# Scale Security – File Audit Logging and Using Vagrant to setup Scale Environments

**Christopher D. Maestas**

**Senior Architect – Spectrum Scale, IBM Systems**

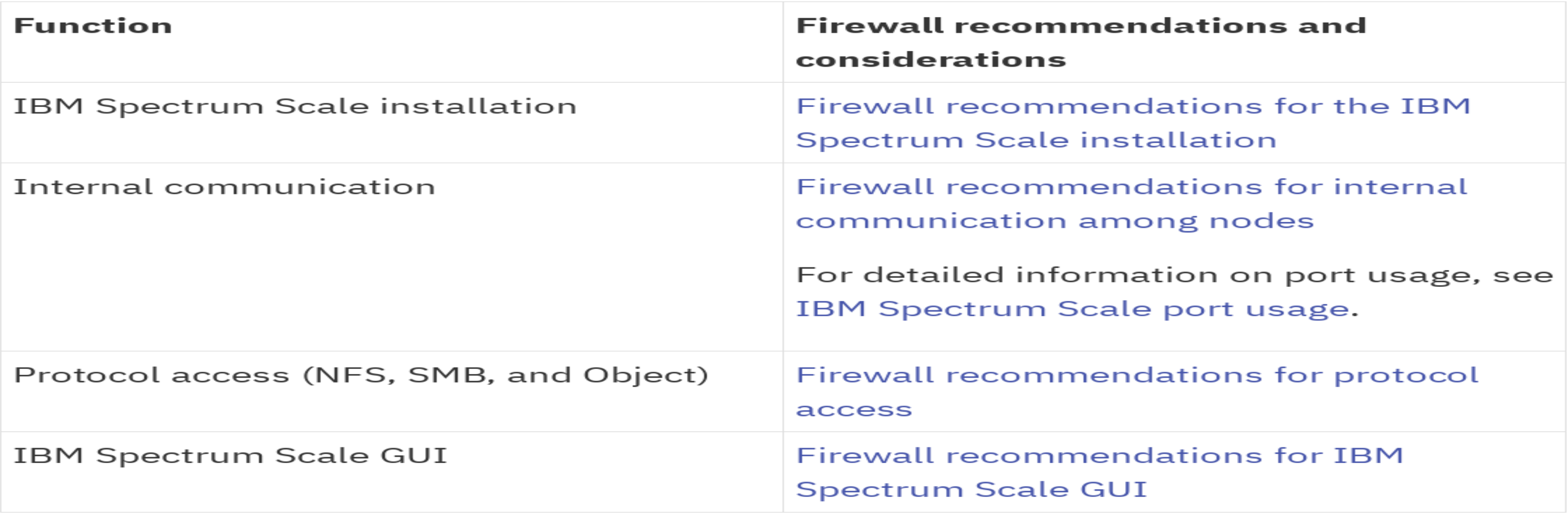
# Firewalls and SELinux

# Spectrum Scale - firewall

gpfs 1191/tcp General Parallel File System
gpfs 1191/udp General Parallel File System
# Dave Craft gpfs@ibm.com November 2004

Ports: https://www.ibm.com/support/knowledgecenter/STXKQY_5.0.0/com.ibm.spectrum.scale.v5r00.doc/bl1adv_firewall.htm

*Table 1. Firewall related information*



| Function | Firewall recommendations and considerations |
|---|---|
| IBM Spectrum Scale installation | Firewall recommendations for the IBM Spectrum Scale installation |
| Internal communication | Firewall recommendations for internal communication among nodes<br>For detailed information on port usage, see IBM Spectrum Scale port usage. |
| Protocol access (NFS, SMB, and Object) | Firewall recommendations for protocol access |
| IBM Spectrum Scale GUI | Firewall recommendations for IBM Spectrum Scale GUI |

# Spectrum Scale - SELinux

GPFS V3.5 and later run in

- 'permissive' mode, and
- 'enforcing' mode with 'SELINUXTYPE=targeted'

GPFS commands have to run unconfined

No SELinux profiles supplied for GPFS daemons and utilities

- Running GPFS command in a confined security context may fail
- Result in a large volume of logged security exception events.

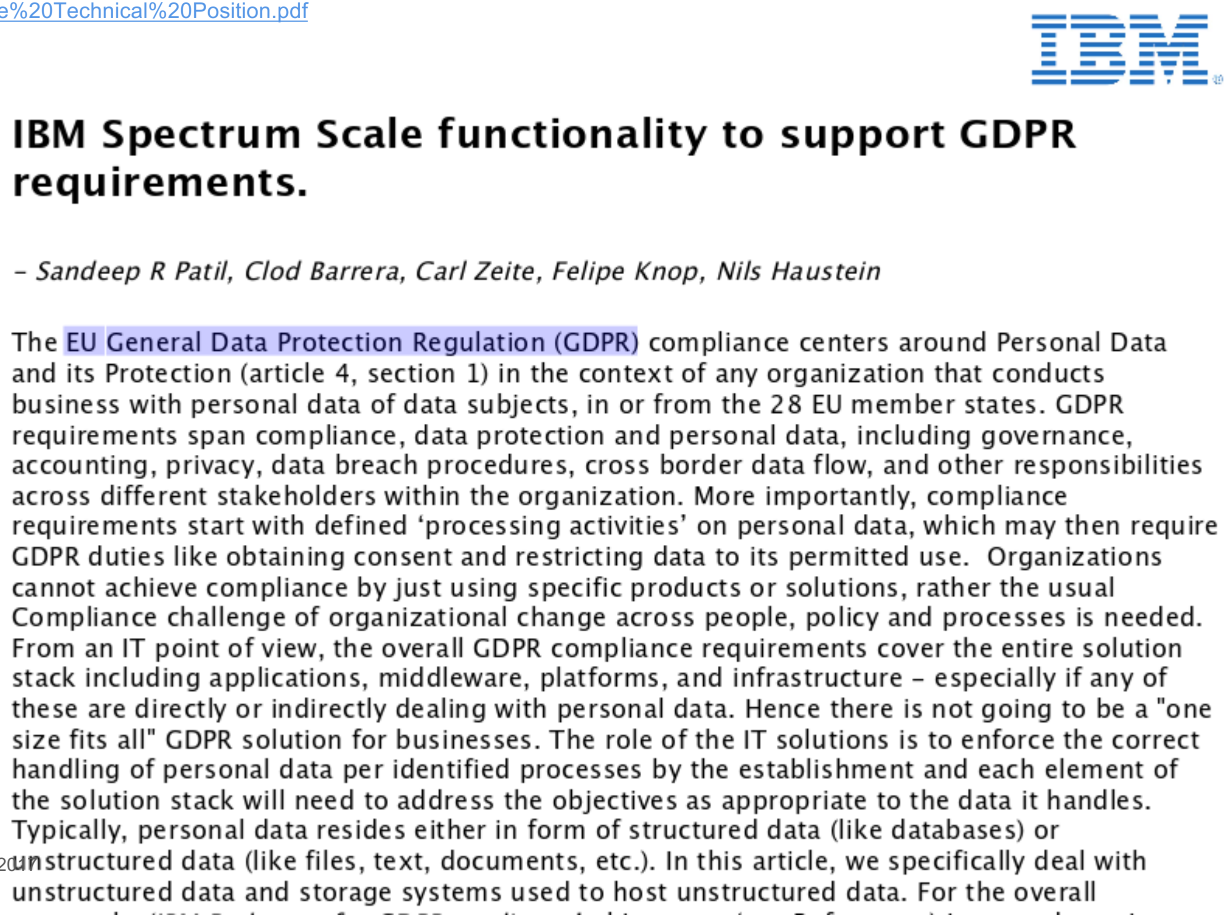
GPFS can hold files with per-inode security labels with limitations

https://www.ibm.com/developerworks/community/wikis/home?lang=en#!/wiki/General%20Parallel%20File%20System%20(GPFS)/page/SElinux

# EU GDPR

# EU General Data Protection Regulation (GDPR)

http://www-03.ibm.com/support/techdocs/atsmastr.nsf/5cb5ed706d254a8186256c71006d2e0a/1d33b61a55b2787185258251004c0566/$FILE/GDPR%20Compliance-%20Spectrum%20Scale%20Technical%20Position.pdf



## IBM Spectrum Scale functionality to support GDPR requirements.

*– Sandeep R Patil, Clod Barrera, Carl Zeite, Felipe Knop, Nils Haustein*

The EU General Data Protection Regulation (GDPR) compliance centers around Personal Data and its Protection (article 4, section 1) in the context of any organization that conducts business with personal data of data subjects, in or from the 28 EU member states. GDPR requirements span compliance, data protection and personal data, including governance, accounting, privacy, data breach procedures, cross border data flow, and other responsibilities across different stakeholders within the organization. More importantly, compliance requirements start with defined 'processing activities' on personal data, which may then require GDPR duties like obtaining consent and restricting data to its permitted use. Organizations cannot achieve compliance by just using specific products or solutions, rather the usual Compliance challenge of organizational change across people, policy and processes is needed. From an IT point of view, the overall GDPR compliance requirements cover the entire solution stack including applications, middleware, platforms, and infrastructure – especially if any of these are directly or indirectly dealing with personal data. Hence there is not going to be a "one size fits all" GDPR solution for businesses. The role of the IT solutions is to enforce the correct handling of personal data per identified processes by the establishment and each element of the solution stack will need to address the objectives as appropriate to the data it handles. Typically, personal data resides either in form of structured data (like databases) or unstructured data (like files, text, documents, etc.). In this article, we specifically deal with unstructured data and storage systems used to host unstructured data. For the overall

# SUDO – don’t run as root

# SUDO wrappers

https://www.ibm.com/support/knowledgecenter/en/STXKQY_5.0.0/com.ibm.spectrum.scale.v5r00.doc/bl1adm_sudowrapper.htm

## Breaking news – installtoolkit mostly works!

caveat with callhome and object configuration for CES

## Configuring sudo – visudo

/usr/lpp/mmfs/samples/sudoers.sample.

## Configuring the cluster to use sudo wrapper scripts

mmchcluster command with the --use-sudo-wrapper option.

## Configuring IBM Spectrum Scale GUI to use sudo wrapper

# Immutability – WORM

# Spectrum Scale immutability - certified for compliance

The immutability function in IBM Spectrum Scale Version 4.2 has been assessed for compliance in accordance to **US SEC17a-4f** rules, **German and Swiss laws and regulations** by a recognized auditor.

Assessment report: http://www.kpmg.de/bescheinigungen/RequestReport.aspx?41742

Certificate: https://www.kpmg.de/bescheinigungen/RequestReport.aspx?41743

KPMG

Review of the software IBM Spectrum Scale version 4.2

REPORT

International Business Machines Corporation
Armonk, NY

August 2016

# Immutability Overview

Immutability means preventing changes and deletion of files during retention time

Spectrum Scale Immutability provides WORM storage in GPFS fileset

- Immutable files cannot be changed or deleted during retention period
  - Deletion is possible when retention time is expired

Managing immutability works similar to other products

- Retention time can be set with last access date
- WORM protection can be set by removing write permission

Spectrum Scale also supports append-only mode

- An empty file can be set to append-only by removing and adding write permission
- Append-only file allows appends at the end
- Append-only file can be made immutable by removing write permission once again

# Fileset Immutability Archive Manager Mode

none: Default setting for a normal fileset

**advisory (ad)**: Allows setting retention times and WORM protection

- But files can be deleted with the proper permission

**noncompliant (nc)**: Advisory mode plus

- Files cannot be deleted if retention time is not expired.
- But retention times can be reset and files can be deleted but not changed

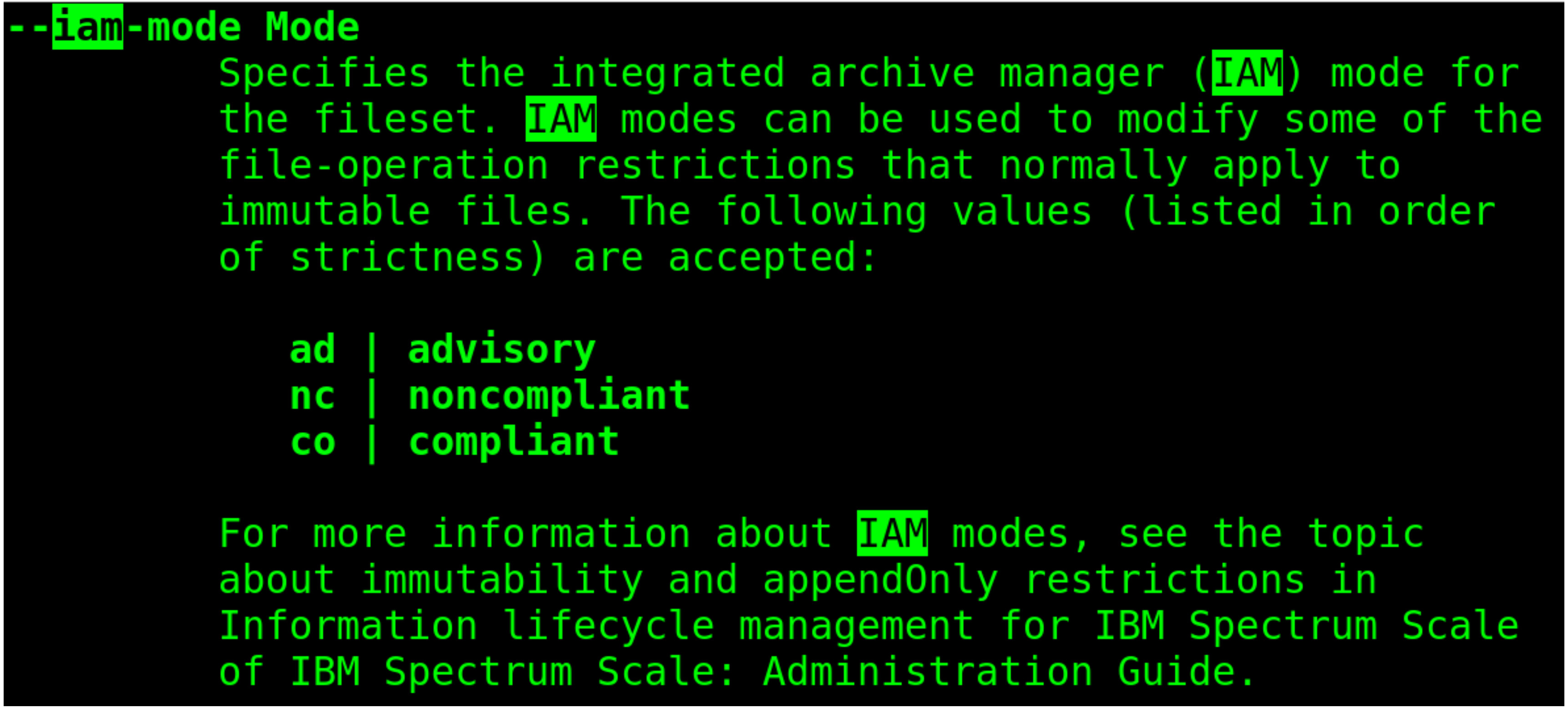
**compliant (co)**: noncompliant mode plus

- Retention time cannot be reset.
- When retention time has expired files can be deleted but not changed

Modes can be upgraded, but not downgraded

To set IAM use command: mmchfileset–iam-mode

# Look a man page! mmchfileset



```
--iam-mode Mode
         Specifies the integrated archive manager (IAM) mode for
         the fileset. IAM modes can be used to modify some of the
         file-operation restrictions that normally apply to
         immutable files. The following values (listed in order
         of strictness) are accepted:

            ad | advisory
            nc | noncompliant
            co | compliant

         For more information about IAM modes, see the topic
         about immutability and appendOnly restrictions in
         Information lifecycle management for IBM Spectrum Scale
         of IBM Spectrum Scale: Administration Guide.
```

# Set commands

## Setting retention time for file

```
touch –at MMddhhmmss filename
mmchattr –E yyyy-mm-dd[@hh:mm:ss] filename
```

## Setting file immutable

```
chmod –w filename
mmchattr –i yes filename
```

## Setting file to append-only

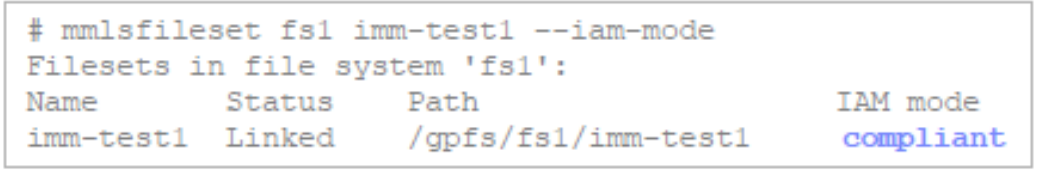
Create Empty file

```
chmod –w filename; chmod +w filename
mmchattr –a yes
```

# Showing commands

View fileset immutability mode

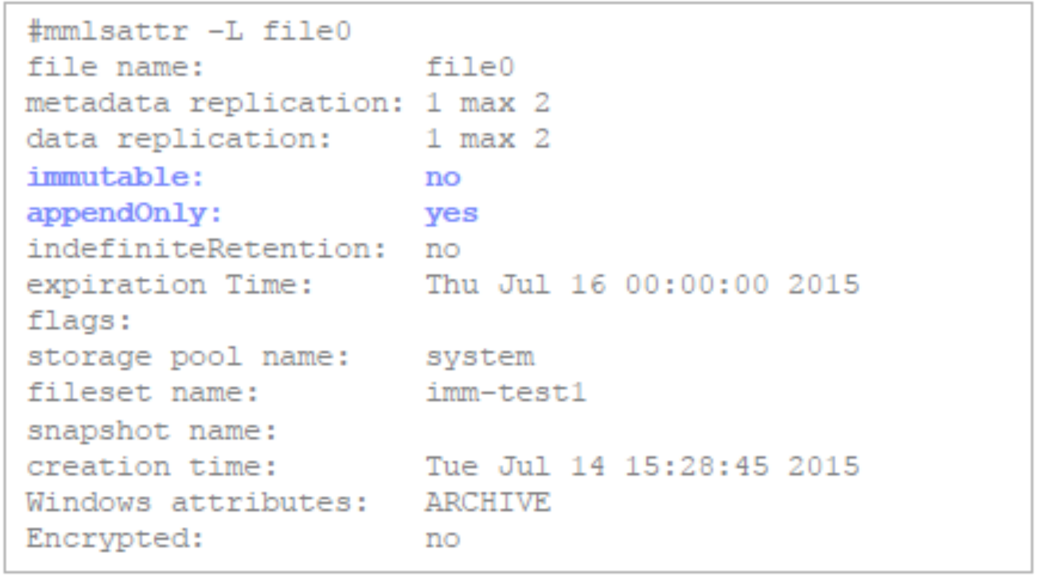
mmlsfilesetfsfset –iam-mode



```
# mmlsfileset fs1 imm-test1 --iam-mode
Filesets in file system 'fs1':
Name          Status    Path                       IAM mode
imm-test1     Linked    /gpfs/fs1/imm-test1         compliant
```

Show file immutability setting

mmlsattr –L filename



```
#mmlsattr -L file0
file name:                 file0
metadata replication:      1 max 2
data replication:          1 max 2
immutable:                 no
appendOnly:                yes
indefiniteRetention:       no
expiration Time:           Thu Jul 16 00:00:00 2015
flags:
storage pool name:         system
fileset name:              imm-test1
snapshot name:
creation time:             Tue Jul 14 15:28:45 2015
Windows attributes:        ARCHIVE
Encrypted:                 no
```

# Additional functions and options

## Deletion of file systems with compliant filesets (mmdelfs)

- Cluster-wide configuration parameter “indefiniteRetentionProtection” prevents this
  - Once set to yes deletion of file system is no longer possible
  - Cannot be set back to no once set to yes

## Deletion of compliant filesets (mmdelfileset)

- Not possible at GPFS 4.2 and higher

## Backup and restore using mmbackup

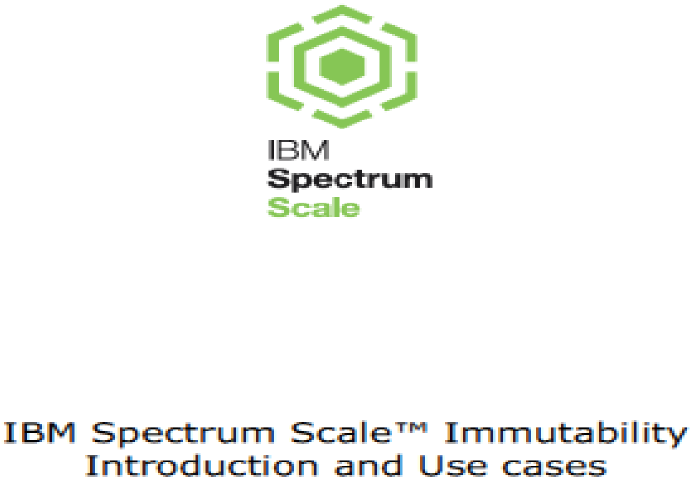
- Works with Spectrum Protect B/A client 7.1.3 and above
- In-place restore cannot overwrite and existing immutable file
- Out-of-place restore does not set the immutability attribute and retention time
  - Last access data will reflect retention time

## Spectrum Protect for Space Management 7.1.4 and above supports this

# Recommended reading

## Spectrum Scale Immutability Whitepaper:

http://www-03.ibm.com/support/techdocs/atsmastr.nsf/WebIndex/WP102620



IBM
Spectrum
Scale

IBM Spectrum Scale™ Immutability
Introduction and Use cases

# File Audit Logging

# Improved security and compliance

## New File Audit Logging capability *(Data Management Edition only)*

Track user accesses to filesystem and events

Supported across all nodes and all protocols

Parseable data stored in secure retention-protected fileset

Events that can be captured are:

- Open, Close, Destroy (Delete), Rename, Unlink, Remove Directory, Extended Attributed Change, Access Control List (ACL) change

# FAL - history

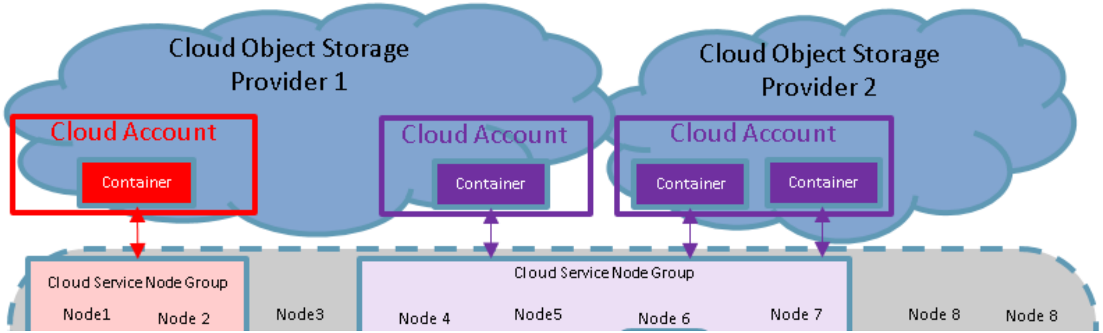
Integration with audit tools like Varonis and IBM Guardium

http://www.redbooks.ibm.com/redpapers/pdfs/redp5426.pdf

https://www.ibm.com/support/knowledgecenter/en/STXKQY_4.2.2/com.ibm.spectrum.scale.v4r22.doc/bl1adv_dpauditlogging.htm

Uses Light Weight Events (LWE) – What uses this today?

Transparent Cloud Tiering - TCT



# Sample Audit POC Tasks

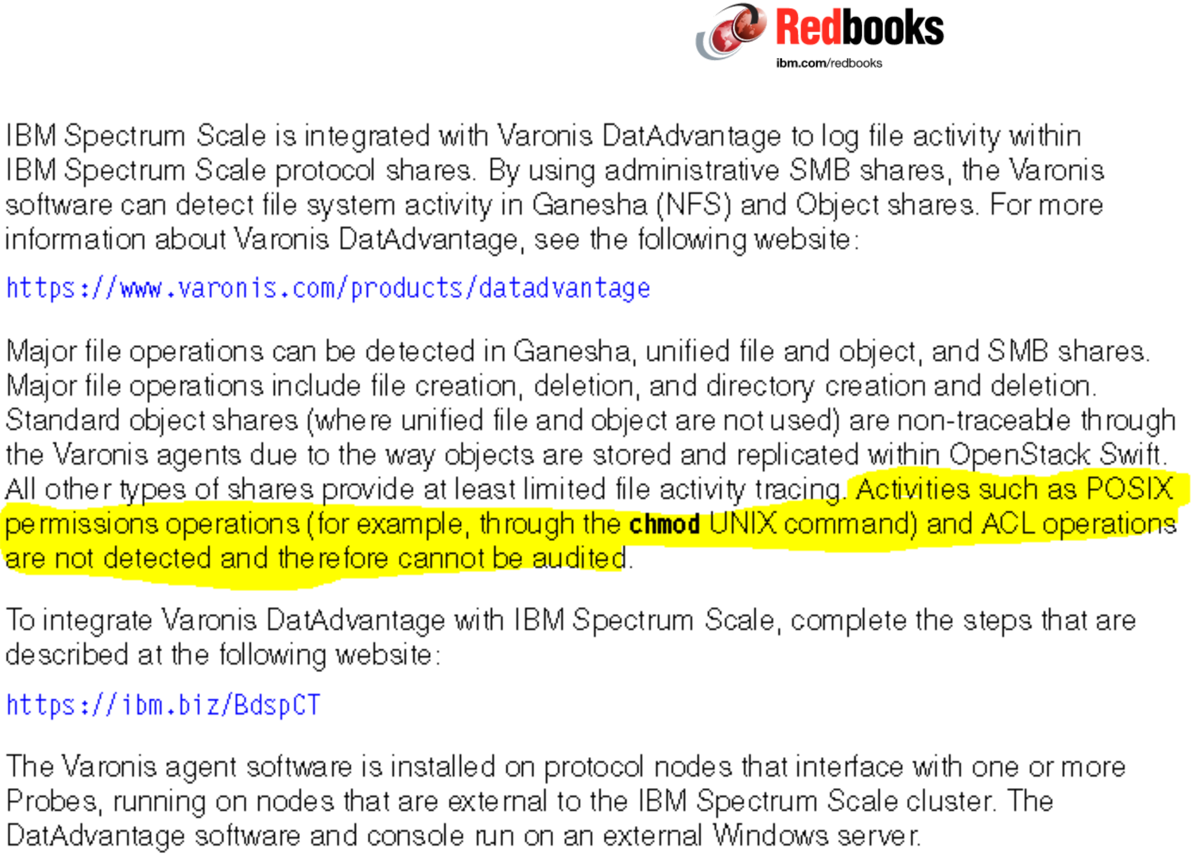
Demonstrate monitoring of file activity including user name, timestamp, and file location regardless of client type

Demonstrate monitoring of file activity without **endpoint (IBM Guardium or Varonis)** agent on clients

Create CSV-formatted reports of file activity and directory activity

Create report containing variable days of activity and deliver via file system, email, and api

# Audit logging with Varonis DatAdvantage



IBM Spectrum Scale is integrated with Varonis DatAdvantage to log file activity within IBM Spectrum Scale protocol shares. By using administrative SMB shares, the Varonis software can detect file system activity in Ganesha (NFS) and Object shares. For more information about Varonis DatAdvantage, see the following website:

https://www.varonis.com/products/datadvantage

Major file operations can be detected in Ganesha, unified file and object, and SMB shares. Major file operations include file creation, deletion, and directory creation and deletion. Standard object shares (where unified file and object are not used) are non-traceable through the Varonis agents due to the way objects are stored and replicated within OpenStack Swift. All other types of shares provide at least limited file activity tracing. Activities such as POSIX permissions operations (for example, through the **chmod** UNIX command) and ACL operations are not detected and therefore cannot be audited.

To integrate Varonis DatAdvantage with IBM Spectrum Scale, complete the steps that are described at the following website:

https://ibm.biz/BdspCT

The Varonis agent software is installed on protocol nodes that interface with one or more Probes, running on nodes that are external to the IBM Spectrum Scale cluster. The DatAdvantage software and console run on an external Windows server.

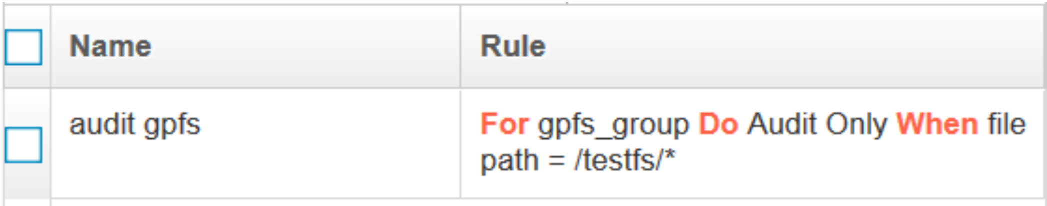
# Spectrum Scale Testing with IBM Guardium

9 node cluster

Traffic

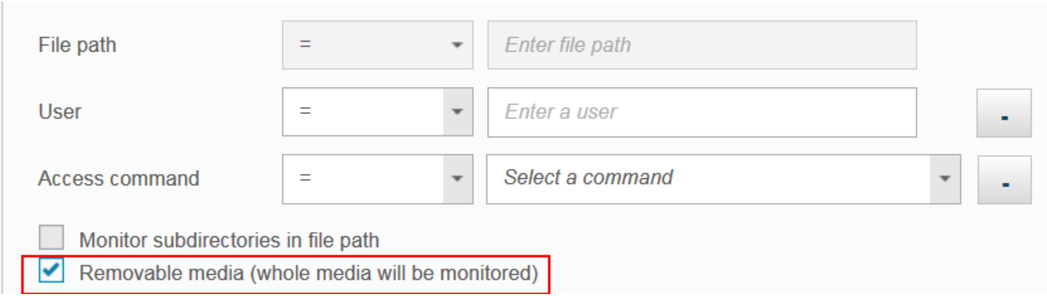
- FVT I/O Stress tests (autotest, mkfiles)
- Command Regression (as root)

STAPs installed on each node



| □ | Name | Rule |
|---|---|---|
| □ | audit gpfs | For gpfs_group Do Audit Only When file path = /testfs/* |

Audit only policy right now



File path = Enter file path

User = Enter a user -

Access command = Select a command -

□ Monitor subdirectories in file path

☑ Removable media (whole media will be monitored)

Audit removable media for NFS

# Spectrum Scale Testing with IBM Guardium

## What do we catch

Commands

- DELETE
- READ
- WRITE
  - Create file thru vi shows as a write
  - We catch data in inode
  - CREATE system call shows up as a WRITE
- EXEC (Execution)
- FILEOP (MKDIR, CHMOD, CHOWN)

Source Program

Db_user

OS-User

Object

## What do we not catch

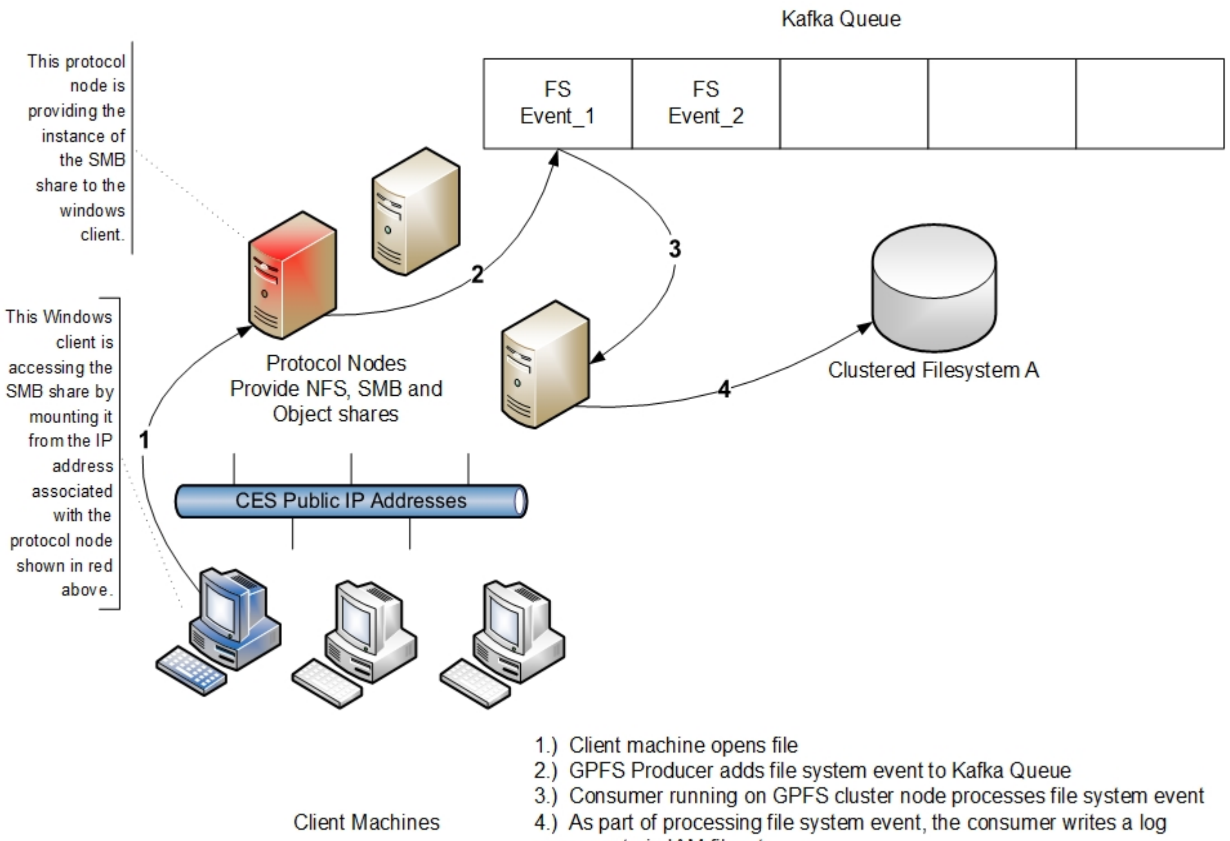
GPFS administration commands like:

```
mmchattr –P sp1 /testfs/subdir/*
#This changes the extended attributes of a file (root only)


mmapplypolicy /testfs/subdir -P mig.pol

# migrates data between storage pools (root only)
```

**To monitor root**

- In guard_tap.ini file add : fam_protect_privileged=1

# Spectrum Scale File Audit Logging - High Level Flow



This protocol node is providing the instance of the SMB share to the windows client.

This Windows client is accessing the SMB share by mounting it from the IP address associated with the protocol node shown in red above.

1.) Client machine opens file
2.) GPFS Producer adds file system event to Kafka Queue
3.) Consumer running on GPFS cluster node processes file system event
4.) As part of processing file system event, the consumer writes a log message to in IAM fileset

# File Audit Logging (FAL)

Now an API for 3rd party software IBM Guardium and Varonis

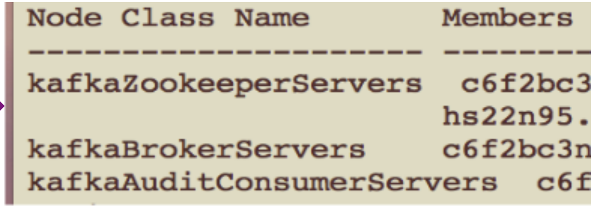
Light Weight Events (LWE) with Apache Kafka

**Producer** to publish stream of records: *1 million msg/s*

Live inside mmfsd (gpfs) daemon

**Consumer** subscribe to one or more topics and process stream: *3 million msg/s*

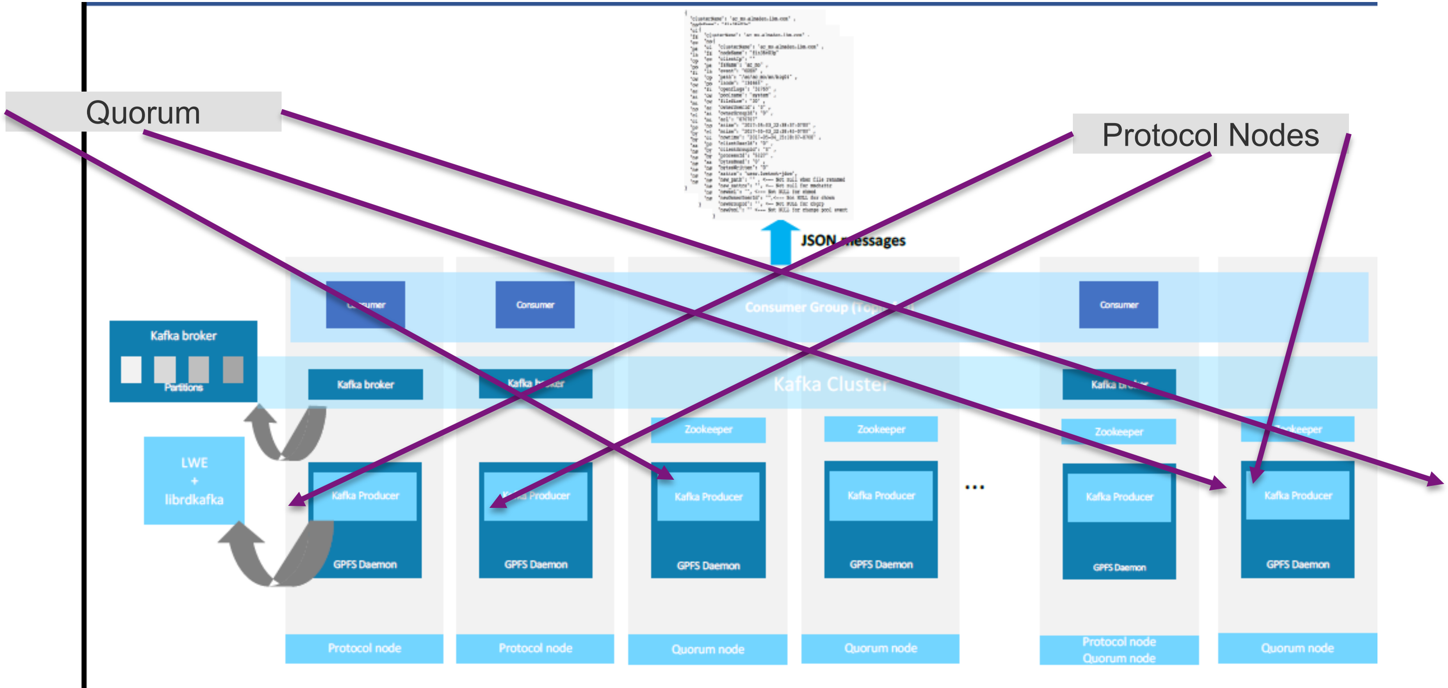
node classes – minimum of 3



```
Node Class Name           Members
---------------------  --------
kafkaZookeeperServers   c6f2bc3
                       hs22n95.
kafkaBrokerServers     c6f2bc3n
kafkaAuditConsumerServers  c6f
```

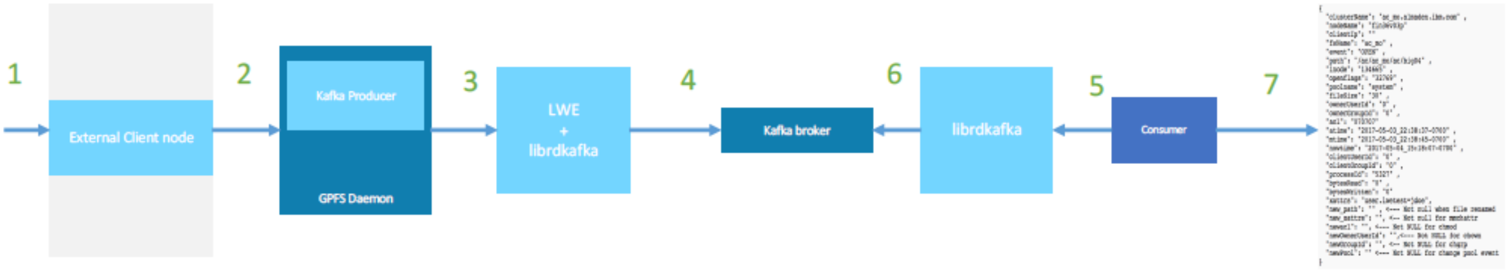
Monitor via CLI, mmhealth ,logfile, msgqueue or GUI (Events panel)!

# FAL - Architecture



*Zookeeper resides on the quorum nodes
**Kafka Brokers can reside on any node (not confined to protocol nodes as depicted in this figure)
***Using the standardized JSON format, client facing API can be derived.

# FAL – event flow



```
{
  "clusterName": "sc_mo.almaden.ibm.com" ,
  "nodeName": "fin36vklp"
  "clientIp": ""
  "fsName": "sc_mo" ,
  "event": "OPEN" ,
  "path": "/sc/sc_mo/sc/big04" ,
  "inode": "134665" ,
  "openFlags": "32769" ,
  "poolName": "system" ,
  "fileSize": "30" ,
  "ownerUserId": "0" ,
  "ownerGroupId": "0" ,
  "acl": "070707"
  "atime": "2017-05-03_22:38:37-0700" ,
  "mtime": "2017-05-03_22:38:45-0700" ,
  "newtime": "2017-05-04_15:18:47-0700" ,
  "clientUserId": "0" ,
  "clientGroupId": "0" ,
  "processId": "5327" ,
  "bytesRead": "0" ,
  "bytesWritten": "0"
  "xattrs": "user.lastest=jdoe",
  "new_path": "" , <--- Not null when file renamed
  "new_xattrs": "", <-- Not null for mmchattr
  "newacl": "", <--- Not NULL for chmod
  "newOwnerUserId": "",<--- Not NULL for chown
  "newGroupId": "", <-- Not NULL for chgrp
  "newPool": "" <--- Not NULL for change pool event
}
```

JSON messages

| SeqNbr | Description |
|---|---|
| 1 | Client performs a file operation ( read/ write/ remove, ..) on a file in an audited filesystem |
| 2 | External client node sends the client request to the relevant gpfs-node |
| 3 | Gpfs daemon using internal LWE (lightweight events) machinery sends the events to the Kafka MsgQueue using librdkafka |
| 4 | Event messages are reliably delivered to the Kafka Broker listening on this topic. |

| SeqNbr | Description |
|---|---|
| 5, 6 | Consumers belonging to a consumerGroup listening on this event topic, will periodically pull events from the Kafka Broker queue via librdkafka |
| 7 | Consumers will write the consumed events from the MsgQueue into the audited filesystem’s “.audit_log” fileset. |

# Install and configuration

Only Linux nodes (RHEL and Ubuntu)

Linux Kernel version above > 3.10

Minimum of 3 Linux quorum nodes

Minimum of 3 nodes must be designated as Broker nodes

Supported hardware platforms

  (x86 and PPCLE)
  RHEL supported on x86 and PPC LE
  Ubuntu is only supported on x86

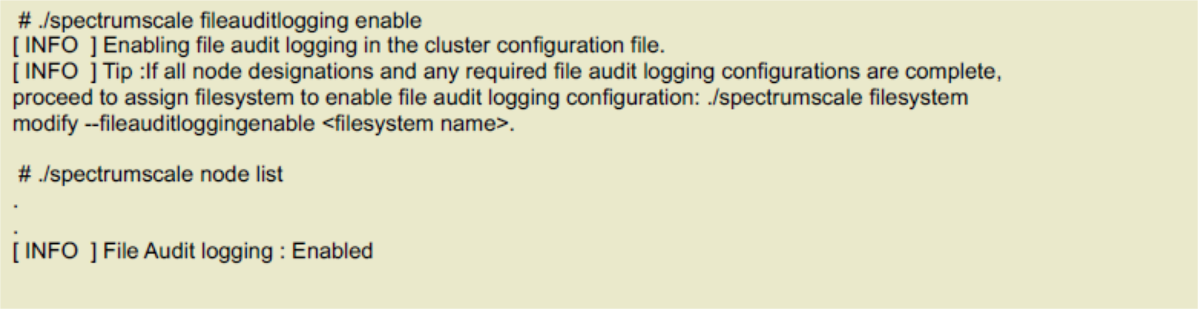
Advanced License edition or the Data Management edition

During Installation, most configuration is automatically done and stored in /opt/kafka folder

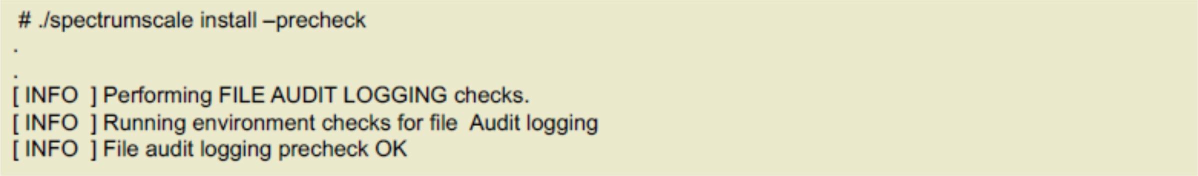
Free space requirements

  >1 GB local disk space per file system being audited
  > 2 GB local disk space per file system being audited on all broker nodes

# Installation

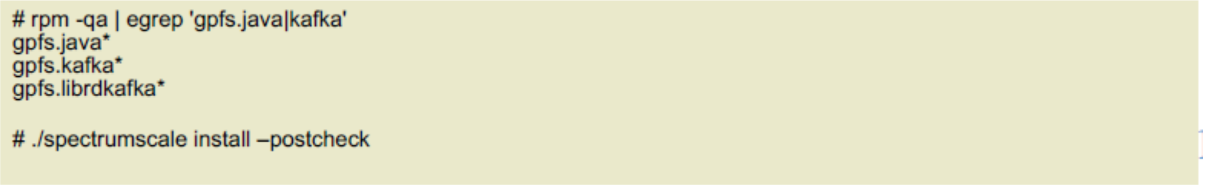


```
 # ./spectrumscale fileauditlogging enable
[ INFO  ] Enabling file audit logging in the cluster configuration file.
[ INFO  ] Tip :If all node designations and any required file audit logging configurations are complete,
proceed to assign filesystem to enable file audit logging configuration: ./spectrumscale filesystem
modify --fileauditloggingenable <filesystem name>.

 # ./spectrumscale node list
.
.
[ INFO  ] File Audit logging : Enabled
```



```
 # ./spectrumscale install –precheck
.
.
[ INFO  ] Performing FILE AUDIT LOGGING checks.
[ INFO  ] Running environment checks for file  Audit logging
[ INFO  ] File audit logging precheck OK
```

After install completes, verify that install installed the necessary GPFS rpms



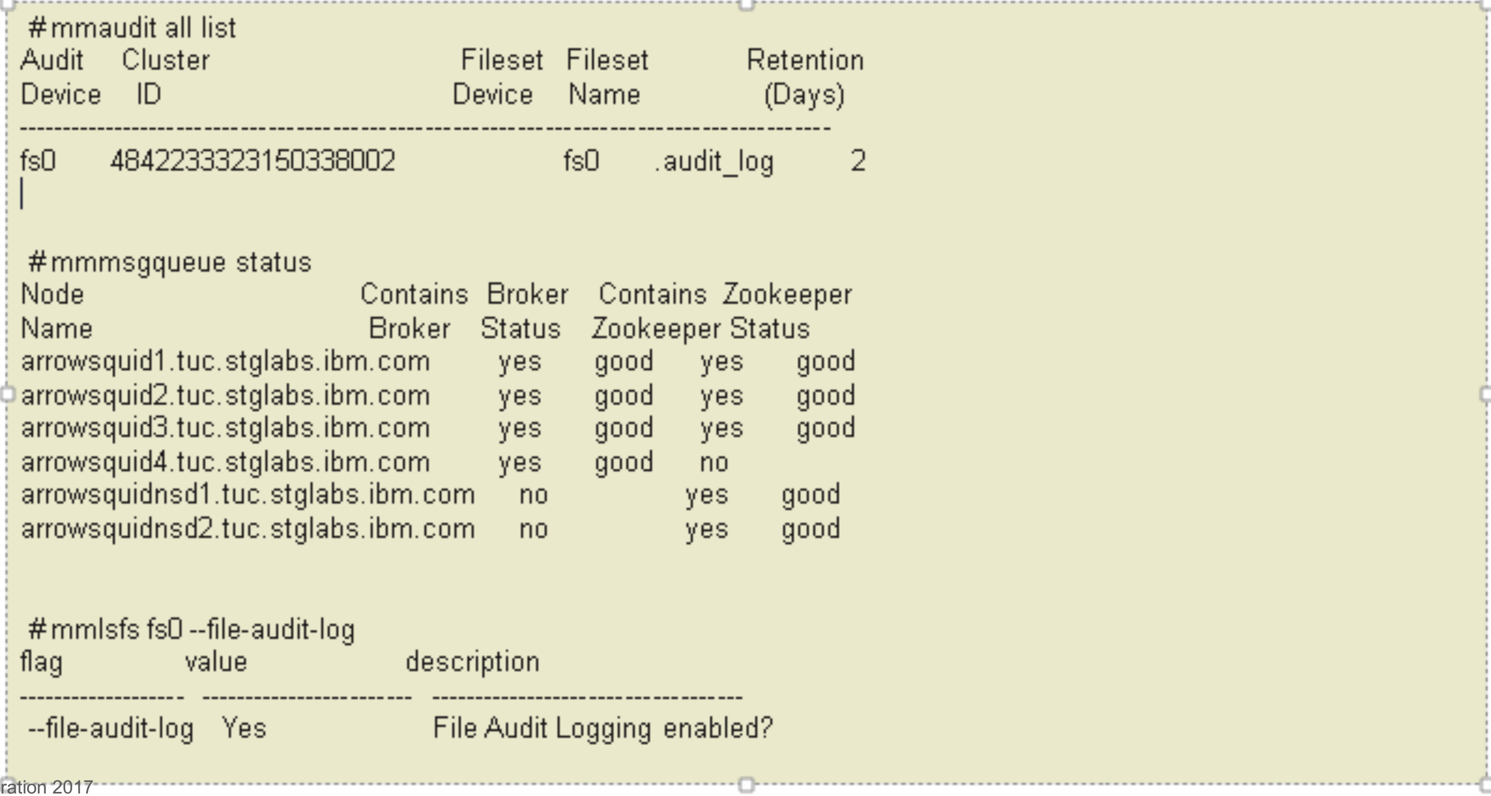
```
# rpm -qa | egrep 'gpfs.java|kafka'
gpfs.java*
gpfs.kafka*
gpfs.librdkafka*

# ./spectrumscale install –postcheck
```

# Installation and verification

Validate using mm-CLI commands to ensure file audit logging is enabled

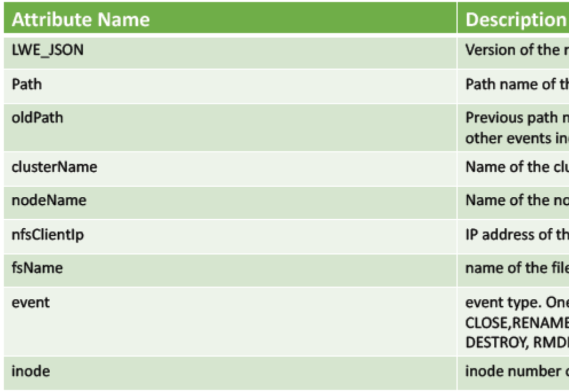
Durir

```
# ./spect
[ INFO ]
[ INFO ]
 './spectr
[ INFO ]
[ INFO ]
```

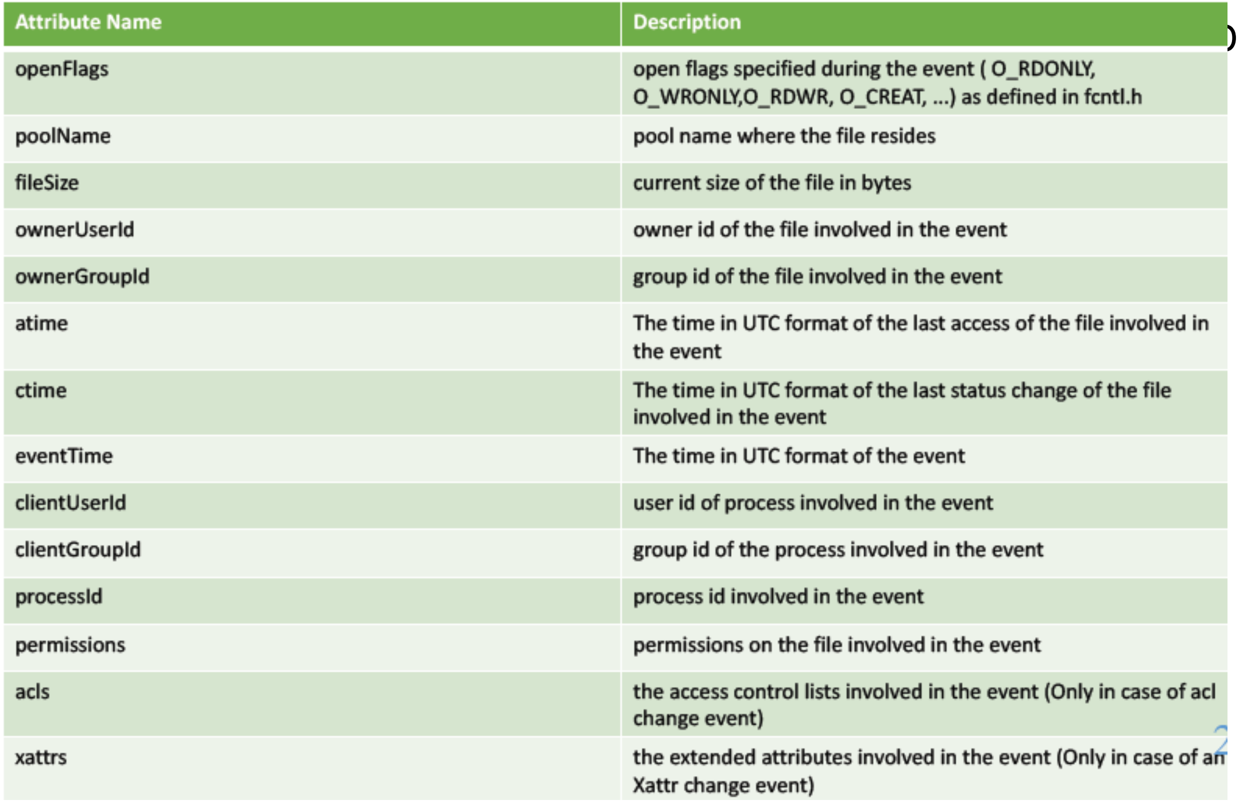


```
 #mmaudit all list
Audit    Cluster                        Fileset  Fileset          Retention
Device    ID                           Device   Name              (Days)
-----------------------------------------------------------------------------
fs0     4842233323150338002            fs0     .audit_log         2
|

 #mmmsgqueue status
Node                              Contains  Broker   Contains  Zookeeper
Name                               Broker   Status   Zookeeper Status
arrowsquid1.tuc.stglabs.ibm.com       yes      good      yes      good
arrowsquid2.tuc.stglabs.ibm.com       yes      good      yes      good
arrowsquid3.tuc.stglabs.ibm.com       yes      good      yes      good
arrowsquid4.tuc.stglabs.ibm.com       yes      good      no
arrowsquidnsd1.tuc.stglabs.ibm.com     no                 yes     good
arrowsquidnsd2.tuc.stglabs.ibm.com     no                 yes     good


 #mmlsfs fs0 --file-audit-log
flag               value                  description
----------------  --------------------  -----------------------------
 --file-audit-log   Yes                   File Audit Logging enabled?
```

# What is logged



| Attribute Name | Description |
|---|---|
| LWE_JSON | Version of the |
| Path | Path name of th |
| oldPath | Previous path<br>other events in |
| clusterName | Name of the cl |
| nodeName | Name of the no |
| nfsClientIp | IP address of th |
| fsName | name of the file |
| event | event type. One<br>CLOSE,RENAME<br>DESTROY, RMD |
| inode | inode number |



| Attribute Name | Description |
|---|---|
| openFlags | open flags specified during the event ( O_RDONLY, O_WRONLY,O_RDWR, O_CREAT, ...) as defined in fcntl.h |
| poolName | pool name where the file resides |
| fileSize | current size of the file in bytes |
| ownerUserId | owner id of the file involved in the event |
| ownerGroupId | group id of the file involved in the event |
| atime | The time in UTC format of the last access of the file involved in the event |
| ctime | The time in UTC format of the last status change of the file involved in the event |
| eventTime | The time in UTC format of the event |
| clientUserId | user id of process involved in the event |
| clientGroupId | group id of the process involved in the event |
| processId | process id involved in the event |
| permissions | permissions on the file involved in the event |
| acls | the access control lists involved in the event (Only in case of acl change event) |
| xattrs | the extended attributes involved in the event (Only in case of an Xattr change event) |

# What gets Monitored

Acquire most common types of file activity:

- **open, close, delete, rename, POSIX permission changes, ACL changes, etc.**
- **Don’t capture internal operations (e.g., restripe)**

Events captured within GPFS daemon – represent attributes of filesystem action at that point

Example audit log entry:

```
{"LWE_JSON": "0.0.1", "path": "/newfs/1Kfile2.restore", "oldPath": null,
"clusterName": "pardie.cluster", "nodeName": "c6f2bc3n10", "nfsClientIp": "",
"fsName": "newfs", "event": "OPEN", "inode": "26626", "openFlags": "32962",
"poolName": "sp1", "fileSize": "0", "ownerUserId": "0", "ownerGroupId": "0",
"atime": "2017-10-25_12:36:22-0400", "ctime": "2017-10-25_12:36:22-0400",
"eventTime": "2017-10-25_12:36:22-0400", "clientUserId": "0", "clientGroupId":
"0", "processId": "10437", "permissions": "200100644", "acls": "u::rwc, g::r,
o::r, ", "xattrs": null }
```

# Log Files for Auditing

Each file system enabled has a dedicated fileset where the audit logs will go.

- Default option is .audit_log at the root of the file system.

.audit_log fileset is created as **IAM mode noncompliant.**

- Advisory mode plus
  - Files cannot be deleted if retention time is not expired.
  - But retention times can be reset and files can be deleted but not changed

AuditLog files are nested within /FSNAME/.audit_log/topic/year/month/date/*

Log file is written in append only mode

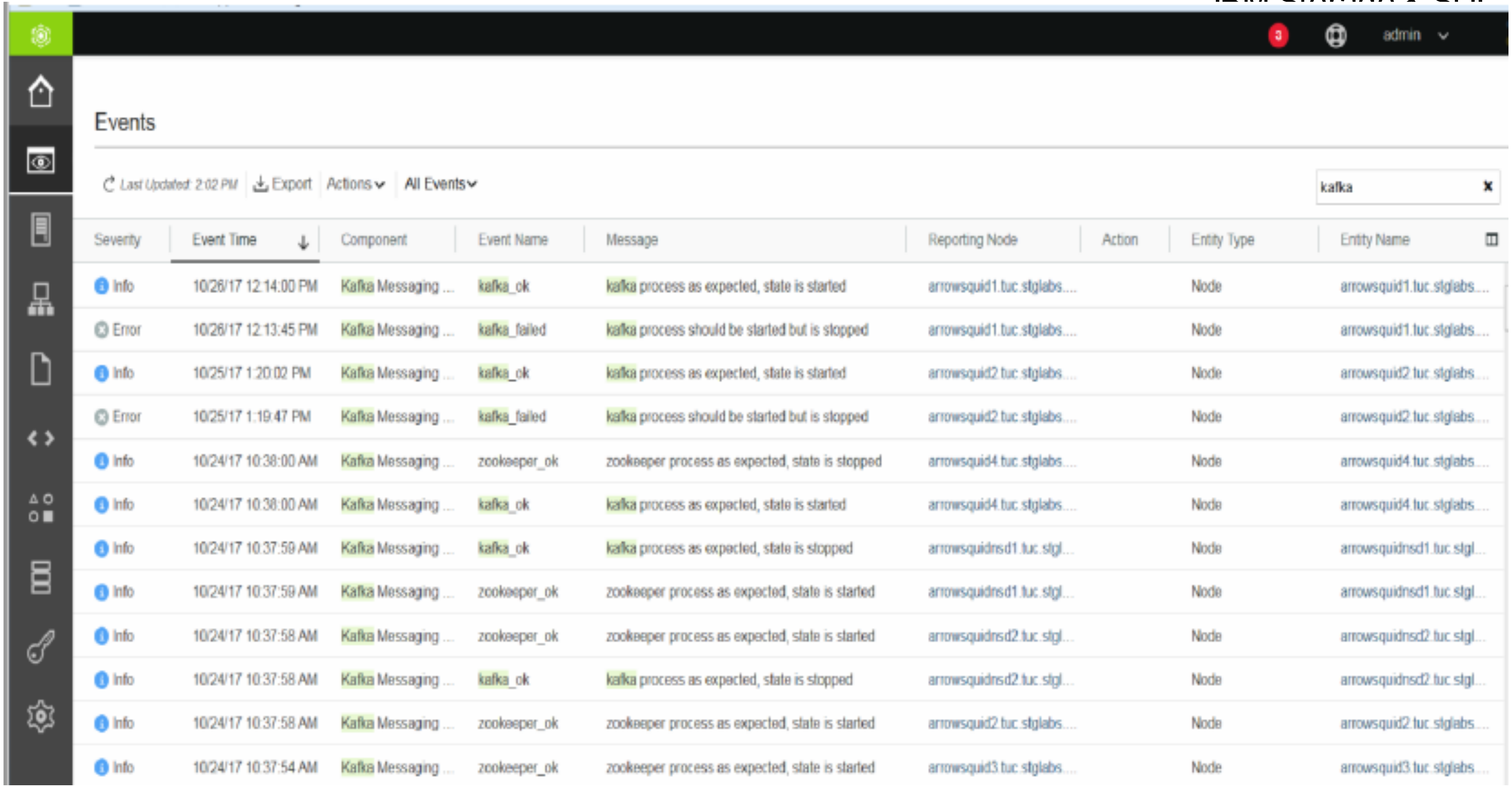
Rotation to a new log file upon reaching a threshold (500,000 events), then compressed and marked immutable for the retention period.

Default retention period is 365 days

Live events can be monitored by tailing the current auditLogFile<…>

Easy to search and consume

# FAL in the GUI

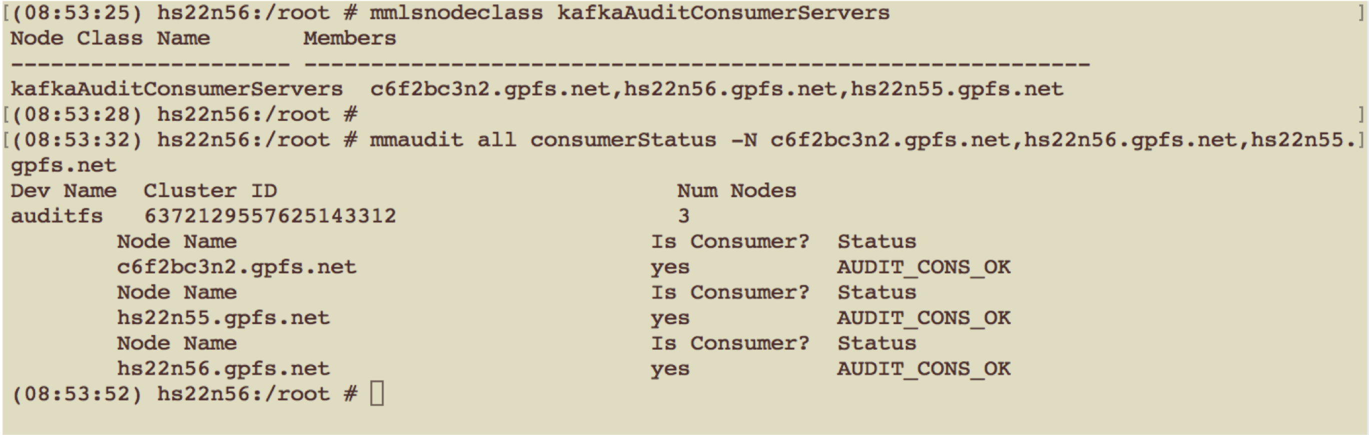


Events

| Severity | Event Time | Component | Event Name | Message | Reporting Node | Action | Entity Type | Entity Name |
|---|---|---|---|---|---|---|---|---|
| Info | 10/26/17 12:14:00 PM | Kafka Messaging ... | kafka_ok | kafka process as expected, state is started | arrowsquid1.tuc.stglabs.... | | Node | arrowsquid1.tuc.stglabs.... |
| Error | 10/26/17 12:13:45 PM | Kafka Messaging ... | kafka_failed | kafka process should be started but is stopped | arrowsquid1.tuc.stglabs.... | | Node | arrowsquid1.tuc.stglabs.... |
| Info | 10/25/17 1:20:02 PM | Kafka Messaging ... | kafka_ok | kafka process as expected, state is started | arrowsquid2.tuc.stglabs.... | | Node | arrowsquid2.tuc.stglabs.... |
| Error | 10/25/17 1:19:47 PM | Kafka Messaging ... | kafka_failed | kafka process should be started but is stopped | arrowsquid2.tuc.stglabs.... | | Node | arrowsquid2.tuc.stglabs.... |
| Info | 10/24/17 10:38:00 AM | Kafka Messaging ... | zookeeper_ok | zookeeper process as expected, state is stopped | arrowsquid4.tuc.stglabs.... | | Node | arrowsquid4.tuc.stglabs.... |
| Info | 10/24/17 10:38:00 AM | Kafka Messaging ... | kafka_ok | kafka process as expected, state is started | arrowsquid4.tuc.stglabs.... | | Node | arrowsquid4.tuc.stglabs.... |
| Info | 10/24/17 10:37:59 AM | Kafka Messaging ... | kafka_ok | kafka process as expected, state is stopped | arrowsquidnsd1.tuc.stgl... | | Node | arrowsquidnsd1.tuc.stgl... |
| Info | 10/24/17 10:37:59 AM | Kafka Messaging ... | zookeeper_ok | zookeeper process as expected, state is started | arrowsquidnsd1.tuc.stgl... | | Node | arrowsquidnsd1.tuc.stgl... |
| Info | 10/24/17 10:37:58 AM | Kafka Messaging ... | zookeeper_ok | zookeeper process as expected, state is started | arrowsquidnsd2.tuc.stgl... | | Node | arrowsquidnsd2.tuc.stgl... |
| Info | 10/24/17 10:37:58 AM | Kafka Messaging ... | kafka_ok | kafka process as expected, state is stopped | arrowsquidnsd2.tuc.stgl... | | Node | arrowsquidnsd2.tuc.stgl... |
| Info | 10/24/17 10:37:58 AM | Kafka Messaging ... | zookeeper_ok | zookeeper process as expected, state is started | arrowsquid2.tuc.stglabs.... | | Node | arrowsquid2.tuc.stglabs.... |
| Info | 10/24/17 10:37:54 AM | Kafka Messaging ... | zookeeper_ok | zookeeper process as expected, state is started | arrowsquid3.tuc.stglabs.... | | Node | arrowsquid3.tuc.stglabs.... |

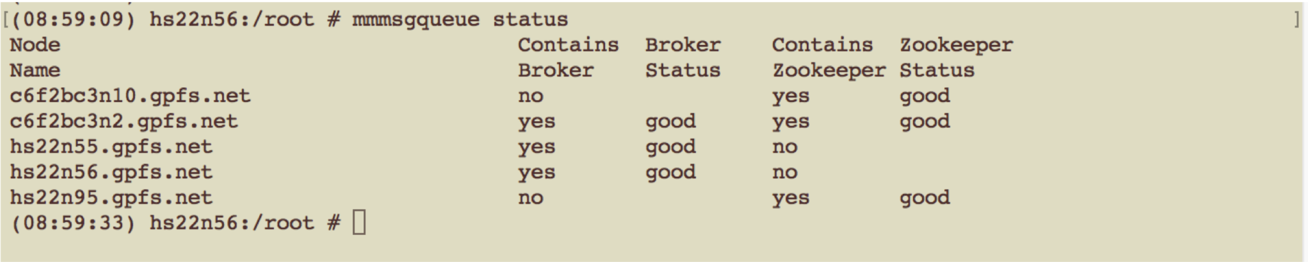
# CLI Monitoring

mmaudit all consumerStatus –N …



```
(08:53:25) hs22n56:/root # mmlsnodeclass kafkaAuditConsumerServers
Node Class Name       Members
--------------------- ----------------------------------------------------------
kafkaAuditConsumerServers  c6f2bc3n2.gpfs.net,hs22n56.gpfs.net,hs22n55.gpfs.net
(08:53:28) hs22n56:/root #
(08:53:32) hs22n56:/root # mmaudit all consumerStatus -N c6f2bc3n2.gpfs.net,hs22n56.gpfs.net,hs22n55.
gpfs.net
Dev Name  Cluster ID                                       Num Nodes
auditfs   6372129557625143312                              3
        Node Name                                        Is Consumer?  Status
        c6f2bc3n2.gpfs.net                               yes           AUDIT_CONS_OK
        Node Name                                        Is Consumer?  Status
        hs22n55.gpfs.net                                 yes           AUDIT_CONS_OK
        Node Name                                        Is Consumer?  Status
        hs22n56.gpfs.net                                 yes           AUDIT_CONS_OK
(08:53:52) hs22n56:/root #
```

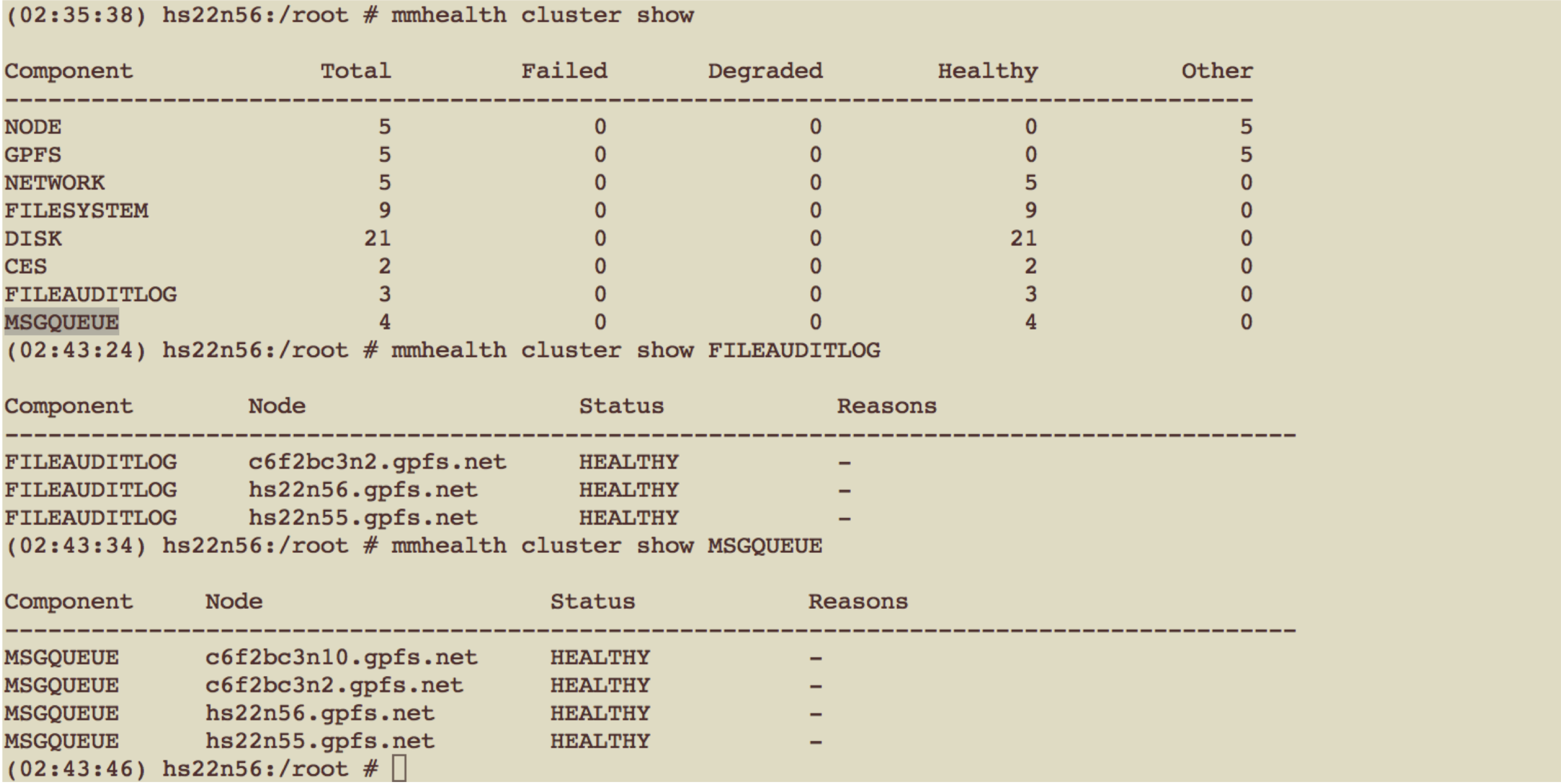
mmmsgqueue status



```
(08:59:09) hs22n56:/root # mmmsgqueue status
Node                                    Contains  Broker    Contains  Zookeeper
Name                                    Broker    Status    Zookeeper Status
c6f2bc3n10.gpfs.net                     no                  yes       good
c6f2bc3n2.gpfs.net                      yes       good      yes       good
hs22n55.gpfs.net                        yes       good      no
hs22n56.gpfs.net                        yes       good      no
hs22n95.gpfs.net                        no                  yes       good
(08:59:33) hs22n56:/root #
```

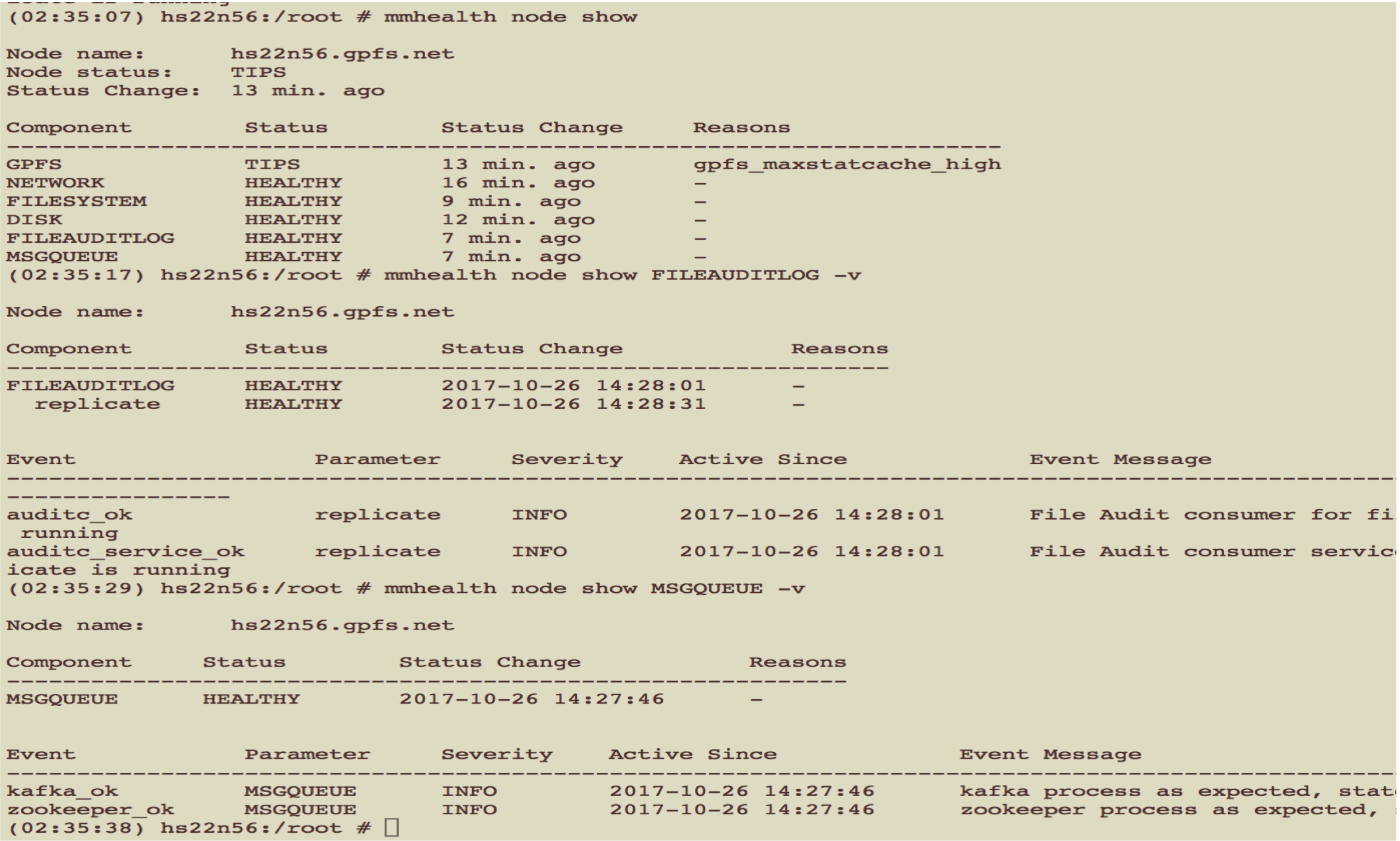
# mmhealth cluster monitoring

Periodic polling and event callback registration mechanism is used.
Possible lag in determining the health due to polling constraints.



```
(02:35:38) hs22n56:/root # mmhealth cluster show

Component             Total         Failed       Degraded        Healthy          Other
---------------------------------------------------------------------------------------
NODE                      5              0              0              0              5
GPFS                      5              0              0              0              5
NETWORK                   5              0              0              5              0
FILESYSTEM                9              0              0              9              0
DISK                     21              0              0             21              0
CES                       2              0              0              2              0
FILEAUDITLOG              3              0              0              3              0
MSGQUEUE                  4              0              0              4              0
(02:43:24) hs22n56:/root # mmhealth cluster show FILEAUDITLOG

Component        Node                   Status            Reasons
------------------------------------------------------------------------------------------
FILEAUDITLOG     c6f2bc3n2.gpfs.net     HEALTHY           -
FILEAUDITLOG     hs22n56.gpfs.net       HEALTHY           -
FILEAUDITLOG     hs22n55.gpfs.net       HEALTHY           -
(02:43:34) hs22n56:/root # mmhealth cluster show MSGQUEUE

Component     Node                    Status            Reasons
------------------------------------------------------------------------------------------
MSGQUEUE      c6f2bc3n10.gpfs.net     HEALTHY           -
MSGQUEUE      c6f2bc3n2.gpfs.net      HEALTHY           -
MSGQUEUE      hs22n56.gpfs.net        HEALTHY           -
MSGQUEUE      hs22n55.gpfs.net        HEALTHY           -
(02:43:46) hs22n56:/root #
```

# mmhealth node monitoring



```
(02:35:07) hs22n56:/root # mmhealth node show

Node name:      hs22n56.gpfs.net
Node status:    TIPS
Status Change:  13 min. ago

Component        Status        Status Change     Reasons
-----------------------------------------------------------------------
GPFS             TIPS          13 min. ago       gpfs_maxstatcache_high
NETWORK          HEALTHY       16 min. ago       -
FILESYSTEM       HEALTHY       9 min. ago        -
DISK             HEALTHY       12 min. ago       -
FILEAUDITLOG     HEALTHY       7 min. ago        -
MSGQUEUE         HEALTHY       7 min. ago        -
(02:35:17) hs22n56:/root # mmhealth node show FILEAUDITLOG -v

Node name:      hs22n56.gpfs.net

Component        Status        Status Change            Reasons
---------------------------------------------------------------
FILEAUDITLOG     HEALTHY       2017-10-26 14:28:01      -
  replicate      HEALTHY       2017-10-26 14:28:31      -


Event                 Parameter     Severity    Active Since              Event Message
---------------------------------------------------------------------------------------------------
----------------
auditc_ok             replicate     INFO        2017-10-26 14:28:01       File Audit consumer for fi
 running
auditc_service_ok     replicate     INFO        2017-10-26 14:28:01       File Audit consumer servic
icate is running
(02:35:29) hs22n56:/root # mmhealth node show MSGQUEUE -v

Node name:      hs22n56.gpfs.net

Component     Status        Status Change            Reasons
------------------------------------------------------------
MSGQUEUE      HEALTHY       2017-10-26 14:27:46      -


Event            Parameter     Severity    Active Since              Event Message
---------------------------------------------------------------------------------------------------
kafka_ok         MSGQUEUE      INFO        2017-10-26 14:27:46       kafka process as expected, stat
zookeeper_ok     MSGQUEUE      INFO        2017-10-26 14:27:46       zookeeper process as expected,
(02:35:38) hs22n56:/root # □
```

# Troubleshooting

## /var/adm/ras/mmmsgqueue.log

Contains information regarding the set up and configuration operations that take place that affect the message queue

Valid on any node containing a broker and/or zookeeper

## /var/adm/ras/mmaudit.log

Contains information regarding the set up and configuration operations that take place that affect the File Audit Logging

Valid on any node running the File Audit Logging command or location where the subcommand may be run (such as a consumer)

## /var/adm/ras/mmfs.log.latest

Daemon log, and contains entries when major message queue or File Audit Logging activity occurs.

## /var/log/messages (Redhat) or /var/log/syslog (Ubuntu)

Contains messages from Kafka components as well as the producer and consumers that are running on a node.

## Logs collected via gpfs.snap

# Where could this go in the future?

Antivirus

Take an action if something happens in a directory

TCT enhancements?!

# Running Spectrum Scale in a Vagrant Environment

# Replicate a repeatable Scale environment

- Yes, we have a VM
- Stemmed from work to do an IBM Scale GUI Lab
  - Spin a VM with an RedHat based OS and kickstart file
  - Use install toolkit and latest version of Scale!
  - Tied to VMWare workstation

```
sudo genisoimage -U -r -v -T -J -joliet-long -V "CentOS 7 x86_64" -volset "CentOS-7.4" -A "CentOS-7.4" -b isolinux/isolinux.bin -c isolinux/boot.cat -no-emul-boot -boot-load-size 4 -boot-info-table -eltorito-alt-boot -e images/efiboot.img -no-emul-boot -o ISONAME .
```

# What is vagrant and why??

HashiCorp Vagrant

Development Environments Made Easy

Build and manage virtual machines on the fly

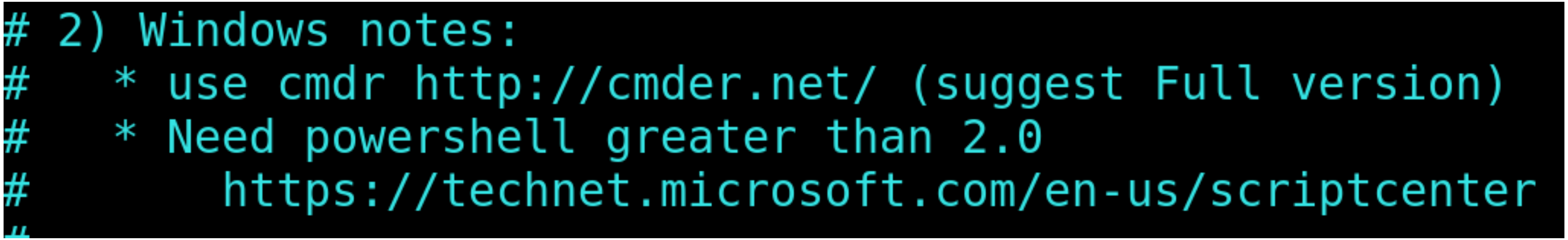
Plugins to configuration management utilities like:
ansible, chef , puppet, salt …

Scale runs anywhere but you need:

1. an OS installed
2. time and name resolution working
3. working network

# Can run on Windows, Linux and OS X

- Windows 7
  - needs a new powershell > 2

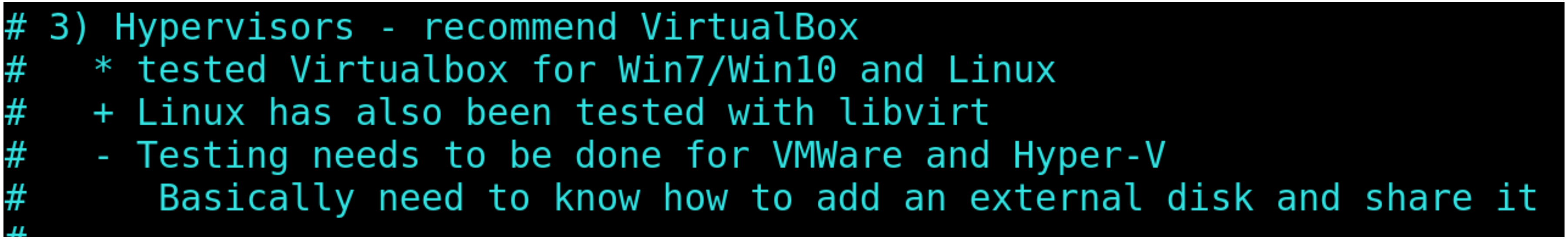


```
# 2) Windows notes:
#    * use cmdr http://cmder.net/ (suggest Full version)
#    * Need powershell greater than 2.0
#        https://technet.microsoft.com/en-us/scriptcenter
#
```

- Linux and OS X environments seem to be fine

# Tested Hypervisors

- Virtualbox
  - Runs the published Scale and Archive VMs today
  - Scale Vagrant files tested on Linux and Windows
- KVM/libvirt
  - No problems with RHEL7, can work with RHEL6



```
# 3) Hypervisors - recommend VirtualBox
#    * tested Virtualbox for Win7/Win10 and Linux
#    + Linux has also been tested with libvirt
#    - Testing needs to be done for VMWare and Hyper-V
#       Basically need to know how to add an external disk and share it
#
```

# Vagrant Mini-HowTo

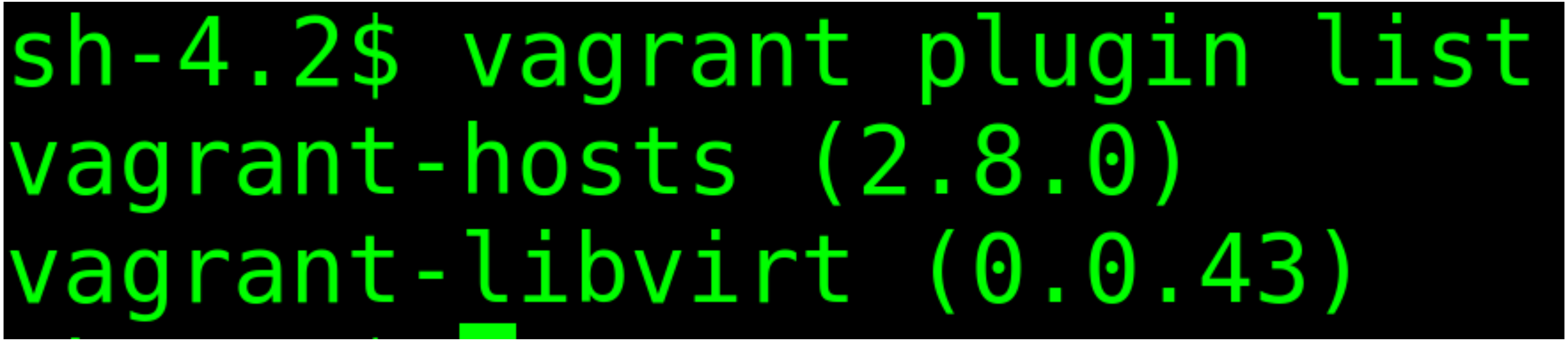
- Everything starts with vagrant
  - To ssh: vagrant ssh VMNAME
  - To start: vagrant up
  - To halt: vagrant halt
  - To reprovision: vagrant destroy

- The main definition is in a file called
  - Vagrantfile – ruby syntax
- To cry or start from scratch: rm –fr $HOME/.vagrant.d

# Setup plugins and add default OS to use

- Certain plugins help with
  - Hosts file update
    - vagrant plugin install vagrant-hosts
- if using Virtualbox, run
  - vagrant plugin install vagrant-vbguest

- else if using libvirt, run
  - vagrant plugin install \ vagrant-libvirt
  - Sometimes trouble starting libvirt  vms, so restart it
    - systemctl restart libvirtd

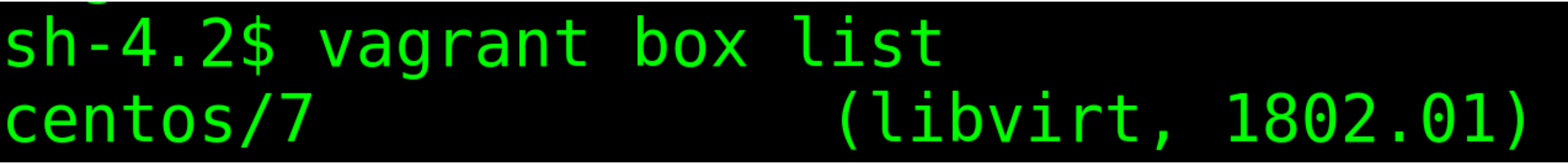


```
sh-4.2$ vagrant plugin list
vagrant-hosts (2.8.0)
vagrant-libvirt (0.0.43)
```

# Setup a local box to work from

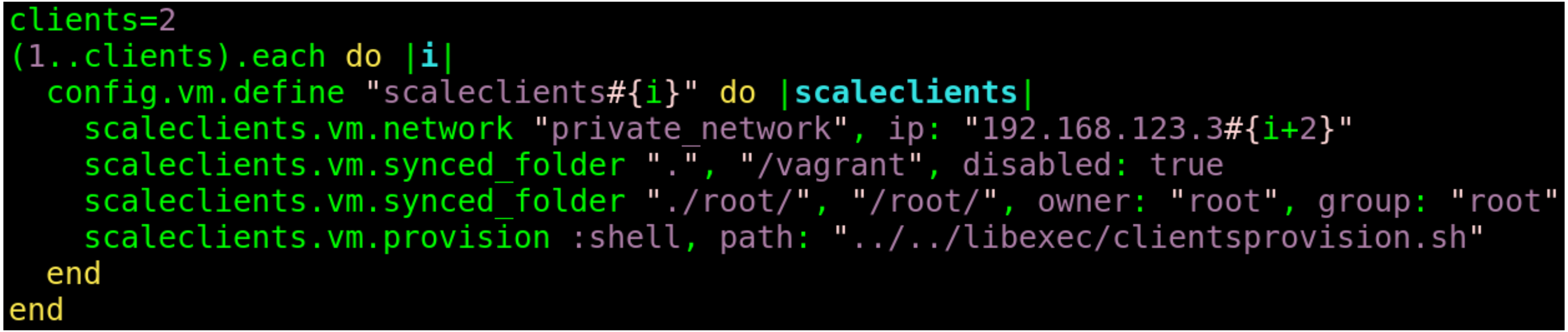
- Select your hypervisor (recommend virtualbox or libvirt)
  - Add centos/7 vagrant box
    - vagrant box add centos/7
    - vagrant box list

- You should see centos/7 listed

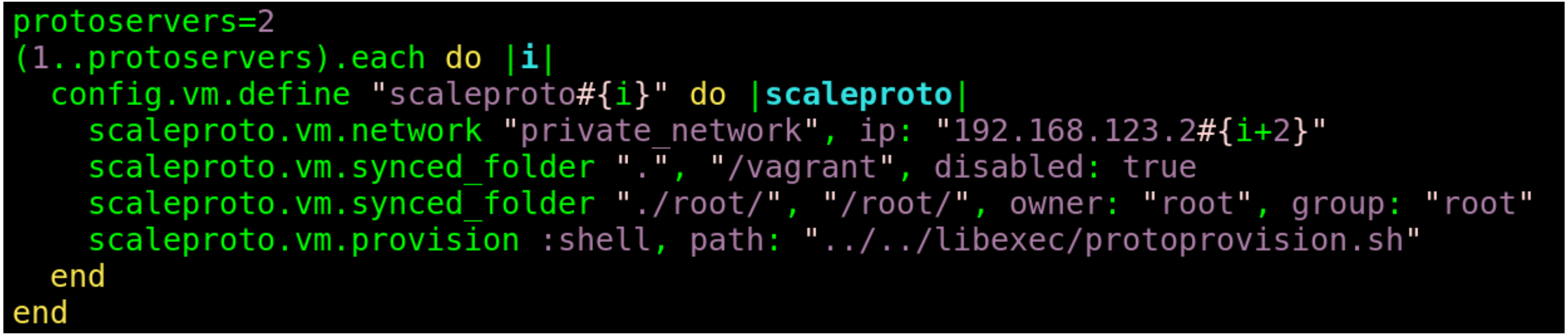


```
sh-4.2$ vagrant box list
centos/7                    (libvirt, 1802.01)
```

# Vagrant file - Clients and Protocol nodes



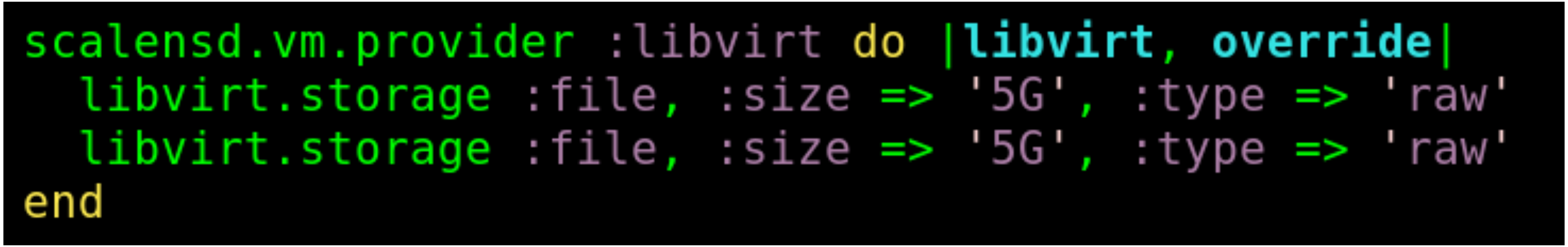
```
clients=2
(1..clients).each do |i|
  config.vm.define "scaleclients#{i}" do |scaleclients|
    scaleclients.vm.network "private_network", ip: "192.168.123.3#{i+2}"
    scaleclients.vm.synced_folder ".", "/vagrant", disabled: true
    scaleclients.vm.synced_folder "./root/", "/root/", owner: "root", group: "root"
    scaleclients.vm.provision :shell, path: "../../libexec/clientsprovision.sh"
  end
end
```



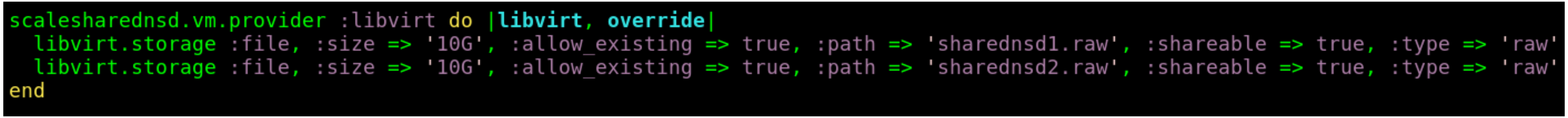
```
protoservers=2
(1..protoservers).each do |i|
  config.vm.define "scaleproto#{i}" do |scaleproto|
    scaleproto.vm.network "private_network", ip: "192.168.123.2#{i+2}"
    scaleproto.vm.synced_folder ".", "/vagrant", disabled: true
    scaleproto.vm.synced_folder "./root/", "/root/", owner: "root", group: "root"
    scaleproto.vm.provision :shell, path: "../../libexec/protoprovision.sh"
  end
end
```

Vagrantfile is Ruby code

# Vagrant file – libvirt SNC vs Shared

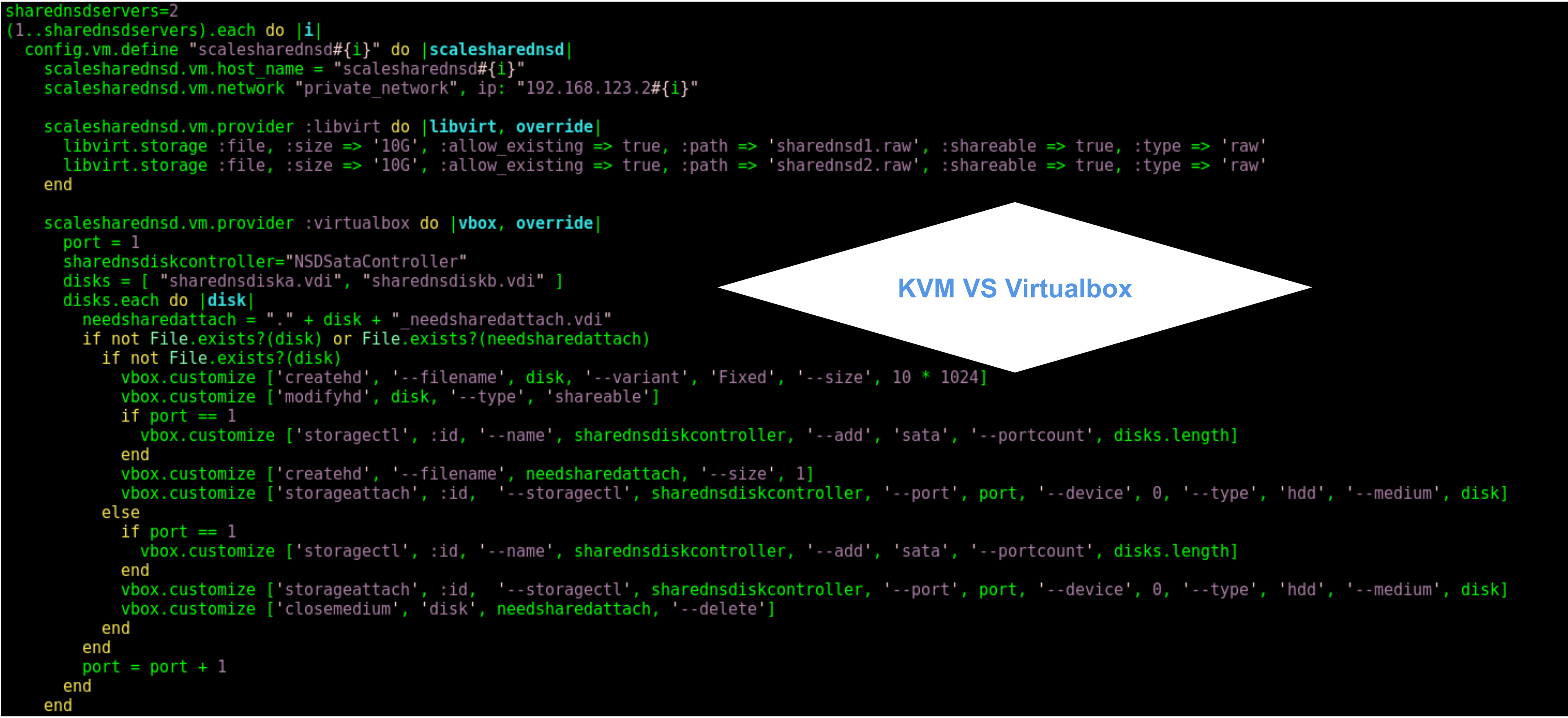


```
scalensd.vm.provider :libvirt do |libvirt, override|
  libvirt.storage :file, :size => '5G', :type => 'raw'
  libvirt.storage :file, :size => '5G', :type => 'raw'
end
```



```
scalesharednsd.vm.provider :libvirt do |libvirt, override|
  libvirt.storage :file, :size => '10G', :allow_existing => true, :path => 'sharednsd1.raw', :shareable => true, :type => 'raw'
  libvirt.storage :file, :size => '10G', :allow_existing => true, :path => 'sharednsd2.raw', :shareable => true, :type => 'raw'
end
```

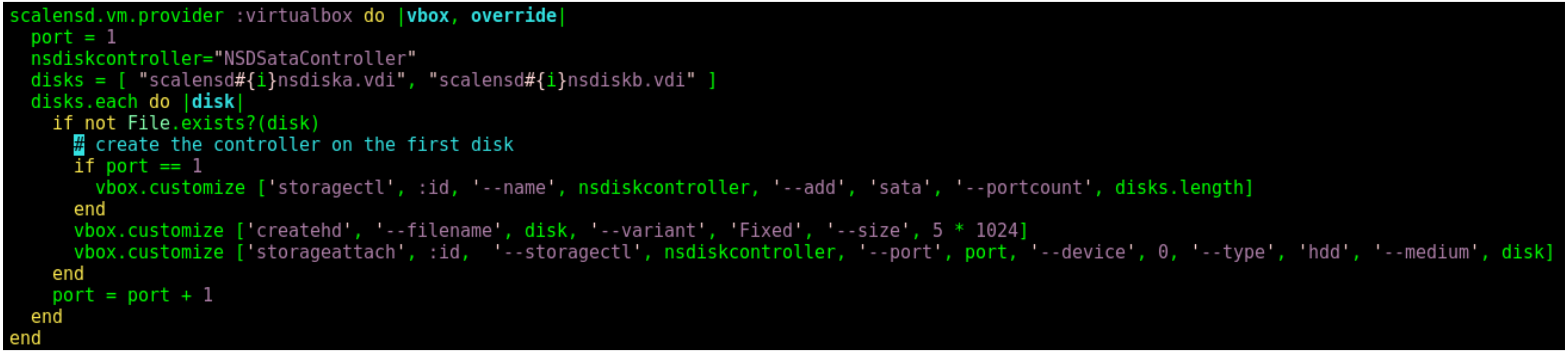
# Shared libvirt vs Virtualbox



```
sharednsdservers=2
(1..sharednsdservers).each do |i|
  config.vm.define "scalesharednsd#{i}" do |scalesharednsd|
    scalesharednsd.vm.host_name = "scalesharednsd#{i}"
    scalesharednsd.vm.network "private_network", ip: "192.168.123.2#{i}"

    scalesharednsd.vm.provider :libvirt do |libvirt, override|
      libvirt.storage :file, :size => '10G', :allow_existing => true, :path => 'sharednsd1.raw', :shareable => true, :type => 'raw'
      libvirt.storage :file, :size => '10G', :allow_existing => true, :path => 'sharednsd2.raw', :shareable => true, :type => 'raw'
    end

    scalesharednsd.vm.provider :virtualbox do |vbox, override|
      port = 1
      sharednsdiskcontroller="NSDSataController"
      disks = [ "sharednsdiska.vdi", "sharednsdiskb.vdi" ]
      disks.each do |disk|
        needsharedattach = "." + disk + "_needsharedattach.vdi"
        if not File.exists?(disk) or File.exists?(needsharedattach)
          if not File.exists?(disk)
            vbox.customize ['createhd', '--filename', disk, '--variant', 'Fixed', '--size', 10 * 1024]
            vbox.customize ['modifyhd', disk, '--type', 'shareable']
            if port == 1
              vbox.customize ['storagectl', :id, '--name', sharednsdiskcontroller, '--add', 'sata', '--portcount', disks.length]
            end
            vbox.customize ['createhd', '--filename', needsharedattach, '--size', 1]
            vbox.customize ['storageattach', :id,  '--storagectl', sharednsdiskcontroller, '--port', port, '--device', 0, '--type', 'hdd', '--medium', disk]
          else
            if port == 1
              vbox.customize ['storagectl', :id, '--name', sharednsdiskcontroller, '--add', 'sata', '--portcount', disks.length]
            end
            vbox.customize ['storageattach', :id,  '--storagectl', sharednsdiskcontroller, '--port', port, '--device', 0, '--type', 'hdd', '--medium', disk]
            vbox.customize ['closemedium', 'disk', needsharedattach, '--delete']
          end
        end
        port = port + 1
      end
    end
```

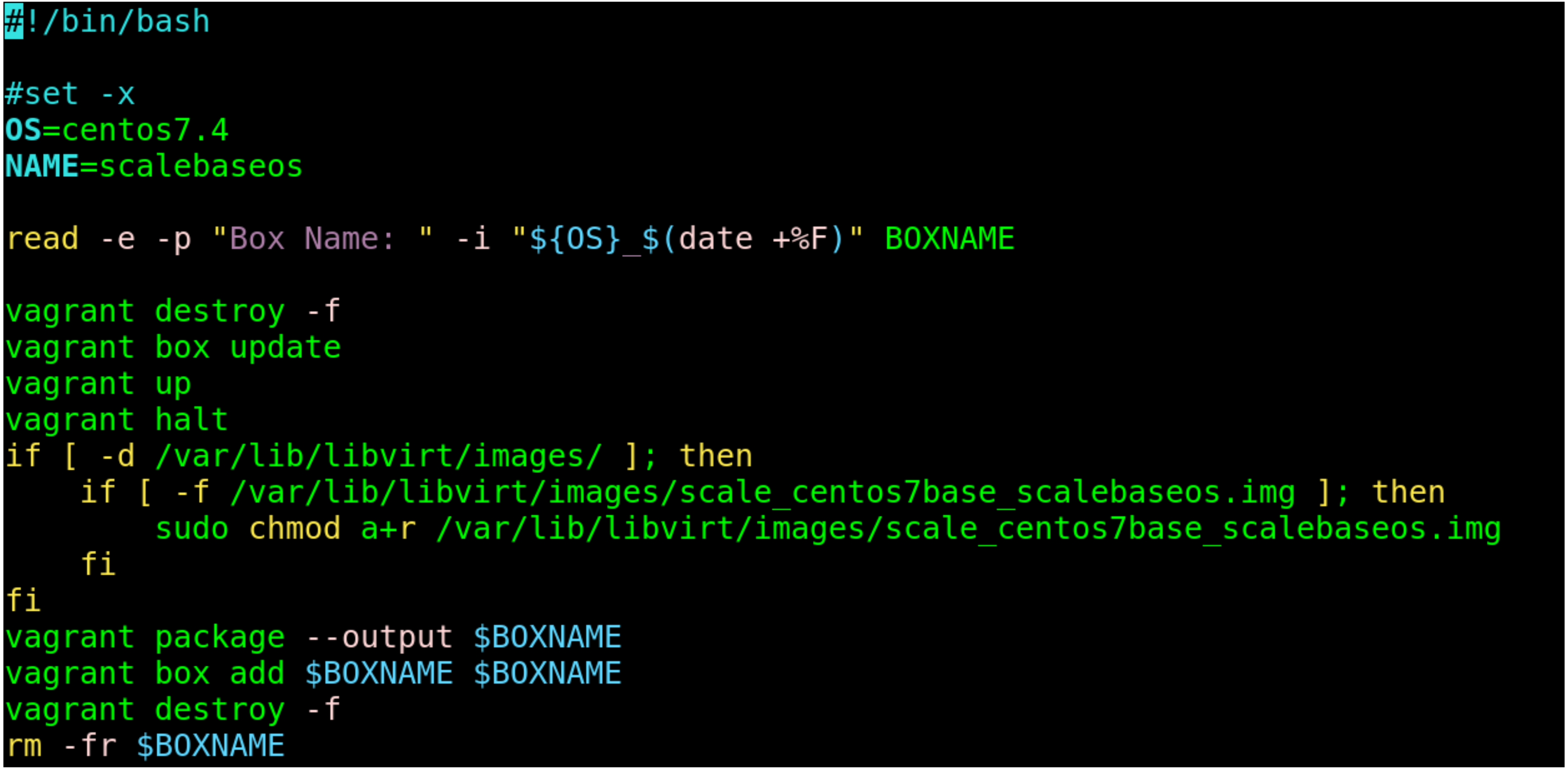
**KVM VS Virtualbox**

# Virtualbox SNC



```
scalensd.vm.provider :virtualbox do |vbox, override|
  port = 1
  nsdiskcontroller="NSDSataController"
  disks = [ "scalensd#{i}nsdiska.vdi", "scalensd#{i}nsdiskb.vdi" ]
  disks.each do |disk|
    if not File.exists?(disk)
      # create the controller on the first disk
      if port == 1
        vbox.customize ['storagectl', :id, '--name', nsdiskcontroller, '--add', 'sata', '--portcount', disks.length]
      end
      vbox.customize ['createhd', '--filename', disk, '--variant', 'Fixed', '--size', 5 * 1024]
      vbox.customize ['storageattach', :id,  '--storagectl', nsdiskcontroller, '--port', port, '--device', 0, '--type', 'hdd', '--medium', disk]
    end
    port = port + 1
  end
end
```

# Install a base box so you don’t have to pull updates



```
#!/bin/bash

#set -x
OS=centos7.4
NAME=scalebaseos

read -e -p "Box Name: " -i "${OS}_$(date +%F)" BOXNAME

vagrant destroy -f
vagrant box update
vagrant up
vagrant halt
if [ -d /var/lib/libvirt/images/ ]; then
    if [ -f /var/lib/libvirt/images/scale_centos7base_scalebaseos.img ]; then
        sudo chmod a+r /var/lib/libvirt/images/scale_centos7base_scalebaseos.img
    fi
fi
vagrant package --output $BOXNAME
vagrant box add $BOXNAME $BOXNAME
vagrant destroy -f
rm -fr $BOXNAME
```

# Provision Scripts

Can call out ansible here

Currently calling a shell script

Points to a SCALESOURCE tree and extracts data

# Provision Scripts

Can call out ansible here

Currently calling a shell script

Points to a SCALESOURCE tree and extracts data

# Let’s demo

Coming soon GIT tree public

- vagrantbuild – sample Vagrant files for Scale
- cssdeployenv – install toolkit and runbooks

Integrate with Ansible form others

ibm.com/storage

# Thank You.

## IBM Storage & SDI