

Container & Object Storage hints and tips

Spectrum Scale German User Meeting 2022
Cologne, Germany – October 19-20, 2022

Harald Seipp (IBM)

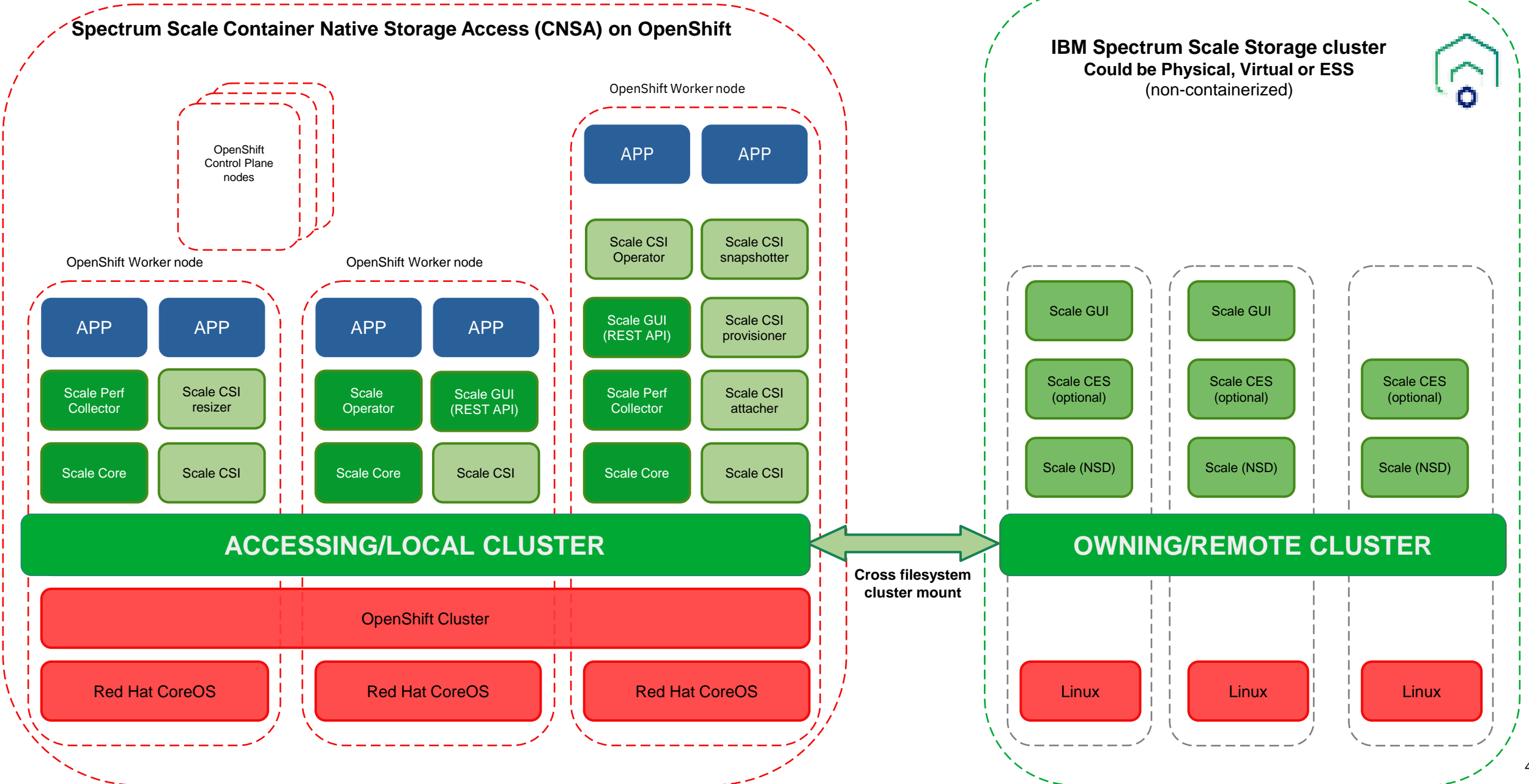


Disclaimer

IBM's statements regarding its plans, directions, and intent are subject to change or withdrawal without notice at IBM's sole discretion. Information regarding potential future products is intended to outline our general product direction and it should not be relied on in making a purchasing decision. The information mentioned regarding potential future products is not a commitment, promise, or legal obligation to deliver any material, code, or functionality. The development, release, and timing of any future features or functionality described for our products remains at our sole discretion.

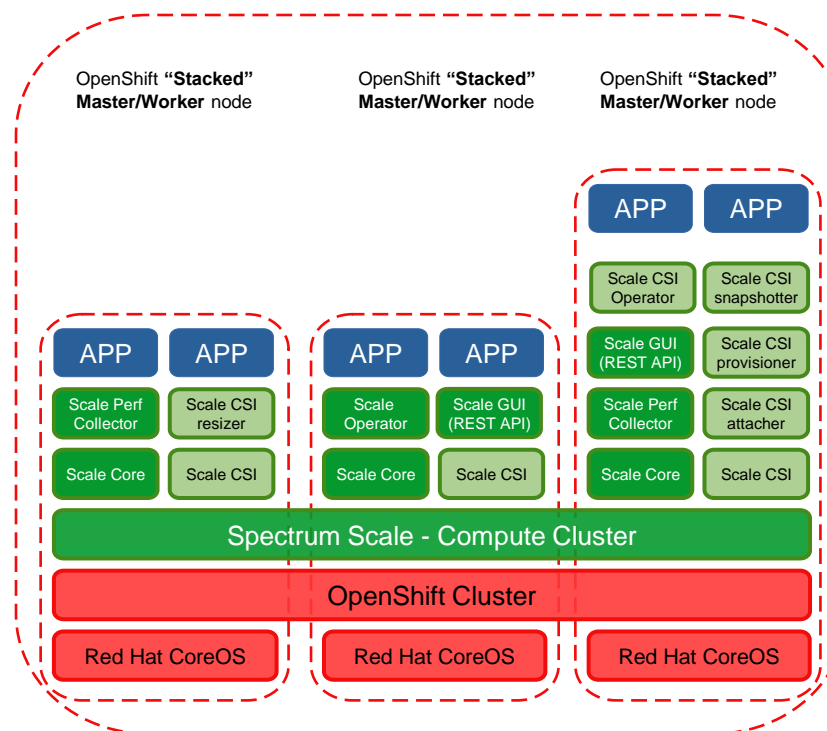
IBM reserves the right to change product specifications and offerings at any time without notice. This publication could include technical inaccuracies or typographical errors. References herein to IBM products and services do not imply that IBM intends to make them available in all countries.

Containers: Spectrum Scale on OpenShift



Spectrum Scale CNSA/CSI on OpenShift Control Plane Nodes?

- Example: Minimum 3-node OpenShift deployment with “Stacked master/workers”
- Supported with CNSA/CSI 5.1.3 and later, driven by Spectrum Scale DAS requirement
- Not recommended when hosting mission critical applications



Spectrum Scale CNSA/CSI Resource Requests/Limits

- For CNSA releases prior to 5.1.5, we just specified a minimum node size
 - Tightly sized clusters with demanding applications require a more detailed resource consumption estimate
- The CNSA Pods specify their requests and limits

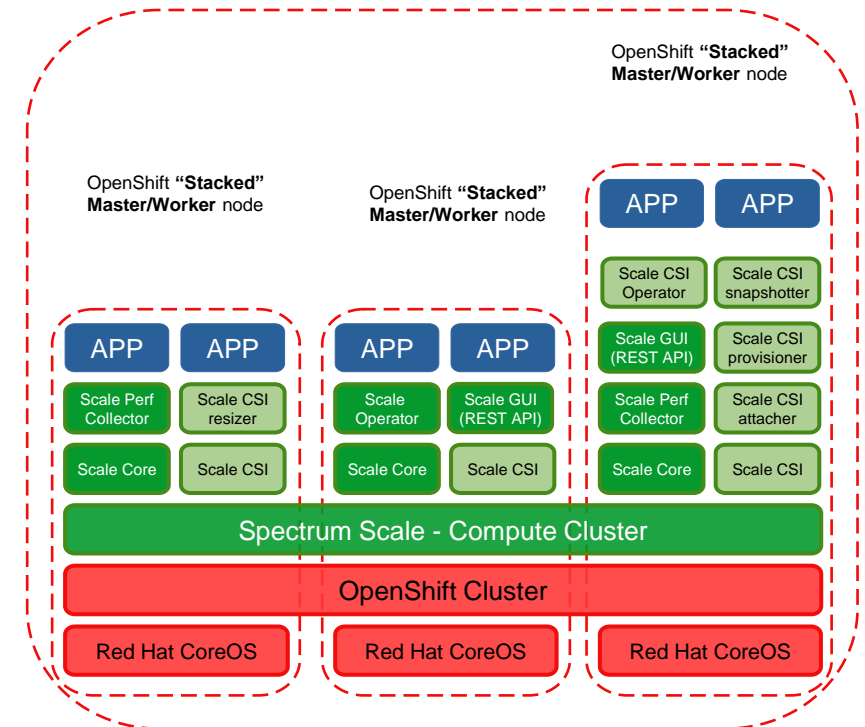
```
for i in `oc get pod -o name -n ibm-spectrum-scale`; do
echo -n $i && oc get $i -o jsonpath='{
"}.spec.containers[*].resources}{'\n"}' -n ibm-spectrum-scale; done
```

```
pod/ibm-spectrum-scale-gui-0 {"limits":{"cpu":"2","memory":"1000Mi"},"requests":{"cpu":"500m","memory":"750Mi"}} {"limits":{"cpu":"1",
pod/ibm-spectrum-scale-gui-1 {"limits":{"cpu":"2","memory":"1000Mi"},"requests":{"cpu":"500m","memory":"750Mi"}} {"limits":{"cpu":"1",
pod/ibm-spectrum-scale-pmcollector-0 {"limits":{"cpu":"500m","memory":"10000Mi"},"requests":{"cpu":"100m","memory":"5000Mi"}} {"limits
pod/ibm-spectrum-scale-pmcollector-1 {"limits":{"cpu":"500m","memory":"10000Mi"},"requests":{"cpu":"100m","memory":"5000Mi"}} {"limits
pod/worker1 {"limits":{"cpu":"24","memory":"148511584Ki"},"requests":{"cpu":"6","memory":"46i"}} {"limits":{"cpu":"100m","memory":"60M
pod/worker2 {"limits":{"cpu":"24","memory":"148512056Ki"},"requests":{"cpu":"6","memory":"46i"}} {"limits":{"cpu":"100m","memory":"60M
pod/worker3 {"limits":{"cpu":"16","memory":"148511508Ki"},"requests":{"cpu":"4","memory":"46i"}} {"limits":{"cpu":"100m","memory":"60M
```

- CSI Pods resource consumption is negligible
- With CNSA 5.1.5 we provide a much more detailed resource consumption overview (see table on the top right)

Table 1. Hardware requirements

Pods	Where deployed	CPU request	Memory request	Storage	Description
core fs	Default all workers	>=1000mCPU, default 25%	>=2GiB, default 25%	Config in /var (~25GiB)	This is the pod that provides the filesystem service for the node. It is required to be deployed on all nodes where PVs are accessed from application pods. The CPU and memory requests can be customized in the cluster CR.
operator	Single node	100mCPU	40MiB	-	The controller runtime that manages all custom resources.
GUI	Two nodes	630mCPU	1.25GiB	Local PV for DB	The graphical user interface and ReST API.
pmcollector	Two nodes	120 mCPU	3-7GiB depending on cluster size	Local PV for DB	The performance collector database.
grafana-bridge	One node	100mCPU	1GiB	-	The bridge for accessing pmcollector from grafana.



Infra Nodes

- Infra nodes in OpenShift are workers with a special label (and license)
- It is allowed* to deploy Spectrum Scale CNSA and CSI to these nodes!

Where applicable, end users can use infrastructure nodes without disqualifying the node for infrastructure licensing to house these software components.
Examples may include:

- CNI and CSI drivers and controllers (also known as plugins).

- Follow Red Hat's best practice to have infra nodes also labeled as worker
 - Avoids issue with machine config pools
- Remove NoSchedule policy for the CNSA and CSI namespaces
 - Edit the namespace and add annotation `openshift.io/node-selector: ""` (empty string is important)

Infrastructure nodes allow customers to isolate infrastructure workloads for two primary purposes:

1. to prevent incurring billing costs against subscription counts and
2. to separate maintenance and management.

Label the CNSA nodes with

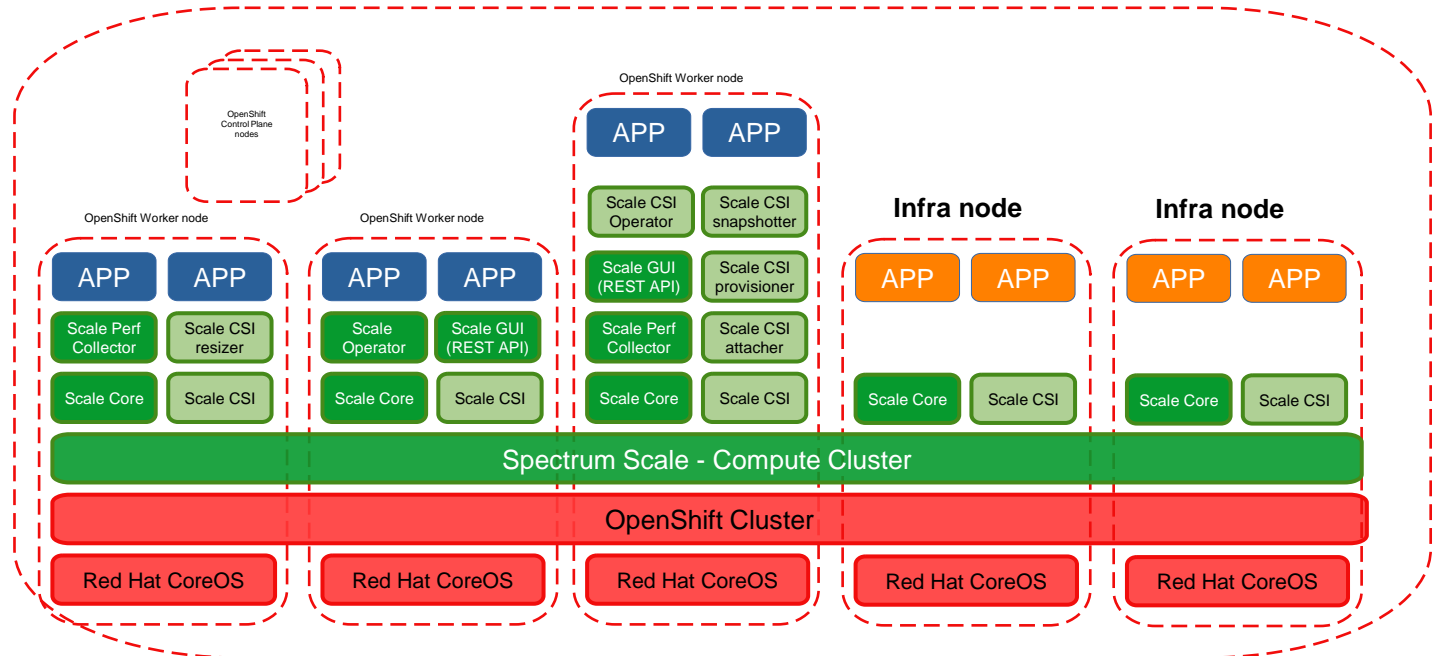
`app.kubernetes.io/component: "scale"`

and then use exactly that node selector in the ScaleCluster Custom Resource (CR):

```
Spec:
  daemon:
    nodeSelector:
      app.kubernetes.io/component: "scale"
```

Important for small clusters: allow GUI and PMCollector to run on Infra nodes:

```
gui:
  nodeSelector:
    node-role.kubernetes.io/infra: ""
  tolerations:
    - effect: NoSchedule
      operator: Exists
pmcollector:
  nodeSelector:
    node-role.kubernetes.io/infra: ""
  tolerations:
    - effect: NoSchedule
      operator: Exists
```



* Reference: <https://www.openshift.com/learn/sizing-subscription-guide>

Prometheus

- Prometheus (used with OpenShift Monitoring) uses advanced Pod Security options:

```
securityContext:  
  seLinuxOptions:  
    level: 's0:c20,c10'  
  runAsUser: 65534  
  runAsNonRoot: true  
  fsGroup: 65534
```

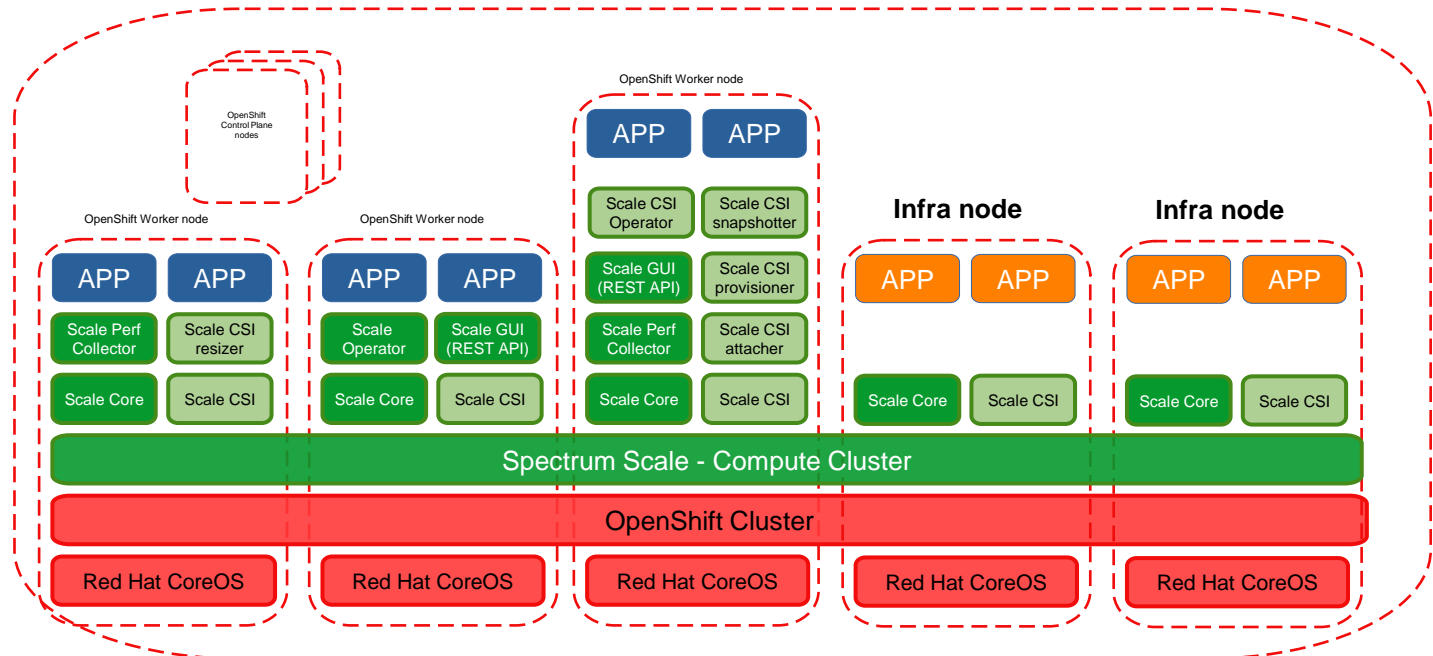
- Setting UID and GID for the storage class was not sufficient, prometheus containers crash with “permission denied”

- Reason is that the containers use the “sub-path” mount Kubernetes feature:

```
oc get pod prometheus-k8s-1 -o  
jsonpath='{.spec.containers[].volumeMounts}' | jq  
[  
  ....  
  {  
    "mountPath": "/prometheus",  
    "name": "prometheus-k8s-db",  
    "subPath": "prometheus-db"  
  },  
  ....  
]
```

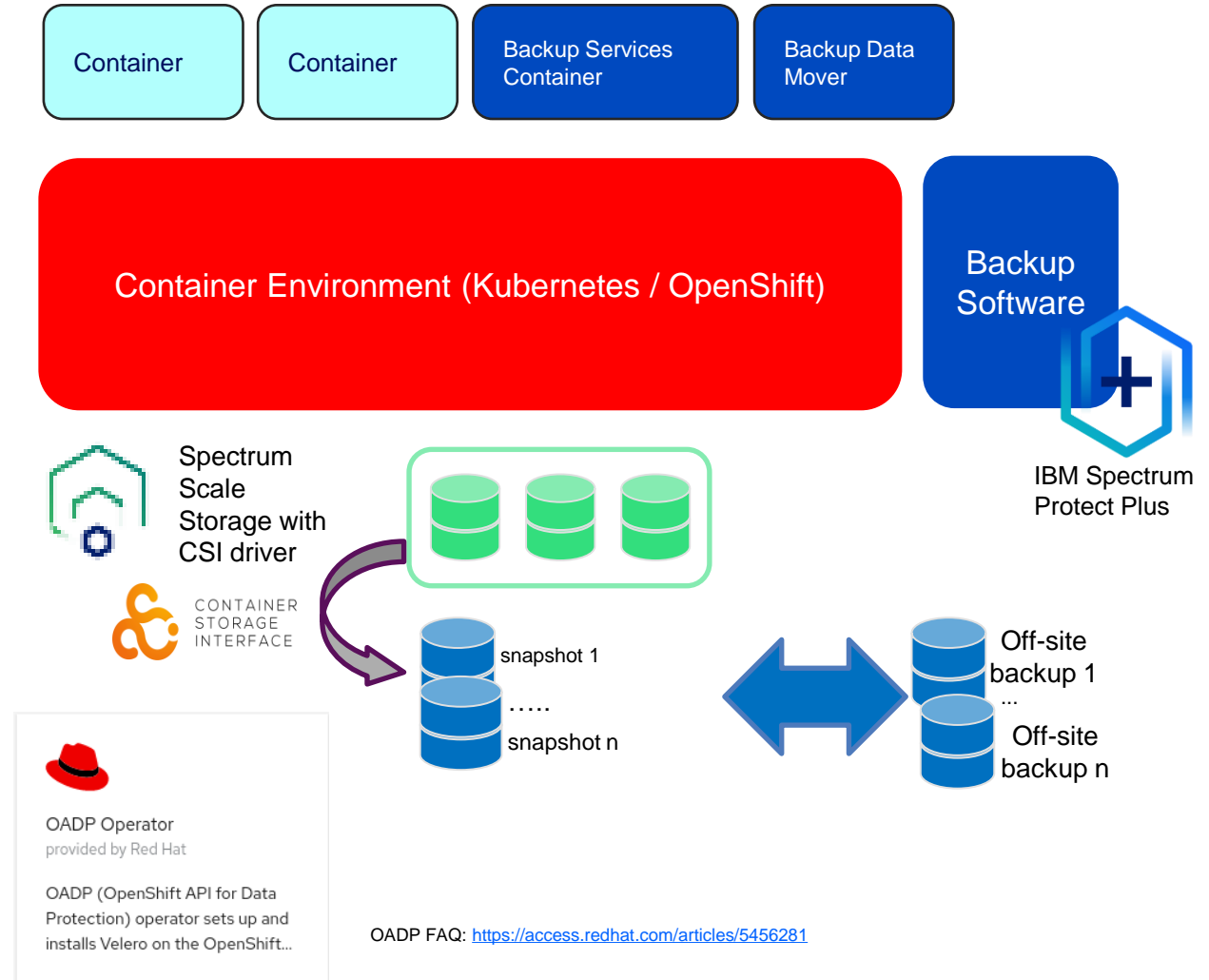
- Solutions:

- CSI prior to 2.6.0: use a Storage Class with permissions: '777'
(Please note that this is not as insecure as it might look like as SELinux policies still apply and overrule the POSIX rights)
- CSI 2.6.0 and later: fsGroup is supported for RWO volumes



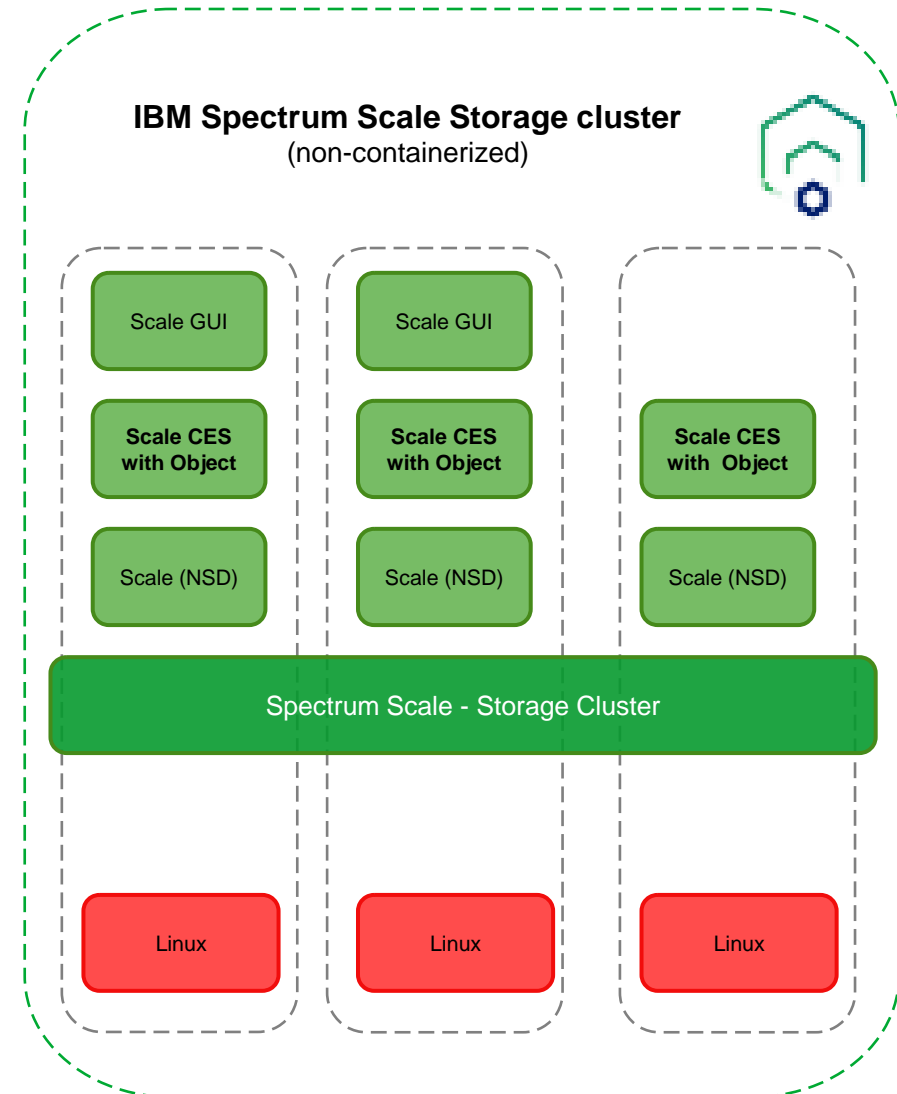
Backup and Restore with Spectrum Protect Plus

- IBM Spectrum Protect Plus Container Backup supports Spectrum Scale CNSA/CSI and protects persistent container data
 - Self service with Kubernetes commands and/or through automated policies as part of application deployment
 - Crash consistency guaranteed by CSI snapshots
 - Application consistent Backup and Recovery through a Backup Service with application integration (quiesce/unquiesce)
 - Protection of application metadata (Namespace)
 - Protection of cluster state (etcd database)
- Spectrum Scale CSI 2.5.0 introduced Consistency Groups
 - All PVs belonging to a namespace are backed up and restored consistently when a storage class with consistency group is used



Spectrum Scale Object Hints and Tips

- Note: The following covers the Spectrum Scale 5.1.x CES Object stack based on OpenStack Swift
 - Deprecated* since 5.1.5, see latest [Spectrum Scale FAQ](#)
- There will be a separate session on the new Spectrum Scale DAS S3 tomorrow

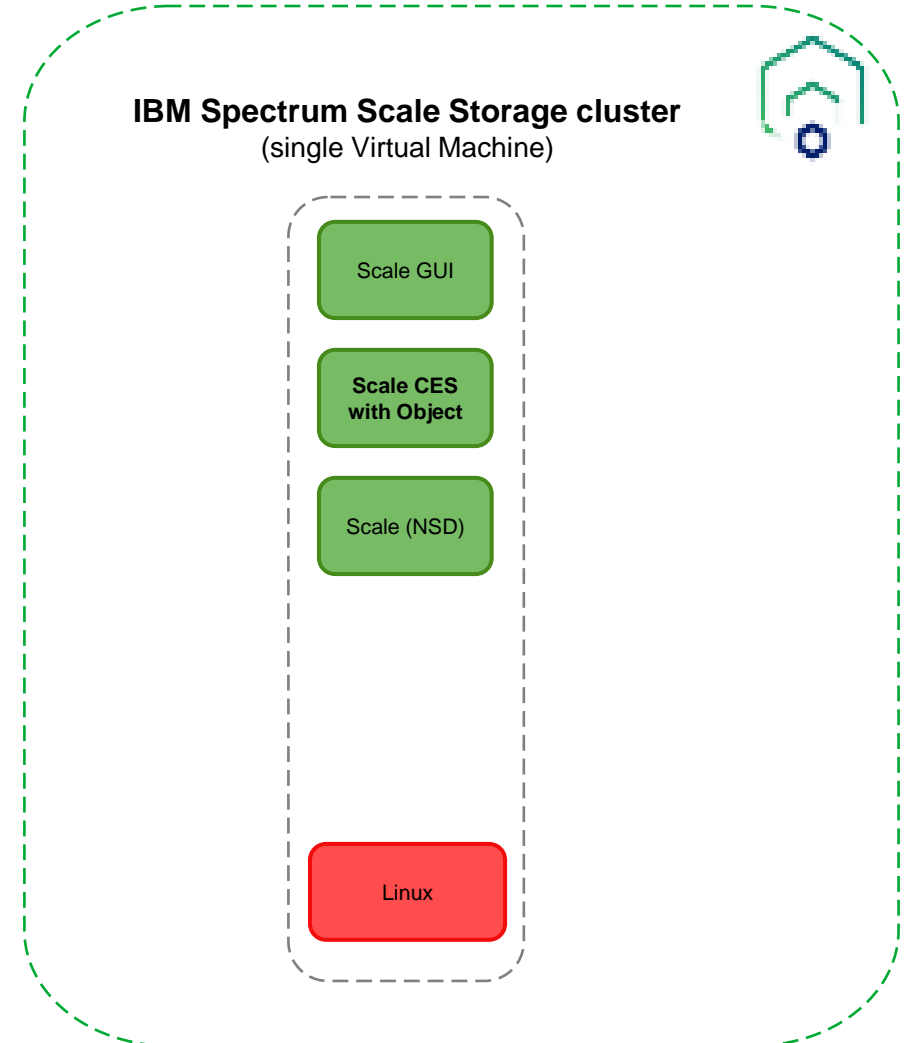


* A feature is a deprecated feature if that specific feature is supported in the current release, but the support might be removed in a future release. In some cases, it might be advisable to plan to discontinue the use of deprecated functionality.

Spectrum Scale Object S3 Test Bed leveraging Spectrum Scale Vagrant

Quick way to test drive Spectrum Scale with S3 Object CES:

```
# git clone https://github.com/IBM/SpectrumScaleVagrant.git  
# cd SpectrumScaleVagrant  
# cp ../Spectrum_Scale_Developer-5.1.5.0-x86_64-Linux software  
# cd virtualbox/prep-box  
# vagrant up  
# vagrant package SpectrumScale_base --output SpectrumScale_base.box  
# vagrant destroy  
# cd ..  
# vagrant up
```




Spectrum Scale Object S3 + Virtual Host Path

- Spectrum Scale supports Path-style URL addressing by default
- Virtual Host Path URL can be enabled:


```
mmobj config change --ccrfile proxy-server.conf --section filter:s3api
-property dns_compliant_bucket_names --value true
mmobj config change --ccrfile proxy-server.conf --section filter:s3api
-property storage_domain --value s3server.scale.com
```
- For more details, please have a look at the excellent paper available at IBM Community:

Setting up an S3 Compliant Object Store in IBM Spectrum Scale with Object Compression, ILM and Tiering


Tue April 26, 2022 06:12 AM



[Nishaan Docrat](#)

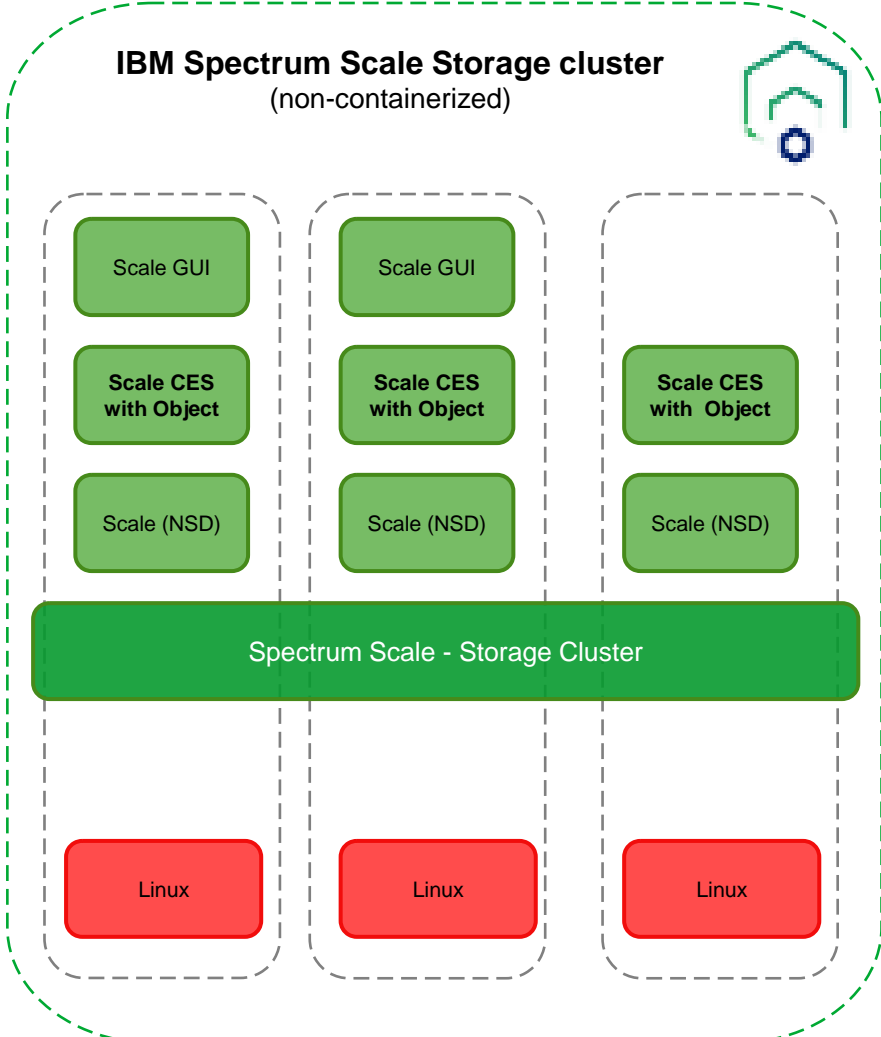
IBM Spectrum Scale is a high-performance clustered filesystem that offers support for multiple protocols including object. Since IBM Spectrum Scale is based on the OpenStack Swift implementation of the object protocol there is a need to also support Amazon's Simple Storage Service (S3) object protocol for applications that require an S3 compatibility. This emulation is achieved by using Swift middleware. The ability to support both AWS S3 and OpenStack Swift protocols in IBM Spectrum Scale Object means that it can support a broader variety of Cloud applications using either object protocol. Amazon has recently announced the deprecation of their path-style addressing scheme in favour of their virtual-hosted style addressing scheme (also referred to as DNS style addressing). Since OpenStack Swift by default supports path-style addressing, it is important to be able to understand what is required to enable this virtual-hosted style addressing in IBM Spectrum Scale to ensure compatibility with S3 applications going forward. This document covers all the steps that are required in order to achieve this. In addition, advanced Spectrum Scale functions like Object compression, Object Information Lifecycle Management (ILM) and Object Tiering are detailed in depth to showcase the benefits of using IBM Spectrum Scale for your Object storage requirements.

Attachment(s)

 [Setting up an S3 Compliant Object Store in IBM Spectrum S...02](#)
6.36 MB | 1 version
Uploaded - Sat May 07, 2022
[Download](#)

Path-style URL:
<https://spectrumscale.cluster/my-bucket>

Virtual Host Path URL:
<https://my-bucket.spectrumscale.cluster>



Spectrum Scale Object S3 + Audit Logging

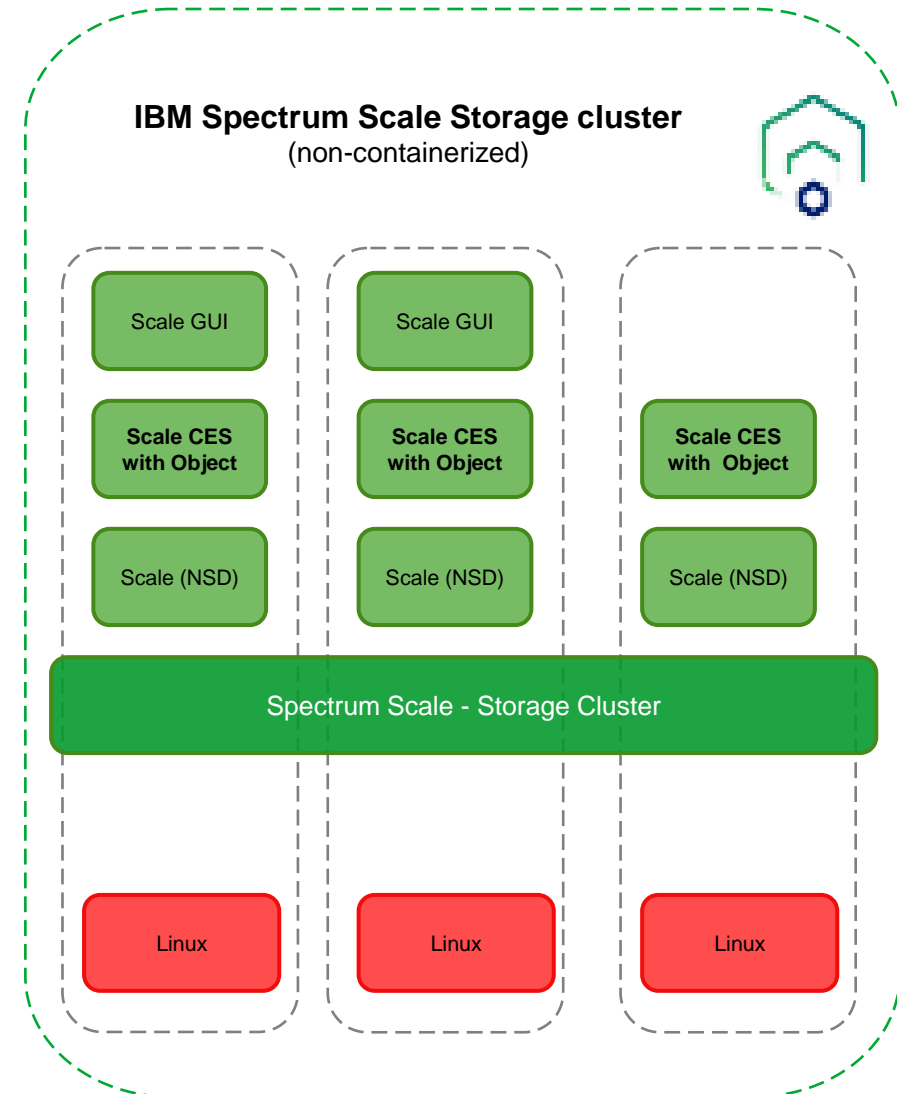
- Challenge: Audit logging S3 requests per user
- When you enable INFO-level logging for the Spectrum Scale Object Proxy server you'll get information about the requests by requesting IP address

```
sudo mmobj config change --ccrfile proxy-server.conf --section DEFAULT --property log_level --value DEBUG
```

- When you enable DEBUG-level logging, the logs will also include the requesting user

```
sudo mmobj config change --ccrfile proxy-server.conf --section DEFAULT --property log_level --value DEBUG
```

- Please note that Spectrum Scale File Audit Logging (FAL) can be used to log object accesses, but all log entries will point to the `swift` user



Thank you for using
IBM Spectrum Scale!