

Special Thanks to development team

Deepavali M Baghwat
Constantine Arnold
Bill Olson
Marc Kaplan



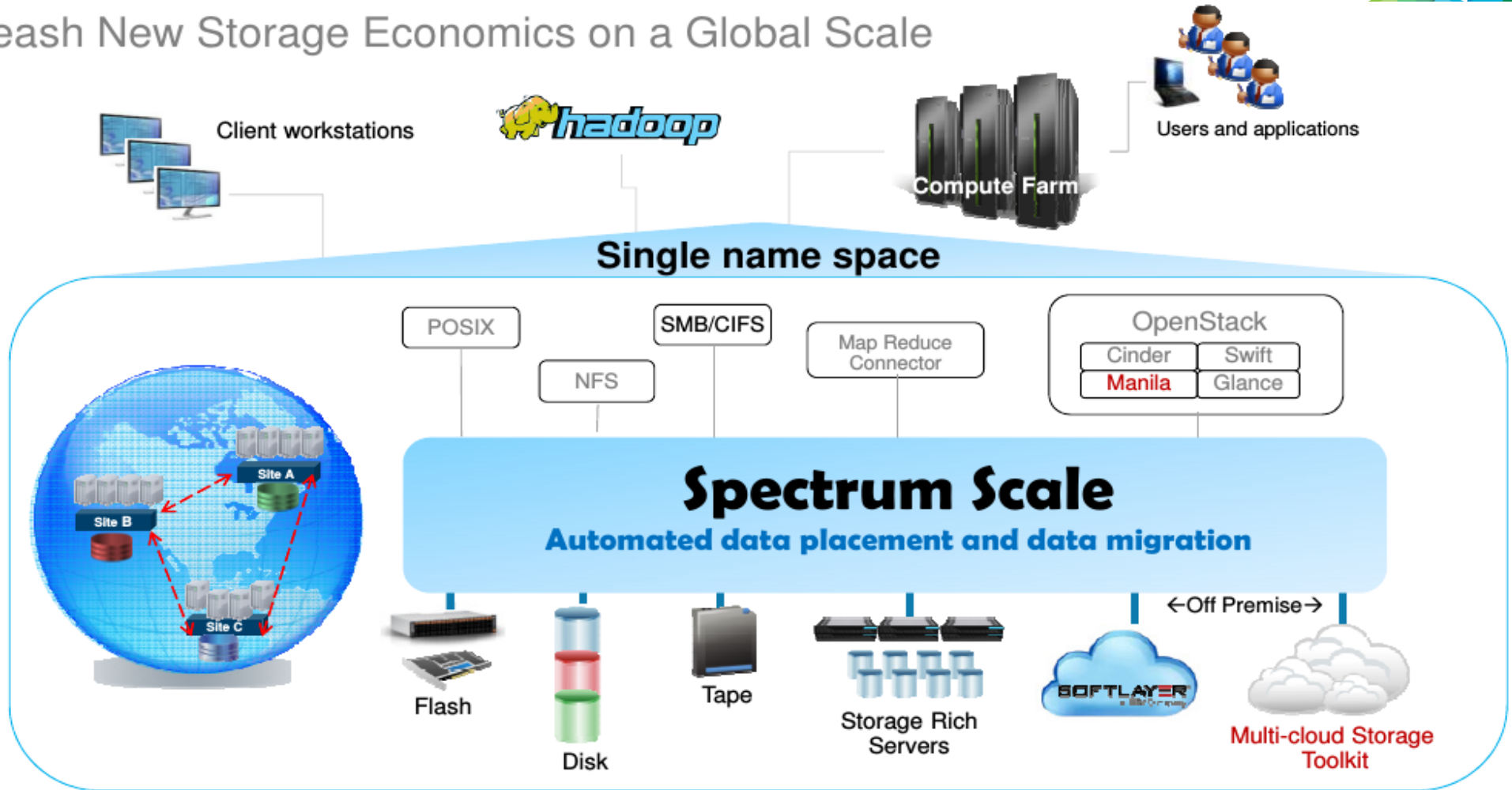
olaf.weiser@de.ibm.com

IBM Deutschland
SpectrumScale voluntary user

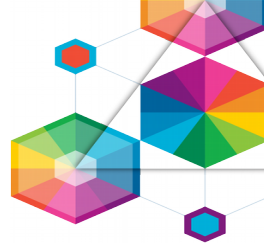
Why watchfolder



Unleash New Storage Economics on a Global Scale

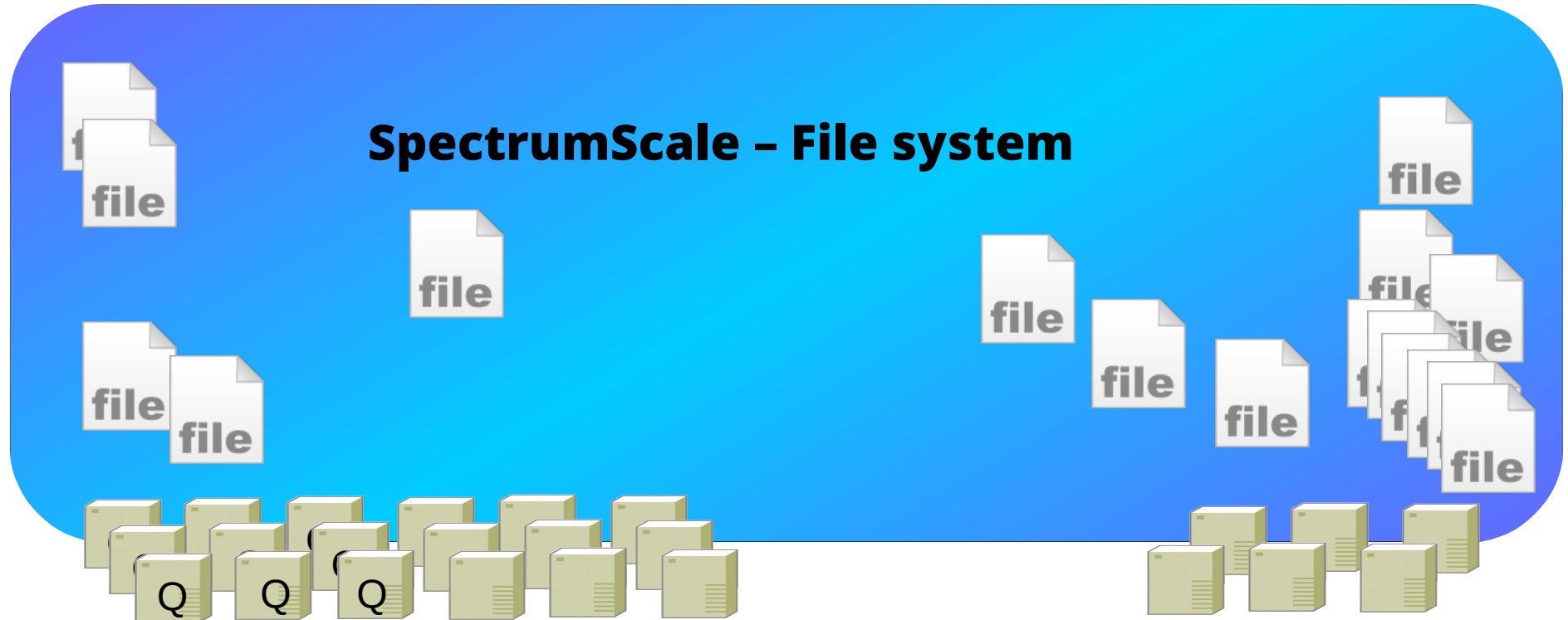


Why watchfolder

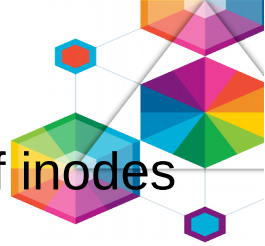


- higher data rates → more data → number of files/objects
- with higher #files in a file system - a POSIX *find* may kick million IO's

– watch folder will help to save a lot of resources for scanning MD



Why watchfolder



- high efficient policy engine helps to find files, but need to scan millions of inodes
- policy engine can generate mio of IOs

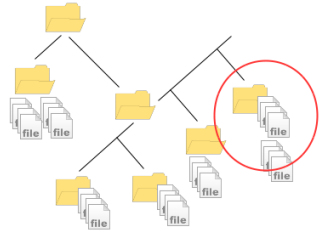
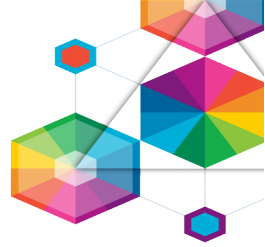
– watch folder will help to save a lot of resources for scanning MD

SpectrumScale - File system

Number of used inodes:	405627348
Number of free inodes:	6250234839
Number of allocated inodes:	6655862187
Maximum number of inodes:	6655862187

The diagram features a blue rounded rectangle containing the title 'SpectrumScale - File system' and a central white box with a dashed border containing the inode statistics table. Surrounding the central box are various icons: 'file' icons, 'TEXT Document' icons, and server rack icons at the bottom. The server racks are arranged in two groups, one on the left and one on the right, with some racks having a 'Q' on them.

Watchfolder – inotify stretched over cluster



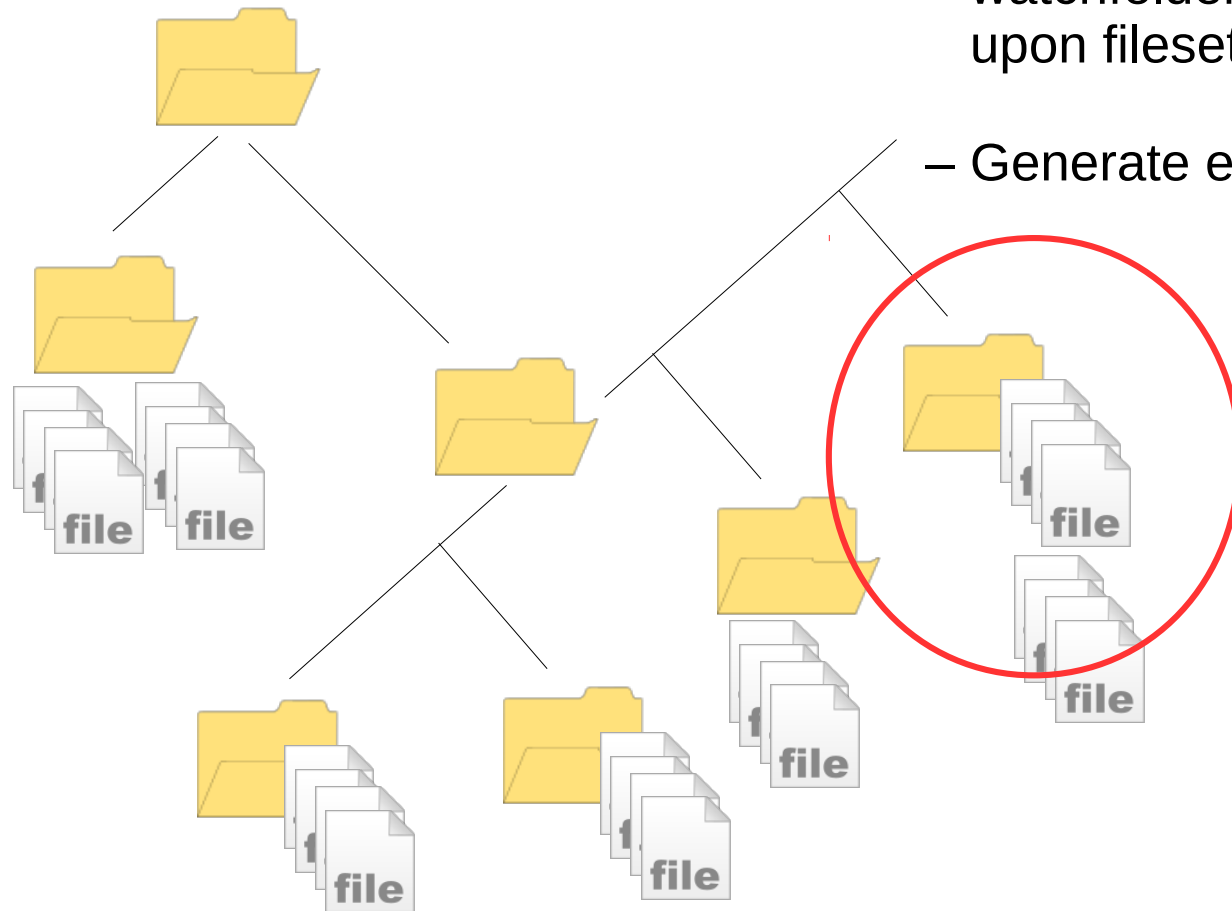
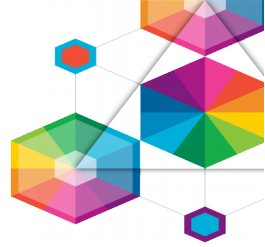
Challenges from inotify:

- inotify only knows „local events“, not sufficient for parallel File systems
- distributed access means distributed events

Solution with SpectrumScale :

- watchfolder is aware of all events from all nodes, by using central queue (kafka)
- SpectrumScale ships with kafka / watchfolder enablement

Watchfolder – overview



– watchfolder can be enabled upon filesets or directories

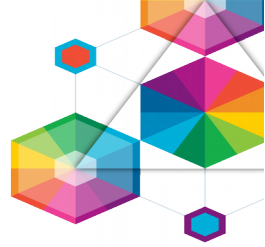
– Generate event on access for

... open, read, write ...

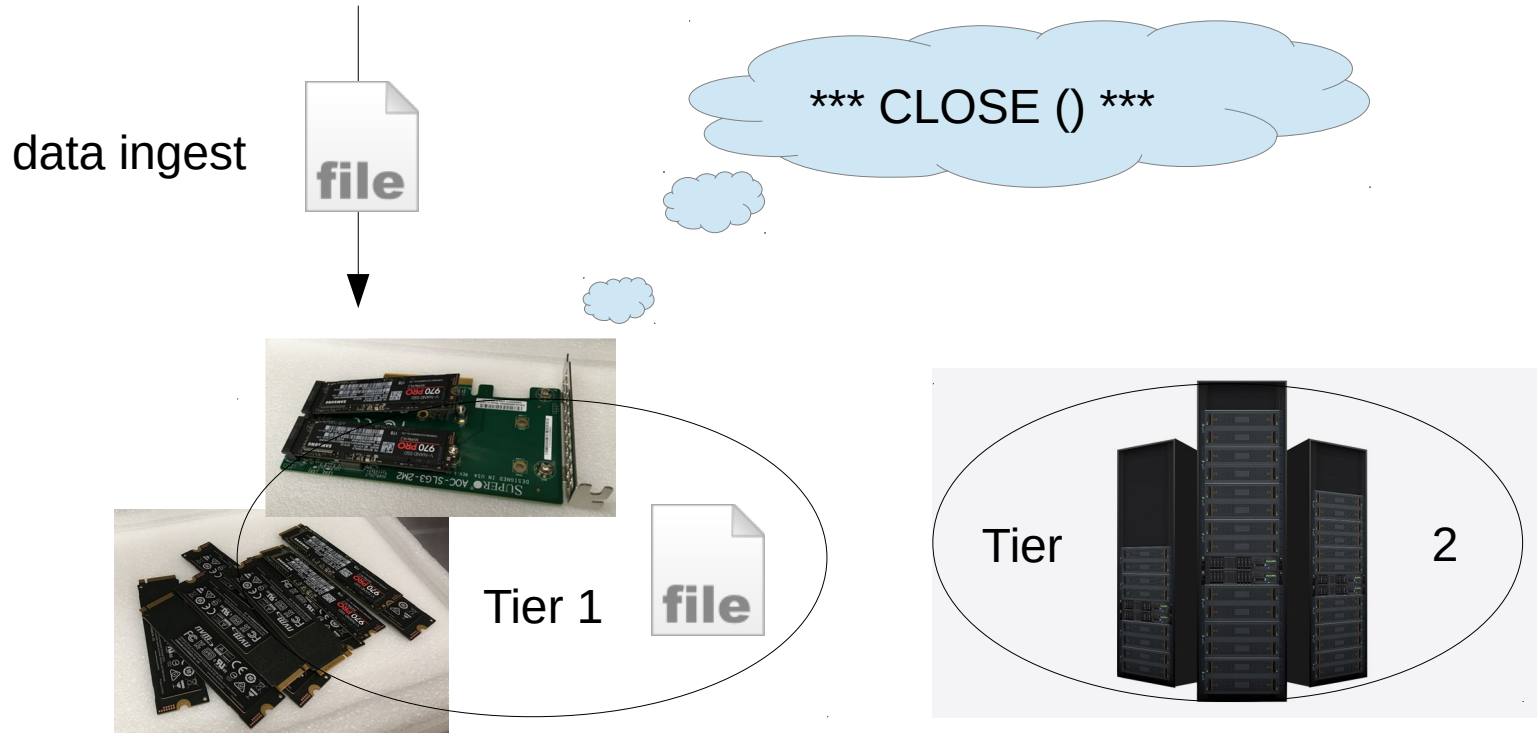
– flexible according to Workload

– uses policy engine / LWE

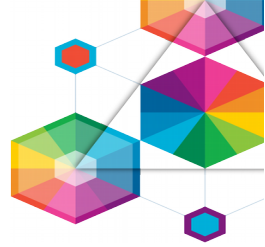
Why watchfolder



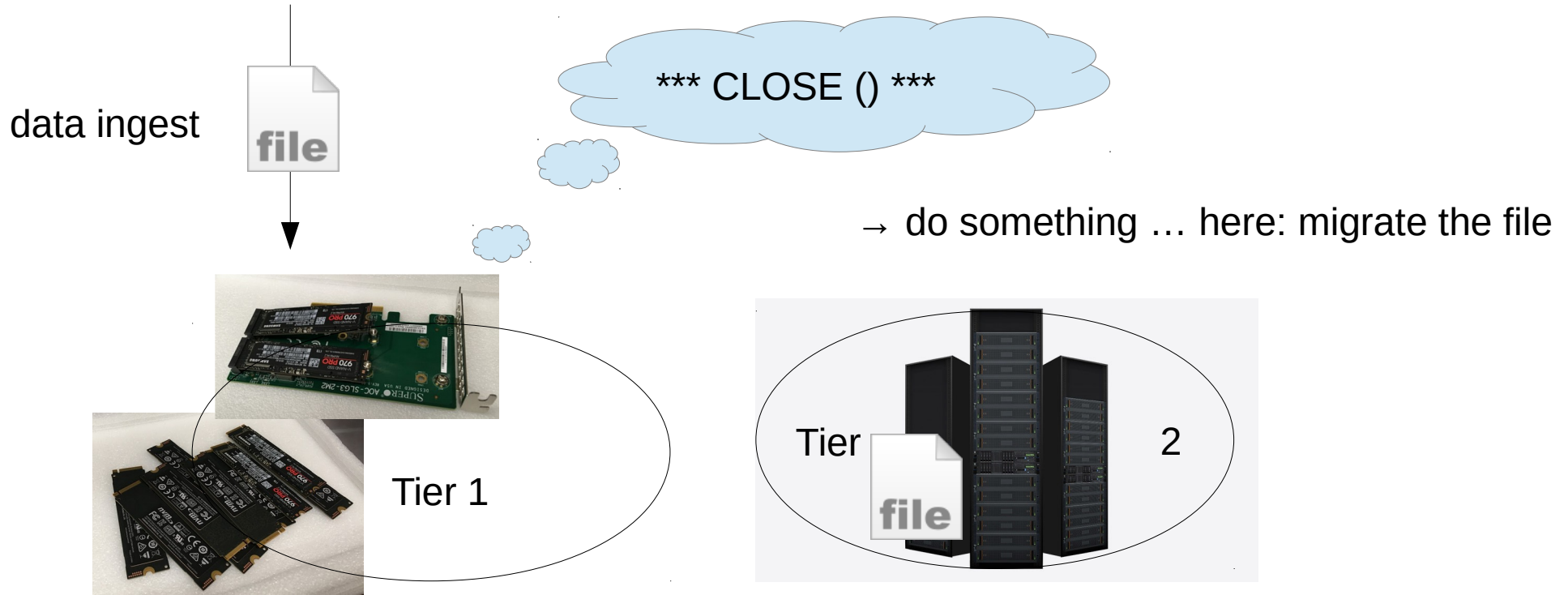
- create your own workflow
- manage placement
- manage post processing



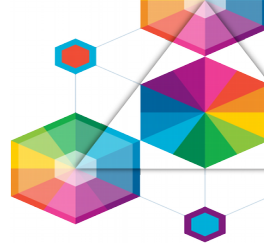
Why watchfolder



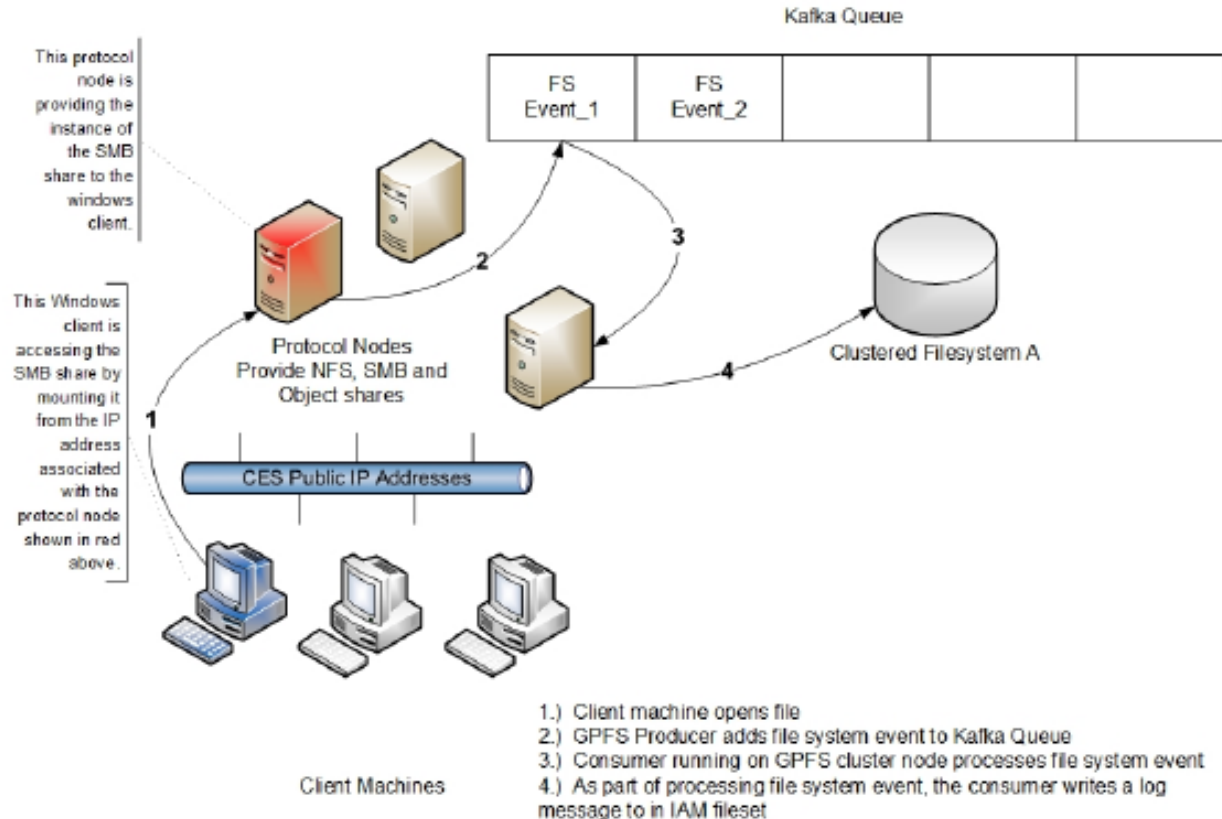
- create your own workflow
- manage placement
- manage post processing



Watchfolder – short introduction



- WF was introduced with FAL
- WF rely on central msg q



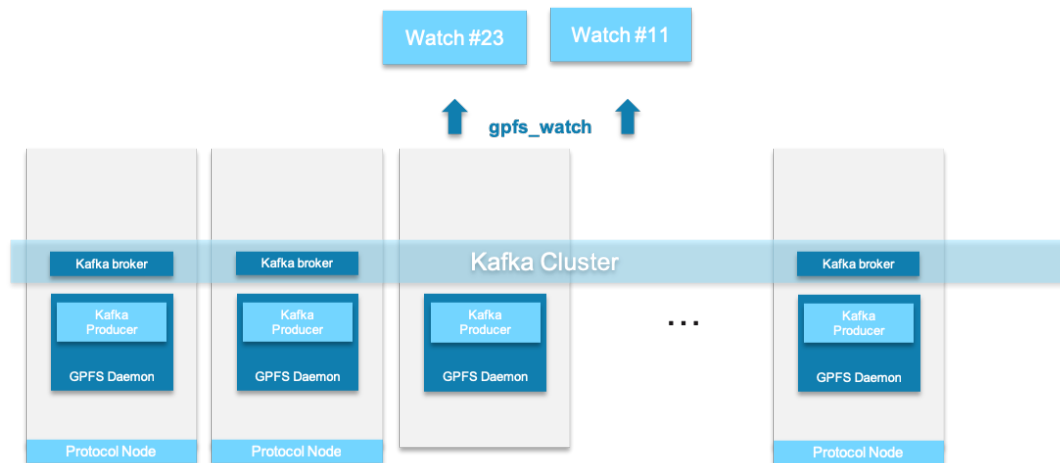
SpectrumScale – comes with Kafka



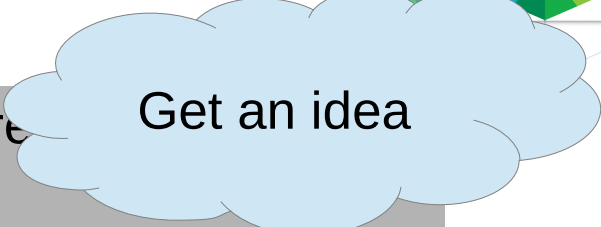
```
[root@tlinc04 ~]# mmmsgqueue status
```

Node Name	Contains Broker	Broker Status	Contains Zookeeper	Zookeeper Status
tlinc04.mainz.de.ibm.com	yes	good	yes	good
tlinc05.mainz.de.ibm.com	yes	good	yes	good
tlinc06.mainz.de.ibm.com	yes	BAD	no	

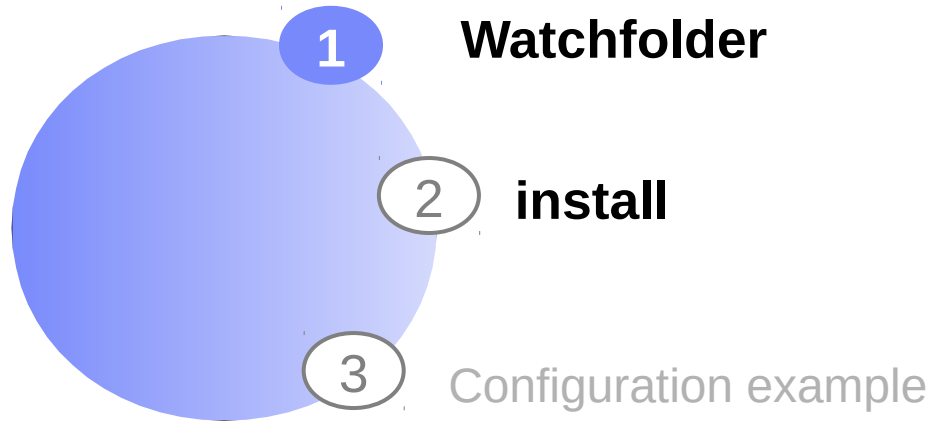
```
[root@tlinc04 ~]#
```



Watchfolder – some little examples / events



IO Operation Command	Events Triggered <i>Inotify</i>	Events Triggered <i>gpfs watch</i>
cat	OPEN, CLOSE_NOWRITE	OPEN, CLOSE_NOWRITE, ACCESS
>>	OPEN, MODIFY, CLOSE_WRITE	OPEN, ACCESS, CLOSE_WRITE
> (Output file doesn't exist)	CREATE, OPEN, MODIFY, CLOSE_WRITE	CREATE, ATTRIB, ACCESS, OPEN, ACCESS, CLOSE_WRITE
> (Output file already exists)	MODIFY, OPEN, MODIFY, CLOSE_WRITE	ACCESS, OPEN, ACCESS, CLOSE_WRITE
cp (source outside watched dir)	CREATE, OPEN, MODIFY, CLOSE_WRITE	CREATE, ATTRIB, OPEN, ACCESS, CLOSE_WRITE





Installation of watch folder with Spectrum Scale Toolkit

```
# ./spectrumscale watchfolder enable
[ INFO ] Enabling watch folder in the cluster configuration file.
[ INFO ] The watch folder feature provides a new method of monitoring and performing actions on file system events in IBM Spectrum Scale. To complete the watch folder configuration please contact your IBM Client Technical Specialist (or send an email to scale@us.ibm.com) to review your use case and receive instructions to enable the feature in your environment.
[ INFO ] if you do not want to enable watch folder, you can run: ./spectrumscale watchfolder disable .
[ INFO ] Tip :If watch folder is enabled for the cluster, the message queue will be enabled. If you have at least 3 protocol nodes defined, you do not need to use the '-b' flag to designate a node as a broker node. Otherwise, you need to specify at least 3 nodes to have the Kafka packages installed on and act as broker nodes for the message queue. Use the following command to designate a node as a broker node: ./spectrumscale node add node_name -b .
[root@hulkprt001 installer] 2018-08-31 21:54:16
# ./spectrumscale watchfolder list
[ INFO ] Watch folder is Enabled.
[ INFO ] User has not defined any broker nodes in the configuration. Use 'spectrumscale node add node_name -b' to add node as a broker node.
```

```
# ./spectrumscale node add hulkprt002 -b
[ INFO ] Setting hulkprt002.tuc.stglabs.ibm.com as a Kafka broker node.
[ INFO ] Configuration updated.
[root@hulkprt001 installer] 2018-09-03 21:06:34
# ./spectrumscale node add hulkprt001 -b
[ INFO ] Setting hulkprt001.tuc.stglabs.ibm.com as a Kafka broker node.
[ INFO ] Configuration updated.
[root@hulkprt001 installer] 2018-09-03 21:06:40
# ./spectrumscale node add hulkprt003 -b
[ INFO ] Setting hulkprt003.tuc.stglabs.ibm.com as a Kafka broker node.
[ INFO ] Configuration updated.
```

[INFO] GPFS	Admin	Quorum	Manager	NSD	Protocol	GUI	Callhome	Perf Mon	FAL/WF	OS	Arch
[INFO] Node	Node	Node	Node	Server	Node	Server	Server	Collector	Broker		
[INFO] hulknsd001.tuc.stglabs.ibm.com			X	X						rhel7	ppc64le
[INFO] hulknsd002.tuc.stglabs.ibm.com			X	X						rhel7	ppc64le
[INFO] hulkprt001.tuc.stglabs.ibm.com	X		X		X	X		X	X	rhel7	ppc64le
[INFO] hulkprt002.tuc.stglabs.ibm.com	X				X	X		X	X	rhel7	ppc64le
[INFO] hulkprt003.tuc.stglabs.ibm.com					X				X	rhel7	ppc64le
[INFO] hulkprt004.tuc.stglabs.ibm.com					X					rhel7	ppc64le
[INFO] hulkprt005.tuc.stglabs.ibm.com					X					rhel7	ppc64le
[INFO] hulkprt006.tuc.stglabs.ibm.com		X			X					rhel7	ppc64le
[INFO] hulkprt007.tuc.stglabs.ibm.com		X			X					rhel7	ppc64le
[INFO] hulkprt008.tuc.stglabs.ibm.com		X			X		X			rhel7	ppc64le

Howto get WF



1. Go to `/usr/lpp/mmfs/5.0.2.0/gpfs_rpms/rhel7` to install the packages.
2. Install the following Kafka packages using the **yum install** command:

```
gpfs.librdkafka-5.0.2-*.x86_64.rpm  
gpfs.kafka-5.0.2-*.x86_64.rpm
```



3. Run the following command to set up the message queue:

```
mmsgqueue enable -N <node1, node2, node3>
```



Note

You must pass in three or more nodes.

4. Check the status of the message queue with the following command:

```
mmsgqueue status
```



5. To test watch folder, go to `/usr/lpp/mmfs/samples/util` and locate the `tswf.C` program.

Note

This is a sample program that IBM® has provided to set up watches on a single file system.



Requirements for WatchFolder

→ 5.0.2 or later

→ watch folder can be **upgraded** to IBM Spectrum Scale 20.01.

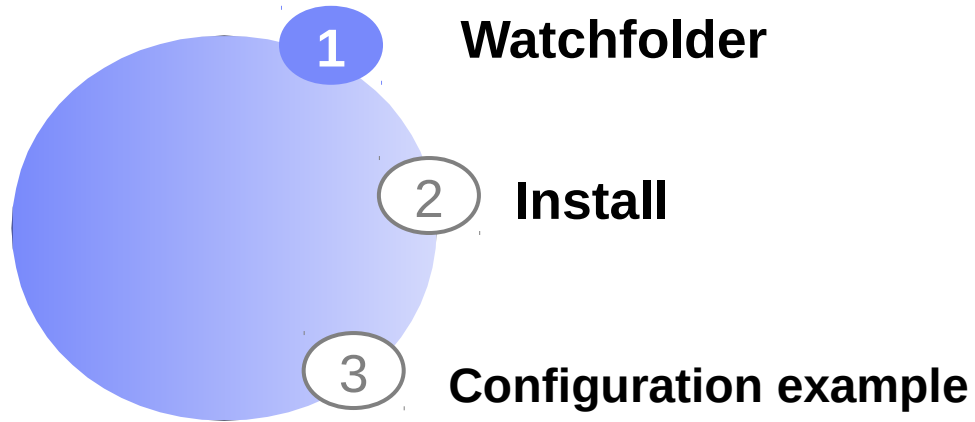
Miscellaneous requirements

Scale Advanced Edition

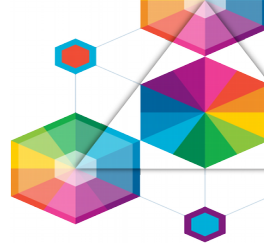
To use watch folder, your environment must be managed by IBM Technical Specialist (or equivalent) and receive instructions to enable the feature in your environment.

The screenshot shows a web browser window with the URL https://www.ibm.com/support/knowledgecenter/en/STXKQY_5.0.2/com.ibm.spectrum.sc. The page title is "Requirements for watch folder". The page content includes:

- RPM and package requirements**
 - Every node that is capable of hosting any combination of brokers, ZooKeepers, producers, and consumers must have the following packages installed:
 - GPFS™ Java™ (gpfs.java rpm/package)
 - For RHEL, the librdkafka package requires the openssl-devel and cyrus-sasl-devel packages
 - For Ubuntu, the librdkafka package requires the libssl-dev and libsasl2-dev packages
 - librdkafka (gpfs.librdkafka rpm/package)
 - Kafka (gpfs.kafka rpm/package)
- OS and hardware requirements**
 - RHEL 7.x on x86, RHEL 7.x on Power8 Little Endian, or Ubuntu 16.04/16.04.01 on x86.
 - Linux Kernel on all platforms must be greater than or equal to 31000123.
 - Minimum of three Linux quorum nodes running on approved OS and hardware (ZooKeepers).
 - Minimum of three nodes to act as message queue servers (brokers) running on approved OS and hardware.



How to use watch folder (5.0.3) - nice example



How to use watch folder (5.0.3) - step 1 / your own Kafka



Quickstart

This tutorial assumes you are starting fresh and have no existing Kafka or ZooKeeper data. Since Kafka console scripts are different for Unix-based and Windows platforms, on Windows platforms use `bin\windows\` instead of `bin/`, and change the script extension to `.bat`.

Step 1: Download the code

[Download](#) the 2.2.0 release and un-tar it.

```
1 > tar -xzf kafka_2.12-2.2.0.tgz
2 > cd kafka_2.12-2.2.0
```

Step 2: Start the server

Kafka uses [ZooKeeper](#) so you need to first start a ZooKeeper server if you don't already have one. You can use the convenience script packaged with kafka to get a quick-and-dirty single-node ZooKeeper instance.

```
1 > bin/zookeeper-server-start.sh config/zookeeper.properties
2 [2013-04-22 15:01:37,495] INFO Reading configuration from: config/zookeeper.properties (org.apache.zookeeper.s
3 ...
```

Now start the Kafka server:

```
1 > bin/kafka-server-start.sh config/server.properties
2 [2013-04-22 15:01:47,028] INFO Verifying properties (kafka.utils.VerifiableProperties)
3 [2013-04-22 15:01:47,051] INFO Property socket.send.buffer.bytes is overridden to 1048576 (kafka.utils.Verifia
4 ...
```

Step 3: Create a topic

Let's create a topic named "test" with a single partition and only one replica:

```
1 > bin/kafka-topics.sh --create --bootstrap-server localhost:9092 --replication-factor 1 --partitions 1 --topic
```

We can now see that topic if we run the list topic command:

```
1 > bin/kafka-topics.sh --list --bootstrap-server localhost:9092
```

<https://kafka.apache.org/quickstart>

How to use watch folder (5.0.3) - step 1 / you own Kafka

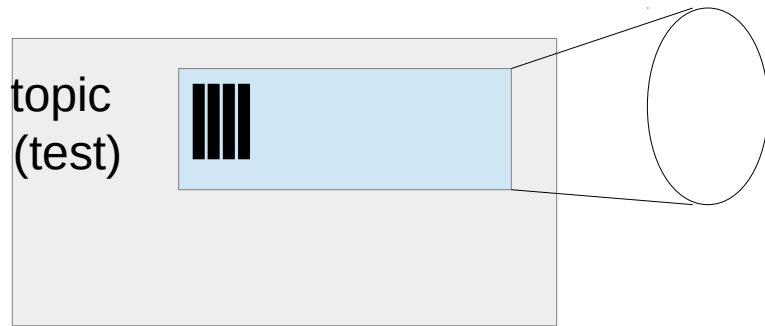


.. have your own Kafka ready ..

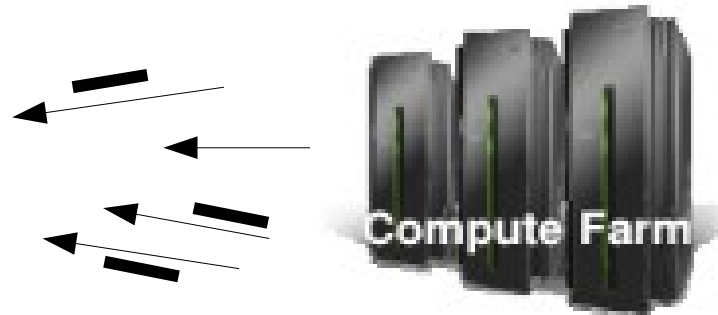


<https://kafka.apache.org/quickstart>

.. start Zookeeper and create a topic ..



KAFKA cluster



How to use watch folder (5.0.3) – step 2 configure mmsqueue



```
mmsgqueue enable -N tlinc04,tlinc05,tlinc06
```

```
[...]
```

```
[root@tlinc04 ~]# mmsgqueue status -q
```

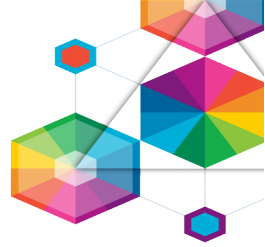
Node Name	Contains Broker	Broker Status	Contains Zookeeper	Zookeeper Status
tlinc04.mainz.de.ibm.com	yes	good	yes	good
tlinc05.mainz.de.ibm.com	yes	good	yes	good
tlinc06.mainz.de.ibm.com	yes	good	yes	good

```
[root@tlinc04 ~]#
```

.. it enables the GPFS embedded Kafka msg queue ..

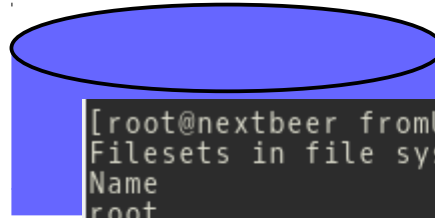
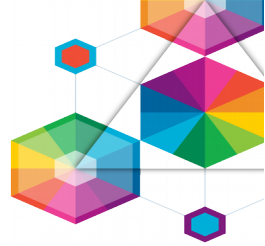
.. and and comes with Zookeeper

How to use watch folder (5.0.3) - step 3 nice example



Node	Daemon node name	IP address	Admin node name	Designation
1	nextbeer	10.11.11.112	nextbeer	quorum-manager
2	nextcan1.beer.loc	10.11.11.201	nextcan1.beer.loc	quorum
3	nextcan2.beer.loc	10.11.11.202	nextcan2.beer.loc	quorum

How to use watch folder (5.0.3) - step 3 nice example

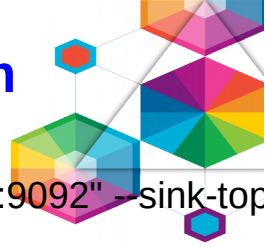


```
[root@nextbeer fromUK]# mmlsfileset beer
Filesets in file system 'beer':
Name          Status      Path
root          Linked     /gpfs/beer
test          Linked     /gpfs/beer/test
fromGermany   Linked     /gpfs/beer/fromGermany
fromFinland   Linked     /gpfs/beer/fromFinland
fromIreland   Linked     /gpfs/beer/fromIreland
fromUK        Linked     /gpfs/beer/fromUK
heavenOnEarth Linked     /gpfs/beer/heavenOnEarth
```



Node	Daemon node name	IP address	Admin node name	Designation
1	nextbeer	10.11.11.112	nextbeer	quorum-manager
2	nextcan1.beer.loc	10.11.11.201	nextcan1.beer.loc	quorum
3	nextcan2.beer.loc	10.11.11.202	nextcan2.beer.loc	quorum

How to use watch folder (5.0.3) - step 3 configure watch



```
mmwatch beer enable --fileset fromUK --events IN_CREATE --event-handler kafkasink --sink-brokers "kafka1:9092" --sink-top
```

```
[root@nextbeer fromUK]# mmwatch beer enable --fileset fromUK --events IN_CREATE --event-handler kafkasink --sink-brokers "kaf
[I] Beginning enablement of Clustered Watch with newly created watch ID: CLW1556875916
[I] Successfully created clustered watch consumer node class kafkaClWatchConsumerServers
[I] Verifying MsgQueue nodes meet minimum local space requirements for Clustered Watch to be enabled for watch: CLW1556875916
    Depending on cluster size, this may take some time
[W] Broker node nextbeer meets the minimum amount of local disk space required 1024KB to run in degraded mode
    but not the recommended amount of local disk space 20971520KB
[W] Broker node nextcan1.beer.loc meets the minimum amount of local disk space required 1024KB to run in degraded mode
    but not the recommended amount of local disk space 20971520KB
[W] Broker node nextcan2.beer.loc meets the minimum amount of local disk space required 1024KB to run in degraded mode
    but not the recommended amount of local disk space 20971520KB
[I] Successfully verified all configured MsgQueue nodes meet minimum local space requirements for Clustered Watch to be enabl
[I] Verified the watch type is INODE for independent fileset fromUK
[I] Successfully created Clustered Watch topic on the MsgQueue for watch: CLW1556875916
[I] Successfully added Clustered Watch configuration file into CCR for watch: CLW1556875916
[I] Successfully enabled Clustered Watch consumers for watch: CLW1556875916
[I] Successfully added Clustered Watch policy rules for watch: CLW1556875916
[I] Successfully enabled Clustered Watch: CLW1556875916
```

```
[root@nextbeer beer]# mmwatch beer status
```

```
beer                CLW1556875916      Active
  Node Name
  nextbeer           HEALTHY
  nextcan1.beer.loc HEALTHY
  nextcan2.beer.loc HEALTHY
```

```
[root@nextbeer fromUK]# █
```

How to use watch folder (5.0.3) - example



mmwatch beer enable --fileset fromUK --events **IN_CREATE** --event-handler kafkasink --sink-brokers "kafka1:9092" --sink-top

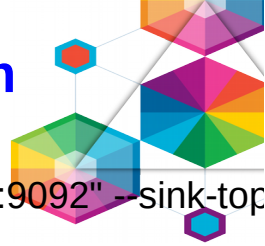
```
[root@nextbeer fromUK]# mmwatch beer enable --fileset fromUK --events IN_CREATE --event-handler kafkasink --sink-brokers "kaf
[I] Beginning enablement of Clustered Watch with newly created watch ID: CLW1556875916
[I] Successfully created clustered watch consumer node class kafkaClWatchConsumerServers
[I] Verifying MsgQueue nodes meet minimum local space requirements for Clustered Watch to be enabled for watch: CLW1556875916
    Depending on cluster size, this may take some time
[W] Broker node nextbeer meets the minimum amount of local disk space required 1024KB to run in degraded mode
    but not the recommended amount of local disk space 20971520KB
[W] Broker node nextcan1.beer.loc meets the minimum amount of local disk space required 1024KB to run in degraded mode
    but not the recommended amount of local disk space 20971520KB
[W] Broker node nextcan2.beer.loc meets the minimum amount of local disk space required 1024KB to run in degraded mode
    but not the recommended amount of local disk space 20971520KB
[I] Successfully verified all configured MsgQueue nodes meet minimum local space requirements for Clustered Watch to be enabl
[I] Verified the watch type is INODE for independent fileset fromUK
[I] Successfully created Clustered Watch topic on the MsgQueue for watch: CLW1556875916
[I] Successfully added Clustered Watch configuration file into CCR for watch: CLW1556875916
[I] Successfully enabled Clustered Watch consumers for watch: CLW1556875916
[I] Successfully added Clustered Watch policy rules for watch: CLW1556875916
[I] Successfully enabled Clustered Watch: CLW1556875916
```

GPFS : beer

For each new file

For a whole fileset : here ... fromUK

How to use watch folder (5.0.3) - step 3 configure watch



```
mmwatch beer enable --fileset fromUK --events IN_CREATE --event-handler kafkasink --sink-brokers "kafka1:9092" --sink-top
```

```
[root@nextbeer fromUK]# mmwatch beer enable --fileset fromUK --events IN_CREATE --event-handler kafkasink --sink-brokers "kaf
[I] Beginning enablement of Cluster
[I] Successfully created cluster
[I] Verifying Msq
[W] Broker
[W] Br
[W] Br
[I] Su
[I] Ve
[I] Su
[I] Su
[I] Su
```

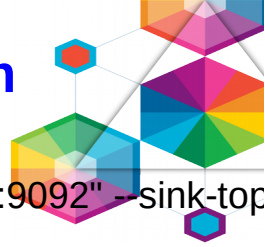
default: watch for all ...

IN_ACCESS
IN_ATTRIB
IN_CLOSE_NOWRITE
IN_CLOSE_WRITE
IN_CREATE
IN_DELETE
IN_DELETE_SELF

IN_IGNORED
IN_ISDIR,IN_MODIFY
IN_MOVE_SELF
IN_MOVED_FROM
IN_MOVED_TO
IN_OPEN
IN_Q_OVERFLOW
IN_UNMOUNT

```
[root@nextbeer]#
```

How to use watch folder (5.0.3) – step 3 configure watch

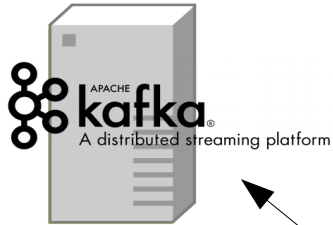


```
mmwatch beer enable --fileset fromUK --events IN_CREATE --event-handler kafkasink --sink-brokers "kafka1:9092" --sink-top
```

The command does

- creates kafka producer for internal Kafka (GPFS)
- creates topic
- creates the policy , enables the policy to the filesystem
- provides a consumer to retrieve events from GPFS Kafka and sends it to the customer owned kafka

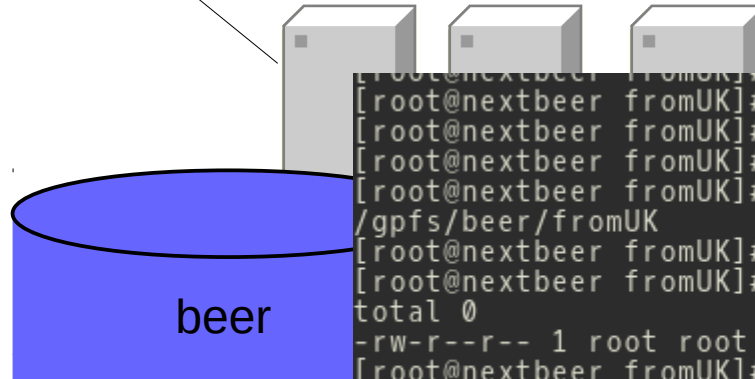
How to use watch folder (5.0.3) - step 3 nice example



```
[[root@kafka1 myown]# ./rescueEurope.py

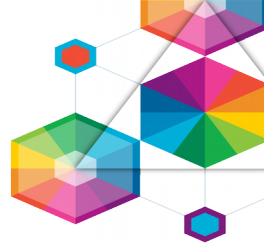
event=IN_CREATE, path=/gpfs/beer/fromUK/get_CamdenPils
/gpfs/beer/fromUK/get_CamdenPils
get_CamdenPils
place a copy in /gpfs/beer/AddTax/get_CamdenPils
```

```
ent": "IN_CREATE", "path": "/gpfs/beer/fromUK/get_CamdenPils", "clusterName": "nextbeer", "nodeName": "nextbeer", "nfsCl
8", "fileSetID": "5", "linkCount": "1", "openFlags": "0", "poolName": "system", "fileSize": "0", "ownerUserId": "0", "o
02-0400", "ctime": "2019-05-03_13:48:02-0400", "mtime": "2019-05-03_13:48:02-0400", "eventTime": "2019-05-03_13:48:02-0
", "processId": "4724", "permissions": "200100644", "acls": null, "xattrs": null, "subEvent": "NONE" }
```

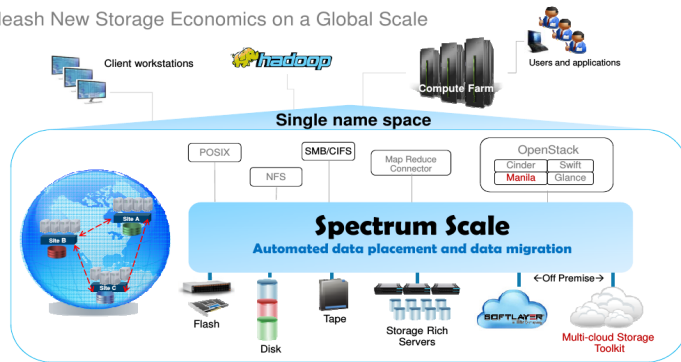


```
[root@nextbeer fromUK]# cd /gpfs/beer/fromUK
[root@nextbeer fromUK]#
[root@nextbeer fromUK]#
[root@nextbeer fromUK]# pwd
/gpfs/beer/fromUK
[root@nextbeer fromUK]# touch get_CamdenPils
[root@nextbeer fromUK]# ll
total 0
-rw-r--r-- 1 root root 0 May  3 13:48 get_CamdenPils
[root@nextbeer fromUK]# ls -l /gpfs/beer/AddTax/get_CamdenPils
-rw-r--r-- 1 root root 0 May  3 13:48 /gpfs/beer/AddTax/get_CamdenP
[root@nextbeer fromUK]#
```

Watch folder (5.0.3)



Unleash New Storage Economics on a Global Scale



– you can create customized workflows

– interact on data ingest, post process data

..some other ideas...

- migrate the file after close to another storage tier
- scan the file / Antivir
- automated process / document management (insurances)
- tag the file with EA's



- **Where to look first**

- mmwatch – verify information about all currently running watches in the cluster
- /var/adm/ras/mmwf.log – primary log file for watch API and mmwatch command
- /var/adm/ras/mmfs.log (major problems with policy, watches, etc.) and /var/adm/ras/mmmqueue.log (problems with the message queue)

- **CallHome integration**

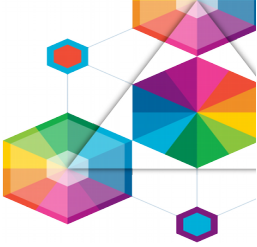
- Output of ‘mmwatch all list -Y’ and inclusion of /var/adm/ras/mmwf.log

- GPFS snap integration

- Collect component logs: mmmqueue.log, mmaudit.log, mmwf.log
- Collect most important Kafka logs
- Collect all configuration in CCR and list of all active watches and topics

- GPFS trace support for watch folder API, level 4 FS usually sufficient

Watchfolder can help to



- post or pre process data , without scanning all MD / name space
- speed up regular workload, by freeing IO resources from long MD scans
- Audits / manage all events in a central queue (kafka)
- create / customize workflow
- have fun

