

---

# Deploying Oracle PeopleSoft 9.2 with NetScaler

---

## Deployment Guide



---

This deployment guide focuses on defining the deployment process for Oracle PeopleSoft 9.2 with Citrix NetScaler. It includes information on setting up basic deployment and optimization.

## Table of Contents

Introduction	3
Configuration	3
NetScaler features to be enabled	4
Steps for load balancing configuration	4
Solution Description	5
Quick Configuration Table	5
Configuring Load Balancing	7
Verification	11
Authentication	11
Configuring Optimization on NetScaler	12
HTTP Compression	12
Integrated Caching	14
Front End Optimization	18
Conclusion	20

Citrix NetScaler is a world-class product with the proven ability to load balance, accelerate, optimize, and secure enterprise applications.

For several years, Citrix has completed certifications and provided deployment guides for key enterprise applications. NetScaler's rich application delivery capabilities significantly enhance the performance of these applications. With a comprehensive feature set, It provides availability, scalability, optimization and security for Oracle PeopleSoft deployments.

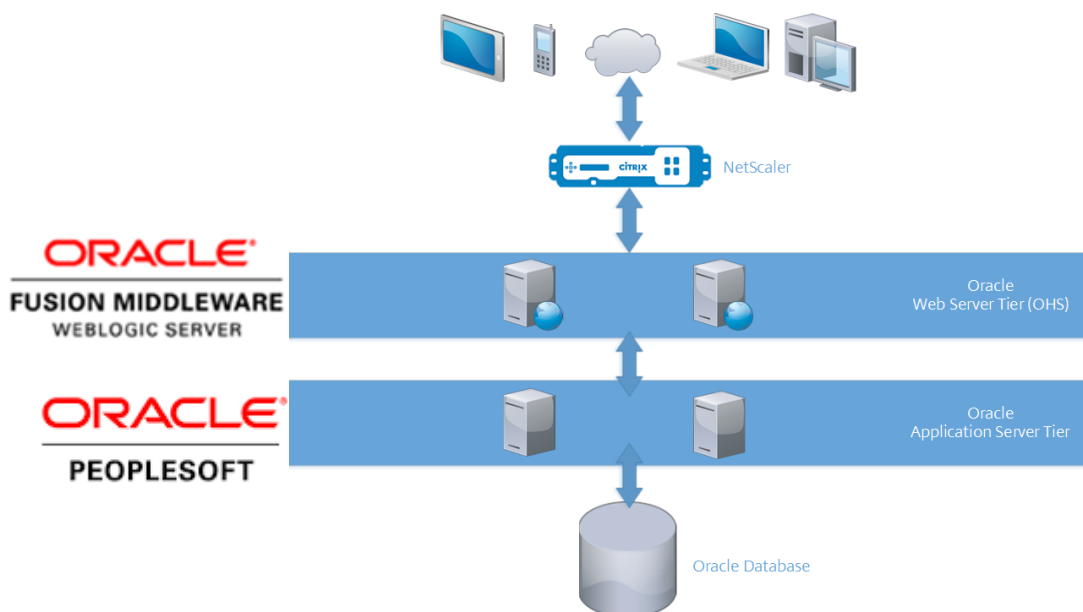
## Introduction

This guide defines the process for deploying Oracle PeopleSoft 12c with NetScaler.

Citrix NetScaler is a world class application delivery controller, with the proven ability to load balance, accelerate, secure and optimize enterprise applications.

Oracle's PeopleSoft applications are designed to address the most complex business requirements. They provide comprehensive business and industry solutions, enabling organizations to increase productivity, accelerate business performance, and provide a lower cost of ownership.

## Configuration



**Recommended Product Versions**

Product	Version
Oracle PeopleSoft	9.2
NetScaler VPX	11.0 (Enterprise/Platinum License) – Load Balancing, Compression, Caching and FEO 11.0 (Standard License) – Only Load Balancing

**NetScaler features**

The following NetScaler features are discussed in this deployment guide.

- Load balancing
- Front End Optimization, Compression, Caching

**Load balancing**

NetScaler load balancing evenly distributes requests to backend servers. Multiple algorithms (such as LEAST-CONNECTION, ROUNDROBIN etc.) are supported to provide efficient load balancing logic for every application server.

**HTTP Compression**

Compression of HTTP traffic using standard GZIP/DEFLATE compression methods.

**Front End Optimization (FEO)**

Advance optimization feature, FEO enables NetScaler to significantly accelerate web content with various acceleration methods such as image compression etc.

**Integrated Caching**

Content caching allows NetScaler to serve frequently used content without requiring round trips to the source webserver.

**Other considerations**

- Make sure you have installed, at a minimum, one license on the NetScaler appliance.
- Set the time zone and a NTP (Network Time Protocol) server, and check the date and time on the NetScaler virtual appliance, as server connections can be sensitive to time differences.
- Configure your DNS settings properly: Note that for the purposes of certificate-based authentication, all addressable hosts that are part of the network setup should have resolvable domain names, not just IP addresses.

## Quick Configuration Table

Configuration Item	Version				
Load Balancing (Traffic Management>Load Balancing>Virtual Servers in the GUI)	Virtual Servers: PeopleSoft_lb_ssl, PeopleSoft_lb (Suggested Names)				
	<table><tr><th>pslb_8080</th><th>pslb_80</th></tr><tr><td><b>Protocol:</b> HTTP <b>Port:</b> 8080 (or alternate as per your configuration) <b>Load Balancing Method:</b> Roundrobin/LeastConnection <b>Services Bound:</b> Ps1 Ps2 <b>Compression Policy:</b> Ps_Compression_Test <b>Cache Policy:</b> Ps_Cache_Test <b>FEO Policy:</b> Ps_Optimization_Test <b>Persistence:</b> Source IP  <b>CLI Commands:</b> add lb vserver pslb_8080 HTTP &lt;IP address for vserver&gt; 8080 -persistenceType SOURCEIP -lbMethod ROUNDROBIN -cltTimeout 180 -downStateFlush DISABLED</td><td><b>Protocol:</b> HTTP <b>Port:</b> 80 (or alternate as per your configuration) <b>Load Balancing Method:</b> Roundrobin/LeastConnection <b>Services Bound:</b> Ps1 Ps2 <b>Compression Policy:</b> Ps_Compression_Test <b>Cache Policy:</b> Ps_Cache_Test <b>FEO Policy:</b> Ps_Optimization_Test <b>Persistence:</b> Source IP  <b>CLI Commands:</b> add lb vserver pslb_80 HTTP &lt;IP address for vserver&gt; 80 -persistenceType SOURCEIP -lbMethod ROUNDROBIN -cltTimeout 180 -downStateFlush DISABLED</td></tr></table>	pslb_8080	pslb_80	<b>Protocol:</b> HTTP <b>Port:</b> 8080 (or alternate as per your configuration) <b>Load Balancing Method:</b> Roundrobin/LeastConnection <b>Services Bound:</b> Ps1 Ps2 <b>Compression Policy:</b> Ps_Compression_Test <b>Cache Policy:</b> Ps_Cache_Test <b>FEO Policy:</b> Ps_Optimization_Test <b>Persistence:</b> Source IP  <b>CLI Commands:</b> add lb vserver pslb_8080 HTTP <IP address for vserver> 8080 -persistenceType SOURCEIP -lbMethod ROUNDROBIN -cltTimeout 180 -downStateFlush DISABLED	<b>Protocol:</b> HTTP <b>Port:</b> 80 (or alternate as per your configuration) <b>Load Balancing Method:</b> Roundrobin/LeastConnection <b>Services Bound:</b> Ps1 Ps2 <b>Compression Policy:</b> Ps_Compression_Test <b>Cache Policy:</b> Ps_Cache_Test <b>FEO Policy:</b> Ps_Optimization_Test <b>Persistence:</b> Source IP  <b>CLI Commands:</b> add lb vserver pslb_80 HTTP <IP address for vserver> 80 -persistenceType SOURCEIP -lbMethod ROUNDROBIN -cltTimeout 180 -downStateFlush DISABLED
pslb_8080	pslb_80				
<b>Protocol:</b> HTTP <b>Port:</b> 8080 (or alternate as per your configuration) <b>Load Balancing Method:</b> Roundrobin/LeastConnection <b>Services Bound:</b> Ps1 Ps2 <b>Compression Policy:</b> Ps_Compression_Test <b>Cache Policy:</b> Ps_Cache_Test <b>FEO Policy:</b> Ps_Optimization_Test <b>Persistence:</b> Source IP  <b>CLI Commands:</b> add lb vserver pslb_8080 HTTP <IP address for vserver> 8080 -persistenceType SOURCEIP -lbMethod ROUNDROBIN -cltTimeout 180 -downStateFlush DISABLED	<b>Protocol:</b> HTTP <b>Port:</b> 80 (or alternate as per your configuration) <b>Load Balancing Method:</b> Roundrobin/LeastConnection <b>Services Bound:</b> Ps1 Ps2 <b>Compression Policy:</b> Ps_Compression_Test <b>Cache Policy:</b> Ps_Cache_Test <b>FEO Policy:</b> Ps_Optimization_Test <b>Persistence:</b> Source IP  <b>CLI Commands:</b> add lb vserver pslb_80 HTTP <IP address for vserver> 80 -persistenceType SOURCEIP -lbMethod ROUNDROBIN -cltTimeout 180 -downStateFlush DISABLED				
Service Configuration (System>Load Balancing>Services) Note: Both backend services are HTTP here	<table><tr><th>Ps1</th><th>Ps2</th></tr><tr><td><b>Protocol:</b> HTTP <b>Port:</b> 80 (or alternate as per your configuration) <b>IP:</b> IP address of 1st PeopleSoft server</td><td><b>Protocol:</b> HTTP <b>Port:</b> 80 (or alternate as per your configuration) <b>IP:</b> IP address of 2nd PeopleSoft server</td></tr></table> <b>CLI Commands:</b> add service PS1 <IP address for 1st front end server> HTTP 80 -gslb NONE -maxClient 0 -maxReq 0 -cip ENABLED X-Forwarded-for -usip NO -useproxyport NO -sp ON -cltTimeout 180 -svrTimeout 360 -CKA NO -TCPB NO -CMP YES add service PS2 <IP address for 2nd front end server> HTTP 80 -gslb NONE -maxClient 0 -maxReq 0 -cip DISABLED -usip NO -useproxyport NO -sp ON -cltTimeout 180 -svrTimeout 360 -CKA NO -TCPB NO -CMP YES	Ps1	Ps2	<b>Protocol:</b> HTTP <b>Port:</b> 80 (or alternate as per your configuration) <b>IP:</b> IP address of 1st PeopleSoft server	<b>Protocol:</b> HTTP <b>Port:</b> 80 (or alternate as per your configuration) <b>IP:</b> IP address of 2nd PeopleSoft server
	Ps1	Ps2			
<b>Protocol:</b> HTTP <b>Port:</b> 80 (or alternate as per your configuration) <b>IP:</b> IP address of 1st PeopleSoft server	<b>Protocol:</b> HTTP <b>Port:</b> 80 (or alternate as per your configuration) <b>IP:</b> IP address of 2nd PeopleSoft server				
Compression Policy Definition (Optimization>Integrated Caching>Policies)	<b>Policy Name:</b> PeopleSoft_Compression_Test <b>Response Action:</b> COMPRESS (GZIP/DEFLATE should work too) <b>Expression:</b> ns_true  <b>CLI Commands:</b> add cmp policy PeopleSoft_Compression_Test -rule ns_true -resAction GZIP bind lb vserver PeopleSoft_lb -policyName PeopleSoft_Compression_Test -priority 100 bind lb vserver PeopleSoft_lb_ssl -policyName PeopleSoft_Compression_Test -priority 100				

Configuration Item	Version
Cache Policy (Optimization>Integrated Caching>Policies)	<p>Policy Name: PeopleSoft_Cache_Test            Actions: CACHE            Cache Content Group: Test            Undefined-Result Action: -Global-undefined-result-action (or NOCACHE/RESET)            Expression: <i>ns_true</i></p> <p>Cache Content Group:            Name: Test            Type: HTTP            Expiry Method: Heuristic (Recommended)/Custom (if specific settings are required)            Default Expiry Times: As per requirement; set to 233 for test deployment.            Parameterization: Leave values as is (unless Cache selectors are in use; not configured for our test setup)            Memory: Define values as per your system limits            Others: Use default settings. All settings have context-sensitive help available if modification is required.</p> <p>CLI Commands:  <i>add cache policy PeopleSoft_Caching_Test -rule "SYS.EVAL_CLASSIC_EXPR(\"ns_true\")" -action CACHE -storeInGroup PeopleSoft_Caching_Test</i></p>
FEO (Front End Optimization) Policy (Optimization>Front end Optimization>Policies)	<p>Optimization Policy Name: PeopleSoft_Optimization_Test            Optimization Action: MODERATE (Preconfigured)            Expression: <i>HTTP.REQ.HEADER("Accept").CONTAINS("html")</i></p> <p><b>Alternate Configuration (Custom Policy):</b>            Optimization Policy Name: PeopleSoft_Optimization_TestCustom            Optimization Action: samplefeo            Expression: <i>HTTP.REQ.HEADER("Accept").CONTAINS("html")</i></p> <p><b>PeopleSoft_Optimization_TestCustom Configuration:</b>            Enabled Settings: JavaScript/Make Inline, JavaScript/Move to End of Body Tag, JavaScript/Minify, Image/Optimize, Image/Lazy Load, Image/Shrink to Attributes, Image/Optimize, Image/Convert to JXR format, Image/Convert GIF to PNG, CSS/Make Inline, CSS/Move to Head Tag, CSS/Minify, CSS/Image Inline, CSS/Combine, CSS/Convert Imports to Links, HTML/Remove Comments from HTML</p> <p>CLI Commands:  <i>add feo policy PeopleSoft_Optimization_Test "HTTP.REQ.HEADER(\"Accept\").CONTAINS(\"html\")" MODERATE</i>    <i>add feo policy PeopleSoft_Optimization_Testcustom "HTTP.REQ.HEADER(\"Accept\").CONTAINS(\"html\")" MS_SP_custom</i>    <i>bind lb vserver PeopleSoft_lb -policyName PeopleSoft_Optimization_Testcustom -priority 100 -gotoPriorityExpression END -type REQUEST</i>  <i>bind lb vserver PeopleSoft_lb_ssl -policyName PeopleSoft_Optimization_Test -priority 100 -gotoPriorityExpression END -type REQUEST</i></p>

## Solution Description

### Configuring Load Balancing

The configuration of load balancing consists of the definition of and load balancing virtual servers, as well as services that are linked to the LB vservers and bound to the individual Oracle PeopleSoft servers.

#### Step 1 - Define the load balancing virtual servers (LB vservers)

Log into the NetScaler GUI. On the Configuration tab, navigate to Traffic Management>Load Balancing>Virtual Servers. For this deployment exercise, we are load balancing two Oracle PeopleSoft server instances. To demonstrate the same, we create two load balancing virtual servers – Pslb\_80 (HTTP Port 80) and Pslb\_8080 (HTTP Port 8080).

When defining a new LB vserver, you will be presented with the settings screen. Here, set the protocol to HTTP for both vservers. Set the IP addresses to appropriate values.

## Load Balancing Virtual Server

**Basic Settings**

Create a virtual server by specifying a name, an IP address, a port, and a public IP address. If the application is accessible only from the local area address.  
You can configure multiple virtual servers to receive client requests, then

Name\*

Protocol\*

 ?

IP Address Type\*

IP Address\*

Port\*

► More

After clicking OK, you will see the Basic Settings screen for the LB vserver. Here, you may change settings such as the session persistence method, authentication and load balancing methods. Set session persistence to SOURCEIP and the load balancing method to LEASTCONNECTION for both virtual servers.

For more information on these features, please refer to <https://docs.citrix.com/en-us/netscaler/11.html>

## Load Balancing Virtual Server

Load Balancing Virtual Server | [Export as a Template](#)

Basic Settings

Name	pslb_8080	Listen Priority	-
Protocol	HTTP	Listen Policy Expression	NONE
State	DOWN	Range	1
IP Address	10.10.15.21	Redirection Mode	IP
Port	80	RHI State	PASSIVE
Traffic Domain	0	AppFlow Logging	ENABLED

Services and Service Groups

A service is a logical representation of an application running on a server.  
A service group enables you to manage a group of services as though it were a single service. After creating a service group, you can bind it to a virtual server, and you can add services to the group. You can also bind monitors to service groups.  
**Note:** Bind at least one service or service group to the virtual server.

Click **Continue** to display the advanced settings and select the method, persistence type, and any other configuration detail that you might need.

No Load Balancing Virtual Server Service Binding

No Load Balancing Virtual Server ServiceGroup Binding

Continue

Optionally, you can also set this vserver up for SSL. More information on this is available on the following page.



To enable an SSL-based LB vserver, you should add an SSL certificate and key pair. For this, you may use either a self-signed certificate generated on the NetScaler appliance or a CA (Certificate Authority) signed one. The steps for generating a self-signed certificate on the NetScaler are as follows –

1. Login to your NetScaler appliance via the Configuration Utility.
2. Select Traffic Management > SSL
3. On the right, under Tools, select Server Certificate Wizard.
4. Here, the wizard will lead you through the series of steps for generating the self signed certificate –
  - Generate the private key
  - Generate the CSR (Certificate Signing Request)
  - Generate the Certificate (using the ns-root.cer NetScaler root certificate)
  - Save the Certificate and Key pair

Alternatively, if a certificate and key pair is already available, the same can be added by navigating to SSL>Certificates and clicking on the Add button. For more details refer to <http://support.citrix.com/article/CTX109260>

To improve site security and achieve an A/A+ rating on the SSL Labs.com evaluation, refer to <https://www.citrix.com/blogs/06/09/scoring-an-a-at-ssl-labs-com-with-citrix-netscaler-update/>

## Step 2 – Define LBVS server service group binding

Now click on the Load Balancing Virtual Server Service Binding tab in the Service and Service Groups section, or alternatively, click on Services in the Traffic Management>Load Balancing subsection and then, click on the Add button.

Every LB service is linked to a server; this can either be a new server or an existing server already defined in the Servers subsection under Load Balancing. Service groups extend this by allowing the creation of a group of services. An LB vserver can use a set of services or a service group.

Here, define the names for the services for each PeopleSoft instance, the IP address (or choose from a list in the case of an existing server) for the new server and the protocol it operates on.

# Load Balancing Service

## Basic Settings

Service Name\*

☐ New Server ☒ Existing Server

Server\*

Protocol\*

Port\*

Traffic Domain



Hash ID

Server ID



Cache Type\*

☐ Cacheable

☒ Enable Service

☒ Health Monitoring

☒ AppFlow Logging

Number of Active Connections

Comments

Monitoring Connection Close Bit

▲ Less

OK

Cancel

## Recommended Best Practices:

- Name your server instances as per their role, not with the IP address (for example, the Oracle PeopleSoft servers can be named PeopleSoft1 and PeopleSoft2)
- As there will be multiple items linked to each application (LB vservers, services, policies among others), it is recommended that they be named appropriately for convenience. This will make using tools such as grep with the CLI a lot easier.

You should enable Health Monitoring if you would like to have NetScaler poll the server periodically to verify its health – it is recommended that this setting should not be disabled except for diagnostic purposes. This and additional settings can be accessed by clicking on the More dropdown (as shown above). If Health Monitoring is disabled, the appliance shows the server UP at all times. Bind these service groups to the appropriate LB vservers and confirm that they have been bound correctly by checking the same in the LB vserver Basic Settings screen. Add all the PeopleSoft servers to be load balanced and bind them to the load balancing virtual server.

## Load Balancing Virtual Server

Load Balancing Virtual Server | [Export as a Template](#)

Basic Settings

Name	pslb_80	Listen Priority	-
Protocol	HTTP	Listen Policy Expression	NONE
State	UP	Range	1
IP Address	10.105.157.213	Redirection Mode	IP
Port	80	RHI State	PASSIVE
Traffic Domain	0	AppFlow Logging	ENABLED

Services and Service Groups

2

Load Balancing Virtual Server Service Bindings

No Load Balancing Virtual Server ServiceGroup Binding

Persistence

Persistence	SOURCEIP	Time-out (mins)	2
		IPv4 Netmask	255.255.255.255
		IPv6 Mask Length	128

Finally, the LB vservers created will be displayed on the configuration screen to the right in the same screen that is obtained by accessing Traffic Management>Load Balancing>Virtual Servers.

This completes the load balancing configuration for Oracle PeopleSoft.

### Verification

The functioning solution can be verified with a default PeopleSoft installation by navigating to `https://<FQDN of LB vserver>/ps/ps/?cmd=login&languageCd=ENG&`

This will show the default login screen for PeopleSoft.

The image shows the Oracle PeopleSoft login interface. At the top, there is a white banner with the 'ORACLE' logo in red and 'PEOPLESOFT' in black. Below this, the background is a solid blue color. In the center, there are three white input fields: 'User ID', 'Password', and 'Select a Language'. The 'Select a Language' dropdown menu is currently set to 'English'. Below the input fields is a green 'Sign In' button. At the bottom, there is a checkbox labeled 'Enable Accessibility Mode'.

## Configuring Optimization on NetScaler

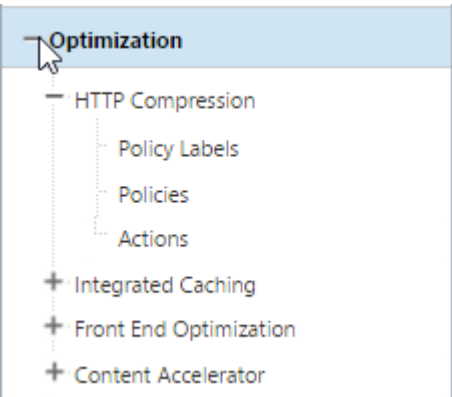
NetScaler provides a flexible, comprehensive suite of optimization capabilities that can be categorized as follows :

- HTTP Compression
- Integrated Caching
- Front End Optimization (additional optimization capabilities)

To configure HTTP Compression, Integrated Caching and Front End Optimization, expand the Optimization tab in the NetScaler GUI's left hand side navigation panel.

### HTTP Compression

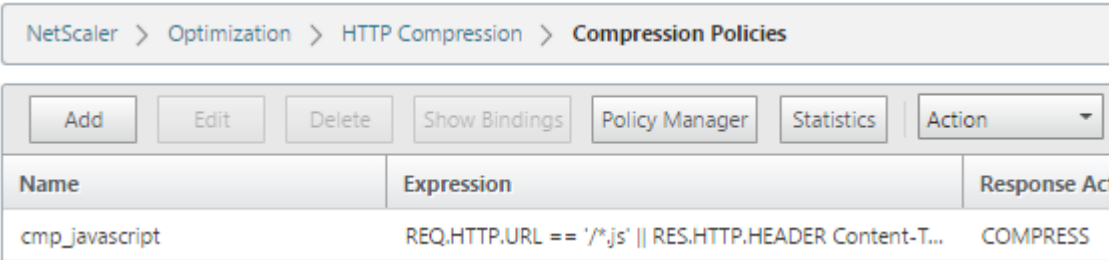
NetScaler's optimization suite is, like other NetScaler features, driven by a policy-action architecture.



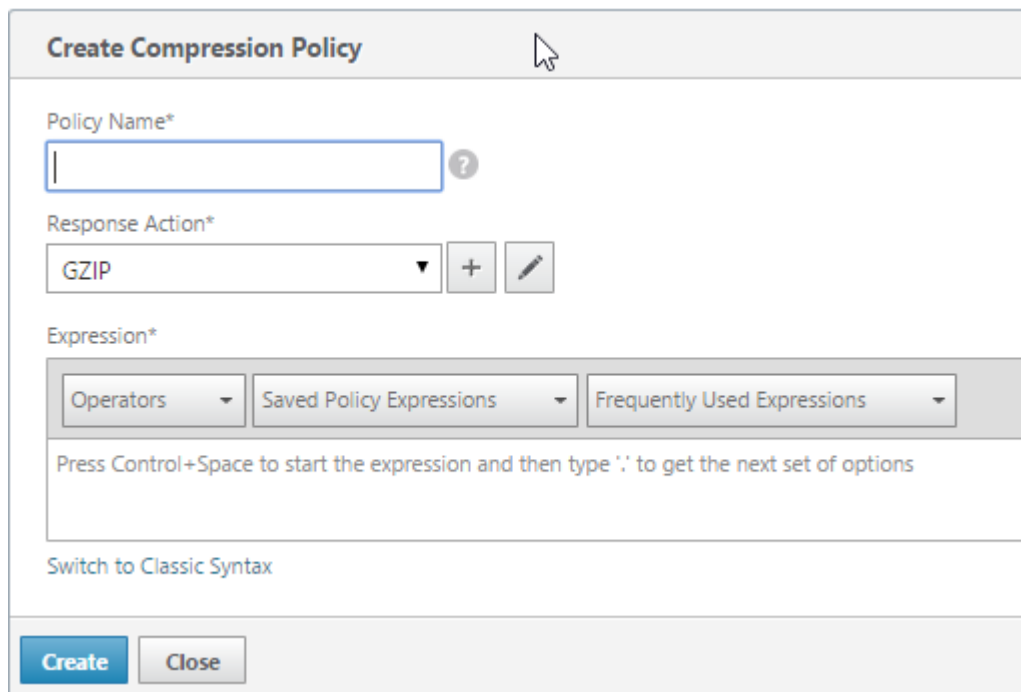
To enable HTTP Compression for a particular service, you should

- Define the HTTP Compression Policy and Action
- Bind the same to the relevant virtual server

To define the Compression Policy and Action, click on the Policies option under HTTP Compression, shown above. This gives you the following screen -



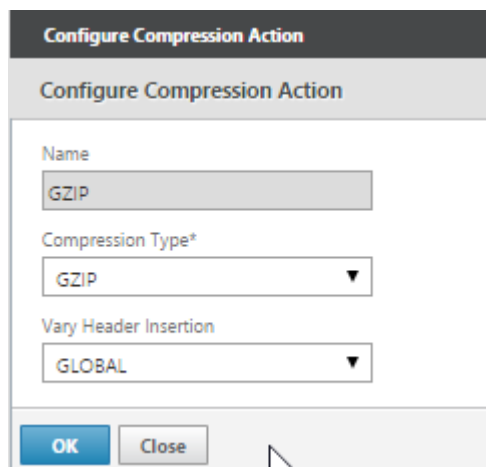
To add a new compression policy, click on the Add button. This will give you the following screen -



The 'Create Compression Policy' dialog box features a title bar with the text 'Create Compression Policy' and a mouse cursor icon. Below the title bar, there are three main sections: 'Policy Name\*' with a text input field and a help icon; 'Response Action\*' with a dropdown menu showing 'GZIP' and buttons for adding (+) and editing (pencil); and 'Expression\*' with three dropdown menus for 'Operators', 'Saved Policy Expressions', and 'Frequently Used Expressions'. Below these is a text area with the instruction 'Press Control+Space to start the expression and then type ':' to get the next set of options' and a link 'Switch to Classic Syntax'. At the bottom are 'Create' and 'Close' buttons.

Here, you can define a name for the policy, an Expression that defines when this policy is triggered (for example, when a particular URL is encountered. To make the policy apply to all content, use `ns_true` in the Expression window. For assistance here, click on the Frequently Used Expressions drop down) and the Response Action that should be taken. Here, the Actions available are COMPRESS (GZIP or DEFLATE compression, with GZIP given priority), GZIP (GZIP standard compression), DEFLATE (DEFLATE compression) and NOCOMPRESS.

Here, you have the option to either add a new Action or reconfigure the existing ones. You can Add using the + button, or edit/configure using the pencil-shaped button. Either option gives you a screen similar to the one shown below



The 'Configure Compression Action' dialog box has a title bar with the text 'Configure Compression Action'. Below the title bar, there are three main sections: 'Name' with a text input field showing 'GZIP'; 'Compression Type\*' with a dropdown menu showing 'GZIP'; and 'Vary Header Insertion' with a dropdown menu showing 'GLOBAL'. At the bottom are 'OK' and 'Close' buttons.

Vary Header Insertion is an option that is relevant for caching; the value of the Vary header allows for different cache results to be returned for similar requests. For now, we will not be changing the options presented here. You can add a new action that uses a compression type of your choice.

For the PeopleSoft deployment, the following settings have been used for HTTP compression –

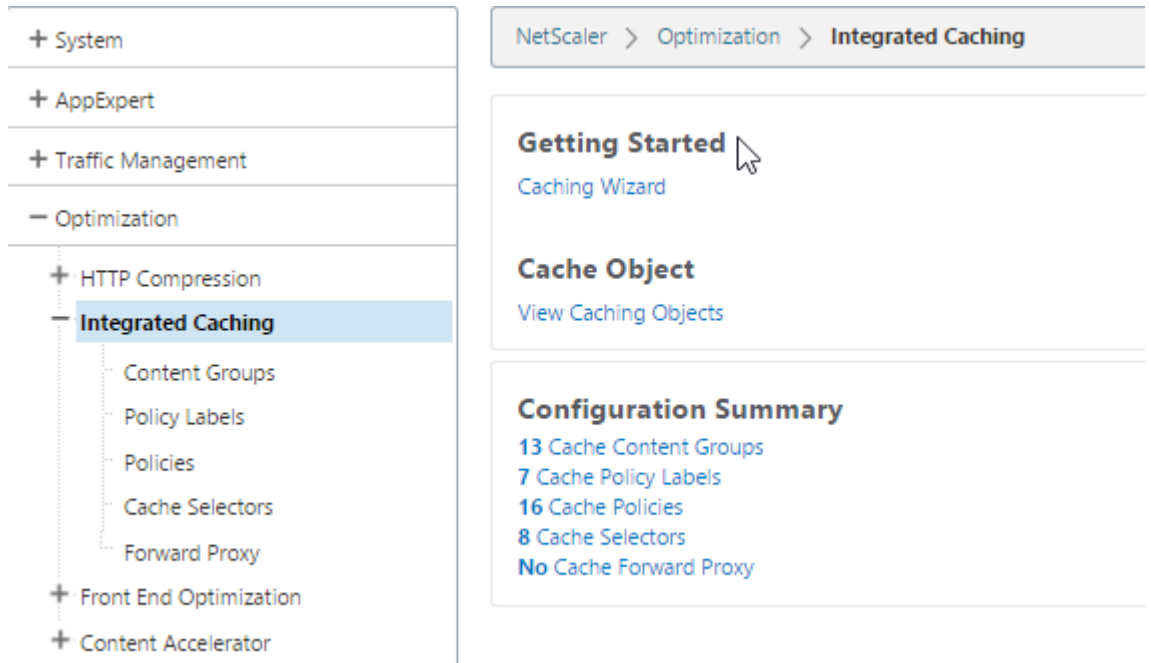
Policy Name: PeopleSoft\_Compression\_Test

Response Action: GZIP (Compress/DEFLATE should work too)

Expression: ns\_true

### Integrated Caching

To configure caching, you can use the integrated wizard that makes configuration very straightforward. To initiate the wizard, navigate to Optimization>Integrated Caching as shown below:



Here, you can initiate the Caching Wizard under Getting Started.

### Specify Content Type

Specify Content Type\*

Static Content ▼

Select Static Content if requests sent to URL always returns the same content.

The following list contains some examples of static content

Style Sheets - /layouts/styles/front.css,  
Scripts - /scripts/helper.js,  
Images - /resources/graphics/thumbnail.gif

Continue

Cancel

The first step requires you to specify the content type. This can be either static (examples given) or dynamic. Helpful hints are provided as shown above to help determine which type of content is relevant for you.

← Back

### Static Content Caching Wizard

#### What to Cache

Policy Name\*

Expression\*

Expression Editor

Operators Saved Policy Expressions Frequently Used Expressions Clear

Press Control+Space to start the expression and then type

Continue Cancel

Select

- HTTP.REQ.URL-Is a Pattern present in HTTP Request URL? [HTTP.REQ.URL.PATH\_AND\_QUERY.CONTAINS("")]
- HTTP.REQ.URL-Compare HTTP URL suffix with a URL [HTTP.REQ.URL.SUFFIX.EQ("")]
- HTTP.REQ.HEADER-Does an HTTP Request Header exists? [HTTP.REQ.HEADER("").EXISTS]
- HTTP.REQ.HEADER-Is a Pattern present in HTTP Request Header? [HTTP.REQ.HEADER("").CONTAINS("")]
- HTTP.REQ.HOSTNAME-Obtain the HTTP Host Name object from this request. [HTTP.REQ.HOSTNAME]
- HTTP.REQ.COOKIE-(Name/Value List) Returns the contents of the HTTP Cookie header as a name/value list. [HTTP.REQ.COOKIE]
- CLIENT.IP.SRC>Returns the source IP of the current packet. [CLIENT.IP.SRC]
- HTTP.RES.HEADER-Is a Pattern present in HTTP Response Header? [HTTP.RES.HEADER("").CONTAINS("")]
- HTTP.RES.CACHE\_CONTROL>Returns the HTTP Cache-Control object [HTTP.RES.CACHE\_CONTROL]
- HTTP.RES.STATUS>Returns the HTTP response status code. [HTTP.RES.STATUS]
- HTTP.REQ.IS\_VALID>Returns TRUE if the HTTP request is properly formed. [HTTP.REQ.IS\_VALID]
- HTTP.REQ.METHOD-Compare HTTP Request Method. [HTTP.REQ.METHOD.EQ(GET)]

The next step involves defining which content should be cached. The Frequently Used Expressions dropdown helps you define the correct expression; however, if you want the caching policy to run for all content, you can use `ns_true` as the expression as shown below:

### Static Content Caching Wizard

#### Cache Policy

Policy Name	Expression
Test	ns_true

#### Specify Content Expiration

☒ Custom ☐ Heuristic

Expire content after

Time Zone

☒ Local ☐ GMT

Absolute Expiry Time

Weak relative expiry for negative (error) responses eg: 4xx 5xx

Continue Cancel



Static Content Caching Wizard

Cache Policy

Policy Name  
Test

Specify Content Expiration

☐ Custom

☒ Heuristic

Weak relative expiry for positive (non-error) responses eg

seconds

Weak relative expiry for negative (error) responses eg: 4xx

seconds

Relative expiry time, in seconds, for expiring positive responses with response codes between 200 and 399. Cannot be used in combination with other Expiry...

More

Continue

Cancel

The next step involves definition of the caching space to be used on the NetScaler and the minimum size of objects to be cached.

Static Content Caching Wizard

Cache Policy

Policy Name  
Test

Content Expiration

Expiry Type	Heuristic
Weak relative expiry for negative (error) responses eg: 4xx 5xx	233
Weak relative expiry for positive (non-error) responses eg: 2xx 3xx	233

Optimize Memory Usage

Quick Abort Size: Continue caching if more than

4194303

KB is already used

Do not cache - if size is less than

0

KB

Do not cache - if size exceeds

80

KB

Do not cache - if hits are less than

0

Finally, the cache policy should be bound to the relevant vserver.

Static Content Caching Wizard

Cache Policy

Policy Name

Test

Content Expiration

Expiry Type

Weak relative expiry for negative (error) responses eg: 4xx 5xx

Weak relative expiry for positive (non-error) responses eg: 2xx 3xx

Heuristic

233

233

Optimize Memory Usage

Quick Abort Size: Continue caching if more than

4194303

Do not cache - if size is less than

0

Do not cache - if size is less than

8000000

Cache Policies

No Load Balancing Virtual Server Request Binding

No Content Switching Virtual Server Request Binding

Continue

These definitions can be made under Cache Policies as shown in the screenshot above.

For the PeopleSoft deployment, the following settings have been used for caching –

Policy Name: PeopleSoft\_Cache\_Test

Actions: CACHE

Cache Content Group: Test

Undefined-Result Action: -Global-undefined-result-action (or NOCACHE/RESET)

Expression: *ns\_true*

Cache Content Group:

Name: Test

Type: HTTP

Expiry Method: Heuristic (Recommended)/Custom (if specific settings are required)

Default Expiry Times: As per requirement; set to 233 for test deployment.

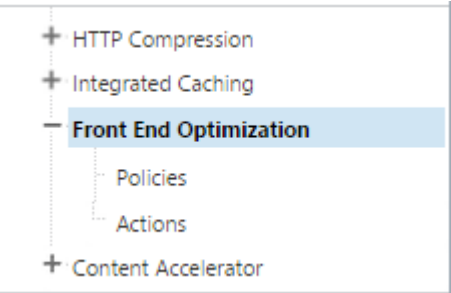
Parameterization: Leave values as is (unless Cache selectors are in use; not configured for our test setup)

Memory: Define values as per your system limits

Others: Use default settings. All settings have context-sensitive help available if modification is required.

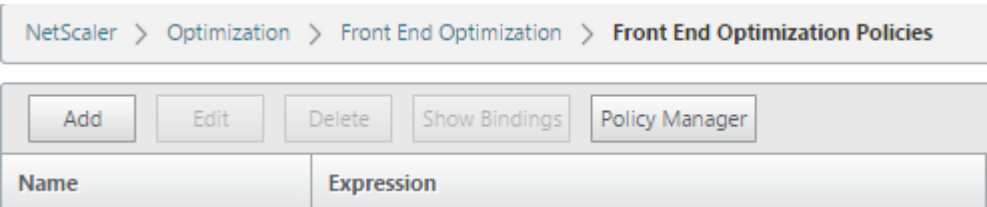
Front End Optimization

The front end optimization feature set makes NetScaler an extremely capable optimization device by implementing enhanced optimization routines for specific front end entities such as images, JavaScript etc. These features provide improved optimization performance than that achieved by compression and caching.



Front End Optimization capabilities can be activated by navigating to Optimization>Front End Optimization. As with all NetScaler features, these are implemented using a policy-action mechanism.

To add a new policy, navigate to Optimization>Front End Optimization and then, click on Policies. To add a new policy, click on Add in the section displayed to the right of the navigation menu.



This will give you the following screen for definition of a new FEO policy.

Create Front End Optimization Policy

Name\*

Action\*

BASIC

+

?

Expression\*

Operators

Saved Policy Expressions

Frequently Used Expressions

Press Control+Space to start the expression and then type '.' to get the next set of options

Create

Close

The Expression here works much on the same lines as expressed for the earlier features; the Frequently Used Expressions drop down can be used for assistance. There are certain predefined actions that can be assigned here, all of which have different configurations for the same settings; you can also either edit or create a custom action, which can be done using the plus or pencil buttons next to the Action name. The Plus icon enables the setup of a custom profile.

Upon clicking either of these buttons, the following screen (or a similar one) is observed:

**Configure Front End Optimization Action**

Name  
MODERATE

**JavaScript**

☒ Make Inline ☐ Move to End of Body Tag  
☒ Minify

**Image**

☒ Shrink to Attributes ☒ Convert GIF to PNG  
☒ Make Inline ☒ Lazy Load  
☒ Optimize ☐ Convert to WEBP  
☐ Convert to JXR format

**CSS**

☒ Make Inline ☐ Combine  
☐ Move to Head Tag ☐ Convert Imports to Links  
☐ Image Inline ☒ Minify

**HTML**

☐ Remove comments from HTML

**Miscellaneous optimization**

☐ Extend Page Cache  
☐ Enable Client Side Measurements

This screen presents all the various front end optimization options available; NetScaler can help to optimize web traffic with JavaScript, Image, CSS (Cascading Style Sheets), HTML and Miscellaneous Optimization. This last section also allows for domain sharding, which splits resources across subdomains to improve optimization and page load times.

For this deployment, the recommended FEO policy setting is Aggressive; this default setting provides a good level of optimization while not affecting the performance of the PeopleSoft setup. The lab tests show an approximate 65 percent reduction in load times, 10-15 percent reduction in the amount of data transferred and 10-15 percent reduction in number of requests on our test setup for generic operations. Results may differ for your setup.

Optimization settings for the Oracle PeopleSoft deployment:

Optimization Policy Name: PeopleSoft\_Optimization\_Test  
Optimization Action: AGGRESSIVE (Preconfigured)  
Expression: HTTP.REQ.HEADER("Accept").CONTAINS("html")

**Alternate Configuration (Custom Policy)**

Optimization Policy Name: PeopleSoft\_Optimization\_TestCustom  
Optimization Action: samplefeo  
Expression: HTTP.REQ.HEADER("Accept").CONTAINS("html")

**PeopleSoft\_Optimization\_TestCustom Configuration:**

Enabled Settings: JavaScript/Make Inline, JavaScript/Move to End of Body Tag, JavaScript/Minify, Image/Optimize, Image/Lazy Load, Image/Shrink to Attributes, Image/Optimize, Image/Convert to JXR format, Image/Convert GIF to PNG, CSS/Make Inline, CSS/Move to Head Tag, CSS/Minify, CSS/Image Inline, CSS/Combine, CSS/Convert Imports to Links, HTML/Remove Comments from HTML

## Conclusion

NetScaler enables highly available Oracle PeopleSoft deployments with its load balancing capabilities. With NetScaler, enterprises can enable a host of additional capabilities including but not limited to authentication offload, end point analysis checks, selective server access, URL rewrites, compression, caching, front end optimizations and much more.

With NetScaler, enterprises can not only enable high availability for their Oracle PeopleSoft environments, but also extend capabilities for security and optimized access. The policy engine used by NetScaler enables enterprises to deploy any specific use cases that they may require, making the NetScaler solution a flexible and robust one that can meet all enterprise requirements.

To learn more about how NetScaler can bring these benefits to Oracle PeopleSoft installations or address other application delivery requirements, please visit <http://www.citrix.com>.



**Enterprise Sales**

North America | 800-424-8749  
Worldwide | +1 408-790-8000

**Locations**

**Corporate Headquarters** | 851 Cypress Creek Road Fort Lauderdale, FL 33309 United States  
**Silicon Valley** | 4988 Great America Parkway Santa Clara, CA 95054 United States

Copyright© Inc. All rights reserved. Citrix, the Citrix logo, and other marks appearing herein are property of Citrix Systems, Inc. and/or one or more of its subsidiaries, and may be registered with the U.S. Patent and Trademark Office and in other countries. All other marks are the property of their respective owner/s.