Deployment Guide

SAP NetWeaver
SAP Enterprise SOA
Deployment Guide

A Step-by-Step Technical Guide

Deployment Guide

Notice:

The information in this publication is subject to change without notice.

THIS PUBLICATION IS PROVIDED “AS IS” WITHOUT WARRANTIES OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING ANY WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NONINFRINGEMENT. CITRIX SYSTEMS, INC. (“CITRIX”), SHALL NOT BE LIABLE FOR TECHNICAL OR EDITORIAL ERRORS OR OMISSIONS CONTAINED HEREIN, NOR FOR DIRECT, INCIDENTAL, CONSEQUENTIAL OR ANY OTHER DAMAGES RESULTING FROM THE FURNISHING, PERFORMANCE, OR USE OF THIS PUBLICATION, EVEN IF CITRIX HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES IN ADVANCE.

The exclusive warranty for Citrix products, if any, is stated in the product documentation accompanying such products. Citrix does not warrant products other than its own.

Table of Contents

Introduction 4
Solution Requirements....5
Prerequisites....5
Network Diagram 6
First time connectivity 8
Serial Connection 8
Ethernet Connection....8
NetScaler Configuration....9
Deployment Model: One-Arm, LB, SSL Offload, Caching, Compression, Re-write. 9
Licensing 10
Features 11
Modes 11
High Availability....12
IP Addresses, Interfaces and VLANs....15
Load Balancing Configuration....18
Create Server Objects....18
Create Service Groups....19
Create LB Virtual Server Objects (VIPs) 21
Load Balancing Methods & Persistence....22
SSL Offload Configuration 24
Keys and Certificates....24
Using the SSL Certificate Wizard 24
Importing the SAP Portal server certificate 25
Create Server Objects....26
Create Service Groups....27
Create SSL Virtual Server Objects (VIPs)....30
SSL Load Balancing Methods & Persistence 31
Caching....34
Caching for SAP Applications 34
Compression....36
Compression for SAP Applications 36
SAP Application non-compressible content types....37
SAP Application compressible content types 37
Citrix automatically compressed content types 37
Configuring Compression for SAP Application 38
Disabling Compression on SAP Application Responses....42
Removing Accept-Encoding headers....42
Disabling compression on the Citrix VIP's....43
Rewrite....44
Rewrite for SAP Applications 44
Rewrite for SAP Composite Application Framework....46
Testing the Composite rewrite connection....49
Rewrite for SAP ERP 50
Testing the ERP rewrite connection....53
Troubleshooting....54
Load Balancing....54
Run a trace....54
Run a trace - on SAP Portal....55
Appendix A - NetScaler Application Switch Configuration....56

Introduction

Citrix® NetScaler® optimizes the delivery of web applications — increasing security and improving performance and Web server capacity. This approach ensures the best total cost of ownership (TCO), security, availability, and performance for Web applications. The Citrix NetScaler solution is a comprehensive network system that combines high-speed load balancing and content switching with state-of-the-art application acceleration, layer 4-7 traffic management, data compression, dynamic content caching, SSL acceleration, network optimization, and robust application security into a single, tightly integrated solution. Deployed in front of application servers, the system significantly reduces processing overhead on application and database servers, reducing hardware and bandwidth costs.

The SAP Enterprise Service Oriented Architecture (SOA) provides a blueprint for services-based, enterprise scale business solutions that are adaptable, flexible, and open. Enterprise Services Architecture takes the concept of service-oriented architecture to a new level by transforming Web services into enterprise services. The SAP NetWeaver® platform provides you with the ability to implement Enterprise Services Architecture tailored to your specific needs at your own pace. SAP is evolving all its solutions to be compliant with the Enterprise Services Architecture blueprint.

Building new, customized solutions that support innovation is expensive and time-consuming because leveraging the functionality of your existing packaged applications is extremely difficult. Bringing Citrix and SAP Enterprise Services Architecture together reduces the dependence on customized applications, and increases flexibility and reduces time to deployment while reducing operational expenses.

This deployment guide was created out of a joint engagement between Citrix and SAP at the Co-Innovation Laboratory in Palo Alto, California, USA. This deployment guide walks through the step-by-step configuration details of how to configure the Citrix NetScaler for use as front-end to SAP Portal for end-user traffic, that is HTTP ~ HTML. To further complement the value of the Enterprise SOA, this guide walks through the details of how to configure the Citrix NetScaler for use as a front-end to the SAP Composite Application Framework and SAP ERP Web Services platforms, providing High Availability, a flexible load balancer and HTTPS encryption point for machine to machine web service traffic, capabilities our competitors still can't live up to. With this deployment Citrix becomes an integral and flexible part of the SAP Enterprise SOA "applistructure" bringing together applications and technology for a fast, flexible and highly effective service oriented IT infrastructure.

Note:

The recommendations in this guide are specific to an SAP deployment and the policies therefore come at the recommendation from SAP. These configurations might deviate from the default or standard Citrix Application Switch configurations and are to be considered as a guideline and reference for SAP Applications, and not the de-facto standard for Citrix Application Switches.

Solution Requirements

- Application Delivery Front-End for SAP Portal, SAP Composite Server, and SAP ERP Server
  - Load Balancing - with tcp multiplexing
  - SSL Offload - using non-standard ports 50001 & 50201
  - Compression - for compressible objects over 8k
  - Caching - following caching rules of SAP servers
  - Re-Write for SOAP:XML, both http & https
  - SAP Requited TCP/IP Port Numbers - https://www.sdn.sap.com/irj/sdn/go/portal/prtroot/docs/library/uuid/4e515a43-0e01-0010-2da1-9bcc452c280b

Prerequisites

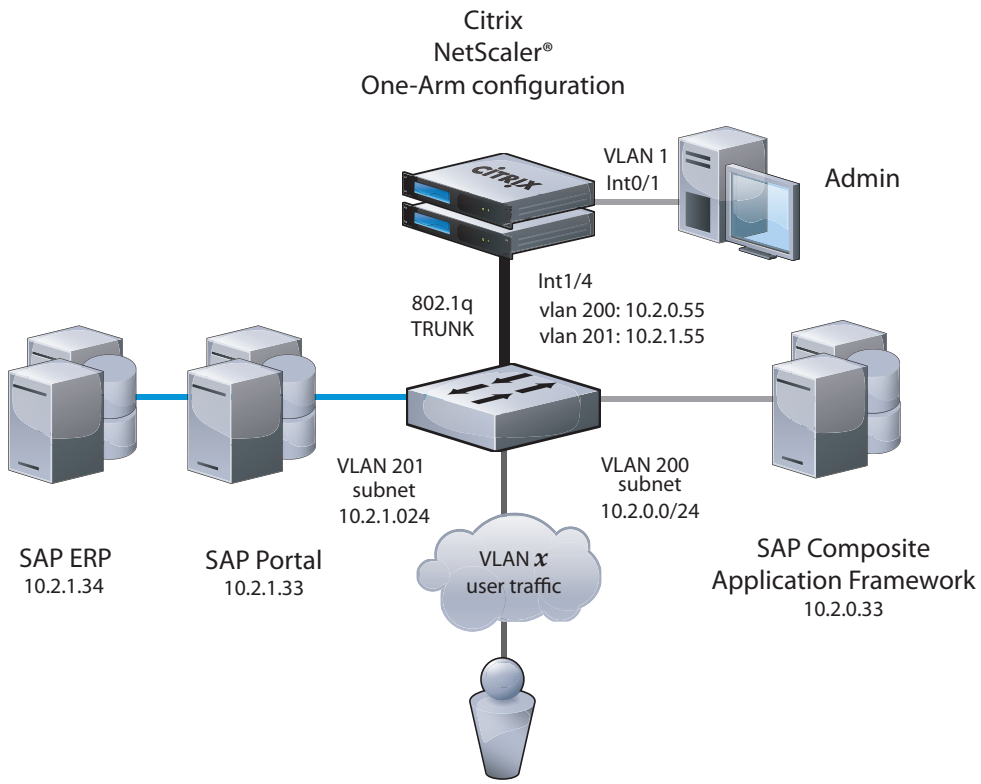
- Citrix NetScaler L4/7 Application Switch, running version 8.0+, (Quantity x 1 for single deployment, Quantity x 2 for HA deployment).
- Layer 2/3 switch, w/support for 802.1q VLANs, (Quantity x 1)
- Client laptop/workstation running Internet Explorer 6.0+, Ethernet port
- 9-pin serial cable -or- USB-to-serial cable
- SAP NetWeaver 7.1 SP3
- SAP Composite Application Framework Environment 7.1
- SAP Enterprise Resource Planning application (ERP) 7.1

Network Diagram

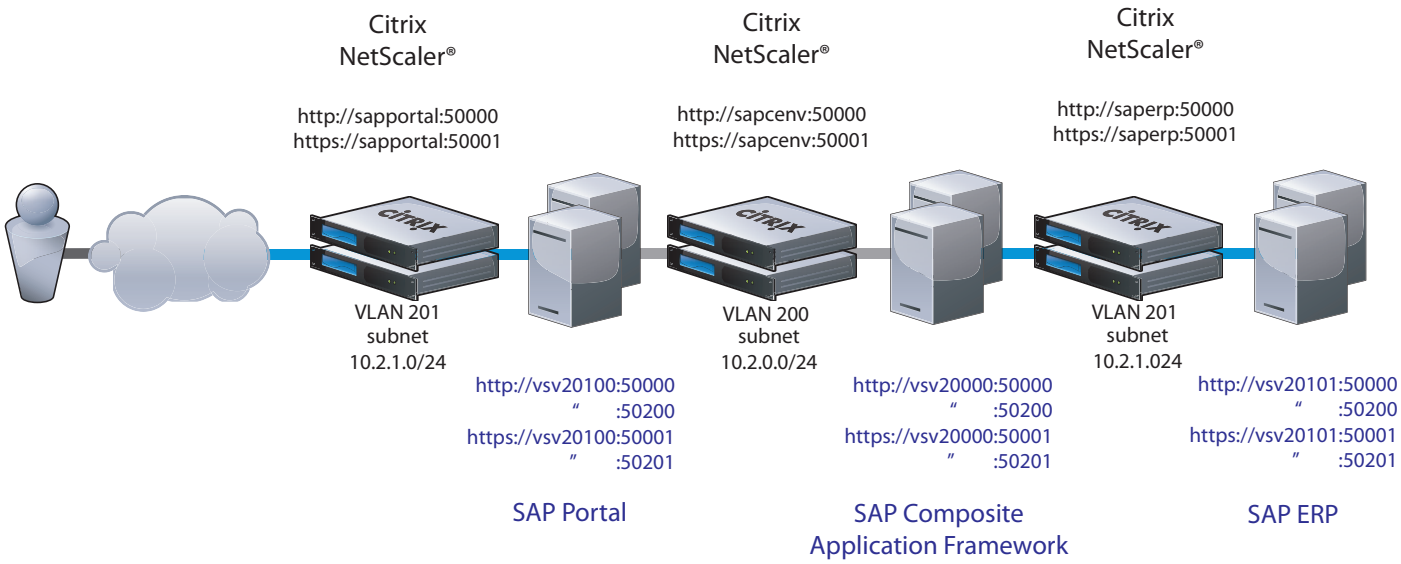
The following is the Network that was used to develop this deployment guide, and is representative of the solution developed at SAP Co-Innovation Lab in Palo Alto, California, USA.

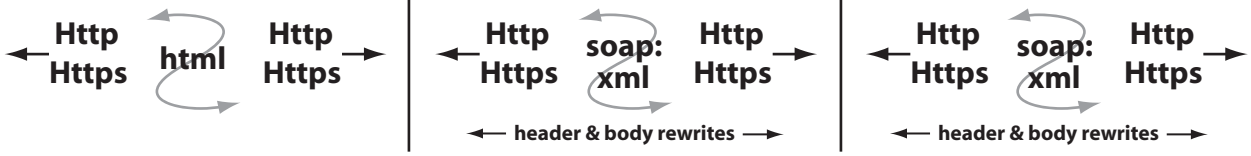
VLAN Legend	Primary NetScaler	Primary/Secondary NetScaler	Secondary NetScaler
VLAN 1 VLAN 200 VLAN 201 802.1q TRUNK	IP Addresses: NSIP: 169.145.91.205 / 24	Shared IP Addresses: VIP: 10.2.1.53 / 24 VIP: 10.2.0.53 / 24 VIP: 10.2.1.54 / 24 VLAN 200: Interface 1/4, Tagged SNIP: 10.2.0.55 / 24 VLAN 201: Interface 1/4, Tagged SNIP: 10.2.1.55 / 24 VLAN 4: Interface 1/4, Untagged VLAN 1: (Mgmt) Interface 0/1, Untagged SNIP: 169.145.91.207 / 24	IP Addresses: NSIP: 169.145.91.206 / 24

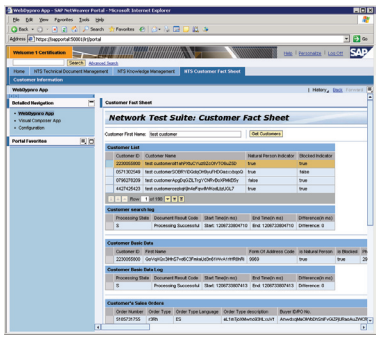
Citrix / SAP Enterprise SOA
Physical Network Diagram

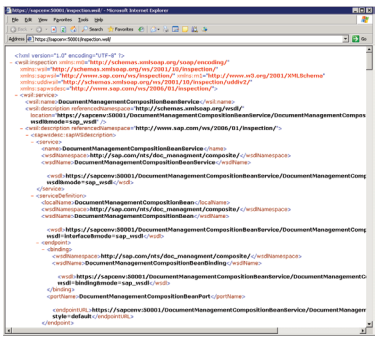


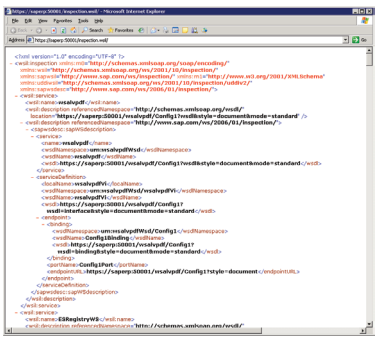
Citrix / SAP Enterprise SOA Logical Network Diagram

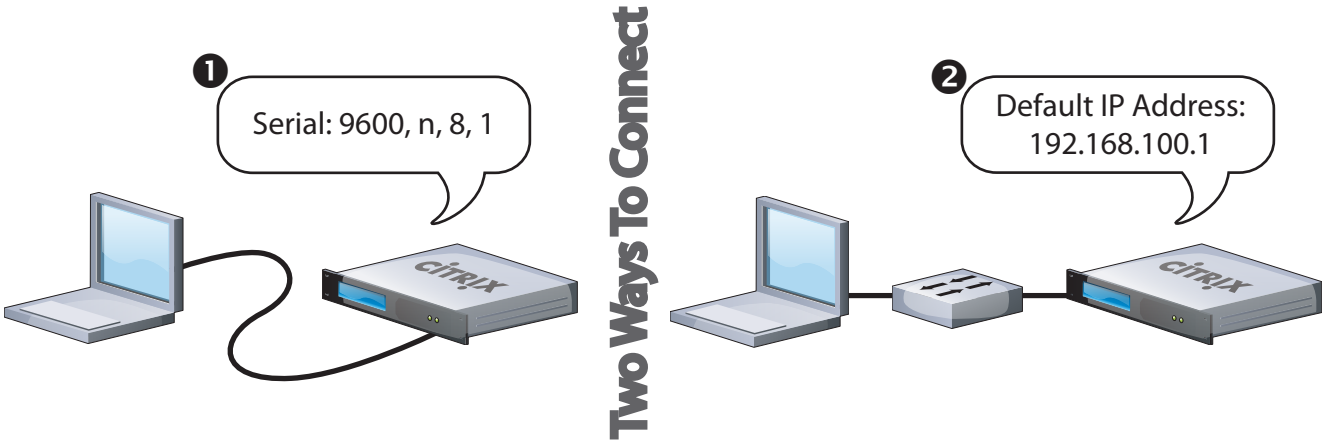












First time connectivity

Serial Connection

The NetScaler can be accessed by the serial port through any terminal emulation program. Windows Hyperterm is commonly used on a laptop or workstation. Connect a 9-pin Null Modem cable (or USB-to-9-pin cable) from the computer to the NetScaler's console port. In the terminal emulation program configure the settings for 9600 baud, No stop bits, 8 data bits, and 1 parity bit. The login prompt should appear. The default login is nsroot, nsroot. It is advisable to change the nsroot password once connected.

Once connected type in the CLI command 'configns' ('nsconfig' if at the shell prompt). Select option 1 to change the NetScaler IP Address and Network Mask. Exit, save and reboot.

Ethernet Connection

The NetScaler can also be accessed by the default IP Address of 192.168.100.1, either through an http, https, telnet or ssh connection. Once connected, the login prompt should appear. The default login is nsroot, nsroot. It is advisable to change the nsroot password once connected.

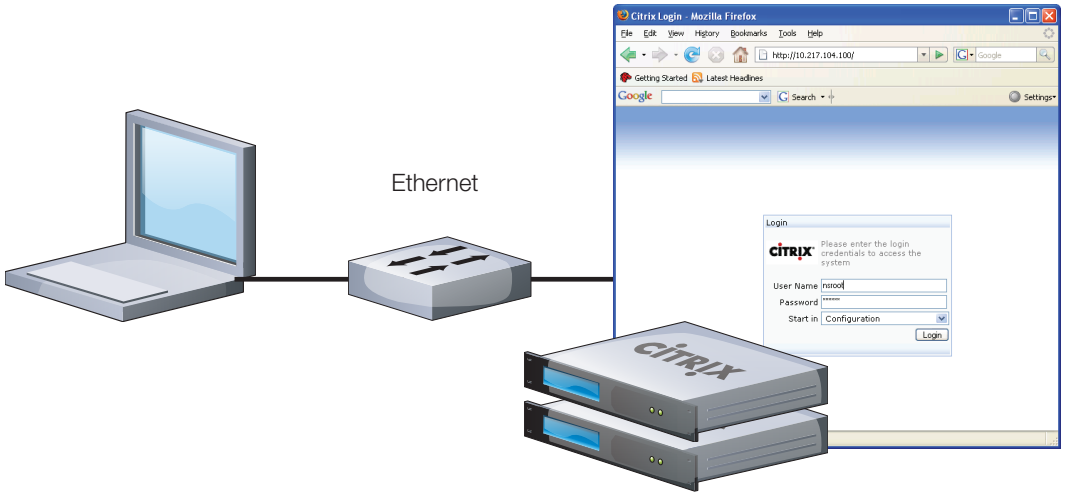
Type in the CLI command 'configns' ('nsconfig' if at the shell prompt). Select option 1 to change the NetScaler IP Address and Network Mask. Exit, save and reboot.

Note: Changing the NetScaler IP Address always requires a reboot.

NetScaler Configuration

Deployment Model: One-Arm, LB, SSL Offload, Caching, Compression, Re-write.

The NetScalers in this example will be deployed as a high availability pair, in one-arm mode. Always start with the first NetScaler. Once the initial NetScaler IP Address (NSIP) has been configured, you can connect to both the Primary and Secondary NetScalers via a http or https web browser connection.



Connect to the NetScaler via the NSIP using a web browser.

In this example:
NS1: http://169.145.91.205
NS2: http://169.145.91.206

Note: Java will be installed.

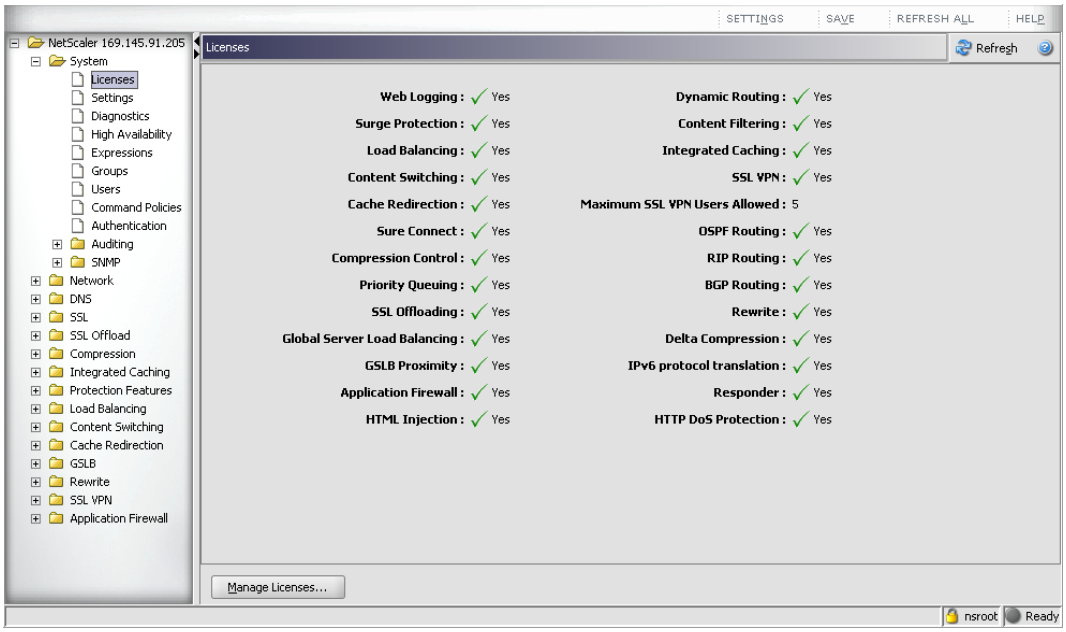
Default login is: nsroot, nsroot.

Licensing

The availability of a feature is controlled by a license key. When using the system for the first time, you need to load the license key and then enable the feature.

To add new licenses.

From the GUI, navigate to NetScaler ➥ System ➥ Licenses ➥ Manage Licenses.



Note:

Licenses are tied to the hostname of the switch and must match. The hostname can be found under NetScaler ➥ System. Make sure the license file is in the correct location. With release 8.0 all license files must be in the /nsconfig/license directory in order to be recognized.

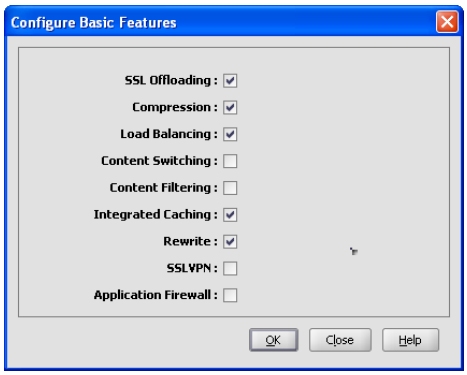
Also, check the “hosts” files in /nsconfig and in /etc, and make sure both include lines for localhost and for the NetScaler hostname as defined in the configuration and /nsconfig/rc.conf.

A properly configured hosts file should look similar to the following (using nshost as the example hostname defined for this NetScaler).

127.0.0.1 localhost
127.0.0.1 nshost

Features

Before configuring the Integrated Caching, Compression, SSL Offloading and Load Balancing features on the system, be sure to enable them. This is important for features such as compression, as the policies won't get applied unless the feature has been enabled beforehand.

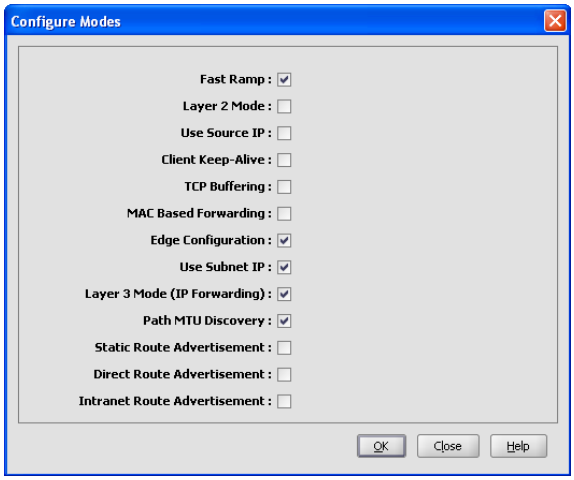


To enable basic features.

From the GUI, navigate to NetScaler ➥ System ➥ Settings ➥ Basic Features.

Modes

Other 'modes' that are important for performance and load balancing are also applied in this section such as Use Subnet IP (USNIP), Client Keep-Alive and TCP Buffering.



To enable modes.

From the GUI, navigate to NetScaler ➥ System ➥ Settings ➥ modes.

High Availability

In a High Availability deployment, one Application Switch actively accepts connections and manages servers, while the second monitors the first. If the first Application Switch quits accepting connections for any reason, the second Application Switch takes over and begins actively accepting connections. This prevents downtime and ensures that the services provided by the Application Switch will remain available even if one Application Switch ceases to function.

Important Considerations for NetScaler High Availability

- The passwords for both NetScalers 'nsroot' account must match. You must change these manually on the switches, they are not synchronized.
- The maximum node ID for Application Switches in an HA pair is 64.
- Both NetScaler HA peers must be running the same version of code.
- The configuration files in 'ns.conf' must match on both NetScalers. For this to happen, the following must occur:
  - » The primary and secondary NetScaler Application Switches must be configured with their own unique NSIP's.
  - » The 'node id' and 'IP Address' of one Application Switch must point to the other Application Switch (it's HA peer).
  - » You must configure RPC node passwords onto both Applicaiton switches. Initially, all Application Switches are configured with the same RPC node password. To enhance security, you should change these default RPC node passwords.

1. While connected to the Primary NetScaler, add the Secondary node.

   In the NetScaler GUI, navigate to: NetScaler ➥ System ➥ High Availability ➥ Add.

   Enter the Node ID and IP address for the Secondary HA peer.

   In this example:
   '2', and 169.145.91.206.

2. Connect to the Secondary NetScaler and tell it to take the Secondary role.

3. Navigate to NetScaler ➥ System ➥ High Availability ➥ Open ➥ "Stay Secondary".

   Connect to the Secondary NetScaler and add the Primary node.

   Enter the Node ID and IP address for the Primary HA peer.

   In this example:
   '1', and 169.145.91.205.

Note:

It is important to turn 'Off' HA Monitoring on interfaces that it is not intended for, otherwise HA Node Synchronization will not be successful.

In the NetScaler GUI: Navigate to NetScaler > Network > Interfaces.

Double-click the interface number(s), and turn 'Off' HA Monitoring.

High Availability Command Synchronization

In a correct HA setup, any command issued on the primary Application Switch will propagate automatically to the secondary Application Switch. Some reasons why command synchronization may not work:

- Network connectivity is down
- Resources are not available on the Secondary Application Switch
- Authentication failure, (nsroot and/or rpc node)
- HA Monitoring is not turned 'On', 'Off' on same interfaces for both nodes

TIP: Disabling the blinking LCD Panel

The LCD panel on the front of the NetScaler will flash intermittently until the unused interfaces are disabled and HA monitoring is turned off on them. In the GUI, Navigate to NetScaler > Network > Interfaces. Select an interface, right-click to disable. Right-click to Open, and disable HA monitoring.

4. Both Primary and Secondary must be configured to Actively participate in HA.

   In the NetScaler GUI on the Primary: Navigate to NetScaler ➥ System ➥ High Availability ➥ ID 0 ➥ Open. Select HA Status 'Enabled'. Enable HA Synchronization. Enable HA Propagation. Click 'Ok'.

   Repeat for Secondary.

5. A successful HA Synchronization can be viewed from the High Availability screen on either the Primary or Secondary node's GUI.

   From the same screen you can 'Force Synchronization' or 'Force Failover'.

Important NetScaler IP Addresses

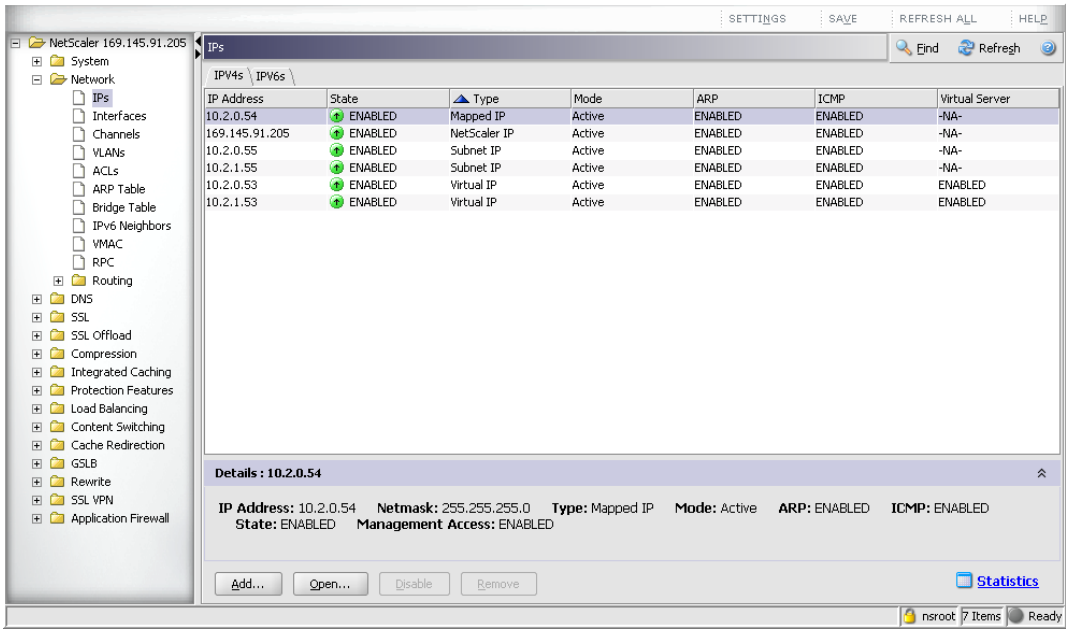
Acronym	Description	Usage
Note: NSIP is Mandatory and requires a reboot.		
NSIP	NetScaler IP Address	The NetScaler IP (NSIP) is the management IP address for the appliance, and is used for all management related access to the appliance. There can only be one NSIP.
SNIP	Subnet IP Address	The Subnet IP address (SNIP) allows the user to access an Application Switch from an external host that is residing on another subnet. When a subnet IP address is added, a corresponding route entry is made in the route table. Only one such entry is made per subnet. The route entry corresponds to the first IP address added in the subnet. In addition, the Application Switch uses the SNIP as the source IP Address for outgoing packets, when the "USNIP" mode is enabled. USNIP is enabled by default. (With USNIP enabled, this removes the necessity of configuring a MIP, thus saving the additional IP Address for other uses). This can also be used as the Tagged VLAN IP. You can use all the Subnet IP addresses as Mapped IP addresses.
MIP	Mapped IP Address	The mapped IP address (MIP) is used by the Application Switch to represent the client when communicating with the backend managed server. Mapped IP addresses (MIP) are used for server-side connections and Reverse NAT. Think of this as the client's source address on the server-side of the Application Switch, assuming a two-arm proxy deployment. Think of it as the Tagged VLAN IP. When using the USNIP mode above, MIP's are unnecessary.
VIP	Virtual IP Address	The Virtual Server IP address (VIP) is used by the Application Switch to represent the public facing ip address of the managed services. ARP and ICMP attributes on this IP address allow users to host the same vserver on multiple Application Switches residing on the same broadcast domain.
DFG	Default Gateway	IP Address of the router that forwards traffic outside of the subnet where the appliance is installed.

Note:

If both USIP mode and USNIP mode are enabled, USIP mode takes precedence over USNIP mode.

Add the remaining IP Addresses

IP Addresses that are added after HA Synchronization is complete, will be replicated on both Primary and Secondary NetScalers. Note that VIP addresses are created later during Load Balancing and SSL Offload configuration, not at this time.



Add the remaining IP Addresses.

NetScaler ➥ Network ➥ IPs ➥ Add.

Make sure you take this opportunity to “Save” the configuration on both the Primary and Secondary NetScalers.

IP Addresses, Interfaces and VLANs

Assigning IP Addresses to Interfaces is done ‘virtually’ through the use of port based VLANs.

By default, all the interfaces on the system are in a single port-based VLAN as untagged interfaces. This VLAN is the default VLAN with a VID equal to 1.

When an interface is added to a new VLAN as an untagged member, the interface is automatically removed from the default VLAN and placed in the new VLAN. This becomes a convenient feature, such that when we plug the Netscaler into a Switch that is using VLANs with tagging, we only need to check the box, to turn on tagging. VLANs are typically used to separate subnet traffic.

If Trunking is turned On, you will see an interface as a member of more than one VLAN.

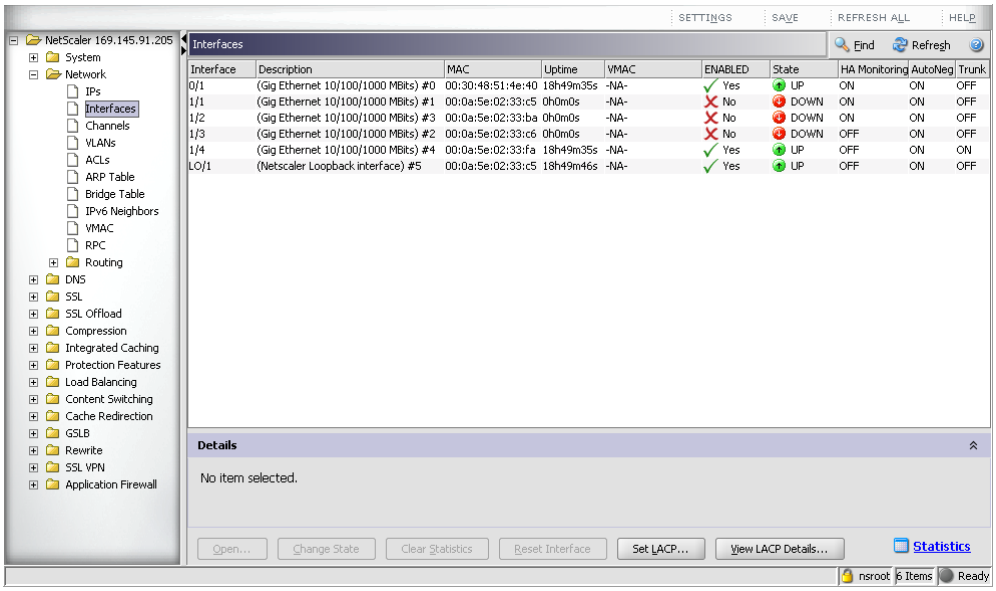
Create VLANs and Assign Subnet IP Addresses to them.

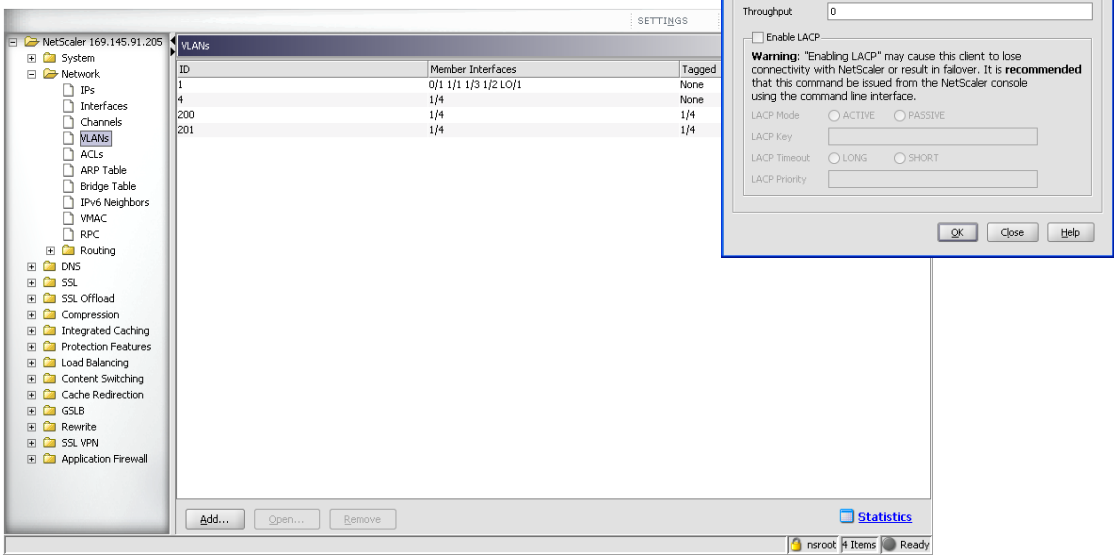
NetScaler ➥ Network ➥ VLANs ➥ Add.

Note: For this example: We create VLANs 4, 200 & 201. We use VLAN Trunking on interface 1/4.

Interface 0/1 is our management interface, in VLAN 1.

NetScaler ➥ Network ➥ VLANs, to view VLAN and Interface assignments on the Application Switch.





Configuring the Virtual MAC

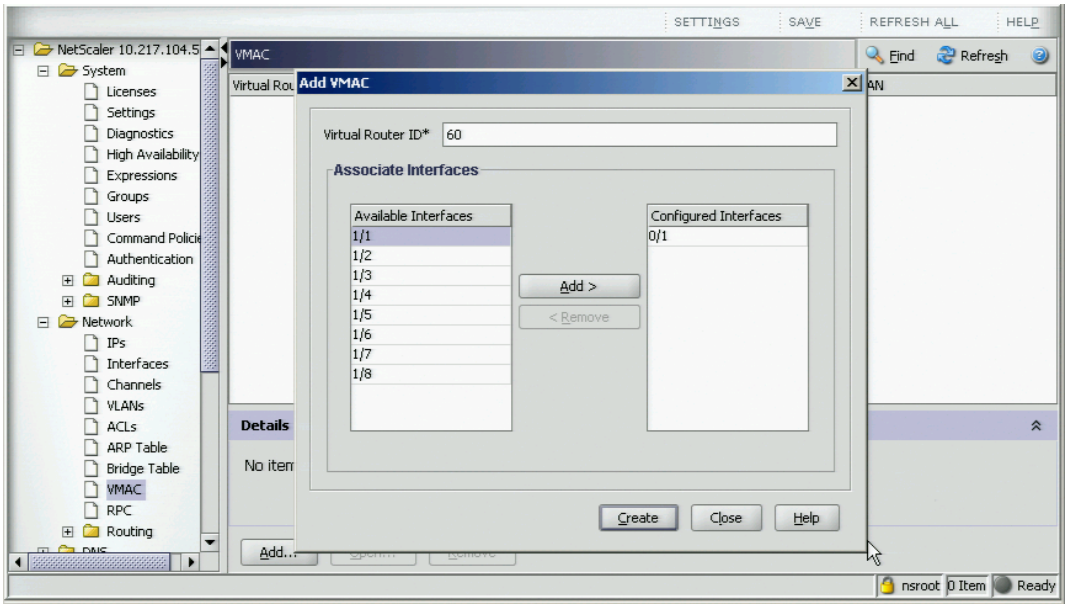
The Virtual MAC address (VMAC) is a floating entity shared by the primary and secondary nodes in an HA setup.

In an HA setup, the primary node owns all of the floating IP addresses such as MIP, SNIP, VIP, etc. It responds to ARP requests for these IP addresses with its own MAC address. As a result, the ARP table of an external device (for example, upstream router) is updated with the floating IP address and the primary node's MAC address.

When a failover occurs, the secondary node takes over as the new primary node. It then uses Gratuitous ARP to advertise the floating IP addresses that it acquired from the primary. The MAC address that the new primary advertises is that of its own interface.

Some devices do not accept Gratuitous ARP messages. You can overcome this problem by configuring a VMAC on both nodes of an HA pair. This implies that both the nodes possess identical MAC addresses. As a result, when failover occurs, the MAC address of the secondary node remains unchanged and ARP tables on the external devices do not need to be updated.

To create a VMAC, you need to create a VRID and bind it to an interface. In an HA setup, you need to bind it to the interfaces on both the primary and secondary nodes. When the VRID is bound to an interface, the system generates a VMAC with the VRID as the last octet. The generic VMAC is of the form 00:00:5e:00:01:<VRID>.



Assign a VMAC.

Navigate to NetScaler ➥ Network ➥ VMAC ➥ Add.

Add a Virtual Router ID to the Interface that HA Monitoring is enabled on.

Load Balancing Configuration

1-2-3:

Configuring Load Balancing is a simple 1-2-3 process performed by creating objects within the Citrix Application Switch. We create the objects in logical formation from the backend servers to the forward facing internet IP Address:
1) Create Servers
2) Create Services
3) Create Load Balancing VIPs w/Persistence

Create Server Objects

Create server objects that point to the backend Application and Database servers. We can refer to these servers by name as opposed to IP Address, and can then assign availability monitors to them.

Create server objects for the Application and Database servers on the backend.

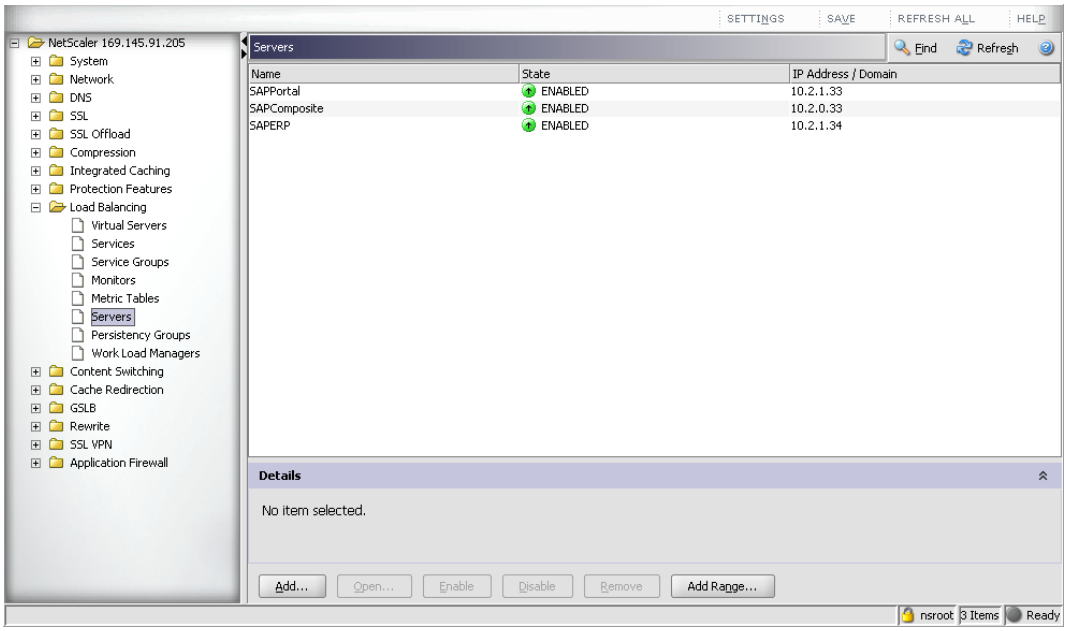
From the GUI, navigate to NetScaler ➥ Load Balancing ➥ Servers ➥ Add.

In this example, our backend servers consist The SAP Portal, Composite and ERP Servers.

SAP Portal: 10.2.1.33

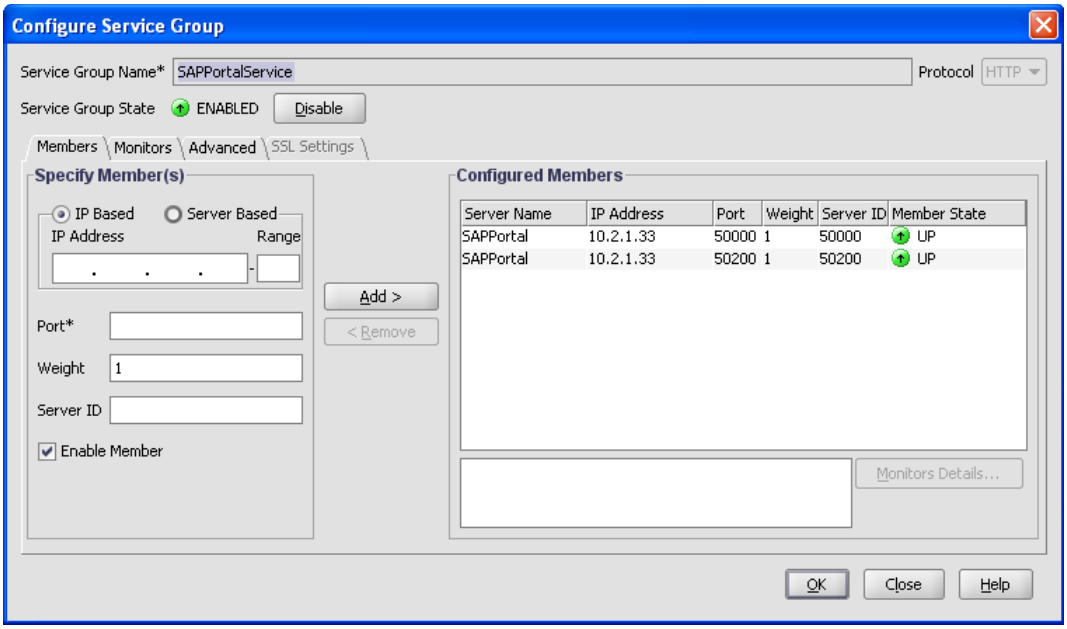
SAP Composite: 10.2.0.33

SAP ERP: 10.2.1.34



Create Service Groups

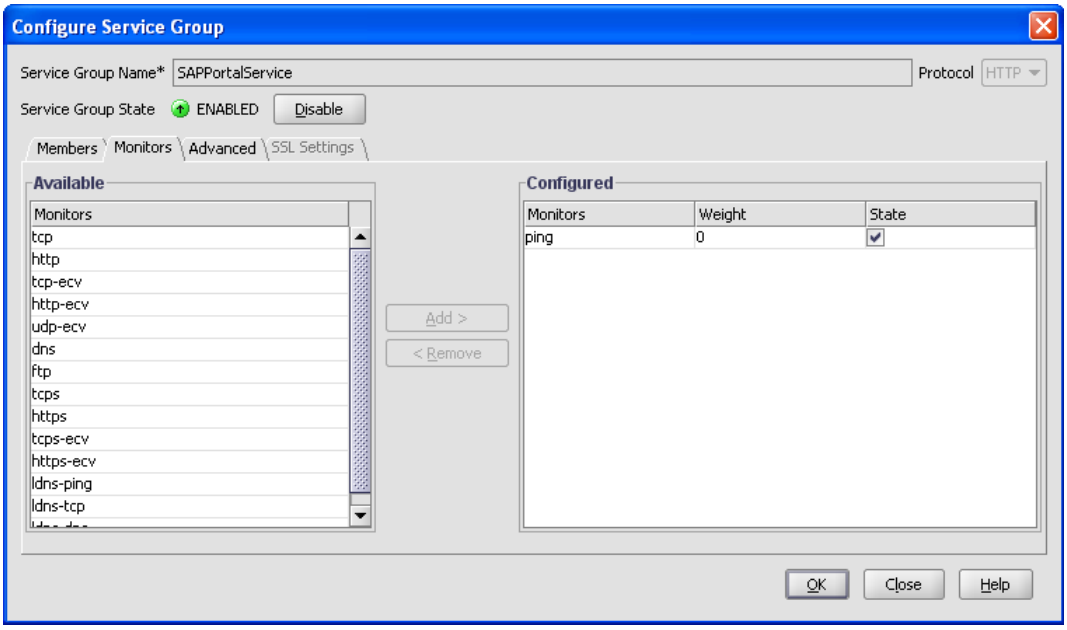
Service Groups are a containers for managing load balancing and SSL services to several instances of the same service (port number) on the same or different servers (ip address).



Add the Service Group for the SAP Portal that will distribute the load across the two SAP backend services on ports 50000 & 50200.

From the GUI, navigate to NetScaler ➥ Load Balancing ➥ Service Groups ➥ Add。

Select an availability monitor to keep in contact with the server/service. If the service goes down, load balancing will mark it down and send traffic to the other available servers/services.

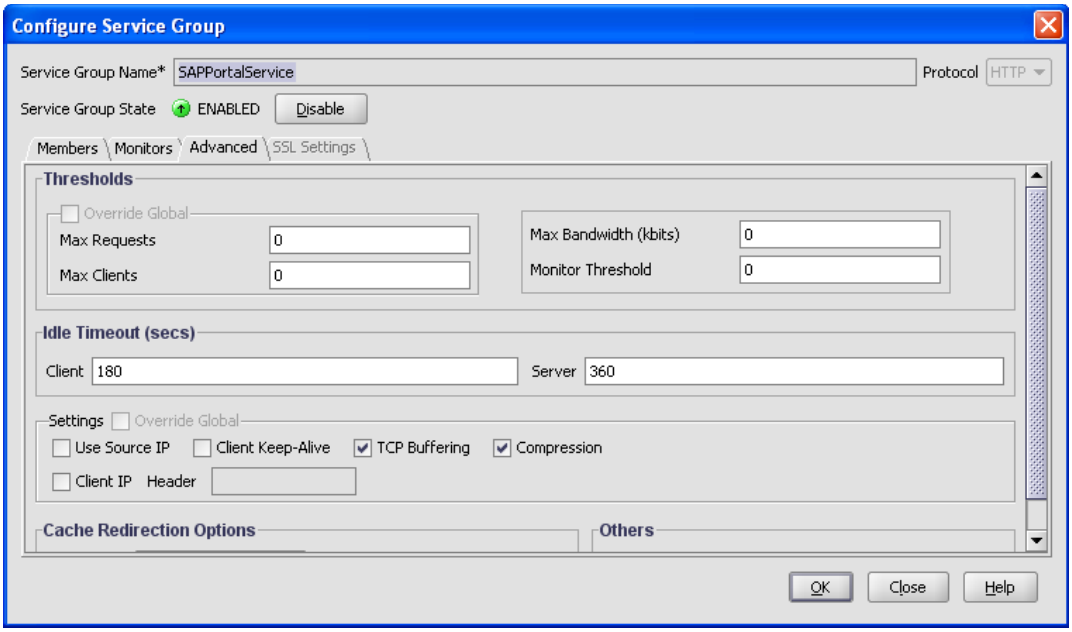


Select the ‘Monitors’ tab. Select ping or http.

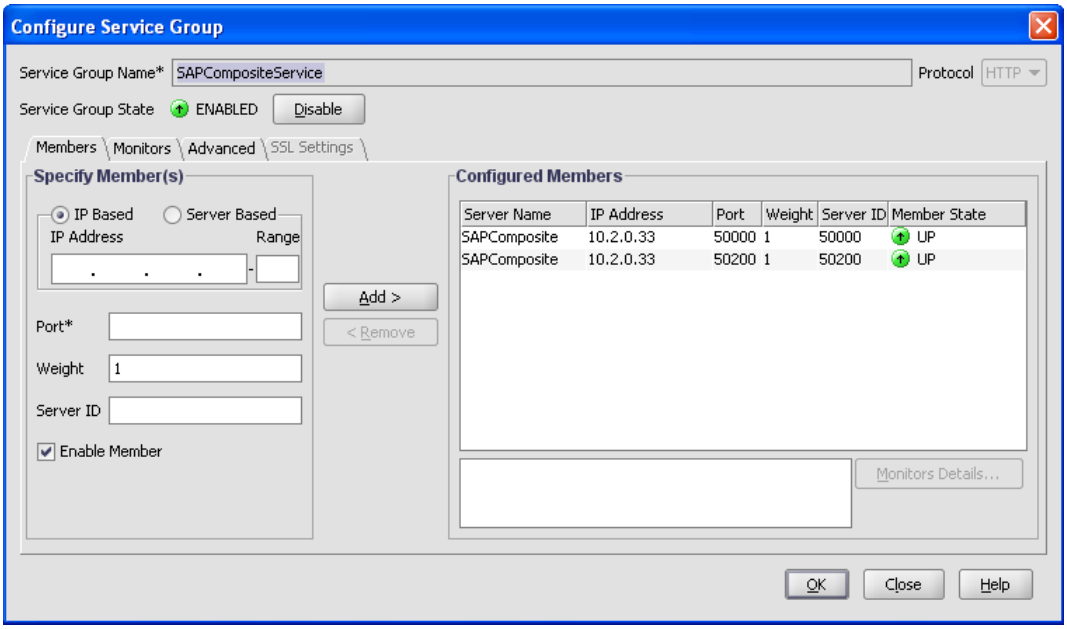
To get the most performance, select the Advanced tab and turn on Compression and TCP Buffering. The compression computation is an off-loaded task for both http and https from the SAP servers.

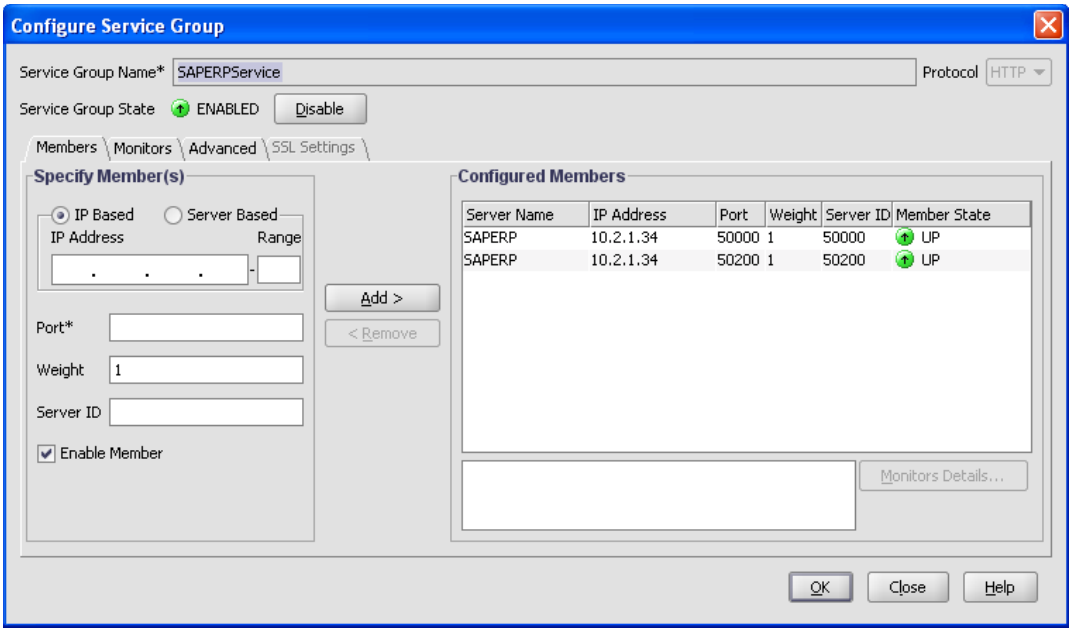
Select the Advanced tab, check TCP Buffering and Compression.

Select OK.



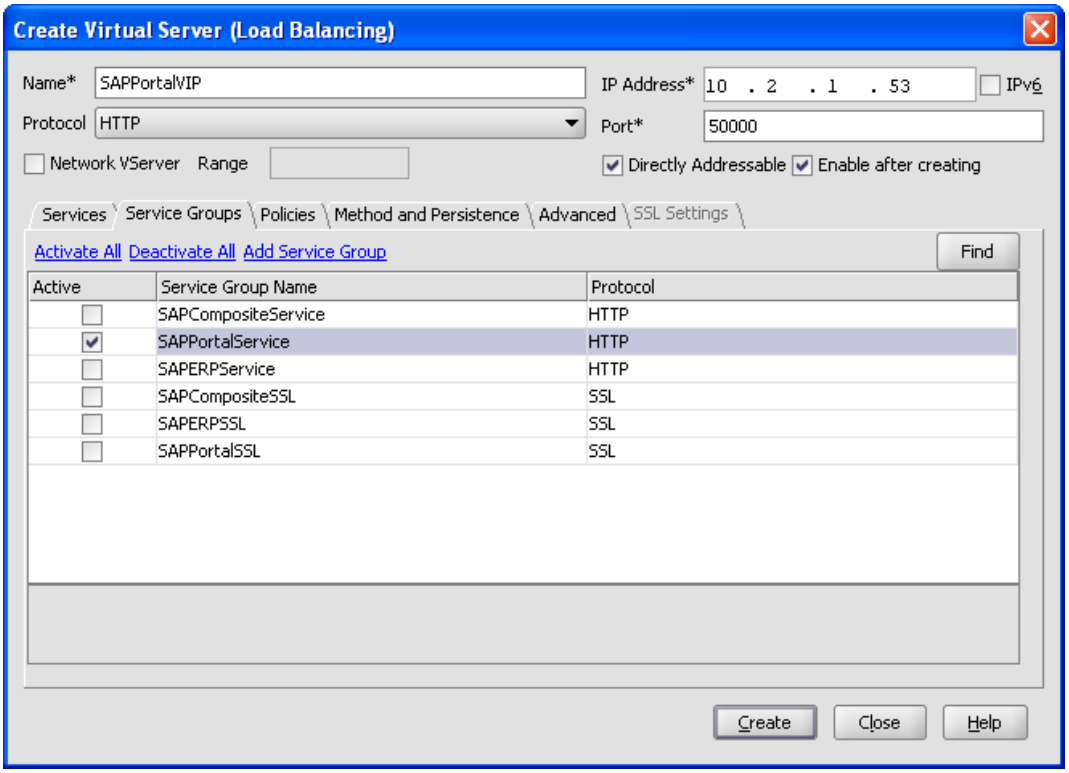
Because the SAP Composite Application Framework and ERP servers are on separate physical servers, and because we want to treat them as separate load balancing groups and services, we created separate Service Groups for those as well.





Create LB Virtual Server Objects (VIPs)

The Virtual Server or Virtual IP Address is the logical entity on the system that accepts client connections from the Internet and distributes them to the service groups/objects. The Vserver or VIP is the public facing internet connection.



Add the SAP Portal Virtual Server.

NetScaler ➥ Load Balancing ➥ Virtual Servers ➥ Add.

In this example:
Our public facing IP Address for SAP Portal is 10.2.1.53, and the default port for SAP is 50000.

Select the Service Groups tab, and check the SAPPortalService. This binds the public VIP to the services.

Load Balancing Methods & Persistence

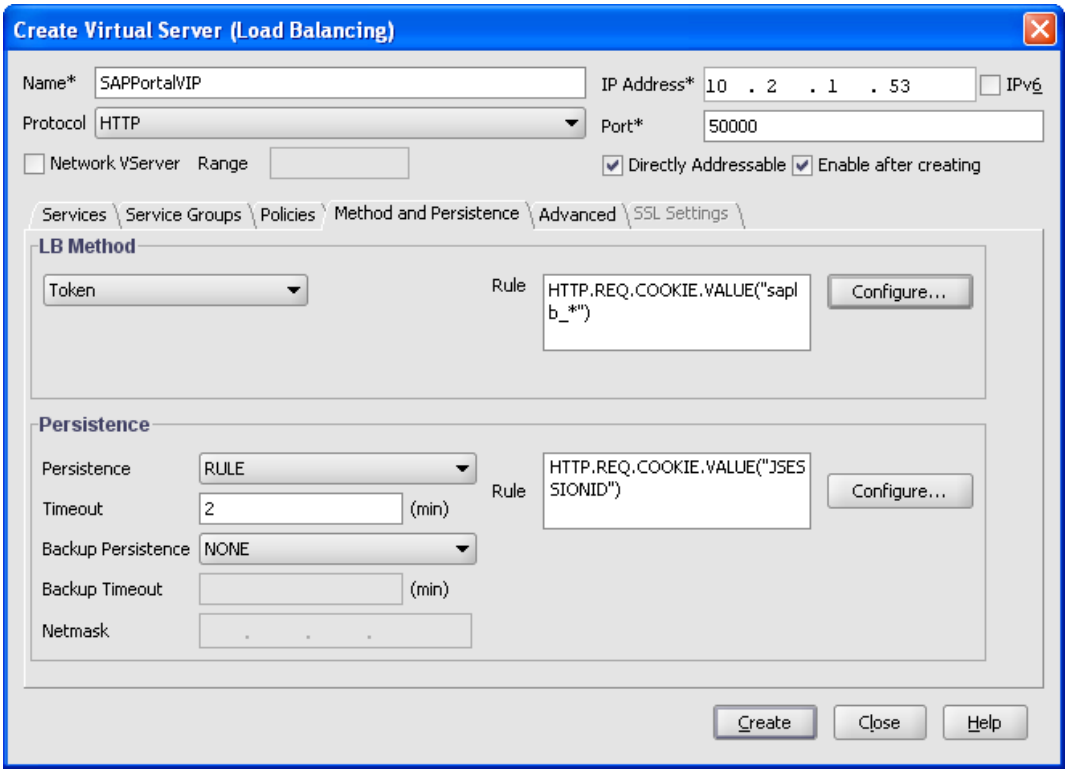
The Citrix Application Switch is capable of several Load Balancing Methods. In order to direct traffic correctly to SAP 7.0+ servers, the Citrix Application Switch must be configured to persist traffic based on the value in the SAP cookie 'saplb_*' issued from the SAP servers. This applies to the SAP Portal, Composite and ERP servers. For backward compatibility with SAP 6.49, the Citrix Application Switch should be configured to persist traffic based on the value in the SAP cookie 'JSESSIONID' issued from the SAP servers.

By default the Citrix Application Switch uses the 'Least Connections' load balancing algorithm, until a value is issued to 'saplb_*' and/or 'JSESSIONID' cookies.

Select the 'Methods and Persistence' tab. Select the LB Method ➥ Token and configure a rule to extract the value from the 'saplb_*' cookie.

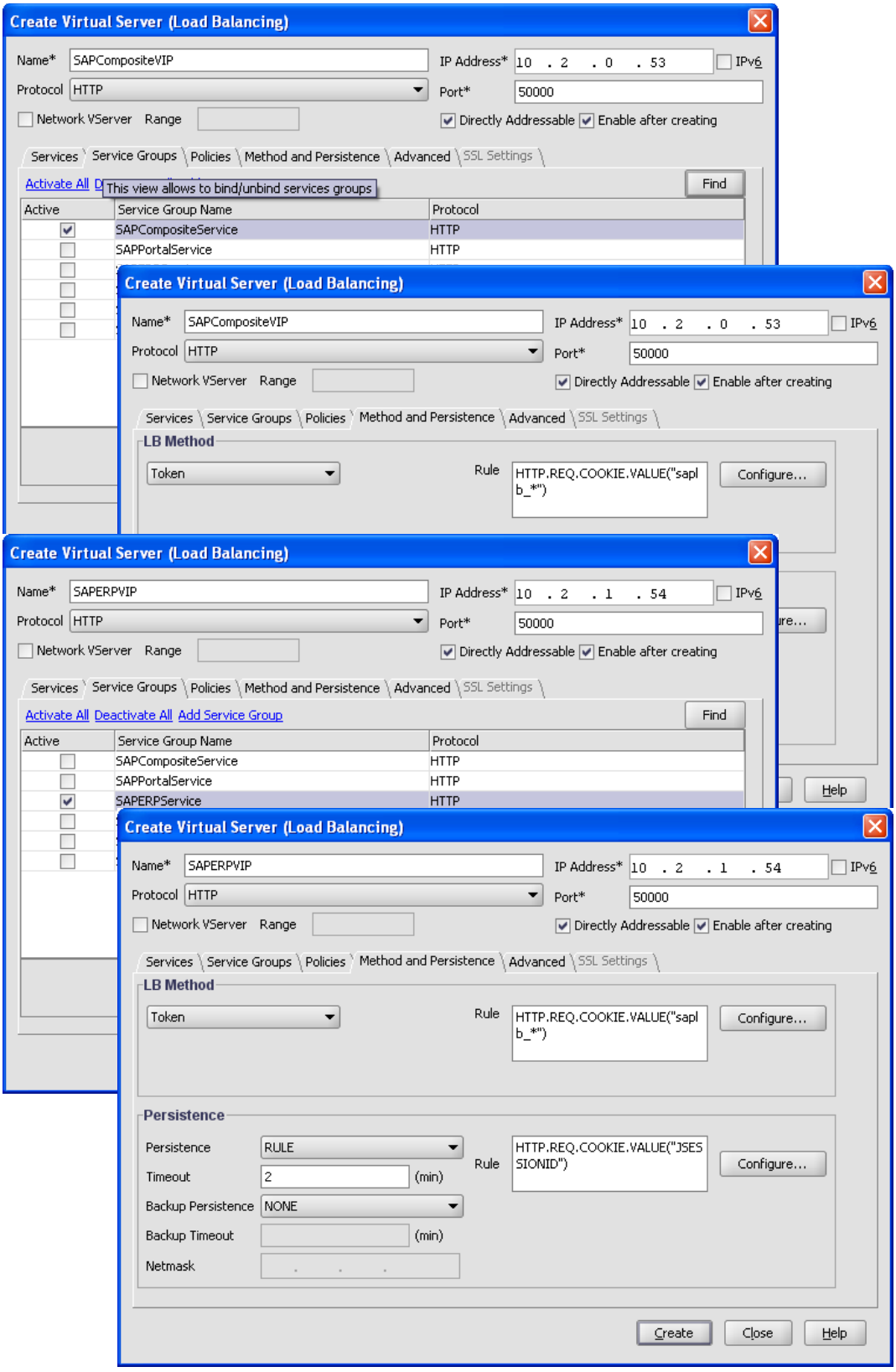
SAP 7.0+:
The rule for 'saplb_*' cookie persistence:
HTTP.REQ.COOKIE.VALUE("saplb_*")

SAP 6.49:
The rule for 'JSESSIONID' cookie persistence:
HTTP.REQ.COOKIE.VALUE("JSESSIONID")



More information regarding the SAP Load Balancing Identifier and it's contents can be found here: http://help.sap.com/saphelp_nw70/helpdata/EN/f2/d7914b8deb48f090c0343ef1d907f0/frameset.htm.

The SAP Composite Application Framework VIPs and ERP VIPs are configured the exact same way, because the communication protocol is HTTP for each of these.



Add additional VIPs for SAP Composite Application Framework and SAP ERP.

Select the appropriate service group.

Select the 'Methods and Persistence' tab, and configured the LB Method ➥ Token rule to extract the value from the 'saplb_*' cookie.

SAP 7.0+:
The rule for 'saplb_*' cookie persistence:
HTTP.REQ.COOKIE.
VALUE("saplb_*")

SAP 6.49:
The rule for 'JSESSIONID' cookie persistence:
HTTP.REQ.COOKIE.
VALUE("JSESSIONID")

Make sure you take this opportunity to "Save" the configuration on both the Primary and Secondary switches.

SSL Offload Configuration

Keys and Certificates

Using any of the SSL features on the NetScaler, such as SSL Offload, requires that you obtain a certificate and private key for the NetScaler. An SSL certificate is a digital data form (X509) that identifies a particular company (domain) or an individual. An SSL key is the private component of the public-private key pair used in asymmetric key encryption (public key encryption). PKCS#12 to PEM conversion

convert pkcs12 /nsconfig/ssl/client_certkey.pem –import –pkcs12File /

nsconfig/ssl//nsconfig/ssl/client_certcertkey.p12

PKCS#12 to PEM conversion, with encrypted key

convert pkcs12 /nsconfig/ssl/client_certkey.pem –import –pkcs12File /

nsconfig/ssl/client_certcertkey.p12 -des

Note: The Application Switch supports a certificate key size of up to 2,048 bits (RSA/DSA).

There are three ways to obtain keys and certificates for use with the Application Switch.

1. Create a self-signed certificate using the SSL certificate wizard.
2. Use an existing one, either root or intermediary, from an existing web server.
3. Obtain one from a public CA-Certificate Authority, such as Verisign.

In this guide we need three certificates.

1. A self-signed server certificate for our front-end HTTPS connections.
2. A self-signed Certificate Authority certificate for our front-end HTTPS connections, to complete the certificate chain.
3. An imported PFXS certificate, from our backend SAP Portal server.

Using the SSL Certificate Wizard

To launch the SSL Certificate Wizard, from the GUI, navigate to NetScaler ➥ SSL.

Click on the <Certificate Wizard>.

Use this wizard to create the self-signed 1) “server certificate” and 2) “CA certificate” for the front-end HTTPS connections.

Importing the SAP Portal server certificate

Providing HTTPS communication to the front-end of the load-balancing connections is the first step in creating secure communications for the organization. Securing the communications on the back-end from the Citrix Application Switch to the SAP servers is the second step. This part, however, is optional. In other words, from the Citrix VIP to the back-end SAP servers, the communications can use either cleartext HTTP or secure HTTPS. SAP asked Citrix to offer the customer the option to use HTTPS on all of it's SAP product's interfaces, in addition to HTTP, as a proof point. Therefore, we configure HTTPS on the backend, as HTTP is the same without the use of certificates.

For more information on setting SSL on SAP: http://help.sap.com/saphelp_nw04s/helpdata/en/9a/53a2a4a45e244aa189c2b7065a0b78/content.htm.

In this guide we chose to use HTTPS as a proof point, that HTTPS communication can be performed between the Citrix Application Switch and the back-end SAP servers using non-standard ports, e.g. 50001, 50201.

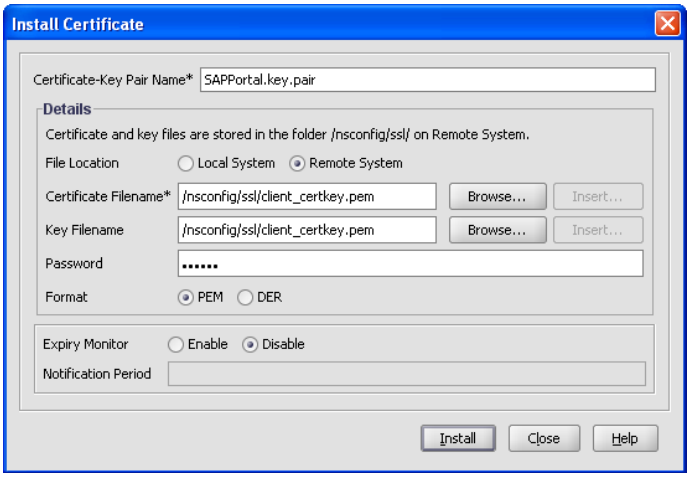
1. For secure communications using HTTPS on the back-end, both the certificate and key from the SAP Portal needs to be EXPORTED into PKCS#12 format.
2. This file needs to be uploaded to the Citrix Application Switch using a tool such as WINScp, http://winscp.net. The file should then be placed in the /nsconfig/ssl directory on the Citrix Application Switch.
3. Connect to the Citrix Application Switch using a SSH terminal emulation program, such as putty, http://putty.org.
4. Type the 'shell' command at the command prompt, and you will be able to enter the import commands:
   - » PKCS#12 to PEM conversion:

     #convert pkcs12 /nsconfig/ssl/client_certkey.pem –import –pkcs12File /nsconfig/ssl//nsconfig/ssl/client_certcertkey.p12
   - » PKCS#12 to PEM conversion, with encrypted key:

     #convert pkcs12 /nsconfig/ssl/client_certkey.pem –import –pkcs12File /nsconfig/ssl/client_certcertkey.p12 -des

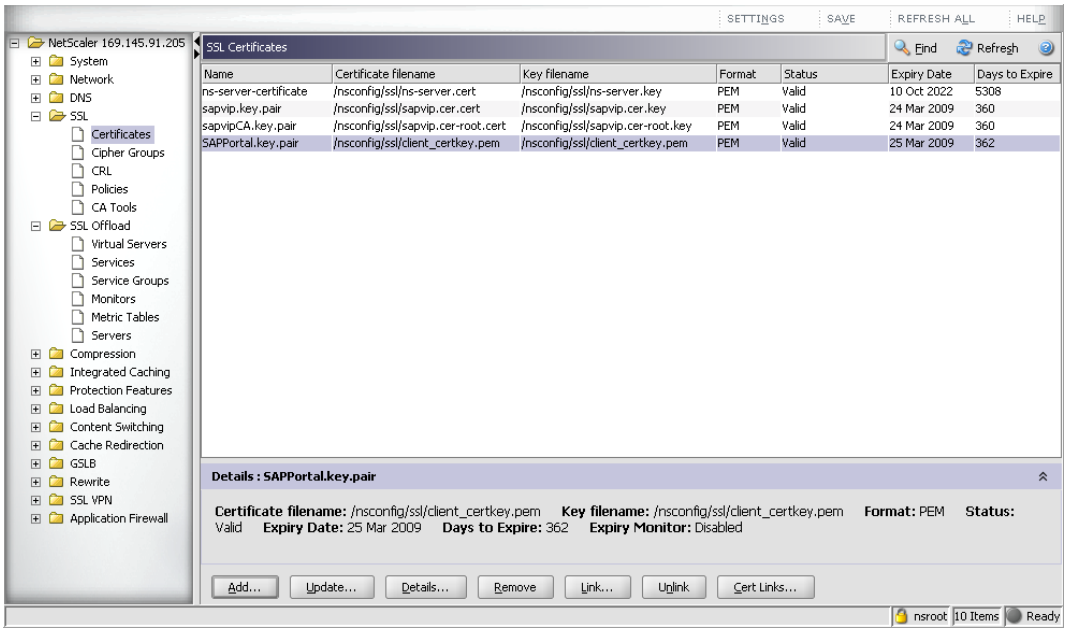
     Note: The -des option will encrypt the output key using the DES algorithm. The user will be prompted to enter the pass-phrase used for encryption.
5. Launch the GUI, and create certificate.

From the GUI, navigate to NetScaler ➥ SSL ➥ Certificates ➥ Add.



After creating the 1) Server certificate 2) CA certificate and importing the 3) SAP certificate, all three certificates should be listed in the certificates folder.

The certificates can also be found in the /nsconfig/ssl directory if logging into the Citrix through an SSH session.



TIP:

Common Name:
The common name in the certificate should match the name used by DNS servers during a DNS lookup of your virtual server (for example, www.sapportal.com). Most browsers use this information for authenticating the virtual server's certificate during the SSL handshake. If the virtual server DNS name does not match the common name as given in the server certificate, browsers will terminate the SSL handshake or prompt the user with a warning message. Do not use wildcard characters such as * or ? and do not use an IP address as a common name. The common name should be without the protocol specifier http:// or https://.

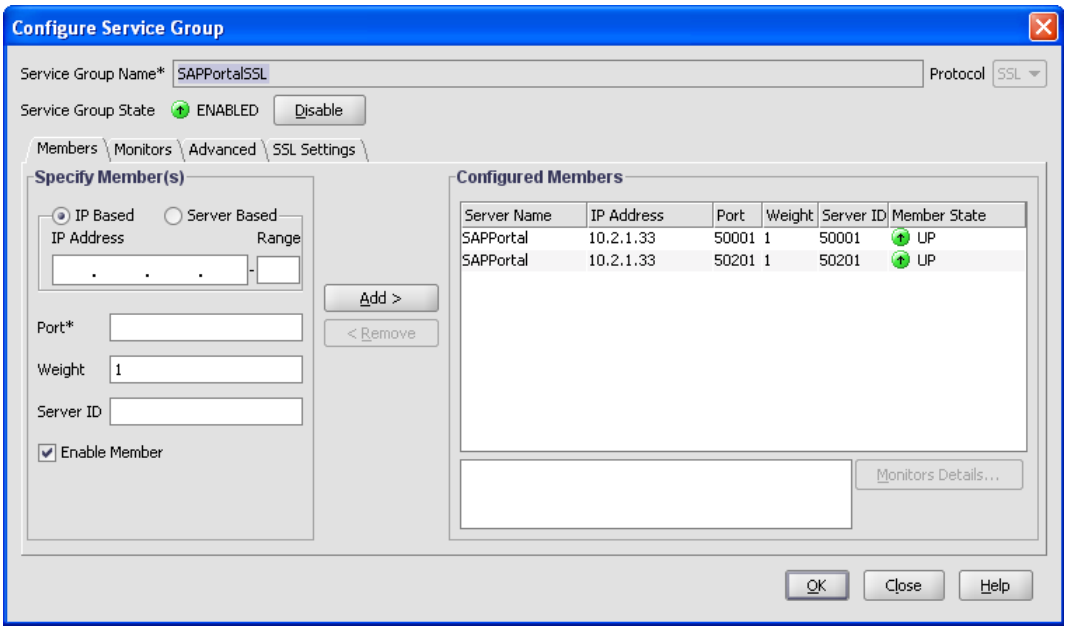
Organization Name:
The organization name (corporation, limited partnership, university, or government agency) must be registered with some authority at the national, state, or city level. Use the legal name under which the organization is registered. Do not abbreviate the organization name and do not use the following characters in the name: < > ~ ! @ # $ % ^ * / \ ( )?. For example, Citrix Systems, Inc.

Create Server Objects

This has already been done when the Load Balancing VIP was created. We use the same Server definitions. If we wanted to point the SSL traffic to a different backend server, we would add a new one here.

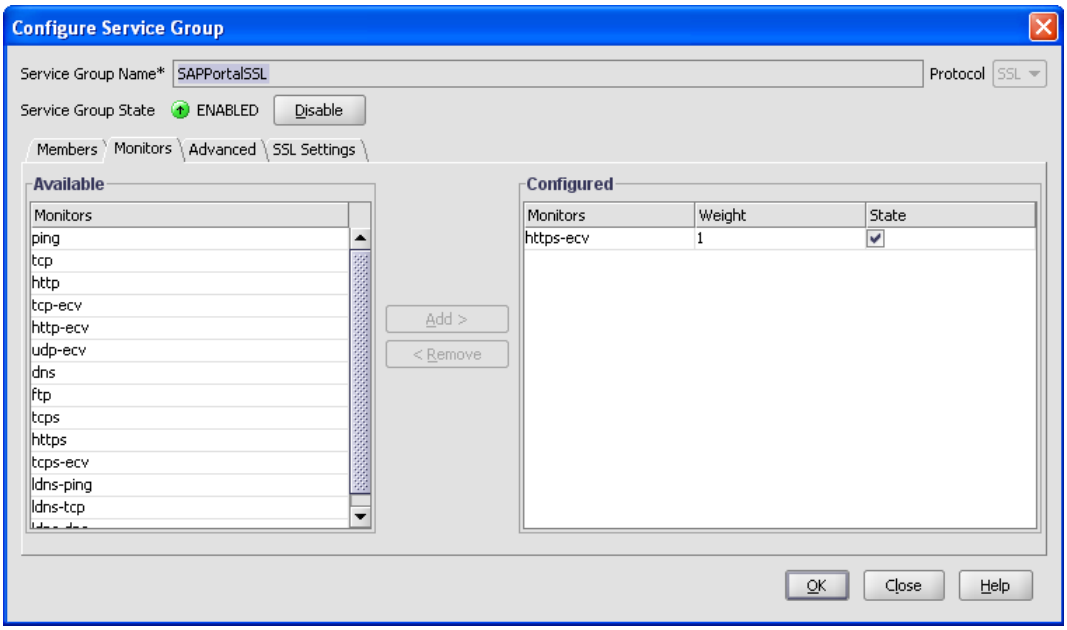
Create Service Groups

Similar to the procedure we implemented for Load Balancing, we create Service Groups for the same servers; however, this time we specify the protocol SSL and port numbers 50001, 50201 - the ports that SAP uses for HTTPS communication.



From the GUI, navigate to NetScaler ➥ Load Balancing ➥ Service Groups ➥ Add.

Add the SAP Portal servers using SSL at the protocol, on ports 50001 & 50201.



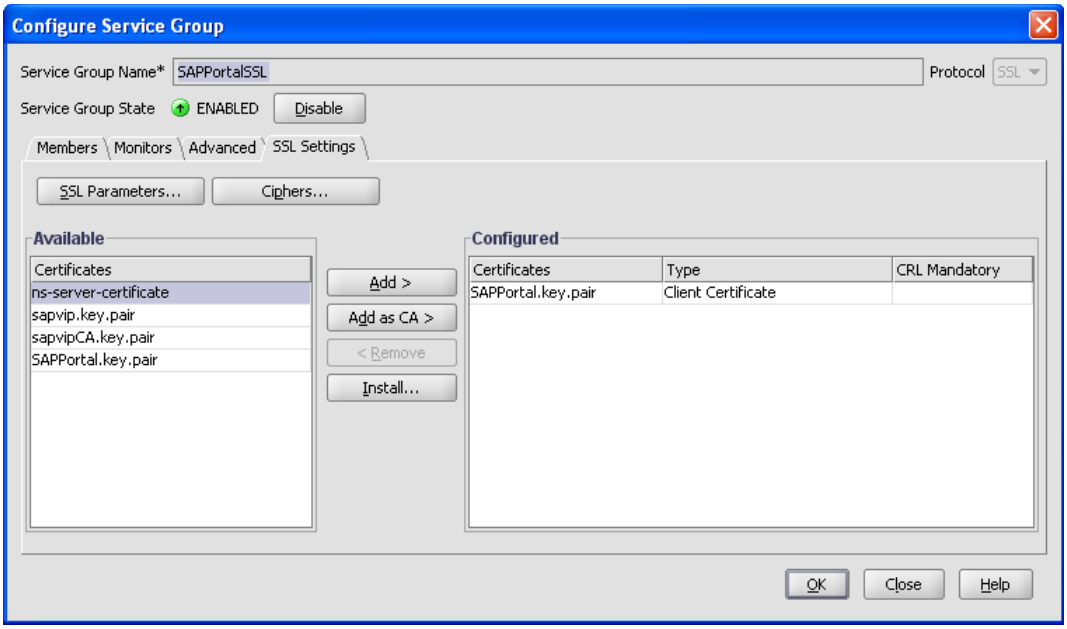
Using the "https-ecv" monitor is a good way to make sure that the SSL handshake is working between the Citrix and the backend SAP Portal connection.

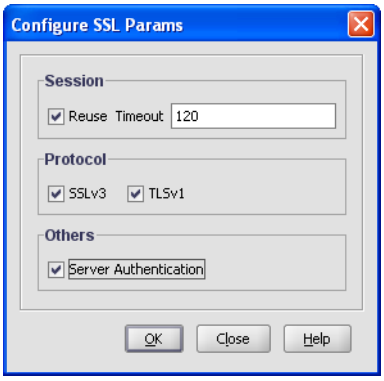
In order to authenticate the SAP Portal server certificate, you need to:

a) Bind the SAP Portal certificate to this server group.

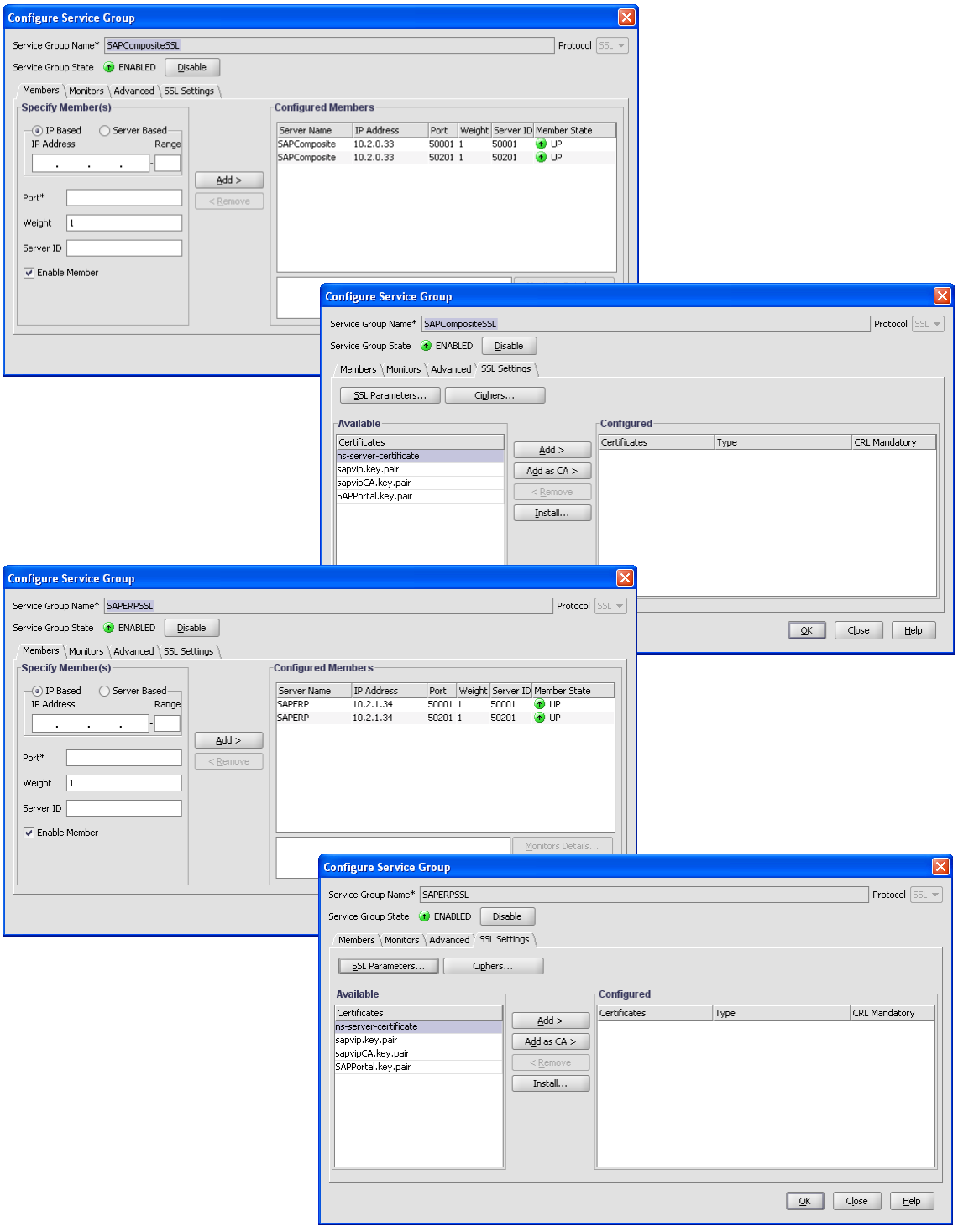
b) Turn on Server Authentication, located under SSL Parameters.

Be sure to turn on Session Reuse to get the performance gain out of TCP Multiplexing and SSL Offload.





Create Server Groups for the SAP Composite and ERP Servers, using the SSL Protocol, on ports 50001 & 50201. Use the https-ecv monitor.



For the SAP Composite and ERP Servers, we did not configure server certificate authentication.

By default the Citrix system will not authenticate the backend web server's certificate, but will still use SSL as the communication method.

This configuration assumes that the server's certificate is trusted by Citrix Application Switch.

Create SSL Virtual Server Objects (VIPs)

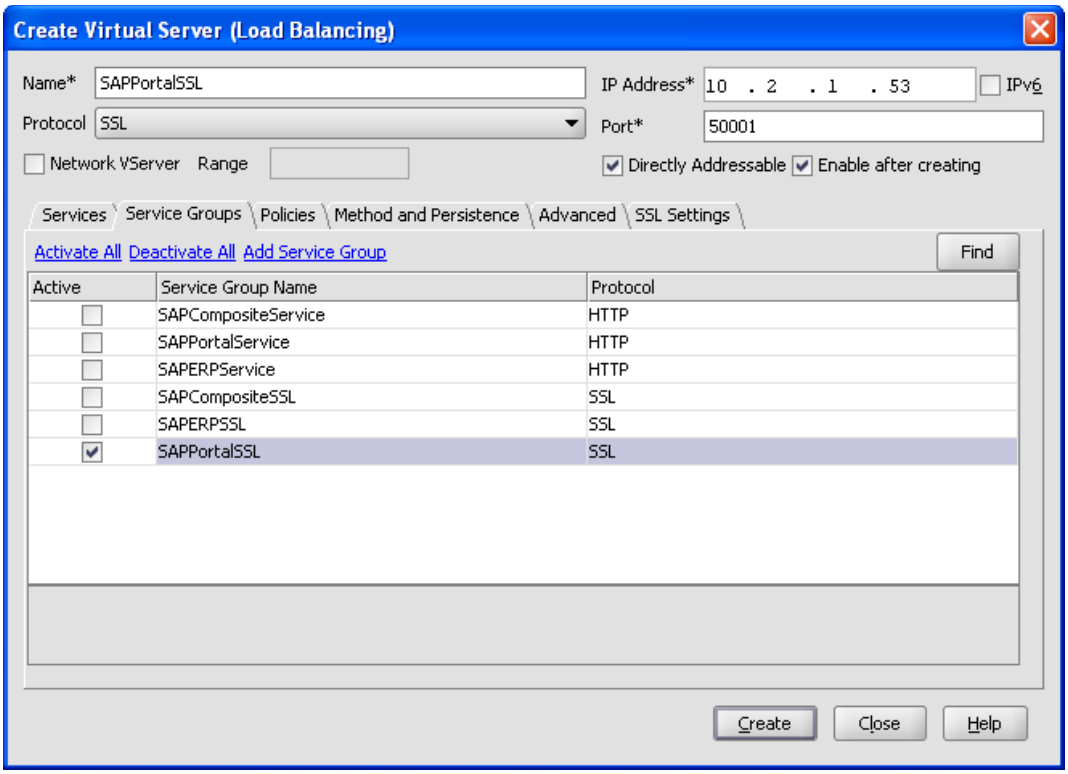
Similar to the procedure we implemented for Load Balancing, we create Virtual Server Objects or Virtual IP Addresses (VIPs) for our front-end SSL connection. The only difference is here we specify the SSL protocol and the port numbers 50001 - the port that SAP uses for HTTPS communication.

From the GUI, select NetScaler ➥ Load Balancing ➥ Add.

Specify the SSL protocol and the port 50001.

Our front-end IP Address for the SAP Portal is the same, 10.2.1.53, as the Load Balancing VIP, and we can do this because the port number is different.

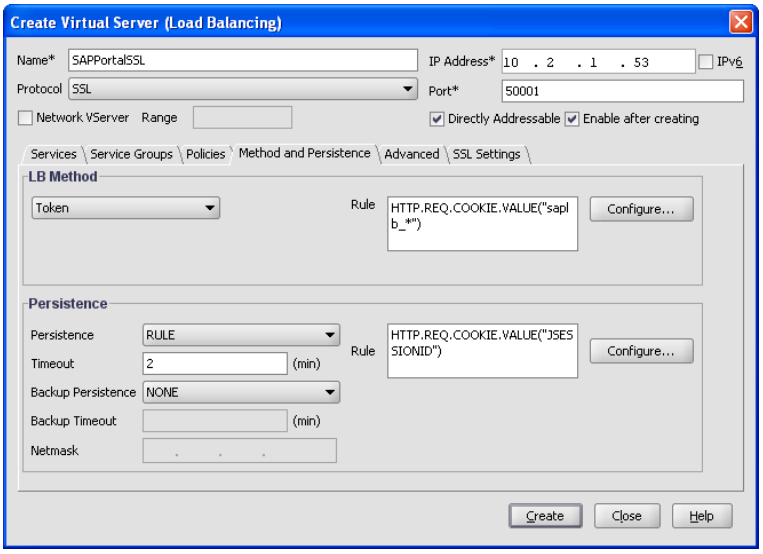
Select the SAPPortalSSL group we created in the previous step.



SSL Load Balancing Methods & Persistence

The same load balancing methods and persistence for HTTP connections are available for HTTPS connections. In order to direct HTTPS traffic correctly to SAP 7.0+ servers, the Citrix Application Switch must be configured to persist traffic based on the value in the SAP cookie 'saplb_*' issued from the SAP servers. This applies to the SAP Portal, Composite and ERP servers. For backward compatibility with SAP 6.49, the Citrix Application Switch should be configured to persist traffic based on the value in the SAP cookie 'JSESSIONID' issued from the SAP servers.

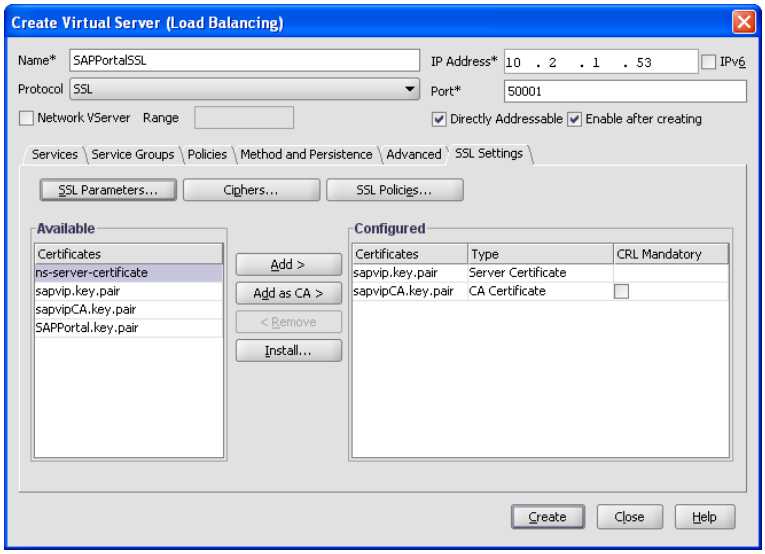
By default the Citrix Application Switch uses the 'Least Connections' load balancing algorithm, until a value is issued to 'saplb_*' and/or 'JSESSIONID' cookies.



Select the 'Methods and Persistence' tab. Select the LB Method ➥ Token and configure a rule to extract the value from the 'saplb_*' cookie.

SAP 7.0+:
The rule for 'saplb_*' cookie persistence:
HTTP.REQ.COOKIE.
VALUE("saplb_*")

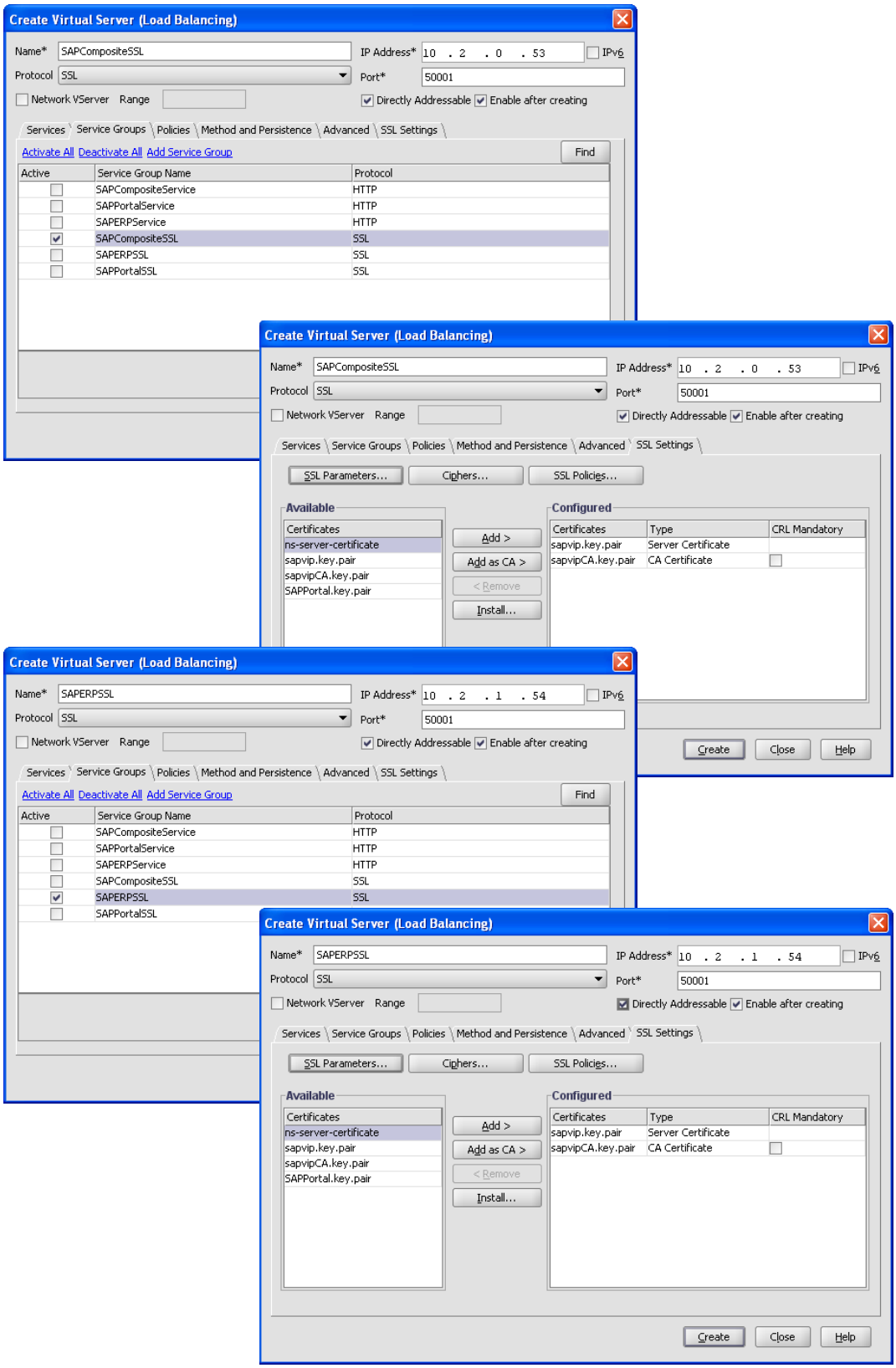
SAP 6.49:
The rule for 'JSESSIONID' cookie persistence:
HTTP.REQ.COOKIE.
VALUE("JSESSIONID")



Configure the Server CertificateandCACertificate for the front-end HTTPS connections.

The Citrix Application Switch will now serve as the SSL termination point for all incoming client traffic.

Repeat the steps and create SSL VIPs for the Composite and ERP servers.



Caching

Caching for SAP Applications

The Integrated Caching feature of the Application Switch helps optimize the delivery of web content and applications. SAP has architected their applications to make use of browser caches for static content, and thus sends directives to the browser cache through the Cache-Control header in responses to the client.

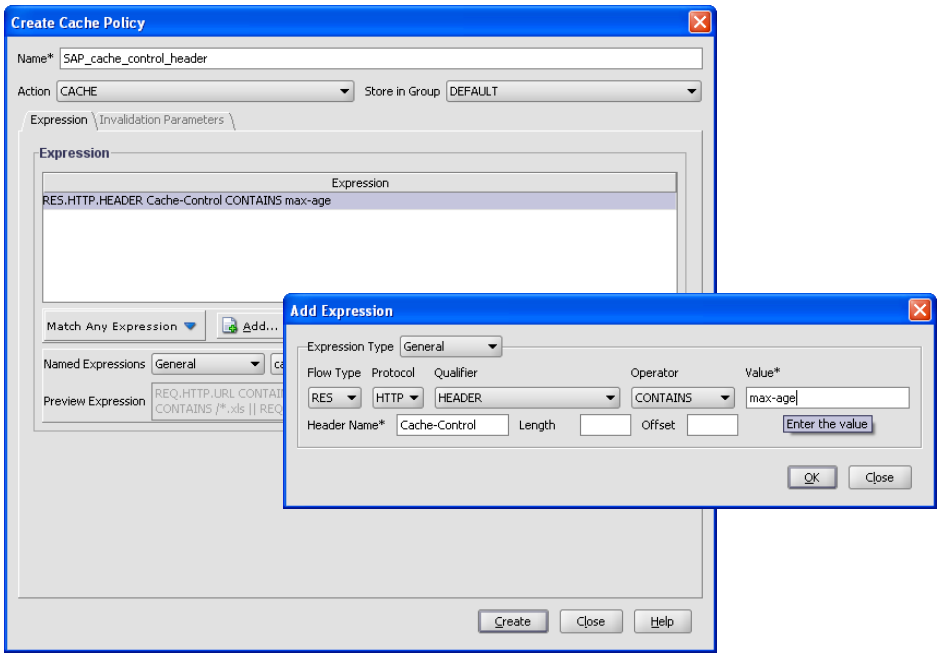
If static content is considered cache-able by SAP, an HTTP Header "Cache-Control: max-age=86400" will be sent along with the object. This is the equivalent to 24 hours, and is configurable by the customer. This is typical for many of the files of type .gif, .jpg, .js and .css. The SAP servers also respond to clients browsers with 304-not modified headers if the content hasn't changed, similar to the function that the Citrix Application Switch would provide.

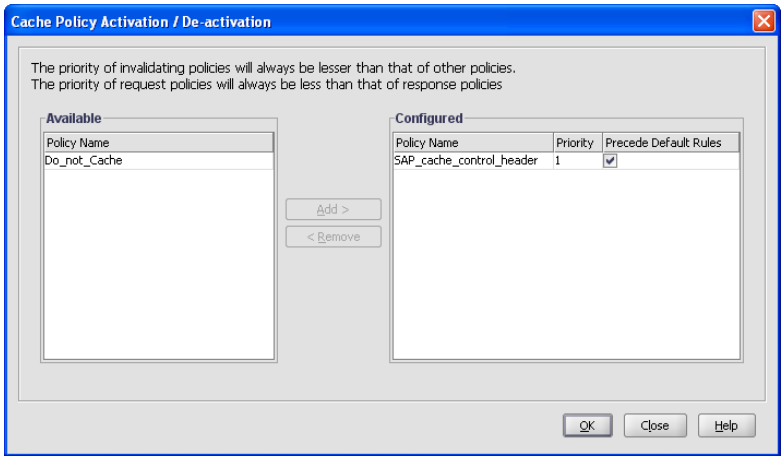
SAP recommends that dynamic caching not be implemented for their applications.

In order to cache static content for SAP Applications, the default Citrix caching policies will catch the header "Cache-Control" with the string "max-age", as well as others that can be found in the Appendix of the Installation and Configuration Guide V2. In Citrix v8.0 the default caching policies are not visible and the hit counters cannot be viewed. If you want to see the hit counter, you can manually add the rule to look for the "Cache-Control" header with the string "max-age". Any other caching policies are at the discretion of the end-user.

From the GUI, select NetScaler ➥ Integrated Caching ➥ Polices ➥ Add.

Add a policy to look in the Responses for the Cache-Control Header "max-age".





To activate the policy, from the GUI, select NetScaler ➥ Integrated Caching ➥ Polices ➥ Activate Policies.

Move the new policy to the right, to override the default rule.

Compression

Compression for SAP Applications

The Citrix Application Switch compresses HTTP responses for browsers that are compression-aware. In other words, for browsers that send the header "Accept-Encoding: gzip,deflate". The Citrix switch automatically compresses text and html. It does not compress images. With the system's integrated caching feature enabled, compressed content can be cached and served to compression-aware clients without re-compression.

The Citrix Application Switch uses Intelligent response filtering: Responses with a content-length of zero are not compressed. If the response is already compressed, it is detected and bypassed by the compression engine. This enables origin sites to use server-based compression in conjunction with Citrix's compression feature.

HTTPS is supported when encryption is performed by the Citrix system. The server's responses are compressed and encrypted before they are sent to the HTTPS client.

Compression works by defining service/service groups and compression polices. Services/service groups are entities that are logical representations of applications on the physical servers. The compression policies enable the system to identify the content that needs to be compressed. A compression policy consists of an expression and an action. An expression is created to identify the files entering the system, for example, HTML files, text files, js files, or css files. An action defines the action the system performs on the file identified by the expression. For example, you can configure a compression policy comprised of an expression that identifies javascript files and an action that compresses the javascript files.

You can enable compression to be applied globally to all traffic or to individual virtual servers (VIPs) where the compression policies are bound to a load balancing vserver. This allows the compression policies to be evaluated for only the services bound to that vserver.

Note:

With the system's integrated caching feature enabled, compressed content can be cached and served to compression-aware clients without re-compression.

When the "**Compress**" action is set, the system uses the GZIP algorithm to compress data for browsers that support either GZIP or both GZIP and DEFLATE. Similarly, the system uses the DEFLATE algorithm to compress data for browsers that support the DEFLATE algorithm. If the browser does not support either algorithm, and the action has been set to COMPRESS, the system does not compress data.

SAP Application non-compressible content types

Flow	Content type	Expression	Action
response	application/zip	RES.HTTP.HEADER Content-Type CONTAINS application/zip	NOCOMPRESS
response	application/pdf	RES.HTTP.HEADER Content-Type CONTAINS application/pdf[1]	NOCOMPRESS
response	content/unknown	RES.HTTP.HEADER Content-Type CONTAINS content/unknown	NOCOMPRESS
response	[unknown]	RES.HTTP.HEADER Content-Type CONTAINS [unknown]	NOCOMPRESS
request	/*.zip	REQ.HTTP.URL CONTAINS /*.zip	NOCOMPRESS
request	/*.cs	REQ.HTTP.URL CONTAINS /*.cs	NOCOMPRESS
request	/*.rar	REQ.HTTP.URL CONTAINS /*.rar	NOCOMPRESS
request	/*.arj	REQ.HTTP.URL CONTAINS /*.arj	NOCOMPRESS
request	/*.z	REQ.HTTP.URL CONTAINS /*.z	NOCOMPRESS
request	/*.gz	REQ.HTTP.URL CONTAINS /*.gz	NOCOMPRESS
request	/*.tar	REQ.HTTP.URL CONTAINS /*.tar	NOCOMPRESS
request	/*.lzh	REQ.HTTP.URL CONTAINS /*.lzh	NOCOMPRESS
request	/*.cab	REQ.HTTP.URL CONTAINS /*.cab	NOCOMPRESS
request	/*.hqx	REQ.HTTP.URL CONTAINS /*.hqx	NOCOMPRESS
request	/*.ace	REQ.HTTP.URL CONTAINS /*.ace	NOCOMPRESS
request	/*.ear	REQ.HTTP.URL CONTAINS /*.ear	NOCOMPRESS
request	/*.compressed	REQ.HTTP.URL CONTAINS /*.compressed	NOCOMPRESS

SAP Application compressible content types

Flow	Content type	Expression	Action
response	text/*	RES.HTTP.HEADER Content-Type CONTAINS text[2]	COMPRESS
response	application/*	RES.HTTP.HEADER Content-Type CONTAINS application	COMPRESS
request	*.html	REQ.HTTP.URL CONTAINS /*.html	COMPRESS
request	*.htm	REQ.HTTP.URL CONTAINS /*.htm	COMPRESS
request	*.txt	REQ.HTTP.URL CONTAINS /*.txt	COMPRESS

Citrix automatically compressed content types

Flow	Content type	Expression	Action
response	text/html	RES.HTTP.HEADER Content-Type CONTAINS text	COMPRESS
response	text/plain	RES.HTTP.HEADER Content-Type CONTAINS text	COMPRESS
response	text/xml	RES.HTTP.HEADER Content-Type CONTAINS text	COMPRESS
response	text/css	RES.HTTP.HEADER Content-Type CONTAINS text/css	COMPRESS
response	text/rtf	RES.HTTP.HEADER Content-Type CONTAINS text	COMPRESS
response	application/msword	RES.HTTP.HEADER Content-Type CONTAINS application/msword	COMPRESS
response	application/vnd.ms-excel	RES.HTTP.HEADER Content-Type CONTAINS application/vnd.ms-excel	COMPRESS
response	application/vnd.ms-powerpoint	RES.HTTP.HEADER Content-Type CONTAINS application/vnd.ms-powerpoint	COMPRESS

1. SAP Recommends not compressing pdf files, although it is technically possible.
2. Optional as Text will automatically be compressed by Citrix default compression policies.

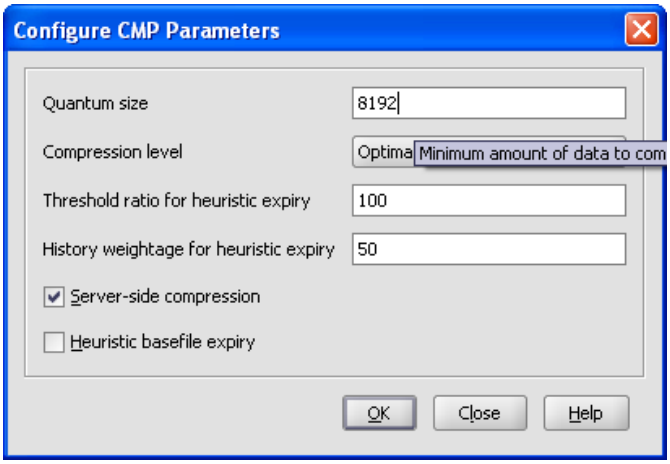
Configuring Compression for SAP Application

For SAP Applications, the Citrix Application Switch uses some default compression policies, along with some custom policies that you must configure. Additionally, in order to offload the compression calculation from the SAP servers, you need to remove the Accept-Encoding header's from the client requests, this way the Citrix Application Switch will end up doing all the compression work, and the SAP servers will not have to be burdened with that. If for some reason, SAP sends a compressed response, the Citrix Applicaiton Switch will not try to re-compress it, and will pass it through.

The default minimum object size for compression in 56k, however SAP recommends setting this to 8k.

From the GUI, select NetScaler ➥ Compression ➥ Configure CMP Parameters.

Change the default from 57344 to 8192.

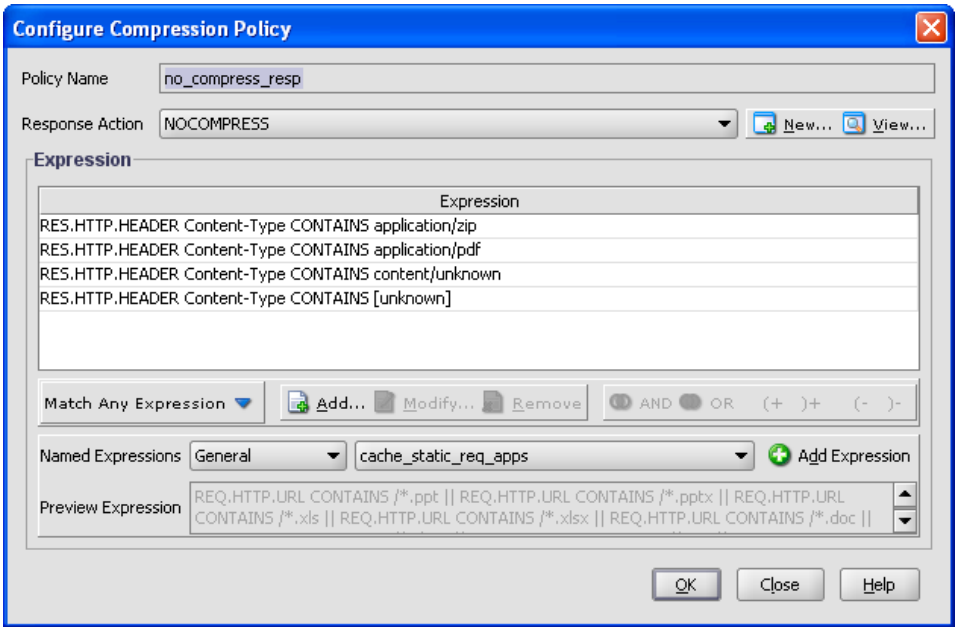


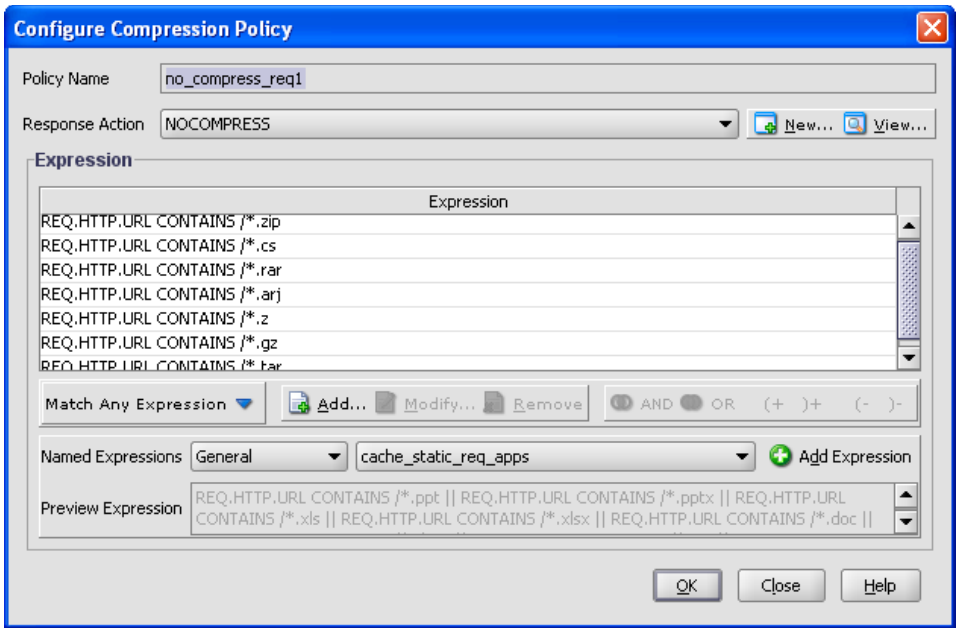
It is important to understand what SAP considers to be non-compressible content vs. compressible content, as referenced in the preceding tables. We configure the expressions for non-compressible content first, and if they match we don't compress them. If any of the later compressible content type expressions match, we then compress that content. We take advantage of some of the built-in policies as well.

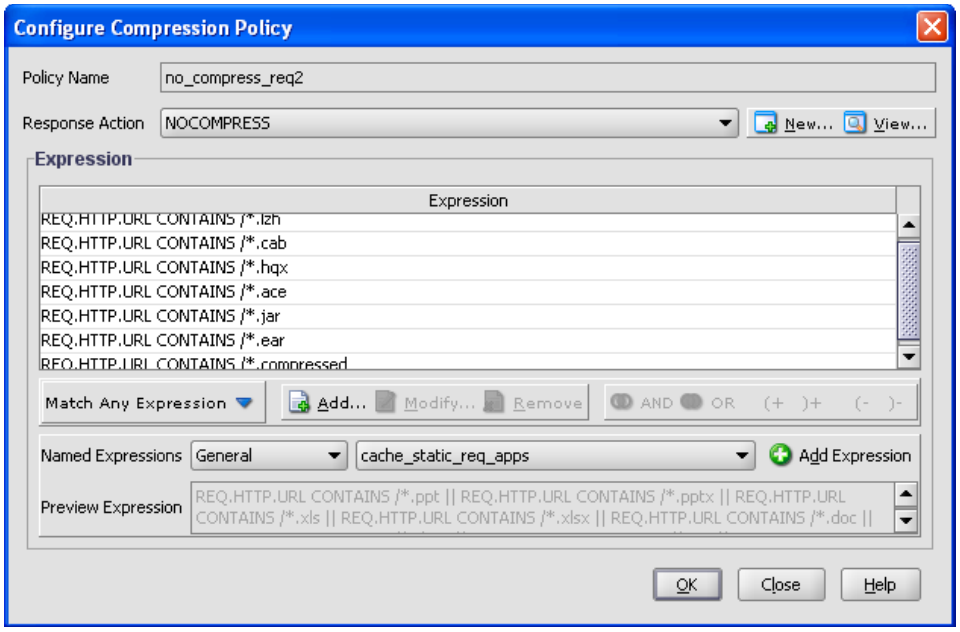
From the GUI, select NetScaler ➥ Compression ➥ HTTP ➥ Add.

Notice this set of compression policies is for the response flows.

Set the compression response action to NOCOMPRESS. Several expressions can be combined in an "OR" matching algorithm.







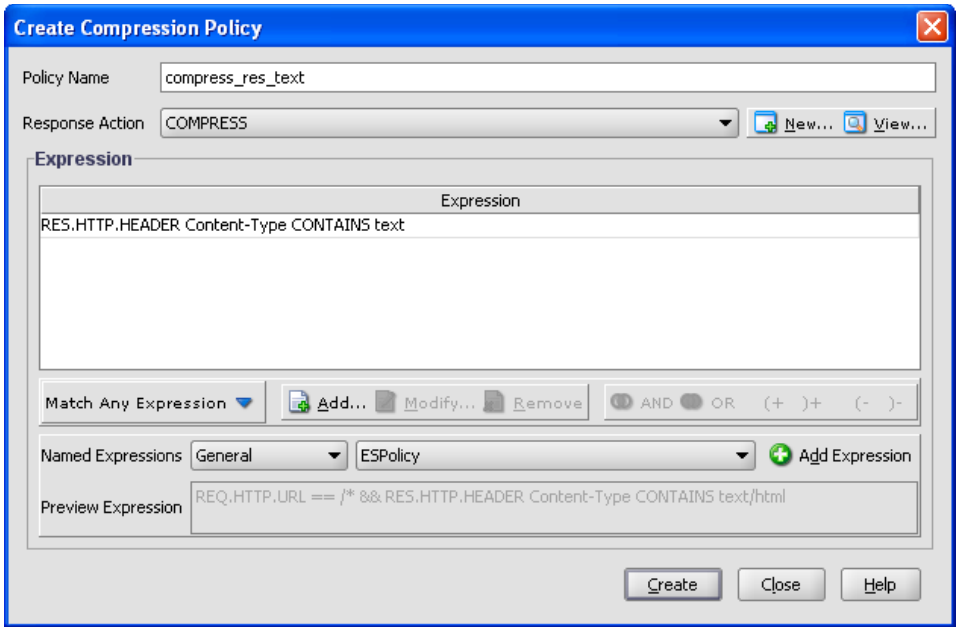
Add the additional policies for the request flows.

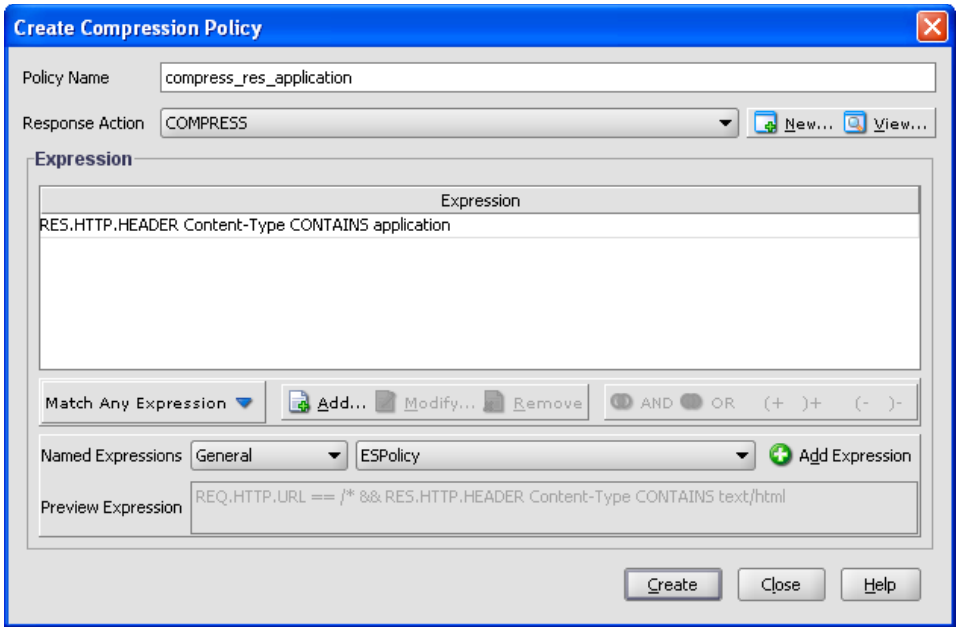
Set the compression response action to NOCOMPRESS. Again, several expressions can be combined in an “OR” matching algorithm.

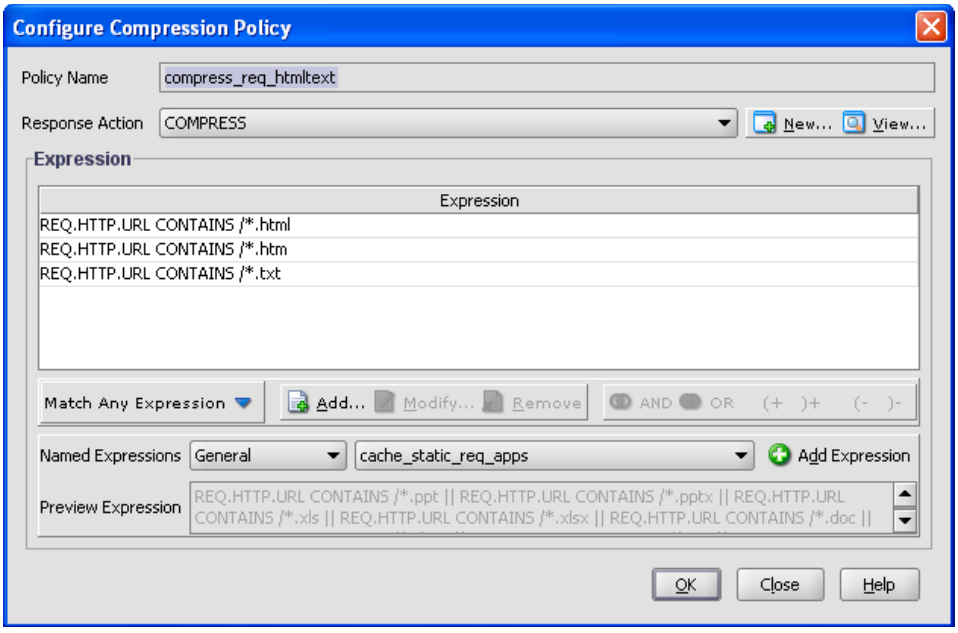
We now add the policies for the compressible content types.

Notice this set of compression policies is for the response flows.

Set the compression response action to COMPRESS. Several expressions can be combined in an “OR” matching algorithm.

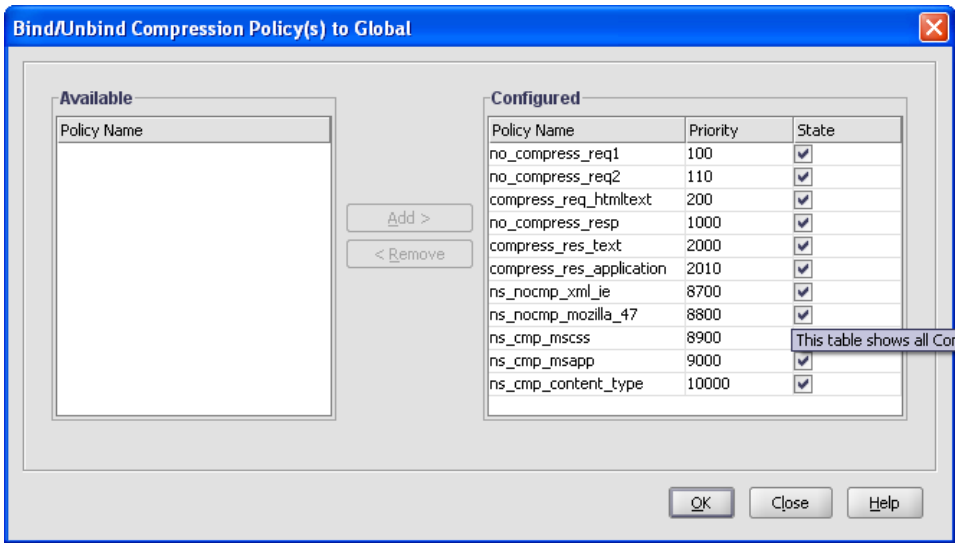






Notice this set of compression policies is for the request flows.

Set the compression response action to COMPRESS. Again, several expressions can be combined in an "OR" matching algorithm.



Now we need to activate the compression policies. From the GUI, select NetScaler ➥ Compression ➥ HTTP ➥ Global Bindings.

We give each policy a priority. The lower the priority takes precedence in evaluation.

We place the no-compress policies higher in the stack, and the compress policies lower in the stack. If the evaluation falls through the no-compress policy evaluation, it is most likely compressible.

Request policies need to be evaluated first before response policies. The default policies contain the prefix 'ns'.

Note:

Compression policies can be enabled on an individual VIP basis, under Load Balancing ➥ Virtual Servers ➥ Open ➥ Policies.

Disabling Compression on SAP Application Responses

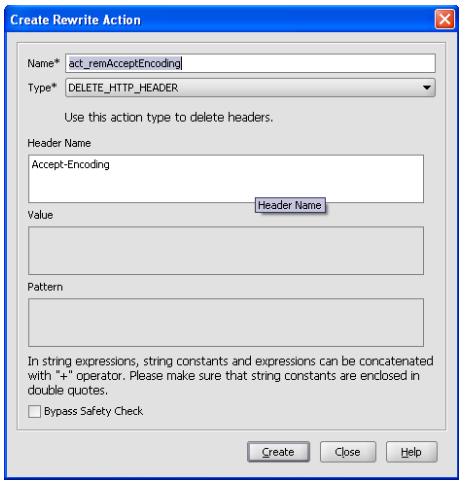
There are two ways to disable compressed content in responses from the SAP Application. 1) Remove the Accept-Encoding headers from the client requests and/or 2) Disable compression on the Service within the Citrix Application Switch. Disabling the compression algorithm on the SAP server free's it up to perform other duties and allows Citrix to offload the compression calculation, along with it's other Applicaiton acceleration technologies. If for some reason, SAP still sends a compressed response, the Citrix Applicaiton Switch will not try to re-compress it, and will pass it through.

Removing Accept-Encoding headers

One way to offload the compression calculation from the SAP servers is to remove the Accept-Encoding header's from the client requests, this way the Citrix Application Switch will end up doing all the compression work, and the SAP servers will not have to be burdened with that workload. This procedure is actually done within the Rewrite engine of the Citrix Application Switch.

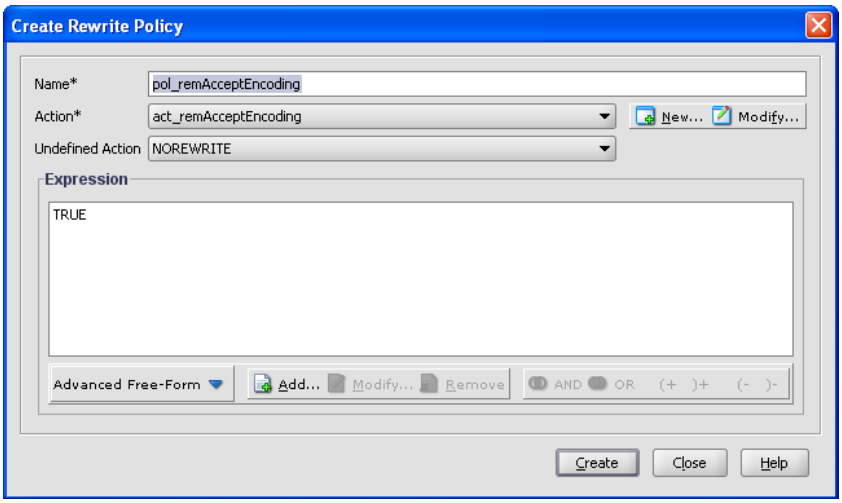
From the GUI, select NetScaler ➥ Rewrite ➥ Actions ➥ Add.

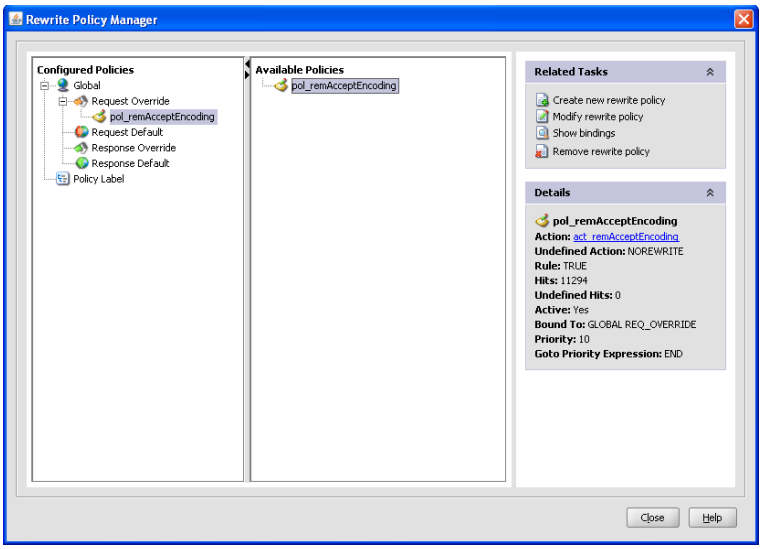
We first create a rewrite 'action' to delete the Accept-Encoding header on it's way to the back-end SAP servers.



From the GUI, select NetScaler ➥ Rewrite ➥ Policies ➥ Add.

We then create a rewrite 'policy' to engage the rewrite action. Give it an expression value of "TRUE".





From the GUI, put the mouse cursor on NetScaler ➥ Rewrite. The select Rewrite Policy Manager.

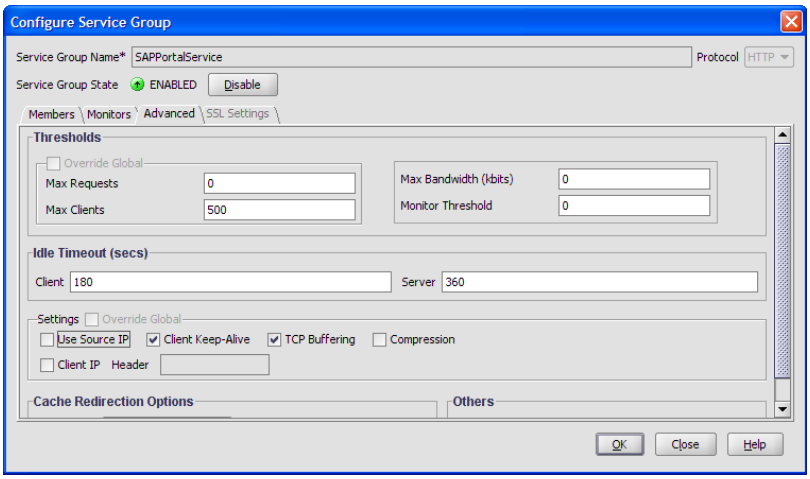
Grab the rewrite policy under Available Policies, and drag it over to Request Overrides.

All requests going to the back-end SAP servers will not have the Accept-Encoding header.

If you don't want to do this on a Global basis, you can assign this policy to the individual VIP.

Disabling compression on the Citrix VIP's

The other method for disabling compression, is to disable it on the Service or Service Group that talks directly to the backend SAP servers.



From the GUI, go to System ➥ Load Balancing ➥ Service Groups ➥ Open ➥ Advanced. De-select the Compression box.

Rewrite

Rewrite for SAP Applications

The Citrix Application Switch Rewrite feature is a general-purpose HTTP header and body modification utility. It allows you to add HTTP headers to an HTTP request or response, make modifications to individual HTTP headers, and delete HTTP headers. It also gives you control over modifying the HTTP body in requests and responses. This becomes a powerful tool for providing flexibility for the SAP Enterprise SOA architecture.

In the proof of concept developed for this guide, the re-write feature was used specifically the re-writing of content body maching-to-machine, soap:xml requests, and that all requests could be served by the Citrix Application Switch, Load Balanced and transported over secure communications using HTTPS with a non-standard port 50001 to the SAP Composite Application Framework and ERP Servers.

Because the Citrix Application Switch performs request and response, header and body rewrites this is made possible. Although in this case, header re-writes were not necessary because the Load Balancing function takes care of this automatically. Here we show how to use content body re-writes.

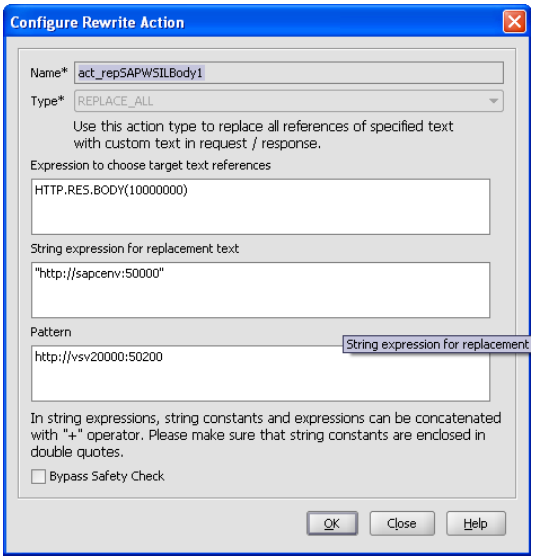
Rewrite for SAP Composite Application Framework

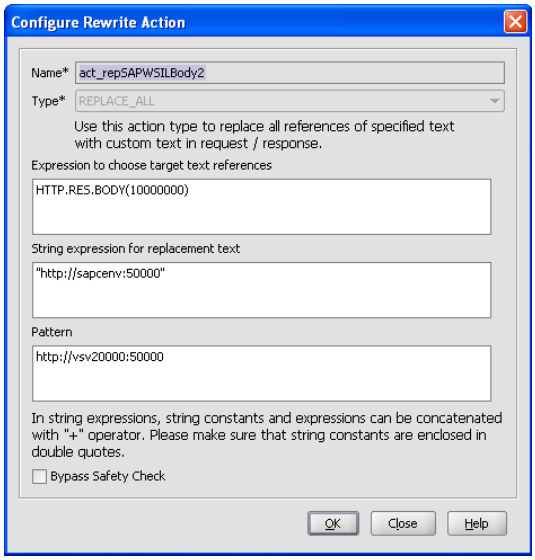
First we configure the rewrite actions for the two backend SAP Composite Application Framework servers.

From the GUI, select NetScaler ➥ Rewrite ➥ Actions ➥ Add.

The first two rewrite actions are performed on the content body for HTTP connections between the Citrix and the SAP Composite server.

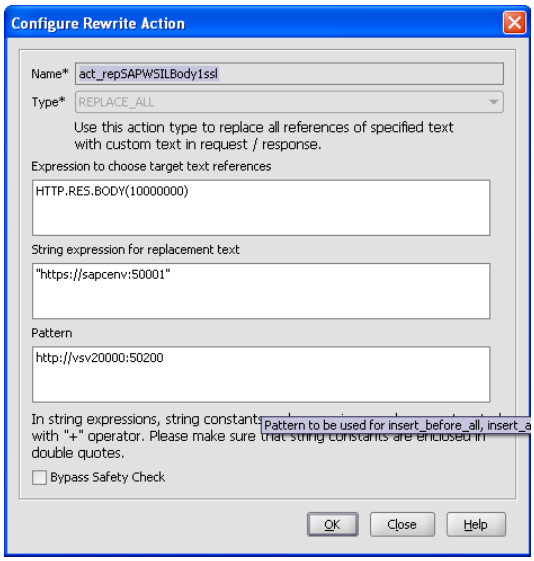
We are replacing any occurrence of the hostname “http://vsv20000:50000” or “http://vsv20000:50200” with our Citrix VIP of “http://sapcenv:50000” in the body of all responses that are sent through the Citrix to clients.

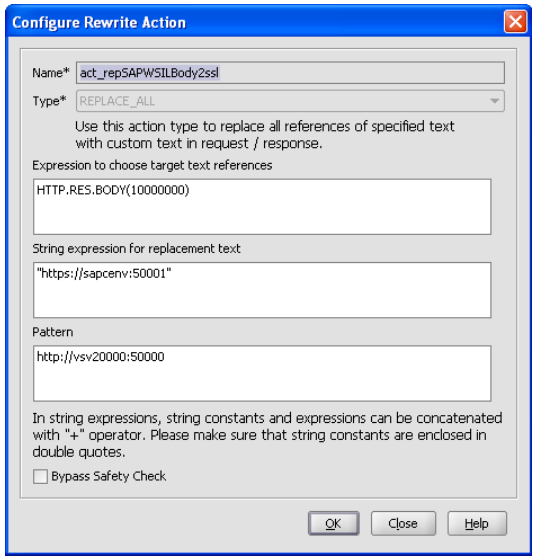


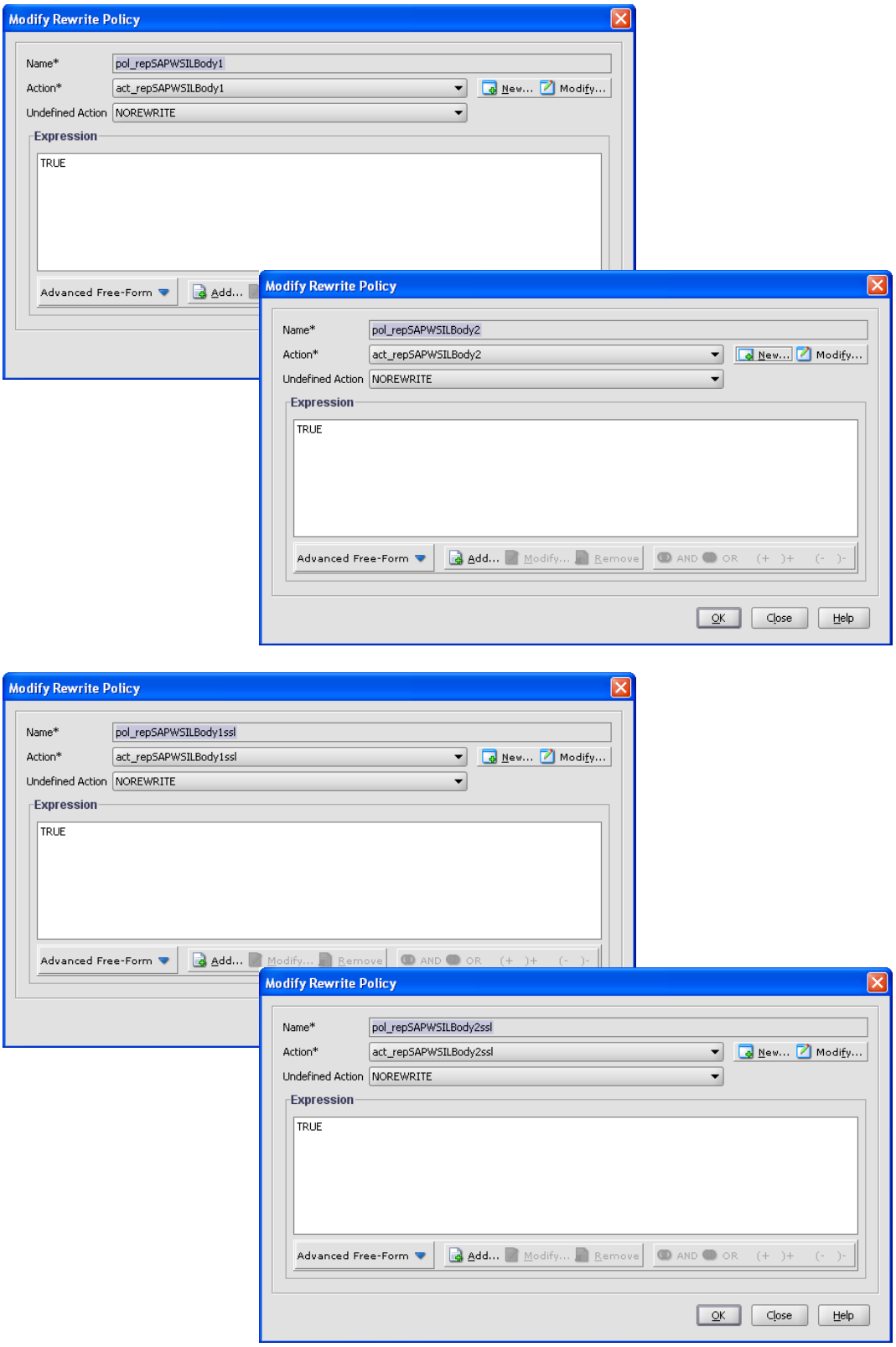


The next two rewrite actions are for HTTPS connections between the Citrix and the SAP Composite server.

We are replacing any occurrence of the hostname “http://vsv20000:50000” or “http://vsv20000:50200” with our Citrix VIP of “https://sapcenv:50001” in the body of all responses that are sent through the Citrix to clients.







Second we configure the rewrite policies for the two backend SAP Composite Application Framework servers, to engage the rewrite actions we just created.

Give the policy a name, invoke the action, and give them an expression value of TRUE.

From the GUI, select NetScaler ➥ Rewrite ➥ Policies ➥ Add.

The first two rewrite policies are for HTTP connections between the Citrix and the SAP Composite server.

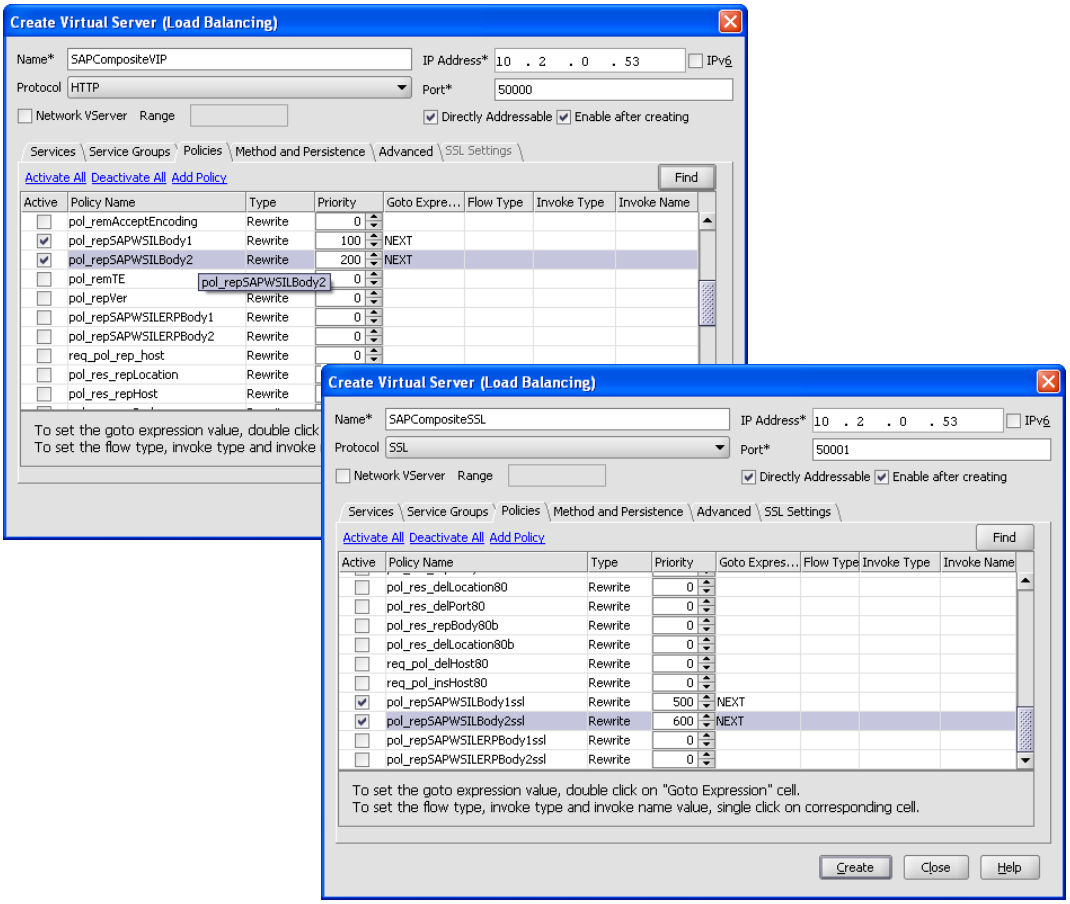
The next two rewrite policies are for HTTPS connections between the Citrix and the SAP Composite server.

The last and most important step is to bind the rewrite policies to the SAP Composite server VIPs, both the HTTP and HTTPS.

From the GUI, go to System ➥ Load Balancing ➥ <lb vip> ➥ Open ➥ Policies.

Find the rewrite policies for the SAP Composite server for HTTP in the list, select them, give them a priority and set the goto expression to NEXT.

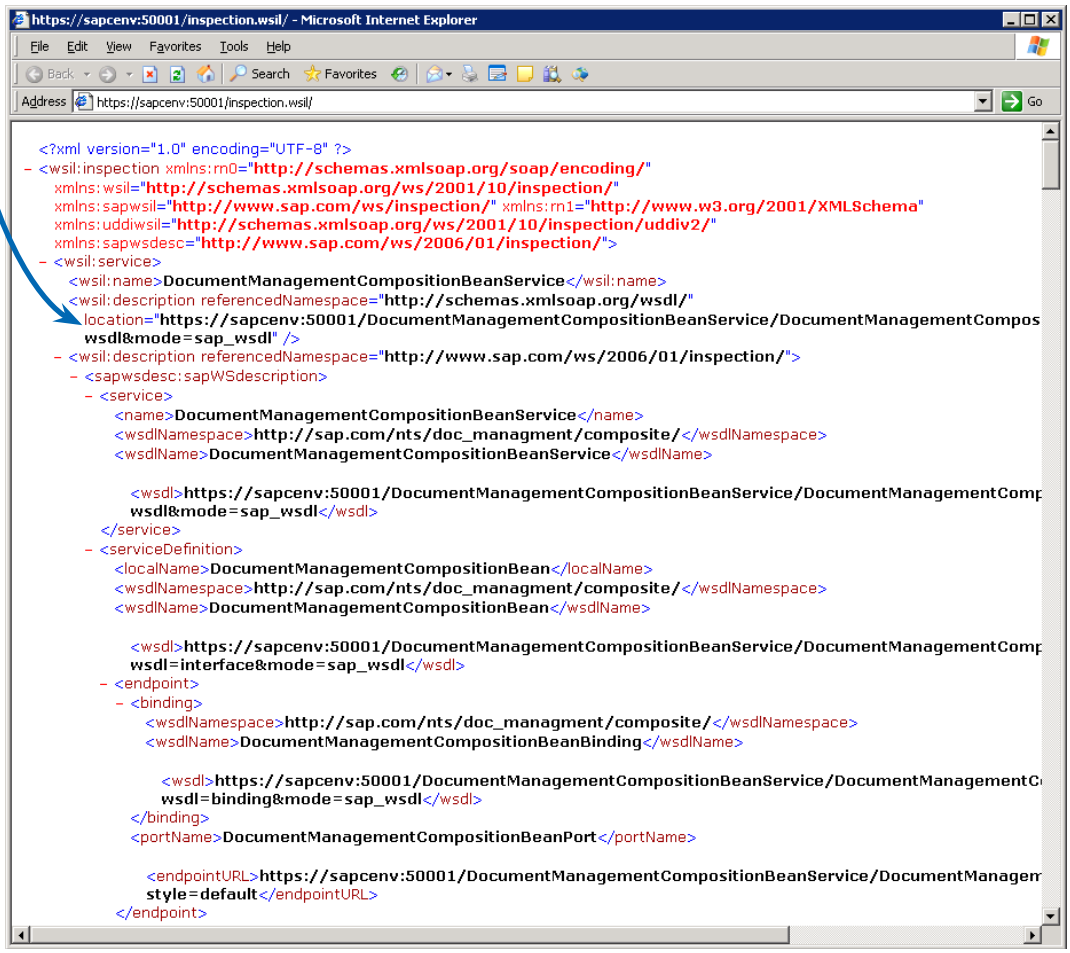
Repeat for the HTTPS policies.



Testing the Composite rewrite connection

To see if the rewrite is working properly, you can simulate the machine-to-machine interface by using a web browser. Open a web browser and connect to the VIP on the Citrix, http://sapcenv:50000/inspection.wsil, and https://sapcenv:50001/inspection.wsil.

If it is working properly the connection will be load balanced, persisted and the body content replaced with http://sapcenv:50000, or https://sapcenv:50001 in the case of https.



Rewrite for SAP ERP

First we configure the rewrite actions for the two backend SAP ERP servers.

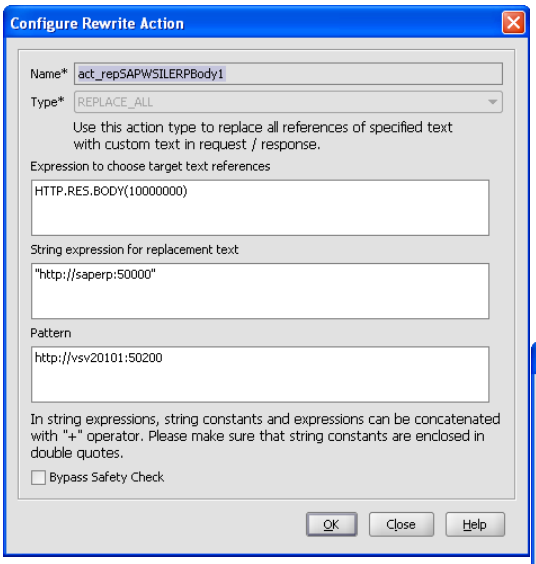
From the GUI, select NetScaler ➥ Rewrite ➥ Actions ➥ Add.

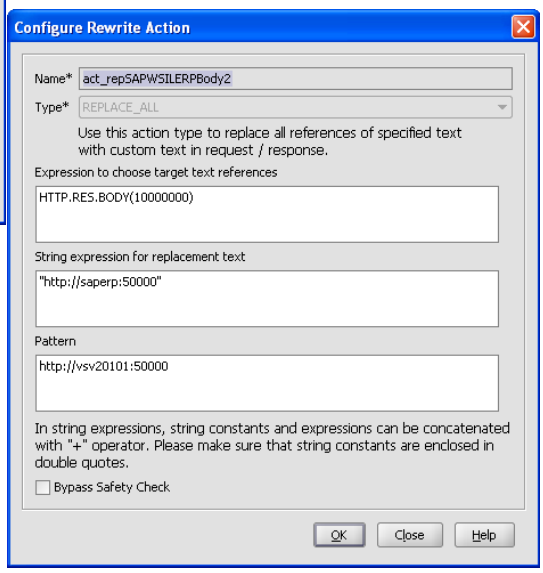
The first two rewrite actions are performed on the content body for HTTP connections between the Citrix and the SAP ERP server.

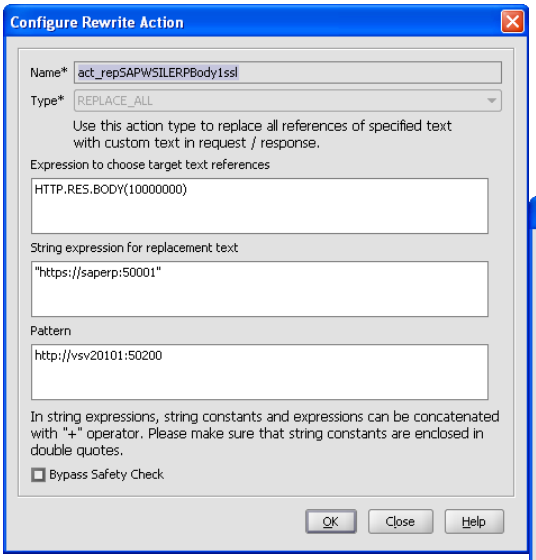
We are replacing any occurrence of the hostname “http://vsv20101:50000” or “http://vsv20101:50200” with our Citrix VIP of “http://saperp:50000” in the body of all responses that are sent through the Citrix to clients.

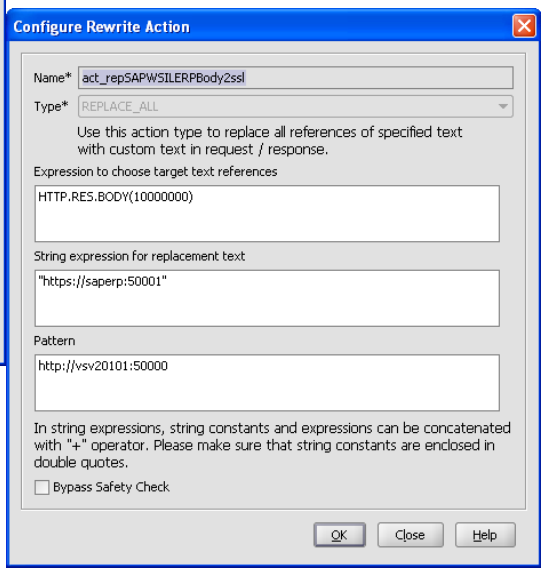
The next two rewrite actions performed on the content body for HTTPS connections between the Citrix and the SAP ERP server.

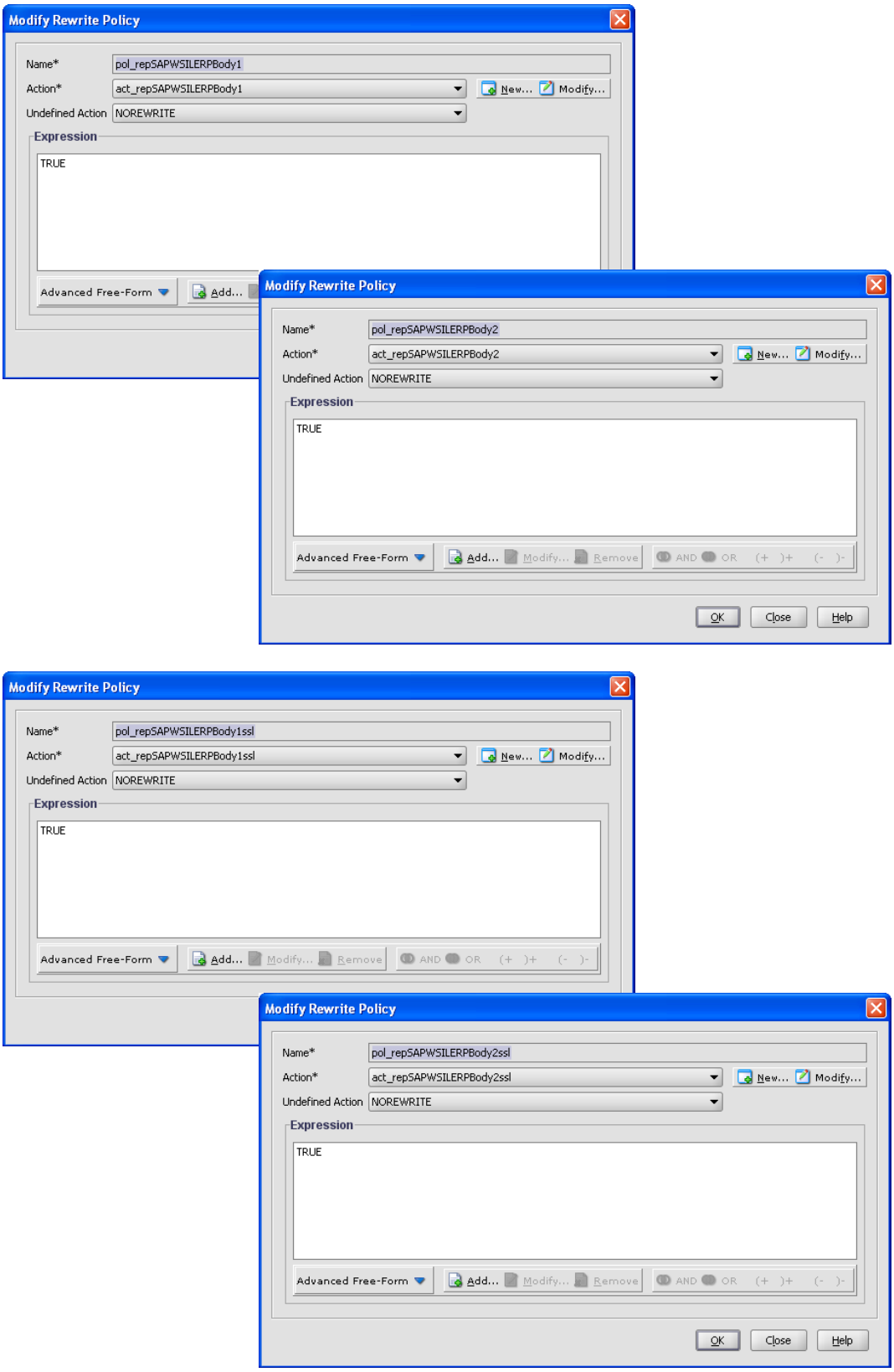
We are replacing any occurrence of the hostname “http://vsv20101:50000” or “http://vsv20101:50200” with our Citrix VIP of “https://saperp:50001” in the body of all responses that are sent through the Citrix to clients.











Second we configure the rewrite policies for the two backend SAP ERP servers, to engage the rewrite actions we just created.

Give the policy a name, invoke the action, and give them an expression value of TRUE.

From the GUI, select NetScaler ➥ Rewrite ➥ Policies ➥ Add.

The first two rewrite policies are for HTTP connections between the Citrix and the SAP Composite server.

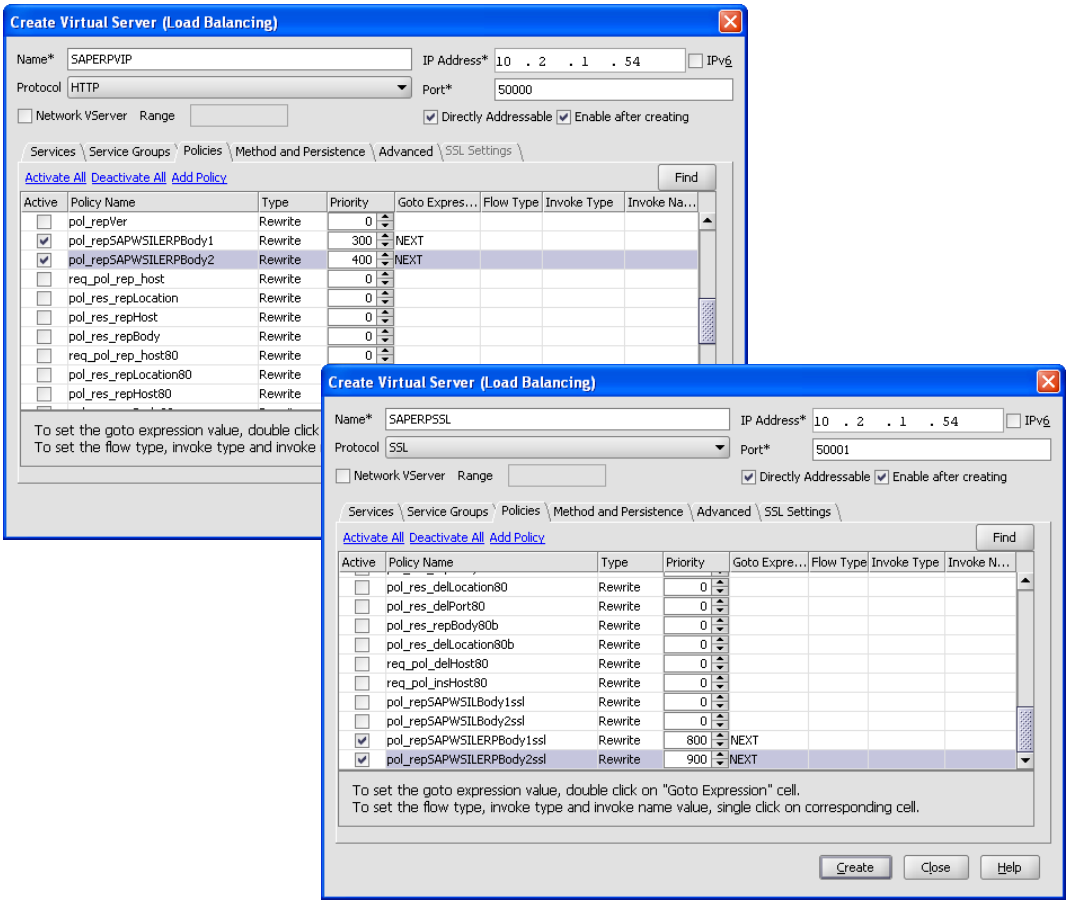
The next two rewrite policies are for HTTPS connections between the Citrix and the SAP Composite server.

The last and most important step is to bind the rewrite policies to the SAP ERP server VIPs, both the HTTP and HTTPS.

From the GUI, go to System ➥ Load Balancing ➥ <lb vip> ➥ Open ➥ Policies.

Find the rewrite policies for the SAP ERP server for HTTP in the list, select them, give them a priority and set the goto expression to NEXT.

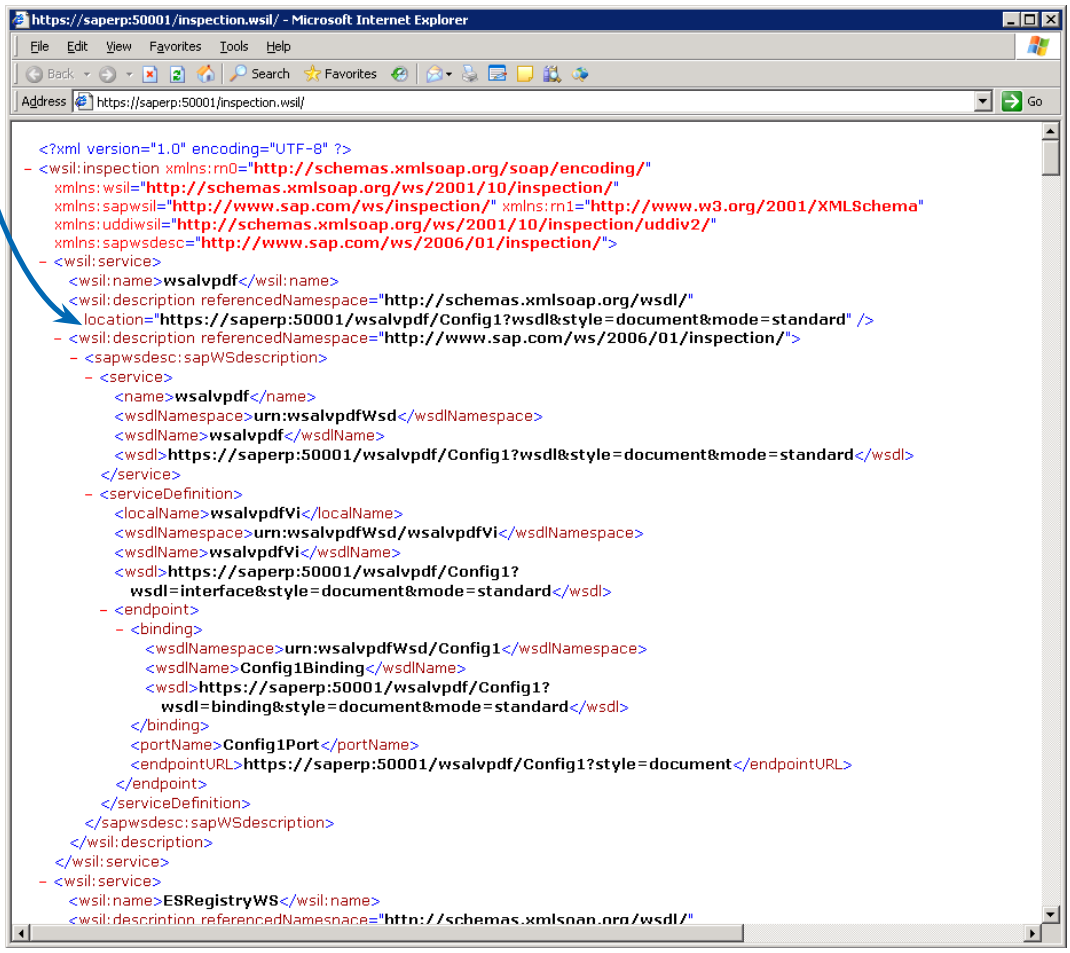
Repeat for the HTTPS policies.



Testing the ERP rewrite connection

To see if the rewrite is working properly, you can simulate the machine-to-machine interface by using a web browser. Open a web browser and connect to the VIP on the Citrix, http://saperp:50000/inspection.wsil, and https://saperp:50001/inspection.wsil.

If it is working properly the connection will be load balanced, persisted and the body content replaced with http://saperp:50000, or https://saperp:50001 in the case of https.

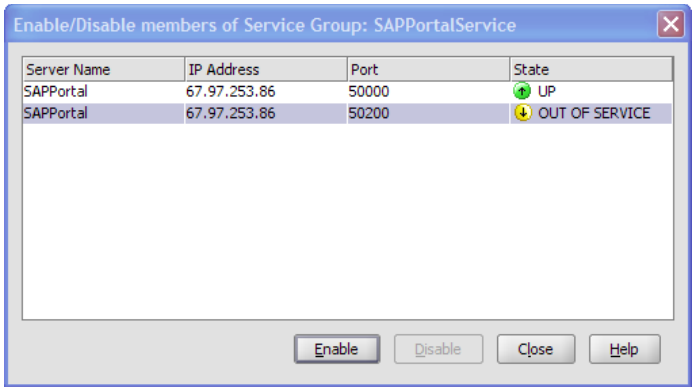


Troubleshooting

Load Balancing

During initial connectivity to the Citrix Application Switch, if things aren't working properly, get back to basics. Try to configure a one-to-one relationship between the client and the backend server, through the virtual server. You can do this by disabling all but one of the servers in the load balancing service group.

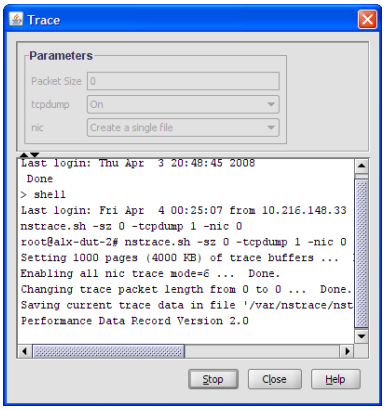
From the GUI, navigate to NetScaler ➥ System ➥ Load Balancing ➥ Service Groups ➥ <svc grp> ➥ Enable/Disable Members.



Run a trace

Running a trace will expose the flow of transactions between all points of interest, especially the client, load balancing VIPs and backend servers. Traces are especially helpful when digging in to find out if the proper headers are being exchanged between client <--> VIP and VIP <--> backend servers. A trace can be run directly on the Citrix Application Switch. Once downloaded this file can be opened and read with Wireshark, a free network trace utility, http://www.wireshark.org. By selecting the stream of packets, and Follow TCP Stream inside of wireshark, you can see the headers for both requests and responses.

From the GUI, navigate to NetScaler ➥ System ➥ Diagnostics ➥ New Trace ➥ Run.



Run a trace - on SAP Portal

Sometimes it is helpful to run a trace on the SAP server itself to see what headers it is receiving. This is especially useful with HTTPS communications as the traces of wireshark taken from the interface level are encrypted and unreadable.

To run a trace on the SAP Server, you have to set the http service property “httptrace” to “enable”.

The instructions for doing this can be found at http://help.sap.com/saphelp_nw70/helpdata/EN/52/46f6a089754e3a964a5d932eb9db8b/frameset.htm.

This property can also be set via Visual Administrator Tool -> Dispatcher -> Services-> Http Provider.

Appendix A - NetScaler Application Switch Configuration

Primary NetScaler

> #NS8.0 Build 51.4

set ns config -IPAddress 169.145.91.205 -netmask 255.255.255.0

enable ns feature WL SP LB CMP SSL IC REWRITE

enable ns mode FR L3 Edge USNIP PMTUD

set interface 0/1 -speed AUTO -duplex AUTO -flowControl RX -autoneg ENABLED -haMonitor ON -trunk OFF -lacpMode DISABLED -throughput 0

set interface 1/1 -speed AUTO -duplex AUTO -flowControl RX -autoneg ENABLED -haMonitor ON -trunk OFF -state DISABLED -lacpMode DISABLED -throughput 0

set interface 1/2 -speed AUTO -duplex AUTO -flowControl RX -autoneg ENABLED -haMonitor ON -trunk OFF -state DISABLED -lacpMode DISABLED -throughput 0

set interface 1/3 -speed AUTO -duplex AUTO -flowControl OFF -autoneg ENABLED -haMonitor OFF -trunk OFF -state DISABLED -lacpMode DISABLED -throughput 0

set interface 1/4 -speed AUTO -duplex AUTO -flowControl RX -autoneg ENABLED -haMonitor OFF -trunk ON -lacpMode DISABLED -throughput 0

add ns ip 10.2.0.54 255.255.255.0 -type MIP -vServer DISABLED

add ns ip 10.2.0.53 255.255.255.255 -type VIP -snmp DISABLED

add ns ip 10.2.0.55 255.255.255.0 -vServer DISABLED

add ns ip 10.2.1.55 255.255.255.0 -vServer DISABLED

add vlan 4

add vlan 200

add vlan 201

bind vlan 4 -ifnum 1/4

bind vlan 200 -ifnum 1/4 -tagged

bind vlan 200 -IPAddress 10.2.0.55 255.255.255.0

bind vlan 201 -ifnum 1/4 -tagged

bind vlan 201 -IPAddress 10.2.1.55 255.255.255.0

set cmp parameter -quantumSize 8192

add policy expression cache_static_res “RES.HTTP.HEADER Content-Type CONTAINS text || RES.HTTP.HEADER Content-Type CONTAINS image || RES.HTTP.HEADER Content-Type CONTAINS application”

add policy expression cmp_do_not_compress_res “RES.HTTP.HEADER Content-Type CONTAINS application/zip || RES.HTTP.HEADER Content-Type CONTAINS application/x-gzip || RES.HTTP.HEADER Content-Type CONTAINS application/pdf || RES.HTTP.HEADER Content-Type CONTAINS content/unknown || RES.HTTP.HEADER Content-Type CONTAINS [unknown]”

add policy expression cmp_compress_res “RES.HTTP.HEADER Content-Type CONTAINS application/* || RES.HTTP.HEADER Content-Type CONTAINS text/*”

add policy expression cmp_compress_req “REQ.HTTP.URL CONTAINS /*.htm || REQ.HTTP.URL CONTAINS /*.html”

add policy expression cache_static_req_text “REQ.HTTP.URL CONTAINS /*.html || REQ.HTTP.URL CONTAINS /*.htm || REQ.HTTP.URL CONTAINS /*.js || REQ.HTTP.URL CONTAINS /*.css”

add policy expression cache_static_req_images “REQ.HTTP.URL CONTAINS /*.gif || REQ.HTTP.URL CONTAINS /*.jpg || REQ.HTTP.URL CONTAINS /*.jpeg || REQ.HTTP.URL CONTAINS /*.png”

add policy expression cache_static_req_apps “REQ.HTTP.URL CONTAINS /*.ppt || REQ.HTTP.URL CONTAINS /*.pptx || REQ.HTTP.URL CONTAINS /*.xls || REQ.HTTP.URL CONTAINS /*.xlsx || REQ.HTTP.URL CONTAINS /*.doc || REQ.HTTP.URL CONTAINS /*.docx || REQ.HTTP.URL CONTAINS /*.txt || REQ.HTTP.URL CONTAINS /*.rtf”

add server SAPPortal 10.2.1.33

add server SAPComposite 10.2.0.33

add server SAPERP 10.2.1.34

add serviceGroup SAPCompositeService HTTP -maxClient 0 -maxReq 0 -cip DISABLED -usip NO -cltTimeout 180 -svrTimeout 360 -CKA

NO -TCPB YES -CMP YES

add serviceGroup SAPPortalService HTTP -maxClient 0 -maxReq 0 -cip DISABLED -usip NO -cltTimeout 180 -svrTimeout 360 -CKA NO -TCPB YES -CMP YES

add serviceGroup SAPERPService HTTP -maxClient 0 -maxReq 0 -cip DISABLED -usip NO -cltTimeout 180 -svrTimeout 360 -CKA NO -TCPB YES -CMP YES

add serviceGroup SAPCompositeSSL SSL -maxClient 0 -maxReq 0 -cip DISABLED -usip NO -cltTimeout 180 -svrTimeout 360 -CKA NO -TCPB NO -CMP YES

add serviceGroup SAPERPSSL SSL -maxClient 0 -maxReq 0 -cip DISABLED -usip NO -cltTimeout 180 -svrTimeout 360 -CKA NO -TCPB NO -CMP YES

add serviceGroup SAPPortalSSL SSL -maxClient 0 -maxReq 0 -cip DISABLED -usip NO -cltTimeout 180 -svrTimeout 360 -CKA NO -TCPB NO -CMP YES

add vpn intranetApplication route_migrate_1 ANY 192.168.0.0 -netmask 255.255.0.0 -destPort 1-65535 -interception TRANSPARENT

add cmp policy no_compress_req1 -rule “REQ.HTTP.URL CONTAINS /*.zip || REQ.HTTP.URL CONTAINS /*.cs || REQ.HTTP.URL CONTAINS /*.rar || REQ.HTTP.URL CONTAINS /*.arj || REQ.HTTP.URL CONTAINS /*.z || REQ.HTTP.URL CONTAINS /*.gz || REQ.HTTP.URL CONTAINS /*.tar” -resAction NOCOMPRESS

add cmp policy no_compress_req2 -rule “REQ.HTTP.URL CONTAINS /*.lzh || REQ.HTTP.URL CONTAINS /*.cab || REQ.HTTP.URL CONTAINS /*.hqx || REQ.HTTP.URL CONTAINS /*.ace || REQ.HTTP.URL CONTAINS /*.jar || REQ.HTTP.URL CONTAINS /*.ear || REQ.HTTP.URL CONTAINS /*.compressed” -resAction NOCOMPRESS

add cmp policy compress_res_text -rule “REQ.HTTP.HEADER Content-Type CONTAINS text” -resAction COMPRESS

add cmp policy compress_res_application -rule “REQ.HTTP.HEADER Content-Type CONTAINS application” -resAction COMPRESS

add cmp policy compress_req_htmltext -rule “REQ.HTTP.URL CONTAINS /*.html || REQ.HTTP.URL CONTAINS /*.htm || REQ.HTTP.URL CONTAINS /*.txt” -resAction COMPRESS

add cmp policy no_compress_resp -rule “RES.HTTP.HEADER Content-Type CONTAINS application/zip || RES.HTTP.HEADER Content-Type CONTAINS application/pdf || RES.HTTP.HEADER Content-Type CONTAINS content/unknown || RES.HTTP.HEADER Content-Type CONTAINS [unknown]” -resAction NOCOMPRESS

add lb vserver SAPCompositeVIP HTTP 10.2.0.53 50000 -persistenceType RULE -lbMethod TOKEN -rule “HTTP.REQ.COOKIE.VALUE(\”saplb_*\”)” -cltTimeout 180

add lb vserver SAPPortalVIP HTTP 10.2.1.53 50000 -persistenceType RULE -lbMethod TOKEN -rule “HTTP.REQ.COOKIE.VALUE(\”saplb_*\”)” -cltTimeout 180

add lb vserver SAPERPVIP HTTP 10.2.1.54 50000 -persistenceType RULE -lbMethod TOKEN -rule “HTTP.REQ.COOKIE.VALUE(\”saplb_*\”)” -cltTimeout 180

add lb vserver SAPPortalSSL SSL 10.2.1.53 50001 -persistenceType RULE -lbMethod TOKEN -rule “HTTP.REQ.COOKIE.VALUE(\”saplb_*\”)” -cltTimeout 180

add lb vserver SAPCompositeSSL SSL 10.2.0.53 50001 -persistenceType RULE -lbMethod TOKEN -rule “HTTP.REQ.COOKIE.VALUE(\”saplb_*\”)” -cltTimeout 180

add lb vserver SAPERPSSL SSL 10.2.1.54 50001 -persistenceType RULE -lbMethod TOKEN -rule “HTTP.REQ.COOKIE.VALUE(\”saplb_*\”)” -cltTimeout 180

set ns rpcNode 169.145.91.205 -password 8a7b474124957776a0cd31b862cbe4d72b5cbd59868a136d4bdeb56cf03b28 -encrypted -srcIP 169.145.91.205

set responder param -undefAction NOOP

add rewrite action act_remAcceptEncoding delete_http_header Accept-Encoding

add rewrite action req_act_replaceHttpVer replace HTTP.REQ.VERSION.MINOR “\”0\””

add rewrite action req_act_removeTEHeader delete_http_header TE

add rewrite action act_repSAPWSILBody1 replace_all HTTP.RES.BODY(10000000) “\”http://sapcenv:50000\”” -pattern http://vsv20000:50200

add rewrite action act_repSAPWSILBody2 replace_all HTTP.RES.BODY(10000000) “\”http://sapcenv:50000\”” -pattern http://vsv20000:50000

add rewrite action act_repSAPWSILERPBody1 replace_all HTTP.RES.BODY(10000000) “\”http://saperp:50000\”” -pattern http://vsv20101:50200

add rewrite action act_repSAPWSILERPBody2 replace_all HTTP.RES.BODY(10000000) “\”http://saperp:50000\”” -pattern http://vsv20101:50000

add rewrite action act_repSAPWSILBody1ssl replace_all HTTP.RES.BODY(10000000) “\”https://sapcenv:50001\”” -pattern http://vsv20000:50200

add rewrite action act_repSAPWSILBody2ssl replace_all HTTP.RES.BODY(10000000) “\”https://sapcenv:50001\”” -pattern http://vsv20000:50000

add rewrite action act_repSAPWSILERPBody1ssl replace_all HTTP.RES.BODY(10000000) “\”https://saperp:50001\”” -pattern http://vsv20101:50200

add rewrite action act_repSAPWSILERPBody2ssl replace_all HTTP.RES.BODY(10000000) “\”https://saperp:50001\”” -pattern http://

vsv20101:50000
add rewrite policy pol_remAcceptEncoding TRUE act_remAcceptEncoding NOREWRITE
add rewrite policy pol_repSAPWSILBody1 TRUE act_repSAPWSILBody1 NOREWRITE
add rewrite policy pol_repSAPWSILBody2 TRUE act_repSAPWSILBody2 NOREWRITE
add rewrite policy pol_remTE TRUE req_act_removeTEHeader NOREWRITE
add rewrite policy pol_repVer TRUE req_act_replaceHttpVer NOREWRITE
add rewrite policy pol_repSAPWSILERPBody1 TRUE act_repSAPWSILERPBody1 NOREWRITE
add rewrite policy pol_repSAPWSILERPBody2 TRUE act_repSAPWSILERPBody2 NOREWRITE
add rewrite policy pol_repSAPWSILBody1ssl TRUE act_repSAPWSILBody1ssl NOREWRITE
add rewrite policy pol_repSAPWSILBody2ssl TRUE act_repSAPWSILBody2ssl NOREWRITE
add rewrite policy pol_repSAPWSILERPBody1ssl TRUE act_repSAPWSILERPBody1ssl NOREWRITE
add rewrite policy pol_repSAPWSILERPBody2ssl TRUE act_repSAPWSILERPBody2ssl NOREWRITE
bind rewrite global pol_remAcceptEncoding 10 NEXT -type REQ_OVERRIDE
bind rewrite global pol_repVer 20 NEXT -type REQ_OVERRIDE
bind rewrite global pol_remTE 30 NEXT -type REQ_OVERRIDE
set rewrite param -undefAction NOREWRITE
bind serviceGroup SAPCompositeService SAPComposite 50000 -serverID 50000
bind serviceGroup SAPCompositeService SAPComposite 50200 -serverID 50200
bind serviceGroup SAPPortalService SAPPortal 50000 -serverID 50000
bind serviceGroup SAPPortalService SAPPortal 50200 -serverID 50200
bind serviceGroup SAPERPService SAPERP 50000 -serverID 50000
bind serviceGroup SAPERPService SAPERP 50200 -serverID 50200
bind serviceGroup SAPCompositeSSL SAPComposite 50001 -serverID 50001
bind serviceGroup SAPCompositeSSL SAPComposite 50201 -serverID 50201
bind serviceGroup SAPERPSSL SAPERP 50001 -serverID 50001
bind serviceGroup SAPERPSSL SAPERP 50201 -serverID 50201
bind serviceGroup SAPPortalSSL SAPPortal 50001 -serverID 50001
bind serviceGroup SAPPortalSSL SAPPortal 50201 -serverID 50201
bind lb vserver SAPCompositeVIP SAPCompositeService
bind lb vserver SAPPortalVIP SAPPortalService
bind lb vserver SAPERPVIP SAPERPService
bind lb vserver SAPCompositeSSL SAPCompositeSSL
bind lb vserver SAPERPSSL SAPERPSSL
bind lb vserver SAPPortalSSL SAPPortalSSL
bind lb vserver SAPCompositeVIP -policyName pol_repSAPWSILBody1 -priority 10 -gotoPriorityExpression NEXT -type RESPONSE
bind lb vserver SAPCompositeVIP -policyName pol_repSAPWSILBody2 -priority 20 -gotoPriorityExpression NEXT -type RESPONSE
bind lb vserver SAPERPVIP -policyName pol_repSAPWSILERPBody1 -priority 100 -gotoPriorityExpression NEXT -type RESPONSE
bind lb vserver SAPERPVIP -policyName pol_repSAPWSILERPBody2 -priority 200 -gotoPriorityExpression NEXT -type RESPONSE
bind lb vserver SAPCompositeSSL -policyName pol_repSAPWSILBody1ssl -priority 300 -gotoPriorityExpression NEXT -type RESPONSE
bind lb vserver SAPCompositeSSL -policyName pol_repSAPWSILBody2ssl -priority 400 -gotoPriorityExpression NEXT -type RESPONSE
bind lb vserver SAPERPSSL -policyName pol_repSAPWSILERPBody1ssl -priority 500 -gotoPriorityExpression NEXT -type RESPONSE
bind lb vserver SAPERPSSL -policyName pol_repSAPWSILERPBody2ssl -priority 600 -gotoPriorityExpression NEXT -type RESPONSE
bind lb monitor ping SAPCompositeService
bind lb monitor ping SAPPortalService
bind lb monitor ping SAPERPService
bind lb monitor https-ecv SAPCompositeSSL
bind lb monitor https-ecv SAPERPSSL
bind lb monitor https-ecv SAPPortalSSL
add ssl certKey ns-server-certificate -cert ns-server.cert -key ns-server.key
add ssl certKey sapvip.key.pair -cert /nsconfig/ssl/sapvip.cer.cert -key /nsconfig/ssl/sapvip.cer.key

```
add ssl certKey sapvipCA.key.pair -cert /nsconfig/ssl/sapvip.cer-root.cert -key /nsconfig/ssl/sapvip.cer-root.key
add ssl certKey SAPPortal.key.pair -cert /nsconfig/ssl/client_certkey.pem -key /nsconfig/ssl/client_certkey.pem -passcrypt qHCCFK04chM=
set ssl service nshttps-10.2.1.55-443 -sessReuse ENABLED -sessTimeout 120 -cipherRedirect DISABLED -sslv2Redirect DISABLED
set ssl service nsrpcs-10.2.1.55-3008 -sessReuse ENABLED -sessTimeout 120 -cipherRedirect DISABLED -sslv2Redirect DISABLED
set ssl service nshttps-10.2.0.55-443 -sessReuse ENABLED -sessTimeout 120 -cipherRedirect DISABLED -sslv2Redirect DISABLED
set ssl service nsrpcs-10.2.0.55-3008 -sessReuse ENABLED -sessTimeout 120 -cipherRedirect DISABLED -sslv2Redirect DISABLED
set ssl service nshttps-10.2.0.54-443 -sessReuse ENABLED -sessTimeout 120 -cipherRedirect DISABLED -sslv2Redirect DISABLED
set ssl service nsrpcs-10.2.0.54-3008 -sessReuse ENABLED -sessTimeout 120 -cipherRedirect DISABLED -sslv2Redirect DISABLED
set ssl service nskrpcs-127.0.0.1-3009 -sessReuse ENABLED -sessTimeout 120 -cipherRedirect DISABLED -sslv2Redirect DISABLED
set ssl service nshttps-127.0.0.1-443 -sessReuse ENABLED -sessTimeout 120 -cipherRedirect DISABLED -sslv2Redirect DISABLED
set ssl service nsrpcs-127.0.0.1-3008 -sessReuse ENABLED -sessTimeout 120 -cipherRedirect DISABLED -sslv2Redirect DISABLED
set cache parameter -memLimit 1024 -via “NS-CACHE-8.0:    1” -verifyUsing HOSTNAME_AND_IP -maxPostLen 0 -prefetchMaxPending 4294967294 -enableBypass YES
set cache contentGroup DEFAULT -ignoreReloadReq NO -removeCookies NO -cacheControl “max-age=43200     “
set cache contentGroup BASEFILE -relExpiry 86000 -maxResSize 256 -memLimit 2
set cache contentGroup DELTAJS -relExpiry 86000 -insertAge NO -maxResSize 256 -memLimit 1 -pinned YES
add cache policy SAP_cache_control_header -rule “RES.HTTP.HEADER Cache-Control CONTAINS max-age” -action CACHE -storeInGroup DEFAULT
add cache policy Do_not_Cache -rule “REQ.IP.DESTIP == 10.2.1.0 -netmask 255.255.255.0 || REQ.IP.DESTIP == 10.2.0.0 -netmask 255.255.255.0” -action NOCACHE
bind cache global SAP_cache_control_header -priority 1 -precedeDefRules YES
bind cmp global no_compress_req1 -priority 100
bind cmp global no_compress_req2 -priority 110
bind cmp global compress_req_htmltext -priority 200
bind cmp global no_compress_resp -priority 1000
bind cmp global compress_res_text -priority 2000
bind cmp global compress_res_application -priority 2010
bind vpn global -intranetApplication route_migrate_1
set lb sipParameters -addRportVip ENABLED
bind ssl service nshttps-10.2.1.55-443 -certkeyName ns-server-certificate
bind ssl service nsrpcs-10.2.1.55-3008 -certkeyName ns-server-certificate
bind ssl service nshttps-10.2.0.55-443 -certkeyName ns-server-certificate
bind ssl service nsrpcs-10.2.0.55-3008 -certkeyName ns-server-certificate
bind ssl service nshttps-10.2.0.54-443 -certkeyName ns-server-certificate
bind ssl service nsrpcs-10.2.0.54-3008 -certkeyName ns-server-certificate
bind ssl service nskrpcs-127.0.0.1-3009 -certkeyName ns-server-certificate
bind ssl service nshttps-127.0.0.1-443 -certkeyName ns-server-certificate
bind ssl service nsrpcs-127.0.0.1-3008 -certkeyName ns-server-certificate
bind ssl vserver SAPPortalSSL -certkeyName sapvip.key.pair
bind ssl vserver SAPPortalSSL -certkeyName sapvipCA.key.pair -CA
bind ssl vserver SAPCompositeSSL -certkeyName sapvip.key.pair
bind ssl vserver SAPCompositeSSL -certkeyName sapvipCA.key.pair -CA
bind ssl vserver SAPERPSSL -certkeyName sapvip.key.pair
bind ssl vserver SAPERPSSL -certkeyName sapvipCA.key.pair -CA
set ns hostName ns
```

Citrix Worldwide

Worldwide headquarters

Citrix Systems, Inc.
851 West Cypress Creek Road
Fort Lauderdale, FL 33309
USA
T +1 800 393 1888
T +1 954 267 3000

Regional headquarters

Americas

Citrix Silicon Valley
4988 Great America Parkway
Santa Clara, CA 95054
USA
T +1 408 790 8000

Europe

Citrix Systems International GmbH
Rheinweg 9
8200 Schaffhausen
Switzerland
T +41 52 635 7700

Asia Pacific

Citrix Systems Hong Kong Ltd.
Suite 3201, 32nd Floor
One International Finance Centre
1 Harbour View Street
Central
Hong Kong
T +852 2100 5000

Citrix Online division

5385 Hollister Avenue
Santa Barbara, CA 93111
USA
T +1 805 690 6400

www.citrix.com

About Citrix

Citrix Systems, Inc. (Nasdaq:CTXS) is the global leader and the most trusted name in application delivery infrastructure. More than 200,000 organizations worldwide rely on Citrix to deliver any application to users anywhere with the best performance, highest security and lowest cost. Citrix customers include 100% of the Fortune 100 companies and 98% of the Fortune Global 500, as well as hundreds of thousands of small businesses and prosumers. Citrix has approximately 6,200 channel and alliance partners in more than 100 countries. Annual revenue in 2006 was $1.1 billion.