

## Migrating SAS® Java EE Applications from WebLogic, WebSphere, and JBoss to Pivotal tc Server

Zhiyong Li and Alec Fernandez, SAS Institute Inc., Cary, NC

### ABSTRACT

SAS® has a large portfolio of Java EE applications. In releases previous to SAS® 9.4, SAS provides support for configuring, deploying, and running these applications in Oracle WebLogic, IBM WebSphere, or Red Hat JBoss. Beginning with SAS® 9.4, SAS has updated the middle-tier architecture to deliver and run these web applications exclusively in the SAS® Web Application Server (a specialized, extended configuration of Pivotal tc Server) rather than the other third-party web application servers. This paper discusses the motivation, technology selections, and architecture on which this change is based. It also describes the advantages that the new approach presents to customers, including increased automation of installation and configuration tasks and improved system administration.

### INTRODUCTION

As SAS has responded to customer needs over the years, the list of products, solution offerings, and features has been greatly expanded. Additionally, the need to meet increased security demands and integrate with a variety of authentication providers has imposed more requirements on SAS software. As these new features and integration points have been added, the amount of complexity has increased. Over time, this complexity has placed an increasing burden on customers in terms of both the initial deployment cost and the total cost required to maintain the software throughout its lifecycle.

Beginning with SAS 9.4, SAS is seeking to reverse the trend toward increasing cost and complexity, while maintaining and even expanding the functionality of the software and expanding the ability to integrate with enterprise IT infrastructure.

SAS seeks to accomplish these goals by embracing the latest technology directions, which lean away from heavyweight J2EE/Enterprise Java Beans and lean toward more flexible, lighter-weight coding paradigms such as SpringSource. By re-architecting the entire middle tier around these paradigms, SAS has been able to eliminate the requirement that the web applications be run on a J2EE server such as Oracle WebLogic, IBM WebSphere, or Red Hat JBoss. Now, all SAS web applications can be run in a lightweight servlet container based on Apache Tomcat.

This approach provides SAS customers with many advantages, including the following:

**Reduced Cost.** SAS no longer requires that customers procure a third-party web application server and an associated maintenance contract. SAS provides the software and the support.

**Reduced Time to Value.** In many cases, once the SAS Deployment Wizard finishes, the software is completely functional. All of the applications are up and running. SAS has eliminated the need to wait for a separate web team to configure the third-party servers per written instructions and then deploy the applications before integration and acceptance testing can proceed.

**Improved Deployment Experience.** Because SAS now controls the entire middle-tier infrastructure, many of the deployment tasks that have heretofore involved manual steps performed by web server teams can now be performed automatically by the SAS Deployment Wizard. This speeds and simplifies the software deployment process. Additionally, automated deployments lead to greater uniformity in the deployed architecture, which eliminates variables in the running applications and the inherent support issues that arise from increased variability. Furthermore, fewer manual steps reduce the likelihood of human errors during the integration process.

**Reduced Complexity/Increased Stability.** Because SAS has standardized on a single web application server, customers are now guaranteed to be running their applications on the very same server, using the very same tuning parameters, that was used to develop and test the applications at SAS. Several moving parts have been eliminated, which allows for simplified deployments and eliminates the need for layers of abstraction that were required when multiple application servers were supported. Also, compromises between tuning options that benefited performance on one application server but were detrimental to another are no longer required. SAS web applications can now be tuned for optimal performance.

**Improved Support for Virtualization and Cloud-Based Deployments.** The new architecture for the SAS Web Application Server tier was specifically selected to work well in the virtualized environments that are increasingly in demand. Because support for virtualization has been considered from the outset, customers who want to utilize a

cloud-based infrastructure will find the process much simpler. Since all the commercial licenses required to run the middle-tier servers are with SAS, transitions to cloud-based deployments will be less costly.

**Improved Customer Support.** SAS offers first and second level support for the HTTP server and web application server, so that customers can now turn to SAS Technical Support to resolve all issues. SAS has a license agreement that provides a mechanism to obtain bug fixes for problems or security vulnerabilities that might be discovered. These fixes will then be committed by Pivotal to the Apache code base and included with subsequent Apache releases of the open-source project. This eliminates the need for customers to contact two different support organizations, which can add delays and increase the amount of effort required to reach a resolution due to the need for coordination among multiple parties.

**Built-in Monitoring and Management.** SAS Environment Manager includes enterprise monitoring software that provides a web interface for managing and monitoring the health and performance of the entire SAS deployment.

## TECHNOLOGY OVERVIEW

**Java Runtime Environment.** The SAS middle-tier environment includes a Java Runtime Environment (JRE) that is included with the SAS 9.4 software deployment. Customers no longer need to install a separate Java instance for the middle-tier environment. This approach ensures that the JRE is isolated from any web browsers that might be running on the same server. It also provides specific versions of Java that have been tested to ensure compatibility with the SAS software products that are being deployed.

The SAS private JRE cannot be replaced by downloading newer versions of Java from vendor sites and using the vendor-provided installers. SAS will evaluate future revisions and patches to the JRE as they become available. SAS will package and deliver these updates to SAS customers as necessary to address security issues and to ensure that the functionality of SAS applications is correct and complete.

**SAS Web Server.** SAS Web Server is an HTTP server that is based on Pivotal Web Server. The Pivotal Web Server is a commercial distribution of the Apache HTTP web server. The Pivotal distribution of the Apache HTTP server is backed with enterprise support. This affords SAS customers the proven stability of the Apache HTTP server coupled with the support infrastructure that enterprise applications require.

**SAS Web Application Server.** SAS Web Application Server is a lightweight server that provides enterprise-class features for running SAS web applications. It is a specialized, extended configuration of Pivotal tc Server. Pivotal tc Server has Apache Tomcat at its core and provides the highly scalable, lightweight application server that customers require, along with operational management, monitoring, diagnostics, security, and enterprise support.

By many measures, Tomcat is the most frequently used web application server in existence today. It powers numerous very large-scale, mission-critical web applications across a diverse range of industries and organizations including Walmart, The Weather Channel, and many others. Some of these users and their stories are listed at <http://wiki.apache.org/tomcat/PoweredBy>.

**Java Message Service Broker.** The Java Message Service (JMS) Broker is based on Apache ActiveMQ.

A JMS broker enables web applications to quickly respond to client requests without requiring the client to wait for completion of all the work that the request requires. For example, when a customer clicks a button on a web page, data might need to be analyzed, reports might need to be generated, and results might need to be e-mailed to many subscribers. The customer would not want to wait for all of these steps to complete before receiving confirmation from the server that the work order has been successfully received and is being processed. The JMS broker is the queuing entity that tracks the requested items from the time that they are requested until they are completed.

Some SAS web applications use JMS connection factories, queues, and topics for implementing business logic. These resources are configured in SAS Web Application Server for use by the SAS web applications.

**Cache Locator.** The Cache Locator is based on Pivotal vFabric GemFire. The software is used by applications on the server-tier and middle-tier machines to locate other members and form a data cache. When the SAS Web Application Server starts, it contacts one of the locators to initialize communication with the distributed cache. With that information, SAS Web Application Server instances form the cache that is needed to share run-time information across server instances, which might be on different machines. A locator is also configured on the server tier to provide access to the data cache for stand-alone client applications such as SAS Web Infrastructure Platform Scheduling Services.

**SAS Environment Manager.** The SAS middle-tier environment includes SAS Environment Manager. SAS Environment Manager is based on VMware vFabric Hyperic, an enterprise system management and monitoring system. Hyperic has capabilities to monitor and manage system and application resources, including OS platforms, services running on the system, different types of servers such as database servers and web application servers, and services implemented in the servers. It also has alert, event, and log management capabilities.

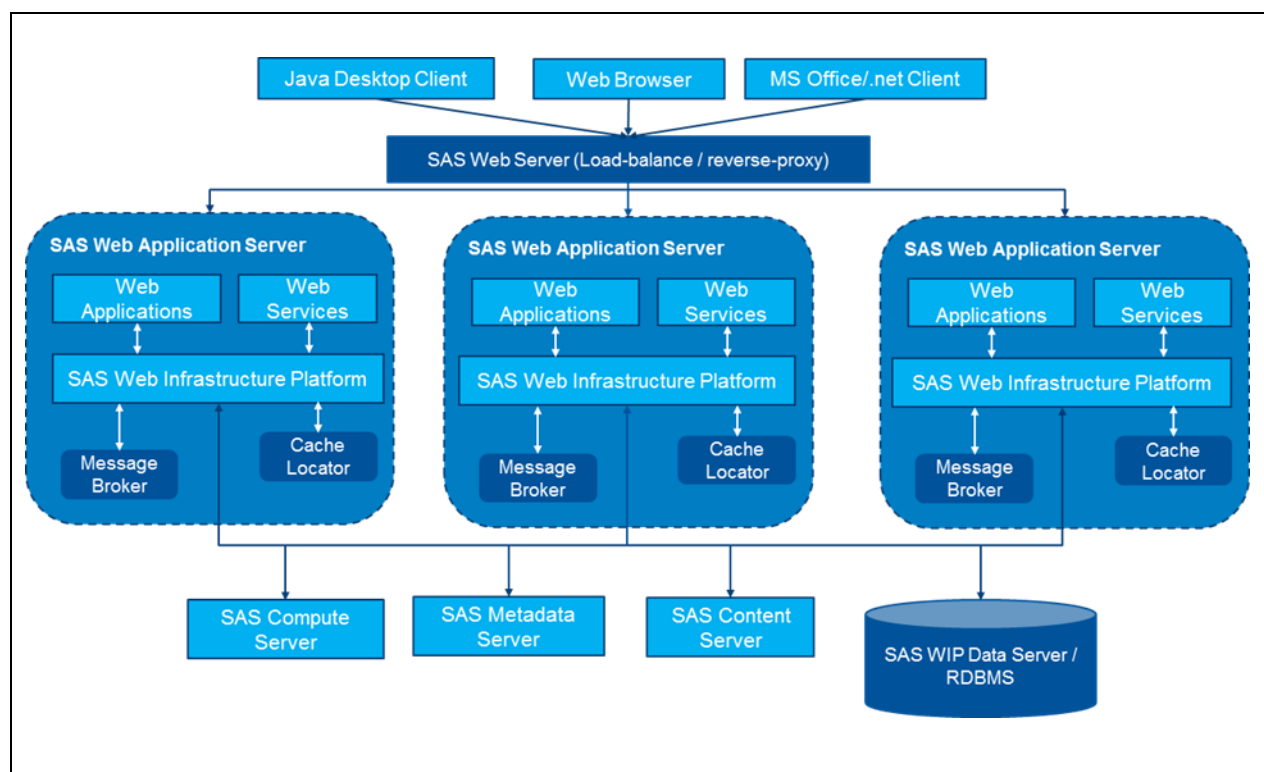
With the extensible architecture of SAS Environment Manager, it is possible to write additional plugins to manage servers, services, and applications in the SAS deployment.

**SAS Web Infrastructure Platform Data Server.** The SAS Web Infrastructure Platform Data Server is a database instance that is used by the SAS web applications to persist data. By default, it is implemented as a single Postgres database instance running on the SAS compute tier. The server contains three individual databases: SharedServices, Administration, and EVManager. By default, the SAS Content Server repository is configured within the SAS Web Infrastructure Platform Data Server's SharedServices database rather than in the file system.

Customers who want to use enterprise database servers instead of the Postgres instance provided by SAS can be accommodated easily by selecting the database of their preference during the deployment process. One exception is that SAS Environment Manager only supports the use of the SAS Web Infrastructure Platform Data Server.

## SAS 9.4 ARCHITECTURE AND MIDDLE-TIER CLUSTERING

The SAS 9.4 enterprise platform uses a multi-tier architecture. A high-level view of the architecture is shown in Figure 1. The client tier consists of browser-based applications, Java desktop applications, and applications running on Microsoft Office. The middle tier has web applications that run in the SAS Web Application Server, which in turn is behind the SAS Web Server. The SAS Web Server can be used to provide load balancing for multiple, clustered SAS Web Application Server instances. It can also be used as the reverse-proxy server when SAS web applications are deployed into multiple SAS Web Application Server instances. The server tier consists of a SAS metadata server and SAS application server components such as SAS workspace servers and SAS stored process servers. The data tier has the SAS Web Infrastructure Platform Data Server, third-party database servers, or both.



**Figure 1. High-Level Architecture of the SAS 9.4 Enterprise Platform**

The SAS middle tier begins with the SAS Web Server, which serves as entry point for all client requests. Because all traffic from the clients to the middle-tier applications is routed through the web server, this becomes the central point for controlling the distribution of traffic to the web application server instances. The web server can act as a load balancer and as both a forward and reverse proxy server. This allows for a great deal of flexibility in handling a very large load, while providing a mechanism for eliminating single points of failure on the middle-tier.

In order to provide greater scalability, availability, and robustness, the SAS Web Application Server supports both vertical and horizontal clustering. With clustering, multiple server instances participate in a load-balancing scheme to handle client requests. Workload distribution is managed by the SAS Web Server. SAS Web Server is configured as a load-balancing HTTP proxy.

The server instances in a cluster can coexist on the same machine (vertical clustering), or the server instances can run on a group of middle-tier server machines (horizontal clustering). The web applications can be deployed simultaneously on both vertical and horizontal clusters.

### **Vertical Clusters**

As previously discussed, a deployment that contains one or more SAS solutions will require that the web application server have access to many gigabytes of memory. If two instances of the web application server are running simultaneously in a vertical cluster, this memory requirement will effectively double. For this reason, it is important to consider hardware sizing constraints when using vertical clusters.

The advantages of vertical clusters include the following:

- If the Java process that underlies one of the instances in the web application server cluster encounters problems that stop the functioning of the web applications, the applications in the other cluster instance will still be able to respond. In this case, it would be possible to stop and restart the web application server that is experiencing problems. Requests would still be serviced by the applications in the other cluster instance. Users who had sessions on the stopped server would lose the session data, but an attempt to reconnect to a clustered application would be successful.
- The load balancer manager (<http://webservermachine.mycompany.com/balancer-manager>) can be used to direct all requests to a single instance of the application, thus “draining” the sessions associated with applications in the other cluster instance. Once an instance is drained, it can be stopped for maintenance without disrupting the service of clustered applications.
- In some cases, the operating system can balance CPU load more effectively if separate Java processes are used.
- Note that if the single machine on which the vertical cluster is deployed experiences an outage, then all the instances in the cluster will be affected. Therefore, the failure of a single machine would cause the application to become unavailable.

### **Horizontal Clusters**

Horizontal clusters are very similar to vertical clusters in that they are comprised of several instances of the same web application server, with the same deployed applications running as different, parallel Java processes. The fundamental difference is that these different web application server instances are on different machines. This allows for more fault tolerance, as well as more flexibility in distributing hardware resources to match the system load.

For example, suppose a customer has two machines, each with 20 gigabytes of memory and two processors (which is approximately equivalent to a single machine with 40 gigabytes and four processors). In this case a horizontal cluster across the two smaller machines would be expected to provide similar performance to that of the larger machine. However, an outage that affects the single, larger machine would cause the web application to become unavailable. In the case of a horizontal cluster, both machines would have to fail simultaneously for the application to become unavailable.

### **Applications That Cannot Be Deployed into a Cluster**

The vast majority of SAS web applications can be deployed into a cluster. However, because of technical limitations, some applications cannot. These exceptions are listed in the *SAS 9.4 Guide to Software Updates*. SAS continues to work to overcome these limitations so that all applications will support clustering in future releases.

## **AUTOMATED CONFIGURATION AND THE BENEFITS OF THE NEW TECHNOLOGY**

Middle-tier components including SAS Web Server, SAS Web Application Server, Cache Locator, and Message Broker can all be automatically deployed and configured. Many server resources (such as data sources and JMS queues) can also be auto-configured, and the SAS web applications are auto-deployed as well. Starting with SAS 9.4, vertical and horizontal clusters can be automatically configured. This automation introduces many additional benefits to our customers.

## SAS WEB SERVER

The benefits afforded by the SAS Web Server include the following:

- The SAS Web Server is automatically configured as a load-balancing HTTP server if the SAS Web Application Server is clustered. This enables fault tolerant, load-balanced deployments right out of the box without the need for manual post-installation configuration steps.
- The SAS Web Server is used and configured as the reverse-proxy server when SAS web applications are deployed into multiple SAS Web Application Server instances.
- The SAS Web Server is automatically configured to cache static web content such as JavaScript files, cascading style sheets, and graphics files. This means the performance benefits are provided at initial deployment, with no manual intervention required.
- The SAS Web Server is automatically configured with session affinity in the case of clustered deployments so that session data is preserved.
- During the initial deployment, you can choose to automatically configure all servers for HTTPS right out of the box. If you plan to use HTTPS, the best practice is to enable the feature during the initial installation and configuration. The SAS Deployment Wizard prompts for a certificate and private key. Both must be in PEM-encoded format. If you have a CA-signed certificate, the SAS Deployment Wizard prompts for the path to the certificate and the path to the RSA private key that is not protected with a pass phrase. Alternatively, you can deploy the software using HTTP during the initial configuration and use the provided instructions to convert to HTTPS.
- Because of the compatibility and close integration with the Apache HTTP server, there is no vendor lock-in.

The following advanced configurations are fully supported, but require documented manual configuration after deployment is complete:

- adding clustered instances of SAS Web Server nodes that eliminate the single point of failure and add capacity
- integrating with customer-supplied load-balancing hardware or software

## SAS WEB APPLICATION SERVER

Although the SAS Web Application Server is based on a commercially available third-party software product, the server is deployed and specifically configured to provide an environment for the SAS web applications and middle-tier platform. The configuration tools that are packaged with the software ease the administration of the server in a SAS environment because the tools are designed to interact with the SAS Metadata Server and other SAS software products to maintain reliability and reduce administration in the SAS deployment.

Because the SAS Web Application Server is dedicated exclusively to running SAS services and applications, it can be tuned very specifically to optimize for specific usage patterns. These optimizations would not be possible if the server had to be generalized in order to run other applications. As described below, this allows for the automation of many tasks that customers had to perform manually in previous versions of SAS.

The following list identifies the automatic configuration features that were inherited from the previous SAS release, as well as some enhancements that are implemented in SAS Web Application Server:

- automatic configuration of multiple managed servers, and distribution of individual web applications into the various server instances for increased performance
- automatic installation and configuration into both vertical clusters and horizontal clusters across multiple machines to provide better performance, scalability, and availability
- automatic connection to Cache Locator on server start-up for distributed communication
- access to the JMS resources provided by JMS Broker
- disabling of automatic directory scanning for changes to files, thus conserving computing resources
- optimized JAR file scanning to reduce start-up times
- packaging of the server together with software that can automate server configuration tasks, thus simplifying the demands for managing a web application server

## **SAS WEB APPLICATIONS AND MULTIPLE SERVER INSTANCE DEPLOYMENT**

The SAS middle tier consists of many different web applications. For example, a typical deployment that contains SAS Visual Analytics consists of approximately three dozen web applications. While capacity planning requires knowledge of the data being analyzed, the types of analysis being performed, expected response times, and the number of customers utilizing the system simultaneously, it is not difficult to estimate that the combined memory demands of all the web applications in a SAS deployment could exceed 15 gigabytes.

Because the JRE is a 64-bit application, a single instance can easily address enough memory to accommodate memory demands. However, care must be taken to ensure that the hardware is sufficient to service the load. If the web application server is deployed on a machine with ample memory and CPUs, then deploying all the web applications into a single server instance is feasible. Alternatively, if the available machines are not sufficiently large to run all the web applications in a single server instance, then electing to distribute the applications among disparate web application server instances is advisable.

During deployment, customers are given the option of deploying all the web applications into a single application server instance or spreading them out among multiple web application server instances. If you select to automatically configure multiple application servers, each of these server instances is created for you. If you choose the automatic deployment option for your SAS web applications, the SAS Deployment Wizard chooses the server on which to deploy each application based on SAS best practices.

For example, in a deployment that includes SAS Visual Analytics and SAS Enterprise Miner, the middle tier would include approximately 40 different web applications. During deployment, one can choose to deploy them into a single web application server instance (the default). In this case, the SAS Deployment Wizard configures a single web application server instance named SAS Server1\_1, all the web applications are deployed into this single instance, and there is one Java process on the machine that services the needs of all these applications.

Alternatively, if one chooses to deploy the application into multiple managed servers, then SAS configures multiple web application server instances with names such as SASServer1\_1, SASServer2\_1 SASServer11\_1, and SASServer13\_1. SAS then distributes the applications among the different server instances according to the specific characteristics of each application, following best practices. The customer can place the different web application server instances on different machines, thus distributing the memory and processing requirements. Note that the exact same applications will be deployed as in the case of a single web application server. It is simply a matter of dividing these applications among the various server instances so that each server instance has a smaller number of applications to service.

## **JAVA MESSAGE SERVICE BROKER**

If, during the SAS configuration process, you elect to perform an automatic configuration of SAS Web Application Server, then the JMS broker is automatically installed and configured for use by the application server. No additional configuration is required. An instance of the broker is installed on the first machine that is used for the SAS middle tier. If you use the SAS Deployment Wizard to configure an additional middle-tier node on another machine, then those server instances are configured with connection information for the broker.

## **CACHE LOCATOR**

In a single-machine deployment (with the middle tier and the server tier on the same machine), only one locator is installed by the SAS Deployment Wizard. The SAS Web Application Server uses this single locator instance. If more than one SAS Web Application Server instance is configured on this single machine (creating a vertical cluster), each instance uses the one locator to learn about the other server instances when forming the cache.

In multiple-machine deployments, a locator is installed on the first middle-tier machine by the SAS Deployment Wizard. A locator is also installed on the first server-tier machine that includes an instance of SAS Web Infrastructure Platform Scheduling Services.

## **ENTERPRISE INTEGRATION**

### **SUPPORT FOR INTEGRATION WITH ENTERPRISE WEB SERVERS**

Some network topologies already have a web server that is used to proxy connections. In these deployments, you can reconfigure the SAS middle tier so that it interacts with the existing web server. It is simplest to keep SAS Web Server in the deployment so that it can continue to load balance connections to a SAS Web Application Server cluster. However, the *SAS 9.4 Intelligence Platform: Middle-Tier Administration Guide* provides the list of specific steps that are required to bypass the SAS Web Application Server for those customers who want to use their existing web server and load balancing infrastructure.

## SUPPORT FOR WEB-BASED AUTHENTICATION

By default, SAS web applications use the host-based authentication that is provided by the SAS Logon Manager application. When credentials are provided to SAS Logon Manager, the credentials are sent to the SAS Metadata Server for authentication. The metadata server then authenticates the credentials against the chosen authentication provider. The default provider is the host operating system.

As an alternative, you can configure the SAS web applications to authenticate on the middle tier. Then, when users logon to a SAS web application, SAS Web Application Server handles the initial authentication using container-managed security.

Performing web authentication facilitates single sign-on. Most likely, your organization has several applications behind a common set of reverse proxy and HTTP servers. When a common server handle authentication is in place, users do not need to re-authenticate for access to each application.

The process of converting from host-based authentication to web-based authentication is not currently automated. Instructions and support are provided to assist customers in performing this conversion. See <http://support.sas.com/documentation/cdl/en/bimtag/66823/HTML/default/viewer.htm#n1bhp608f0hsoen10i1vi0p9l5f7.htm>.

## SUPPORT FOR ALTERNATE AUTHENTICATION MECHANISMS

For customers who choose web-based authentication, the *SAS 9.4 Intelligence Platform: Middle-Tier Administration Guide* provides step-by-step instructions for integrating with a variety of authentication providers, including the following:

- Integrated Windows Authentication (IWA) is a Microsoft technology that is used in environments where users have Windows domain accounts. With IWA, the credentials (user name and password) are hashed before being sent across the network. The client browser proves its knowledge of the password through a cryptographic exchange with the web application server.
- IBM Tivoli Access Manager WebSEAL IBM WebSEAL is an important component of IBM Tivoli Access Manager. Access Manager provides complete authorization, network security, and policy management functions. It is used for end-to-end protection of resources over intranets and extranets. WebSEAL is a high-performance, multi-threaded web server that applies fine-grained security policy to resources in the protected web object space of Tivoli Access Manager. WebSEAL can provide single sign-on solutions and incorporate back-end web application server resources into its security policy.
- CA SiteMinder is widely used as a user authentication and single sign-on solution. It also provides policy-based authorization, identity federation, and auditing to the protected web applications. SAS 9.4 supports CA SiteMinder through the use of both the SAS Web Server and the SAS Web Application Server.

## MANAGE SAS ENVIRONMENTS

In releases previous to SAS 9.4, customers use various tools to manage SAS software. Operating system (OS) shell commands are used to manage the OS platforms; SAS Management Console is used to manage the server tier; and web application server-specific administrative tools (such as WebSphere Admin Console) are used to manage the middle tier. For the applications, each application has its own management functions that are typically exposed through its web application user interface.

With the introduction of SAS Environment Manager in SAS 9.4, we would like to provide a single management environment that can manage all SAS application stacks.

SAS Environment Manager installs an agent process on each server-tier and middle-tier machine in the deployment. Each agent gathers performance metrics and transfers the data to a server process that runs on a middle-tier machine. The server process includes a web application server that provides a web-based administrative interface. Administrators use a web browser to monitor and manage numerous components in the SAS environment. See Figure 2 for an example of the metrics that are made available via the SAS Environment Manager.

While the management feature of SAS Environment Manager enables customers to monitor the run-time metrics of the SAS Web Server and SAS Web Application Server, the administration features in the web interface enable tuning of these servers to optimize performance.

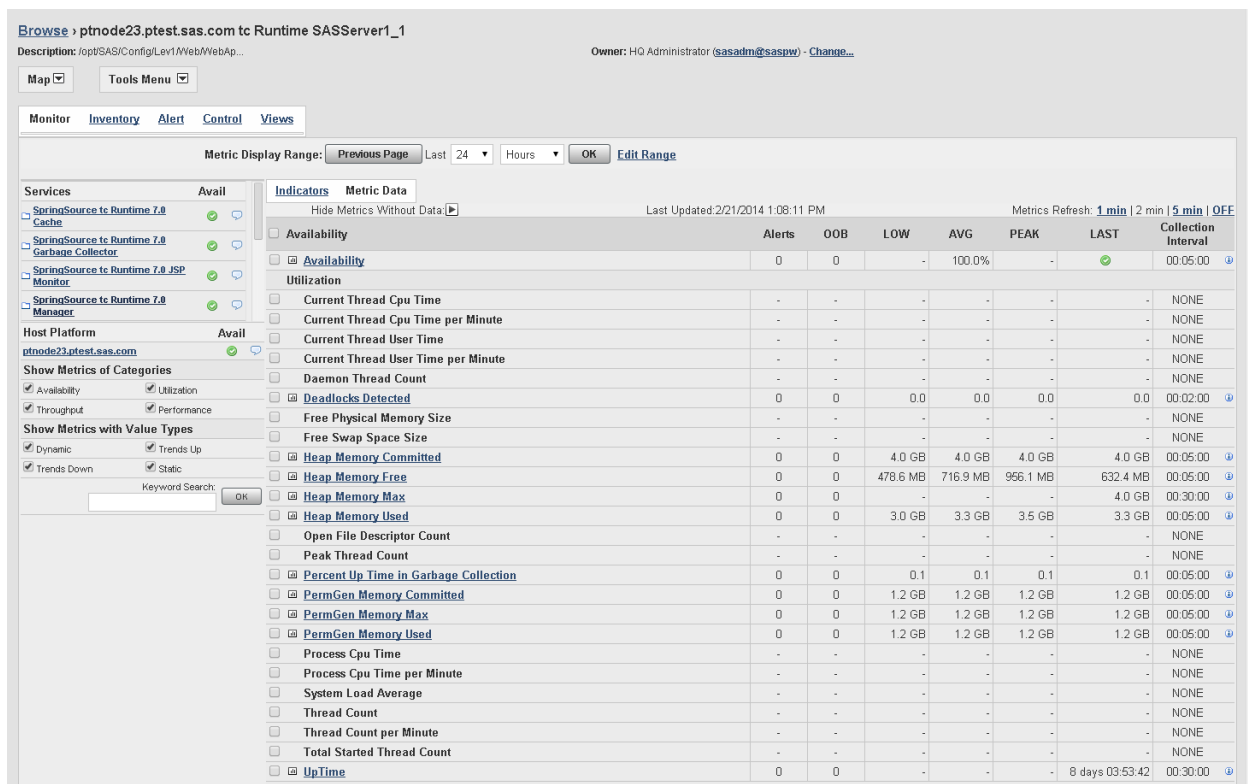
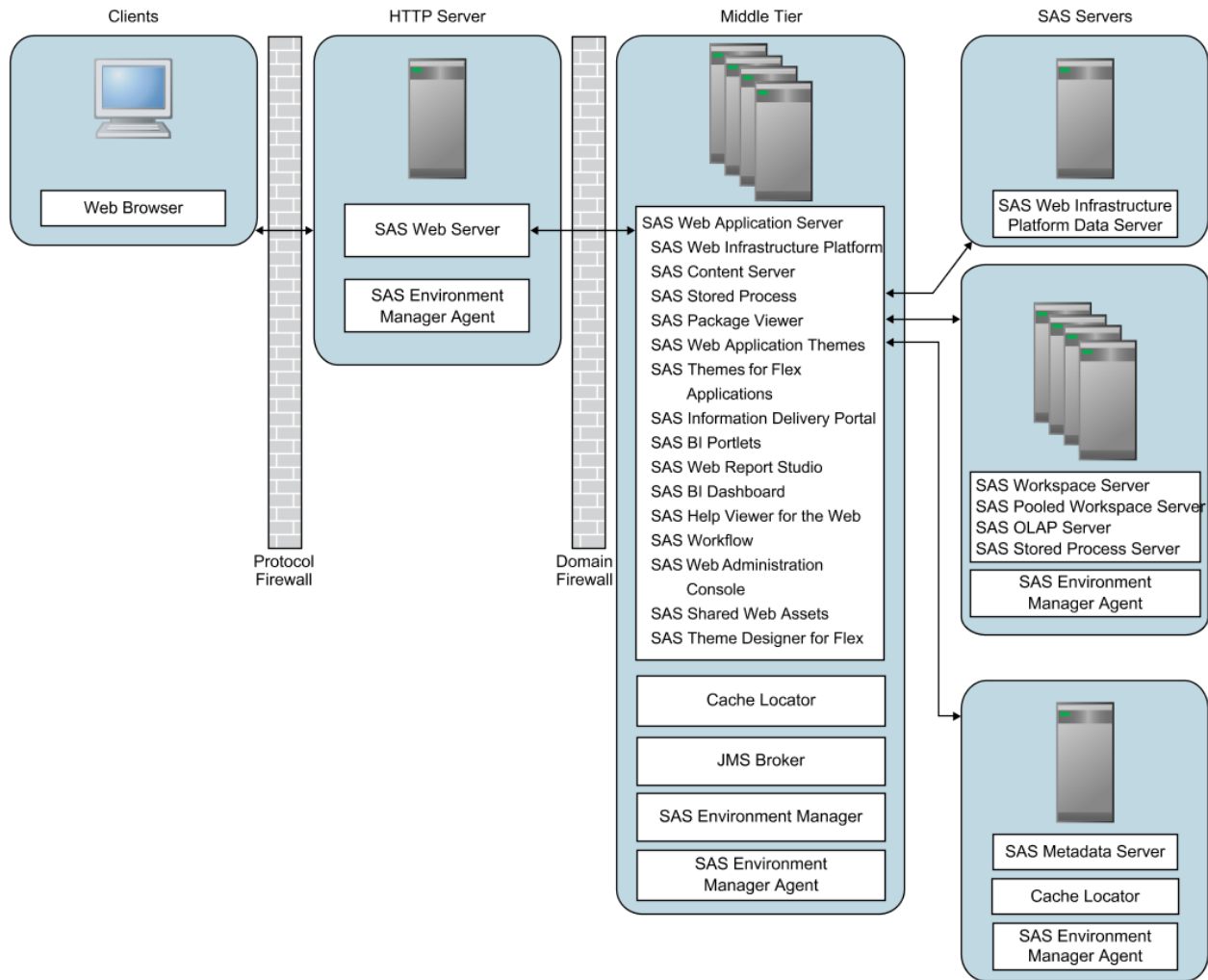


Figure 2. Sample of Environment Manager Page Showing SAS Web Server Performance Metrics

## EXAMPLE ARCHITECTURE

Let us use an example deployment to further demonstrate the SAS 9.4 architecture and its load-balancing implementation. This deployment has a cluster of web application servers and deploys SAS Web Server on its own machine.

Figure 3 illustrates the example topology. In most cases, the SAS Web Application Server instances and applications are configured identically. However, as stated previously, some applications such as SAS BI Dashboard Event Generator, as well as some SAS solution applications, cannot be clustered. These are examples of situations in which the server instances and applications are not identically configured.



**Figure 3: Example Clustered SAS Web Application Topology**

Most aspects of this topology can be configured automatically with the SAS software. Here are the advantages and disadvantages of this topology:

Topic	Advantages	Disadvantages
Security	<p>The SAS web applications and the web application server cluster are protected by firewalls.</p> <p>The web application server and SAS web applications can be configured to perform web authentication for single sign-on to the applications and other web resources in the network.</p>	None
Performance	<p>Response time is improved because static content is cached by the SAS Web Server.</p> <p>The greater computing capacity of the web application server cluster also improves performance.</p>	None
Scalability	Once the cluster is established, additional server instances can be added to support larger numbers of	None

	concurrent users.	
Availability	<p>Clustering provides fault isolation that is not possible with a single web application server. If a machine in the cluster fails, then only the users with active sessions on that machine are affected.</p> <p>You can plan downtime for maintenance by taking some servers offline. While maintenance is performed, new requests are directed to the applications deployed on the remaining servers.</p>	SAS Web Server remains a single point of failure. Software and hardware high-availability options exist to mitigate this disadvantage.
Maintainability	Configuration and deployment of the cluster and the applications can still be automated with the SAS Deployment Wizard.	Some operations, such as redeploying web applications, can require more effort when more machines are used.

**Table 1. Example Deployment Advantages and Disadvantages**

## Load Balancing

Consider the architecture above with the SAS Web Server deployed on one machine and the SAS web applications deployed in both vertical and horizontal clusters on a two different machines. We now have the same set of web applications deployed a total of four times each. Each of these individually deployed apps would have a different URL. For example, the SAS Visual Analytics Explorer application would be available via the following four URLs:

- <http://WebAppSrv1:Port1/SASVisualAnalyticsExplorer>
- <http://WebAppSrv1:Port2/SASVisualAnalyticsExplorer>
- <http://WebAppSrv2:Port1/SASVisualAnalyticsExplorer>
- <http://WebAppSrv2:Port2/SASVisualAnalyticsExplorer>

As noted previously, all traffic to instances of the SAS Web Application Server comes through the SAS Web Server. Therefore, the web server needs to be aware of all the deployed instances of each application so that the requests can be forwarded to the corresponding applications. Therefore, users would expect to find the SAS Visual Analytics Explorer application at the following URL:

<http://WebServer/SASVisualAnalyticsExplorer>

This single URL is possible because in the SAS 9.4 middle-tier architecture, SAS controls the configuration of both the web server and the web application server. SAS can alter these configurations as new software is added. When the web server is first deployed, the load balancing infrastructure is initialized with the expectation that SAS web applications will be configured later. Then, as web applications are deployed either on the same machine or on subsequent machines in the deployment, proxy entries are added to the following configuration file:

`/SASConfig/Lev1/Web/WebServer/conf/sas.conf`

The added entries contain the URLs for each deployed instance of a SAS web application. If web applications are deployed on other machines, then the deployment scripts on those machines direct the deployment agent on the SAS Web Server machine to make the appropriate entries.

In this use case, once the deployment process is complete on all three machines (WebServer, WebAppServer1, and WebAppServer2), the following entries would be visible in the `sas.conf` file:

```
ProxyPass /SASVisualAnalyticsExplorer balancer:// WebAppServer1_Cluster/SASVisualAnalyticsExplorer
ProxyPassReverse /SASVisualAnalyticsExplorer balancer:/
WebAppServer1_Cluster/SASVisualAnalyticsExplorer
[...lines deleted for brevity...]
<Proxy balancer:// WebAppServer1_Cluster12>
```

```
BalancerMember http://WebAppSrv1:8080 route=ee4bd_SASServer12_1
BalancerMember http:// WebAppSrv1:8081 route=ee4bd_SASServer12_2
BalancerMember http:// WebAppSrv1:8080 route=ee4be_SASServer12_1
BalancerMember http:// WebAppSrv1:8081 route=ee4be_SASServer12_2
ProxySet scolonpathdelim=on stickysession=ee47373439506acc86f4e80bbcc316bd_Cluster
</Proxy>
```

**Example: Extract from the sas.conf File Showing Load Balancing Entries**

## CONCLUSION

SAS has discontinued support for a variety of middle-tier platforms in favor of a single standard platform. Through this platform, SAS provides a turn-key solution that SAS delivers, configures, tunes, and maintains. This approach provides myriad benefits including the ability to automate a vast number of deployment tasks that customers formerly had to perform manually. These tasks are now part of the automated deployment process. This approach also provides SAS customers with a more streamlined support process, because all the software that is required to use SAS software products is provided in-house by SAS.

## REFERENCES

- SAS Institute Inc. 2013. *SAS 9.4 Intelligence Platform: Middle-Tier Administration Guide, Second Edition*. Cary, NC: SAS Institute Inc. Available at <http://support.sas.com/documentation/cdl/en/bimtag/66823/PDF/default/bimtag.pdf>.
- SAS Institute Inc. 2013. *SAS 9.4 Intelligence Platform: Security Administration Guide, Second Edition*. Cary, NC: SAS Institute Inc. Available at <http://support.sas.com/documentation/cdl/en/bisecag/67045/PDF/default/bisecag.pdf>.
- SAS Institute Inc. 2013. *SAS 9.4 Intelligence Platform: Installation and Configuration Guide*. Cary, NC: SAS Institute Inc. Available at <http://support.sas.com/documentation/cdl/en/biig/63852/PDF/default/biig.pdf>.
- SAS Institute Inc. 2013. *SAS 9.4 Guide to Software Updates*. Cary, NC: SAS Institute Inc. Available at <http://support.sas.com/documentation/cdl/en/whatsdiff/66129/PDF/default/whatsdiff.pdf>.
- Li, Zhiyong, and Roda, Mike, 2014. "An Advanced Fallback Authentication Framework for SAS® 9.4 and SAS® Visual Analytics." *Proceedings of the SAS Global Forum 2014 Conference*. Cary, NC: SAS Institute Inc.
- Peters, Amy, Bonham, Bob, and Li, Zhiyong, 2013. "Monitoring 101: New Features in SAS 9.4 for Monitoring Your SAS Intelligence Platform." *Proceedings of the SAS Global Forum 2013 Conference*. Cary, NC: SAS Institute Inc. Available at: <https://support.sas.com/resources/papers/proceedings13/463-2013.pdf>.
- Doninger, Cheryl, Li, Zhiyong, and Wolfe, Bryan, 2014. "Best Practices for Implementing High Availability for SAS® 9.4." *Proceedings of the SAS Global Forum 2014 Conference*. Cary, NC: SAS Institute Inc.
- Siooss, Rob, and Li, Zhiyong, 2014. "SAS® 9.4 Web Application Performance: Monitoring, Tuning, Scaling, and Troubleshooting." *Proceedings of the SAS Global Forum 2014 Conference*. Cary, NC: SAS Institute Inc.
- VMWare Inc. (2013). VMware vFabric 5 Documentation. Available at <http://pubs.vmware.com/vfabric5/index.jsp>.

## CONTACT INFORMATION

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

Zhiyong Li  
10 SAS Campus Drive  
Cary, NC 27513  
SAS Institute Inc.  
[Zhiyong.Li@sas.com](mailto:Zhiyong.Li@sas.com)  
<http://www.sas.com>

Alec Fernandez  
100 SAS Campus Drive  
Cary, NC 27513  
SAS Institute Inc.  
Alec.Fernandez@sas.com  
<http://www.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.