Introducing the Linux Health Checker

Peter Oberparleiter <peter.oberparleiter@de.ibm.com>

© 2012 IBM Corporation

Trademarks

The following are trademarks of the International Business Machines Corporation in the United States and/or other countries.

AIX*	IBM*	PowerVM	System z10	z/OS*
BladeCenter*	IBM eServer	PR/SM	WebSphere*	zSeries*
DataPower*	IBM (logo)*	Smarter Planet	z9*	z/VM*
DB2*	InfiniBand*	System x*	z10 BC	z/VSE
FICON*	Parallel Sysplex*	System z*	z10 EC	
GDPS*	POWER*	System z9*	zEnterprise	
HiperSockets	POWER7*			

* Registered trademarks of IBM Corporation

The following are trademarks or registered trademarks of other companies.

Adobe, the Adobe logo, PostScript, and the PostScript logo are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States, and/or other countries.
Cell Broadband Engine is a trademark of Sony Computer Entertainment, Inc. in the United States, other countries, or both and is used under license there from.
Java and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.
Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.
Windows Server and the Windows logo are trademarks of the Microsoft group of countries.
InfiniBand is a trademark and service mark of the InfiniBand Trade Association.
Intel, Intel logo, Intel Inside, Intel Inside logo, Intel Centrino, Intel Centrino logo, Celeron, Intel Xeon, Intel SpeedStep, Itanium, and Pentium are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.
UNIX is a registered trademark of The Open Group in the United States and other countries.
Linux is a registered trademark of Linus Torvalds in the United States, other countries, or both.
ITIL is a registered trademark, and a registered community trademark of the Office of Government Commerce, and is registered in the U.S. Patent and Trademark Office.
IT Infrastructure Library is a registered trademark of the Central Computer and Telecommunications Agency, which is now part of the Office of Government Commerce.

* All other products may be trademarks or registered trademarks of their respective companies.

Notes:

Performance is in Internal Throughput Rate (ITR) ratio based on measurements and projections using standard IBM benchmarks in a controlled environment. The actual throughput that any user will experience will vary depending upon considerations such as the amount of multiprogramming in the user's job stream, the I/O configuration, the storage configuration, and the workload processed. Therefore, no assurance can be given that an individual user will achieve throughput improvements equivalent to the performance ratios stated here.

IBM hardware products are manufactured from new parts, or new and serviceable used parts. Regardless, our warranty terms apply.

All customer examples cited or described in this presentation are presented as illustrations of the manner in which some customers have used IBM products and the results they may have achieved. Actual environmental costs and performance characteristics will vary depending on individual customer configurations and conditions.

This publication was produced in the United States. IBM may not offer the products, services or features discussed in this document in other countries, and the information may be subject to change without notice. Consult your local IBM business contact for information on the product or services available in your area.

All statements regarding IBM's future direction and intent are subject to change or withdrawal without notice, and represent goals and objectives only.

Information about non-IBM products is obtained from the manufacturers of those products or their published announcements. IBM has not tested those products and cannot confirm the performance, compatibility, or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

Prices subject to change without notice. Contact your IBM representative or Business Partner for the most current pricing in your geography.

Tux logo by Larry Ewing.

© 2012 IBM Corporation

Agenda – Part 1

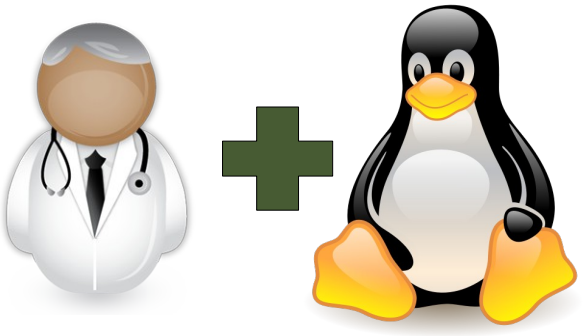
1. **Introducing health checking**
2. **Using the Linux Health Checker**
3. **How to write a check**

© 2012 IBM Corporation

Introducing health checking

- **What is a health check?**
  - A process that identifies conditions which may lead to problems
- **What is the Linux Health Checker?**
  - A tool that performs an automated health check of a Linux system
  - Checks status and configuration
  - Presents report on identified problems

→ **Helps keeping Linux systems healthy (operational)**



© 2012 IBM Corporation

What does it do?

- **Example problem classes**
  - Configuration errors
  - Deviations from best-practice setups
  - Hardware running in degraded mode
  - Unused accelerator hardware
  - Single point-of-failures

- **Detailed problem report**
  - Enable users to understand and solve problems
  - Make expert knowledge available to wider audience

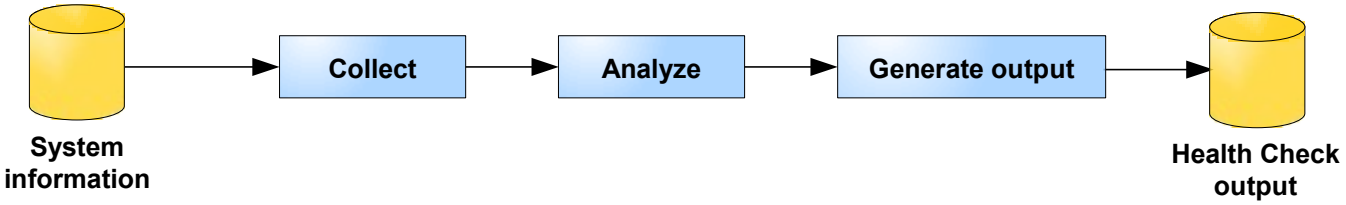
© 2012 IBM Corporation

Goals

- **Ease of use**
  - Simple setup: Install and run
  - Primary tasks easily accessible through command line interface
- **Flexibility through Framework/Plug-in concept**
  - Health check plug-ins
    - Contain all problem area specific knowledge
  - Consumer plug-ins
    - Handle output processing
  - Extend functionality by adding new plug-ins

© 2012 IBM Corporation

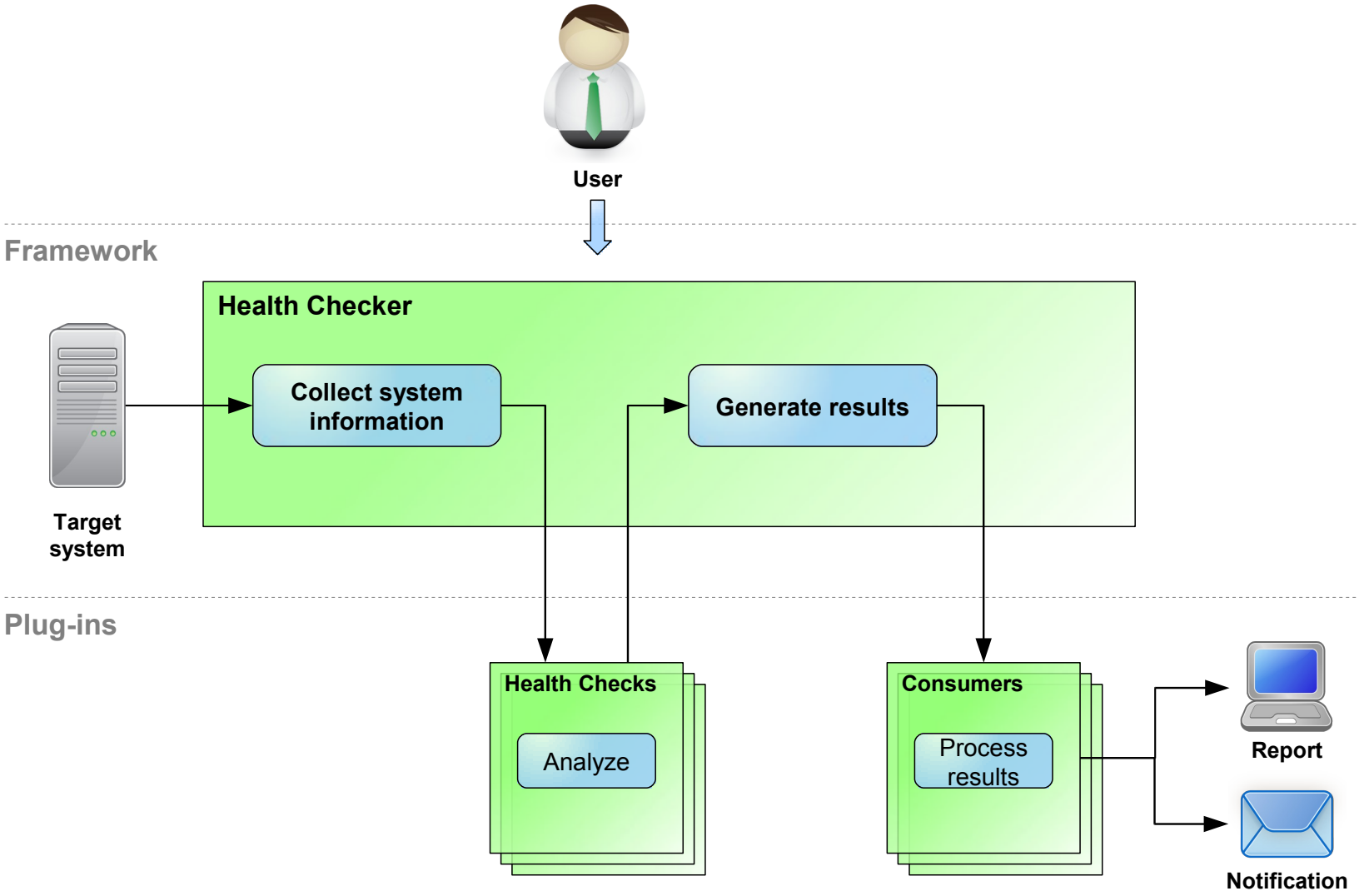
Basic approach to health checking



- **Collect system information**
  - File contents, for example /var/log/messages
  - Program output, for example /bin/df
- **Analyze information**
  - Find relevant data points
  - Compare with best-practice values
- **Produce report**

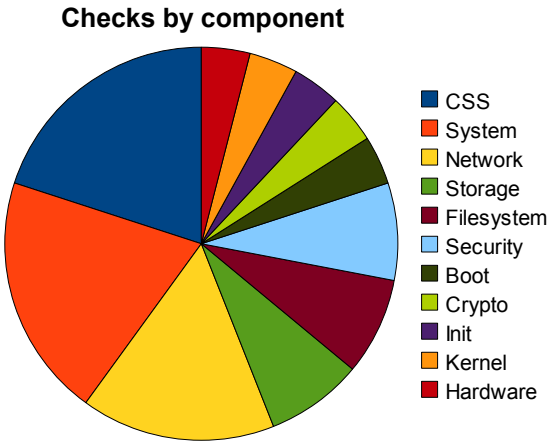
© 2012 IBM Corporation

System overview



© 2012 IBM Corporation

Health checks in version 1.0

Verify that the bootmap file is up-to-date	Screen users with superuser privileges
Check whether the path to the OpenSSL library is configured correctly	Identify network services that are known to be insecure
Identify unusable I/O devices	Identify multipath setups that consist of a single path only
Check for CHPIDs that are not available	Confirm that automatic problem reporting is activated
Identify I/O devices that are in use although they are on the exclusion list	Ensure that panic-on-oops is switched on
Identify I/O devices that are not associated with a device driver	Check whether the CPUs run with reduced capacity
Check for an excessive number of unused I/O devices	Spot getty programs on the /dev/console device
Check Linux on z/VM for the "nopav" DASD parameter	Identify unused terminals (TTY)
Check file systems for adequate free space	
Check file systems for an adequate number of free inodes	
Check whether the recommended runlevel is used and set as default	
Check the kernel message log for out-of-memory (OOM) occurrences	
Identify bonding interfaces that aggregate qeth interfaces with the same CHPID	
Check for an excessive error ratio for outbound HiperSockets traffic	
Check the inbound network traffic for an excessive error or drop ratio	
Identify qeth interfaces that do not have an optimal number of buffers	
Confirm that the dump-on-panic function is enabled	

© 2012 IBM Corporation

Agenda – Part 2

1. Introducing health checking
2. Using the Linux Health Checker
3. How to write a check

© 2012 IBM Corporation

Preparations

- **Requirements**
  - Linux
    - Framework should run on any hardware platform
    - Health checks may be platform specific
  - Perl 5.8 or later
    - Additional Perl modules which are usually part of default installation

- **Obtaining the Linux Health Checker**
  - Open source under Eclipse Public License v1.0
  - Download RPM or source package from http://lnxhc.sourceforge.net
  - Install using RPM command or **`make install`**

© 2012 IBM Corporation

First health check run

```
[user@lnxhost ~]$ lnxhc run
Collecting system information
Running checks (12 checks)
CHECK NAME                                HOST                      RESULT
=============================================================================
boot_zipl_update_required ..... lnxhost                   SUCCESS
css_ccw_availability ..... lnxhost                   SUCCESS
css_ccw_chpid ..... lnxhost                   SUCCESS
css_ccw_no_driver ..... lnxhost                   SUCCESS
css_ccw_unused_devices ..... lnxhost                   EXCEPTION-LOW

 >EXCEPTION css_ccw_unused_devices.many_unused_devices(low)
    Of 4664 I/O devices, 4659 (99.89%) are unused

fs_disk_usage ..... lnxhost                   SUCCESS
mm_oom_killer_triggered ..... lnxhost                   SUCCESS
net_hsi_tx_errors ..... lnxhost                   NOT APPLICABLE
ras_dump_on_panic ..... lnxhost                   EXCEPTION-HIGH

 >EXCEPTION ras_dump_on_panic.no_standalone(high)
    The dump-on-panic function is not enabled

sec_services_insecure ..... lnxhost                   SUCCESS
sys_sysctl_call_home ..... lnxhost                   NOT APPLICABLE
sys_sysinfo_cpu_cap ..... lnxhost                   SUCCESS

10 checks run, 2 exceptions found (use 'lnxhc run --replay -V' for details)
```


© 2012 IBM Corporation

Interpreting list view

- **Some health checks found no problems**

```
boot_zipl_update_required ..... lnxhost                SUCCESS
css_ccw_availability ........... lnxhost                SUCCESS
css_ccw_chpid ................. lnxhost                SUCCESS
css_ccw_no_driver ............. lnxhost                SUCCESS
```

- **A health check did not run**

```
net_hsi_tx_errors ............. lnxhost                NOT APPLICABLE
```

  - A requirement was not met, for example platform, hypervisor or Linux version

- **More reasons why a health check cannot run**
  - FAILED_SYSINFO
    - Some of the required input data could not be collected
  - FAILED_CHKPROG
    - There was a run-time error in the analysis step

© 2012 IBM Corporation

Interpreting list view (continued)

- **A potential problem was found**

```
css_ccw_unused_devices ................. lnxhost                    EXCEPTION-LOW

 >EXCEPTION css_ccw_unused_devices.many_unused_devices(low)
    Of 4664 I/O devices, 4659 (99.89%) are unused
```

  - Full exception ID
    - `css_ccw_unused_devices.many_unused_devices`
  - Exception severity
    - `low`
  - Exception summary
    - `Of 4664 I/O devices, 4659 (99.89%) are unused`

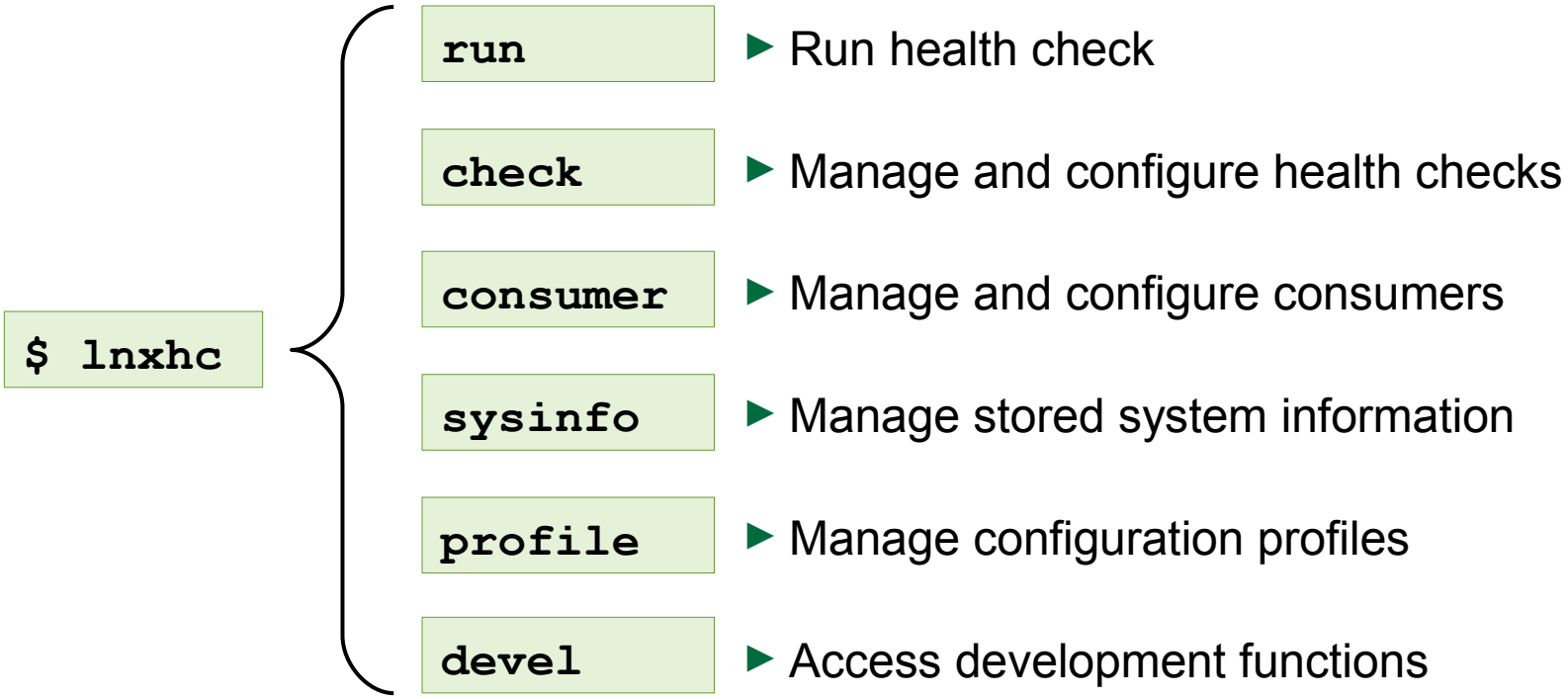
© 2012 IBM Corporation

Getting more details

```
[user@lnxhost ~]$ lnxhc run -V css_ccw_unused_devices
CHECK NAME                                HOST                                RESULT
=============================================================================================
css_ccw_unused_devices ................. lnxhost                             EXCEPTION-LOW

 >EXCEPTION css_ccw_unused_devices.many_unused_devices(low)

  SUMMARY
    Of 4664 I/O devices, 4659(99.89%) are unused

  EXPLANATION
    The  number  of  unused (offline) I/O devices, 4664 (99.89%) of a total of 4659,
    exceeds the specified threshold. During the boot process, Linux senses and analyzes
    All available  I/O devices, including unused devices. Therefore, unused devices
    unnecessarily consume memory and CPU time.

  SOLUTION
    Use the "cio_ignore" feature to exclude I/O devices that you do not need from  being
    sensed  and  analyzed. Be sure not to inadvertently exclude required devices. To ex-
    clude devices, you can use the "cio_ignore" kernel parameter or a command like this:

    echo "add <device_bus_id>" > /proc/cio_ignore

    where <device_bus_id> is the bus ID of an I/O device to be excluded.

  REFERENCE
    For  more  information  about  the  "cio_ignore"  feature, see the section about the
    "cio_ignore" kernel parameter in "Device Drivers, Features, and Commands".
```


© 2012 IBM Corporation

Additional functions



© 2012 IBM Corporation

Viewing list of available health checks

```
[user@lnxhost ~]$ lnxhc check --list

CHECK NAME                                COMPONENT                      STATE
=============================================================================
boot_zipl_update_required                 boot                           active
crypto_openssl_ibmca_config               crypto                         inactive
css_ccw_availability                      channel subsystem              active
css_ccw_chpid                             channel subsystem              active
css_ccw_ignored_online                    channel subsystem              active
css_ccw_no_driver                         channel subsystem              active
css_ccw_unused_devices                    channel subsystem              active
dasd_zvm_nopav                            storage                        active
fs_disk_usage                             filesystem                     active
fs_inode_usage                            filesystem                     active
init_runlevel                             init                           active
mm_oom_killer_triggered                   kernel                         active
net_bond_dev_chpid                        network                        active
net_hsi_tx_errors                         network                        active
net_inbound_packets                       network                        active
net_qeth_buffercount                      network                        active
ras_dump_on_panic                         system                         active
sec_non_root_uid_zero                     security                       active
sec_services_insecure                     security                       active
...
```


© 2012 IBM Corporation

Viewing health check information

```
[user@lnxhost ~]$ lnxhc check --info fs_disk_usage

Check fs_disk_usage (active)
============================

Title:
  Check file systems for adequate free space

Description:
  Some applications and administrative tasks require an adequate amount of free space on
  each mounted file system. If there is not enough free space, these applications might
  no longer be available or the complete system might be compromised. Regular monitoring
  of disk space usage averts this risk.

Exceptions:
  critical_limit=high (active)
  warn_limit=low (inactive)

Parameters:
  critical_limit=95
        File system usage (in percent) at which to raise a high-severity exception.
        Valid values are integers in the range 1 to 100.

        Default value is "95".

...
```


© 2012 IBM Corporation

Health check properties

- **Fixed properties**
  - Name
  - Meta-data
    - Component to be checked
    - Author name
  - Exceptions
    - Exception IDs
- **Some properties can be modified**
  - Activation state
  - Parameter values
  - Exception activation state and severity

© 2012 IBM Corporation

Modifying health check properties

- **Activation state**
  - Specifies if a check should be performed during health check run

```
[user@lnxhost ~]$ lnxhc check fs_disk_usage --state inactive
Setting state of check 'fs_disk_usage' to 'inactive'
Done.
```

- **Parameter values**
  - Values defined by health checks
  - Enable users to customize certain aspects of the health check

```
[user@lnxhost ~]$ lnxhc check --param fs_disk_usage.critical_limit=99
Setting value of parameter fs_disk_usage.critical_limit to '99'
Done.
```

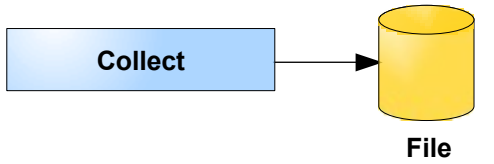
- **See man page for full list of properties**
  - `man lnxhc_properties.7`

© 2012 IBM Corporation

Advanced health checking modes

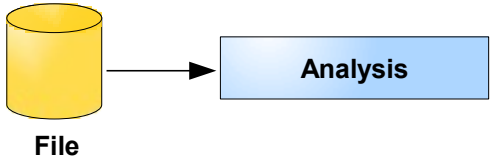
- **Collect data to file**

```
lnxhc sysinfo --collect --file lnxhost.sysinfo
```



- **Analyze from file**

```
lnxhc run --file lnxhost.sysinfo
```

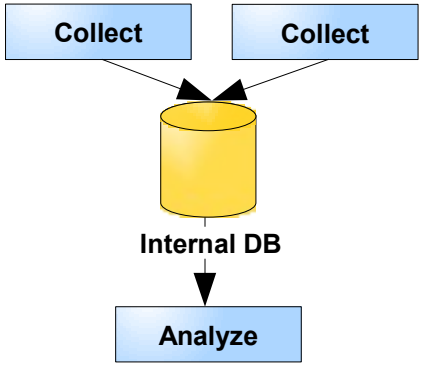


- **Analyze from remote host**

```
ssh user@remote lnxhc sysinfo -c -f - | lnxhc run -f -
```

- **Analyze from multiple hosts**

```
lnxhc sysinfo –clear
ssh user@remote1 lnxhc sysinfo -c -f - | lnxhc sysinfo -–merge -
ssh user@remote2 lnxhc sysinfo -c -f - | lnxhc sysinfo -–merge -
...
lnxhc run --current
```



© 2012 IBM Corporation

Agenda – Part 3

1. Introducing health checking
2. Using the Linux Health Checker
3. How to write a check

© 2012 IBM Corporation

Preparations for writing a health check

- **Required skills**
  - Basic programming skills
  - Any programming language
    - Shell
    - Perl
    - C, C++, etc.
- **Required resources**
  - Linux system
    - lnxhc package installed
    - Text editor
  - Some time

© 2012 IBM Corporation

Finding a suitable idea

- **Sources for health check ideas**
  - Own experience
  - Read documentation
  - Analyze previous outages
- **What to look for**
  - Single points of failure
  - System values reaching limits
  - Unhealthy combinations of configurations

© 2012 IBM Corporation

Example idea

- **What to check?**
  - Value of sysctl setting 'panic_on_oops' should be '1'
- **Why?**
  - “Kernel oops” = severe kernel error
  - Indication that the kernel can no longer be trusted
  - Kernel will continue anyway if panic_on_oops is '0'
- **How to check**

```
[user@lnxhost ~]$ cat /proc/sys/kernel/panic_on_oops
0
```

- **Solution**

```
echo 1 > /proc/sys/kernel/panic_on_oops
```


© 2012 IBM Corporation

Implementation without framework

- **Check program 'check.sh'**

```
#!/bin/bash

FILENAME="/proc/sys/kernel/panic_on_oops"
PANIC_ON_OOPS=`cat $FILENAME`

if [ "$PANIC_ON_OOPS" -eq 0 ] ; then
        echo "The panic-on-oops setting is disabled"
        echo "Enable it using 'echo 1 > /proc/sys/kernel/panic_on_oops'"
fi

exit 0
```

- **Sample output**

```
[user@lnxhost ~]$ ./check.sh
The panic-on-oops setting is disabled
Enable it using 'echo 1 > /proc/sys/kernel/panic_on_oops'
```


© 2012 IBM Corporation

Writing checks for the Linux Health Checker framework

- **One directory per check**
  - Directory name is check name
- **Files for**
  - Meta data
  - Text
  - Check program

```
panic_on_oops
├── definitions
├── descriptions
├── exceptions
└── check
```


© 2012 IBM Corporation

Definitions file

- **Contains data about the health check**

```
[check]
author = user@host
component = system

[sysinfo panic_on_oops]
file = /proc/sys/kernel/panic_on_oops

[exception no_panic_on_oops]
severity = high
```

- **Meta-data**
- **System information**
  - Files, command output, etc.
- **Exceptions**
  - ID and severity
- **Optional parameters**

© 2012 IBM Corporation

Descriptions file

- **Contains health check and parameter descriptions**

```
[title]
Ensure that panic-on-oops is enabled

[description]
The panic-on-oops setting ensures that a
Linux instance is stopped if a kernel
oops occurs.
```

- **Check title**
- **Basic check description**
- **Description of parameters**

© 2012 IBM Corporation

Exceptions file

- **Contains problem report text**

```
[summary no_panic_on_oops]
The panic-on-oops setting is disabled

[explanation no_panic_on_oops]
Without the panic-on-oops setting, a
Linux instance might keep running after
an oops.

[solution no_panic_on_oops]
Use the following command to enable the
panic-on-oops setting

echo 1 > /proc/sys/kernel/panic_on_oops

[reference no_panic_on_oops]
See kernel documentation on panic-on-oops
setting.
```

- **Problem summary**
- **Explanation**
  - Why is this a problem?
- **Solution**
  - Step-by-step instruction
- **Reference for further reading**
  - If available

© 2012 IBM Corporation

Check program

- **Implements health check analysis logic**

```
#!/bin/bash

FILENAME=$LNXHC_SYSINFO_panic_on_oops
PANIC_ON_OOPS=`cat $FILENAME`

if [ "$PANIC_ON_OOPS" -eq 0 ] ; then
  echo "no_panic_on_oops" >> $LNXHC_EXCEPTION
fi

exit 0
```

- **Access system information**
- **Analyze and report exception**
- **Indicate result code**
  - 0 = Success
  - 64 = Missing dependency
  - Other = Run-time error

© 2012 IBM Corporation

Putting it all together

```
[user@lnxhost ~]$ lnxhc run -V ./panic_on_oops
Collecting system information
Running checks (1 checks)
CHECK NAME                                HOST                                RESULT
=============================================================================================
panic_on_oops ........................... lnxhost                             EXCEPTION-HIGH

 >EXCEPTION panic_on_oops.no_panic_on_oops(high)

  SUMMARY
    The panic-on-oops setting is disabled

  EXPLANATION
    Without  the  panic-on-oops  setting,  a Linux instance might
    keep running after an oops.

  SOLUTION
    Use the following command to enable the panic-on-oops setting
    echo 1 > /proc/sys/kernel/panic_on_oops

  REFERENCE
    See kernel documentation on panic-on-oops setting.
```

- **If it doesn't work, add more “-V”s**
  - Increase level of verbosity to help debugging

© 2012 IBM Corporation

Wrap-up

- **To implement a check**
  - Create a directory
  - Add files
    - Meta-data
    - Text files
    - Check program
  - Run/debug until it works
- **Health check creation dialog** `lnxhc devel –-create-check my_check`
  - Creates template files according to dialog input
- **Once done, consider contributing it!**
  - Post to lnxhc-list@lists.sourceforge.net
  - See contribution guidelines at http://lnxhc.sourceforge.net/contributing.html

© 2012 IBM Corporation

Further reading

- **Man pages**
  - Once installed use 'apropos lnxhc' to list man pages
  - Also available on the web: http://lnxhc.sourceforge.net/manpages.html
- **User's Guide**
  - http://lnxhc.sourceforge.net/documentation.html
- **Main web page**
  - http://lnxhc.sourceforge.net/
- **Mailing list**
  - Open for questions, comments, ideas, contributions, etc.
  - lnxhc-list@lists.sourceforge.net

© 2012 IBM Corporation