PowerScale and ECS: Data Orchestration with CloudSoda

Data Movement as a Service for Media Workflows

March 2023

H19520

White Paper

Abstract

This document describes a distributed storage environment with PowerScale file storage and ECS object storage. CloudSoda orchestrates the movement of data between storage types. The paper introduces data movement as a service (aaS) with CloudSoda and concepts such as agents, accessors, and storage targets.

DELL Technologies

Copyright

Contents

Executive summary

Overview

Media workflows leverage multiple sites and storage types. The storage types run the gambit: high-performance on-premises PowerScale file storage, private object Elastic Cloud Storage (ECS), public cloud, and even direct attached workstation storage. CloudSoda provides a modern, cloud-native, and on-premises aware data orchestration tool called SoDA. SoDA gives media companies the power to orchestrate sophisticated data moves through a single pane of glass.

SoDA is data movement as a service. A cloud console connects to local agents to perform WAN/LAN optimized data moves. Transfers are configured with a policy-based engine for copies, moves, and syncs of file and object data.

This paper introduces SoDA workflows and how these workflows integrate with PowerScale file storage and ECS object storage. SoDA concepts such as storage targets, accessors, and agents are described in detail.

Revisions

Date	Part number/ revision	Description
March 2023	H19520	Initial release

We value your feedback

Dell Technologies and the authors of this document welcome your feedback on this document. Contact the Dell Technologies team by email.

Author: Gregory Shiff

Special Thanks: Greg Holick

Note: For links to other documentation for this topic, see the PowerScale Info Hub and the ECS Info Hub.

Distributed media ecosystems

Managing a mix of storage types

File Storage File Storage

Media production is global business. Productions are spread between multiple facilities in various geographies. Often, locations each have their own silos of storage, such as high-performance PowerScale file storage. Meanwhile, the organization might use a massive object store (private ECS or public cloud) as the central storage core. In addition to these shared-storage repositories, data may be at remote locations on internal direct attached storage. These parts come together to form a data ecosystem.

Orchestrating data movement to make sure that the right files are where they need to be at the right time is a serious undertaking. Successfully moving production assets has implications that directly impact budgets and production timelines.

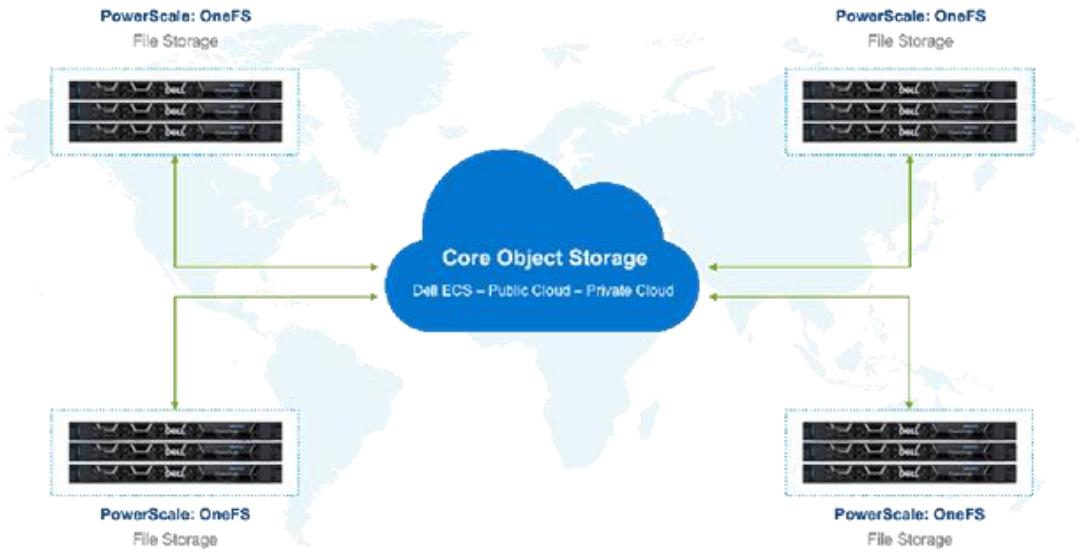

Figure 1. Storage ecosystem

Data movement as a service

SoDA is designed to tackle these challenges. The software provides a central console with visibility into all types of storage: object, file, and local. Data movement policies are configured and monitored through this console. Data can be moved, copied, or synchronized (one way) in an on-going or one time basis. The cloud console communicates to local agents at each site, where each site uses WAN optimized data movement directly among all sites

The SoDA Controller is deployed as a service in AWS, on Prem, or Hosted by SoDA. Users log into a fully managed SoDA instance. From here, agents for Windows, Linux, and macOS can be downloaded for deployment near storage resources. The local agents may communicate to one or more storage targets simultaneously. Multiple agents can connect to the same storage target in different ways, such as a Windows agent using SMB and a Linux agent using NFS. CloudSoda has the smarts to differentiate between these various scenarios while providing agent based access to shared storage.

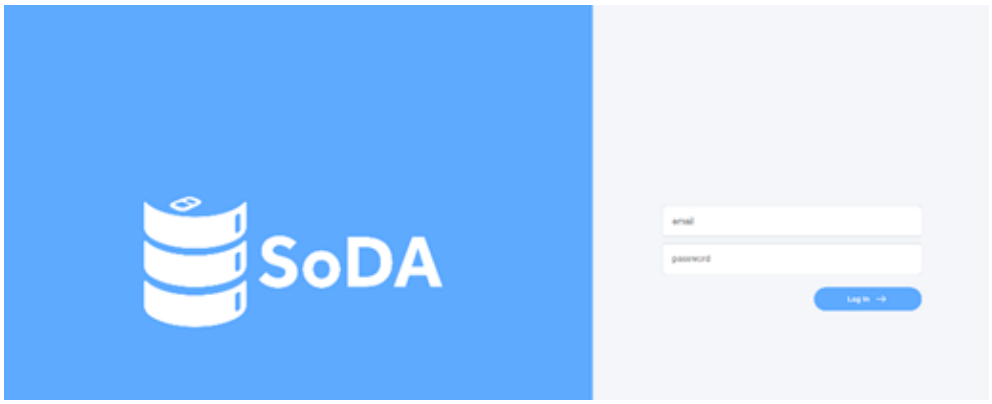

Figure 2. CloudSoda login

CloudSoda concepts and architecture

CloudSoda Architecture

There are two main components of the SoDA architecture: the cloud management console (or SoDA Controller) and local agents.

The SoDA Controller and associated services are fully managed and offered as a service. Administrators access the SoDA console to configure storage targets, local agents, and accessors (how the agents connect to the storage). The SoDA Controller provides an interface for configuring and monitoring data movement policies and one-time transfers.

Local agents perform the data movement. Agents are installed on machines that have access to the various storage targets. These storage targets could be direct-attached block, shared NAS, or object storage. The CloudSoda agents run as background processes on the local machines while all management is configured through the single cloud management console.

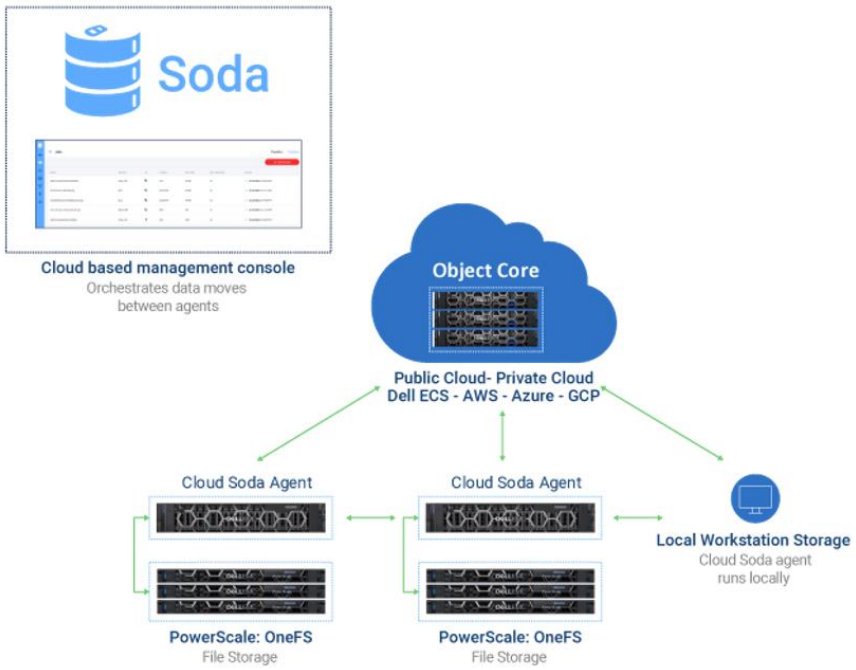

Figure 3. CloudSoda architecture

Storage, agents, accessors

CloudSoda breaks out storage targets, local agents, and accessors. This modular approach provides enough flexibility for a wide range of scenarios, especially when the same storage is accessed through various protocols.

Storage targets are the storage silos themselves. Broadly, there are two types of storage that CloudSoda supports: file and object. File storage can be shared storage or local direct attached drives. Object storage can be private, such as Dell ECS, or public cloud.

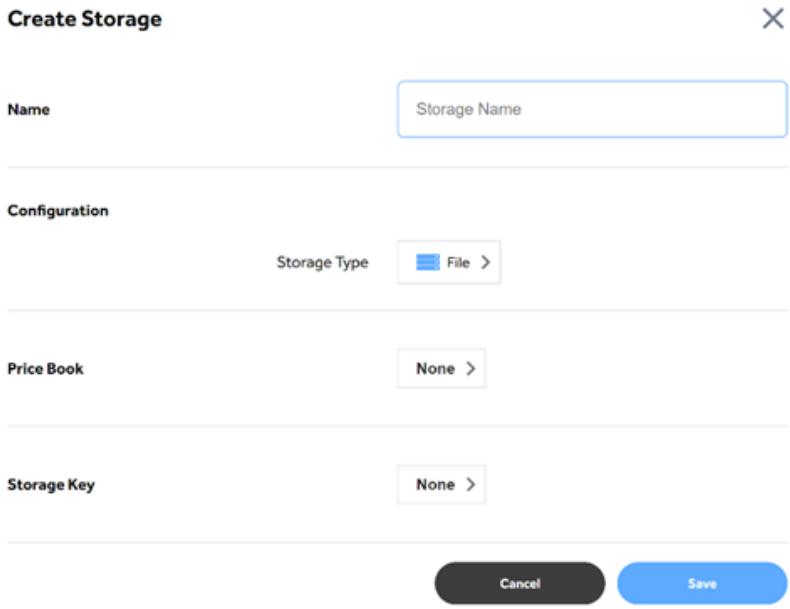

Figure 4. Storage targets

Agents are the next layer in SoDA. Agents can be thought of as the computers that have access to storage targets and will be doing data movement. CloudSoda supports Windows, Linux, and macOS agents. When an agent has been configured in the CloudSoda console, a configuration file for that agent is automatically generated and available for download from the SoDA Controller.

Finally, CloudSoda uses the concept of storage accessor. An accessor defines how a particular agent connects to a storage target. For example, a Linux client connecting to an NFS share will have a particular path to that storage. A Windows client will use SMB and have a UNC path with a drive letter to the same storage. This allows multiple agents to access the same storage device which is represented as a single storage in SoDA. Both of those clients can also be connected to the same S3 storage using the same bucket credentials or separate credentials for security needs. In this example, there would be three accessors, an NFS accessor, an SMB accessor, and an S3 accessor. The Linux agent would use the NFS and S3 accessors while the Windows agent would get the SMB and S3 agent. This level of configuration greatly reduces the need for complex path mapping or translation.

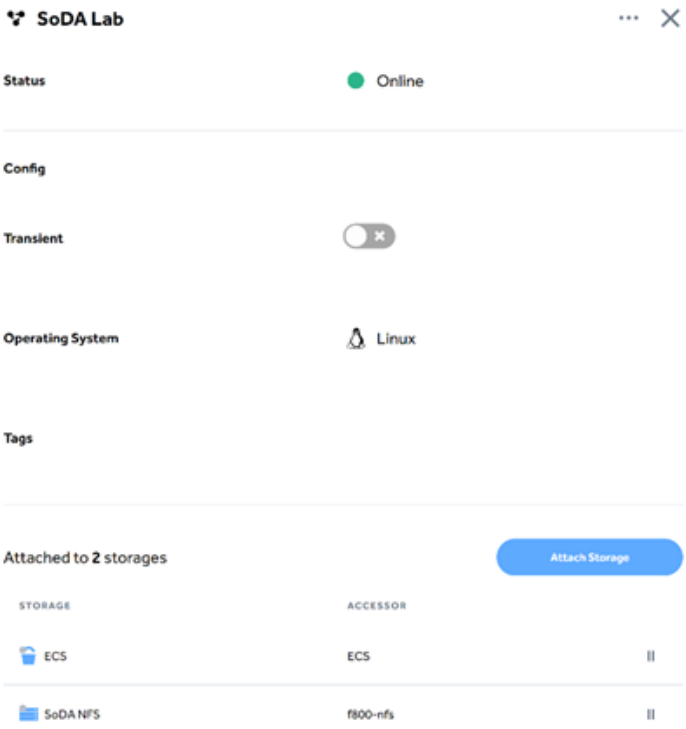

Figure 5. Agent configuration

The agent configuration in Figure 5 is such an example. The Linux agent is connected to two storage targets: an NFS share on PowerScale and an S3 Bucket on ECS.

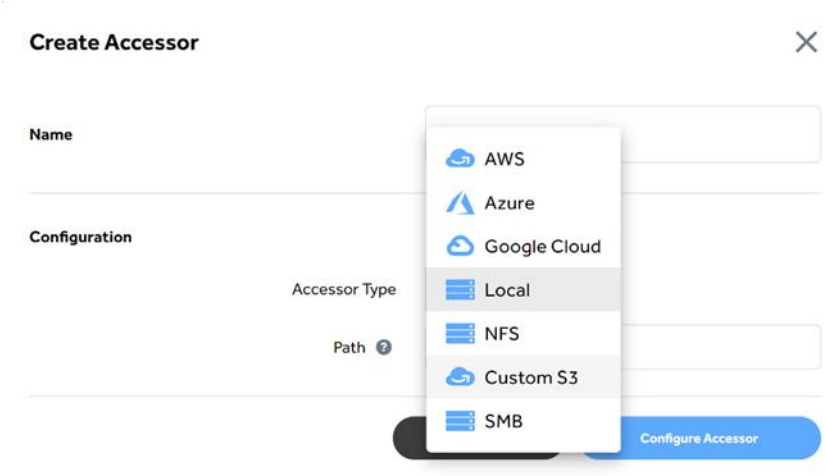

Figure 6. Accessor configuration

CloudSoda data movement and policies

How CloudSoda moves data

SoDA agents move data between themselves using encrypted, WAN optimized UDP transfers. Data is check-summed at either end to ensure integrity. Agents use the S3 API to communicate to object storage and support multipart uploads of large files. Storage performance, compute resources available on the agent host, and speed of the network

link between agents determine transfer speeds. Whether file or object, SoDA looks at the current data set and dynamically spins up and down threads based on file size to maximize performance across a WAN or LAN link.

Data moved by SoDA is accessible at either end. File data moved to an object store can be accessed natively with S3 APIs. Objects moved back to file storage can be accessed natively there as well. SoDA does not require or use any proprietary gateways.

Policies and quick transfers

A policy-based engine within the CloudSoda console controls when data moves within the CloudSoda ecosystem. User configured policies can run on a regular schedule and can be used to create drop folders, or a user can do a quick one-time transfer. The policy engine provides options for error and conflict resolution. Helpfully, there is also the option to dry run a policy before setting it in motion on production datasets, which is part of the SoDA patented technology.

SoDA policies provide control over exactly what data is moved and only moves what is required. Data can be filtered in many ways, including identifying hot and cold data or certain file types and names.

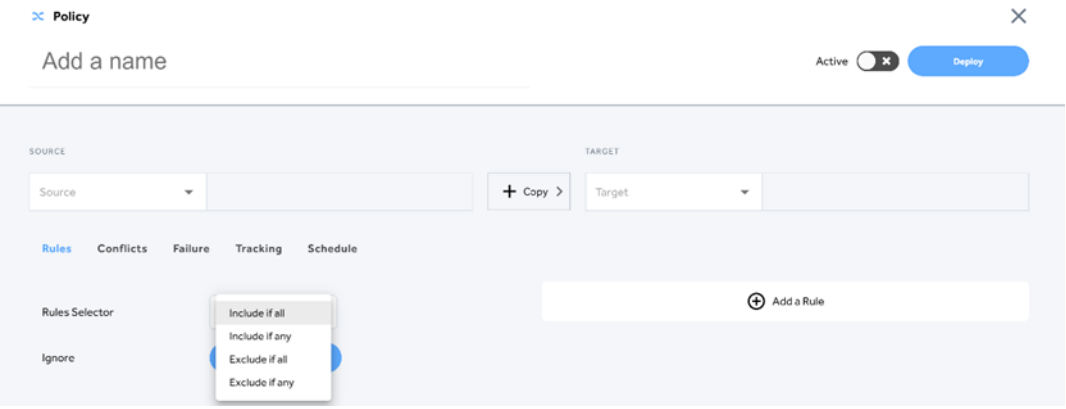

Figure 7. Storage policy

SoDA API and Webhook support

SoDA provides a REST API for external systems to monitor and manage data orchestration. While not meant as a complete replacement for the management console, the API allows systems such as Media Asset Management platforms to leverage SoDA for data movement at scale.

The SoDA API allows for programmatic creation of transfer jobs along with all associated parameters. This can include whether the job is a move, sync, or copy, along with various filtering parameters such as file type or name. After the transfer is initiated, the API can be used to monitor status of this new job or any existing SoDA jobs.

SoDA API documentation can be found here: SoDA API.

Figure 8. SoDA API

SoDA also supports webhooks to proactively push notifications to systems about job completion. Conceivably, this could be used to initiate further data processing when a data move has finished.

Webhooks are configured in the SoDA console. While job completion might be the most obvious use case, there is a range of job conditions that can be set to trigger a particular webhook.

For additional details, refer to the SoDA Webhook documentation (account required) for Creating a Webhook.

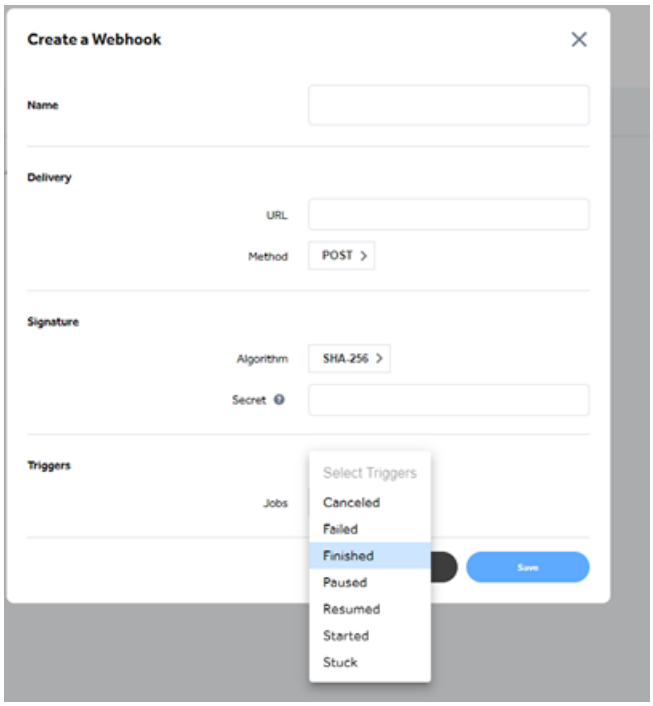

Figure 9. Webhook configuration

Media workflow use cases

Remote production

For remote productions, getting the camera data to the post-production facility as quickly as possible can be a challenge. With CloudSoda, remote productions can move data from a local RAID or hard drive to object storage or a PowerScale NAS back at the post facility. CloudSoda agents running on dedicated data transfer machines at the remote location can be configured to move data to one or more targets.

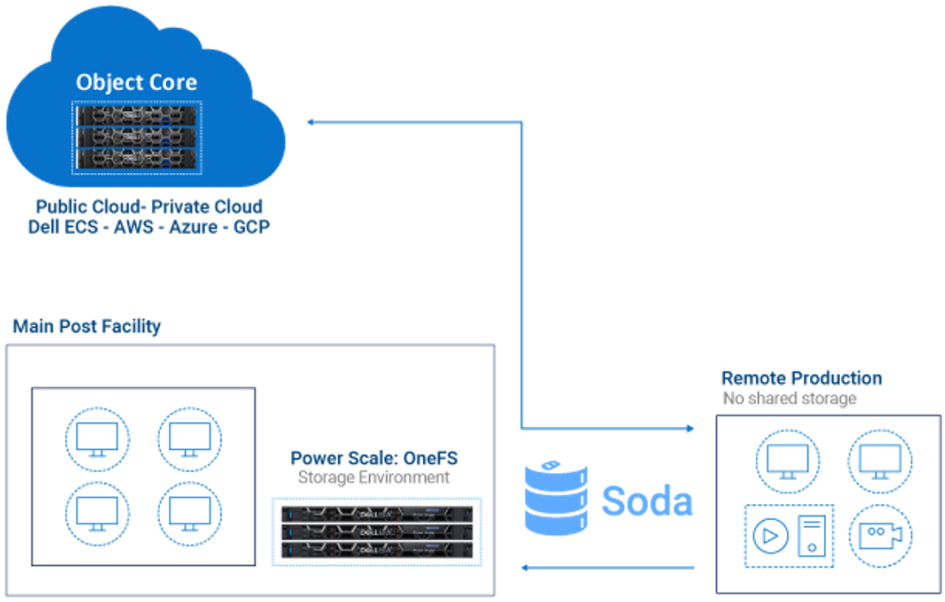

Figure 10. Remote production

Multisite production with object core

Content creation is often spread between multiple facilities that all need access to the underlying media. CloudSoda's policy-based engine can synchronize files to an object repository while simultaneously copying data directly between file storage at each remote location. In this manner, underlying media assets that may not change can be protected on object storage while smaller, transitory project files can be copied directly between locations.

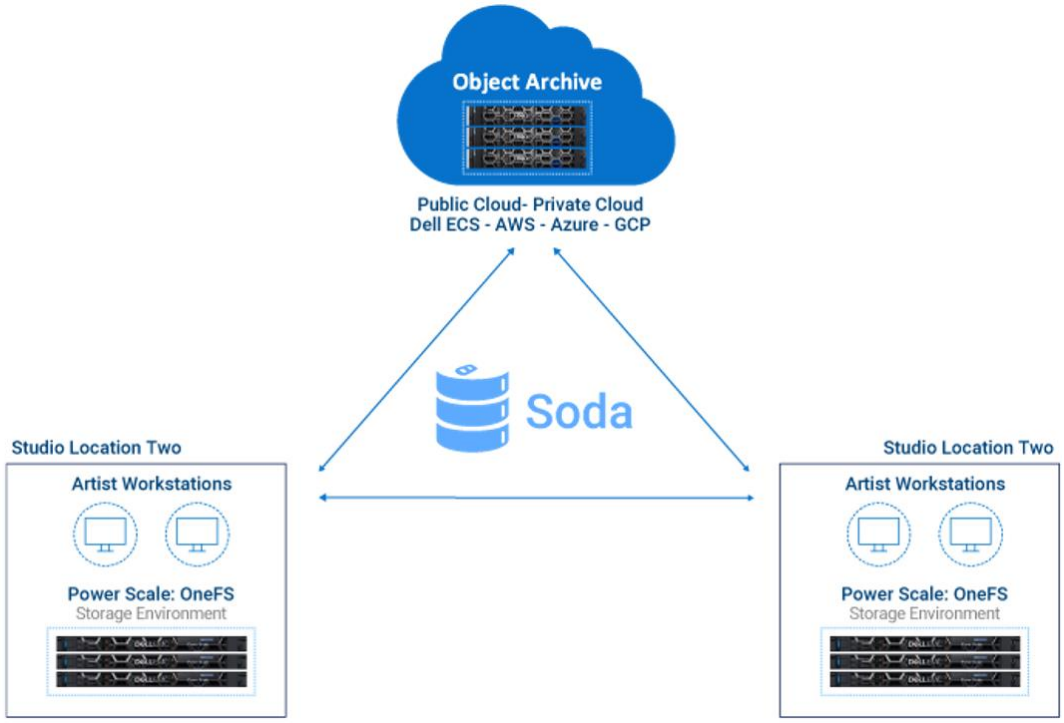

Figure 11. Multi-site production

Remote users and main production facility

In another common scenario, there are remote users working from home who need access to production data and an automated approach for moving newly created media back to the main facility. CloudSoda provides a secure way for automated data moves to remote users. Complementary policies can be configured to move data back to an object core or to performant file storage at the main studio.

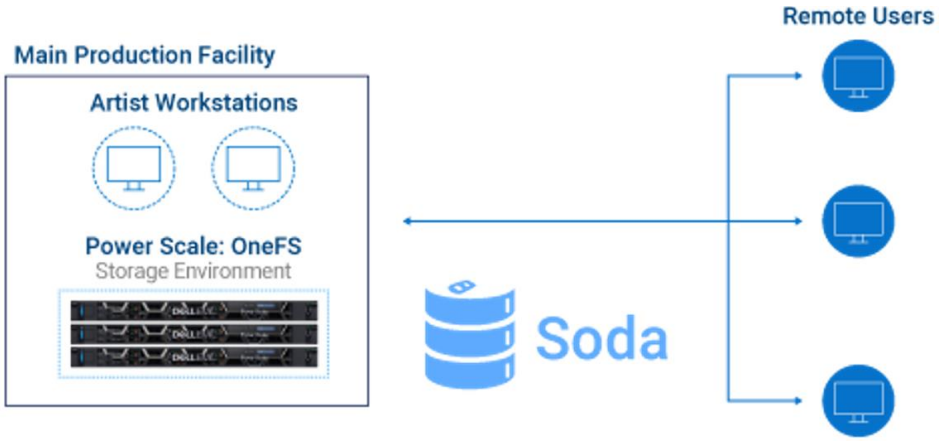

Figure 12. Remote users

Media asset management data mover

The CloudSoda API allows for external applications to trigger data moves or archives programmatically. For instance, a media asset management system can send job parameters by API to SoDA for a large data move. SoDA will in turn move the files or objects and can inform the asset management system with a webhook when the job is complete.

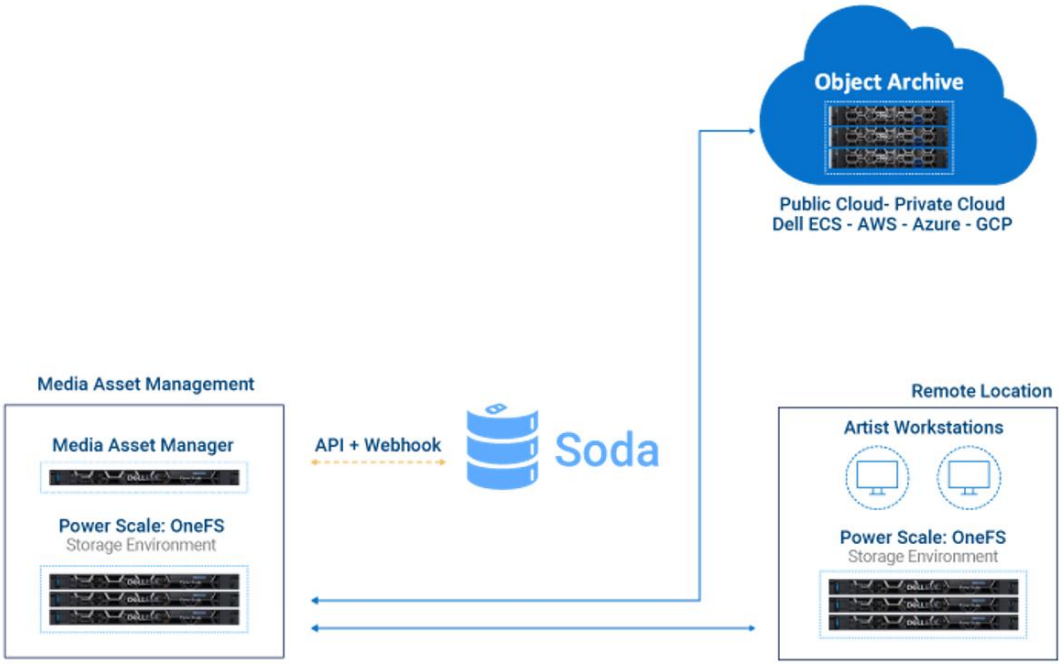

Figure 13. Media asset management data

References

Dell Technologies documentation

The following Dell Technologies documentation provides other information related to this document.

- PowerScale Info Hub
- ECS Info Hub.

CloudSoda documentation

See also the following CloudSoda documentation.

- SoDA API
- Creating a Webhook