SNIA DEVELOPER CONFERENCE
SDC 25
By Developers FOR Developers

Hyatt Regency Santa Clara, CA
September 15-17, 2025

# Storage for AI in Public Clouds: Case Study of Vela in IBM Cloud

Vasily Tarasov, Principal Research Scientist, IBM Research

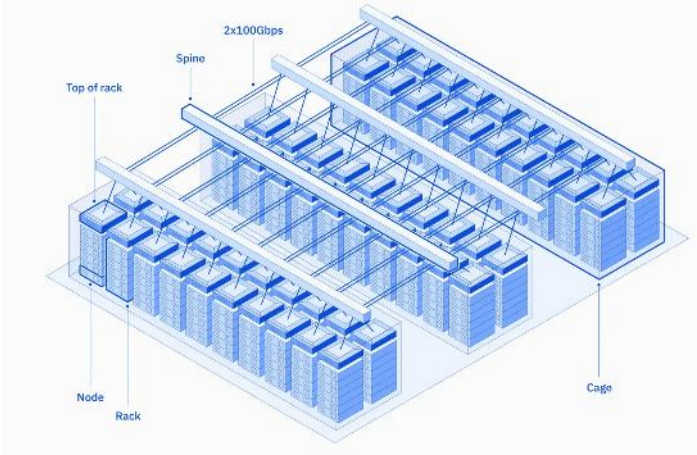
# Agenda

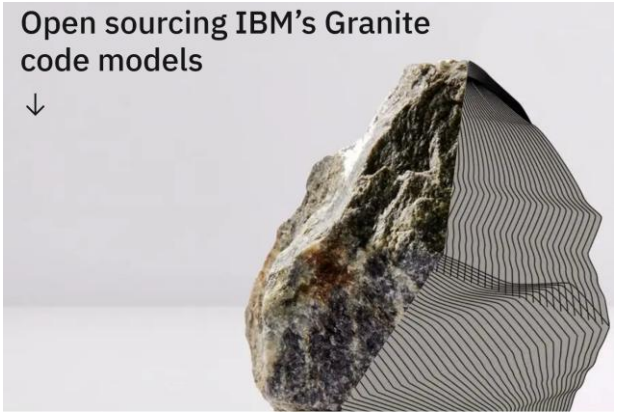
1. Vela(s): cloud-native cluster(s) for large model trainings
2. Data access in model training workloads
3. Storage for training
    1. Naïve approach and problems
    2. Elastic file system cache over object storage
4. Self-serviced caching volumes
5. Data prefetch and scheduler integration
6. As-a-service solution

# Vela(s): Cloud-native model training cluster(s)

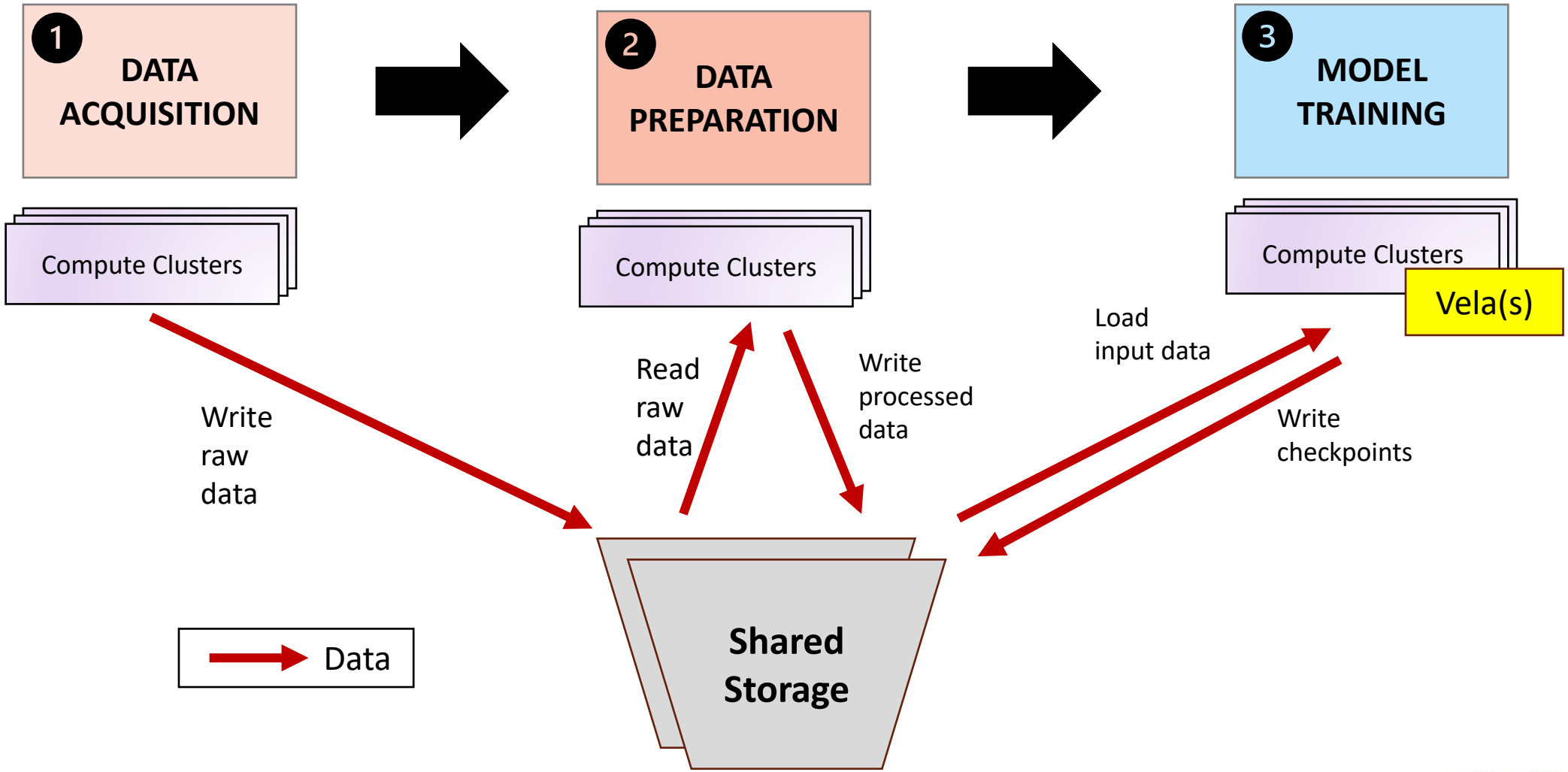
- IBM Research needs infrastructure to train ML models
  - Large Language Models (LLMs) with 100s of billions of parameters
- Can we build a **cloud-native** training cluster?
  - Instead of traditional on-prem HPC data center
- Of-the-shelf GPU-rich host servers added to IBM Cloud
  - 8× NVIDIA A100, H100, H200, …
- KVM-based Virtual Machines as building blocks
- Standard Ethernet networking
  - RoCE for GPU-GPU communication
- Red Hat OpenShift (OCP) for resources and training job management
  - MLBatch / Kueue job queuing and quota management system https://github.com/project-codeflare/mlbatch/blob/main/CODEFLARE.md#mlbatch-for-codeflare-users
- Success! IBM Granite models:
  - https://huggingface.co/ibm-granite



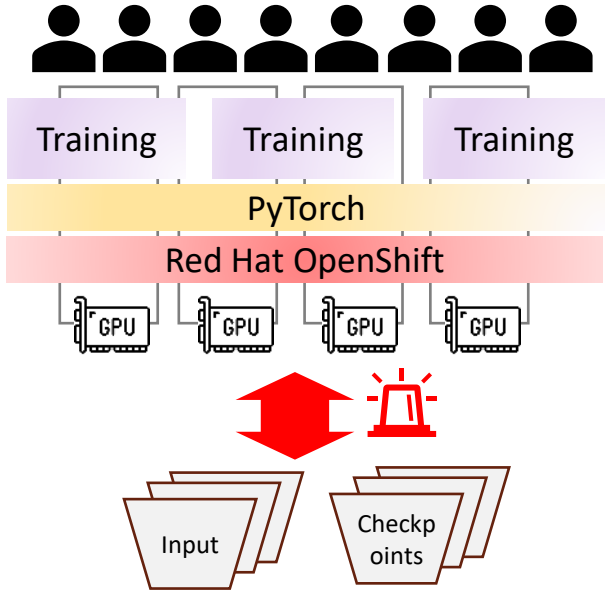
# Data Access in AI training workflows

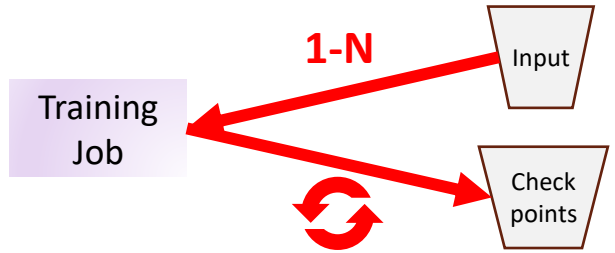


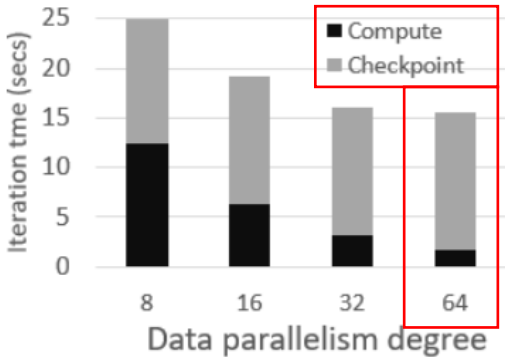
# Workload and storage for model training

- Multiple users run concurrent training jobs
  - Some jobs for weeks – months
- Native shared storage services in IBM Cloud
  - A. IBM Cloud Object Storage (COS)
  - B. ~~IBM Cloud File Storage (NFS) – 1GB/s per file share~~
- Typical training job uses
  - 1 bucket for input data – read all input data a few times
  - 1 bucket for checkpoints – **many periodic writes, infrequent reads**
- Everything gets bigger
  - Input data (more tokens): 1TB → 20TB
  - Checkpoints (larger models): 100GB → 1TB
  - Cluster size (100s of GPUs → 1000+ GPUs)
- **Problems**
  - × Slow checkpointing
    - × High variability in performance over time
  - × Storage backend "overload"
  - × Slow training data loading
  - × Lack of file system semantics



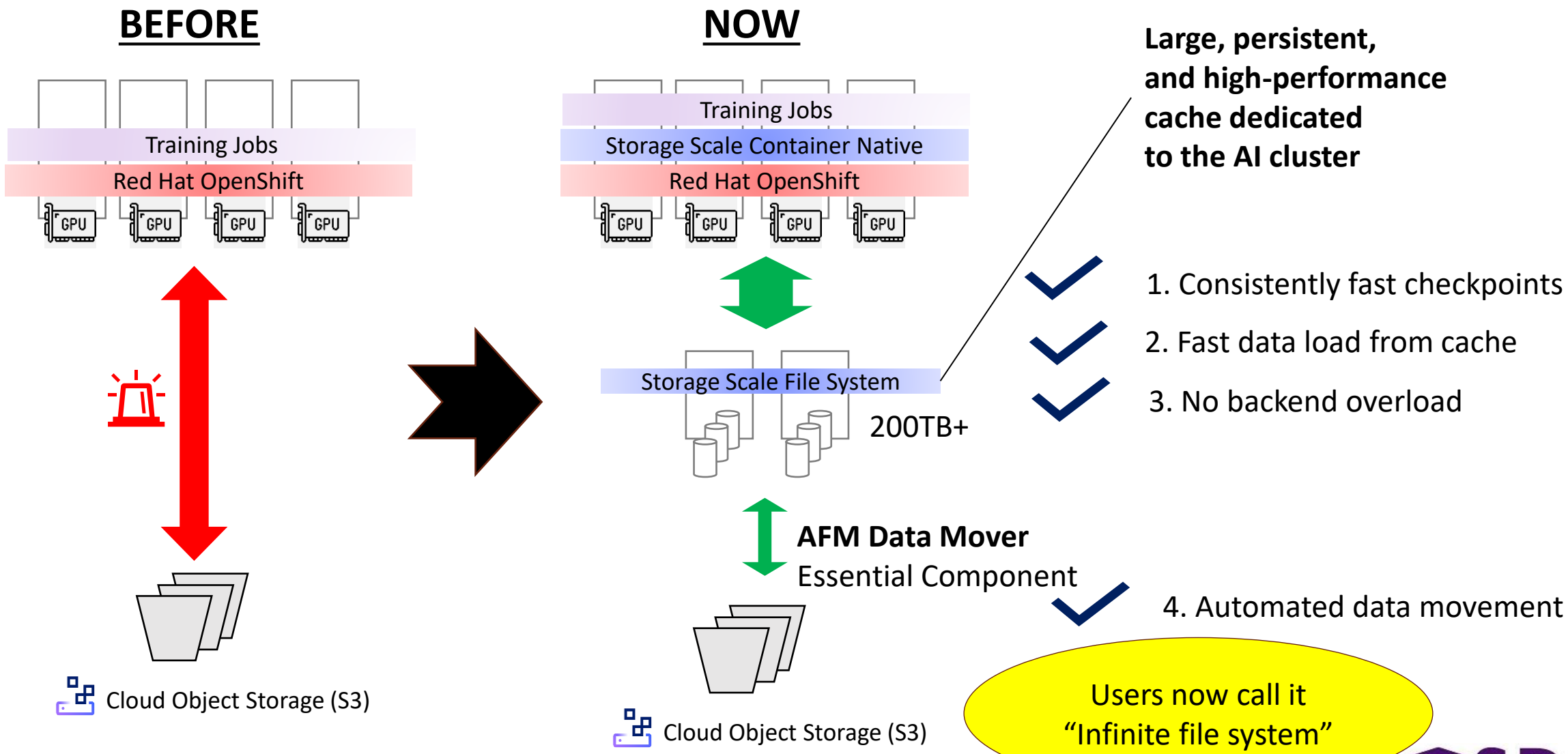




- Extended training times
- Idling expensive GPUs
- Can't perform checkpoints as frequently as desired




(a) gpt3-1.3B (i.e., dense model)

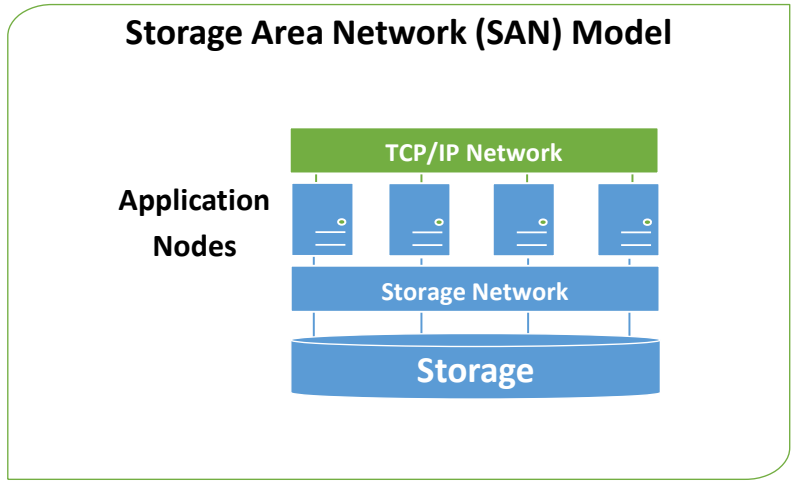
# Multi-user Elastic Cache w/ Storage Scale

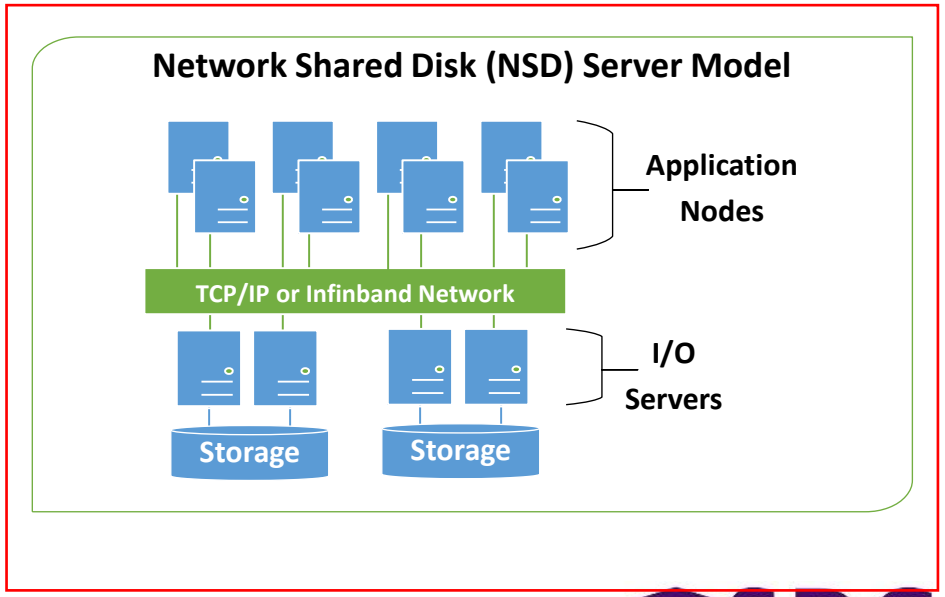


# IBM Spectrum Scale – IBM's shared disk, parallel cluster file system

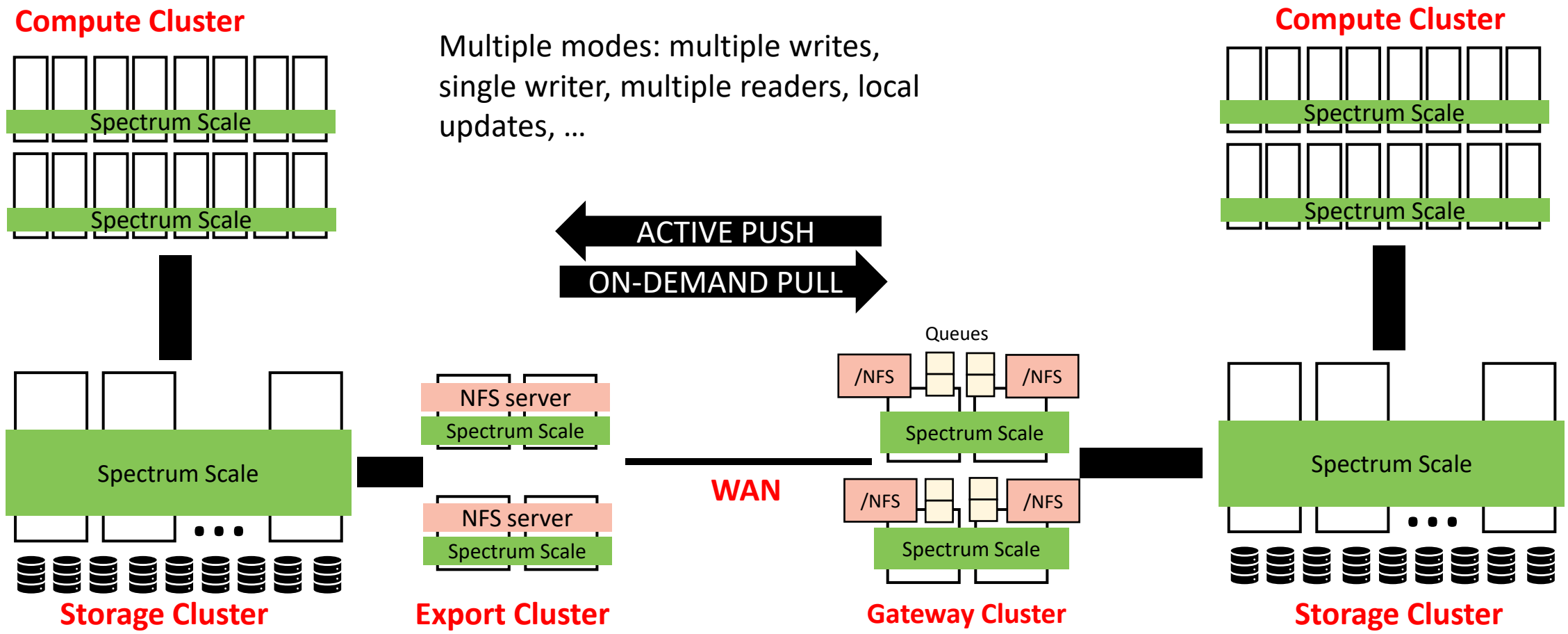
- Standard file system interface with **POSIX semantics**
- Shared disk: all data and metadata on storage devices accessible from any node through block I/O interface ("disk": any kind of block storage device)
- Parallel: data and metadata flow from all of the nodes to all of the disks in parallel.
  - Wide striping
  - Distributed locking for read/write semantics
- Filesets: logical split of a file system namespace



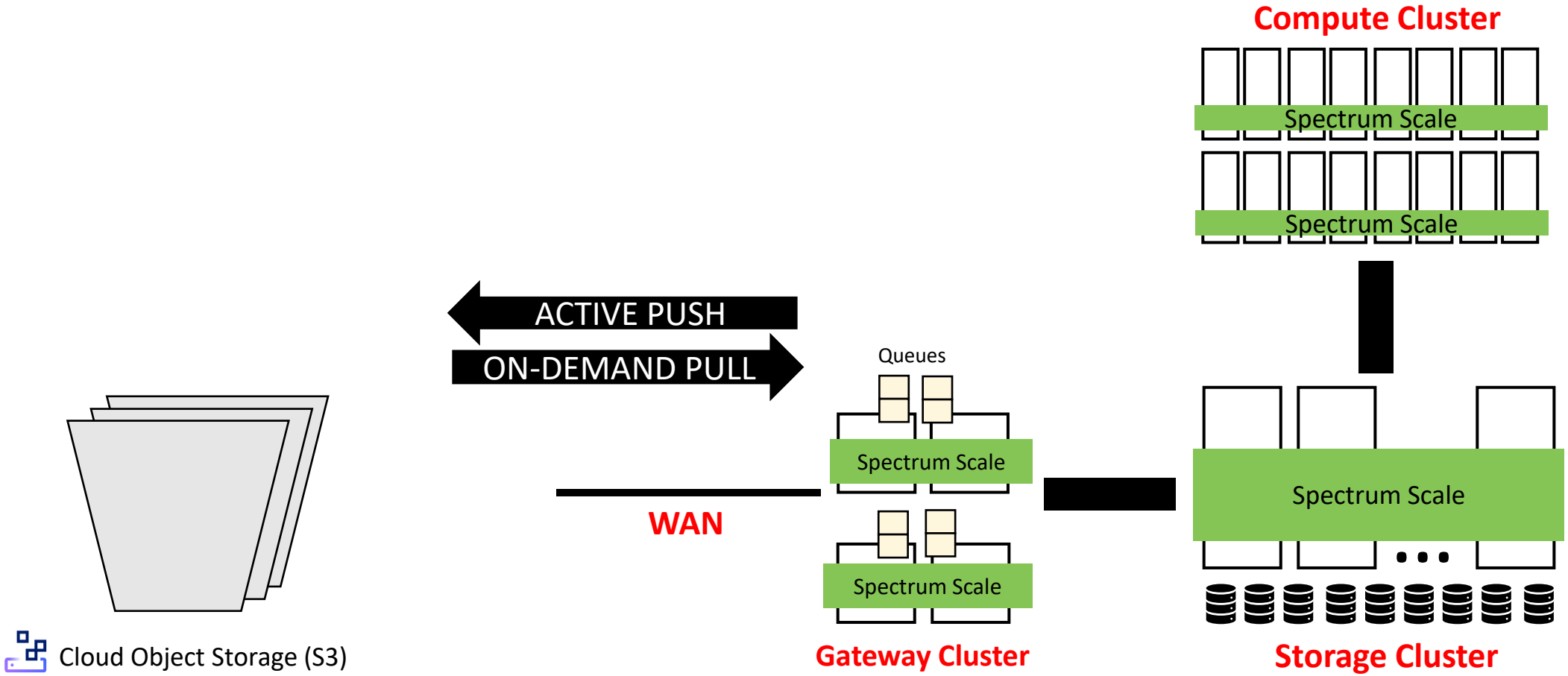
# Active File Management - Traditional




**Home site**

**Cache site**

# Active File Management - Object




**Home site**

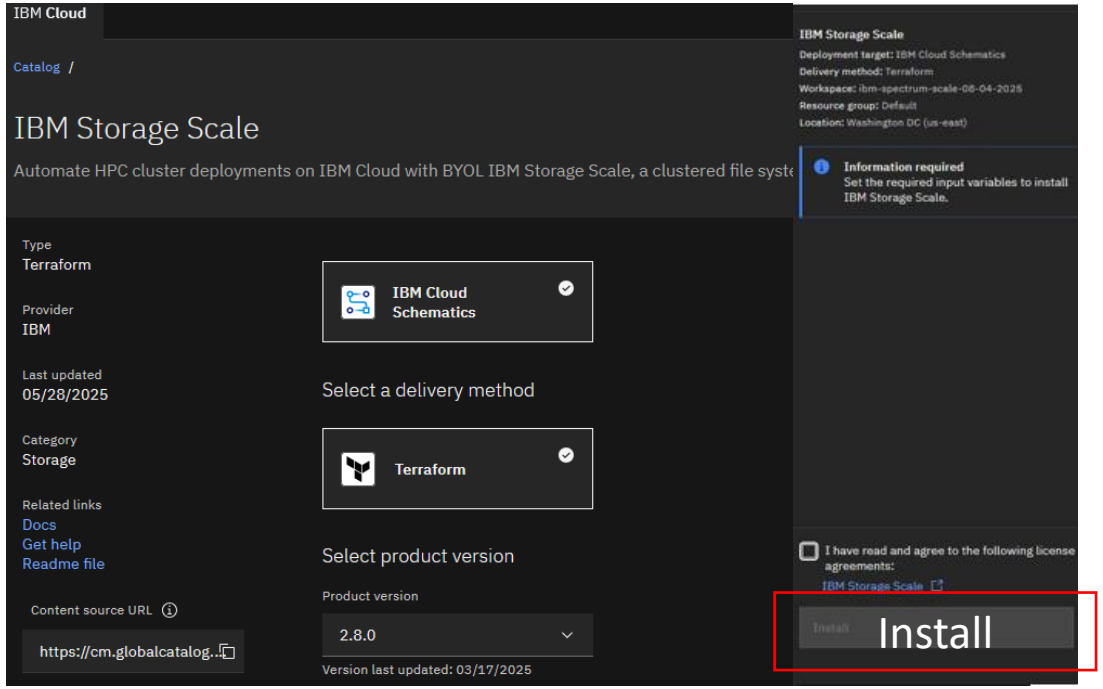
**Cache site**

# Automation for IBM Storage Scale in IBM Cloud

## 1. Deploy with IBM Cloud Tile

- IBM Schematics (Terraform) and Ansible based
- UI, CLI, API




https://cloud.ibm.com/catalog/content/ibm-spectrum-scale

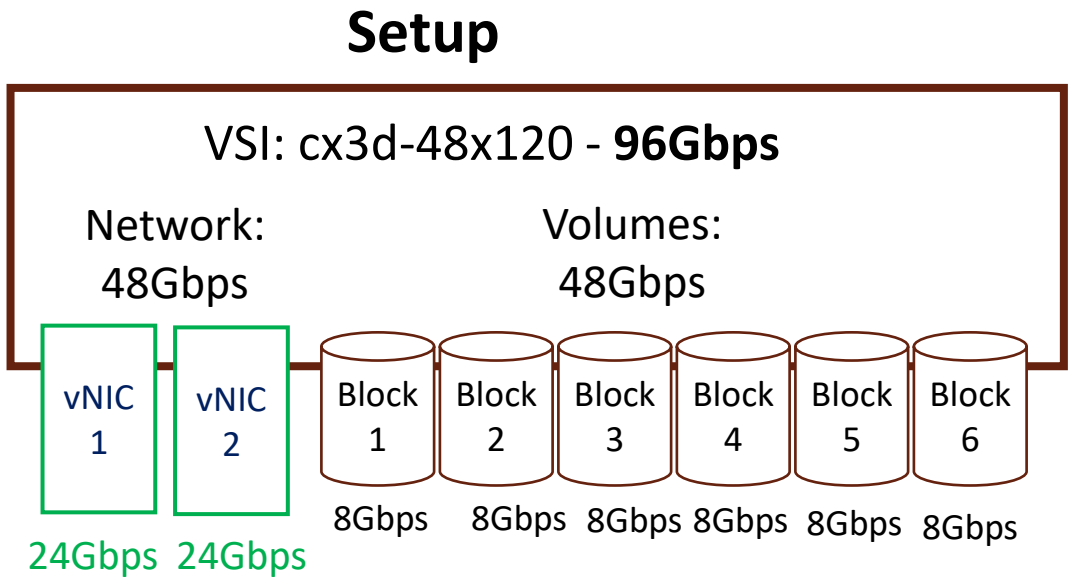
## 2. Use Terraform/Ansible from Github

- https://github.com/IBM/ibm-spectrum-scale-cloud-install
  - AWS, GCP, Azure, (IBM Cloud)
- https://github.com/IBM/ibm-spectrum-scale-install-infra
- More customizable

# Storage Cluster

Every client write results in a networked write to block: split available VSI network 50/50 between block and vNICs



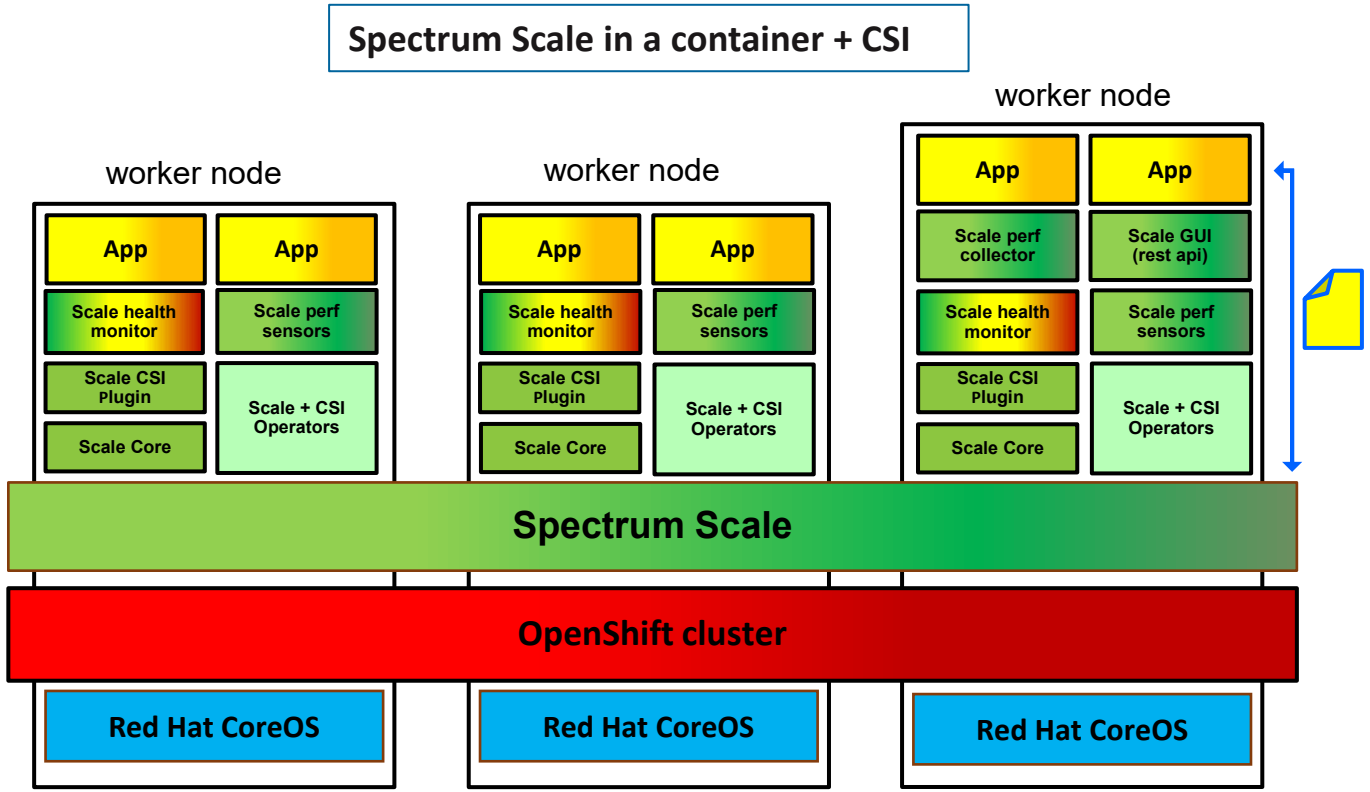

6GB/s (48Gbps) per server
96GB/s per 16-node cluster
240TB if 2.5TB volumes used

- Use of (new) networked block storage type in IBM Cloud
- Alternatives
  - Baremetal
  - ECE / Replication
- Multiple interfaces support
  - MROT
- Some nodes are AFM gateways
- Scale GUI / REST API Server

Scale setup: No ECE, not data replication, 2-way metadata replication

# Compute Cluster - Container Native



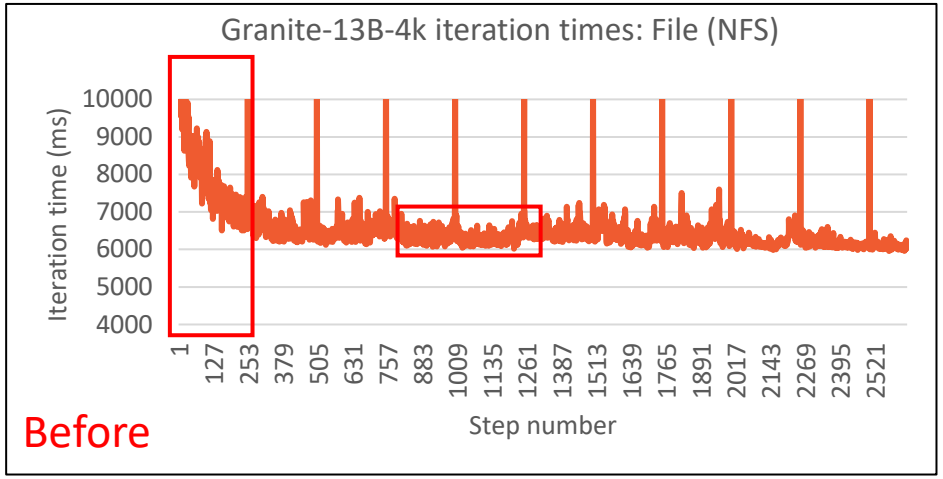

- Dynamic OCP clusters
  - Failures, node movement
  - Node removal process
- Largest Container Native Scale cluster – over 200 nodes
  - Dedicated non-GPU VSIs for quorum nodes
  - Kernel module compilation adjustments
    - Machine Configuration Operator

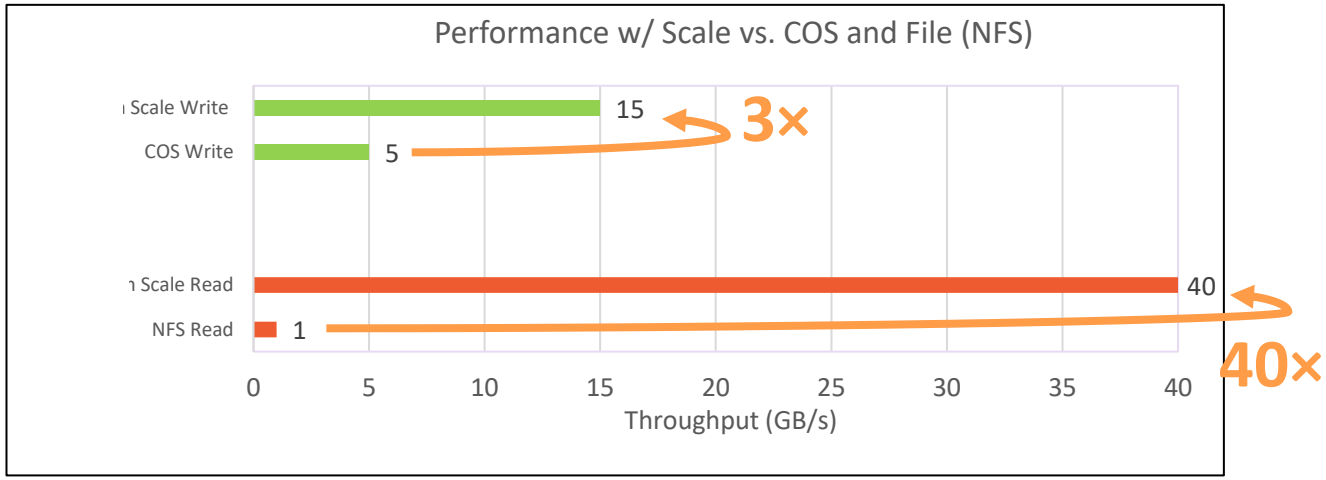
 

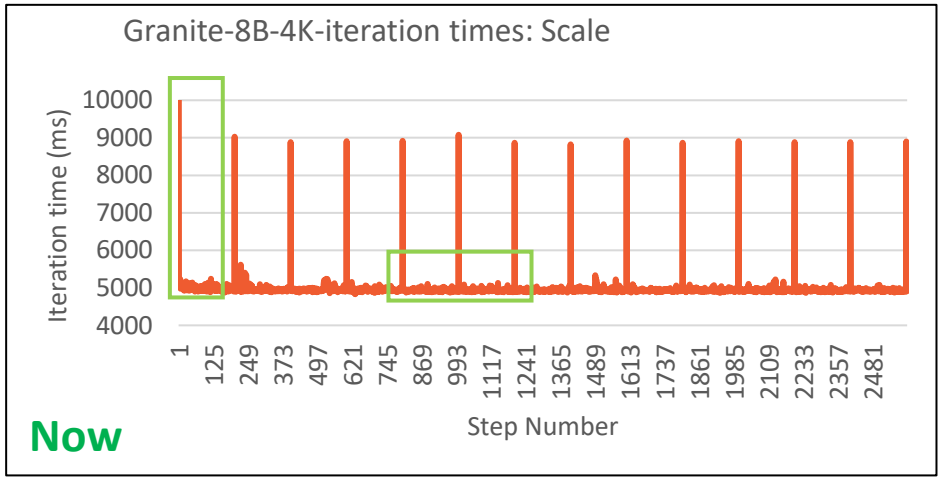
# Production Deployment Results

Summary

- **3×** faster writes (than COS) and **40×** faster reads (than NFS)
- **10+%** improvement in training speed
- Consistent training step times
  - w/ File – 9-6 seconds, 50% variation
  - w/ Scale – 4.8 – 5.2 seconds, **only 10% variation**
  - No ramp down of the training time



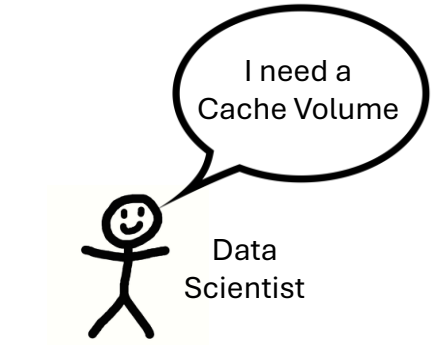




Users demand need for more capacity, efficiency, and operational simplicity

# Scale Cache Volumes

***Earlier process***

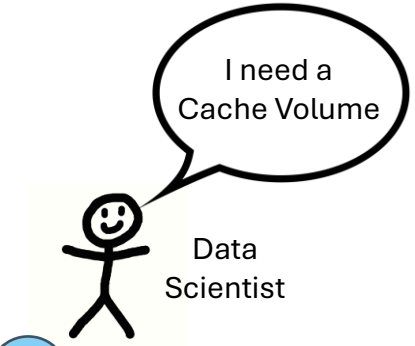



1 Opens Ticket,
Provides S3 keys provides


2 *Manually* sets up Scale Cache Volume
**(Data Scientist waits hours/days)**

- ✗ Manual process & requires AFM expertise
- ✗ Lacks self-service, does not scale well
- ✗ Process is NOT secure!




1 Apply YAML for PVC & Secret for *each* Bucket*

* - Admin creates StorageClass one-time only


2 Automates setting up Cache Volume
**(Cache Volume ready within Minutes)**

- ✓ Reduces setup time from Hours/Days to seconds/minutes
- ✓ Simple interface for non-storage experts!
- ✓ Secure, Self-Service, Scales well

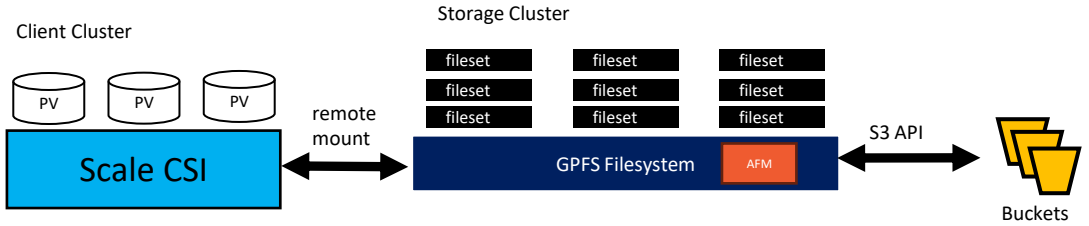
 

# What are Scale Cache Volumes?

1. Maps PV to S3 Bucket
   - Uses AFM fileset as a caching layer
   - *Most* AFM modes are supported

2. Scale CSI driver supports…
   - Dynamic & Static provisioning
   - 1:1 or Many:1 (PV → Bucket)

3. Advanced Features
   - Adjust AFM parameters (fine tuning)
   - Manual cache prefetch / eviction



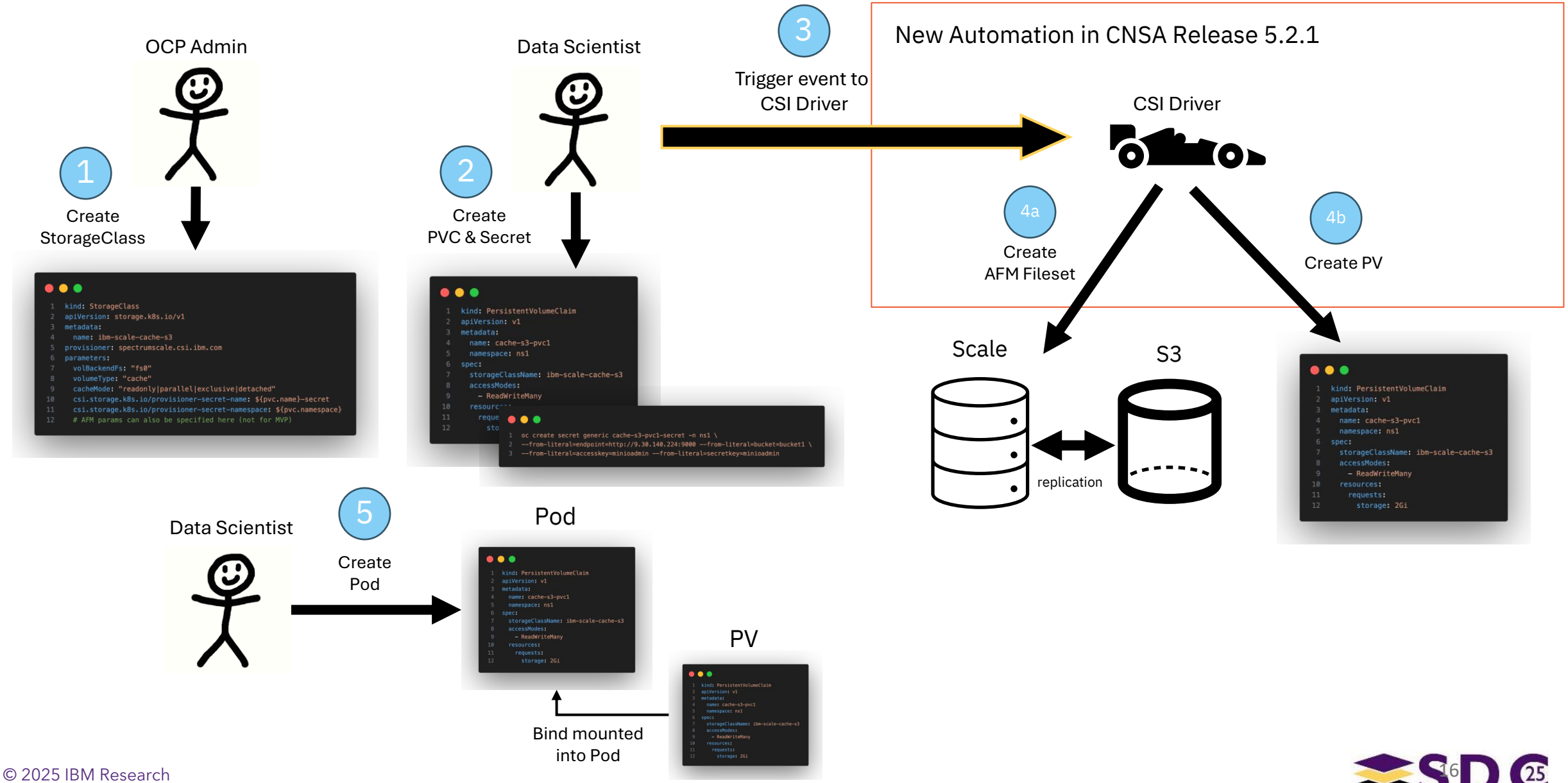

**Scale Cache Volume** is PV → Bucket mapping

**Mapping:** PV → AFM fileset → Bucket

**AFM** does File → Object translation

https://www.ibm.com/docs/en/scalecontainernative/5.2.3?topic=caching-data-from-object-storage

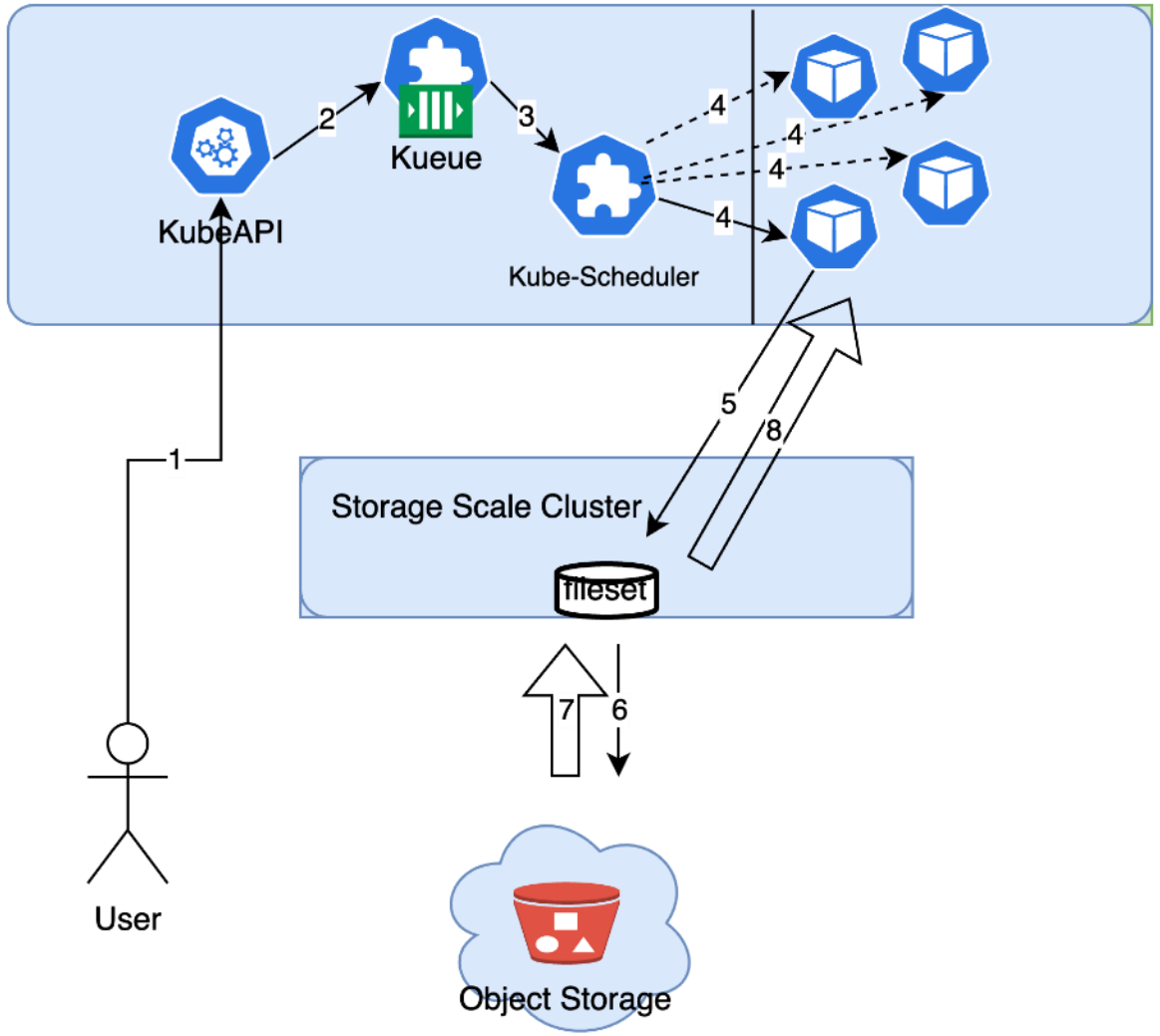
# Scale Cache Volumes - workflow



# Current architecture overview with Kueue

***Problem***: kueue admits job as soon as GPUs (and other resources) are available, but data it needs is not yet available in the storage cluster (still needs to be fetched from object store).

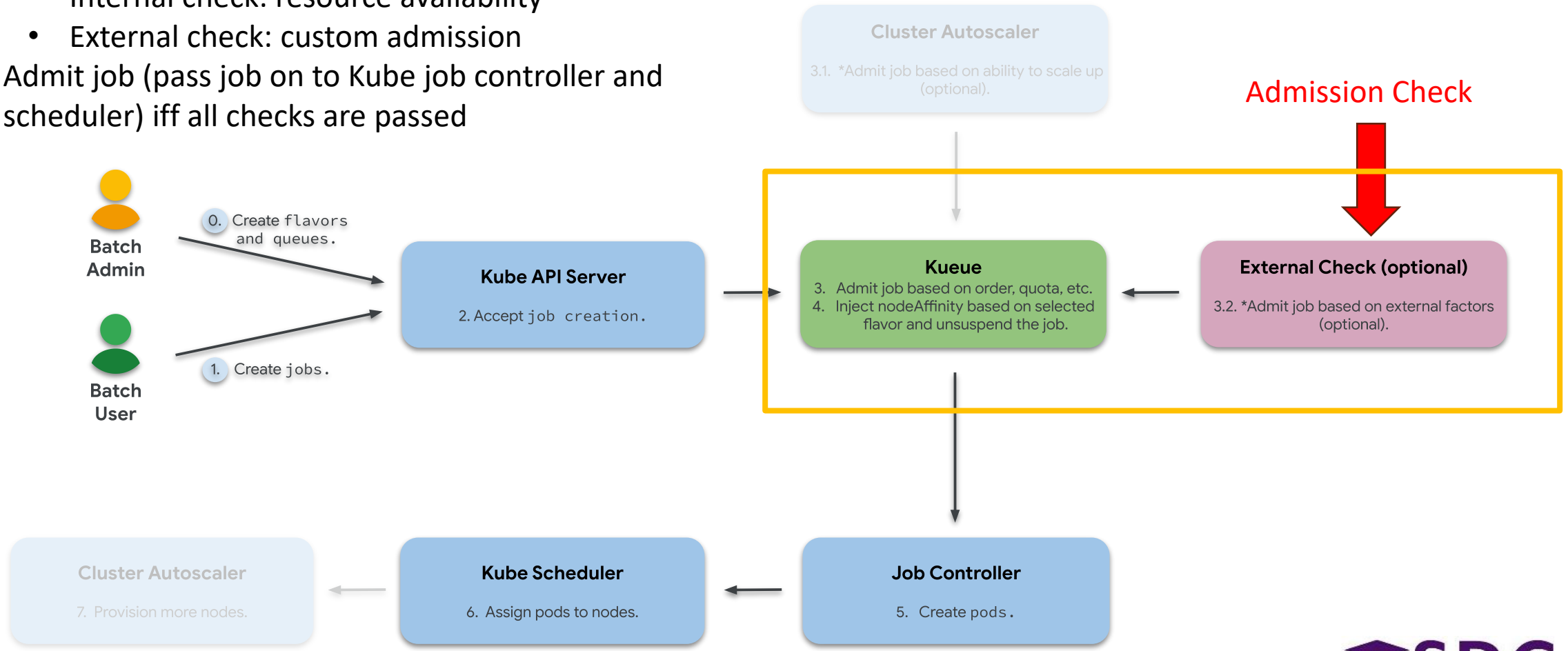
1. User submits job to the OCP/Kubernetes cluster
2. KubeAPI sends job to Kueue for scheduling
3. Kueue
    1. queues workload
    2. reserves resources
    3. admits workload
4. Kube-scheduler assign node to pod
5. Pod runs training jobs and fetches dataset from storage cluster
6. **Storage cluster fetches data from object storage**



# Custom Admission Check in Kueue

- Job scheduling with Kueue
  - Internal check: resource availability
  - External check: custom admission
- Admit job (pass job on to Kube job controller and scheduler) iff all checks are passed



***Solution***: use external admission check to fetch data from object store before admitting the job.

1. User submits job to the Kubernetes cluster
2. KubeAPI sends job to Kueue for scheduling

Job 1

3a. Kueue triggers prefetch of dataset to storage cluster

4a. Storage cluster fetches data from object storage

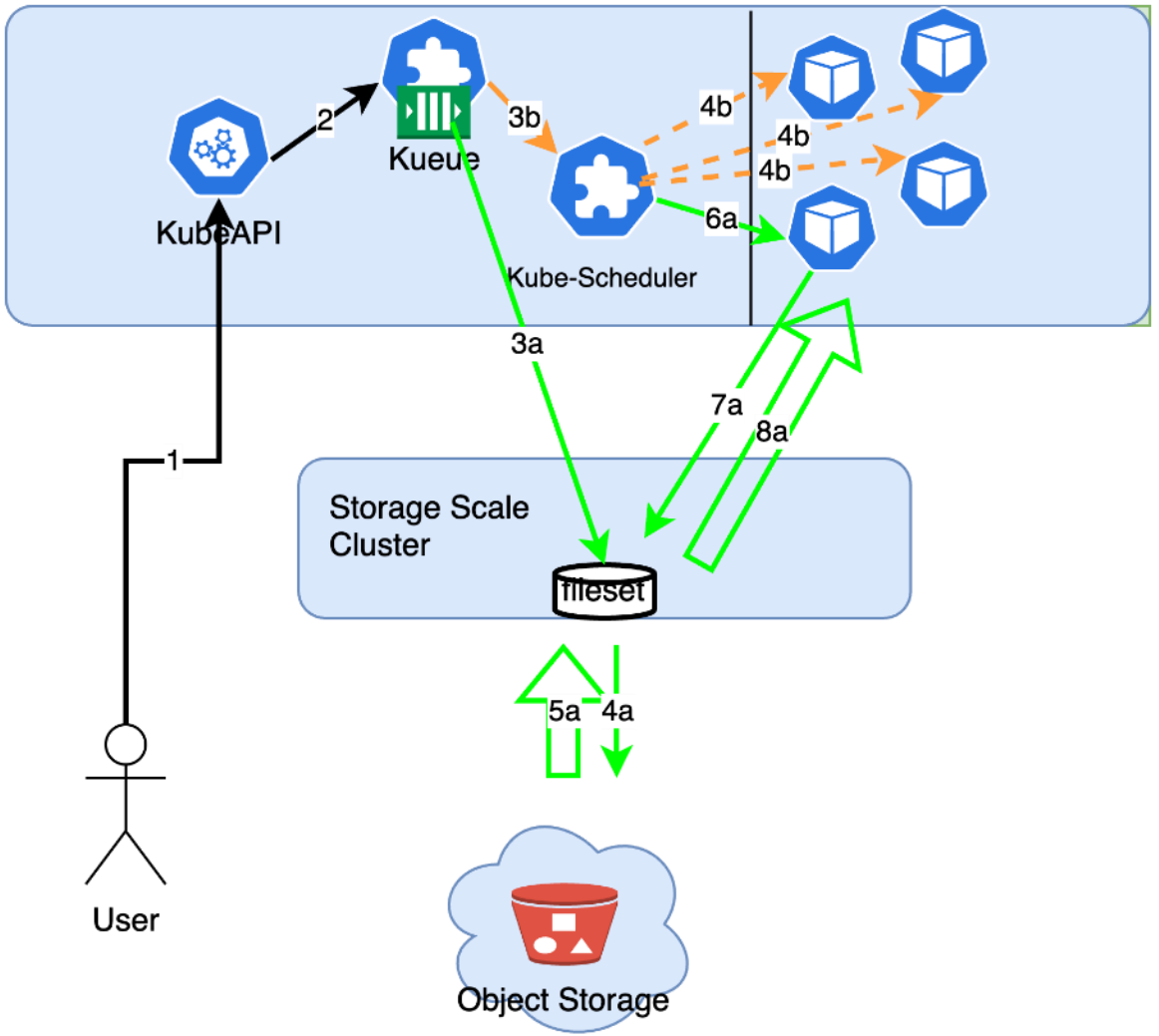
5a. Object storage sends data to storage cluster

6a. Kube-scheduler assign node to pod

7a. Pod runs training jobs and fetches dataset from storage cluster

8a. Storage cluster sends data to training pod

# Dataset prefetch



# Cluster Setup

- ClusterQueue

```yaml
apiVersion: kueue.x-k8s.io/v1beta1
kind: ClusterQueue
metadata:
  name: "cluster-queue"
spec:
  namespaceSelector: {} # match all.
  resourceGroups:
  - coveredResources: ["cpu", "memory"]
    flavors:
    - name: "default-flavor"
      resources:
      - name: "cpu"
        nominalQuota: 1
      - name: "memory"
        nominalQuota: 1Gi
  admissionChecks:
  - custom-ac
```

- Custom Admission Check

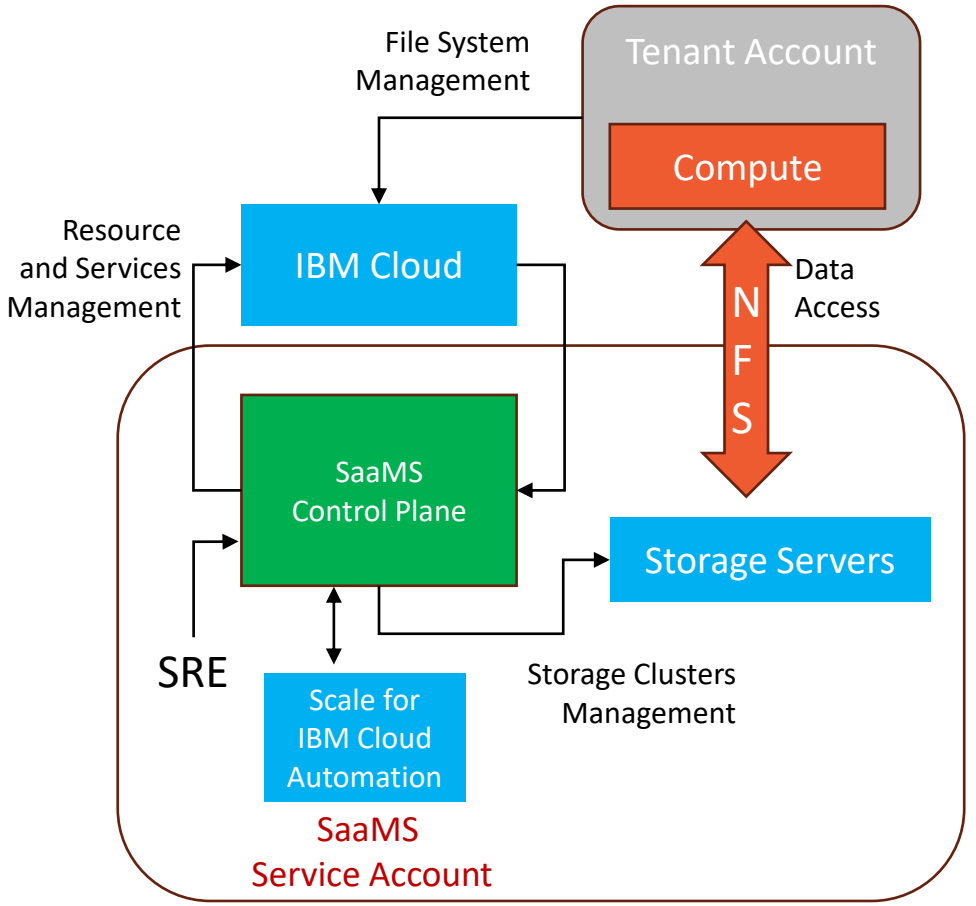
```yaml
apiVersion: kueue.x-k8s.io/v1beta1
kind: AdmissionCheck
metadata:
  name: custom-ac
  namespace: teammlb
spec:
  controllerName: kueue.sandbox.com/prefetch-request
```

- Job Specification

```yaml
spec:
  pytorchReplicaSpecs:
    Master:
      replicas: 1
      restartPolicy: Never
      template:
        metadata:
          namespace: teammlb
          annotations:
            kueue.sandbox.com/pvc-name: "prefetch-pvc"
            kueue.sandbox.com/directory: "/dir"
            kueue.sandbox.com/sub-directory: "subdir"
        spec:
          volumes:
              - name: topology-volume
          containers:
              - name: pytorch
                image: ghcr.io/foundation-model-stack/base:pytorch-latest-nightly-20230126
                imagePullPolicy: IfNotPresent
```

# Scale as a Managed Service

- Managing and operating Scale clusters requires expertise
  - Initial deployment, upgrades, resizing, troubleshooting, etc.
- Cloud users expect more simplicity
  - "I need a highly-available file system volume of size 100TB w/ 100GB/s performance (1GB/s/TB)"
- Storage cluster in service account
  - Managed by storage SREs
- NFS – so that client clusters don't run Scale software
- Clusters not shared between tenants
- Zonal service
- Integrated billing, catalog, monitoring, etc.
- Re- or multi-attachment of block devices



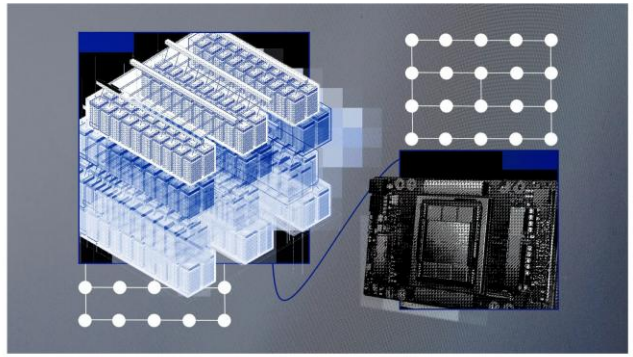

 

# Research Blogs and Papers

Research 7 minute read

## Why we built an AI supercomputer in the cloud

Introducing Vela, IBM's first AI-optimized, cloud-native supercomputer.



https://research.ibm.com/blog/AI-supercomputer-Vela-GPU-cluster

## Vela: A Virtualized LLM Training System With GPU Direct RoCE


Apoorve Mohan* IBM Research Yorktown Heights, USA

Robert Walkup IBM Research Yorktown Heights, USA

Bengi Karacali IBM Research Yorktown Heights, USA

Ming-hung Chen IBM Research Yorktown Heights, USA

Abdullah Kayi IBM Research Bethesda, USA

Liran Schour IBM Research Haifa, Israel

Shweta Salaria IBM Research Yorktown Heights, USA

Sophia Wen IBM Research Yorktown Heights, USA

I-hsin Chung IBM Research Yorktown Heights, USA

Abdul Alim IBM Research Yorktown Heights, USA

Constantinos Evangelinos IBM Research Cambridge, USA

Lixiang Luo IBM Research Yorktown Heights, USA

Marc Dombrowa IBM Research Yorktown Heights, USA

Laurent Schares IBM Research Yorktown Heights, USA

Ali Sydney IBM Research Cambridge, USA

Pavlos Maniotis IBM Research Yorktown Heights, USA

Sandhya Koteshwara IBM Research Yorktown Heights, USA

Brent Tang IBM Cloud Rochester, USA

Joel Belog IBM Cloud Lowell, USA

Rei Odaira IBM Cloud Austin, USA

Vasily Tarasov IBM Research Almaden, USA

Eran Gampel IBM Cloud Haifa, Israel

Drew Thorstensen IBM Cloud Durham, USA

Talia Gershon IBM Research Yorktown Heights, USA

Seetharami Seelam* IBM Research Yorktown Heights, USA



**Abstract**

Vela is a cloud-native system designed for LLM training workloads built using off-the-shelf hardware, Linux KVM-based virtualization, and a virtualized RDMA over Converged Ethernet (RoCE) network. Vela virtual machines (VMs) support peer-to-peer DMA between the GPUs and SRIOV-based network interface.

In this paper, we share Vela's key architectural aspects with details from an NVIDIA A100 GPU-based deployment in one of the IBM Cloud data centers. Throughout the paper, we share insights and experiences from designing, building, and operating the system over a ~2.5 year timeframe to highlight the capabilities of readily available software and hardware technologies and the improvement opportunities for future AI systems, thereby making AI infrastructure more accessible to a broader community. As we evaluated the system for performance at ~1500 GPU scale, we achieved ~80% of the ideal throughput while training a 50 billion parameter decoder model using model parallelism, and ~70%



*Corresponding Authors: apoorve.mohan@ibm.com, sseelam@us.ibm.com



*ASPLOS '25, Rotterdam, Netherlands*



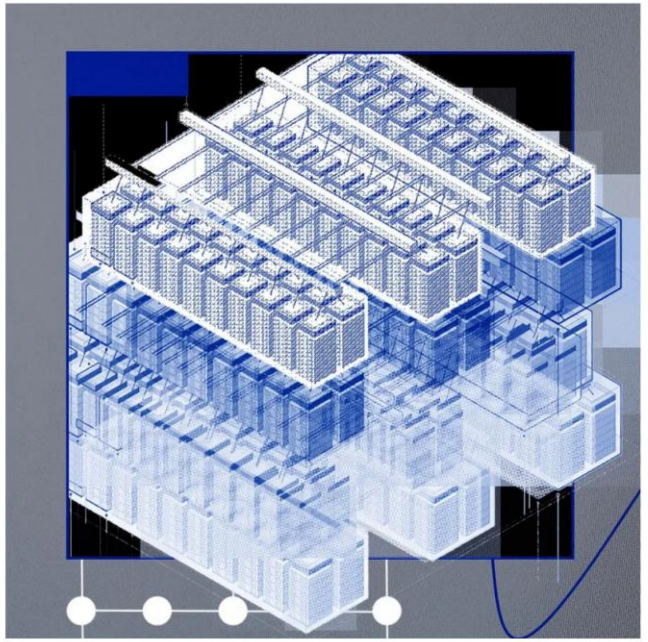
ACM ISBN 979-8-4007-1079-7/2025/03


ASPLOS '25: Proceedings of the 30th ACM International Conference on Architectural Support for Programming Languages and Operating Systems

https://dl.acm.org/doi/pdf/10.1145/3676641.3716280

## IBM uses Storage Scale in its AI model training

By Chris Mellor - August 1, 2024



Big Blue developed its own Vela cluster, using Storage Scale, to train its AI models.

https://blocksandfiles.com/2024/08/01/ibm-uses-storage-scale-in-its-ai-model-training/

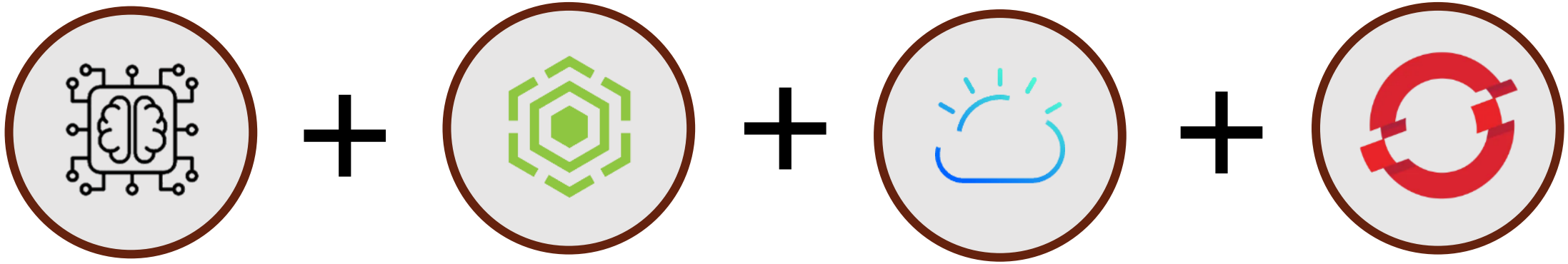
# Team behind this work

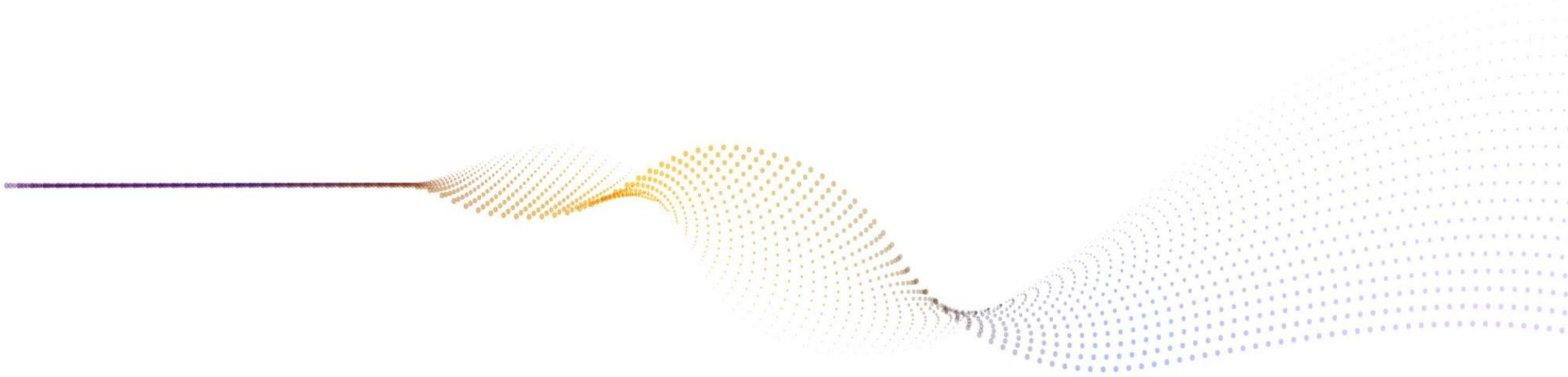

Vasily Tarasov, Scott Guthridge, Jeremy Cohn,
Marc Eshel, Leo Luan, Travis Janssen,
Alex Merenstein, Frank Schmuck,
Lei Pan, Thanh Pham, Veera Deenadhayalan,
Swami Sundararaman, Seelam Seetharami,
Sophia Wen, Talia Gershon
**IBM Research - Hybrid Cloud Infrastructure**

Kevin O'Connor, Abdoulaye Traore,
Chris Laibinis, Brent Wolfe, Carlos Fonseca
**IBM Research – Emerging Technology Engineering**

Piyush Chowdhary
**IBM Cloud – Scale**

Brian Reitz, Steve Pritko, Piyush Shivam
**IBM Cloud – Block Storage**



# Thank you for attending!

Please remember to rate this session. You get access the presentations at
http://sniadeveloper.org/conference