Admin CLI LB Enhancement
AS9.1EE Test Spec

2.Revision History

Version	Comments	Date	Author
1.0 1.1	Initial Draft for 9.0ee Ready for 9.1ee Review	04/20/05 10/18/06	Davis Nguyen Davis Nguyen

Table of Content

1.Introduction

This document provides a list of procedures to test Admin CLI Loadbalancer enhancement in AppServer 9.1ee release. For detail test case execution and configuration, please see section 2, Table of Content.

3.Existing 8.1EE Commands Test Cases
1.0 create-http-lb-config

1.0 Definition:

The create-http-lb-config command is used to create a lb configuration. If --target option is not specified and config_name operand is specified, then a load balancer configuration is created but with no references to any target. If config_name operand is not specified and --target option is specified, then a load balancer configuration is implicitly created with an unique name as targetName-http-lb-config. Text **color in purple** is referenced for new attributes and properties for the command

1.1 Usage:

```
create-http-lb-config [--responsetimeout 60] [--httpsrouting=false] [--reloadinterval 60] [--
monitor=false] [--routecookie=true] [--property (name=value)[:name=value]*] --target target |
config_name
```

1.2 Test cases: create-http-lb-config

Basic test (no target)

1. create a lb-config with `only config_name option specified. Verify if the lb-config was created with all default assigned values stated in the usage`
2. create a lb-config with both required parameters `target and config_name` options and verify for appropriate error messages are thrown
3. miss both the required parameters `target and config_name` and verify appropriate error messages are thrown

resposetimeout option

4. create a lb-config with responsetimoeout other than 60 and verify its value by using get command
5. create a lb-config with responsetimeout=0 and verify its validation error
6. create a lb-config with responsetimeout as a non integer number and make sure appropriate error messages are thrown

httpsrouting option

5. create a lb-config with httpsrouting=false and verify its value
6. create a lb-config with httpsrouting without any values and verify its validation error
7. create a lb-config with httpsrouting=true and verify its value
8. create a lb-config with httpsrouting other than true/false and verify if it is reported as error

reloadinterval option

9. create a lb-config with reloadinterval=60 and verify its value
10. create a lb-config with reloadinterval=0 and verify its value
11. create a lb-config with reloadinterval other than 60 and verify its value
11b. create a lb-config with reloadinterval=-1 and verify its validation error

routecokkie option

12. create a lb-config with routecookie=true and verify its value
13. create a lb-config with routecookie=false and verify its value
14. create a lb-config with routecookie without any value specified and verify it is set to to true by default

target option and config

15. create a lb-config without a config name and verify if the lb config name is derived from the default server-referenced and it named after the default server - server-http-lb-config
16. create a lb-config without a config name to the cluster target and verify the lb config name is derived from the cluster-referenced and it named by the cluster reference - cluster1-http-lb-config
17. create a lb-config without a config name to the server target and verify the lb config name is derived from the server-referenced and it named by the server reference - server1-http-lb-config
18. create a lb-config with a assigned name to a standalone server.
19. create a lb-config with target as a nonexisting cluster/server, verify it is reported as a error
20. create a lb-config with target as a non standalone instance of a cluster - verify if the command reports an error message
21. create lb-config with target name as configuration name which already exists in the domain and verify it is reported as a error
22. create lb-config with config name and with the same target

property
23. create a lb-config with property test1="This is Test" and verity if property created correctly
24. create a lb-config property test2=123:value3="Second Test"

2.0 delete-http-lb-config

2.0 Definition:

The delete-http-lb-config command deletes a load balancer configuration. Currently, the load balancer must not reference any clusters or instances before it can be deleted.

2.1 Usage:

```
delete-http-lb-config config_name
```

1.2 Test cases: delete-http-lb-config

1. delete the lb-config which is referenced to a cluster and verify it is reported an error
2. delete the lb-config which is referenced by a standalone server and verify it is reported an error
3. delete the existing unreferenced lb-config (having no server-ref) and verify it is deleted

4. delete the existing unreferenced lb-config (having no cluster-ref) and verify it is deleted
5. delete a non existent lb-config and verify it is reported as a error
6. delete an existent lb-config with a wrong target name and verify it is reported as a error

3.0 list-http-lb-configs
3.0 Definition:

The list-http-lb-configs command lists the load balancer config and its clusters/instances, or all the load balancer config in the domain.

3.1 Usage:
```
list-http-lb-configs [target]
```
3.2 Test cases: list-http-lb-configs
1. list the lb-config associated with this particular clusters by giving cluster name for the target option
2. list the lb-config associated with this particular servers by giving server name for the target option
3. try to list the lb-config associated with a particular cluster/server by giving cluster/server name for the target option and there is no instance/cluster is referencing the lb-config. Verify the list command does not list any cluster/instance
4. list the lb-configs in a domain by omitting the target name and verify it lists all the lb config in the domain

4.0 create-http-lb-ref
4.0 Definition:

The create-http-lb-ref command is used to add an existing cluster to an existing load balancer configuration. Text **color in purple** is referenced for new atributes and properties for the command.

4.1 Usage:
```
create-http-lb-ref [--lbpolicy lb_policy] [--lbpolicymodule lb_policy_module] [--healthcheckerurl
url] [--healthcheckerinterval=30] [--healthcheckertimeout=10] [--lbEnableAllInstances] [--
lbEnableAllApplications] --config config_name | --lbname loadbalancer_name target
```
4.2 Test cases: create-http-lb-ref
1. Add server-ref to an existing server lb-config without specifying any optional attributes and verify if the server-ref is created sucessfully with default attribute values
2. Add cluster-ref to an existing cluster lb-config without specifying any optional attributes and verify if the cluster-ref is created sucessfully with default attribute values
3. Add a cluster-ref twice with the same lb-config name and verify for the error message
   4. Add a cluster-ref with --lbpolicy=weighted-round-robin and verify for lbpolicy values and other attribute values as well
   5. Add a cluster-ref with --lbpolicy=round-robin and verify for lbpolicy values and other attribute values as well
   6. Add a cluster-ref with --lbpolicy=user-defined & specified --lbpolicymodule=....\my-lbLib.so and verify for lbpolicy values and other attribute values as well
   7. Add a cluster-ref with --healthcheckerurl=/ and verify for healthcheckerurl values and other attribute values as well
   8. Add a cluster-ref with --healthcheckerinterval=99 and verify for healthcheckerinterval values and other attribute values as well
   9. Add cluster-ref with --healthcheckertimeout=100 and verify for healthcheckertimeout values and other attribute values as well
   10. Add a cluster-ref with --lbEnableAllInstances=true and verify for lbEnableAllInstances values and other attribute values as well
   11. Add a cluster-ref with --lbEnableAllApplications=true and verify for lbEnableAllApplications values and other attribute values as well
   13. Add a cluster-ref with both --config and --lbname options are specified and verify for error message thown.
   14. Add a server-ref with --lbname option to a targer server and verify if lb-config and lbname are created for the server target
   14. Add a cluster-ref with --lbname option to a cluster target and verify if lb-config and lbname are created for the cluster target

5.0 delete-http-lb-ref
5.0 Definition:

The delete-http-lb-ref command is used to delete a cluster/instance reference from a load balancer config. Text **color in purple** is referenced for new atributes and properties for the command.

5.1 Usage:
```
delete-http-lb-ref [--force=true] --config config_name | --lbname loadbalancer_name target
```
5.2 Test cases: create-http-lb-ref
1. delete the server-ref with having all applications disable and verify it is deleted
2. delete the cluster-ref and with having all applications and servers disable and verify it is deleted
3. try delete a nonexisting server-ref and verify appropriate error is thrown

4. try delete a nonexisting cluster-ref and verify appropriate error is thrown

   5. try to delete a cluster-ref with having all applications and server enabled, and verify appropriate error is thrown (force=false)
    6. try to delete a cluster-ref with having all applications enabled and servers disabled, and verify appropriate error is thrown (force=false)
    7. try to delete a cluster-ref with having all servers enabled and application disabled, and verify appropriate error is thrown (force=false)
    8. try to delete a cluster-ref with having force=true and all applications disabled, and verify it is deleted
    9. try to delete a cluster-ref with having force=true and all servers disabled, and verify it is deleted
   10. try to delete a cluster-ref with having force=true and all applications and server disabled, and verify it is deleted
   11. try to delete a server-ref with force=true, config and lbname options and verify if error message is thrown
   12. try to delete a cluster-ref with force=true and lbname options and verify if lb-config and loadbalancer-name all deleted
   13. try to delete a cluster-ref with having all applications and servers disabled with force=false and lbname options. Verify if only loadbalancer-name is deleted and an error message should appears for trying to delete the lb-config
   14. try to delete a server-ref with having all applications disabled with force=false and lbname options. Verify if only loadbalancer-name is deleted and an error message appears for trying to delete the lb-config

6.0 disable-http-lb-server
6.0 Definition:
The disable-http-lb-server command is used to disable a cluster/server managed by the load balancer.
6.1 Usage:
```
disable-http-lb-server [--timeout 30] target
```
6.2 Test cases: disable-http-lb-server
1. disable the server in the specified target and specific time. Verify if the server is actually disable
2. disable the cluster in the specified cluster name and verify it is disabled
3. try to disable a server in the specified server name which has lb-enabled in a false state and verify if a warning is thrown
4. disable the server in the specified server name with timeout option value is different than 30 and verify the value is set

7.0 enable-http-lb-server
7.0 Definition:
The enable-http-lb-server command is used to re-enable a previously disabled load balancer server instance
7.1 Usage:
```
enable-http-lb-server target
```
7.2 Test cases: enable-http-lb-server
1. enable an instance server and verify it is enabled
2. enable an instance which is in already enabled state and verify appropriate INFO message is thrown
3. enable a cluster and verify it is enabled

8.0 disable-http-lb-application
8.0 Definition:
The disable-http-lb-application command is used to disable a specific application managed by the load balancer.
8.1 Usage:
```
disable-http-lb-application [--timeout 30] --name application_name target
```
8.2 Test cases: disable-http-lb-application
1. disable the application for a server-config with the required parameters and verify it is disabled in 30 minnutes
2. disable the application for a server-config with missing required parameters one by one and verify f appropriate error is thrown (invalidate test)
3. disable the application for a server-config with timeout value other than 30 and verify it is set
4. disable the application for a server-config with timeout value as 0 and verify it is set
5. disable the application for a server-config which is not existing and verify appropriate message is thrown
6. disable the application for a cluster-config with all the parameters and verify it is disabled
7. disable the application for a cluster-config with missing required parameters one by one and verify appropriate error is thrown
8. disable the application for a cluster-config with timeout value other than 30 and verify it is set
9. disable the application for a cluster-config with timeout value as 0 and verify it is set
10. disable the application for a cluster-config which is not existing and verify appropriate message is thrown

9.0 enable-http-lb-application

9.0 Definition:

The enable-http-lb-application command is used to re-enable a previously disabled load balancer application.

9.1 Usage:

```
enable-http-lb-application --name application_name target
```

9.2 Test cases: enable-http-lb-application

1. enable an application for a server-ref with all the parameters and verify it is enabled
2. enable an application for a server-ref with missing required parameters one by one and verify error messages are thrown (invalidate test)
3. enable an application for a server-ref with application name which is already in the enabled state and verify for an appropriate error message
4. enable a non-existing application for a server-ref and verify error is thrown
5. enable an application for a cluster-ref with all the parameters and verify it is enabled for all instances in that cluster
6. enable an application for a cluster-ref with missing required parameters one by one and verify error messages are thrown
7. enable an application for a cluster-ref with application name which is already in the enabled state and verify for an appropriate error message
8. enable a non-existing application for a cluster-ref and verify error is thrown

10.0 create-http-health-checker

10.0 Definition:

The create-http-health-checker command is to create a health checker to a load balancer (LB) configuration. The command will require a load balancer reference configuration

10.1 Usage:

```
create-http-health-checker [--url "/"] [--config config_name] [--interval 30] [--timeout 10] target
```

10.2 Test cases: create-http-health-checker

1. create a health checker with all the parameters and verify it is created with the correct default values
2. create health checker with all the parameters and with custom values and verify it is created
3. create health checker with missing required parameters one by one and verify appropriate error messages are thrown
4. create a health checker with " " as url and verify it is reported as a error
5. create a health checker with url which doesnot exist in the specified target and verify it is get created
6. create a health checker with url having HTTP GET parameters and verify it is created (/test/getApp?name=value)
7. create a health checker with interval as 0 and verify it is reported as error
8. create a health checker with interval as a negative number and verify it is reported as a error
9. create a health checker with interval as some custom numbers
10. create a health checker with timeout as 0 and verify it is reported as error
11. create a health checker with timeout as a -ve number and verify it is reported as error
12. create a health checker with non existing config and verify it is reported as a error
13. create a health checker with existing config which already has health checked created and it should be modified with new configuration
14. create a health checker with cluster name as the config name and verify health checker created.

11.0 delete-http-health-checker

11.0 Definition:

The delete-http-health-checker command is to delete a health checker to a load balancer configuration

11.1 Usage:

```
delete-http-health-checker --config config_name target
```

11.2 Test cases: delete-http-health-checke

1. delete the health checker for the specified config
2. delete the health checker for a non existing lb config and verify error is thrown
3. try to delete the health checker with config name as name of the cluster and verify it is deleted
4. try to delete the health checker twice and verify it is reported as a error (what will be health checker when a health checker is deleted ?. When there is no health checker in a config how does a lb will check the instance state ?.) - optional test if time permitted

12.0 export-http-lb-config

12.0 Definition:

The export-http-lb-config command is used to explicitly create the loadbalancer.xml file consumed by the load balancer

plugins. Text `color in purple` is referenced for new atributes and properties for the command.
12.1 Usage:
```
export-http-lb-config --config config_name | --lbname loadbalanacer_name [--retrieve] [file_name]
```
12.2 Test cases: export-http-lb-config
    1. export a lb-config with the specified config and a file_name option to the specified absolute path; verify the result

    2. export a lb-config with the specified config option and verify if the loadbalancer.xml.LB-CONFIG_NAME file is export to the <appserver-install>domains/domain1/generated dirrectory

    3. export the config which is not existing and verify error is thrown

    4. export the config to a file location for which there is no write permission and verify error is thrown

    5. export the config to a file which is already existing and verify it is overwritten (BUG)

    6. try file name is a directory and verify loadbalancaer.xml is created in the current directory

    7 try to export a lb-config with having --retrieve=true option specified and verify if target_loadblanacer.xml file is exported to the current machine ../domains/domain1/generated directory

    8 try to export a lb-config from a client agent with having --retrieve=true option specified and file_name=./test-loadbalancer.xml and verify if the test-loadbalancer.xml file is exported to the current directory

    9 try to export a lb-config from a client agent with having --retrieve=true option specified and verify if target_LB_CONFIG_loadbalancer.xml file is exported to the client specified machine location

    10. try to export a cluster lb-config with lbname option specified and verify if loadbalancer.xml.LB-CONFIG_name file is created correctly

    11. try to export a server lb-config with lbname option specified and verify if loadbalancer.xml.LB-CONFIG_name file is created correctly

    12. try to export a server lb-config with lbname and config options specified, verify the error message thrown

4.New 9.1EE Commands Test Cases

13.0 create-http-lb
13.0 Definition:
    command to create the loadbalancer in the appserver config. Text `color in purple` is the latest added attributes of a new command
13.1 Usage:
    asadmin create-http-lb --devicehost device_host_or_ip --deviceport device_port [--autoapplyenabled=false] [--sslproxyhost proxy_host] [--sslproxyport proxy_port] **[--target]**
 [--lbpolicy lb_policy] [--lbpolicymodule lb_policy_module]

[--healthcheckerurl url] [--healthcheckerinterval 10] [--healthcheckertimeout 10] [--lbEnableAllInstances] [--lbEnableAllApplications] [--isDeviceSSLEnabled] [--property (name=value)[:name=value]*] <loadbalancer-name>
13.2 Test cases: create-http-lb
  Basic test

    1. create a loadbalancer_config with only required attributes options and verify for default values (**loadbalancer-name are created for the default server target)**

    2. createa loadbalancer_config with all parameters and verify for the assigned parameter values (**lb-config and loadbalancer-config are created with all applications, servers and autoApply enabled for the specified target)**

    3. miss the required parameters one by one and verify appropriate error messages are printed

  autoapplyenabled option

    4. create a loadbalancer_config with autoapplyenabled equals to true and verify the set value can be retrieved by get command

    5. create a loadbalancer_config with autoapplyenabled value not true or false and verify this is reported as a error

    6. create a loadbalancer_config with autoapplyenabled value not set and verify if the default value is false

  devicehost option

    8. create a loadbalancer_config without devicehost set and verify appropriate error messages are printed

    9. create a loadbalancer_config with devicehost set to non-existing system and verify appropriate error messages are printed

    10. create a loadbalancer_config with devicehost set an ip address and verify if the value was received correctly

    11. create a loadbalancer_config with devicehost set to a valid hostdevice and verify if the value was received correctly

  devicehost port

    10. create a loadbalancer_config without deviceport set and verify appropriate error messages are printed

    11. create a loadbalancer_config with incorrect deviceport set and verify if the value was received correctly

    12. create a loadbalancer_config with a valid deviceport and devicehost and verify if the value was received correctly

  sslproxyhost option

    13. create a loadbalancer_config with sslproxyhost set to non-existing user and verify appropriate error messages are

printed
   14. create a loadbalancer_config with sslproxyhost set to a proxy_host and verify if the command is executed successfully
   sslproxyport option
   15. create a loadbalancer_config with sslproxyport set to non-existing user and verify appropriate error messages are
printed
   16. create a loadbalancer_config with sslproxyport set to a proxy_port and verify if the command is executed successfully
   property option
   17. create a loadbalancer_config with property set to some values and verify if the property values are set correctly
   lbpolicy & lbpolicymodule
   18. create a loadbalancer_config with
lbpolicy=weighted-round-robin and verify result

19. create a loadblanncer_cofig with lbpolicy=user-defined and lbpolicymodule=../define.so, verify result

healthcheckerurl

20. create a loadbalancer_config with healthcheckerurl=/test and verify result

healthcheckerinterval

21. create a loadbalancer_config with healthcheckerinterval=30 and verify result

healthcheckertimeout

22. create a loadbalancer_config with healthcheckertimeout=15 and verify result

lbEnableAllInstances
   23. create a loadbalancer_config with lbEnableAllInstances=true and verify result

24. create a loadbalancer_config with lbEnableAllInstances=false and verify result

lbEnableAllApplications
   25. create a loadbalancer_config with lbEnableAllApplications=true and verify result

26. create a loadbalancer_config with lbEnableAllApplications=false and verify result

isDeviceSSLEnabled
   26. create a loadbalancer_config with isDeviceSSLEnabled=true and verify result

27. create a loadbalancer_config with isDeviceSSLEnabled=false and verify result

14.0 delete-http-lb
14.0 Definition:
   command to delete the loadbalancer element. Text **color in purple** is the latest added attributes of a new command
14.1 Usage:
   `delete-http-lb  <loadbalancer-name>`
14.2 Test cases: delete-http-lb
   1. delete a loadbalancer-config with the specified load-balancer-name
   2. delete a loadbalancer-config without a load balancer name and verify error is thrown
   3. delete a loadbalancer-config with an incorrect load balancer name and verify error is thrown

15.0 list-http-lbs
15.0 Definition:
   command to list the loadbalancers
15.1 Usage:
   `list-http-lbs`
15.2 Test cases: list-http-lbs
   1. Create a loadbalancer--name and run list-http-lb command to verify if the name is displayed correctly
   2. Create few load-balancer--name and run list-http-lb and verify if all names are displayed correctly

16.0 apply-http-lb-change
16.0 Definition:
   Publishes the changes to load balancer (loadbalancer.xml file) in WebServer
16.1 Usage:
   `apply-http-lb-change <lb-name>`

16.2 Test cases: apply-http-lb-change

    1. Use create-http-lb-config to create a lb_config for a cluster (eg, named lb_config_cluster1)

    2. Use command create-http-b to create a loadblancer_config for cluster1 (eg, named loadbalancer_config_cluster1)

    3. Execute command apply-http-change and verify if loadbalancer.xml file is created in Web Server configuration directory

    4. Repead step 1 & 2 but reset `--lbEnableAllInstances` and `--lbEnableAllApplications=true`, execute apply-http-change and verify the content of loadbalancer.xml file

    5. Modify properties element (lb-configs & loadbalancer) in domain.xml files, `execute` apply-http-change and verify the content of loadbalancer.xml file with the changes

17.0 config-lb-weight

17.1 Usage:

    configure-lb-weight --cluster cluster_name <instance-name=weight[:instance-name=weight]>

17.2 Test cases: config-lb-weight

    1. Create lb_config and loadbalancer_config with having `--lbpolicy=weighted-round-robin`, `execute command` configure-lb-weight to assign weighted value to each instance. Open domain.xml file and verify the result for weighted assigment

    2. Use export-http-lb-config command to create loadbalancer.xml file and verify if those weighted values exported correctly to LB xml file

5.End-to-End Scenario

Scenario A: (existing & new commands)

```
   1. Create a cluster (cluster1)
   2. Create 2 instances in cluster1 (instance1 and instance2)
   3. Deploy jaxrpc-simplebean sample to cluster1
   4. Create a lb-config to cluster1 (http-lb-config-cluster1)
   5. Create a cluster-ref to cluster1 with --lbpolicy=weighted-round-robin, --healthcheckerurl=/, --
lbEnableAllInstances & --lbEnableAllApplications=true options specified.
   6. Create a health-checker for cluster1
   7. Use export-http-lb-config command to export the recently created cluster1 lb-config to
loadbalancer.xml file in Webserver (LB server) config directory.
   8. Create a loadbalancer-config for cluster1 lb-config
   9. Use config-lb-weight comand to config weighted-round-robin for instance1 and instance2 as ratio
(2/5)
   10. Use apply-http-lb-change command to verify the connection.
   11. Launch a client browser verify if you can access jaxrpc-simplebean application through
Loadbalancer port.
   12. Repeat step 11 with simultaneously access from different browsers, view log files for the results.
Access & server log files should show equally level of weighted-round-robin values which was assigned in
step 9.
```

Scenario B: (new commands with a cluster in 2 machines)

```
   1. Create a cluster (cluster1) with 2 remoted instances (instance1 on machine1 and instance on
machine2)
   2. Deploy a test application to cluster1
   3. Use the new create-http-lb command to create lb-config and loadbalancer-config to cluster1
   4. Use apply-http-lb-change command to verify the connection
```
    5. Use `create-http-lb-ref` and configure-lb-weight commands to set value for `the weighted-round-robin` `(100/300)`
```
   6. Use apply-http-lb-change command to update the changes to loadbalancer.xml file and verify the
result.
   7. Launch a browser to access to jaxrpc-simplebean application through Loadbalancer port.
   8. Repeat step 7 with simultaneously access from different browsers, view log files for the results.
Access & server log files should show equally level of weighted-round-robin values which was assigned in
step 5.
```

Scenario C: (auto-apply feature for mutiple clusters)

```
   1. Create a cluster (cluster1)
   2. Create 2 instances in cluster1 (instance1 and instance2)
   3. Deploy a test application to cluster1
   4. Create a loadbalancer_config for cluster1 with autoapplyenabled=true
   5. Use apply-http-lb-change command to verify the connection and export the loadbalancer.xml file
   6. Create cluster2 with 3 instances (instance1, instance2 & instance3)
   7. Deploy a test application to cluster2
   8. Create a loadbalancer_config for cluster2 with autoapplyenabled=true
   9. Use apply-http-lb-change command to verify the connection and update the loadbalancer.xml file for
cluster2
```

10. Launch a browser to access cluster2 application and verify the result
11. Undeploy the application from cluster2 - expecting a warning message?
12. Verify if the loadbalancer.xml and domain.xml file were updated accordingly

6.Review Section

Name	Date	Comments
Satish Viswanatham		
Prashanth Abbagani		
Jane Young		
Nazrul Islam		
Vivekanandh Sedhumadhavan		

7.Reference Documents

Name	Location	Authors
9.x CLI Command Lines	http://appserver.sfbay/apollo/cli/sjsas9_cli_commands.html	CLI Developers
9.1ee LB CLI New Commands	http://www.glassfishwiki.org/gfwiki/Wiki.jsp?page=LBCommands	Jane Young
Loadbalancer User Experience	http://appserver.sfbay.sun.com/twiki/bin/view/Apollo/LoadbalancerUserExperience	Nazrul Islam, Sreeram Duvur
Http Load Balancer Enhancements	http://appserver.sfbay.sun.com/as9ee/eng/http_loadbalancer/lb-onepager.txt	Satish, Nazrul & Harsha
Updating and Monitoring HTTP Load Balancer Blog	http://blogs.sun.com/sv96363/entry/ssl_setup_of_http_load	Satish Viswanatham