

BIGdiff - A Little Help For Software Upgrades



Chase Abbott, 2018-24-08

Published on behalf of DevCentral MVP [Leonardo Souza](#)

If you have been to F5 Agility in Boston and went to my presentation, you should have already an idea of what I will talk about in this article, but you will learn more things, so continue reading.

If you haven't heard of BIGdiff yet, have you been living in Mars?

Don't worry I will explain what that is and how it can help you with software upgrades, and whatever you find useful.

It is not an AI that will do the upgrade for you but will help you with the upgrade.

Challenges

These are the challenges BIGdiff addresses:

You are upgrading a F5 device with 1,000 virtual servers and 1,000 wide IPs.

1. How do you know if you have the same number of virtual servers and wide IPs after the upgrade?
2. How do you know if you have the same number of available virtual servers and wide IPs after the upgrade?
3. If the number of available virtual servers or wide IPs changed after upgrade, how can you find what changed?

Existing Solutions

First Challenge:

There are multiple solutions already for this challenge.

Both for LTM and GTM, you can take a print screen of the statistics before the upgrade and compare after the upgrade.

For LTM, Statistics > Module Statistics > Local Traffic

For GTM, Statistics > Module Statistics > DNS > GSLB

This in 13.1.0.1, but I think this exist since v9, and will be in similar place in all versions.

Qkview and iHealth combination.

iHealth will show you configuration totals but is mainly LTM and does not show you GTM objects.

Network Map is another option.

However, network map is only for LTM.

Also, that is a map that start from a virtual server, so if you have a pool that is not linked to a virtual server that will not count in the totals.

Second Challenge:

The statistics also tell you the status of the objects, so that solution works for both challenges.

Third Challenge:

There is no automated way to get this.

You could run multiple tmsh commands to get the status before the upgrade, or just generate a qkview that will run those commands for you.

However, you will still need to compare the objects one by one.

If the only slot you got for the software upgrade was 3am in a Sunday, I am sure you will miss some objects or fall asleep.

Solution

I hope you are thinking about the same, computers don't need to sleep, and they are better/faster than humans to compare 2 strings or numbers (that is basically 0 or 1, so they are not that smart).

So, the conclusion is simple, let the computer do the work comparing objects while you drink another coffee to keep you awake to complete the software upgrade.

The idea is simple, get the list of objects, and their respective status, before and after the software upgrade, then compare them and report the result.

In this context, object is any entity that has a status in a BIG-IP device that may be affected by the software upgrade.

Looking BIG-IP modules, that translate to LTM and GTM objects, for example, virtual servers and wide IPs.

That is where BIGdiff script comes to help you and automate that process.

You run BIGdiff before the upgrade, upgrade the device, and run again after the upgrade.

The script will then generate a HTML file with the results.

Technical Bits

BIGdiff is a bash script and uses dialog program to generate the graphical menus.

Dialog is a common program for CLI menus and is what F5 uses for the config command for example.

The script uses snmpwalk to query locally the device for the object status, because so far has been the faster option I tested.

That basically generates the same text file before and after the upgrade.

Those text files that are used after to compare the objects.

The script will generate the results in a HTML format, with tables.

If something already exists and do the job well, there is no reason why not to use, so the script uses the [TableFilter](#) JavaScript library, that provide filter functionalities for HTML tables.

You just need to have the JavaScript library file in the same folder that you have the HTML file, and the magic will happen.

If you don't need the filter functionality, you don't need to have the JavaScript library, and static tables will be presented.

The script is optimized to use mainly bash functionality, to be as faster as possible.

I tested the script to compare 13 thousand objects, and it complete the task in a couple minutes.

13K objects is a really big configuration, so even if the device you plan to run the script has a large configuration, that will be just a couple minutes in your change window to run the script.

Support

The script only supports BIG-IP software, no support for EM or BIG-IQ.

The reason is simple, there is no use case for those software.

Versions 11.x.x/12.x.x/13.x.x/14.x.x were tested and are supported.

As new versions are released, I will be testing to see if any change is needed to support that version.

LTM objects are supported and will be listed even if LTM is not provisioned, as majority of the other modules do use LTM internally.

GTM objects and partitions are also supported.

Using BIGdiff

Go to the code share link: [BIGdiff](#)

Download the tablefilter.js file, if you want to use the table filter functionality as described above.

Download the bigdiff.sh that is the script file.

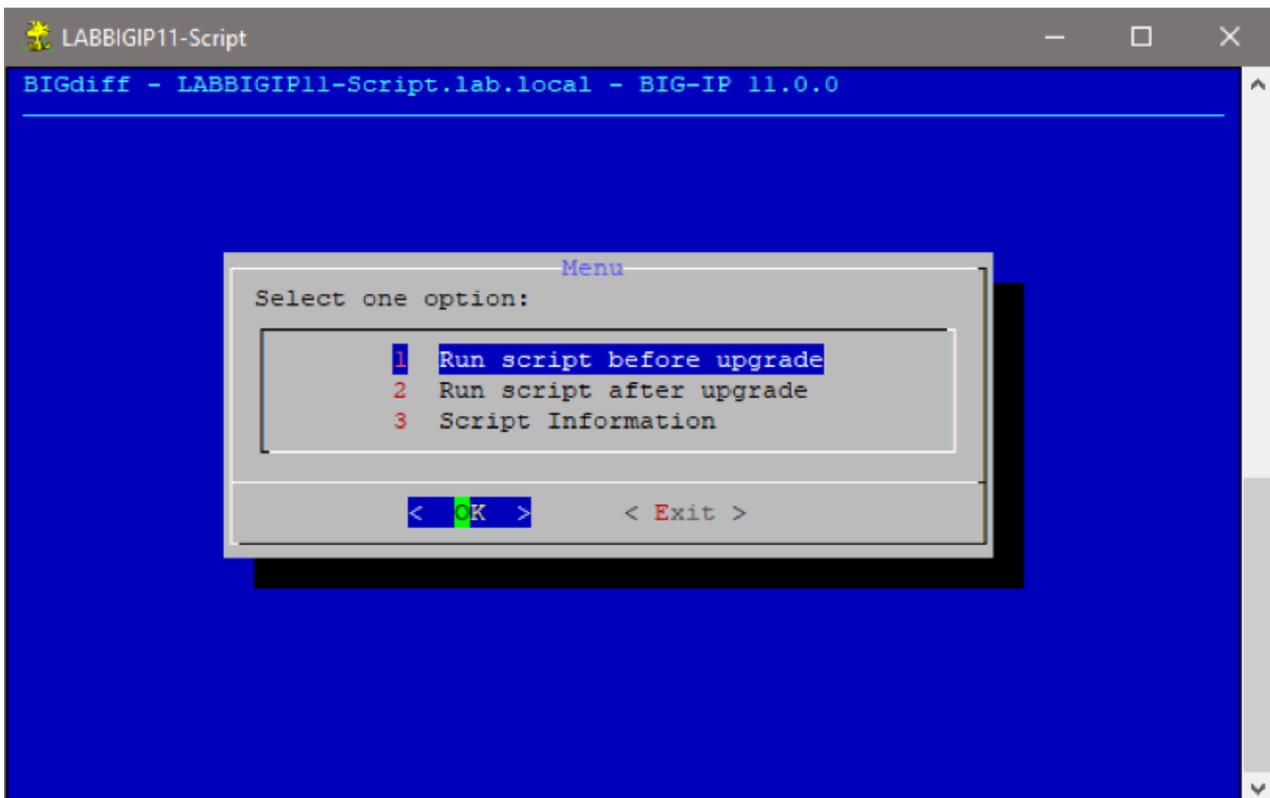
In the F5 device, create a folder in `/shared/tmp`, as `/shared` is shared between all volumes.
Upload the file `bigdiff.sh` to the F5 device.

Change the file permission to run:

```
chmod +x bigdiff.sh
```

Run the script:

```
./bigdiff.sh
```



Run the script before the upgrade.
Upgrade the F5 device.
Run the script after the upgrade.

Download the file ending in `.html` from the F5 device.
Open the HTML file with your favourite browser.
Make sure you have the `tablefilter.js` in the same folder as the HTML file, if you want the filter functionalities.

BIGdiff

2018/08/16 19:51

Upgrade Status

Changes found

Number of Changes

8

LTM Total

Object	Before Upgrade	After Upgrade
Virtual Server	3	3
Pool	3	3
Pool Member	4	4
Node	2	2

GTM Total

Object	Before Upgrade	After Upgrade
Data Center	1	1
Server	1	1
Wide IP	3	3
Pool	3	3
Pool Member	4	4
Virtual Server	3	3
Prober Pool	0	0
Prober Pool Member	0	0
Link	0	0

LTM - Virtual Server

	Clear	Clear	Clear	Clear	
Virtual Server	Before Upgrade - Enabled	After Upgrade - Enabled	Before Upgrade - Status	After Upgrade - Status	Result
partitiona/vsa1	enabled	enabled	green	green	Same
partitiona/vsa2	enabled	enabled	green	green	Same
partitionb/vsb1	enabled	enabled	green	red	Changed

LTM - Pool

	Clear	Clear	Clear	Clear	
Pool	Before Upgrade - Enabled	After Upgrade - Enabled	Before Upgrade - Status	After Upgrade - Status	Result
partitiona/poola1	enabled	enabled	green	green	Same
partitiona/poola2	enabled	enabled	green	green	Same
partitionb/poolb1	enabled	enabled	green	red	Changed

LTM - Pool Member

	Clear	Clear	Clear	Clear			
Pool	Node	Port	Before Upgrade - Enabled	After Upgrade - Enabled	Before Upgrade - Status	After Upgrade - Status	Result
partitiona/poola1	partitiona/172.16.255.254	80	enabled	enabled	green	green	Same
partitiona/poola2	partitiona/172.16.255.254	80	enabled	enabled	green	green	Same
partitiona/poola2	partitiona/172.16.255.254	84	enabled	enabled	green	red	Changed
partitionb/poolb1	partitionb/172.16.255.253	80	enabled	enabled	green	red	Changed

LTM - Node

	Clear	Clear	Clear	Clear	
Node	Before Upgrade - Enabled	After Upgrade - Enabled	Before Upgrade - Status	After Upgrade - Status	Result
partitiona/172.16.255.254	enabled	enabled	blue	blue	Same
partitionb/172.16.255.253	enabled	enabled	blue	blue	Same

GTM - Wide IP

	Clear	Clear	Clear	Clear	
Wide IP	Before Upgrade - Enabled	After Upgrade - Enabled	Before Upgrade - Status	After Upgrade - Status	Result
Common/wideip1.lab.local	enabled	enabled	green	green	Same
Common/wideip2.lab.local	enabled	enabled	green	green	Same
Common/wideip3.lab.local	enabled	enabled	green	red	Changed

GTM - Pool

	Clear	Clear	Clear	Clear	
Pool	Before Upgrade - Enabled	After Upgrade - Enabled	Before Upgrade - Status	After Upgrade - Status	Result
Common/pool1	enabled	enabled	green	green	Same
Common/pool2	enabled	enabled	green	green	Same
Common/pool3	enabled	enabled	green	red	Changed

GTM - Pool Member

	Clear	Clear	Clear	Clear			
Pool	Server	Virtual Server	Before Upgrade - Enabled	After Upgrade - Enabled	Before Upgrade - Status	After Upgrade - Status	Result
Common/pool1	Common.LABBIGIP12-Script.lab.local	partitiona/vsa1	enabled	enabled	green	green	Same
Common/pool2	Common.LABBIGIP12-Script.lab.local	partitiona/vsa1	enabled	enabled	green	green	Same
Common/pool2	Common.LABBIGIP12-Script.lab.local	partitiona/vsa2	enabled	enabled	green	green	Same
Common/pool3	Common.LABBIGIP12-Script.lab.local	partitionb/vsb1	enabled	enabled	green	red	Changed

GTM - Virtual Server

	Clear	Clear	Clear	Clear		
Server	Virtual Server	Before Upgrade - Enabled	After Upgrade - Enabled	Before Upgrade - Status	After Upgrade - Status	Result
Common.LABBIGIP12-Script.lab.local	partitiona/vsa1	enabled	enabled	green	green	Same
Common.LABBIGIP12-Script.lab.local	partitiona/vsa2	enabled	enabled	green	green	Same
Common.LABBIGIP12-Script.lab.local	partitionb/vsb1	enabled	enabled	green	red	Changed

Other Use Cases

The reason I wrote the script was to help with the software upgrades, but you are not limited to software upgrades. You can use the script to compare the objects after you have done something, that can be an upgrade or something else.

You can use the software for consolidations, for example 2 devices that will be replaced by a single device.

You run the script in the old devices, merge the txt files that are created with the list of objects.

Import the configuration in the new device, upload the script and merged txt file you created.

Run the script in the new device, and the script will report to you if the objects have the same status as in the old devices.

Another use case is for major changes.

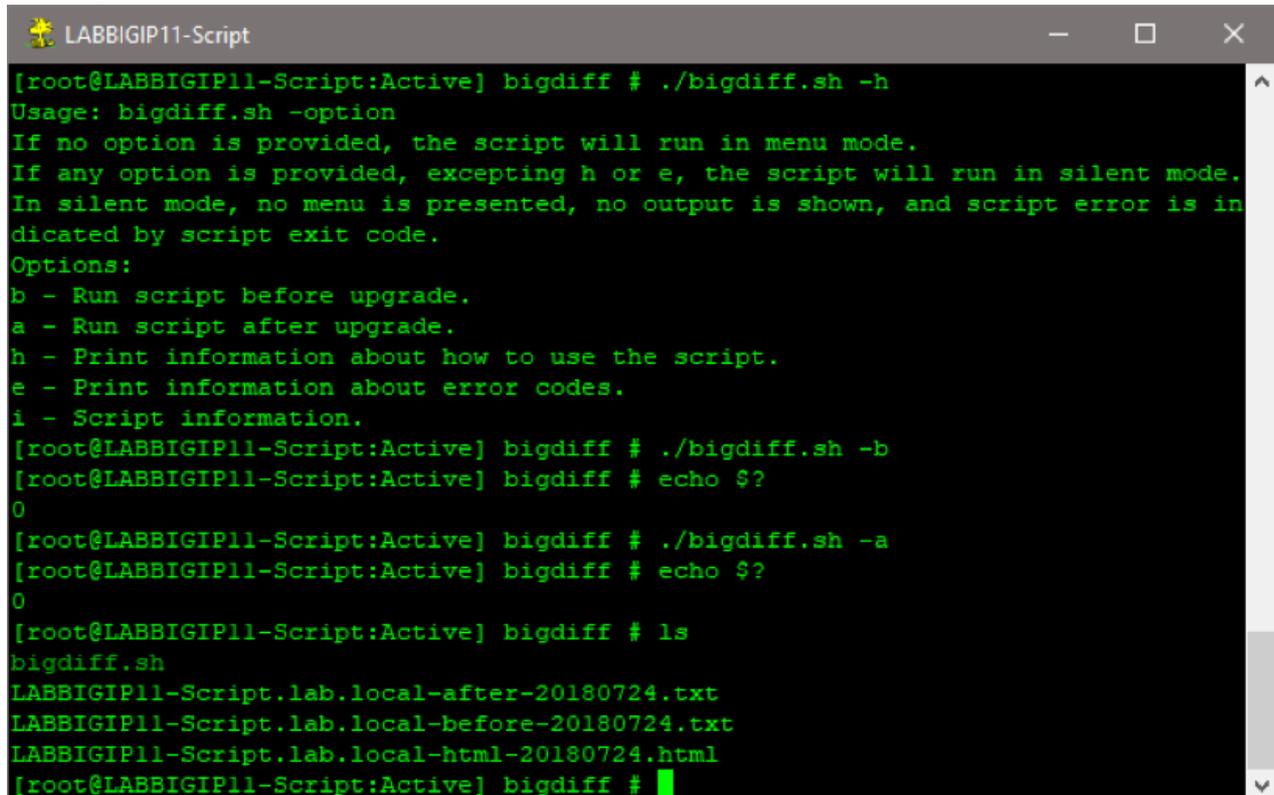
You can run the script, do the changes, and run the script again.

The script will then tell you if you broke something.

Silent Mode

Silent mode is mainly to be used to integrate with other tools.

The image above explains how to use.



```
LABBIGIP11-Script
[root@LABBIGIP11-Script:Active] bigdiff # ./bigdiff.sh -h
Usage: bigdiff.sh -option
If no option is provided, the script will run in menu mode.
If any option is provided, excepting h or e, the script will run in silent mode.
In silent mode, no menu is presented, no output is shown, and script error is indicated by script exit code.
Options:
b - Run script before upgrade.
a - Run script after upgrade.
h - Print information about how to use the script.
e - Print information about error codes.
i - Script information.
[root@LABBIGIP11-Script:Active] bigdiff # ./bigdiff.sh -b
[root@LABBIGIP11-Script:Active] bigdiff # echo $?
0
[root@LABBIGIP11-Script:Active] bigdiff # ./bigdiff.sh -a
[root@LABBIGIP11-Script:Active] bigdiff # echo $?
0
[root@LABBIGIP11-Script:Active] bigdiff # ls
bigdiff.sh
LABBIGIP11-Script.lab.local-after-20180724.txt
LABBIGIP11-Script.lab.local-before-20180724.txt
LABBIGIP11-Script.lab.local-html-20180724.html
[root@LABBIGIP11-Script:Active] bigdiff #
```

Conclusion

Read the information in the [code share page](#) about know issues.

I hope you find the script useful.

F5 Networks, Inc. | 401 Elliot Avenue West, Seattle, WA 98119 | 888-882-4447 | f5.com

F5 Networks, Inc.
Corporate Headquarters
info@f5.com

F5 Networks
Asia-Pacific
apacinfo@f5.com

F5 Networks Ltd.
Europe/Middle-East/Africa
emeainfo@f5.com

F5 Networks
Japan K.K.
f5j-info@f5.com