

Paper 301-2010

Scaling the SAS® 9.2 EBI Web Tier with Multiple Managed Servers and Server Clusters

Scott Sweetland, SAS Institute Inc., Cary, NC

ABSTRACT

Exploring several concepts for scaling the SAS® 9.2 EBI Web tier, this paper will present an overview of SAS 9.2 Web architecture and deployed components, and discuss a recent successful customer deployment. We will look at strategies and details for distributing SAS 9.2 Web Infrastructure Platform components and EBI Web applications across multiple managed servers and server clusters. Deployment examples will be given from field engagements, with an analysis of benefits and potential pitfalls.

INTRODUCTION

Don't you wish you could clone yourself for vast gains in productivity? This day dream introduces us to the main concepts of Web server clustering. Let's say you make three 'Me-Clones' for the perfect car pool. Now you have four times the brain power, memory, and personal I/O channels to solve problems and attend meetings. Your 'Team-Me' car pool is like a vertical server cluster: multiple servers running on the same hardware. But if your hardware fails, like the car for your car pool, then no one from Team-Me gets to work.

You decide to expand the reliability of Team-Me and let your Me-clones have their own cars. Now you are emulating a horizontal server cluster with the reliability of independent hardware.

Soon you discover that some tasks just can't be done as a team. You decide to use a cold standby backup strategy for these tasks: an on-demand Me-clone, only to be activated for emergencies, when the primary Me (that's you) fails.

Finally, you know that the productivity of your Team-Me cluster will depend on the details of workload distribution. With a bad configuration, you might end up with all of your clones doing the same task while no one is working on a critical task.

All of these concepts translate directly to our topic. Just think of your team of Me-clones as a cluster of Web servers as we work through the Web tier deployment of the SAS 9.2 EBI Server:

- How do we design our Web server cluster for performance and availability?
- Which applications in the EBI Web tier can be deployed to a server cluster?
- How do we provide high availability for applications that cannot be clustered?
- How do we distribute the workload to the servers in our cluster?

The SAS 9.2 EBI Web tier consists of the Web Infrastructure Platform (WIP) applications and the EBI client applications, including the SAS Information Delivery Portal, SAS Web Report Studio, SAS Web OLAP Viewer, and SAS BI Dashboard. By deploying to a server cluster, we can improve performance through multiple instances of applications, allowing for greater memory utilization and more server resources. Additional benefits come in the area of high availability. With applications hosted on multiple servers, the loss of one server has limited impact. If we build our cluster using multiple machines, we gain protection from both server failure and hardware failure.

This paper will address key concepts of cluster deployment at a high level. However, SAS Technical Support provides excellent documentation that gives detailed step-by-step instructions for successful cluster deployment on the SAS 9.2 supported Java™ Web Applications Servers. **SAS 9.2 Web Application: Clustering** will be your trusted reference as you begin planning and carrying out your cluster deployment.

BUILDING A CLUSTER

How you structure the server cluster depends on your goals for performance and availability, and available hardware. A vertical cluster (Figure 1) consists of multiple managed servers on a single hardware platform. More servers give you better performance for large numbers of concurrent users. Because each server's Java™ Virtual Machine (JVM) has limits on memory utilization, having multiple servers expands our memory utilization for the machine, while multiple servers provide additional capacity for user sessions. The failure of one server has limited impact with the availability of applications on other servers. However, a failure of the entire machine would be catastrophic. Therefore, applications requiring the highest levels of availability should be deployed to a horizontal cluster (Figure 1). This form of clustering uses multiple servers hosted on different machines. Many customers decide to implement a combination of both vertical and horizontal clustering as we will see in our example deployments.

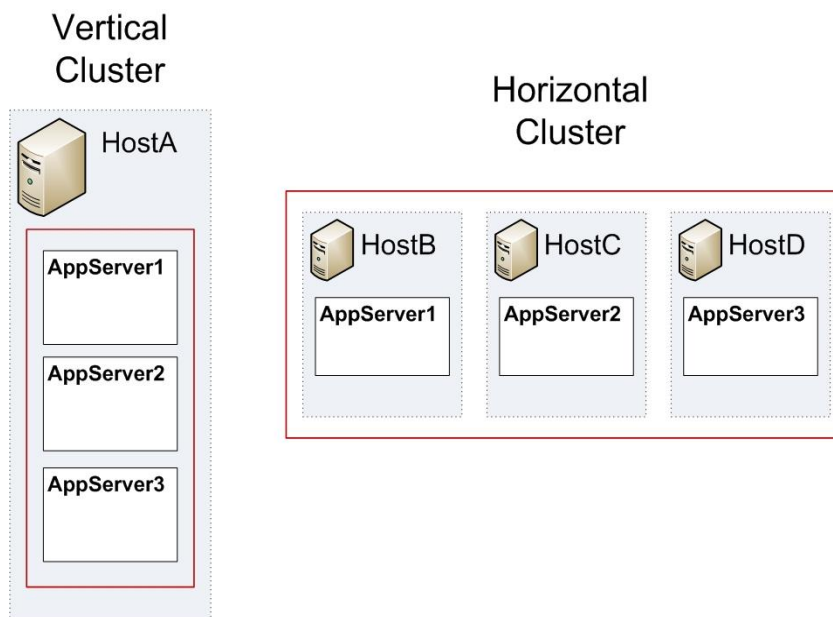


Figure 1. Vertical Application Server Cluster and Horizontal Application Server Cluster

WEB APPLICATION DEPLOYMENT

SAS WEB INFRASTRUCTURE PLATFORM

The SAS Web Infrastructure Platform is a collection of middle tier services and applications that provide integration capabilities for SAS 9.2 EBI Web applications. SAS Web Infrastructure Platform services provide authentication, WebDAV content management, and Web Theme support. SAS Web Infrastructure Platform applications include the Logon Manager, which handles authentication requests for all Web applications, the Preferences Manager, which manages user preferences, and the Stored Process Web Application, which manages the execution of stored processes and the display of results. The SAS Content Server provides WebDAV services and the SAS OnlineDoc application provides a library of reference documentation for the SAS System.

Clustering the SAS Web Infrastructure Platform services and applications provides enhanced performance and reliability, but as we will see in a later section, not all of these applications can be clustered. We must look to different strategies to provide high availability.

SAS 9.2 EBI WEB APPLICATIONS

The Web client applications of the SAS Enterprise Business Intelligence Server provide the means to create, share, and personalize SAS BI content while providing a secure framework to aggregate content with custom dashboards and alerting. Clustering these applications provides performance enhancements as well as high availability. A few features of these applications are not supported on a cluster deployment. In the next section, we will look at strategies to support these features in a stand-alone server deployment.

LIMITATIONS FOR CLUSTER DEPLOYMENT

Certain features of the SAS 9.2 EBI Server applications are not supported in a clustered deployment:

- SAS Content Server.
- SAS Web Report Studio in-process job scheduling is supported only for single deployments of the application. Job scheduling through Platform Process Manager (a feature of Platform Suite for SAS) is supported in a cluster deployment.
- SAS OnlineDoc for the Web.
- SAS BI Dashboard Event Generator, a Web application bundled in the SAS BI Dashboard EAR.

To provide high availability for the Content Server, OnlineDoc, and BI Dashboard Event Generator, we can deploy these applications to a stand-alone server, and use a 'Hot with Cold Standby' strategy (Figure 2). We configure two stand-alone servers, each with identical application deployments. The 'hot' server provides the primary deployment of the applications, while the other server is 'cold' or inactive until needed for a primary server failure. The servers can be maintained on the same or different machines, but can never be running at the same time.

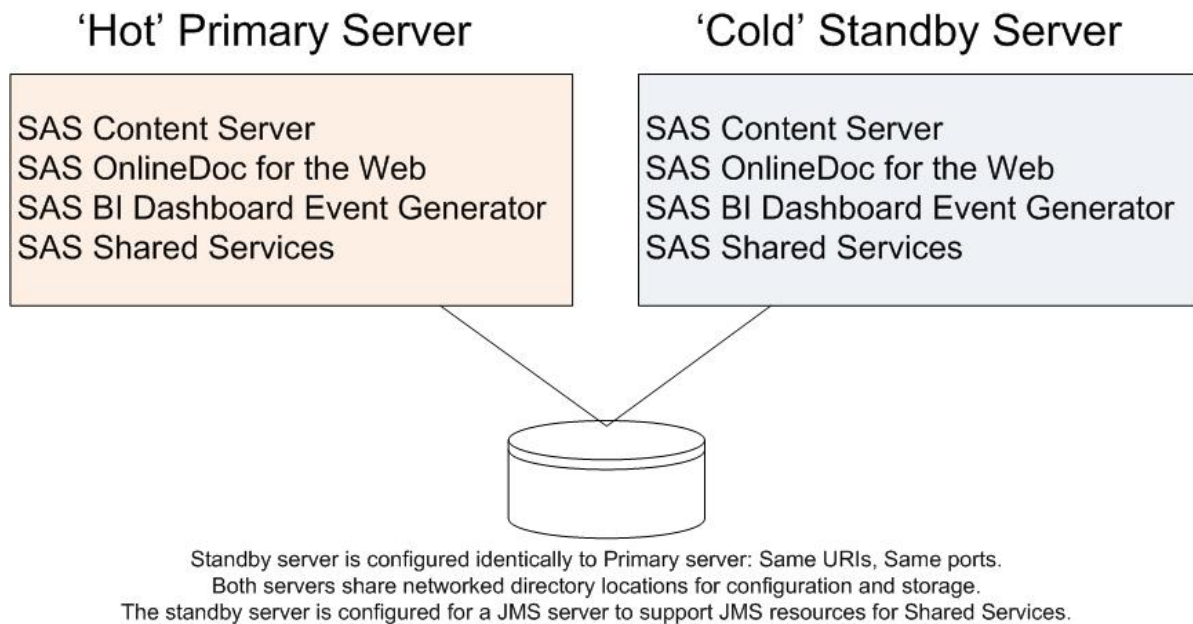


Figure 2. Hot with Cold Standby Server Strategy for High Availability of a Stand-Alone Server

Separate deployment of the BI Dashboard Event Generator requires that the WAR file be removed from the BI Dashboard EAR file. The BI Dashboard EAR file is reconstructed without the Event Generator WAR file. This reconstruction enables us to deploy the BI Dashboard application to a server cluster, while deploying the Event Generator application WAR file to our Hot and Cold Standby server implementation.

The SAS Web Infrastructure Platform Shared Services uses a Java Messaging Service (JMS) server to provide JMS resources. In the EBI Web tier deployment, only the BI Dashboard Event Generator uses the JMS resources of the SAS Infrastructure Platform Shared Services. Since the Event Generator application cannot be clustered, there is little benefit in clustering the Shared Services and implementing JMS over all servers in the cluster. So we can simplify our deployment by hosting Shared Services only on our stand-alone server and only configuring JMS on this server.

The documentation provides details for re-organizing and re-deploying the BI Dashboard EAR file. It describes how to deploy the SAS Shared Services and supporting JMS resources on the stand-alone server. Our strategy for application deployment to the stand-alone and clustered servers is presented in Table 1.

Table 1 SAS 9.2 EBI Server Web Tier Application Deployment

Application		Deployment Target
Web Infrastructure Platform	SAS Web Infrastructure Platform Services <code>sas.wip.services9.2.ear</code>	Target all servers in the cluster.
	SAS Web Infrastructure Platform Applications <code>sas.wip.apps9.2.ear</code>	Target all servers in the cluster.
	SAS Shared Services <code>sas.shared9.2.ear</code>	Target the stand-alone server.
	SAS OnlineDoc for the Web <code>sas.webdocmd9.2.ear</code>	Target the stand-alone server.
	SAS Content Server <code>sas.wip.scs9.2.ear</code>	Target the stand-alone server.
	SAS Stored Process Web Application <code>sas.storedprocess9.2.ear</code>	Target all servers in the cluster.
	SAS Themes Web Application <code>sas.themes.ear</code>	Deploy to an HTTP server for best performance.
EBI Server Web Applications	SAS Information Delivery Portal <code>sas.portal4.2.ear</code>	Target all servers in the cluster.
	SAS Web Report Studio <code>sas.webreportstudio4.2.ear</code>	Target all servers in the cluster.
	SAS Web OLAP Viewer for Java <code>sas.webolapviewer4.2.ear</code>	Target all servers in the cluster.
	SAS BI Dashboard <code>sas.bidashboard4.2.ear</code>	Target this EAR file all servers in the cluster if the <code>sas.eventsgenerationframework.war</code> file was removed. Otherwise, target this EAR to the stand-alone server.
	SAS BI Dashboard Event Generator <code>sas.eventsgenerationframework.war</code>	If this WAR was extracted from the <code>sas.bidashboard4.2.ear</code> file, deploy this file to the stand-alone server.
	SAS Package Viewer <code>sas.packageviewer.ear</code>	Target all servers in the cluster.

	SAS BI Portlets sas.biportlets4.2.ear	Target all servers in the cluster.
--	--	------------------------------------

PLANNING AND PROCESS

Begin your deployment process by completing a sizing estimate for concurrent users and a profile of typical user tasks. Based on these requirements, you will develop a capacity plan for the number of servers in your cluster and the hardware capacity to support the cluster. The final plan will also be influenced by your availability requirements, such as adding servers and hardware to provide redundant capacity. With a plan to meet both performance and availability requirements, you are ready to begin a process of deployment following these high-level steps:

1. Develop the deployment plan for your SAS 9.2 EBI software.

While we are focused on the Web tier, this plan must also include the SAS application server tier. You will specify a single server for the Web tier.

2. Install SAS software using the SAS Deployment Wizard.

The SAS Deployment Wizard will create a server instance with all EBI Server applications targeted to this server. Use this server instance as a template to create the additional servers for the cluster deployment.

3. Prepare for clustering.

As noted in the previous section, the SAS BI Dashboard EAR file must be prepared for clustering by extracting the BI Dashboard Event Generator WAR file.

4. Cluster the applications.

Using the server instance from Step 2 as a template, create the additional server instances for your cluster. As noted in previous sections, not all applications will be targeted to the cluster. Figure 4 shows a simple single machine cluster of 2 servers with one stand-alone server, and how the Web applications would be targeted for deployment. If you have multiple machines, you will need to copy the Web tier configuration directories to each machine. See the **SAS 9.2 Web Application: Clustering** documentation for application server-specific details. Figures 3, 4, and 5 show the steps as a generalized process.

5. Configure an HTTP server as a reverse proxy to distribute workload.

This step varies in complexity and automation depending on the Application Server. You must install the server-specific plug-in to the HTTP server and configure it to correctly distribute HTTP requests to the clustered servers and stand-alone server. Most of the Web Infrastructure Platform applications do not have an end-user interface and are accessed only by the EBI Server Web applications, and in general, have no need to be surfaced by the proxy server. The SAS Technical Support Web site, support.sas.com, provides detailed instructions to configure a reverse proxy server. Under **Knowledge Base**, click the **Third-Party Software Reference** link.

6. Change the configuration for the clustered deployment.

Having completed the clustered deployment, we must modify the original SAS Deployment Wizard configuration to reflect the new deployment targets for key Web tier components. These configuration changes are made in the metadata for the Web tier components and applications. In addition, log file targets for applications deployed to the cluster must be reconfigured to be unique for each application instance in the cluster.

7. Validate the cluster.

In the next sections, we will look more closely at steps 5, 6, and 7.

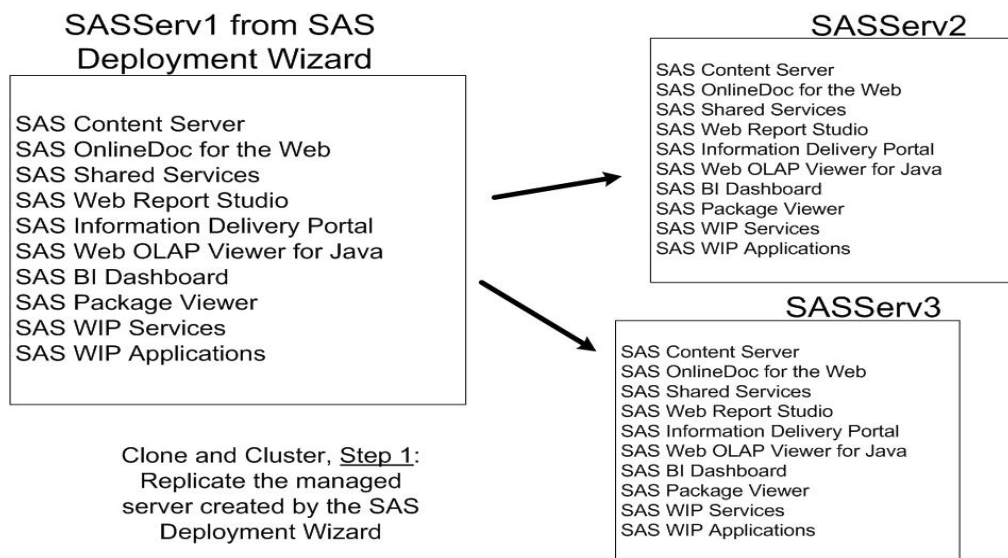


Figure 3. Replication of the Server Created by the SAS Deployment Wizard

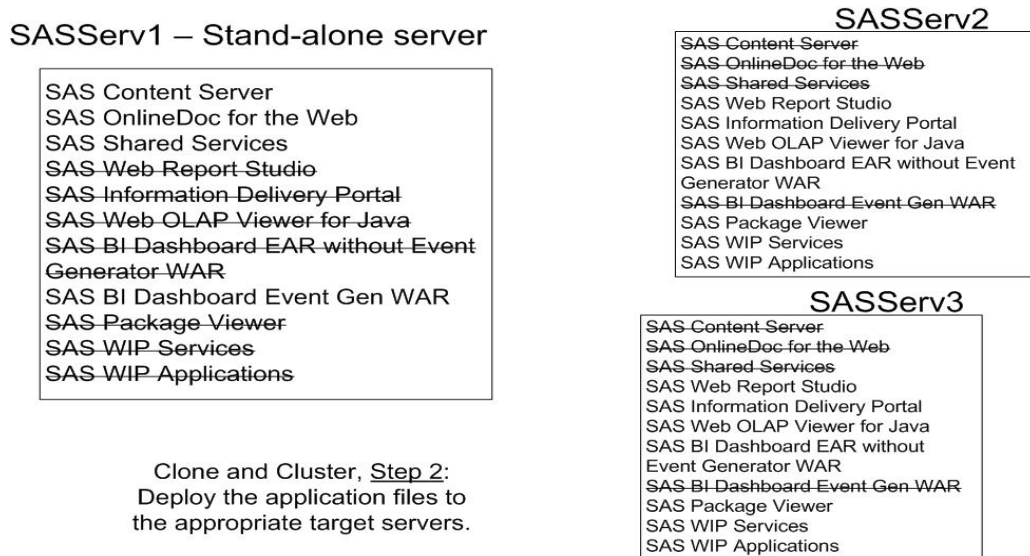
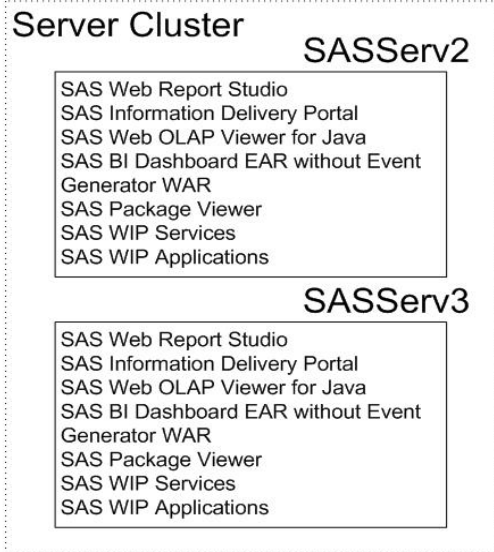


Figure 4. Deployment of the Application Files to the Appropriate Targets

SASServ1 – Stand-alone server

- SAS Content Server
- SAS OnlineDoc for the Web
- SAS Shared Services
- SAS BI Dashboard Event Gen WAR



Clone and Cluster, Step 3:
Define our cluster as SASServ2
and SASServ3.

Figure 5. Simple Cluster Deployment

DISTRIBUTING THE WORKLOAD

REVERSE PROXY SERVER

For our clustered deployment of the EBI Server Web tier (Figure 5), HTTP requests must be distributed to the appropriate servers. Some HTTP requests are simply forwarded to the stand-alone server, while requests to the clustered applications are load balanced for best performance. An HTTP server configured with a reverse proxy plug-in module provides this service. Application server plug-in modules provide load balancing to clustered servers while also preventing direct access to the application servers (Figure 6).

The configuration of a reverse plug-in module is similar for all application servers: specific URI requests are identified for distribution to one or more application servers. Configuration details are provided in the documentation.

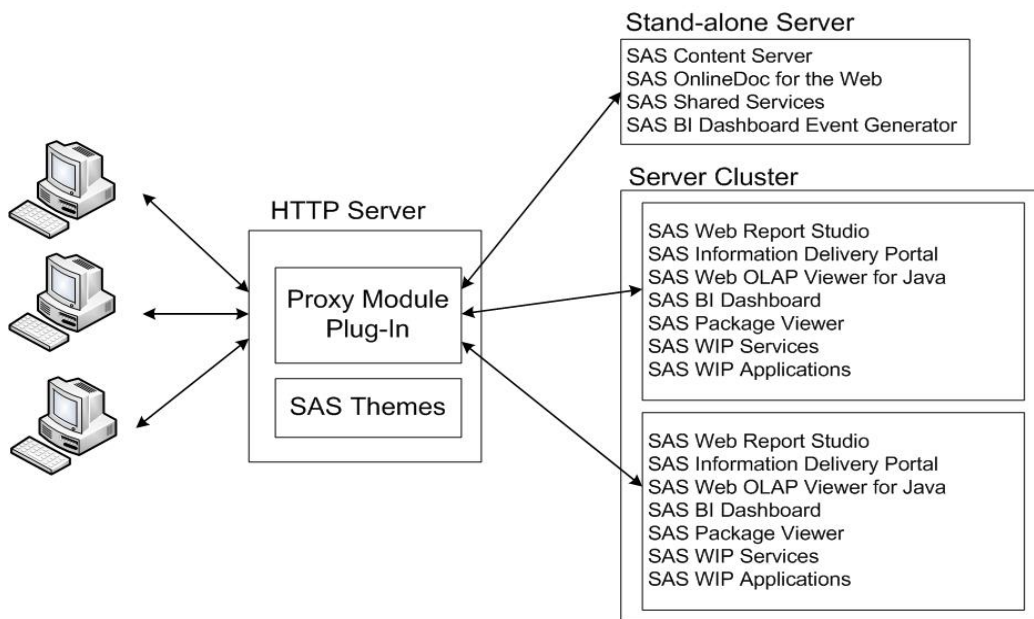


Figure 6. Reverse Proxy Server for a Simple Cluster Deployment

SAS THEMES

SAS Theme content should be deployed to an HTTP server to improve the performance of serving this static HTML content. In our simple cluster deployment example, the reverse proxy server is a convenient location to host this content. As described in the “SAS 9.2 Intelligence Platform: Web Application Administration Guide,” this is as simple as extracting the `sas.theme.default.war` file to a new directory named `/SASTheme_default`.

METADATA CONFIGURATION

Integration of the SAS EBI Server Web tier relies on metadata configuration to provide connection information for applications and services. The initial metadata configuration is established by the SAS Deployment Wizard during the deployment of the application server, and the connection information for all applications and services is based on the host and port of this server.

After re-deployment to a server cluster, the metadata configuration must be updated to use the proxy server host name and port as the new connection information. The SAS Management Console is used to make these changes. The Configuration Manager is used to update the host name and port for the SAS Web Infrastructure Platform and EBI applications. The Server Manager is used to update this information for the SAS Content Server, and the Foundation Services manager is used to update the WebDAV URL (SAS Content Server host and port) for the Information Service WebDAV repository in local and remote instances of the SAS Foundation Services. Step-by-step details are provided in the documentation.

APPLICATION LOGGING FILE CONFIGURATION

Web application logging file locations are configured in metadata. The SAS Deployment Wizard creates an initial configuration for each application based on a single file path location for a single server. When deployed to a server cluster, each of the multiple instances for an application attempts to write to the same file location. Two JVM options enable us to uniquely identify the server JVM where the application is deployed, and generate a unique logging file location for each application instance in the cluster. For each server JVM, add these options:

```
-Dsas.appserver.instance.id=SASServerN
-Dsas.wrs.keyUserActionLog.filename=
    SASWebReportStudio4.2_KeyActions_SASServerN
```

where **N** is a unique number identifying the server in a cluster on a machine.

Use the SAS Management Console Foundation Services manager to edit the logging file pathname for each Web application (not for the Remote Services configuration):

1. In the **Core** folder, right-click **Logging Service** and select **Properties**.
2. Select the **Service Configuration** tab and the **Configuration** button.
3. Select the **Outputs** tab and edit the SAS_LS_File output.
4. Change the **File** parameter to use the unique identifier defined in the JVM option:
 `/SAS/Config/Lev1/Web/Logs/${sas.appserver.instance.id}/SASBIDashboard4.2.log`
5. Exit with **OK** for each dialog box.

When you restart your Application Server, confirm that each application instance is properly logging to a file path that enables you to identify the server instance.

VALIDATE THE CLUSTER

You can validate the proper function of your clustered deployment by tracing a series of logons through the access logs for each Web server. This may require a separate configuration change for the reverse proxy plugin module. The validation can be structured through the use of automated testing software to simulate expected concurrent user workloads. Based on simulated workloads, you can choose to expand your cluster size, or tune the deployment with additional hardware.

EXAMPLE DEPLOYMENTS

Figure 7 shows an IBM® WebSphere™ Application Server clustered deployment for the SAS 9.2 EBI Server. It is defined as a single cell of two machines, with the Deployment Manager controlling a node agent on each machine. Horizontal clustering provides high availability. An IBM HTTP server (not depicted) is used as a reverse proxy server and hosts the SAS Theme content. This deployment provides performance gains and high availability through the use of horizontal clustering.

WebSphere Application Server 6.1 with Horizontal Clustering

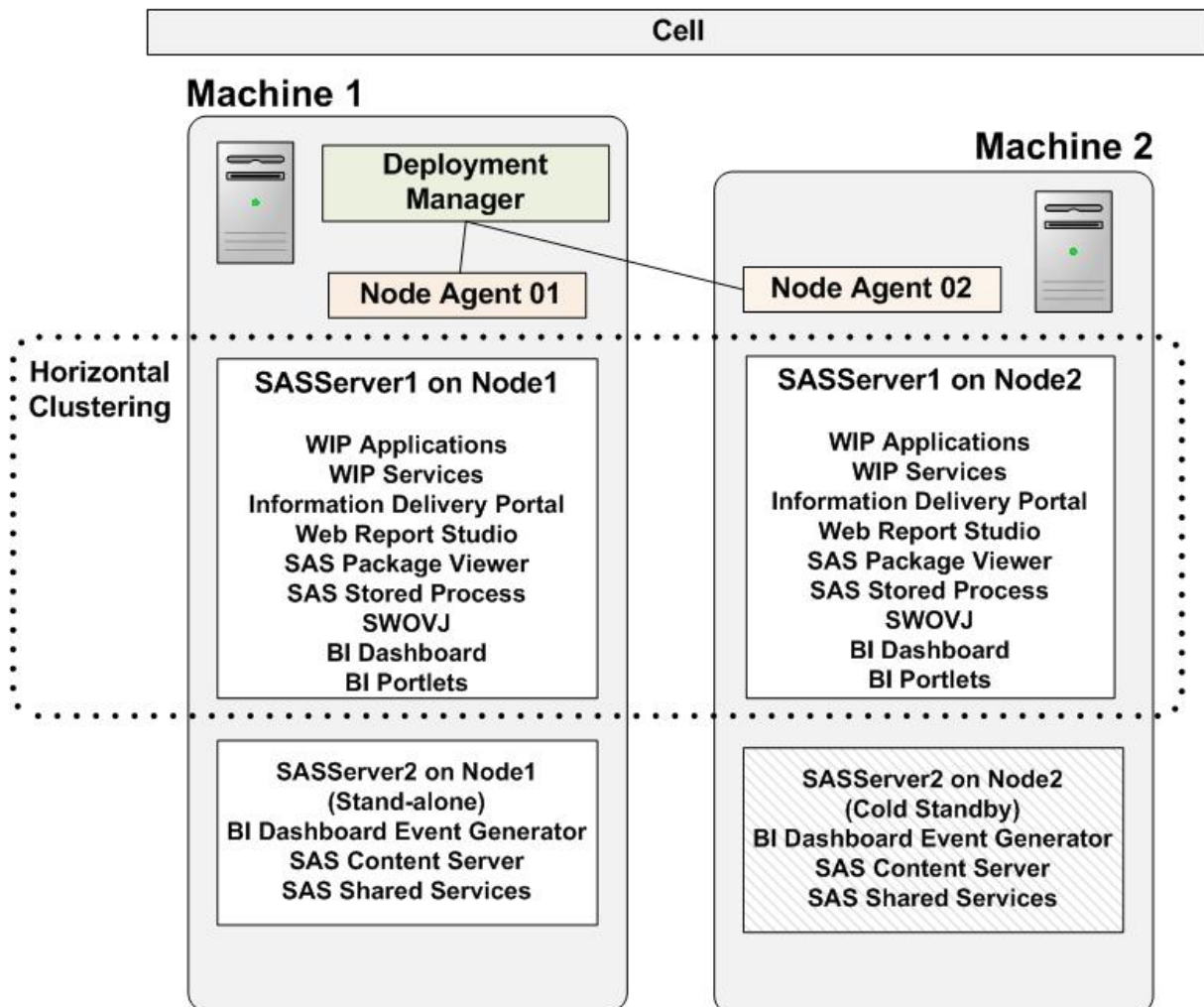


Figure 7. WebSphere Application Server 6.1 Horizontal Clustering Deployment

Figure 8 depicts both vertical and horizontal cluster deployment. Each application server machine has three clustered servers. The Hot with cold backup strategy is used for non-clustered applications and the Remote Services application. This deployment has very high performance characteristics while providing very high availability.

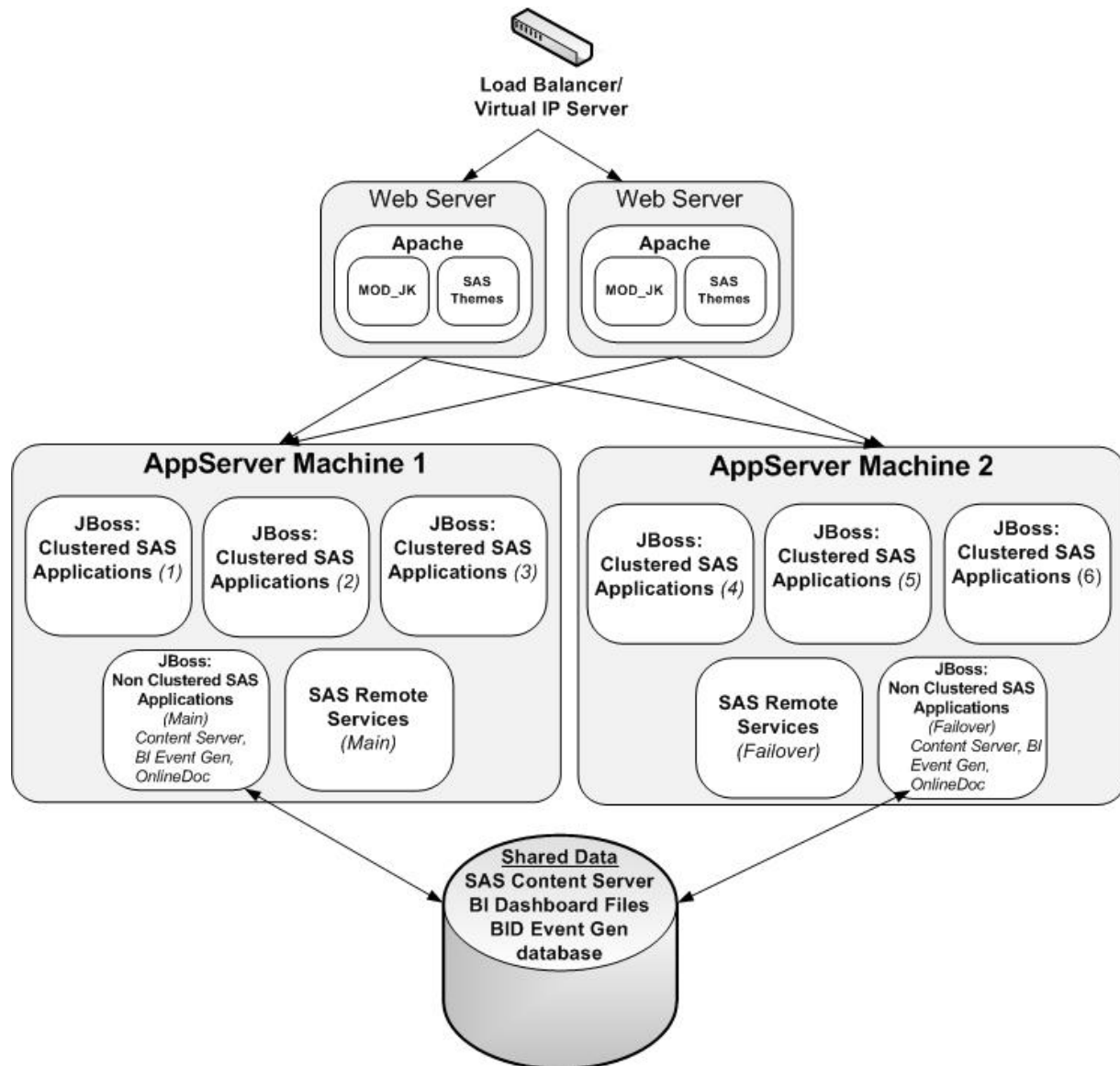


Figure 8. High Performance Clustered Deployment with High Availability from Horizontal Clustering.

CONCLUSION

Web server clustering provides performance gains through the expanded use of memory and operating system resources. An additional feature is improved reliability, with server clusters that span multiple machines providing the highest availability. The applications and services of the SAS 9.2 EBI Server Web tier acquire all of these features when deployed to a Web server cluster, enabling you to scale for larger numbers of users while providing dependable and reliable service. SAS is committed to supporting customers as they explore the benefits of server cluster deployment with ongoing research and testing of best practice deployments for supported application servers.

REFERENCES

SAS Institute Inc. 2009. *SAS 9.2 Intelligence Platform: Web Application Administration Guide*, 2d ed. Cary, NC: SAS Institute Inc.

SAS Institute Inc. 2009. *SAS 9.2 Web Applications: Tuning for Performance and Scalability*. Cary, NC: SAS Institute Inc.

SAS Institute Inc. 2009. *SAS 9.2 Web Application: Clustering*. Available on request.

ACKNOWLEDGMENTS

Saravana Chandran, SAS R&D Java Development Technologies, for his excellent documentation and assistance in exploring the many configurations that are possible with Web server clustering.

Meera Venkataramani, SAS R&D BIC Development Support, for her extensive best practices work for customer clustering environments.

Steve Clark, SAS Professional Services, who contributed with requirements and provided best practice knowledge from the field.

RECOMMENDED READING

IBM Corporation. IBM WebSphere Application Server 6.1 Information Center.

http://publib.boulder.ibm.com/infocenter/wasinfo/v6r1/index.jsp?topic=/com.ibm.websphere.nd.iseries.doc/info/seriesnd/ae/crun_srv_grp.html

IBM Corporation. (2005). *IBM Redbooks® Documentation for WebSphere Application Server 6.1*.

<http://www.redbooks.ibm.com/abstracts/sg246688.html?Open>

Red Hat, Inc. 2007. *JBoss Enterprise Application Platform 4.2.0 Clustering Guide*.

http://www.redhat.com/docs/manuals/jboss/jboss-eap-4.2/doc/Clustering_Guide.pdf

JBoss, Inc. 2004-2005. *The JBoss 4 Application Server Guide, JBoss AS 4.0.3, Release 4*.

<http://docs.jboss.org/jbossas/jboss4guide/r4/html>

CONTACT INFORMATION

Your comments and questions are valued and encouraged. Contact the author at:

Name:	Scott Sweetland
Enterprise:	SAS Institute Inc
Address:	SAS Campus Drive
City, State ZIP:	Cary, NC 27513 USA
Work Phone:	919 531 7174
Fax:	919 677 4444
E-mail:	scott.sweetland@sas.com
Web:	support.sas.com

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.