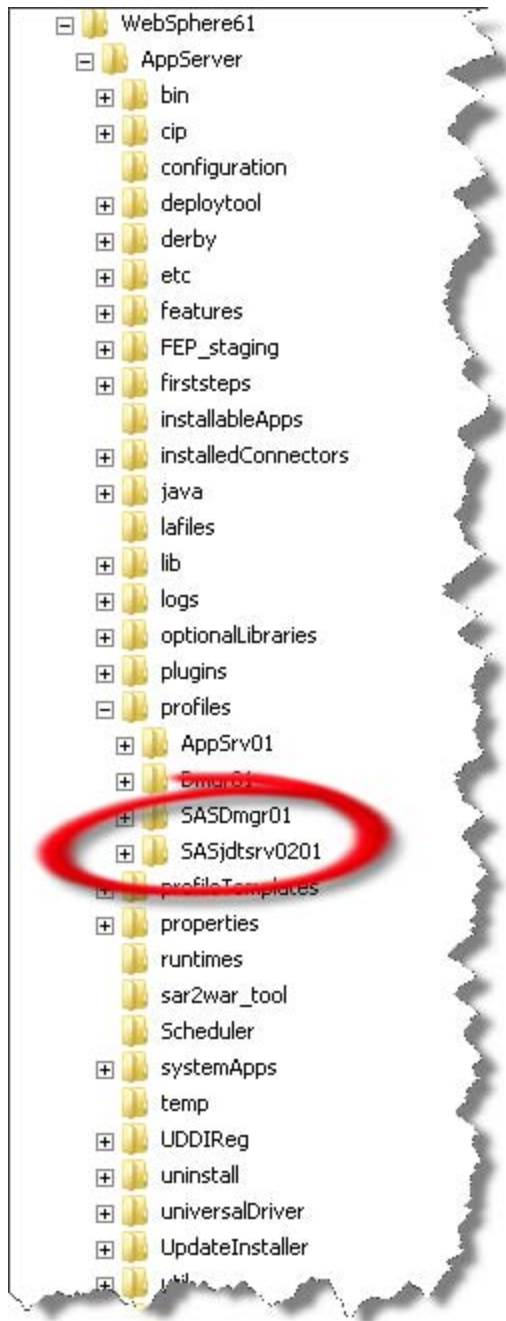


Figure 4. WebSphere SASDomain Config File Structure

Best Practice: The space that is required for a SAS 9.2 EBI deployment is in the range of 6.5 GB for WebSphere (with two profiles created) and 5 GB for WebLogic (for SASDomain) and JBoss (SASServer(s)). More space is required when you cluster vertically, and a log rotation cycle determines the space requirement. For clusters, plan an average of 3 to 4 times more space.

Because SAS Themes contains static files, always deploy them in a Web server or a reverse proxy.

BACKUP AND RESTORE

JBoss: JBoss configuration is stored in XML files, and changes to them are through manual edits. You need to back up the entire file system under JBoss_HOME. In the event of corruption or configuration loss, you can simply replace the entire JBoss file structure with backup.

WebLogic: Oracle WebLogic provides the **pack and unpack** utility to back up a subset or entire domain. This utility can be used as the primary backup and restore facility for the SASDomain. You can use the pack and unpack to restore the configuration on a local or remote machine.

WebSphere: WebSphere provides the **backupConfig** and **restoreConfig** utilities to back up and restore the SAS 9.2 WebSphere profile.

For WebLogic and WebSphere, you can effectively use their backup and restore utilities to migrate from one environment to another.

SAS 9.2 MIDDLE TIER COMMUNICATION PROTOCOLS AND PORTS

HTTP is the communication protocol used between the back-end servers and the front-end clients, such as Web browsers, Java desktop clients, and MS Office / .Net clients. SAS 9.2 Web applications also use HTTP. The SAS Web infrastructure and shared services communicate with the foundation services through **RMI**. The foundation services communicate with back-end metadata, SAS, and SAS STP Servers through **IOM**. The SAS 9.2 middle tier uses multicast technology and communicates through UDP. The following figure shows the communication protocol flow among the SAS middle tier, front-end clients, and back-end servers.

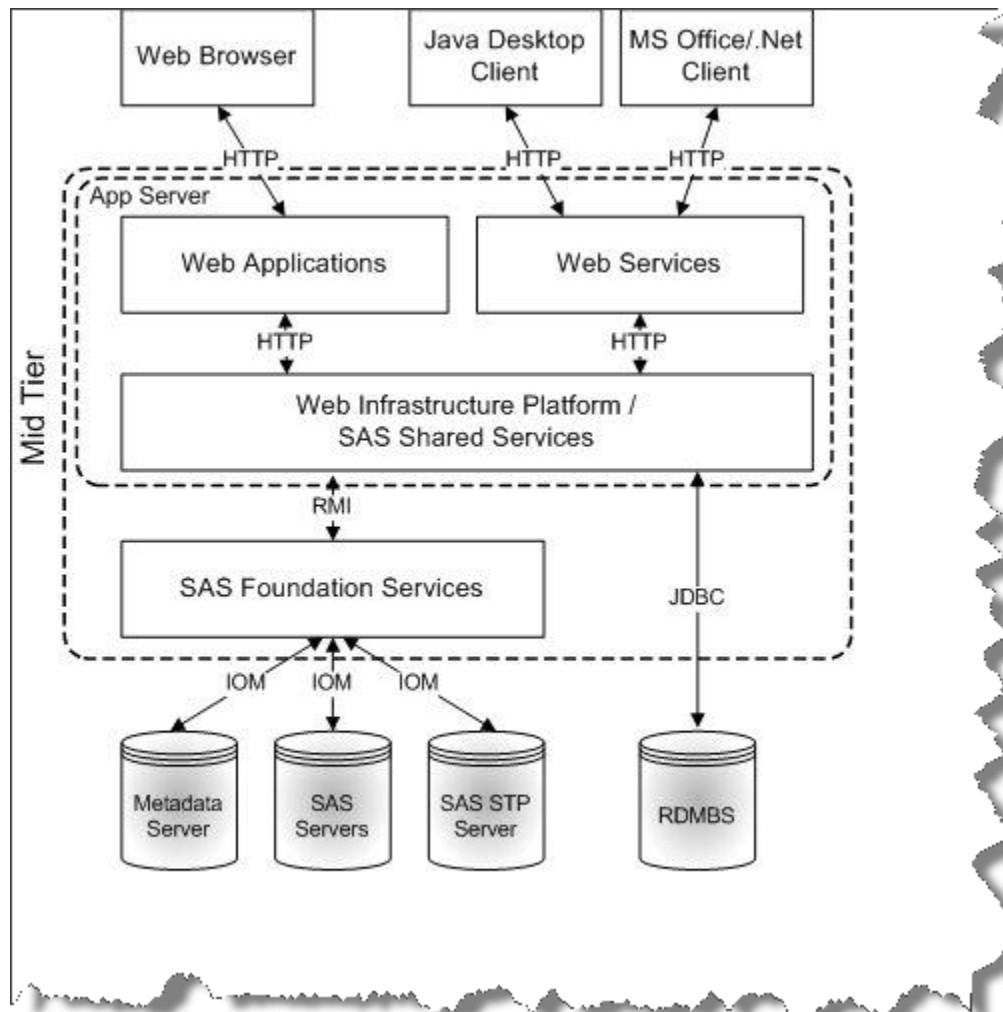


Figure 5. SAS 9.2 Communication protocol flow

During installation, you will have an installation [plan checklist](#) that contains the list of ports or port ranges that will be used. The ports vary depending on the Web application server and the deployment plan that you choose.

For the JBoss default installation, the first SAS 9.2 server starts with 8080 for the HTTP port and increments by 100 for the second server, that is, 8180 for SASServer2. Similarly for HTTPS, the first server starts with 8443, and the second server (SASServer2) is 8543. The RMI port for SASServer1 is 1099, SASServer2 is 1199.

In WebLogic SASServer1 – HTTP = 7001, HTTPS=7002; SASServer2 – HTTP=7101, HTTPS=7102

In WebSphere SASServer1 – HTTP=9080, HTTPS=9443; SASServer2 – HTTP=9081, HTTPS=9444.

SAS Remote Services uses a port in the range of 5090-5099 (the default install uses 5090). The default multicast port is 8561 (UDP)

Best Practice: When configuring the firewall between the layers, make sure that the SAS 9.2 middle tier and respective back-end servers stay in same DMZ. If there is a firewall between the back-end servers and the SAS 9.2 middle tier, make sure the respective ports are open for the back-end server and use the pre-plan checklist as a cross reference.

MULTICAST

The SAS 9.2 middle tier uses multicast. The SAS Web Infrastructure Platform provides several services under the covers such as configuration and authentication. The Web Infrastructure Platform provides load balancing capabilities with those service instances.

The multicast contains two parts: address and port. The SAS Remote Services **environment.properties** contains the value of the multicast address and port. The JVM arguments for each SASServer (Application Server Instance) instance have the “-D” parameter for both address and port.

`-Dmulticast.address=<multicast host address> -Dmulticast.port=<multicast port>`

The multicast address and port must match across the entire middle tier – that is, the SAS Remote Services and all the instances of the SASServer (Application Server Instance) that are deployed with the SAS 9.2 Web applications.

It is very common to have multiple NIC (network interface card) in single machine, listening on two different IP addresses. In those types of machines, you need to use **-Djgroups.bind_addr=<IP address that you want use>** in the SAS Remote Services and all instances of the Web application server that deployed with the SAS 9.2 Web applications.

If the multicast communication fails, the SAS 9.2 Web applications won't function properly. Debugging multicast issues is cumbersome. If you encounter a multicast issue, first check the subnet of each machine that is involved in the SAS 9.2 middle tier and make sure each is in the same subnet.

Next, enable Jgroups logging at the debug level. To do this:

1. Start the **SAS Management Console**.
2. In the **Plugins** window, select **Foundation Services Manager -> Remote Services -> Core -> Logging Service**.
3. Right-click on **Logging Service** and select **Properties**. Select the **Service Configuration** property sheet.
4. Click the **Configuration** button. If org.jgroups is not in the Contexts list, click the **New** button. The new context name will be org.jgroup.
5. Change the Priority to DEBUG. Perform these steps for Remote Services, SASWIPServices9.2 Local Services, SASWIPSoapServices9.2 Local Services, and SASWIPTools Local Services.

If you still have multicast issues, then run a simple multicast debug utility:

1. Extract jgroup.jar from one of the EAR files.
2. Run `java -Djgroups.bind_addr=<host or IP> org.jgroups.tests.McastReceiverTest -mcast_addr <address> -port <port>`

3. Run `java -Djgroups.bind_addr=<host or IP> org.jgroups.tests. McastSenderTest -mcast_addr <address> -port <port>`
4. Enter the value in the sender window. You should see it in the receiver side; if not then you have a network issue.

Best Practice: The default IP version (v4) can be overridden during installation and configuration. Each SAS 9.2 middle tier server and remotes service JVM has a property “`-Djava.net.preferIPv4Stack=true`”. If you want to use IPv6, then you need remove the IPv4 JVM argument or set it to false, then add the IPv6 property set to true “`-Djava.net.preferIPv6Addresses=true`”.

SAS 9.2 MIDTIER DEPENDENCIES

The SAS 9.2 middle tier uses resources such as JDBC, JMS, and Mail in the Web application server. SAS Shared Service depends on the JDBC, JMS, and Mail resources. SAS BI Dashboard uses the JMS. Since all the communication between the applications is **HTTP**, there is no hard dependency between the applications, except for the SAS BI Portlets. BI Portlets must be deployed in same server as the SAS Information Delivery Portal. In JBoss 4.2 using remote JMS servers is not possible, so you have to deploy SAS BI Dashboard and SAS Shared Service together in order to function properly. This is not the case with Oracle WebLogic and IBM WebSphere.

DEPLOYMENT TOPOLOGY

The deployment topology differs from customer to customer, mostly driven by customer-specific performance and scalability requirements. In a typical enterprise, four environments are common: development, testing, preproduction, and production. As a best practice, try to mimic the preproduction environment as a production environment.

In a typical deployment topology, the Web application server is front ended by a Web server (HTTP server). In most cases these HTTP servers are front ended by a hardware-based IP sprayer / load balancer or a reverse proxy such as Tivoli Access Manager or SiteMinder. The following diagram shows the typical deployment topology.

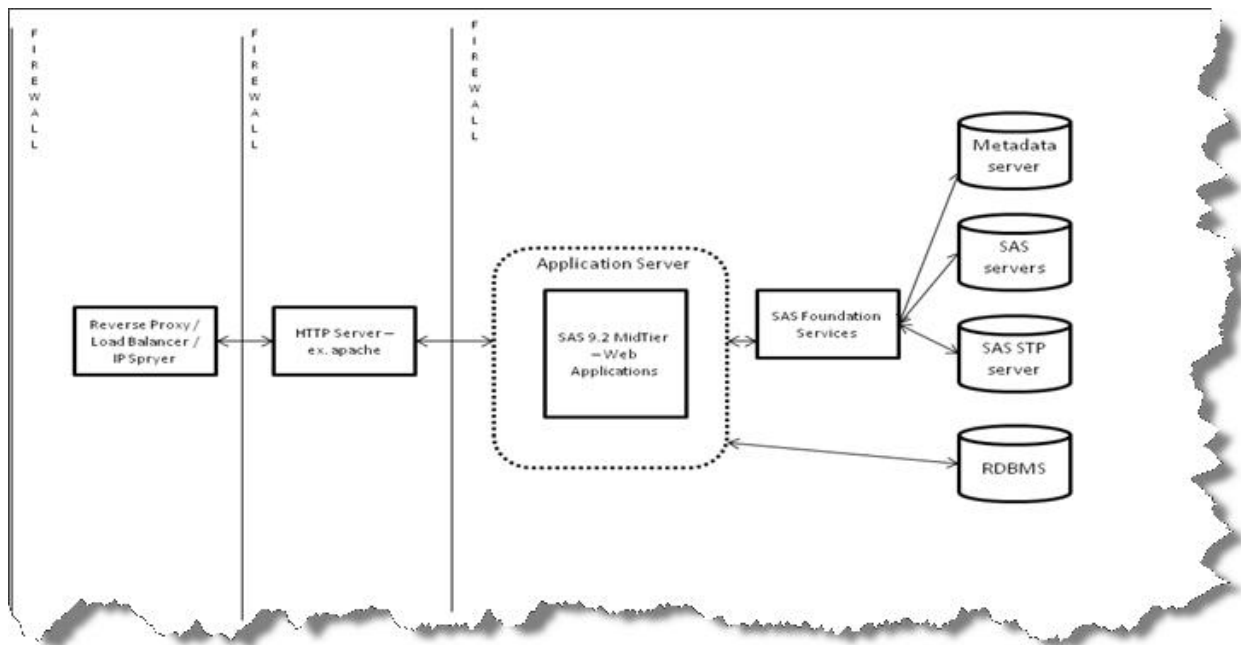


Figure 6. Deployment Topology

Best Practice: In SAS 9.2 the host and port information about the individual SAS 9.2 Web applications is kept in the SAS Metadata Server. When you cluster or distribute SAS 9.2 Web application across multiple servers, you have to make changes in the metadata server. It always best practice to keep the host name and port as your HTTP server's

host name and port, so that any changes to topology do not require changes in the metadata server. These changes are handled easily by the HTTP server.

TUNING

32 BIT VS 64 BIT

A Java process (or application server) is another operating system process like any other application, which is subject to addressability restrictions and OS memory model restrictions. The addressability is restricted by architecture. The Java process is combination of process address space, Java, and native heaps.

The 32-bit architecture has an addressable range of 2^{32} , 0x00000000 – 0xFFFFFFFF, which is 4GB.

A 64-bit architecture has a range of 2^{64} and 0x0000000000000000 – 0xFFFFFFFFFFFFFFFF, which is 16EiB.

The memory not allocated to the Java heap is available to the native heap. Native Heap = Available Memory Space – Max Java Heap Size

Due to the nature of the 32-bit architecture, your overall memory is limited to 2 GB. . In the 32-bit version of Windows you use the 3 GB switch in boot.ini to increase the address space. If you want to scale up memory for large numbers of users, you either create a new JVM (server) to distribute the applications or use the cluster. With the 64-bit JVM you do not encounter the memory restrictions.

GARBAGE COLLECTION

The name itself implies what it means: the objects that are not needed by the program are “garbage”. SAS 9.2 supports Sun JDK, IBM JDK, and HP JDK. When choosing the garbage collection scheme, several performance metrics need to be examined carefully: throughput, GC overhead, pause time, collection frequency, heap usage, and promptness. Both Sun JDK and HP JDK behaviors are similar, and so is the tuning. Each JDK provides several varieties of garbage collection schemes. We do not want to get into each one, but we will talk about the ones that work better for SAS 9.2 Web applications.

CMS: For both Sun and HP JDK we use the Concurrent Mark-Sweep (CMS) collector. GC pause time seriously affects the response time. Young generation collections do not typically cause long pauses. However, old generation collections, though infrequent, can impose long pauses, especially when large heaps are involved. CMS addresses this key issue and is often called a low-latency collector

In the IBM JDK, optavgpause and gencon work better for SAS 9.2 Web applications.

optavgpause: Application response time is important, so the GC pause time has to be minimal. optavgpause works better than the default IBM JDK GC and keeps the pause time to a minimum. It does not guarantee a particular pause time, but pauses are shorter than those produced by the default GC policy. The idea is to perform some garbage collection work concurrently while the application is running.

gencon: A generational garbage collection strategy considers the lifetime of objects and places them in separate areas: a nursery and a tenured area. The idea is that most objects are short-lived; by collecting the nursery frequently, these objects can be freed without paying the cost of collecting the entire heap. The tenured area is garbage collected less often.

SAS 9.2 WEB APPLICATION TUNING

You can distribute the SAS 9.2 Web applications across multiple Web application servers. For ways you can distribute the Web applications, the most common scenarios, and respective tuning, see [SAS 9.2 Web Applications: Tuning for Performance and Scalability](#) (PDF file).

Best Practice: When you are tuning the heap, you do not want to go beyond 4 GB, mainly due to garbage collection pause. If you need to exceed 4GB then use multiple servers to distribute the SAS 9.2 Web application, or use clusters.

MONITORING

Though several commercial monitoring solutions are available in the marketplace, we will look into the one available with the application server and JDK that we use with SAS 9.2.

Java VisualVM is a tool that provides a visual interface for viewing detailed information about Java applications. It is part of JDK 6, and can be downloaded separately when using with JDK 1.5. It provides plug-ins to various tools such as JConsole, jstatm, jinfo, jstack, and jmap. With VisualVM you can use all the tools from one interface. You can use VisualVM to monitor the JVM.

JOPR is an open source version of JBoss ON, a monitoring product for JBoss. You can monitor the JBoss and deployed applications (SAS 9.2 Web applications) using JPOR.

Oracle WebLogic provides WebLogic Diagnostic Framework as part of the WebLogic server. It provides WLDF Dashboard through its console extension. You can enable the diagnostic tool through the admin console: **Admin Console --> Click Preference --> Click the Extensions --> Select diagnostics-console-extension, and then enable it.**

IBM WebSphere provides the Performance Monitoring Infrastructure (PMI) through its console. PMI provides the fine grain details about system usage, application usage, connection pool, JVM, and more.

MIGRATION

SAS 9.2 middle tier applications are self contained in the EAR. The operational data exist in the back-end servers such as the SAS Metadata Server. The SAS Content Server uses a file-based repository; all the configuration and content server data exist in the **AppData** directory under the **SAS Config** directory. Generally, migrating from one environment to another is a combination of automatic SAS Deployment Wizard installation plus manual file copying and restoring. Any changes to BI Dashboard configuration under **AppData** need to carry over. If you have configured custom portlets, then those need to be carried over (copy the PAR file to the new environment). In WRS if you had any specific configuration changes under the **SASConfig\Lev1\Web\Applications\SASWebReportStudio4.2** directory, then those need to carry over to the new environment.

CLUSTERING

The SAS middle tier can be clustered for performance and high availability requirements. The clustering techniques as a whole are very common across the Web application servers, but require a few specific changes to individual application servers. The Web (HTTP) server is configured to manage the workload requests to multiple instances of SAS middle tier applications that are deployed in the Web application server. The round-robin algorithm used to distribute the workload. For example, if there are two instances of a SAS middle tier application involved, the workload is distributed 50/50 between the individual instances. The Web server maintains the session affinity based on the cookie information, so that the subsequent requests are guaranteed to target the right instance of the SAS middle tier application. The following figure shows a typical clustering topology for a SAS 9.2 Web application.

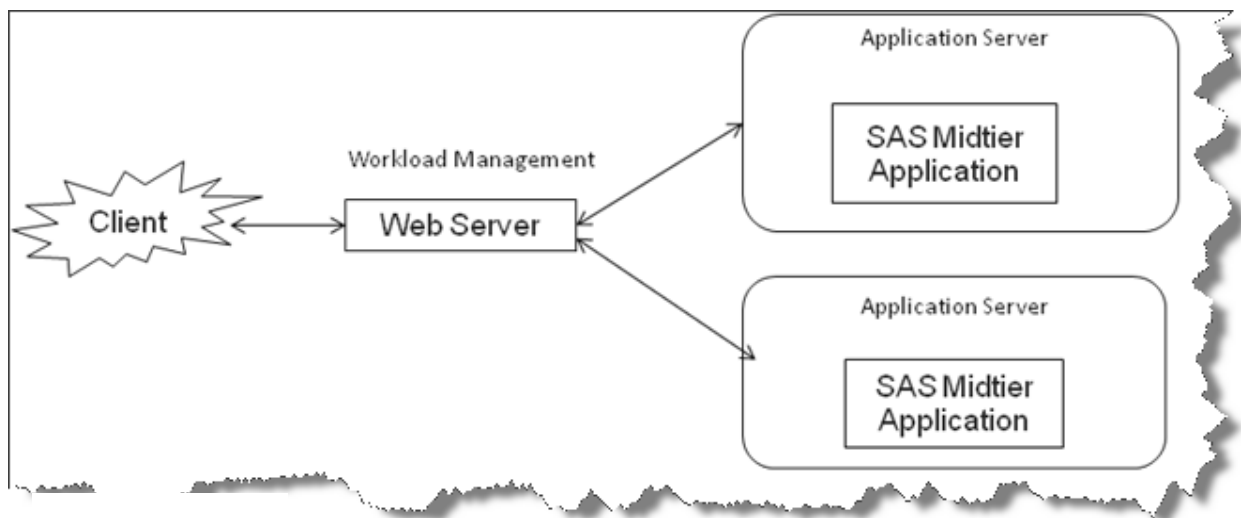


Figure 7. Clustering Deployment Topology

There are two types of clustering: vertical and horizontal. Vertical cluster scales the application server in a single physical machine, whereas horizontal cluster scales application servers in multiple physical machines. Horizontal clustering addresses, both software and hardware level failures, whereas vertical clustering addresses only software level failures. When a single point of failure is of primary concern at the hardware level, then choose horizontal clustering.

All SAS 9.2 installs are driven by a SAS install plan. The SAS Deployment Wizard installs software based the deployment plan that you choose. During the installation you will deploy and configure a fully functional SAS 9.2 middle tier with a single Web application server. Then you configure the server instance, configure the cluster, configure the ports in the metadata directory, and configure the HTTP server to distribute the workload. The following figure shows the high-level steps to achieve the clustering.

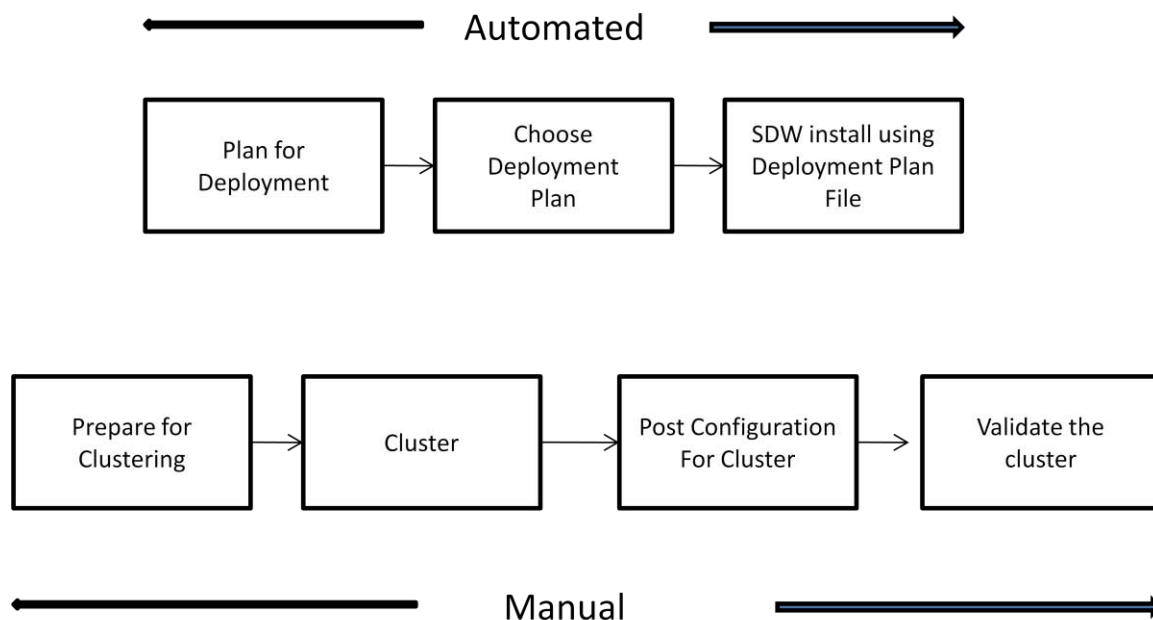


Figure 8. Clustering High Level Steps

For a step-by-step guide to clustering, see [SAS 9.2 Web Applications: Clustering \(Secured only available licensed customers\)](#).

CONCLUSION

The SAS 9.2 platform is more robust than previous versions; it provides various ways to scale the middle tier applications in both single JVM and clustered JVMs. You can easily monitor the SAS 9.2 middle tier using freely available tools that are part of the JDK or application server. You can fine tune the SAS 9.2 middle tier, the Web application server, and the OS based on your performance and scalability needs.

REFERENCES

SAS 9.2 Web applications: tuning for performance and scalability

<http://support.sas.com/resources/thirdpartysupport/v92/appservers/WebTierScalabilityAndPerformance.pdf>

ACKNOWLEDGMENTS

This paper is not possible without the assistance of SAS R&D; I would like to thank the entire R&D team.

RECOMMENDED READING

Third Party Software for SAS 9.2 Foundation

<http://support.sas.com/resources/thirdpartysupport/v92m2/index.html>

SAS 9.2 Support for Java Runtime Environments

<http://support.sas.com/resources/thirdpartysupport/v92m2/jres.html>

SAS 9.2 Support for Java Development Kits

<http://support.sas.com/resources/thirdpartysupport/v92m2/jdks.html>

SAS 9.2 Support for Web Application Servers and HTTP Servers

<http://support.sas.com/resources/thirdpartysupport/v92m2/appservers/index.html>

SAS 9.2 Planning Application Generated Pre-Install Checklists

http://www.sas.com/apps/demosdownloads/samplePlans_PROD_sysdep.jsp?packageID=000501

SAS 9.2 Intelligence Platform: Web Application Administration Guide, Second Edition

<http://support.sas.com/documentation/cdl/en/biwaag/63059/HTML/default/titlepage.htm>

SAS 9.2 Web Applications: Tuning for Performance and Scalability

<http://support.sas.com/resources/thirdpartysupport/v92/appservers/WebTierScalabilityAndPerformance.pdf>

SAS 9.2 Web Applications: Clustering (Secured only available licensed customers)

http://www.sas.com/pub/pub_solutions/bizdev/solutions_doc/thirdparty_support.php

CONTACT INFORMATION

Saravana R. Chandran
SAS Institute Inc.
500 SAS Campus Dr.
Room R4409
CARY, NC 27513

SAS and all other SAS Institute Inc. product or service names are registered trademarks or trademarks of SAS Institute Inc. in the USA and other countries. ® indicates USA registration.

Other brand and product names are trademarks of their respective companies.