DELL Technologies

Technical White Paper

Dell EMC Streaming Data Platform: Architecture, Configuration, and Considerations

Abstract

This document provides a technical overview and describes the design of Dell EMC Streaming Data Platform.

October 2021

H18162

Revisions

Date	Description
February 2020	Initial release
May 2020	Updated for 1.1
March 2021	Updated for 1.2
October 2021	Updated for 1.3

Acknowledgments

Author: Damien Mas

Reviewers: Russ Caldwell, Ted Schachter, Ashish Batwara, Amy Tenanes

Table of contents

Executive summary

This document describes the Dell EMC™ Streaming Data Platform (SDP), a scalable solution that is used to ingest, store, and analyze streaming data in real-time. This paper provides information about the solution components, logical and physical infrastructure, configuration details, and considerations to make when selecting and deploying a solution.

1 Introduction

The Internet of Things (IoT) brings the promise of new possibilities, but to unlock them, organizations must change how they think about data. With the emergence of IoT, there is a new class of applications that processes streaming data from sensors and devices that are spread around the globe. In theory, the solution is simple: turn massive amounts of data into real-time insights by immediately processing and analyzing it in a continuous and infinite fashion. However, managing streaming IoT data is not that simple. Legacy infrastructure is not made to support IoT data streaming from millions of data sources with varying data types. The world of streaming IoT requires a shift to the world of real-time applications consuming continuous and infinite streams.

Today, there are hundreds of applications trying to solve different pieces of the IoT puzzle. This scenario makes it difficult to build a full, end-to-end solution as the applications keep changing, have various interoperability requirements, and require their own infrastructure. Managing this complex system is costly and time consuming and requires substantial maintenance.

The Dell EMC Streaming Data Platform is designed to solve these problems. It is an ideal enterprise solution designed to address a wide range of use cases by simplifying the infrastructure stack. The solution described in this document will help customers to reunite the Operational Technology (OT) world and the IT world by providing the following key features:

- Ingest of data whether it is IOT data, sensor data, video data, log files, high frequency data, at the edge or at the core data center.
- Analyze data in real-time or batch and alert based on that specific data set.
- Centralize data from the edge to the core data center for analysis or model development (training).

1.1 Product overview

The Streaming Data Platform is an elastically scalable platform for ingesting, storing, and analyzing continuously streaming data in real-time. The platform can concurrently process both real-time and collected historical data in the same application.

The Streaming Data Platform ingests and stores streaming data from a range of sources. These sources can include IoT devices, web logs, industrial automation, financial data, live video, social media feeds, applications and event-based streams. The platform can process millions of data streams from multiple sources while ensuring low latencies and high availability.

The platform manages stream ingestion and storage, and hosts the analytic applications that process the streams. It dynamically distributes data processing and analytical jobs over the available infrastructure. Also, it dynamically and automatically scales resources to satisfy processing requirements in real-time as the workload changes. The Streaming Data Platform integrates the following capabilities into a single software platform:

- Stream ingestion: The platform ingests all types of data, whether static or streaming, in real-time. Even historical files of data, when ingested, become bounded streams of data.
- Stream storage: Elastic tiered storage provides instant access to real-time data and historical data. This loosely coupled long-term storage is what enables an unbounded digital video recorder (DVR) for all streaming data sources.
- Stream analytics: Real-time stream analysis is possible with an embedded analytics engine. Analyzing historical and real-time streaming data is now unified to simplify the application-development process.

- Real-time and historical unification: The platform can process real-time and historical data, create and store new streams, send notifications to enterprise alerting tools, and send output to third-party visualization tools.
- Platform management: Integrated management provides data security, configuration, access control, resource management, an intuitive upgrade process, health and alerting support, and network topology oversight.
- Run-time management: A web portal lets users configure stream properties, view stream metrics, run applications, and view job status.
- Application development: APIs are included in the distribution. The web portal supports application deployment and artifact storage.

In summary, the platform enables storing continuously streaming data, analyzing that data in real-time, and supports historical analysis on the stored stream.

1.2 Architecture

The Streaming Data Platform architecture contains the following key components:

- Pravega: Pravega is an open-source streaming storage system that implements streams and acts as first-class primitive for storing or serving continuous and unbounded data. This open-source project is driven and designed by Dell Technologies. See the Pravega site for more information.
- Unified Analytics: SDP includes the following embedded analytic engines for processing data stream.
  - Apache Flink®: Flink is a distributed computing engine to process large-scale unbounded and bounded data in real-time. Flink is the main component to perform streaming analytics in the Streaming Data Platform. Flink is an open-source project from the Apache Software Foundation.
  - Apache Spark™ is a unified analytics engine for large-scale data processing. SDP ships with images for Apache Spark.
  - Pravega Search (PSearch) provides search functionality against Pravega streams.
  - GStreamer is a pipeline-based multimedia framework that links together a wide variety of media processing systems to complete complex workflows. GStreamer supports a wide variety of media-handling components, including simple video playback, recording, streaming and editing. Integrated with SDP, GStreamer can record video from supported network-connected cameras to Pravega and perform real-time GPU-accelerated object detection inference on the recorded video.
- Kubernetes: Kubernetes (K8s) is an open-source platform for container orchestration. K8s is distributed through two different flavors. Kubespray for Edge deployment, and OpenShift by RedHat for Core deployment.
- Management platform: The management platform is Dell Technologies™ proprietary software. It integrates the other components and adds security, performance, configuration, and monitoring features. It includes a web-based user interface for administrators, application developers, and end users.

The Streaming Data Platform supports options for edge and core deployments. These new deployment options will allow customers to stream data from the edge to the core. Figure 1 shows a high-level overview of the Streaming Data Platform architecture streaming data from the edge to the core.

- **SDP Edge** is a small footprint deployment. Deploying SDP at the edge, where the data is generated, has the advantage of local ingestion. In addition, SDP Edge can process, filter, or enrich the collected

data at the edge, as opposed to sending all data upstream to the core. SDP Edge can be deployed on a single physical node for edge sites that don't require High Availability (HA) or on three physical nodes for edge sites that require HA.

- **SDP Micro** is a lightweight version of SDP Edge that can ingest low-volume data from sensors without gateways. SDP Micro comes with limited analytics capabilities , and can only be installed on a single virtual or physical node.
- **SDP Core** provides all the advantages of on-premise data collection, processing, and storage. It provides real-time data ingestion and also accepts data collected by SDP Edge and streamed up to the core. Deployments can start with a minimum of 3 nodes and expand out up to 12 nodes, with built-in scaling of added resource.

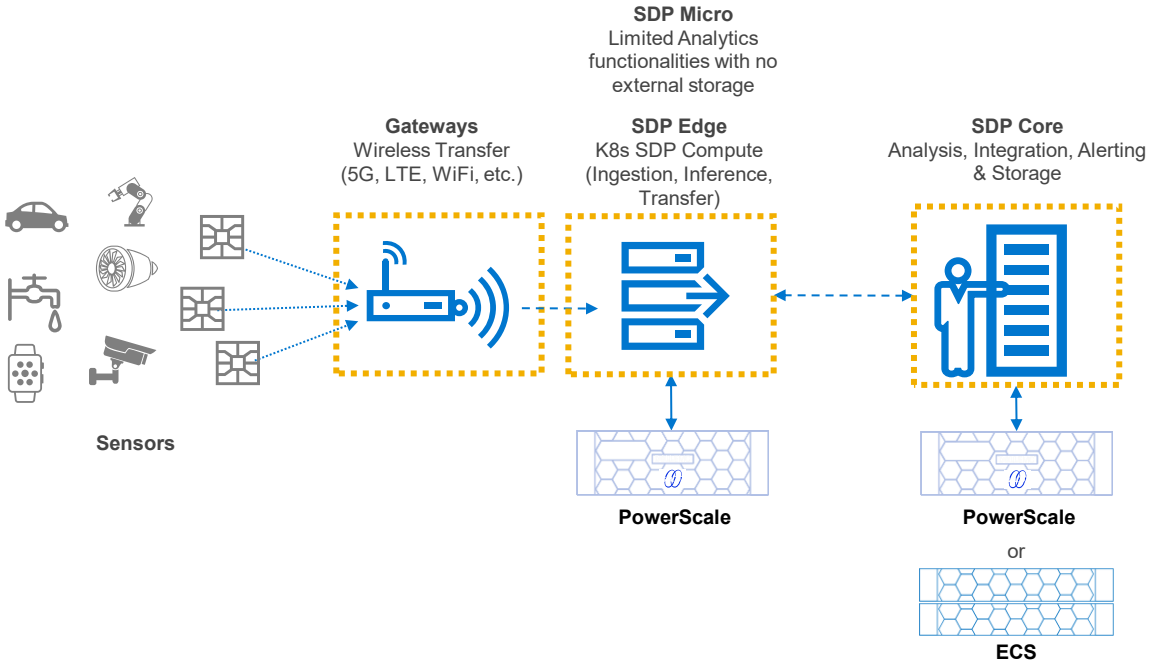

Figure 1 High Level Architecture Overview

1.2.1 SDP Edge architecture overview

Figure 2 shows a high-level depiction of the Streaming Data Platform architecture at the edge.

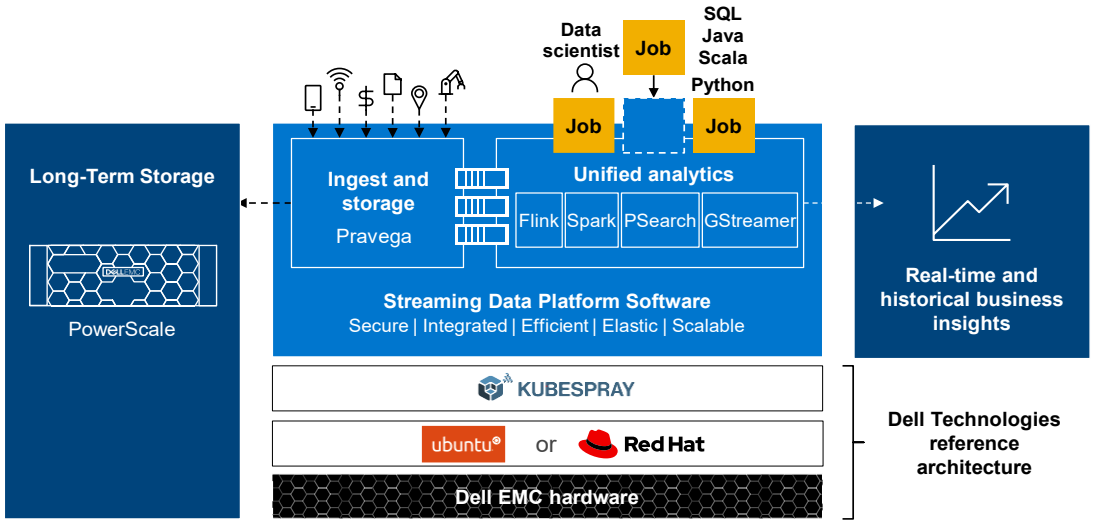

Figure 2 Streaming Data Platform architecture overview at the Edge

Note: SDP Edge only supports Dell EMC PowerScale systems for Long-Term Storage (LTS).

1.2.2 SDP Micro architecture overview

Figure 3 shows a high-level depiction of the Streaming Data Platform Micro architecture

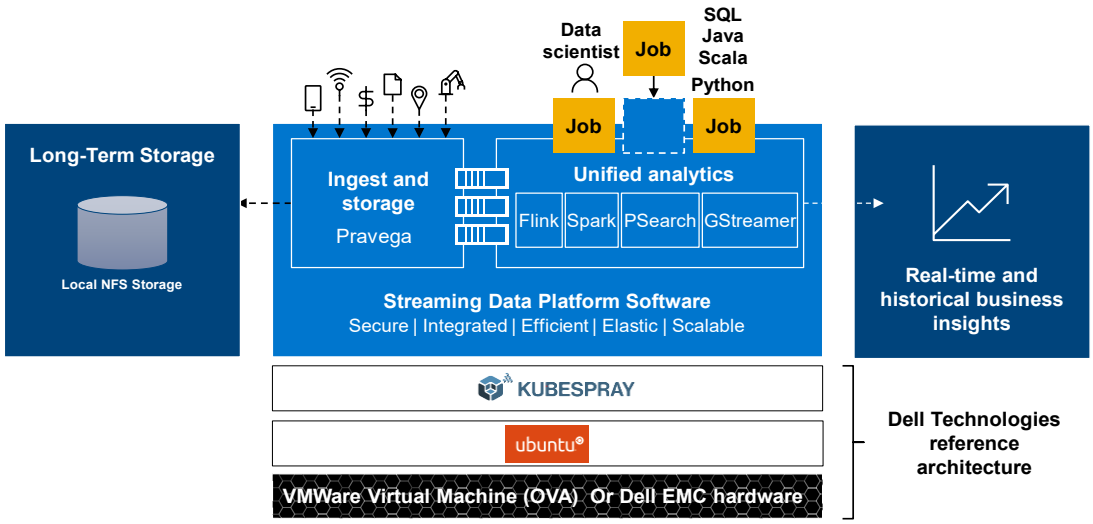

Figure 3 SDP Micro architecture overview

1.2.3 SDP Core architecture overview

Figure 4 shows a high-level depiction of the Streaming Data Platform architecture at the core. SDP Core architecture is based on Dell EMC Ready Stack for Red Hat OpenShift Container Platform 4.6.

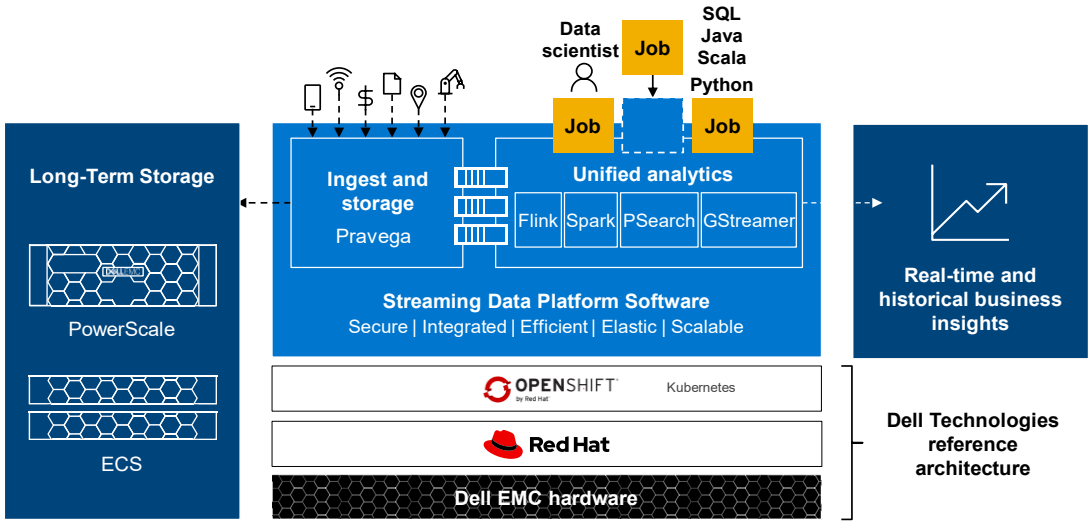

Figure 4 Streaming Data Platform architecture overview at the Core

1.3 Stream definition and scope

Pravega organizes data into Streams. According to the Pravega site, a Stream is a durable, elastic, append-only, unbounded sequence of bytes. Pravega streams are based on an append-only log-data structure. By using append-only logs, Pravega rapidly ingests data into durable storage.

When a user creates a stream into Pravega, they give it a name such as **JSONStreamSensorData** to indicate the types of data it stores. Pravega organizes Streams into Scopes. A Pravega Scope provides a secure namespace for a collection of streams and can contain multiple streams. Each Stream name must be unique within the same Scope, but there can be identical Stream names within different Scopes.

A Stream is uniquely identified by its name and the scope it belongs to. Clients can append data to a Stream (writers) and read data from the same stream (readers).

Within the Streaming Data Platform, a Scope is created in the UI by creating an analytics project. A Pravega Scope is automatically created once the analytics project is created. The name of the Pravega Scope is automatically inherited from the analytics project name, so choose the name carefully. Both names are identical.

Prior to SDP 1.2 version, each analytics project was associated to a single Pravega scope, which means that each project was completely isolated from each other. Since SDP 1.2 and later versions, it's possible to allow members of a project to be able to read Pravega streams from a different project. This will help data scientists who wants to share streams between multiple projects without the need of duplicating the data. This feature is called **Cross Project Pravega Scope Sharing.**

2 The Streaming Data Platform

This section provides an overview of the Streaming Data Platform and its components: Pravega, Flink, Spark and Pravega Search.

2.1 Pravega

Pravega is deployed as a distributed system, it forms the Pravega cluster inside Kubernetes.

The Pravega architecture presents a software-defined storage (SDS) architecture that is formed by Controller instances (control plane) and Pravega Servers (data plane) also known as Pravega Segment Store. Figure 5 illustrates an overview of the default architecture. Most of the components can be customized such as the volume size or number of replicas per stateful set or replica set.

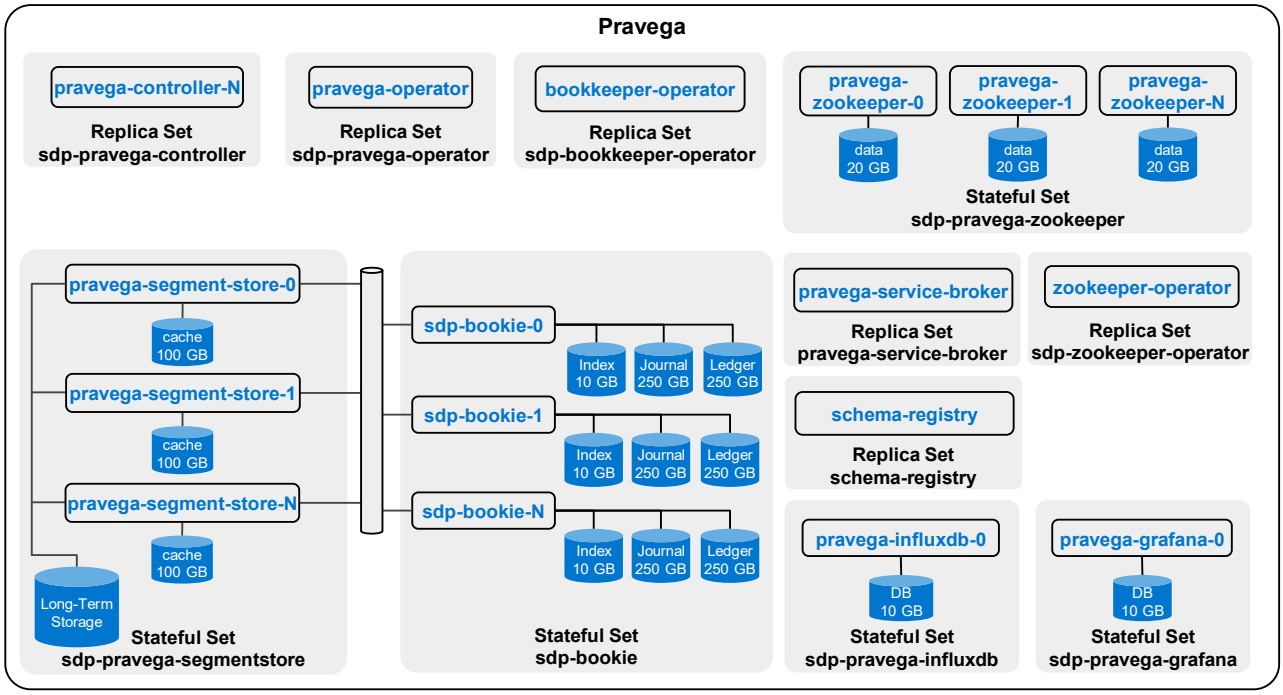

Figure 5 Pravega architecture diagram

2.1.1 Pravega Operator

The Pravega Operator is a software extension to Kubernetes. It manages Pravega clusters and automates tasks such as creation, deletion, or resizing of a Pravega cluster. Only one Pravega operator is required per instance of the Streaming Data Platforms. For more details about Kubernetes operators, see the Kubernetes page Operator pattern.

2.1.2 Bookkeeper Operator

The Bookkeeper Operator manages Bookkeeper clusters deployed to Kubernetes and automates tasks related to operating a Bookkeeper cluster such as Create and destroy a Bookkeeper cluster, Resize cluster and Rolling upgrades.

2.1.3 Zookeeper Operator

Manages the deployment of Zookeeper clusters in Kubernetes.

2.1.4 Pravega service broker

The Pravega service broker creates and deletes Pravega Scopes. It also registers them as protected resources in Keycloak along with related authorization policies.

2.1.5 Pravega Controller

The Pravega Controller is a core component in Pravega that implements the Pravega control plane. It acts as central coordinator and manager for various operations that are performed in the Pravega cluster such as actions to create, update, seal, scale, and delete streams. It is also responsible for distributing the load across the different Segment Store instances. The set of Controller instances form the control plane of Pravega. They extend the functionality to retrieve information about the Streams, monitor the health of the Pravega cluster, gather metrics, and perform other tasks. Typically, there are multiple Controller instances (at least three instances are recommended) running in a cluster for high availability.

2.1.6 Pravega Segment Store

The Segment Store implements the Pravega data plane. It is the main access point for managing Stream Segments, which enables creating and deleting content. The Pravega client communicates with the Pravega Stream Controller to identify which Segment Store must be used. Pravega Servers provide the API to read and write data in Streams. Data storage includes two tiers:

- Tier 1: This tier provides short-term, low-latency data storage, guaranteeing the durability of data written to Streams. Pravega uses Apache Bookkeeper™ to implement tier 1 storage. Tier 1 storage typically runs within the Pravega cluster.
- Long-Term Storage (LTS): This tier provides long-term storage for Stream data. The Streaming Data Platform supports Dell EMC Isilon and Dell EMC ECS to implement Long-Term Storage. LTS is commonly deployed outside the Pravega cluster.

The number of segment store is customizable and can be scaled depending on the workload.

2.1.7 Pravega Zookeeper

Pravega uses Apache Zookeeper™ to coordinate with the components in the Pravega cluster. By default, three Zookeeper servers are installed.

2.1.8 Pravega InfluxDB

The Pravega influxDB is used to store Pravega metrics.

2.1.9 Pravega Grafana

Pravega Grafana dashboards show metrics about the operation and efficiency of Pravega.

2.1.10 Pravega Schema Registry

Pravega Schema Registry is the latest service offering from Pravega family. The registry service is designed to store and manage schemas for the unstructured data stored in Pravega streams. The service is designed to not be limited to the data stored in Pravega and can serve as a general purpose management solution for storing and evolving schemas in wide variety of streaming and non-streaming use cases. Schema Registry

provides RESTful interface to store and manage schemas under schema groups. Users can safely evolve their schemas within the context of the schema group based on desired schema compatibility policy configured at a group level. For more details about Schema registry please see official repository link.

2.1.11 Pravega Bookkeeper

Pravega uses Apache Bookkeeper. It provides short-term, low-latency data storage, guaranteeing the durability of data written to Streams. In deployment, use at least five bookkeepers (bookies): three bookies for a quorum plus two bookies for fault-tolerance. By default, three replicas of the data must be kept in Bookkeeper to ensure durability.

Table 1 describes the four parameters in Bookkeeper that are configured during the Streaming Data Platform installation. For more details, please refer to the Installation and Administration Guide document.

Table 1 Bookkeeper parameters

Parameter name	Description
bookkeeper replicas	The number of bookies needed in the cluster
bkEnsembleSize	The number of nodes the ledger is stored on. bkEnsembleSize = bookkeeper replicas - F **F** represents the number of bookie failures tolerated. For instance, wanting to tolerate two failures, at least three copies of the data are needed (bkEnsembleSize = 3). To enable two faulty bookies to be replaced, instantiate two additional bookies, with a total of five bookkeeper replicas.
bkWriteQuorumSize	This parameter corresponds to the number of replicas of the data to ensure durability.
bkAckQuorumSize	By default, the following is true: `bkWriteQuorumSize == bkAckQuorumSize` The platform waits for the acknowledgment of all bookies on a write to go to the next write.

2.1.12 Pravega data flow

The following steps and diagrams outline the processes for write and read data flows.

Write data flow (Figure 6):

1. The Client contacts the Controller to identify where to perform the write.
2. The Controller returns the segment and the Segment Store URL where to write the data.
3. The Client writes to the Segment Store.
4. The data is written to Tier-1 in Apache Bookkeeper.
5. The Client receives an acknowledgment from Pravega confirming that the data has been written. In parallel the data is stored in the Segment Store cache volume.
6. Asynchronously, the data is copied to long-term storage.

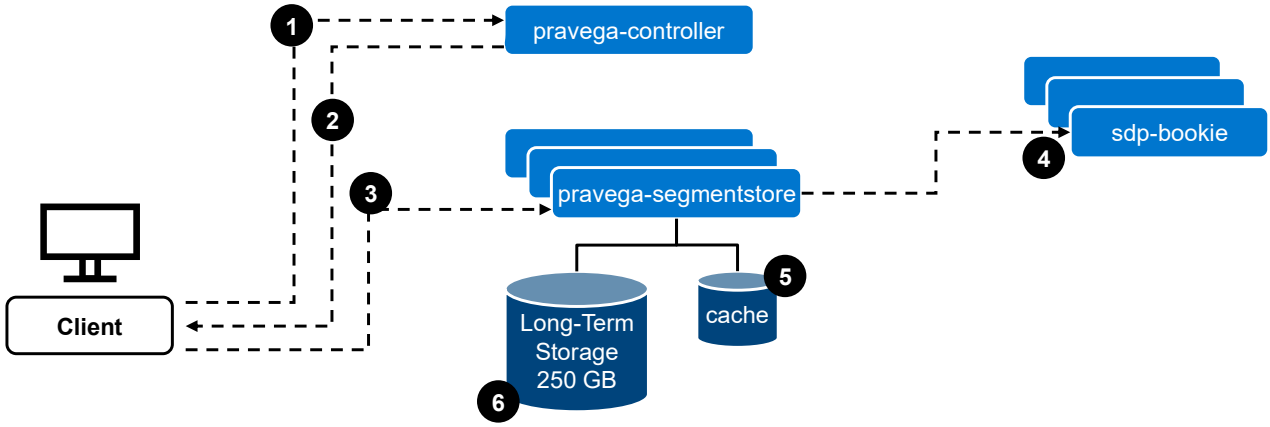

Figure 6 Write Data Flow

Read data flow (Figure 7):

7. The client contacts the Controller to identify where to perform the read.
8. The Controller returns the segment and the Segment Store url where to read the data.
9. Data is requested to the Segment Store.
10. The Segment Store reads from cache or Long-Term Storage, depending on where the data is stored. This information is hidden from the client point of view.
11. The data is returned to the client.

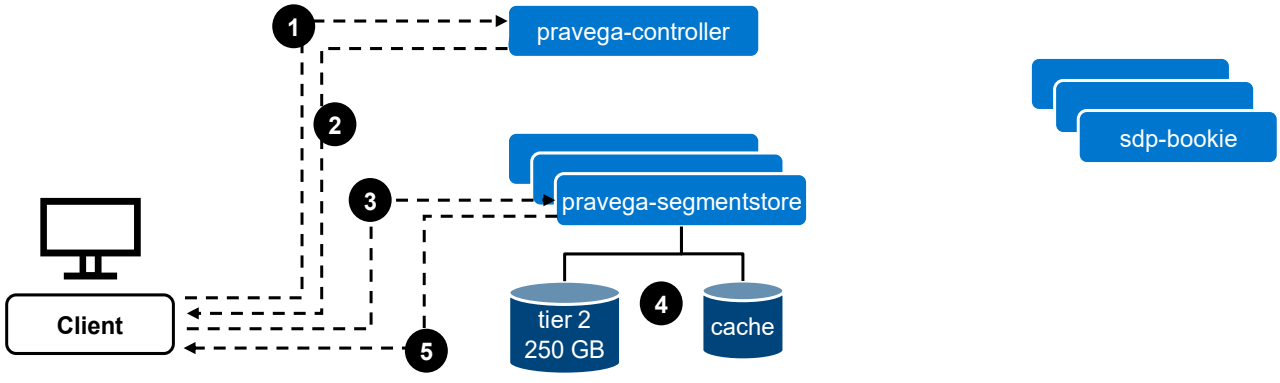

Figure 7 Read Data Flow

Note: Apache Bookkeeper is not used in 'read data flow' scenario. The data that is stored in Apache Bookkeeper is only used for recovery purposes.

2.2 Analytics project

The Streaming Data Platform provides analytic compute capabilities in the form of a managed Apache Flink or Apache Spark environments. In SDP, each Flink or Spark environments are tied to an analytics project. An analytics project is an isolated environment for streaming or analytic processing. The provisioning process of an analytic project creates the following components:

- Security credentials for the project
- A Pravega Scope (with the same name as the project) secured by the project credentials
- Storage for project analytic components (backed by NFS or ECS S3)
- Integrated metrics for monitoring
- A Kubernetes namespace (with the same name as the project) containing common infrastructure components:
  - A Zookeeper cluster (three nodes by default)
  - A secure Project artifact repository (accessible from outside the cluster with a dedicated DNS name). SDP supports Maven coordinates for or file path.
  - Kubernetes secrets containing the project credentials

SDP (1.3 and later versions) takes advantage of GPUs (Graphics Processing Unit) for image and video processing, stream processing, and machine learning. The GPU-accelerated workload adds support for both Apache Flink and Apache Spark applications, which enables data scientists to use machine learning on analytic workloads. For the moment, only NVIDIA's GPUs are supported.

Please note that since SDP 1.3 version, SDP supports MQTT and REST-based ingest gateway. That means any application that can use these protocols can directly ingest the data into SDP without integrating with the Pravega client.

2.2.1 Apache Flink

Flink Clusters can be easily deployed into analytics projects with SDP automatically configuring Flink clusters with Pravega access credentials, storage and HA configuration. The Flink Application lifecycle is also managed by SDP providing an easy way to deploy, stop, start and migrate Flink Applications onto Flink clusters.

SDP 1.3 ships with images for Flink 1.10.x, 1.11.x, and 2.12.

Once the analytics project has been created, the user can create one or more Flink clusters depending on their needs. By default, a Flink cluster is composed of one job manager and *n* task managers. The number of task managers within the cluster can be scaled at any time. SDP automatically configures Flink Clusters with the correct Pravega credentials, storage and high availability configuration reducing the burden on administrators. See Figure 8 for a diagram of a Flink cluster within an analytics project.

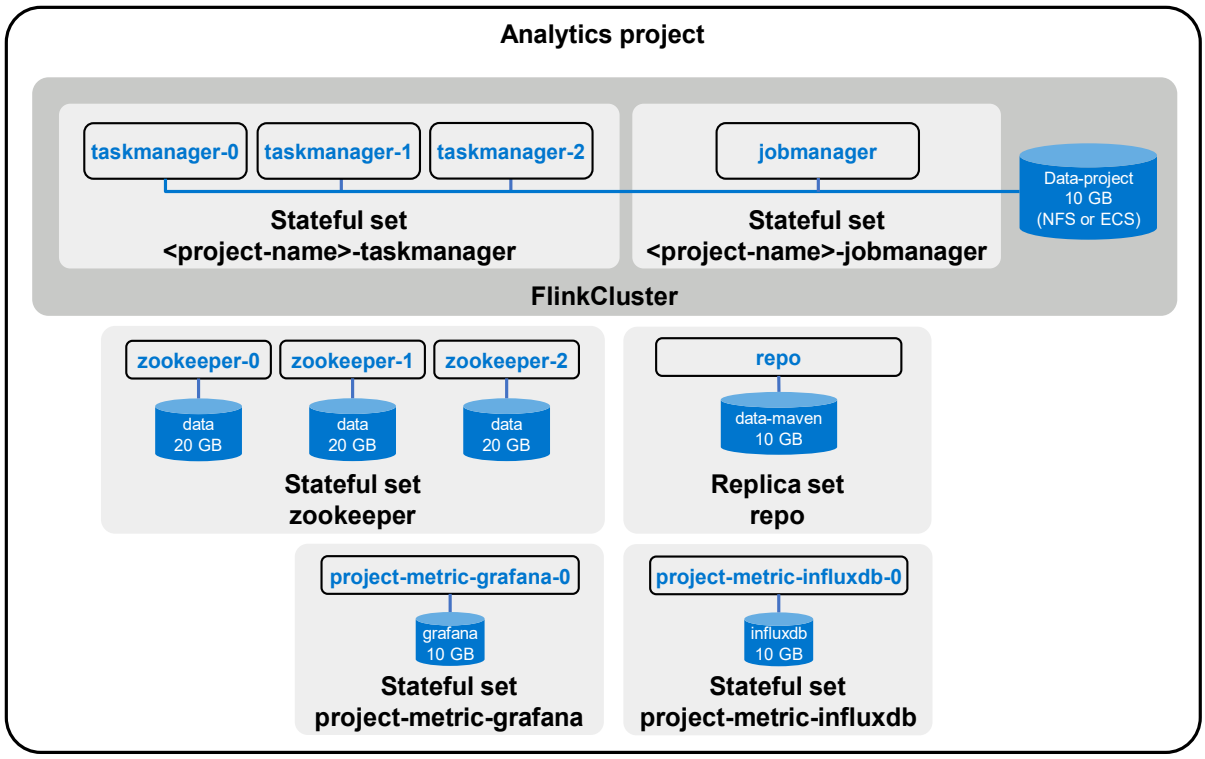

Figure 8 Flink analytics project diagram

2.2.2 Apache Spark

Spark applications can be easily deployed into analytics projects with SDP automatically configuring Spark stacks with Pravega access credentials, storage and HA configuration. The Spark Application lifecycle is also managed by SDP providing an easy way to deploy, stop, start and migrate Spark Applications.

SDP 1.3 ships with images for Spark 2.4.7 and 3.0.1. Java, Scala, and Python are the supported programming languages.

The main difference is, the Spark cluster is the application. With Spark, there is no concept of a session cluster as there is with Flink. The Spark cluster is split into the driver and the executors. The driver runs the main application and manages the scheduling of tasks. The executors are workers that perform tasks.

When you deploy a Spark application within SDP, SDP will automatically create the Driver pod, which contains the SparkContext also known as the cluster manager. The SparkContext is responsible for communicating with Kubernetes to create executor pods. Application actions are split into tasks and scheduled on executors. See Figure 9 for a diagram of a Spark application within an analytics project.

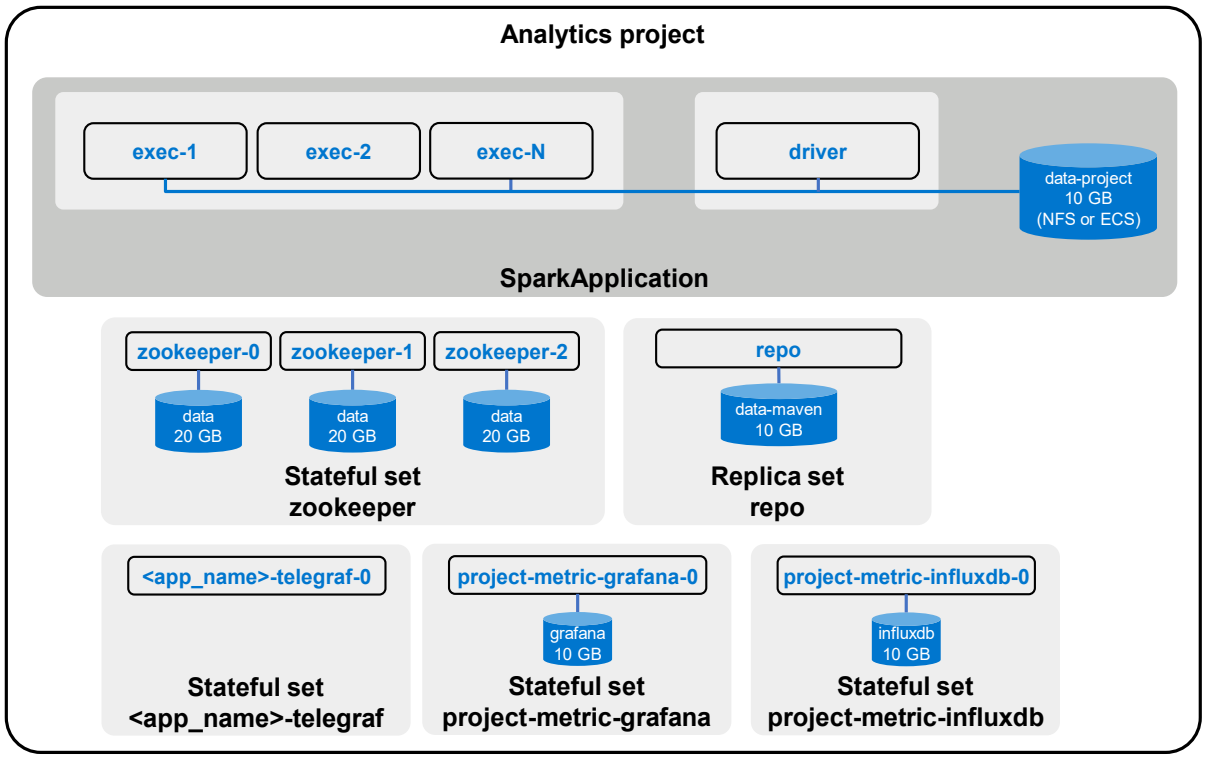

Figure 9 Spark analytics project diagram

2.2.3 Pravega Search (PSearch)

Pravega Search (PSearch) provides search functionality against Pravega streams. Pravega Search is available only as part of SDP. It is not available with Open Source Pravega. SDP provides all Pravega Search functionality in addition to comprehensive deployment, management, security, and access to the PSearch REST API. Pravega Search is implemented in the context of an SDP project. It runs in its own cluster associated with a project. A project has only one PSearch cluster. Search can be enabled or disabled at any time in the stream lifecycle.

Access to a PSearch cluster is based on project membership. Only project members and SDP Administrators can make streams searchable, submit queries, and view query results. Applications using the Pravega Search REST API must obtain project credentials.

With Pravega Search deployed in its own separated clusters, the stream indexing and query processing does not affect the efficiency of other stream processing functions. Each PSearch cluster has its own CPU and memory resources. The resources used for stream ingestion, stream storage, and analytic operations are not affected by the volume or timing of indexing and querying.

Pravega Search maintains index metadata in an infinitely expanding series of index documents, stored in system-level Pravega streams. SDP manages the index documents and all the indexing worker threads that perform the indexing and query processing. Autoscaling for Pravega Search resources is built into SDP.

For more details about PSearch, please refer to the Dell EMC Streaming Data Platform Developer's Guide.

2.2.4 Video analytics with GStreamer

The Streaming Data Platform can record video from supported network-connected cameras to Pravega and perform real-time GPU-accelerated object detection inference on the recorded video. Object detection is a type of deep learning inference that can identify the coordinates (left, top, width, height) and class objects (for example, car, bike, person) in an image. Object detection is a core requirement for many advanced video intelligence tasks, such as counting the number of people, measuring automobile traffic flow rates, and reading street signs.

SDP (1.3 and later) provides a web UI, which allows users to specify the connectivity parameters for network-connected cameras, such as IP addresses and login credentials, using Real-Time Streaming Protocol (RTSP). RTSP cameras often support both TCP and UDP transport modes of Real-Time Protocol (RTP). **SDP only supports the TCP transport mode because it is more reliable and can easily travel through firewalls.**

Retention policies can be applied to video streams, enabling automatic deletion of older videos based on bytes used or age.

Both camera recording and object detection processes use the GStreamer multimedia framework, which allows developers to build flexible and high-performance video processing pipelines.

To provide object detection, SDP developers can use the NVIDIA DeepStream SDK. The SDK provides GPU-acceleration of common video processing tasks, such as video decoding and object detection. NVIDIA DeepStream SDK delivers a complete streaming analytics toolkit for AI-based multi-sensor processing, video and image understanding. It is built around open source GStreamer framework and provides GStreamer plugins, such as GPU accelerated H.264 video encoding/decoding and deep learning inference. NVIDIA DeepStream requires a NVIDIA GPU.

3 Logical infrastructure

The Streaming Data Platform is a software-only platform running in a Kubernetes environment. This section describes the recommended architectures.

3.1 SDP Edge

The Edge is a small computer cluster that aggregates data from multiple sensors in the field. In a highly available configuration, three nodes form a Kubernetes cluster, and a distributed file system is used to store Pravega streams. A single server can be used if high availability is not required.

SDP Edge is running on a Kubernetes cluster deployed with Kubespray. Kubespray is an open-source project developed to facilitate Kubernetes deployment on Cloud platforms or Bare-metal servers. In this solution, we are using Kubespray to deploy Kubernetes cluster on Bare-metal servers at the Edge. By using Kubespray, customers can deploy SDP at the edge with a small footprint that requires fewer physical resources. Kubespray also offers the possibility to deploy Kubernetes on a single node for edge sites that do not require High Availability (HA) or on three nodes for edge sites that require HA. Ubuntu 18.04.x and RHEL 8.4.x are supported.

From a storage point of view, there are two deployment options based on the number of nodes within the Kubernetes cluster:

- Single node install can be self-contained using local NFS server as Long Term Storage (LTS), but if the customer is planning to expand from one node to three nodes cluster, PowerScale is mandatory.
- Three nodes install will use PowerScale for Long Term Storage (LTS).

From a processing point of view, there are a few options within SDP: Apache Flink, Spark, Pravega Search, and Video Analytics.

For the networking configuration, Kubespray is using calico. There is nothing to do from a user point of view as the installer package will take care of everything. The important thing to know is that for each edge deployment, end users will have to provide a range of available and unused IPs. See Table 2 for details.

Table 2 IP Reservation

Deployment Options	Number of IPs	Consumers
Single Node	3 Floating IPs, within the same subnet as the Node	1 IP for service pod 1 IP for ingress service 1 IP for Segment Store
3 Nodes	5 Floating IPs, within the same subnet as the Nodes	1 IP for service pod 1 IP for ingress service 3 IPs for Segment Store

3.2 SDP Micro

As described in Section 1.2, SDP Micro is a lightweight version of SDP Edge that can ingest low-volume data from sensors without gateways. SDP Micro and SDP Edge are very similar in terms of architecture. Both are using Kubernetes deployed with Kubespray.

The main differences between SDP Micro and SDP Edge are :

- Limited analytics capabilities
- Can only be installed on a single node (VMWare VM with an OVA file or bare-metal server)
- Only support Ubuntu 18.04
- No external storage for LTS

3.3 SDP Core

SDP Core architecture is using the Dell EMC Ready Stack for Red Hat OpenShift Container Platform 4.6, a proven design to help organizations accelerate their container deployments and cloud-native adoption. Dell EMC Ready Stack for Red Hat OpenShift Container Platform 4.6 is a flexible infrastructure that has been designed, optimized, and validated for an OpenShift Container Platform 4.6 on-premises bare-metal deployment. OpenShift Container Platform 4.6 consists of many open-source components that have been carefully integrated to provide a consistently dependable platform on which you can develop and deploy scalable containerized applications. OpenShift Container Platform provides great flexibility for accommodating platform deployment preferences.

Ready Stack for OpenShift Container Platform 4.6 includes Dell EMC hardware (servers, switches, and storage) to enable you to develop, validate, and deploy the Streaming Data Platform and more.

For more details about Dell EMC Ready stack for Read Hat OpenShift Container Platform 4.6, see the Dell Technologies Solutions Info Hub for Containers website.

4 Physical infrastructure

This section describes the recommended physical infrastructure for the Streaming Data Platform.

4.1 Servers

The solution offers three architecture options. Each architecture has different configuration options listed in Table 3.

- SDP Edge
- SDP Micro
- SDP Core

Table 3 Cluster Size

	Size	Nodes	High Availability	Kubernetes	OS	RAM	LTS	Total Cores including Hyper-Threading
SDP-Micro	X-Small	1-node bare metal or VM	No	Kubespray 2.14.2	Ubuntu 18.04.x	32 GB	Local (1x 1TB)	6
	Small	1-node bare metal or VM	No	Kubespray 2.14.2	Ubuntu 18.04.x	48 GB	Local (1x 1TB)	10
	Medium	1-node bare metal or VM	No	Kubespray 2.14.2	Ubuntu 18.04.x	96 GB	Local (2x 1TB)	12
	Large	1-node bare metal or VM	No	Kubespray 2.14.2	Ubuntu 18.04.x	128 GB	Local (2x 1TB)	16
	X-Large	1-node bare metal or VM	No	Kubespray 2.14.2	Ubuntu 18.04.x	192 GB	Local (2x 1TB)	24
SDP-Edge	Edge/Dev	1	No	Kubespray 2.14.2	Ubuntu 18.04.x, RHEL 8.4.x	256 GB	Local (2x 2TB) or PowerScale	12 Minimum
	Edge/HA	3	Yes	Kubespray 2.14.2	Ubuntu 18.04.x, RHEL 8.4.x	256 GB	PowerScale	36 Minimum
SDP-Core	Small	3	Yes	OpenShift	RHCOS 4.6.8	384 GB	ECS or PowerScale	192
	Medium	6	Yes	OpenShift	RHCOS 4.6.8	384 GB	ECS or PowerScale	384
	Large	12	Yes	OpenShift	RHCOS 4.6.8	384 GB	ECS or PowerScale	768

4.2 Switches

The Streaming Data Platform requires two top-of-rack switches. Dell EMC PowerSwitch S5200-ON series switches are recommended. They provide dual-speed 10/25 GbE (SFP+/SFP28) ports and 40/100 GbE uplinks.

The following switches are recommended based on the number of servers and future growth requirements.

- Dell EMC PowerSwitch S5248-ON
- Dell EMC PowerSwitch S5232-ON

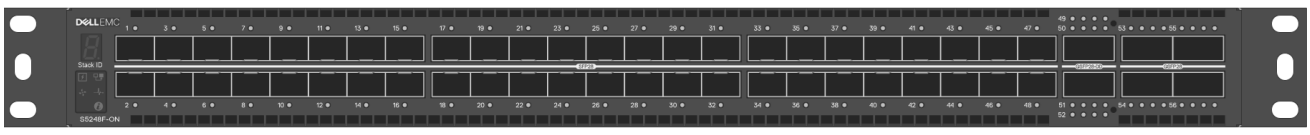

Figure 10 Dell EMC PowerSwitch S5200-ON series

4.3 Long-Term Storage (LTS)

SDP supports PowerScale and ECS as LTS. The decision must be made at installation time. It is not possible to use both storage options at the same time on the same SDP instance. Note that migration from one storage option to the other is not supported.

4.3.1 PowerScale

The Streaming Data Platform supports PowerScale systems with NFSv4/v3 as LTS for long-term and persistent storage.

PowerScale OneFS 8.2.x and 9.x are supported. Carefully select the appropriate PowerScale model depending on the expected data growth over time.

Highlights and recommendations for the PowerScale configuration include the following:

- NFSv4 is enabled on the Isilon system.
- PowerScale storage can be shared with other data center resources and does not need to be dedicated to the Streaming Data Platform.
- The best option is to connect PowerScale data network interfaces to the Streaming Data Platform infrastructure switches. If this option is not possible, ensure that the number of network HOPs are at a minimum to get the best latency.
- A best practice is to configure LACP on switches for PowerScale network interfaces data ports, but it depends on the specific configuration.

4.3.2 ECS S3 Buckets

The Streaming Data Platform supports ECS systems with S3 buckets as LTS for long-term and persistent storage.

Highlights and consideration for the ECS configuration:

- Supports ECS 3.5.1.4 and later, ECS 3.6.1.1 and later, and 3.7.0.9 and later.
- SDP 1.2 supports ONLY S3 Head (no NFS Head access).
- ONLY Access Key Security is supported (for both Pravega and Analytic Projects).
  - NO support for IAM in SDP 1.2.
- GEO replication is NOT supported.
  - All access to buckets must be via primary owning site.
- Load Balancers are supported but are not part of SDP.
  - Pravega uses ECS Smart Client and therefore can load balance at the application layer.
  - Flink is not able to load balance at application layer.
- Both HTTP and HTTPS communication is supported.
  - Also support for custom trust (i.e. Self-Signed Certificates).

A Technical support and resources

Dell.com/support is focused on meeting customer needs with proven services and support.

Storage technical documents and videos provide expertise that helps to ensure customer success on Dell Technologies storage platforms.

SDP Code Hub provide customers, developers, and integrators with a common place to find articles, guides and sample code so they can get started developing applications and integrating with the Streaming Data Platform product and Pravega streaming storage.

Dell Technologies Solutions Info Hub for Containers provides to learn about Dell Technologies solutions for Red Hat OpenShift Container Platform, a Kubernetes-based DevOps platform.

A.1 Related resources

See the following additional resources:

- http://pravega.io/
- https://kubernetes.io/