# Deployit NetScaler Application Delivery Controller Plugin Manual

**Version 3.9.1**

# Table of Contents

Table of Contents 2
Preface 3
Overview 3
Features 3
Requirements 3
Functionality 3
Service Group based 3
Server/Service based 3
Behavior 4
Setting up a loadbalancing configuration 4
Service Group based 4
Server/Service based 4
Loadbalancing a mixed Application Server and Web Server environment 4
Customization 4
Disable/Enable server scripts 4
CI Reference 5
Configuration Item Overview 5
Containers 5
Other Configuration Items 5
Configuration Item Details 5
lb.LoadBalancer 5
netscaler.NetScaler 5
netscaler.NetScalerParameters 6
netscaler.NetScalerScriptParameters 6

Copyright XebiaLabs B.V. © 2008-2012. All rights reserved.

## Preface

This document describes the functionality provided by the Citrix NetScaler Application Delivery Controller (netscaler) plugin.

Refer to **Deployit Reference Manual** for background information on Deployit and deployment concepts.

## Overview

The NetScaler plugin is a Deployit plugin that adds capability for managing deployments done to application and web servers whose traffic is managed by a NetScaler loadbalancing device.

See the **Features** section below for details.

## Features

- Allows servers / services to be taken out of the load balancing pool before a deployment and put back after the deployment has been performed.

## Requirements

- **Deployit requirements**
  - **Deployit**: version 3.9+
  - **NetScaler**: 8.1, 9.x and 10.x
  - **Other Deployit Plugins**: loadbalancer-plugin (Distributed together), group-orchestrator-plugin, webserver-plugin

## Functionality

The plugin supports two modes of working:

1. Service group based
2. Server/Service based

Furthermore the plugin works in conjunction with the "group-based" orchestrator to disable and enable containers which are part of a single deployment group in one go.

The group-based orchestrator will divvy up the deployment into multiple phases, based on the 'deploymentGroup' property of the containers that are being targeted. Each of these group will be disabled in the NetScaler just before they're deployed to, and will be re-enabled right after the deployment to that group. This will ensure that there is no downtime during the deployment.

### Service Group based

The plugin will add the following 4 properties to every deployable and deployed to control which service in which service group this deployed affects.

**netscalerServiceGroup** : STRING
The name of the service group that the service running on the targeted container is registered under. (default: {{NETSCALER_SERVICE_GROUP}}

**netscalerServiceGroupName** : STRING
The name of the service in the service group. (default: {{NETSCALER_SERVICE_GROUP_NAME}})

**netscalerServiceGroupPort** : STRING
The port the service in the service group runs on. Note: This is a string on the deployable to support placeholder replacement. (default: {{NETSCALER_SERVICE_GROUP_PORT}})

### Server/Service based

The plugin will add the following 3 properties to every container to control how the server is known in the NetScaler ADC, and how long it should take to do a graceful disable of the server:

**netscalerAddress** : STRING
The IP address or name this server is registered under in the NetScaler load balancer

Copyright XebiaLabs B.V. © 2008-2012. All rights reserved.

**netscalerType** : STRING
Whether this is a 'server' or a 'service' in the NetScaler load balancer (default: server)

**netscalerShutdownDelay** : INTEGER
The amount of seconds before the server is disabled in the NetScaler load balancer. A value of -1 triggers use of the defaultShutdownDelay of the NetScaler device. (default: -1)

### Behavior

The plugin will add three steps to the deployment of each deployment group.

1. A disable server step, this will stop the traffic to the servers that are managed by the loadbalancer.
2. A wait step, this will wait the maximum shutdown delay
3. An enable server step, this will start the traffic to the servers that were previously disabled.

## Setting up a loadbalancing configuration

In order to setup Deployit to use your NetScaler ADC device, you will need to take the following steps.

1. Create a NetScaler (`netscaler.NetScaler`) configuration item in the 'Infrastructure' tree under a host, and add it as a `member` to the `udm.Environment`. The host configuration item indicates how to connect to the NetScaler device.
2. Add all the containers that the NetScaler device manages to the `managedServers` collection of the created NetScaler configuration item.

### Service Group based

For the service group based setup, you can create dictionaries restricted to containers in the environment. Each of these dictionaries will need to contain the following keys:

- NETSCALER_SERVICE_GROUP
- NETSCALER_SERVICE_GROUP_NAME
- NETSCALER_SERVICE_GROUP_PORT

As a second option, you could immediately do an initial deployment and set the values correctly on all the deployeds. During an upgrade deployment these values will be copied from the previous deployment.

### Server/Service based

1. Configure the `netscalerAddress` property of each of the containers so that the NetScaler configuration item knows how the container is known within the NetScaler ADC device.

Once that is done, during any deployment to the Environment, the *netscaler-plugin* will ensure that the loadbalancing logic is implemented.

### Loadbalancing a mixed Application Server and Web Server environment

If you have an Apache httpd server which fronts a website backed by one or more application servers, it is possible to setup a more complex loadbalancing scenario, thus ensuring that the served website is not broken during the deployment. For this, the `www.ApacheHttpdServer` configuration item from the standard *webserver-plugin* is augmented with a property called `applicationServers`.

Whenever a deployment is done to one or more of the containers mentioned in the `applicationServers` residing in the same Environment as the web server, the following happens in addition to the standard behavior:

1. Just before the first application server will be deployed to, the webserver will be removed from the loadbalancing configuration.
2. After the last application server linked to the web server has been deployed to, the webserver is put back into the loadbalancing configuration

## Customization

### Disable/Enable server scripts

Copyright XebiaLabs B.V. © 2008-2012. All rights reserved.

The disable and enable server scripts, by default called:

- netscaler/disable-server.cli.ftl
- netscaler/enable-server.cli.ftl

contain the NetScaler CLI commands to influence the loadbalancing. They are FreeMarker templates which have access to the following variables during resolution:

- *servers*: A list of `NetScalerItem` (Either `ServiceGroup` or `ServerOrService`) that are to be enabled/disabled.
- *loadBalancer*: The `netscaler.NetScaler` loadbalancer that manages the servers.

## CI Reference

## Configuration Item Overview

### Containers

| CI | Description |
|---|---|
| *lb.LoadBalancer* | The Load Balancer super type |
| netscaler.NetScaler | A Citrix NetScaler LoadBalancer |

### Other Configuration Items

| CI | Description |
|---|---|
| *lb.LoadBalancer* | The Load Balancer super type |
| netscaler.NetScaler | A Citrix NetScaler LoadBalancer |
| netscaler.NetScalerParameters | Default implementation of a CI that provides ID, name and type information |
| netscaler.NetScalerScriptParameters | Default implementation of a CI that provides ID, name and type information |

## Configuration Item Details

### lb.LoadBalancer

**Virtual Type**
**Type Hierarchy** udm.BaseContainer >> udm.BaseConfigurationItem
**Interfaces** udm.Taggable, udm.ConfigurationItem, udm.Container, overthere.HostContainer

The Load Balancer super type.

| Public Properties |
|---|
| **managedServers** : SET_OF_CI<udm.Container><br>The servers managed by this load balancer |
| **Hidden Properties** |
| **disableInLoadBalancer** : BOOLEAN<br>Whether this server should be disabled in the load balancer when it is being deployed to |
| **netscalerAddress** : STRING<br>The IP address or name this server is registered under in the NetScaler load balancer |
| **netscalerShutdownDelay** : INTEGER<br>The amount of seconds before the server is disabled in the NetScaler load balancer (A value of -1 triggers use of the defaultShutdownDelay of the NetScaler device) |
| **netscalerType** : STRING = server<br>Whether this is a 'server' or a 'service' in the NetScaler load balancer |
| **tags** : SET_OF_STRING<br>If set, only deployables with the same tag will be automatically mapped to this container. |

### netscaler.NetScaler

**Type Hierarchy** lb.LoadBalancer >> udm.BaseContainer >> udm.BaseConfigurationItem
**Interfaces** udm.Taggable, udm.ConfigurationItem, udm.Container, overthere.HostContainer

A Citrix NetScaler LoadBalancer

Copyright XebiaLabs B.V. © 2008-2012. All rights reserved.

| Parent | |
|---|---|
| * | **host** : CI<overthere.Host><br>Host |

| Public Properties | |
|---|---|
| * | **defaultShutdownDelay** : INTEGER = 0<br>The amount of seconds to wait before the servers are disabled in the load balancer |
| | **managedServers** : SET_OF_CI<udm.Container><br>The servers managed by this load balancer |

| Hidden Properties | |
|---|---|
| * | **disableServersOrder** : INTEGER = 8<br>Disable Servers Order |
| * | **disableServersScript** : STRING = netscaler/disable-server.cli.ftl<br>Disable Servers Script |
| * | **enableServersOrder** : INTEGER = 94<br>Enable Servers Order |
| * | **enableServersScript** : STRING = netscaler/enable-server.cli.ftl<br>Enable Servers Script |
| * | **runScript** : STRING = netscaler/run-script.cli.ftl<br>Run Script |
| * | **sourceCommand** : STRING = source<br>Source Command |
| | **disableInLoadBalancer** : BOOLEAN<br>Whether this server should be disabled in the load balancer when it is being deployed to |
| | **netscalerAddress** : STRING<br>The IP address or name this server is registered under in the NetScaler load balancer |
| | **netscalerShutdownDelay** : INTEGER<br>The amount of seconds before the server is disabled in the NetScaler load balancer (A value of -1 triggers use of the defaultShutdownDelay of the NetScaler device) |
| | **netscalerType** : STRING = server<br>Whether this is a 'server' or a 'service' in the NetScaler load balancer |
| | **tags** : SET_OF_STRING<br>If set, only deployables with the same tag will be automatically mapped to this container. |

| Control task | Parameter CI | Attributes | Description |
|---|---|---|---|
| **disableService** | netscaler.NetScalerParameters | | no description |
| **enableService** | netscaler.NetScalerParameters | | no description |
| **runScript** | netscaler.NetScalerScriptParameters | | no description |

## netscaler.NetScalerParameters

**Type Hierarchy** udm.Parameters >> udm.BaseConfigurationItem
**Interfaces** udm.ConfigurationItem

Default implementation of a CI that provides ID, name and type information.

| Public Properties | |
|---|---|
| * | **serviceGroup** : STRING<br>NetScaler service group |
| * | **serviceGroupName** : STRING<br>NetScaler service name |
| * | **serviceGroupPort** : INTEGER<br>NetScaler service port |

## netscaler.NetScalerScriptParameters

**Type Hierarchy** udm.Parameters >> udm.BaseConfigurationItem
**Interfaces** udm.ConfigurationItem

Default implementation of a CI that provides ID, name and type information.

| Public Properties | |
|---|---|
| * | **netScalerScript** : STRING<br>Script to execute on the NetScaler device |

Copyright XebiaLabs B.V. © 2008-2012. All rights reserved.

Copyright XebiaLabs B.V. © 2008-2012. All rights reserved.