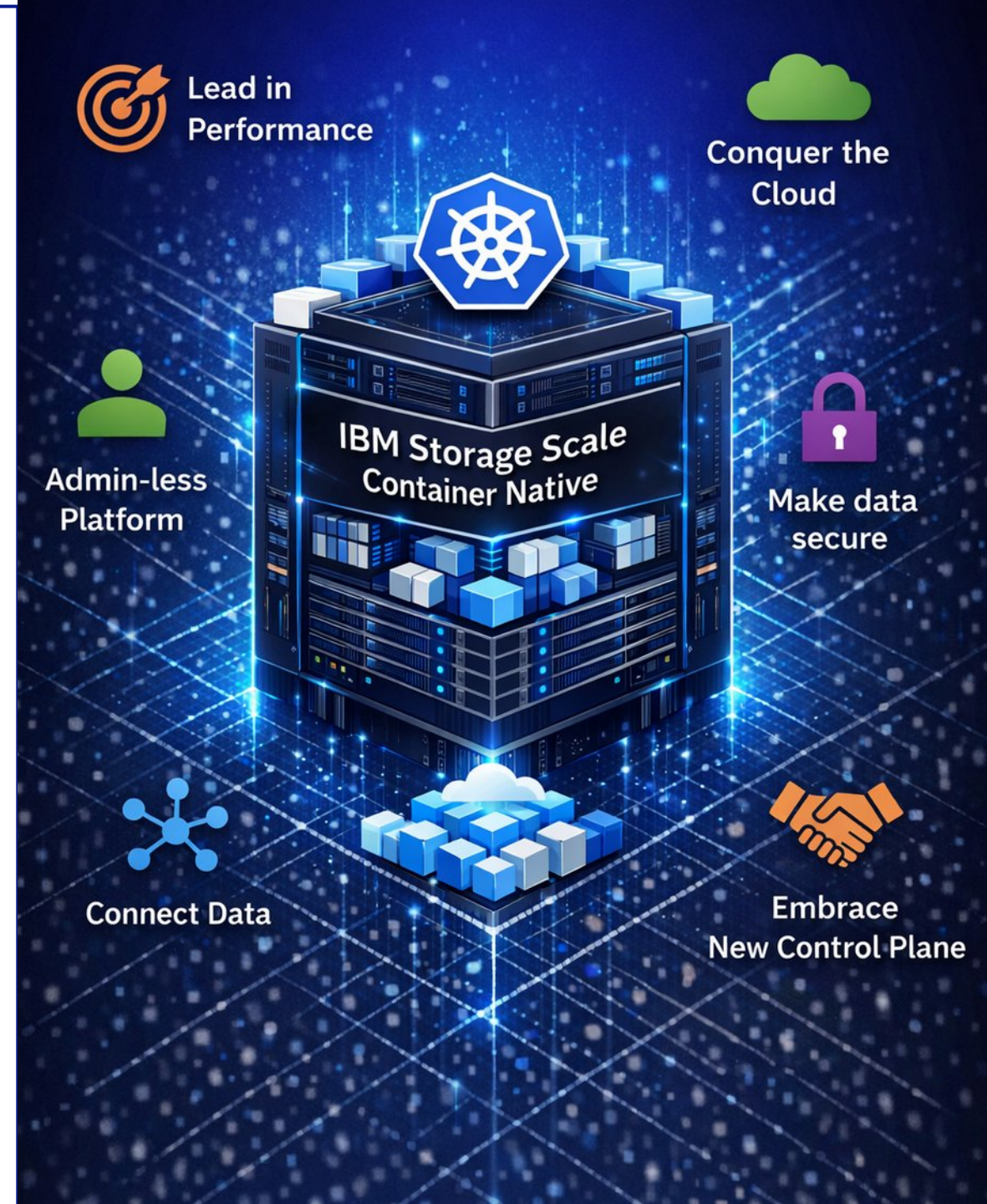


Advancing Declarative Filesystem Management in Kubernetes

Storage Scale UK User Meeting 2026

IBM Innovation Center, London
May 19-20, 2026

Abhishek Jain (abjain39@in.ibm.com)



Container Storage Interface (CSI)

A standard for exposing block and **file storage** to containerized workloads on Container Orchestration Systems (COs) like **Kubernetes**.

Dynamic Provisioning - Create/Delete Volume

Static Provisioning- Dynamic Workflow

Volume Snapshot, Cloning, Expansion

Volume Shallow Copy

Compression, Tiering

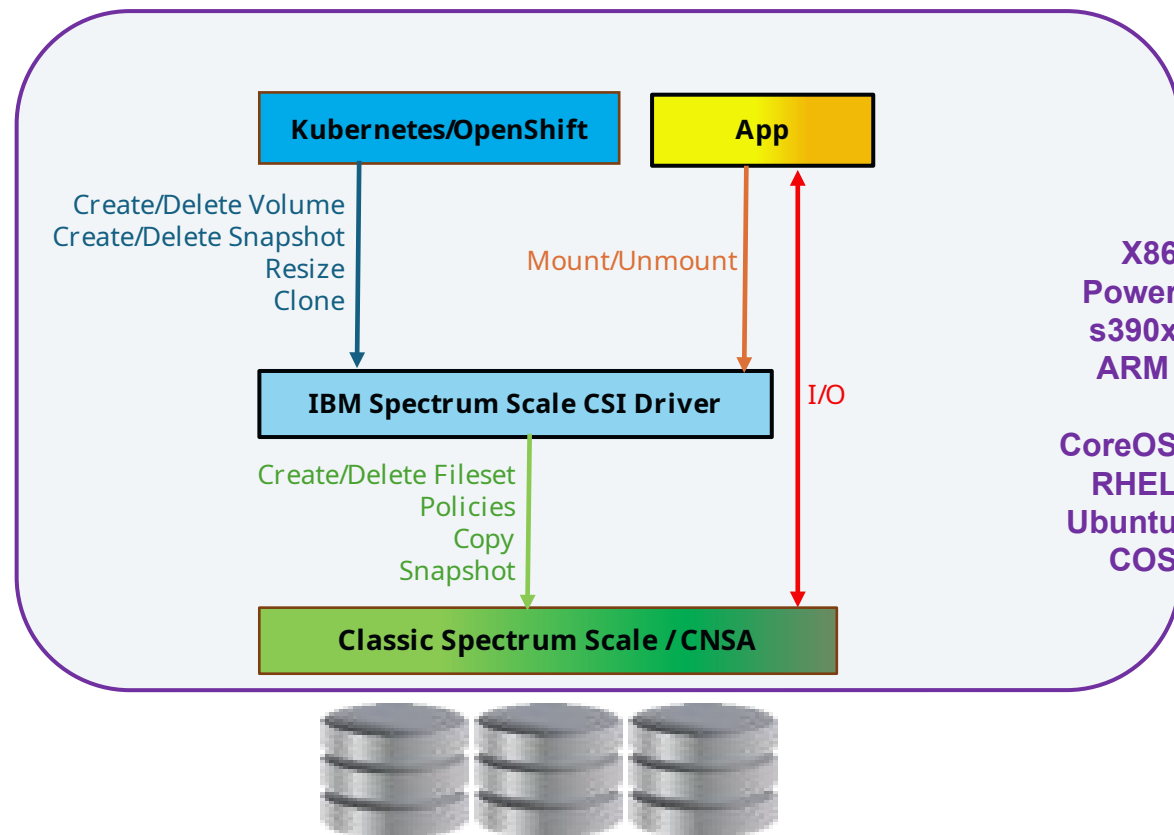
Application Consistent Snapshot

Remote & Local multiple Filesystems

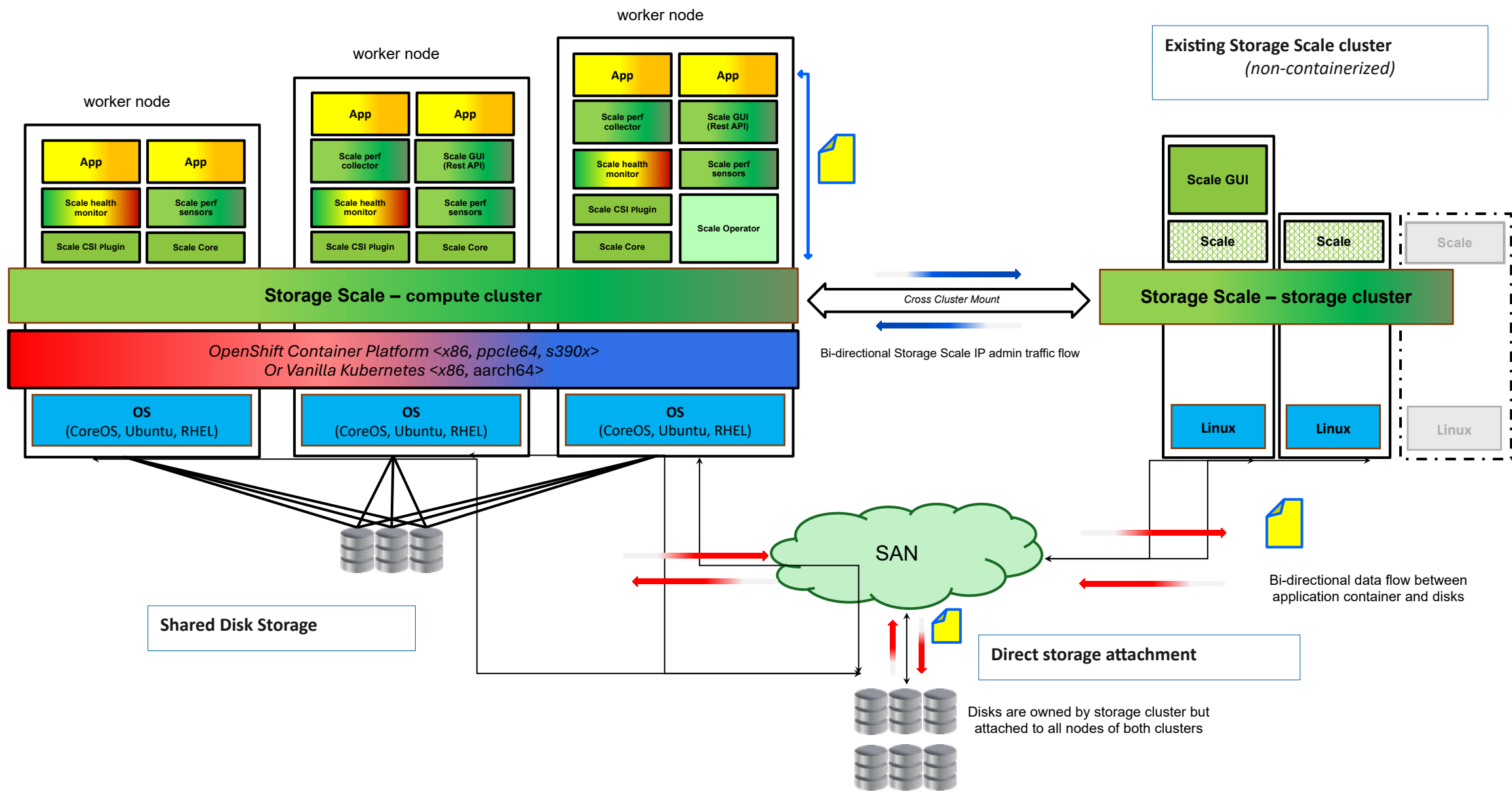
UID/GID & FsGroup Support

Scale GUI HA

Scale Dir/Fileset Based Volume



Container Native Storage Scale



Remote Cluster workflow

```
apiVersion: scale.spectrum.ibm.com/v1beta1
kind: RemoteCluster
metadata:
  labels:
    app.kubernetes.io/instance: ibm-spectrum-scale
    app.kubernetes.io/name: cluster
  name: remotecuster-sample
  namespace: ibm-spectrum-scale
spec:
  contactNodes:
    - storagecluster1node1
    - storagecluster1node2
  gui:
    cacert: cacert-storage-cluster-1
    csiSecretName: csi-remote-mount-storage-cluster-1
  hosts:
    - guihost1.example.com
    - guihost2.example.com
  insecureSkipVerify: false
  secretName: cnsa-remote-mount-storage-cluster-1
```

1

```
apiVersion: scale.spectrum.ibm.com/v1beta1
kind: Filesystem
metadata:
  labels:
    app.kubernetes.io/instance: ibm-spectrum-scale
    app.kubernetes.io/name: cluster
  name: remote-sample
  namespace: ibm-spectrum-scale
spec:
  remote:
    cluster: remotecuster-sample
    fs: fs1
  seLinuxOptions: {}
```

2

```
apiVersion: scale.spectrum.ibm.com/v1beta1
kind: LocalDisk
metadata:
  labels:
    app.kubernetes.io/instance: ibm-spectrum-scale
    app.kubernetes.io/name: cluster
  name: disk0
  namespace: ibm-spectrum-scale
spec:
  device: /dev/sdb
  node: worker0.example.com
  nodeConnectionSelector:
    matchExpressions:
      - key: kubernetes.io/hostname
        operator: In
        values:
          - worker0.example.com
          - worker1.example.com
          - worker2.example.com
```

1

Shared Disk workflow

```
apiVersion: scale.spectrum.ibm.com/v1beta1
kind: Filesystem
metadata:
  labels:
    app.kubernetes.io/instance: ibm-spectrum-scale
    app.kubernetes.io/name: cluster
  name: local-sample
  namespace: ibm-spectrum-scale
spec:
  local:
    blockSize: 4M
    pools:
      - disks:
          - disk0
          - disk1
          - disk2
        name: system
        replication: 1-way
        type: shared
      seLinuxOptions: {}
```

2

```
[root@ocp ~]# oc get ld -n ibm-spectrum-scale
```

NAMESPACE	NAME	TYPE	READY	USED	AVAILABLE	FILESYSTEM	SIZE	AGE
ibm-spectrum-scale	disk0	shared	True	True	True	local-sample	50.0 GiB	10d
ibm-spectrum-scale	disk1	shared	True	True	True	local-sample	50.0 GiB	10d
ibm-spectrum-scale	disk2	shared	True	True	True	local-sample	50.0 GiB	10d

3

Kubernetes Architecture

- RHEL OS
 - X86 , Power, ARM
 - 9.x, 10.x
- Ubuntu
 - X86
 - 22.04,24.04

Kubernetes Architecture

- RHEL OS
 - X86,ARM
 - 9.x, 10.x
 - 1.32,1.33, 1.34
- Ubuntu
 - X86
 - 22.04,24.04
 - 1.32,1.33, 1.34

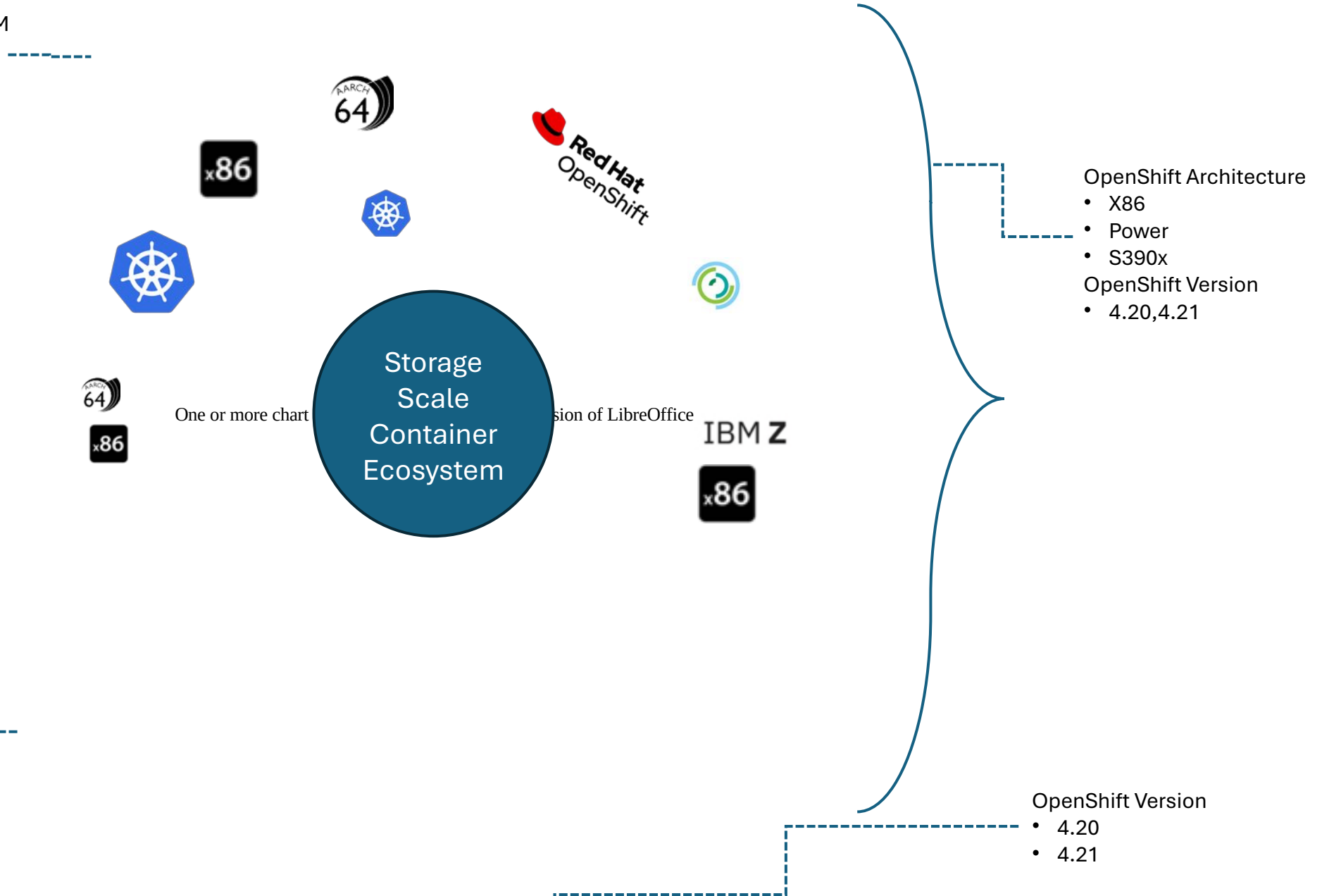
Tech Preview

Google Container Optimized OS

- COS 121, 125

Kubernetes

- 1.32,1.33, 1.34



Release Updates & improvements

Release 6.0.0

Ease of Use & Operations

- Consolidate CSI Operator into CNSA Operator
- Removal of primary filesystem and fileset dependency.
- Migration from IBM standalone CSI driver to CNSA on vanilla Kubernetes

High Performance Network

- High-speed, low-latency networking support with NVIDIA MOFED driver integration
- GPU Direct Storage (GDS) support via RoCE or InfiniBand

Enhancements

- CSI quota fixes for small volumes
- Shared disk technology improvements
- Automated deadlock detection and data collection
- Enable linking fileset at customized path
- CNSA deployment support for CoreWeave (AI factory) environment

Platform & OS Support

- OCP 4.20, 4.21(4.17) & k8s 1.34 (1.30)
- Red Hat Enterprise Linux 9.7
- aarch64 (ARM64) CNSA support for vanilla k8s
- New COS 121 Milestone GKE Support (Tech Preview)

Release 6.0.1

Ease of Use & Operations

- Container application-aware CNSA upgrades
- Single Namespace CNSA deployment
- Automated Kubernetes Network Policy Support
- Declarative GPFS tuning enhancement for node and cluster level parameter
- Automated management for node profile to configure memory and CPU for CNSA resource management
- Automated GPFS quorum management

Enhancements

- Data caching persistent volume capabilities with NFS backend, S3 is already supported
- Pod level bonding that uses Container Network Interface (CNI) overlay networks.
- Support for new GPFS kernel module management method (Kernel Module Management) support
 - Migration from MCO based kernel module management to new KMM based module management
 - Support for building & signing kernel module for Secure Boot
- Fusion Access for SAN Support :
 - Introducing disaster recovery with MetroDR support
 - Improve efficiency with enhanced volume cloning for disk volumes

Platform & OS Support

- OCP 4.21(4.20) & k8s 1.34 (1.32)
- Support for Red Hat Enterprise Linux 10.0 ,10.1 & 9.8
- New COS 125 Milestone GKE Support (Tech Preview)

CNSA-CSI Integration Strategy

Removal of Primary filesystem/fileset

Background :

In releases prior to **CNSA v6.0.0**, IBM Storage Scale CSI Driver used to create **soft link** in Primary Filesystem that points to the actual volume directories in same or another filesystem , which leads to

- Dependency of dedicated primary filesystem and fileset
- Downtime to primary filesystem cluster meaning, data access lost to all the container applications.

Spec:

Attacher Node Selector:

Key: scale

Value: true

Clusters:

Cacert: storage-cacert

Id: 5980831202266627691

Primary: // not required

Primary Fs: fs1 // not required

Rest API:

Gui Host: 10.11.126.148

Secrets: guisecret

Secure Ssl Mode: true

Consistency Group Prefix: 20ac81f2-96a9-4351-a102-7579acae04ac

Local Scale Cluster: 5980831202266627691

One Operator (CNSA + CSI)

Background:

In releases prior to CNSA v6.0.0, CSI Operator and CNSA Operator exist as separately into respective namespaces, each maintaining independent controllers and managing resources. As CNSA evolves to primarily support CSI, maintaining two separate operator stacks which leads to

- increases operational complexity
- Debugging issues and deployment overhead.

With v6.0.0, One Operator (CNSA + CSI) resulting in:

- Simplified installation and upgrade process.
- Single point of reconciliation for CNSA and CSI resources.
- Better maintainability and testing.
- Reduced debug efforts.
- Reduced chances of conflict during upgrades or runtime.

Single Namespace CNSA Deployment

Background:

In releases prior to CNSA v6.0.1, IBM Storage Scale container native components were deployed across multiple namespaces such as:

- [ibm-spectrum-scale-operator](#)
- [ibm-spectrum-scale-csi](#)
- [ibm-spectrum-scale-dns](#)
- [ibm-spectrum-scale](#)

With v6.0.1, One Namespace for Full CNSA deployment resulting in:

- Simplified management and debugging
- Consolidated Network Policy
- Enabled Deployment using Red hat OLM in Fusion Data Foundation Product.

Container Application Awareness: Simplified Upgrade

Prior to 6.0.1.0 release

- CNSA operator **evicts all pods on node** except DaemonSet pods before deleting core pod

With 6.0.1.0

- Container Operator **evicts only application pods that use Scale-PVs** before deleting core pod
- In rare cases operator evicts all pods on node and reboots node
 - Rare case: mmshutdown fails to unload kernel module
- NodeDraining event in Daemon CR shows evicted pods

```
Normal NodeDraining 64s Daemon Node worker8.example.com drained by Storage Scale operator. Evicting pod test/nginx-pod-6.
```

Benefits:

- Optimized upgrade performance !
- No impact on workloads that do not use Scale-PVs !

Migration from CSI to Storage Scale Container Native

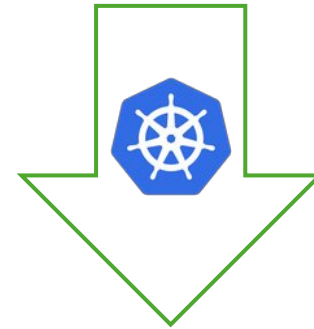
An existing IBM Storage Scale Container Storage Interface Driver cluster can be migrated into a new CNSA cluster.

Outcome :

- The native installation of IBM Storage Scale CSI (standalone CSI) need to be removed from the K8s nodes
- All persistent volumes and volume claims are transferred (using the migration script) into CNSA.

Refer : <https://www.ibm.com/docs/en/scalecontainernative/6.0.0?topic=kubernetes-migration-from-storage-scale-csi>

IBM Storage Scale Container Native Interface (CSI)



IBM Storage Scale Container Native

GPFS Kernel Module Management (RedHat OpenShift)

Cluster Creation with MCO

Prerequisite

- Apply Machine Config Operator configuration

```
apiVersion: machineconfiguration.openshift.io/v1
kind: MachineConfig
metadata:
  labels:
    machineconfiguration.openshift.io/role: worker
  name: 00-worker-ibm-spectrum-scale-kernel-devel
spec:
  config:
    ignition:
      version: 3.2.0
    extensions:
      - kernel-devel
```

Pros:

- No External Container Dependency
- GPFS kernel module built automatically

Cons:

- Reboots all OCP nodes
 - Hence impact on running container workload
- GPFS kernel module built on each pod/node restart
 - Cluster creation and pod startup is slow
- No Secure Boot support

Cluster Creation with KMM

Prerequisite

- Apply Kernel Module Management Operator configuration
- Enable KMM section in cluster.yaml

```
gpfsModuleManagement:
  kmm:
    imageRepository:
      registry: quay.io
      repo: abcuser/gpfs-kmod
      registrySecret: quayio-secret
    moduleSigning:
      certSecret: my-signing-key-pub
      keySecret: my-signing-key
```

Pros:

- Does not reboot OCP nodes, hence no impact on running container workload
- GPFS Kernel module is built once, until OCP or GPFS is changed.
- Kernel module can be signed with secure boot keys

Cons:

- Internal/External container registry dependency to push the dynamically built container image which container kernel module

Serviceability Enhancements (Collecting Kernel Threads)

IBM Storage Scale container native 6.0.0, enhances support for automated debugging workflows, with a key focus on deadlock identification and resolution.

Previous Approach (Pre-6.0.0)

- Required manual steps:
 - Apply custom YAMLS
 - Modify deployment to fetch and run special container images
 - Restart pods to enable debug mode
- Time-consuming during critical failures (e.g., deadlocks)

What's New?

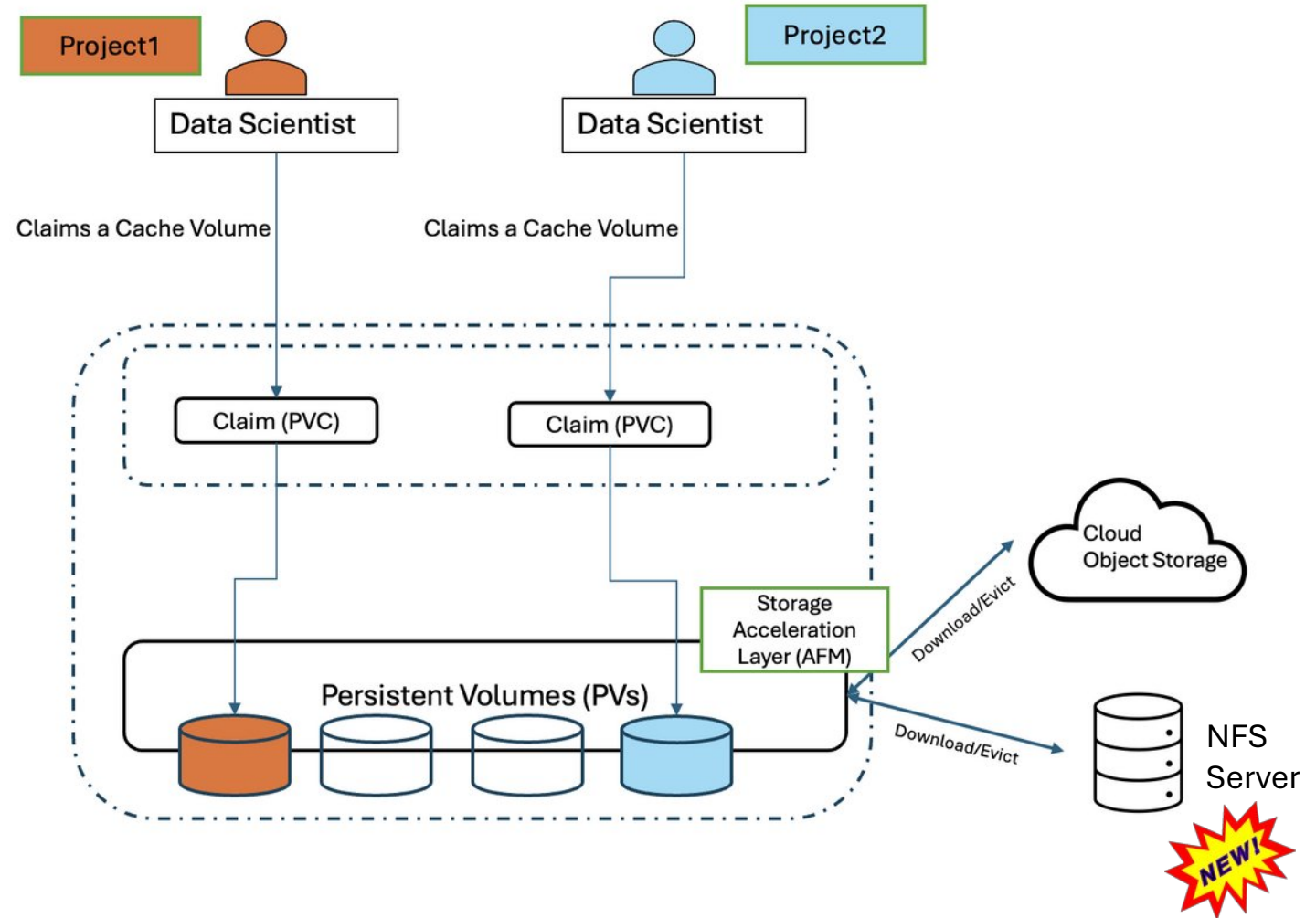
- Just enable **threadDeadlockCapture = true**
- Now when deadlock is detected, deadlock related logs will be auto collected similar to classic Scale
- This feature is disabled by default.

Enable it directly in cluster CR with

```
kubectl patch cluster.scale.spectrum.ibm.com <namespace> --type='json' -p='[{"op": "add",  
"path": "/spec/debugConfig", "value": {"threadDeadlockCapture": true}}]'
```

Cache Volume

- Dynamically provisioned cache volumes from Object storage
- Supported cache modes:
 - Readonly
 - Parallel
 - Exclusive
 - Detached
- CSI will provision an AFM cache fileset and attaches as `PersistentVolume` to the application pod
- `CacheVolumeOperation` allows to automate download and eviction of data on `PersistentVolume`
- CSI `VolumeAttributeClass` to change AFM volume's tuning attributes



Cache Volume Workflow

```
kind: StorageClass
apiVersion: storage.k8s.io/v1
metadata:
  name: sc-scale-cache
provisioner: spectrumscale.csi.ibm.com
parameters:
  volBackendFs: "fs0"
  volumeType: "cache"
  csi.storage.k8s.io/provisioner-secret-name: ${pvc.name}-secret
  csi.storage.k8s.io/provisioner-secret-namespace: ${pvc.namespace}
reclaimPolicy: Delete
allowVolumeExpansion: true
```

```
graph TD
    1[1] --> 2[2]
    2 --> 4[4]
    4 --> 1
    3[3] --> 1
```

1

```
kubectl create secret generic cache-pvc1-secret -n project1 \
--from-literal=nfsServers=remote-node12.example.com
--from-literal=nfsPath="/home/user/data"
```

3

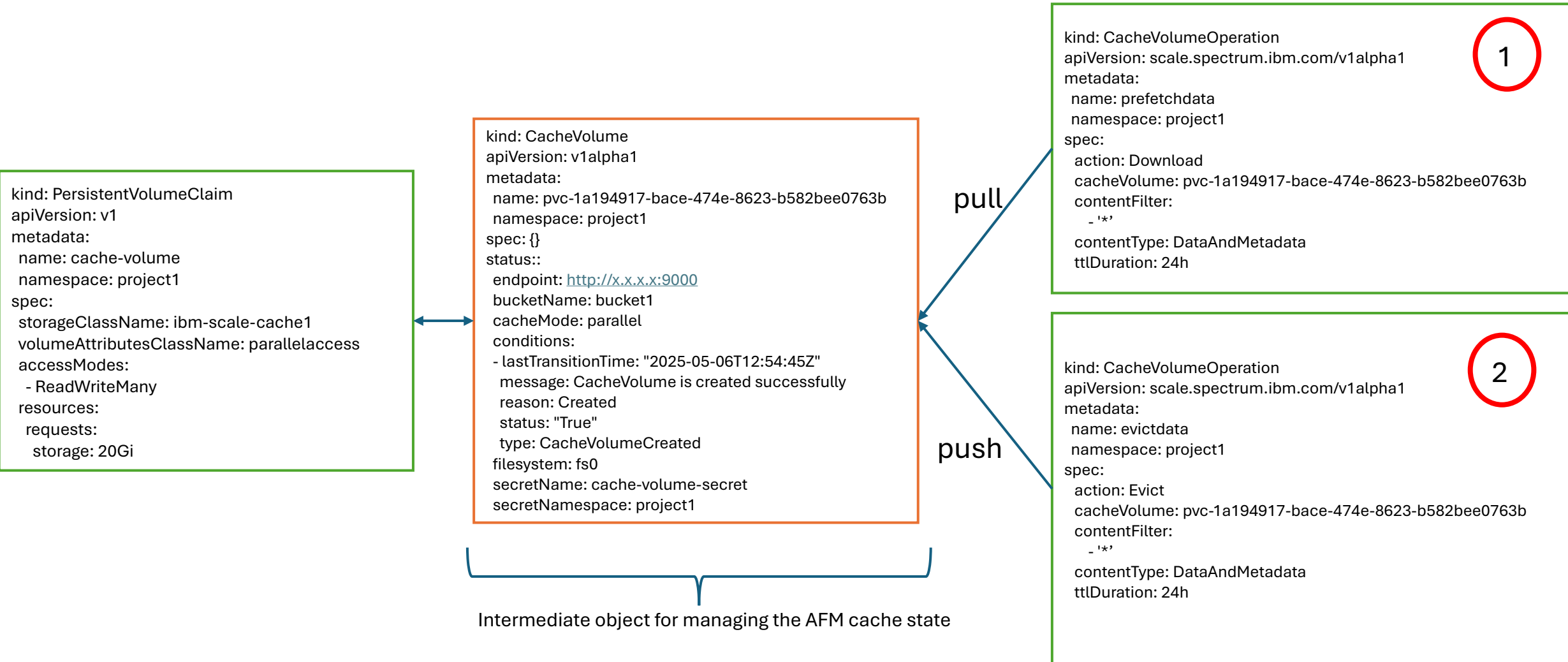
```
kind: PersistentVolumeClaim
apiVersion: v1
metadata:
  name: cache-volume
  namespace: project1
spec:
  storageClassName: sc-scale-cache
  volumeAttributesClassName: parallelaccess
  accessModes:
    - ReadWriteMany
  resources:
    requests:
      storage: 20Gi
```

2

```
kind: VolumeAttributesClass
apiVersion: storage.k8s.io/v1
metadata:
  name: parallelaccess
driverName: spectrumscale.csi.ibm.com
parameters:
  afmNumFlushThreads: "16"
  afmDirLookupRefreshInterval: 60
```

4

Cache Volume Workflow (Cont..)

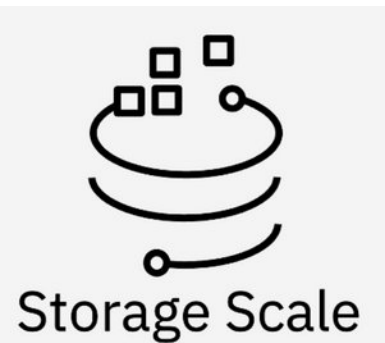


Outlook

- Embrace new scale control plane
- Reinforce k8s integration and admin-less operation
- Must-gather improvement for Improved Diagnostics and Supportability
- Enterprise features
 - Regional Disaster Recovery
 - Encryption
 - Resiliency
 - Multi-Tenancy
- FIPs Support with OCP and Kubernetes
- Expand support matrix (more cloud providers)
- Scale-NFS-CSI driver
- Single node cluster



Thank you for using Storage Scale



Storage Scale

