STORAGE DEVELOPER CONFERENCE
SDC 21
BY Developers FOR Developers
Virtual Conference
September 28-29, 2021
A SNIA Event

# Name Based Addressing for Kubernetes Storage

## Container Native Storage and Optimizing DevOps

Jacob Cherian, CPO, Ionir Systems Ltd.
Additional Contributors: Nir Peleg (CTO), Or Sagi (Chief Architect)

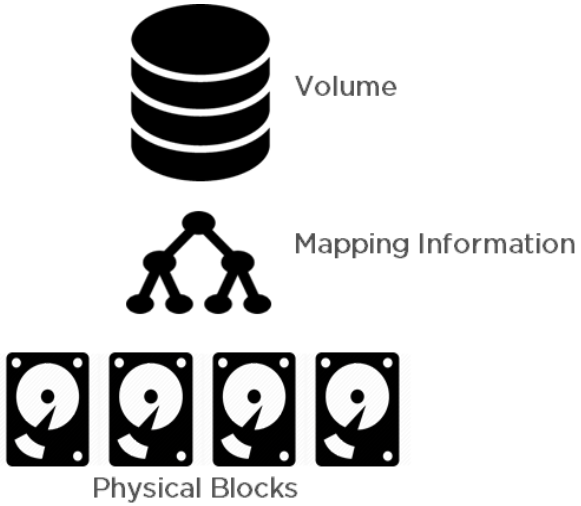
# Agenda

- Addressing Today and Challenges
- Name Based Addressing
- Microservices Architecture for Storage Systems
- Capabilities Enabled by Name Based Addressing
- DevOps Use Case Review

# Storage Metadata and Location Based Addressing

- Data addressing based on location of a data block in the system
  - Metadata maps between logical location on a logical object to physical location of data on a physical device
  - The location eventually maps to a location (block address) on a media device
  - There may be multiple layers of indirection depending on the system
- A logical object (volume, file, object, others?)
  - Defined in terms of the locations where the data for object is stored



# Challenges of Location Based Addressing

- Definitions of logical objects are not transportable
  - Location is only meaningful within the scope of the instance of the storage system
- Metadata design must be tailored for IO profile and data objects
  - Random vs. sequential, read vs. write, large requests vs. small requests
- What if logical and/or device addressing is not block based (object, KV, others?)
  - Translation between addressing schemes
  - Data management operations add additional complexity

# A New Approach - Name Based Addressing

- Metadata built around the concept of named data blocks
- Each data block on ingest is assigned a world-wide unique name
  - Name based on the content of the data block and not location
  - Can be computed by generating a secure hash
  - Chunking, Optimal chunk size(s)?
- Any logical object (volume, file, object, others?)
  - Defined in terms of the names corresponding to contents of each data block in the logical object
- Addressing based on <u>what the data block is, instead of where it is stored</u>


Definition of a Logical Object

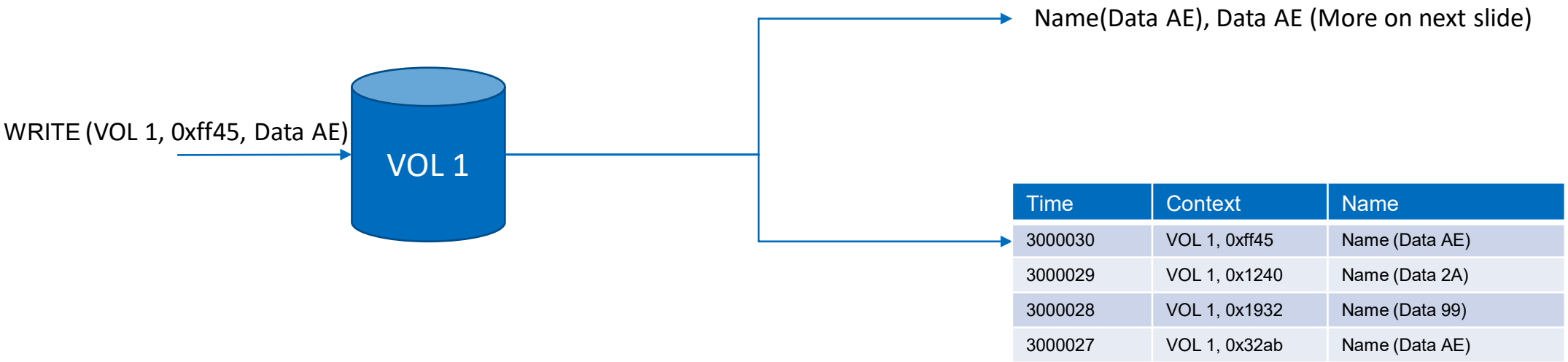
 

# Advantages of Name Based Addressing

- The definition of a logical object is independent of the system on which it lives
  - Definition of the logical object is inherently portable
- Data addressing independent of logical object or media addressing
  - Extensible beyond block-based addressing
- Within a system data can be moved without updating metadata
  - Most of the time and in cases where it needs to be, the complexity of updates is minimal

# Metadata based on Name Based Addressing



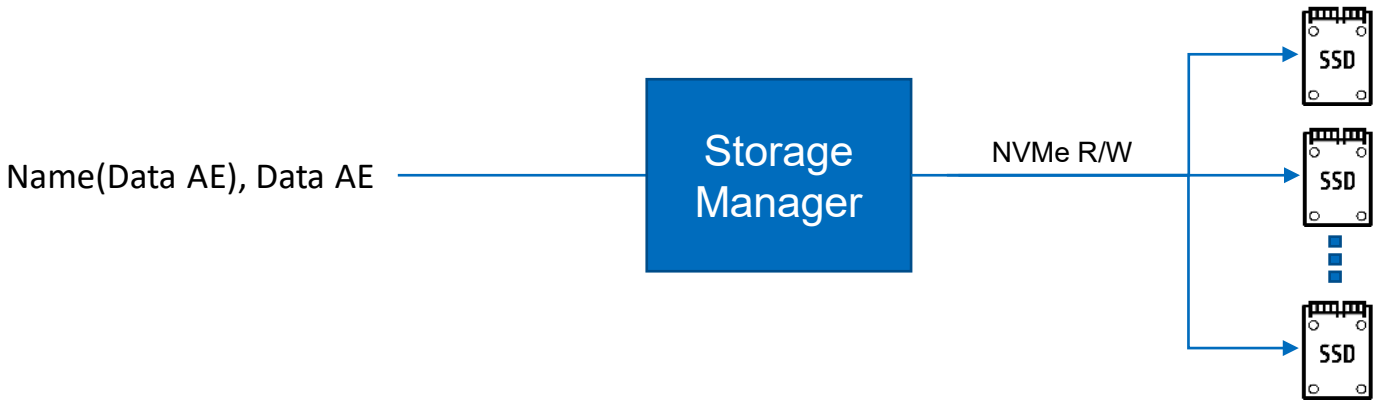

| Time | Context | Name |
|---|---|---|
| 3000030 | VOL 1, 0xff45 | Name (Data AE) |
| 3000029 | VOL 1, 0x1240 | Name (Data 2A) |
| 3000028 | VOL 1, 0x1932 | Name (Data 99) |
| 3000027 | VOL 1, 0x32ab | Name (Data AE) |

Core Metadata (volume example)

- Metadata defines the relationship between the logical context associated with a data block and its name
  - The logical context is opaque and is only used during data retrieval
  - Allows system to support any arbitrary logical abstraction for data (volume, object, KV, others?)
  - Time record allows reconstruction of state of a logical object from arbitrary point in the past

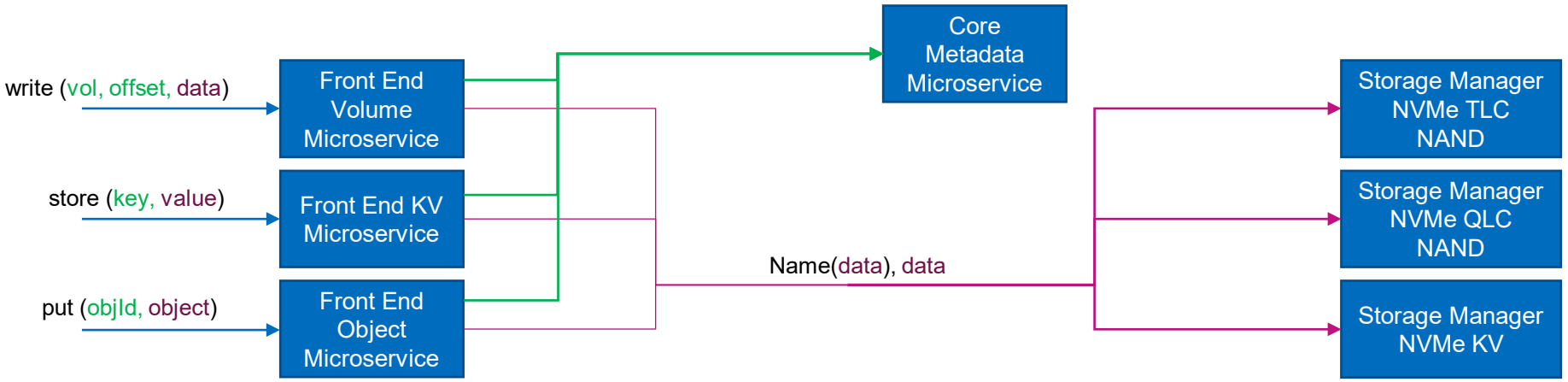
# Storing Data Blocks and Managing Media




- A storage manager component translates between data names and to location on media
  - Storage manager responsible behavior required for a specific device type
  - Manages resiliency across multiple devices
  - A given data block is only stored once – providing inline deduplication
- Allows system to be extended to support multiple device types

# Name Based Metadata and Microservices




- Platform can support multiple front-ends and media types
  - Performance dependent on the combination of front end and storage manager service
  - Core metadata service just records updates and performance is not dependent on IO profile

# Capabilities Enabled by Name Based Metadata

- Instant Mobility of Persistent Volumes
  - Move volumes between clusters instantly by moving names of data objects
  - All writes are local at the new location before any data is copied
  - Data for read can be moved on demand/as a background process/heat based
  - Names also reduces the amount of data that needs to be moved, by identifying duplicate data

# Capabilities Enabled by Name Based Metadata

- Clones to Any Arbitrary Point in the Past
  - Enabled by time information recorded in the core metadata
  - Volume state from the past for a given logical context recreated by querying metadata store with desired time stamp

# Eliminating DevOps Data Delays with Name Based Container Native Storage

Instant Data Mobility and Agility

ionir

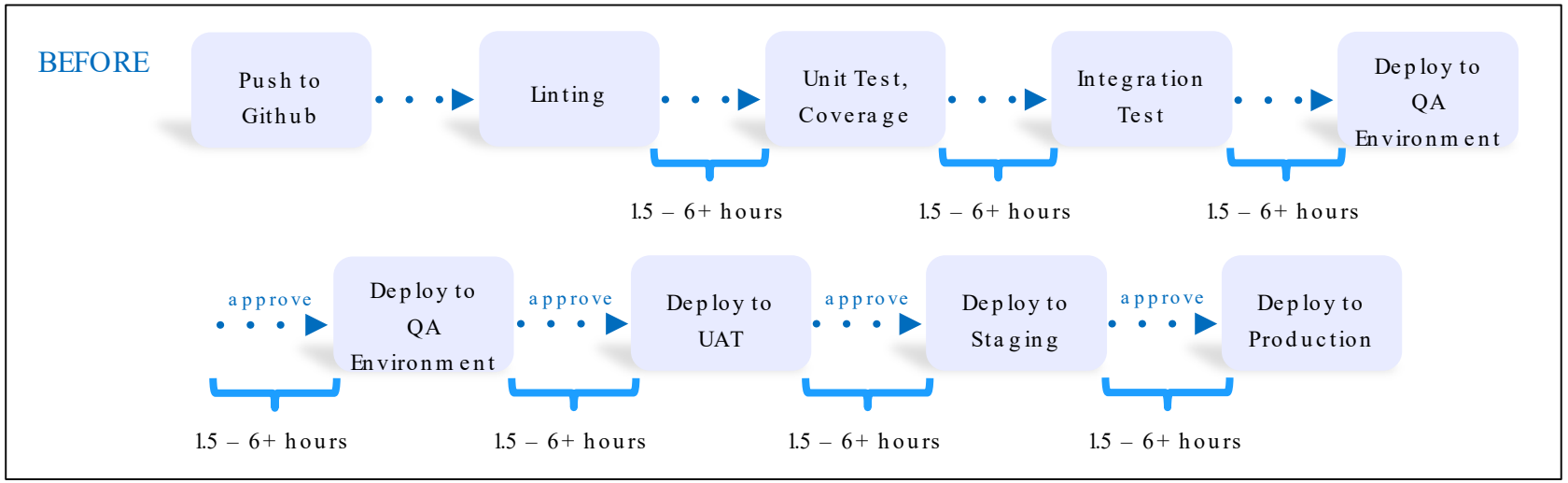
STORAGE DEVELOPER CONFERENCE SDC 21

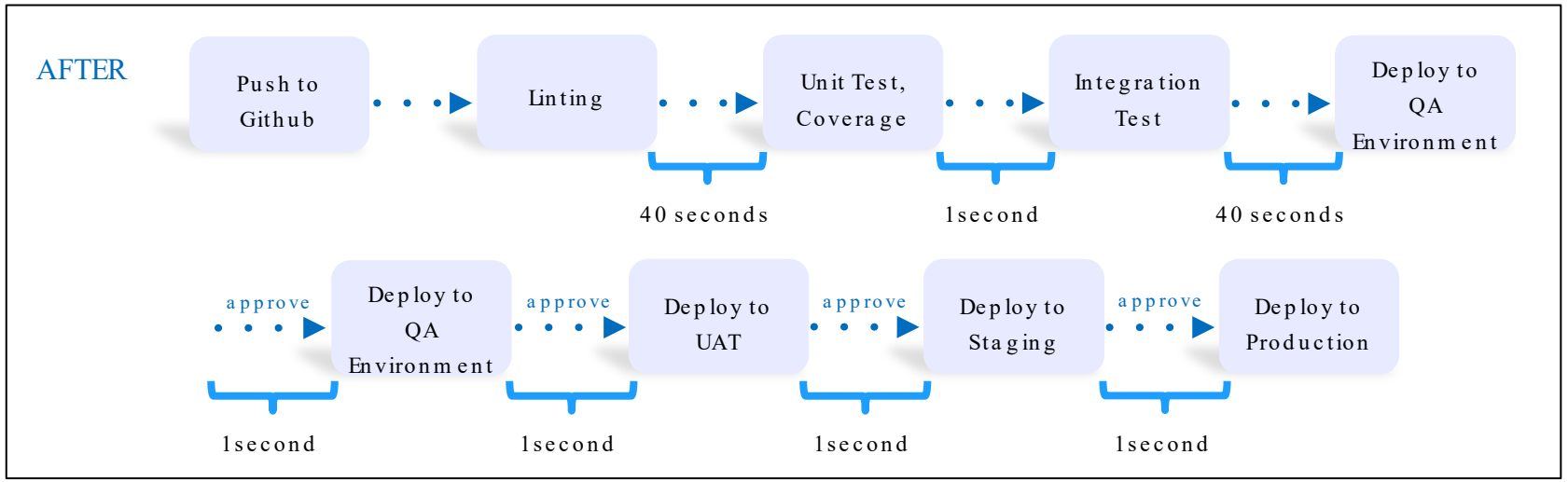
# DevOps Data Challenges

- DevOps infrastructure that spans multiple K8s clusters across clouds
- CI/CD and other processes require:
  - Movement of data sets between K8 clusters and clouds
  - Provisioning of multiple copies of data sets
  - Reverting data to reset operations or clear errors

# DevOps Acceleration Example




Delays between stages due to time to set up data.

Optimized Pipeline

- 40s for data operations between K8s clusters using Instant Mobility
- 1s for data operations with a K8s cluster

Pipeline time reduced from many hours to a few minutes

ionir

STORAGE DEVELOPER CONFERENCE SDC 21

# Thank You

info@ionir.com