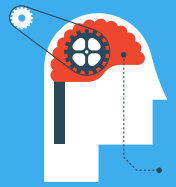


Spectrum Scale Expert Talks

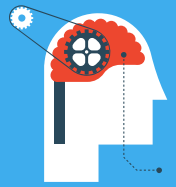
Episode 6:

Persistent Storage for Kubernetes and OpenShift environments

IBM Spectrum Scale

Show notes:
www.spectrumscaleug.org/experttalks

Join our conversation:
www.spectrumscaleug.org/join



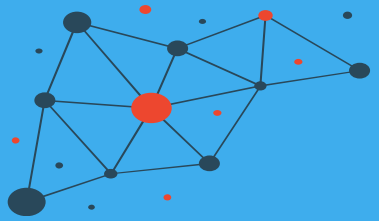
SSUG::Digital

Welcome to digital events!

IBM Spectrum Scale

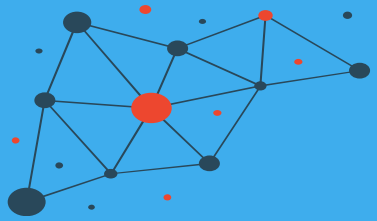
Show notes:
www.spectrumscaleug.org/experttalks

Join our conversation:
www.spectrumscaleug.org/join



About the user group

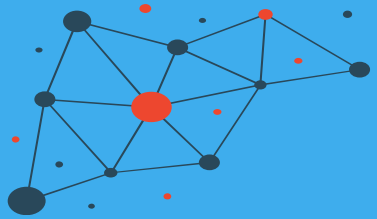
- Independent, work with IBM to develop events
- Not a replacement for PMR!
- Email and Slack community
- https://www.spectrumscaleug.org/join



We are …

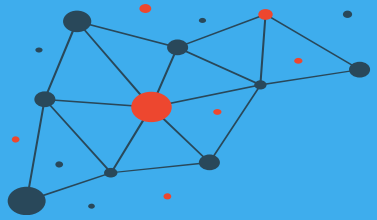
- Simon Thompson (UK)
- Kristy Kallback-Rose (USA)
- Bob Oesterlin (USA)
- Bill Anderson (USA)
- Chris Schipalius (Australia)

IBM CHAMPION



Check https://www.spectrumscaleug.org/experttalks for charts, show notes and upcoming talks

- Past talks:
  - 001: What is new in Spectrum Scale 5.0.5
  - 002: Best practices for building a stretched cluster
  - 003: Strategy update
  - 004: Update on performance enhancements in Spectrum Scale (file create, MMAP, direct IO, ESS 5000)
  - 005: Update on functional enhancements in Spectrum Scale (inode management, vCPU scaling, NUMA considerations)
- Today:
  - Oct 6: Persistent Storage for Kubernetes and OpenShift environments
- Next:
  - Oct 21: Best practices for information lifecycle management (ILM)
  - Nov 4: Multi-node scaling of AI workloads using Nvidia DGX, OpenShift and Spectrum Scale
  - Nov 16: User Meeting at SC20 (Session 1) (more details will follow)
  - Nov 18: User Meeting at SC20 (Session 2) (more details will follow)



Speakers

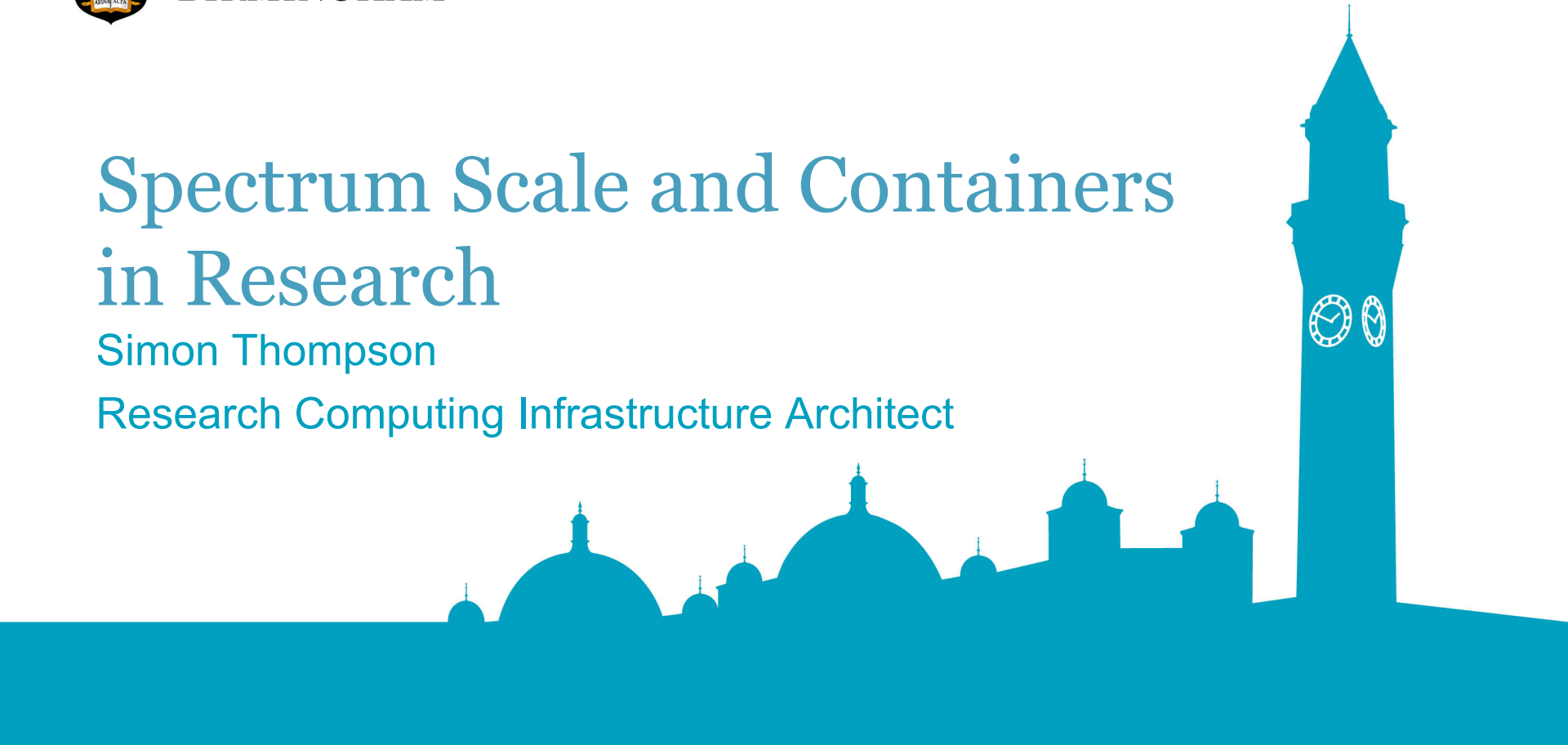
- Simon Thompson (University of Birmingham)
  - Spectrum Scale and containers in research
- Smita Raut (IBM)
  - Spectrum Scale CSI driver
- Renar Grunenberg (HuK-Coburg)
  - Use case: Kafka self-service
- Harald Seipp (IBM)
  - More use cases

UNIVERSITY OF BIRMINGHAM

Spectrum Scale and Containers in Research

Simon Thompson

Research Computing Infrastructure Architect



Supercomputing, AI, Storage, Software Services,
Training, Private Cloud

Delivering storage for researchers

What our researchers want

- Play worlds to:
  - Experiment with code
  - Develop systems
- Access to their data
- r00t

- VMs + NFS
  - NFS is kinda slow
  - Tricky to manage root+permissions
  - We end up having to install software for them

Enter containers

- Give researchers an environment they can control inside
- Use CSI to mount selected file-sets into containers
- Allow self-provisioning of containers with storage
- Rapid deployment/prototyping environments

- Thanks to Christopher Hall, Alex Lloyd who did most of the work on this…

UNIVERSITY OF BIRMINGHAM

Challenges

- Vagueness about using existing file-sets (you can)
- Can’t map all IO to a single user
  - Data in the ”real world” can be owned by many different people
- We were stupid
  - You need a k8s node (VM?) which is a Scale client
  - Our k8s deployment didn’t lend itself to this
  - Name mapping … use mmlscluster output, not FQDN of VM

UNIVERSITY OF BIRMINGHAM

Things to note

- K8s node needs to be CentOS or RHELS (not CoreOS)
- Systemd dependency needed for k8s services -> Scale
- We provisioned VMs for k8s nodes
  - Kubespray
  - Need post spray tool for Scale client
  - IB Pass-Through required for performance (= no live migration)
- Some networking types didn’t work for us!

UNIVERSITY OF BIRMINGHAM

Introduction to CSI Driver

Disclaimer

IBM Spectrum Scale

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.

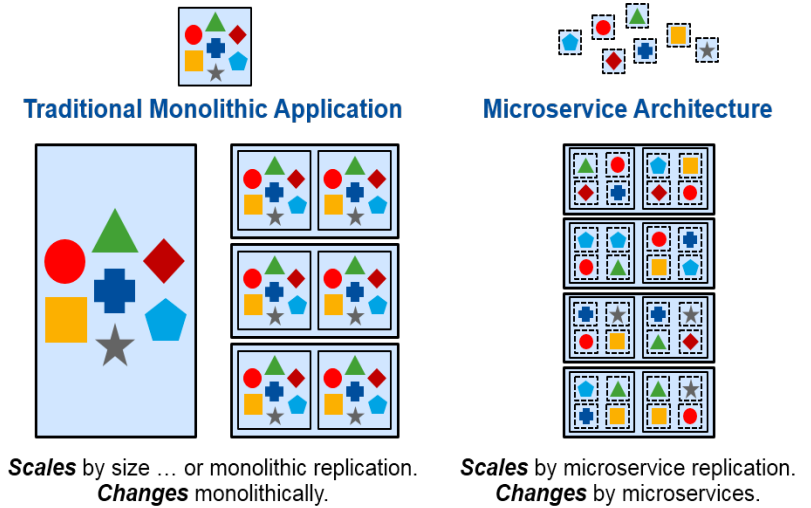
Outline

- ***Introduction***
- ***Spectrum Scale CSI Driver***
- ***Use Cases***
- ***Spectrum Scale On OpenShift***

Baseline: Containers Everywhere

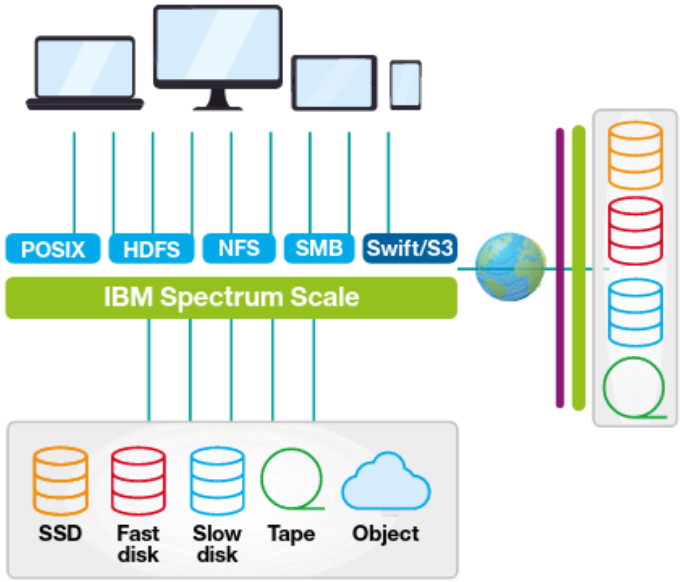
IBM Spectrum Scale

- Multicloud: On-premises and Public Clouds
  - Elastic scheduling and auto-scaling
  - Improved resource utilization
  - Secure isolation and Multi-Tenancy
  - Portable and reproducible service
  - One-click Laptop to Supercomputer
- Development, DevOps and continuous integration
  - Re-use of applications and services
  - Simplify and accelerate application deployment
- Microservices Architecture
  - Programming language and technology stack independence
  - Faster and easier development



Baseline: IBM Spectrum Scale

Highly scalable high-performance unified storage for files and objects with integrated analytics



Remove data-related bottlenecks
- with a parallel, scale-out solution
- 2.5TB/s demonstrated throughput

Enable global collaboration
- with unified storage and global namespace
- Data Lake serving HDFS, files and object across sites

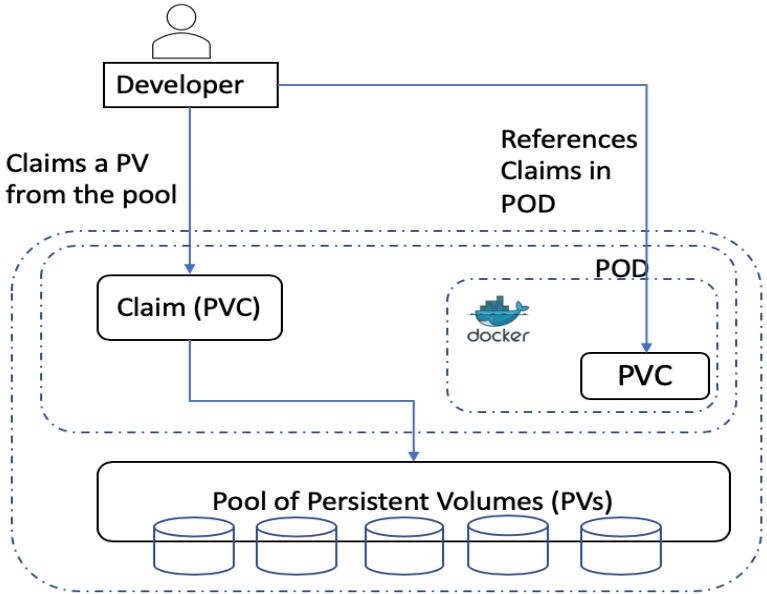
Optimize cost and performance
- with automated data placement
- thin-provisioning preview and TRIM support, QOS on project preview

Ensure data availability, integrity and security
- with erasure coding, replication, snapshots, and encryption
- End-to-end checksum, Spectrum Scale RAID, NIST/FIPS certification

IBM Spectrum Scale
CSI Driver

Persistent Storage For Containers

Stateful containers: *Persistent data with high availability and data protection is one of the biggest barriers for container adoption in the enterprise for production workloads*



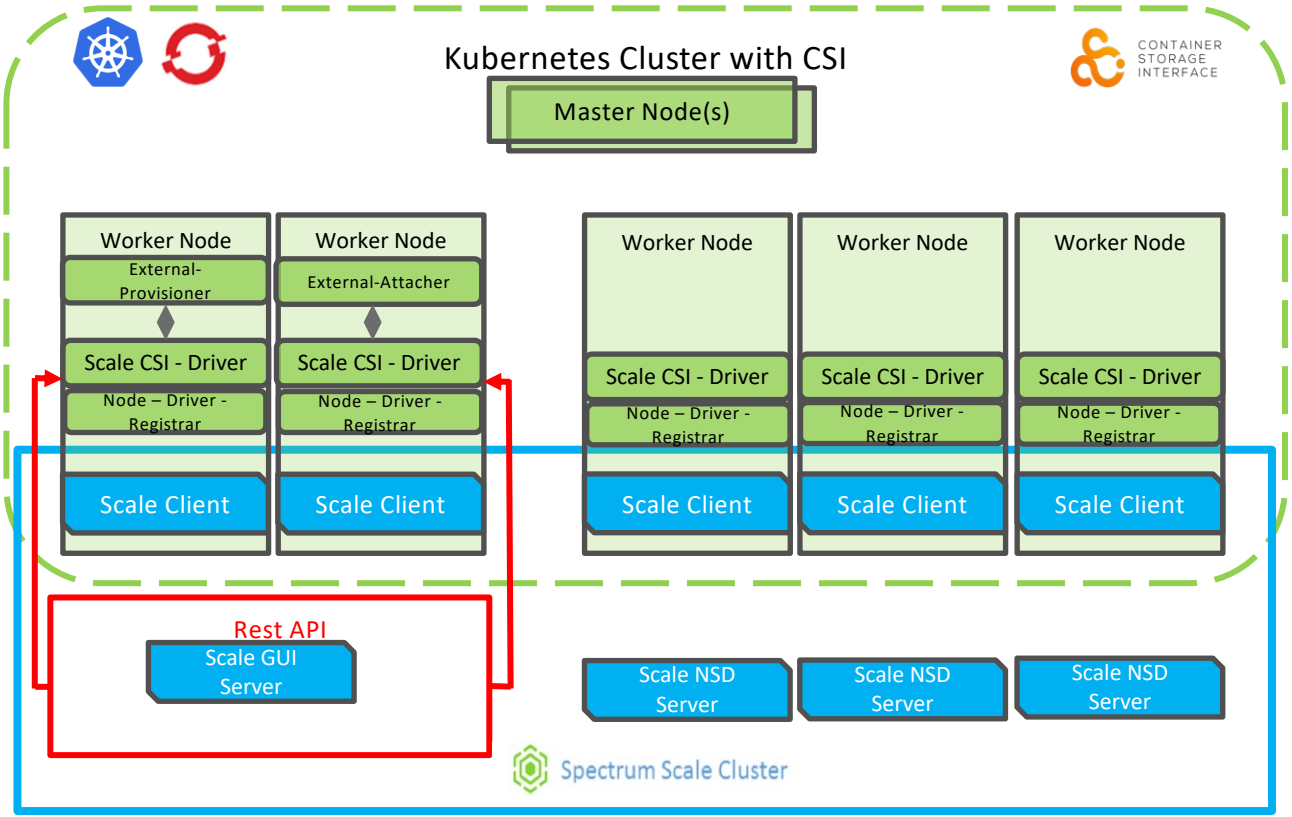
Kubernetes Volumes: PVC or Persistent Volume Claim is a request request for storage by a user. PVCs consume PV (Persistent Volume) resource.

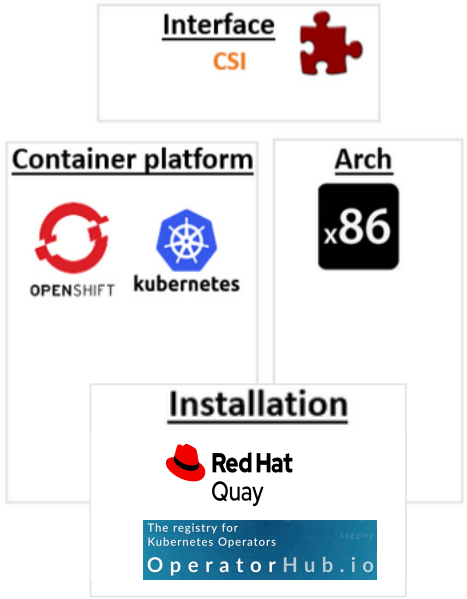
Dynamic Provisioning: Allows storage volumes to be created on-demand. Eliminates the need for cluster administrators to pre-provision storage.

Static Provisioning: Creates PVs upfront that carry details of the real storage. Administrator should know the storage requirements upfront.

Storage Class: Provides a way for administrators to define “classes” of storage.

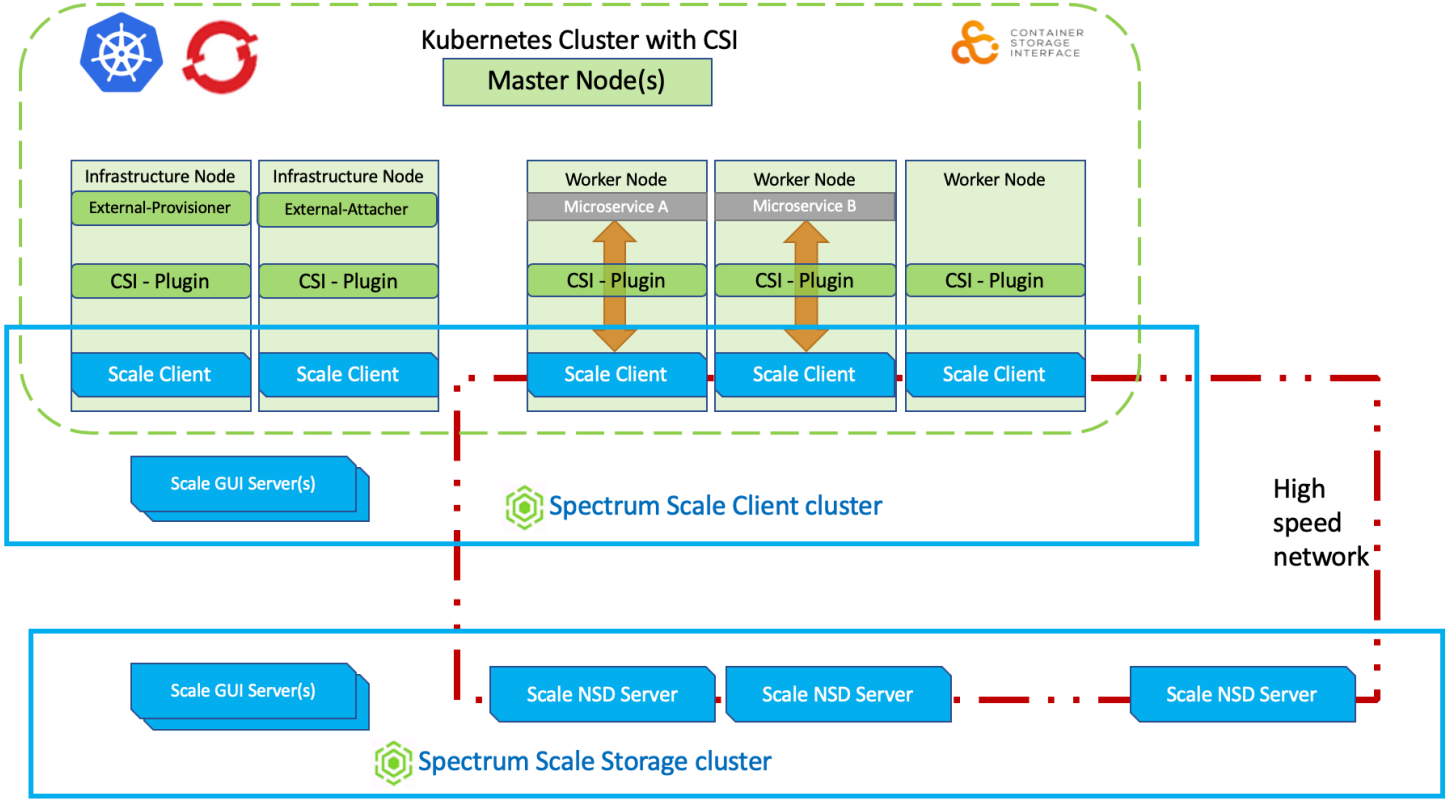
Spectrum Scale CSI Driver – Architecture



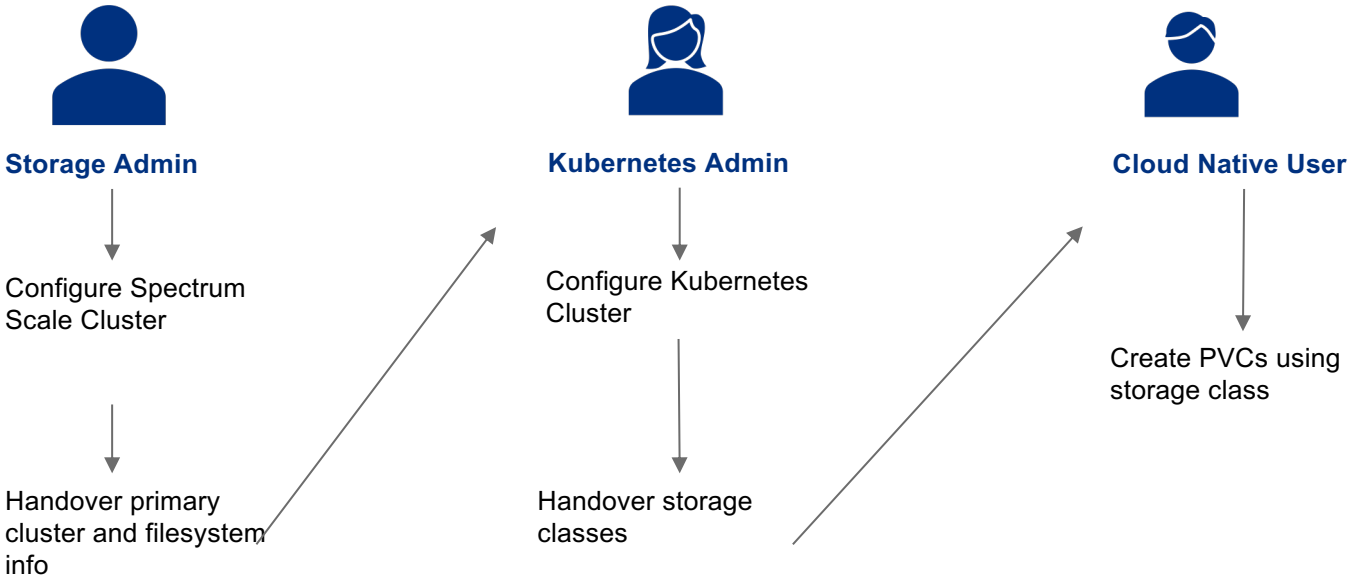


IBM Spectrum Scale

Spectrum Scale CSI Driver With Remote Cluster



Personas and Workflow



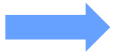
IBM Spectrum Scale

Static Provisioning vs Dynamic Provisioning

- **Static Provisioning**

Kubernetes Admin

Admin provisions static **PVs** in OpenShift/Kubernetes



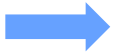
Cloud Native User

User claims volumes from pool of pre-provisioned **PVs** through **Persistent Volume Claim (PVC)**

- **Dynamic Provisioning**

Kubernetes Admin

Admin creates **StorageClass** in OpenShift/Kubernetes



Cloud Native User

User claims volumes from **StorageClass** through **Persistent Volume Claim (PVC)** *(self-service provisioning)*

IBM Spectrum Scale

Static Provisioning

Persistent Volume

```
apiVersion: v1
kind: PersistentVolume
metadata:
      name: static-scale-static-pv
spec:
     capacity:
          storage: 1Gi
     accessModes:
        - ReadWriteMany
     csi:
        driver: spectrumscale.csi.ibm.com
        volumeHandle:
"clusterID;FSUID;path=/gpfs/fs1/staticdir"
```

created

Persistent Volume Claim

```
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
    name: static-pvc
spec:
    accessModes:
       - ReadWriteMany
    resources:
    requests:
       storage: 1Gi
```

bound

Pod Definition

```
apiVersion: v1
kind: Pod
metadata:
      name: csi-scale-fsetdemo-pod
      labels:
          app: nginx
spec:
     containers:
        - name: web-server
        image: nginx
        volumeMounts:
              - name: mypvc
                mountPath:
/usr/share/nginx/html/scale
        ports:
              - containerPort: 80
  volumes:
      - name: mypvc
        persistentVolumeClaim:
            claimName: static-pvc
```

IBM Spectrum Scale

Dynamic Provisioning

StorageClass

```
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
      name: fileset-sc
provisioner:
spectrumscale.csi.ibm.com
parameters:
      volBackendFs: "gpfs1“
reclaimPolicy: Delete
```

Persistent Volume Claim

```
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
    name: scale-pvc
spec:
    accessModes:
      - ReadWriteMany
    resources:
    requests:
       storage: 1Gi
    storageClassName: fileset-sc
```

created bound

Pod Definition

```
apiVersion: v1
kind: Pod
metadata:
       name: csi-scale-demo-pod
       labels:
            app: nginx
spec:
      containers:
          - name: web-server
          image: nginx
          volumeMounts:
                - name: mypvc
                  mountPath: /usr/scale
         ports:
                - containerPort: 80
  volumes:
        - name: mypvc
              persistentVolumeClaim:
                  claimName: scale-pvc
```

IBM Spectrum Scale

Dynamic Provisioning - Storage Classes

Independent Fileset Provisioning

```
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
      name: csi-spectrum-scale-fileset
provisioner: spectrumscale.csi.ibm.com
parameters:
      volBackendFs: "gpfs1“
      uid: "1000"
      gid: "1000"
      inodeLimit: "1024"
reclaimPolicy: Delete
```

Dependent Fileset Provisioning

```
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
     name: spectrum-scale-fileset-
dependent
provisioner: spectrumscale.csi.ibm.com
parameters:
      volBackendFs: "gpfs1"
      uid: "1000"
      gid: "1000"
      filesetType: "dependent"
      parentFileset: “fset1"
reclaimPolicy: Delete
```

Lightweight Directory Provisioning

```
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
      name: csi-spectrum-scale-lt
provisioner: spectrumscale.csi.ibm.com
parameters:
      volBackendFs: "gpfs1"
      volDirBasePath: "DiffFS/N1"
      uid: "1000"
      gid: "1000"
reclaimPolicy: Delete
```

Use Case:
Kafka self service
HUK Coburg Insurance Group

HUK Coburg Insurance Group – Container Storage Interface Usecase KSS

Kafka Self Services with K8s and Spectrum Scale-CSI

October 2020

Contents

The HUK Coburg insurance group

HUK-COBURG
Aus Tradition günstig

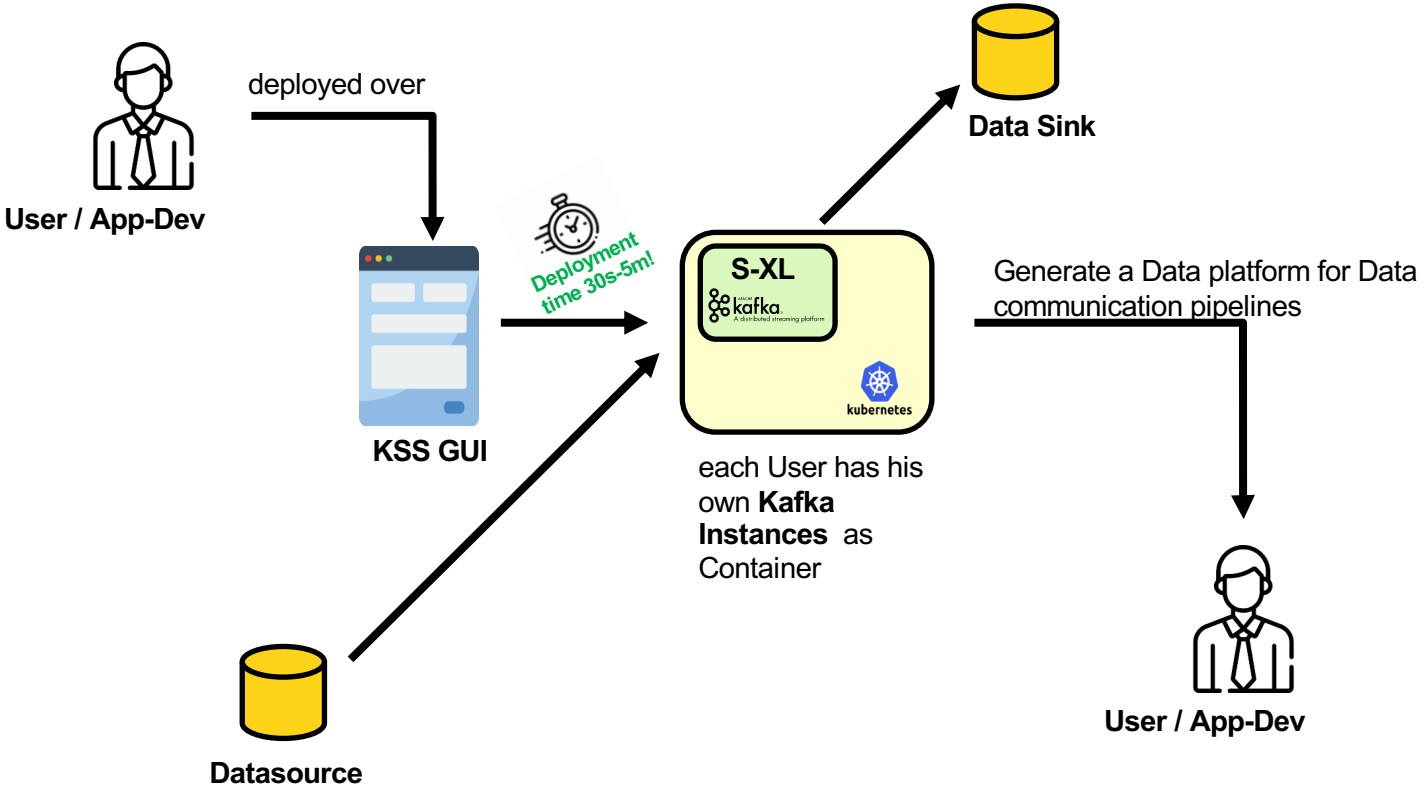
- With **wide more than 12 million customers**, HUK-COBURG are the large insurers for private households in Germany.
- **Traditionally offering favourable prices**
- **Largest German motor insurers** with more than 11 million insured vehicles
- **Second rank** in personal liability and home contents insurance, **fifth rank** in **legal expenses insurance**
- **Life insurance:** low costs, low lapse rates, high benefits
- **Health insurance:** most successful newly-founded company in recent decades

Requirements:

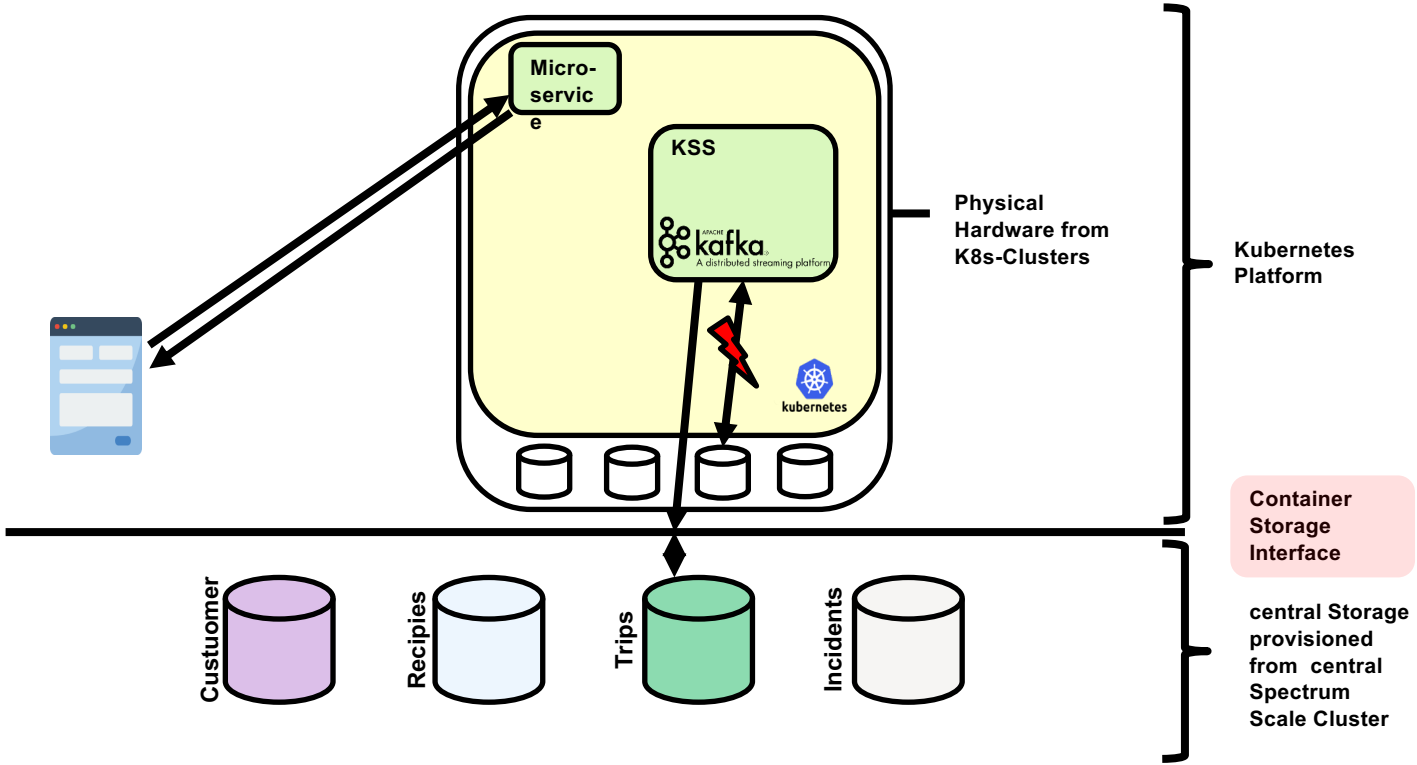
HUK-COBURG
Aus Tradition günstig

- build a Self Service for application development to use kafka as event messaging bus system
- implement a microservice
- generate a enterprise ready storage interface for high availability

Kafka Self-Service – Requirement process



Kafka Self-Service – first CSI-Implementation in HUK



CSI-Driver History

HUK-COBURG
Aus Tradition günstig

Version status csi 2020

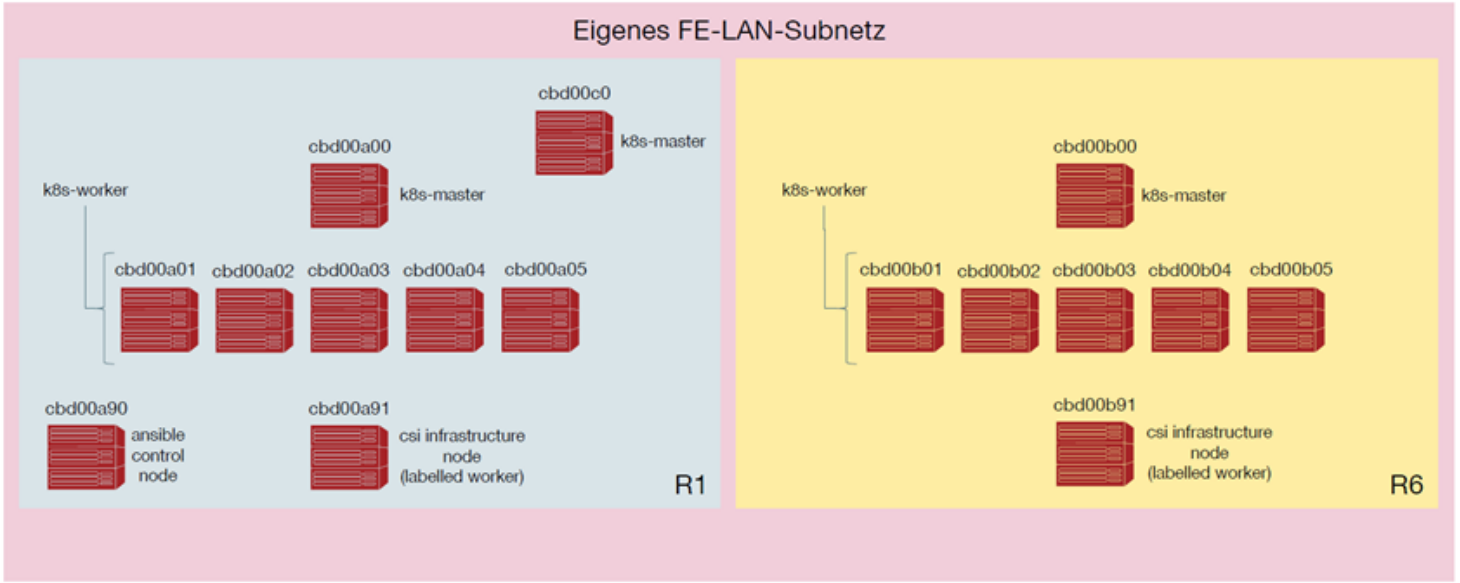
Since 15.01.2020 – csi 1.0.0

Since 14.02.2020 – csi 1.0.1

Since 25.03.2020 – csi 1.1.0

Since 03.07.2020 – csi 2.0.0

Cluster Overview:



Storage-Cluster
remote mounted

Some Issues

HUK-COBURG
Aus Tradition günstig

-Incompatibility with scale 5.0.5 and 5.0.4 in the rest response of GUI ClusterID (fixed with GA)

- Disconnected Internet is already a problem during deployment of new Versions. (enhanced documentation are missing here)
- GUI-HA only with a LoadBalancer currently possible
- Sidecar container where updated

quay.io/k8scsi/csi-attacher:v2.1.1

quay.io/k8scsi/csi-provisioner:v1.5.0 quay.io

/k8scsi/csi-node-driver-registrar:v1.2.0

- k8s etcd asynchronity
- GUI-Node not on same CSI-Node because of Port conflicts

The Next

HUK-COBURG
Aus Tradition günstig

- container native Storage
- Integration of Csi in OpenShift
- GUI Containerisation for a simple HA (hopefully)

More Use Cases:
(based on various customers)

Spectrum Scale use case at automotive client

- Containerized platform to train and test the AI for an ADAS (Advanced Driver Assistance Systems) project
- High-bandwidth data ingest (double-digit TB per day) through Spectrum Scale/ESS
- Sophisticated cloud architecture (see next slide)
- Skilled admins and developers

Spectrum Scale use case
at automotive client – Cluster architecture

IBM Spectrum Scale

ANSIBLE

HashiCorp Terraform

RED HAT OPENSHIFT Container Platform

RedHat OpenShift

RHEL

OpenStack

VMware ESX

x86 Servers + Nvidia GPUs

IBM **Spectrum Scale** / IBM **Elastic Storage Server** (ESS)

Spectrum Scale use case at automotive client – Lessons learned

- To get the Spectrum Scale client up & running
  - Assigned an additional OpenStack network with a dedicated NIC to the VM
    - With OpenStack floating IP the Scale Cluster IP was not visible within VM
  - Adjusted OpenStack security groups to allow inbound traffic to the Spectrum Scale ports
- To ensure that persistent Pods are placed on the Spectrum Scale node(s)
  - Labeled the node and added a nodeSelector to the persistent Pod deployment configs

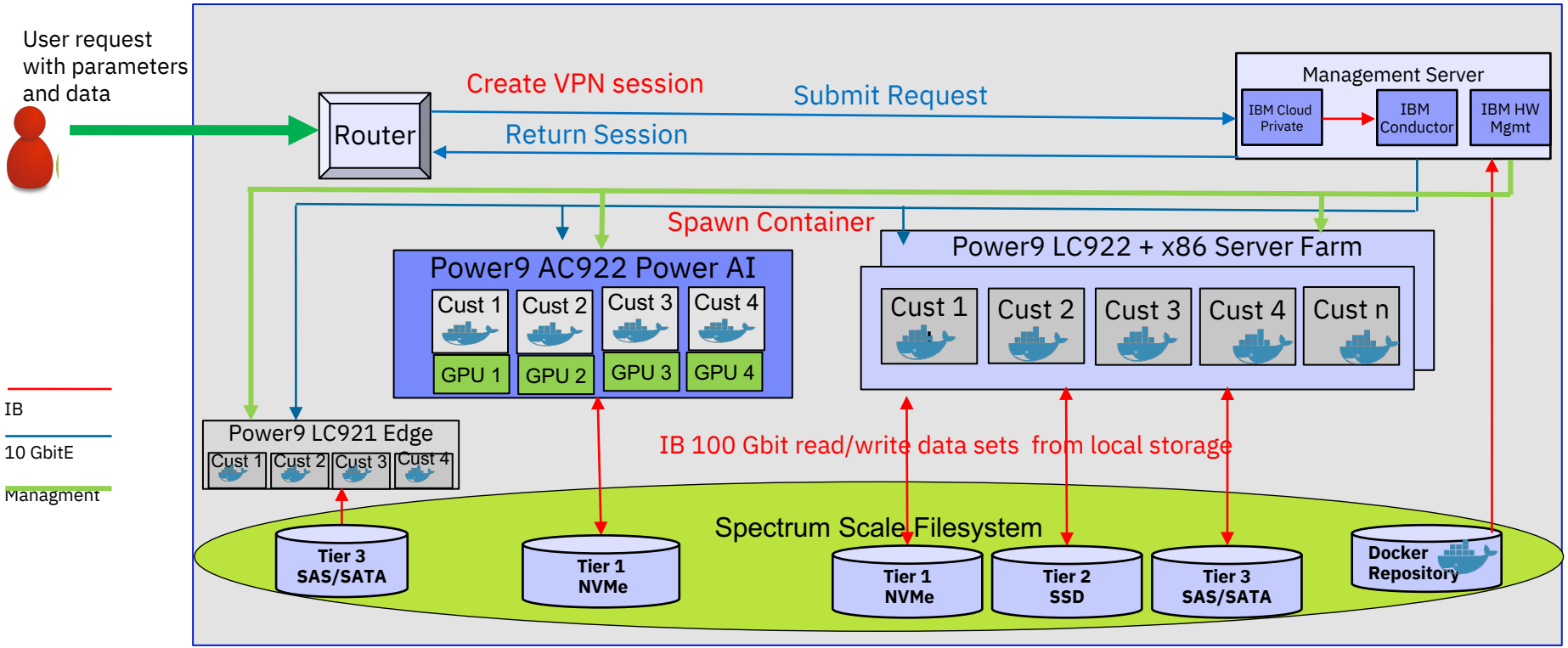
Spectrum Scale use case at automotive client – Lessons learned (cont.)

- Made Storage Enabler for Containers 2.0.0 work
  - Steps are now documented as solution blueprint
- Re-installed with SEC 2.1.0
  - SEC Helm Chart 1.0.1 requires container privilege adjustments for the deployment to succeed
- Existing data can be integrated through additional storage class for existing filesets
  - Created a PVC to re-use an existing Spectrum Scale fileset (ingest directory)
  - Ensure sufficient access rights for the container process

Spectrum Scale use case at automotive client – Summary

- Spectrum Scale is the perfect fit for data intensive engineering
  - High-performance I/O ingest
  - Sub-microseconds access time
  - Containerized AI app data access without duplication or movement
  - Information Lifecycle Management

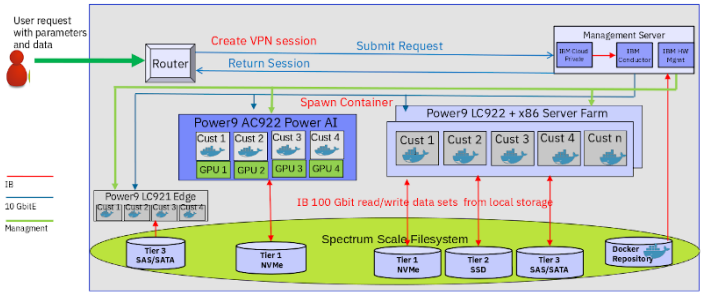
Spectrum Scale use case at Cloud Service Provider: AIaaS*



* Artificial Intelligence as a service

Spectrum Scale use case at CSP: AIaas – Lessons learned

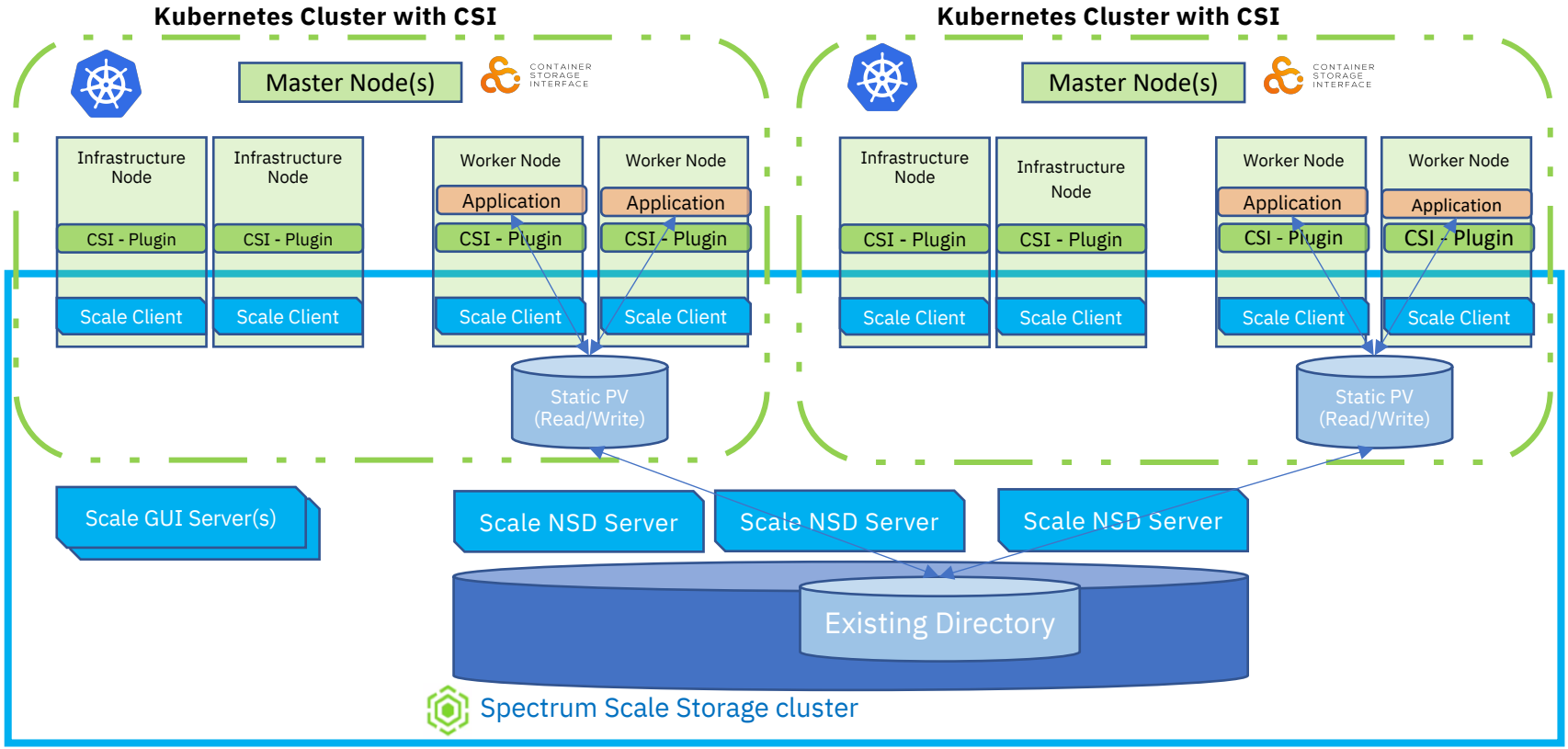
- Multi-tenant isolation and data management is key for the CSP
  - (Semi-)static provisioning for more control over Filesets
  - Pre-created PVs/Filesets with own naming conventions
  - Strict control of per-tenant services (Snapshots, Backup)
- Integration with 3-Tier concept
  - Fileset placement policies
  - One Kubernetes Storage Class per Tier
- CSI Driver works with IBM Cloud Private 3.2.1 on Power and x86-64
  - Leveraging CSI driver lightweight volumes and static provisioning
  - No IBM CSI Driver Support here, CSP is supporting



Spectrum Scale use case
Cloud Computing – Summary

- IBM Spectrum Scale provides state-of-the-art support for cloud and AI use cases through the IBM Spectrum Scale CSI driver
- We are further investing in that area towards container-native

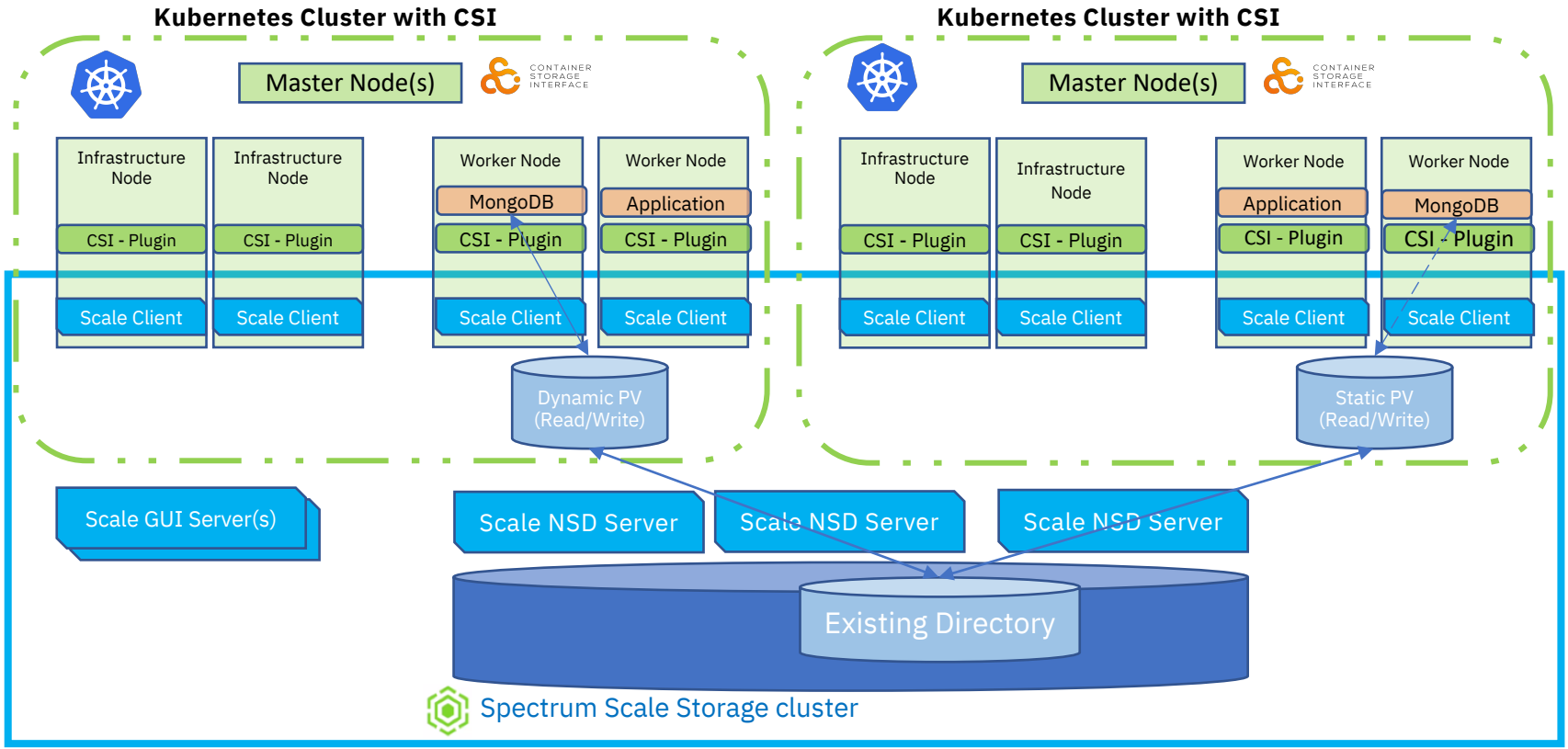
Spectrum Scale use case
Two-site* Disaster Recovery



* Quorum tiebraker needed at 3 site or in separate fire compartment on one site

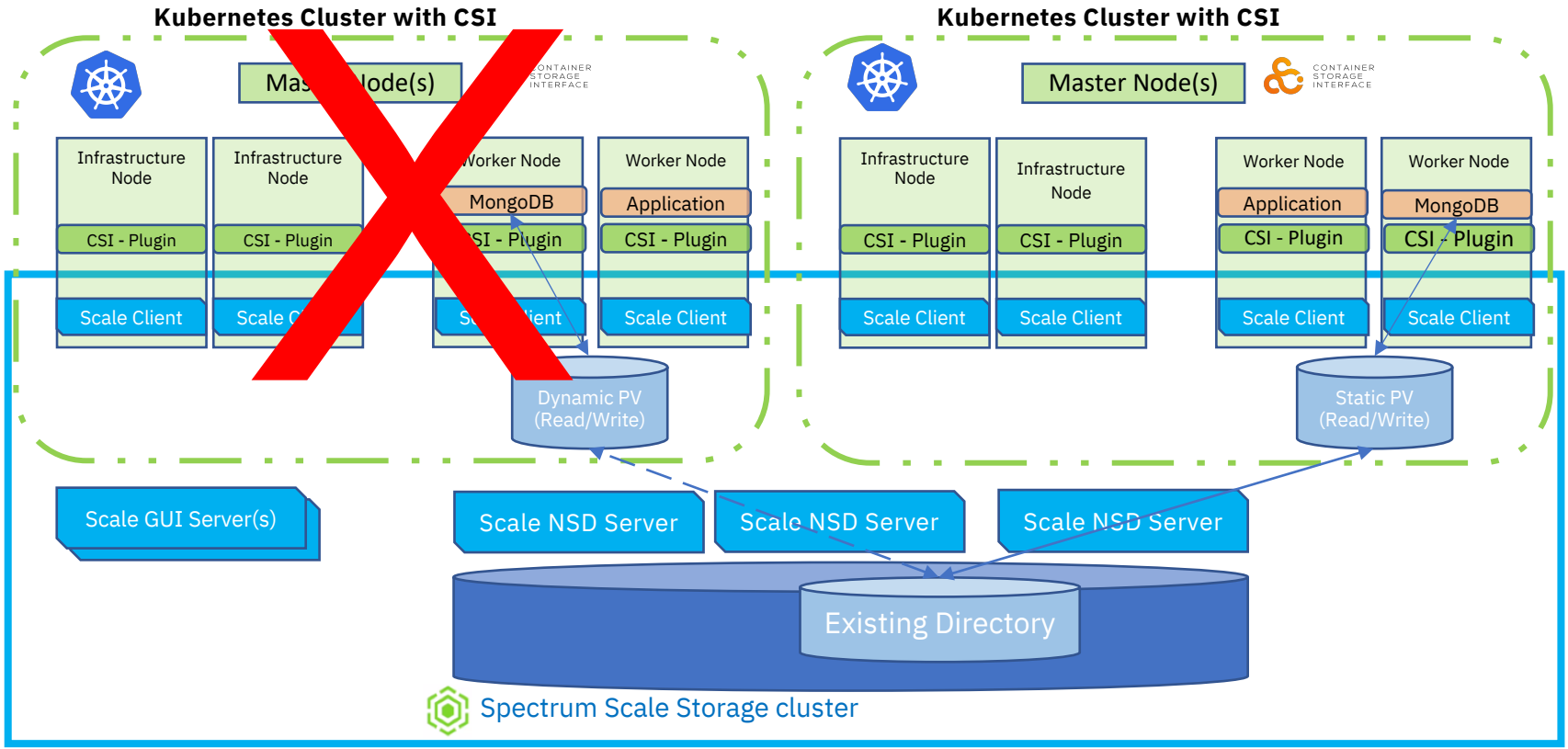
Spectrum Scale two-site* MongoDB DR

IBM Spectrum Scale



* Quorum tiebraker might be needed at 3 site or in separate fire compartment on one site

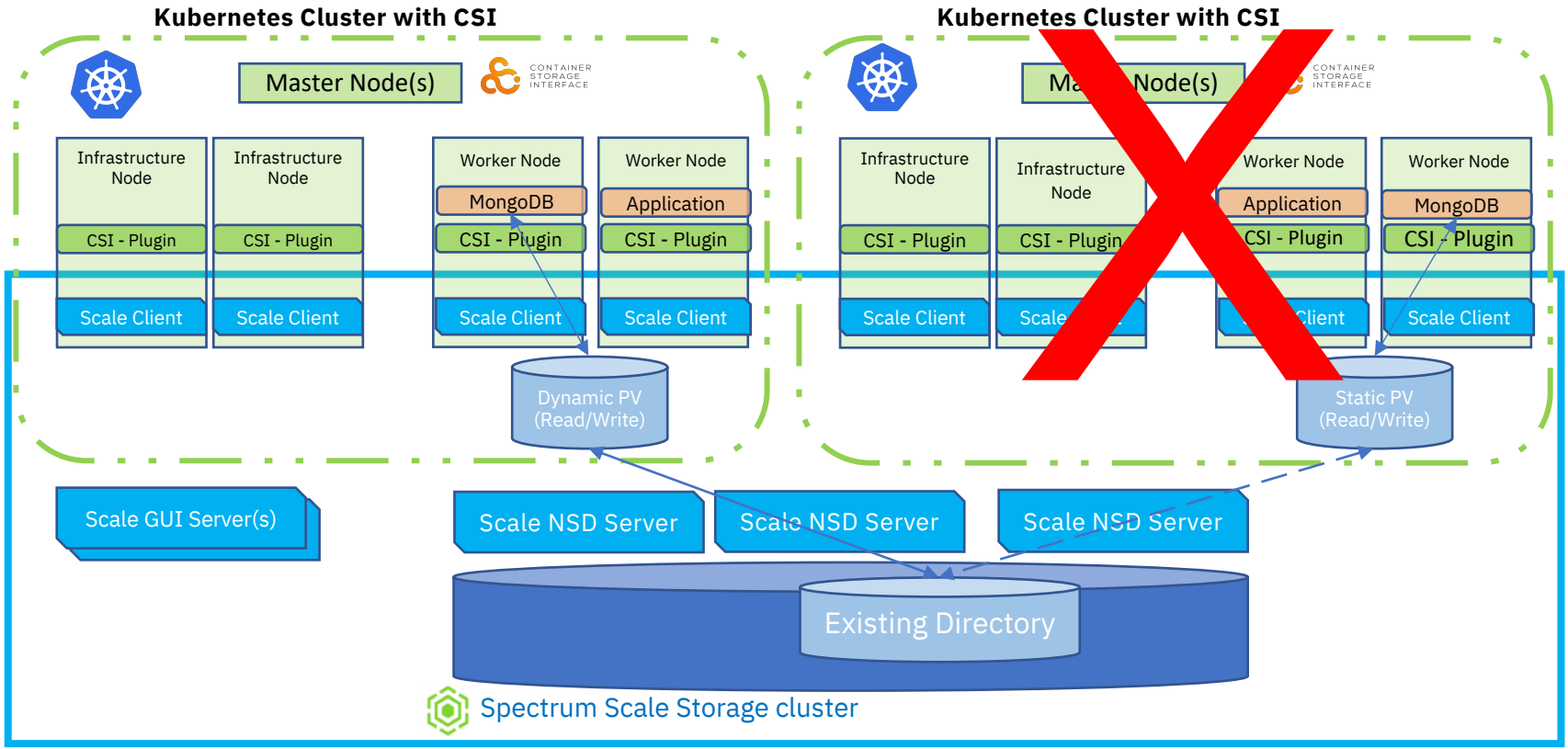
Spectrum Scale two-site* MongoDB DR - failover



* Quorum tiebraker might be needed at 3 site or in separate fire compartment on one site

Spectrum Scale two-site* MongoDB DR - failback

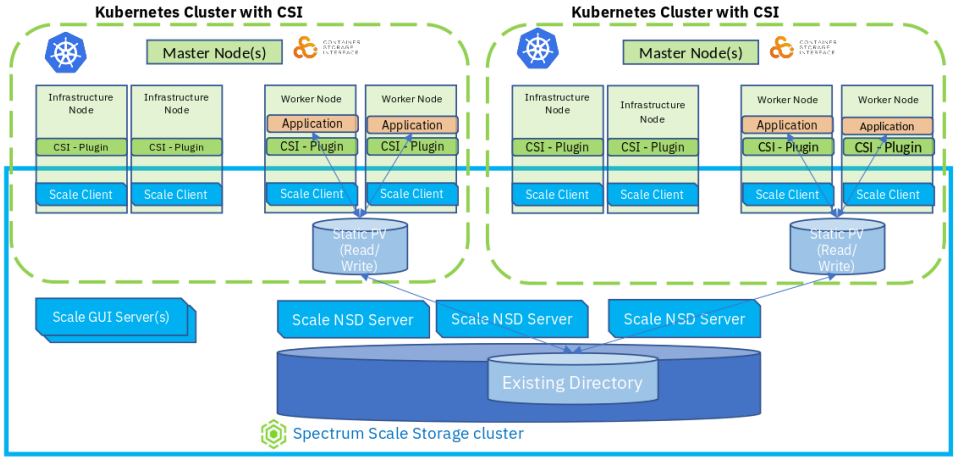
IBM Spectrum Scale



* Quorum tiebraker might be needed at 3 site or in separate fire compartment on one site

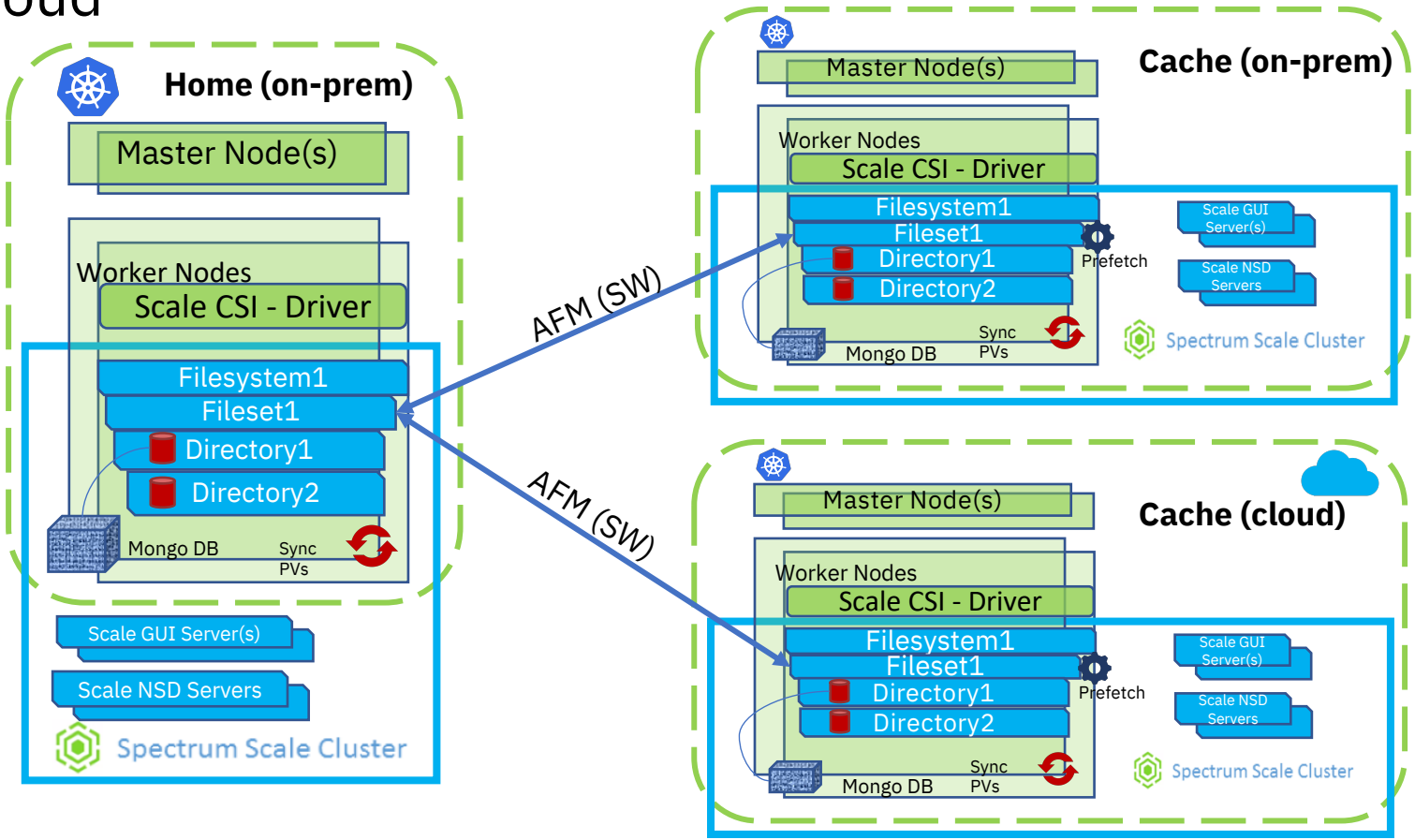
Spectrum Scale use case
Two-site DR – Lessons learned

- Kubernetes clusters do not like to be stretched, etcd is very latency-sensitive
- Spectrum Scale streched cluster can serve as the common data plane
- Additional use case “strong tenant isolation”
  - Separate Kubernetes clusters might have different cluster admins
  - Stronger isolation than Kubernetes namespace separation
  - Still single data plane possible



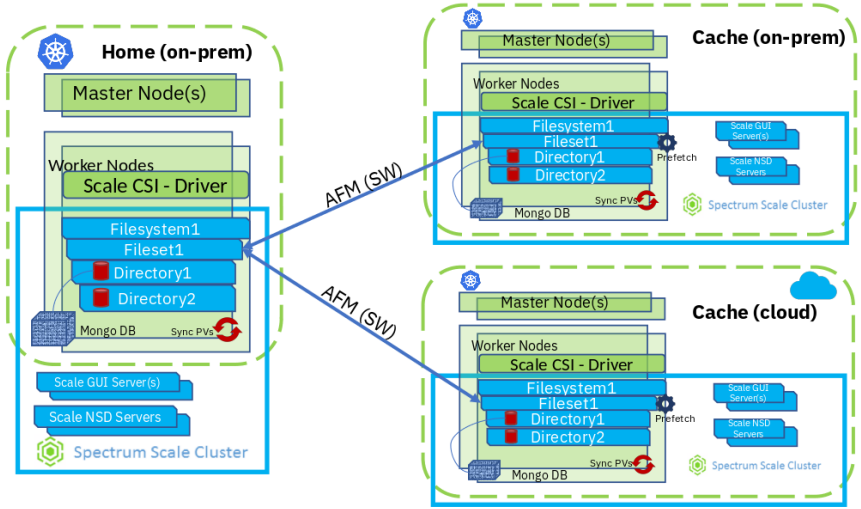
Spectrum Scale use case
Multicloud

IBM Spectrum Scale



Spectrum Scale use case
Multicloud – Lessons learned

- To service workload on cloud
  - Single-writer (home site) only
  - Processed data should be pushed to separate file system
- For DR purposes
  - Only one workload container should be up at given time
  - Independent writer can be used, but monitor cloud data outgoing traffic (EGRESS)
- Can be used with Spectrum Scale on AWS automation
  - Upcoming release will support the required Spectrum Scale GUI
  - Available today on Azure and Oracle cloud



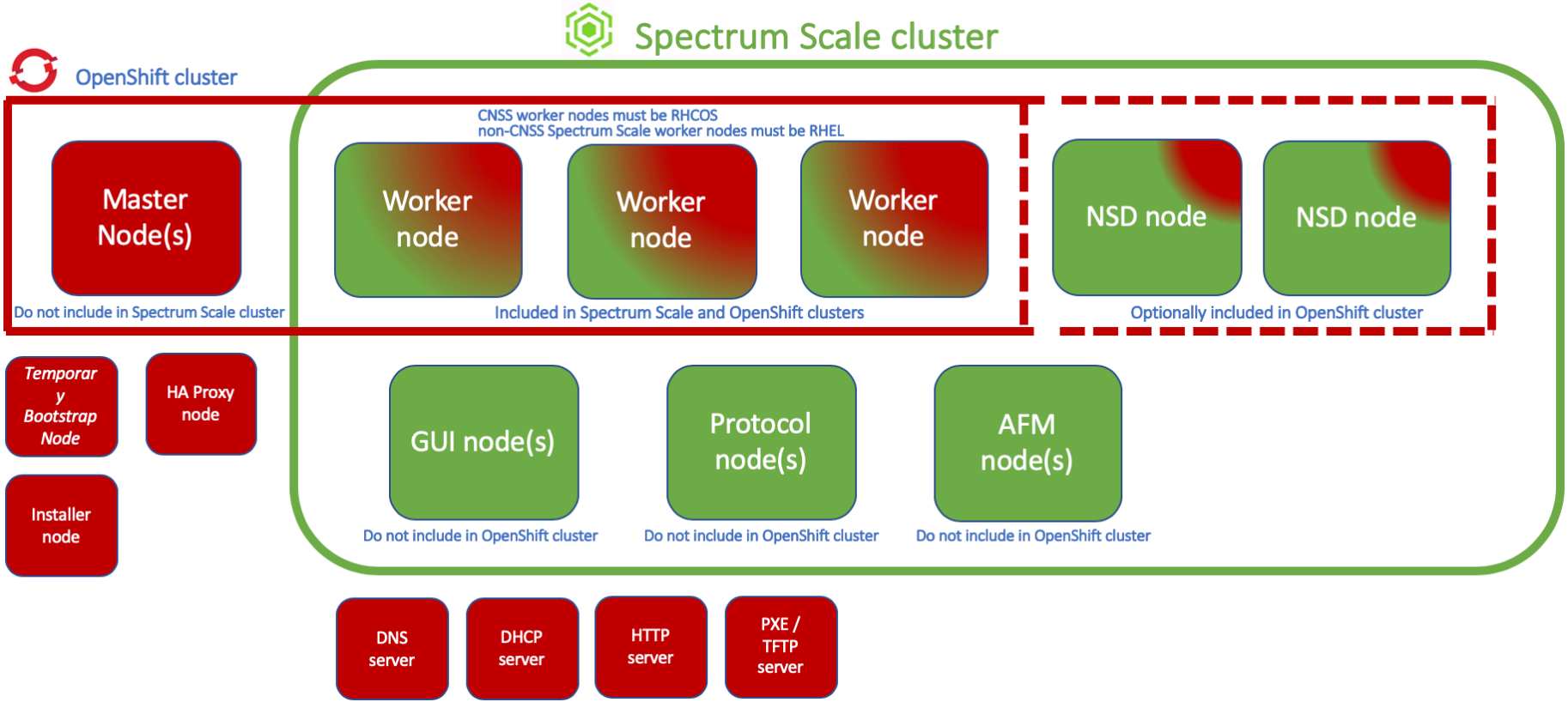
Spectrum Scale use case Multicloud – Summary

- Spectrum Scale is an ideal multicloud data plane
  - Common namespace
  - Consistent cache
  - Transparent data migration

Spectrum Scale On OpenShift

IBM Spectrum Scale

OpenShift - High Level Cluster Configuration

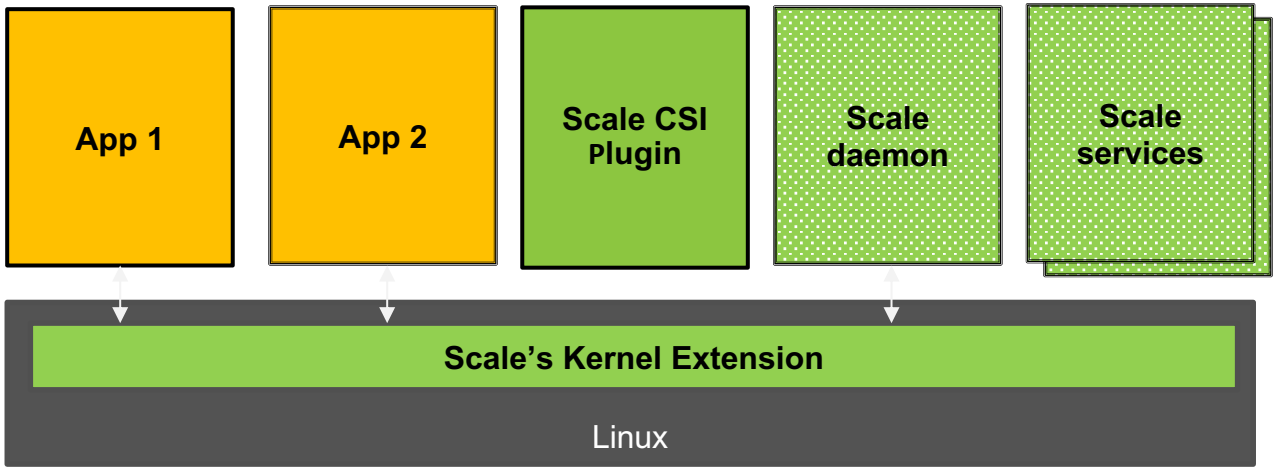


Containerized Spectrum Scale

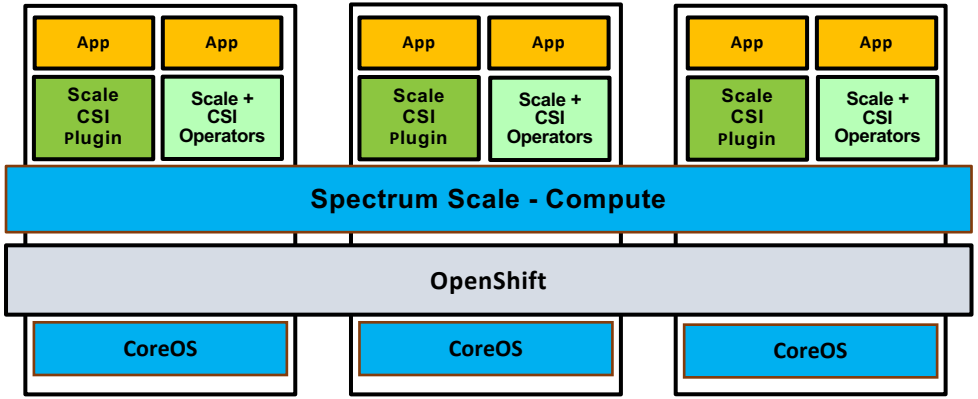
The core of containerized Scale is a daemonset of containers that run on every worker node, load the kernel extension and start the filesystem daemon. This set of containers needs special privileges to load kernel modules, mount file systems, access block devices and access the host network.

Other Scale services like ReST API, GUI, performance data collection, etc. run in separate containers.

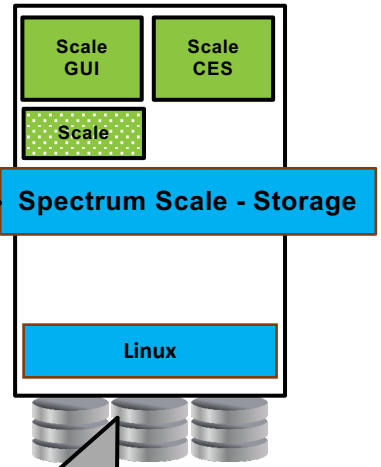
Deployment and operation is implemented via operators.



Cloud-Native IBM Spectrum Scale (CNSS)



Cross Cluster Mount



Cloud Native Spectrum Scale

Scale Containers running in Openshift
Kernel module for CoreOS

Needs „Classic Spectrum Scale“ to provide Storage

„Classic“ Spectrum Scale

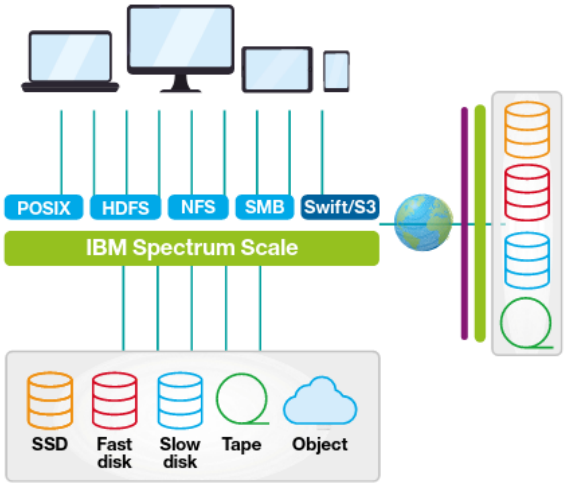
Provides Storage to Cloud Native Spectrum Scale

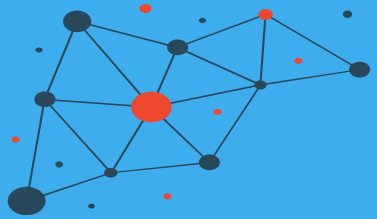
CNSS Beta1 started on 30th July 2020

Demo

Summary and call to action

- Spectrum Scale is a perfect match for Kubernetes and OpenShift container environments featuring
  - Virtually unlimited throughput
  - Exabyte scalability
  - Single namespace eliminating data silos
  - Multicloud readiness
- Talk to your IBM sales contact for an RPQ or write to scale@us.ibm.com





Check https://www.spectrumscaleug.org/experttalks for charts, show notes and upcoming talks

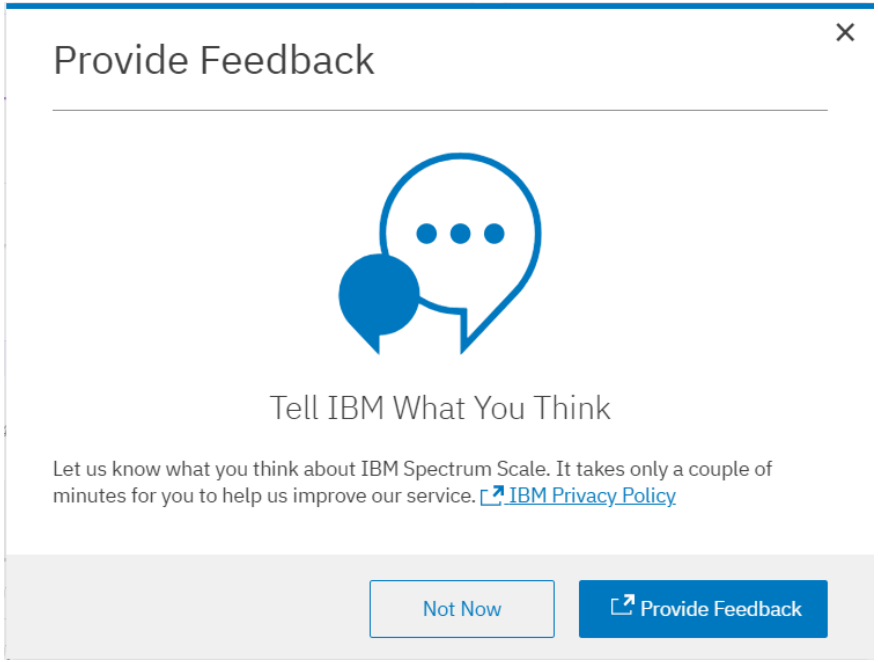
- Past talks:
  - 001: What is new in Spectrum Scale 5.0.5
  - 002: Best practices for building a stretched cluster
  - 003: Strategy update
  - 004: Update on performance enhancements in Spectrum Scale (file create, MMAP, direct IO, ESS 5000)
  - 005: Update on functional enhancements in Spectrum Scale (inode management, vCPU scaling, NUMA considerations)
- Today:
  - Oct 6: Persistent Storage for Kubernetes and OpenShift environments
- Next:
  - Oct 21: Best practices for information lifecycle management (ILM)
  - Nov 4: Multi-node scaling of AI workloads using Nvidia DGX, OpenShift and Spectrum Scale
  - Nov 16: User Meeting at SC20 (Session 1) (more details will follow)
  - Nov 18: User Meeting at SC20 (Session 2) (more details will follow)

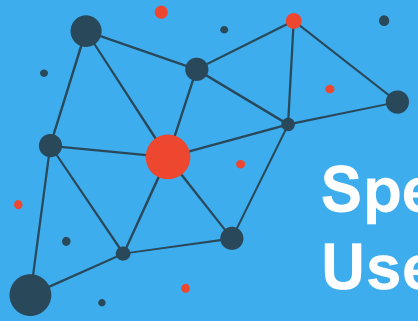
Thank you!

IBM Spectrum Scale

Please help us to improve Spectrum Scale with your feedback

- If you get a survey in email or a popup from the GUI, please respond
- We read every single reply





Spectrum Scale User Group

The Spectrum Scale (GPFS) User Group is free to join and open to all using, interested in using or integrating IBM Spectrum Scale.

The format of the group is as a web community with events held during the year, hosted by our members or by IBM.

See our web page for upcoming events and presentations of past events. Join our conversation via mail and Slack.

www.spectrumscaleug.org