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.

May 2020

H18162

Revisions

Date	Description
February 2020	Initial release
May 2020	Updated for 1.1

Acknowledgments

Author: Damien Mas

Table of contents

Executive summary

This document describes 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.

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.

1.1 Product overview

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.

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. 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 infinite storage, and access to 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.
- 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.
- Kubernetes: Kubernetes (K8s) is an open-source platform for container orchestration. K8s is distributed through the Pivotal Container Service (PKS) running on VMware® vSphere®.
- 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.

Figure 1 shows a high-level depiction of the Streaming Data Platform architecture.

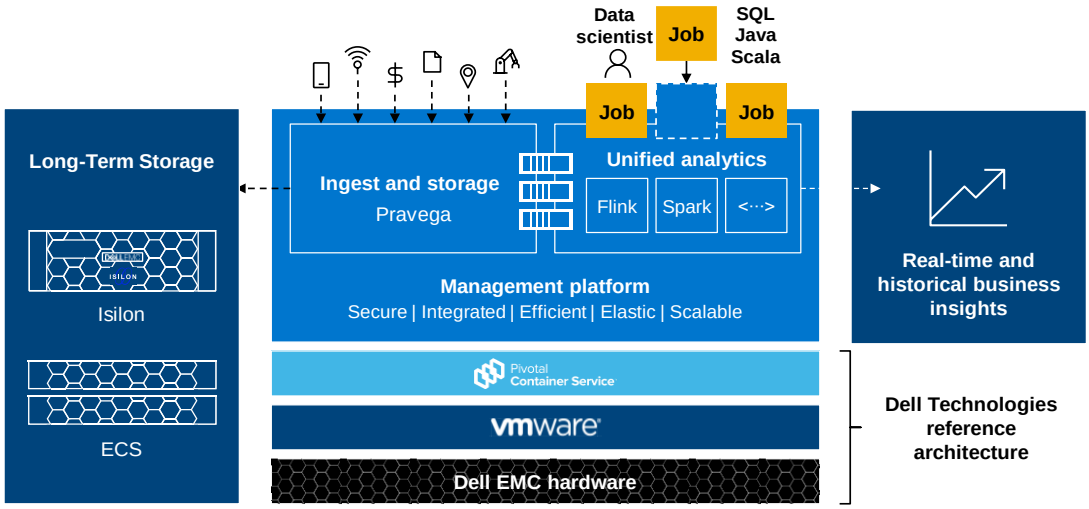


Figure 1 Streaming Data Platform architecture overview

Note: Streaming Data Platform supports Dell EMC Isilon™ systems and Dell EMC ECS for persistent storage and Apache Flink for the steaming analytics engine. Dell EMC ECS is now supported since SDP 1.1 release.

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 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.

2 Streaming Data Platform

This section provides an overview of the Streaming Data Platform and its components: Pravega, Flink, and the Pivotal Container Service (PKS).

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 2 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.

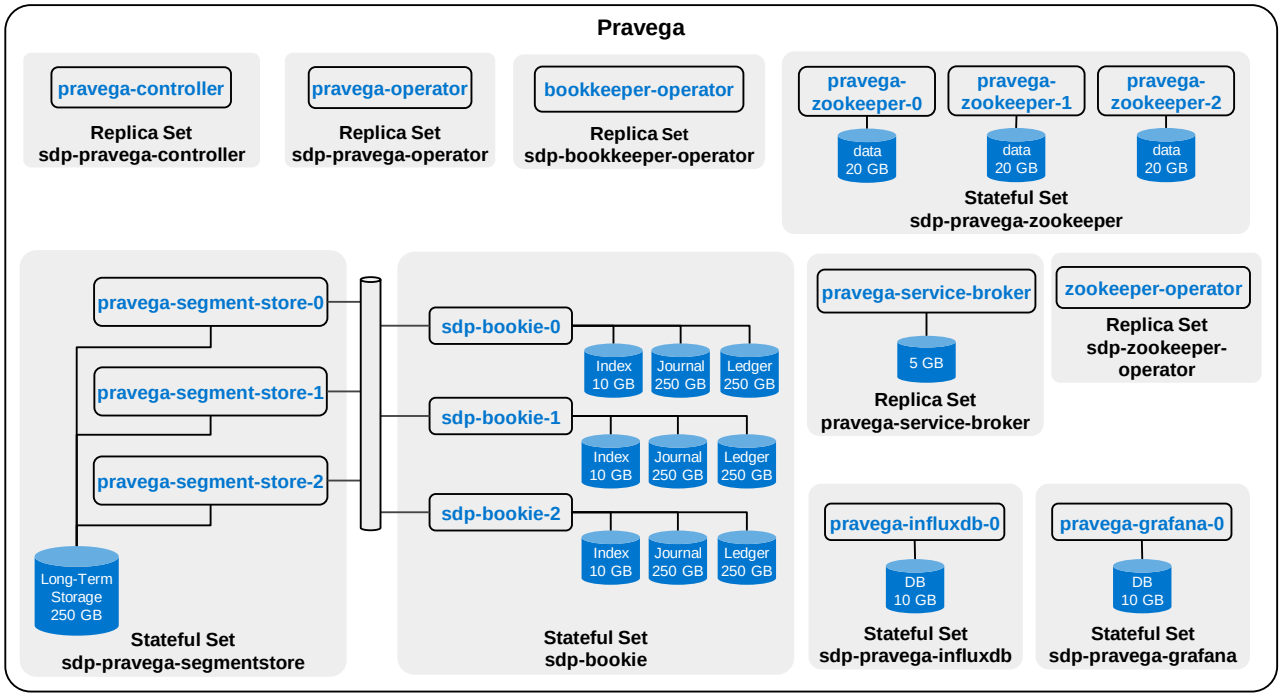


Figure 2 Pravega architecture diagram

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 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 registered 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. Streaming Data Platform supports Dell EMC Isilon and Dell EMC ECS to implement Long-Term Storage. LTS is commonly deployed outside the Pravega cluster.

By default, six Segment Stores are installed, but it is possible to increase this number 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 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.

SDP release 1.1 introduces a new configuration option to improve Bookkeeper performance. Compare to SDP 1.0 release, where Bookkeeper was provisioned with PVC coming from vSAN storage, in 1.1 customer will have the option to provision Bookkeeper with local datastores disks. In this scenario each Bookkeeper will run on a dedicated local datastore for better performance. This new option is called **Bookkeeper on BOSH**.

Note: One major change with Bookkeeper on BOSH is that Bookkeeper will not run within Kubernetes pods. Each Bookkeeper instance will be deployed within dedicated Virtual Machines. Migration from SDP 1.0 using Bookkeeper on vSAN to SDP 1.1 using Bookkeeper on BOSH is not supported.

Table 1 describes the four parameters in Bookkeeper that are configured during the Streaming Data Platform installation.

Table 1 Bookkeeper parameters

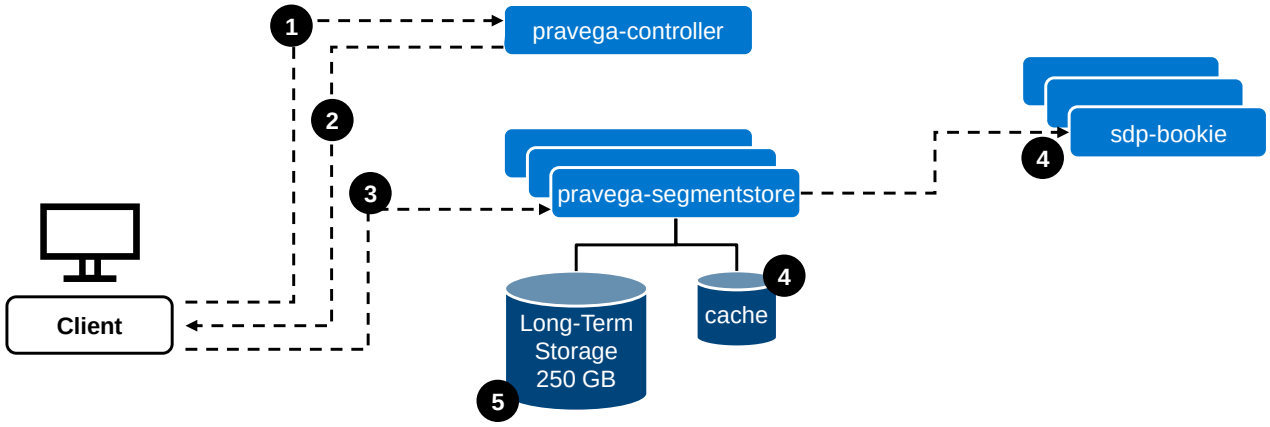
Parameter name	Description
bookkeeper replicas	The number of bookies needed in the cluster
bkEnsembleSize	The number of nodes the ledger is stored on. bkEnsembleSize = bookkeeper replicas - F **F** represents the number of bookie failures tolerated. For instance, wanting to tolerate two failures, at 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. The default value is 3, which means that the data is replicated three times on three different bookies.
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.

2.1.11 Pravega data flow

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

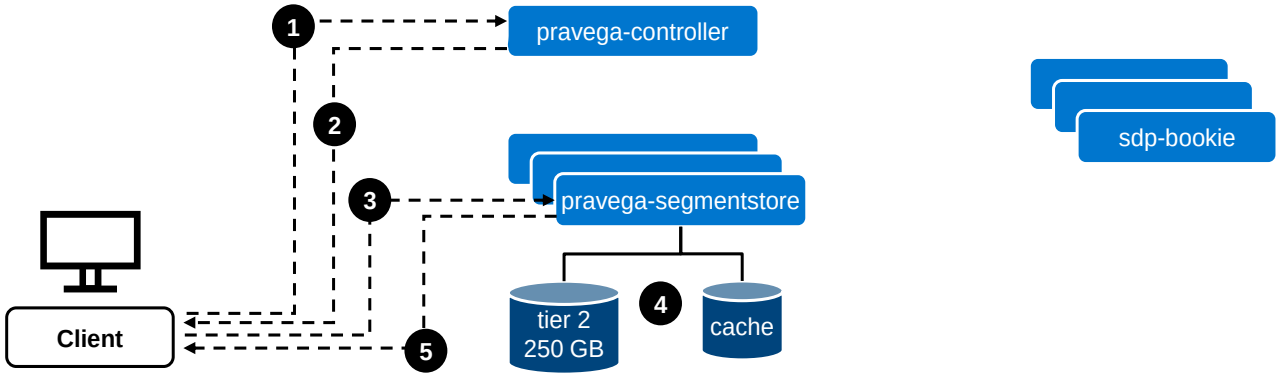
Write data flow:

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.



Read data flow:

1. The client contacts the Controller to identify where to perform the read.
2. The Controller returns the segment and the Segment Store url where to read the data.
3. Data is requested to the Segment Store.
4. 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.
5. The data is returned to the client.



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 Flink

SDP provides analytic compute capabilities in the form of a managed Apache Flink environment. Flink Clusters can be easily deployed into Analytic 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.

The SDP 1.1 Flink image environment ships with images for Flink 1.8.3, 1.9.2 and Flink 1.10.0. It also supports custom Flink images.

In Streaming Data Platform, Flink is 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:

- 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)
- A Kubernetes namespace (with the same name as the project) containing common infrastructure components:
  - A Zookeeper cluster (three nodes by default)
  - A secure Maven repository (accessible from outside the cluster with a dedicated DNS name)
  - Kubernetes secrets containing the project credentials

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 3 for a diagram of an analytics project.

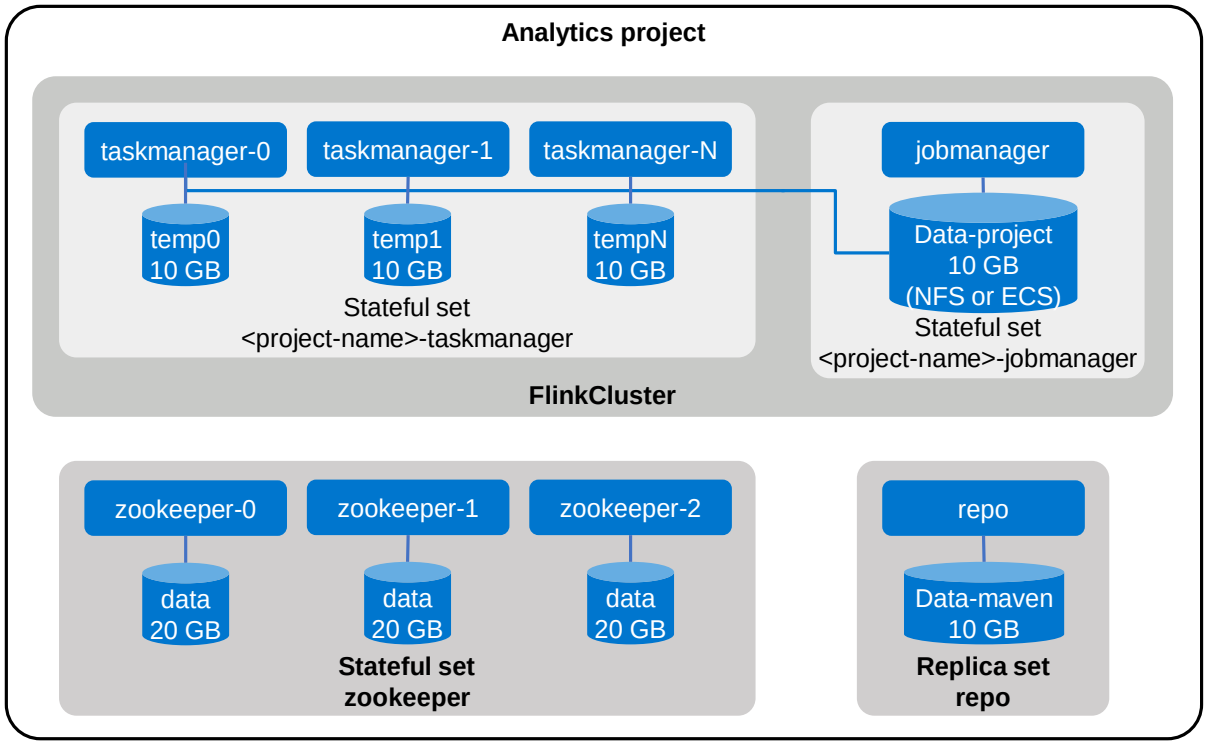


Figure 3 Analytics project diagram

2.3 Pivotal Container Service (Kubernetes)

Within the Pivotal Container Service (PKS), a Kubernetes platform, deployment configurations are known as plans. Plans contain configuration for items such as the number of workers, number of masters, and CPUs, memory, or disks per VM. These plans are used to create a PKS cluster.

Streaming Data Platform offers 2 plans:

Small (for testing):

- Name: small
- Master/ETCD Node instances: 1
- Master/ETCD VM Type: medium.disk (CPU: 2, RAM: 4 GB, disk: 32 GB)
- Master persistent disk size: 50 GB
- Master/ETCD Availability Zone: az1
- Maximum number of workers on a cluster: 50
- Worker Node instances: 3
- Worker VM Type: xlarge (CPU: 4, RAM: 16 GB, disk: 32 GB)
- Worker persistent disk size: 50 GB
- Worker Availability Zone: az1

Large (for production):

- Name: large
- Master/ETCD Node instances: 3
- Master/ETCD VM Type: medium.disk (CPU: 2, RAM: 4 GB, disk: 32 GB)
- Master persistent disk size: 30 GB
- Master/ETCD Availability Zone: az1
- Maximum number of workers on a cluster: 50
- Worker Node instances: 5
- Worker VM Type: 2xlarge (CPU: 8, RAM: 32 GB, disk: 64 GB)
- Worker persistent disk size: 50 GB
- Worker Availability Zone: az1

Note: The number of worker node defined in these plans are corresponding to the default values. In real SDP deployment this value may change.

3 Logical infrastructure

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

VMware ESXi™ is installed on each physical server. Within SDP 1.1 there is no more dedicated management nodes. Every physical servers are now part of a single VMware cluster.

Deployed within VMware vCenter® are NSX-T, OPS Manager, Enterprise Pivotal Container Service (PKS), BOSH, and VMware Harbor Registry.

SDP supports PKS version 1.6.1.

PKS is responsible for managing each new VM and deploying K8s clusters. Only one SDP instance can run in a K8s cluster, forming a one-to-one relationship. Deploying multiple SDP instances requires deploying others K8s clusters. The K8s cluster is the PKS cluster. The creation of PKS cluster is simple and is performed with a single command. The only limitation is the physical resources available in the VMware vCenter cluster.

In Streaming Data Platform 1.1 there are four deployments options. See Table 2 for more details:

Table 2 Cluster Size

Size	Physical Servers	Logical Infrastructure
Minimal	4	See Figure 4
Small	6	See Figure 4
Medium	12	See Figure 4
Large	24	See Figure 4

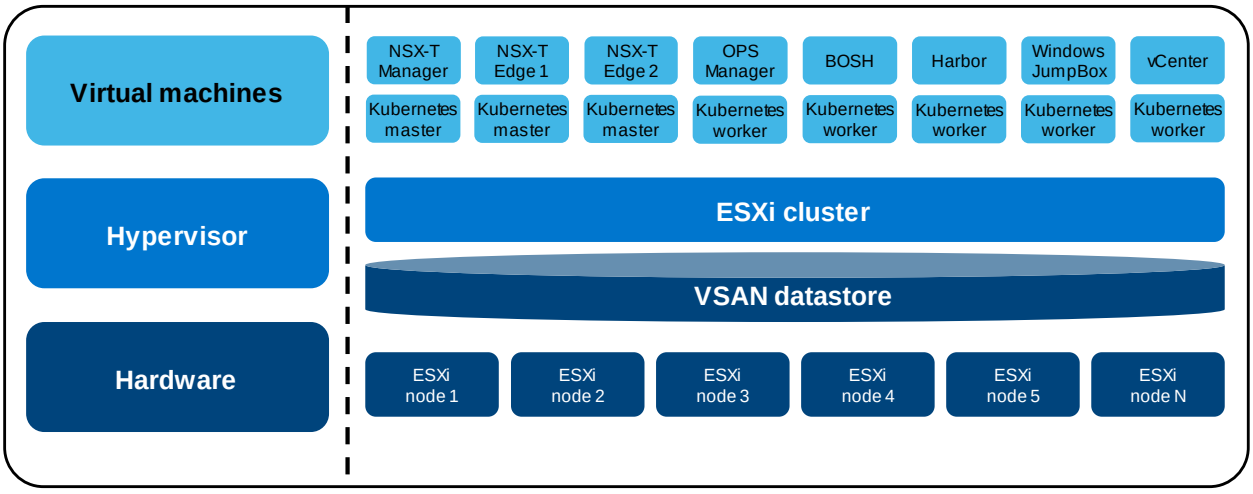


Figure 4 Logical diagram of the Streaming Data Platform infrastructure

3.1 Pivotal components

This section describes the Pivotal components of the solution.

3.1.1 Operations Manager

Pivotal Operations Manager (Ops Manager) provides a user interface to manage the deployment of Pivotal components like Enterprise PKS, BOSH, and Harbor Registry.

3.1.2 Pivotal Container Service

Streaming Data Platform requires a Kubernetes (K8s) environment to run. Pivotal Container Service (PKS) is used to run the K8s cluster. PKS is an enterprise Kubernetes platform that simplifies managing the Kubernetes cluster. It also provides functionalities to quickly scale up or scale down the environment, based on the current workload.

3.1.3 BOSH Director for vSphere

BOSH Director for vSphere is a powerful tool that can provision and deploy software over multiple VMs. It is a key element within the Pivotal platform. PKS uses BOSH to run and manage Kubernetes clusters.

3.1.4 Harbor

Harbor is a Docker registry that comes with PKS. It is used to store Streaming Data Platform Docker images.

3.2 vSAN

VMware vSAN is a storage virtualization software that allows managing storage with a single platform. It joins all storage devices accessible from a vSphere cluster into a shared data pool. All local disks that are provisioned from the physical cluster nodes are merged together to form the vSAN storage pool. The pool does not include nodes that are dedicated for booting or local resources. With vSAN, there is no requirement to deploy or maintain separate arrays and storage networking hardware.

Streaming Data Platform uses vSAN to provision storage for VMs and also as a storage class in the Kubernetes cluster. The storage class in Kubernetes is used to dynamically provision persistent volumes (PV) to the different pods and containers. A pod consumes a persistent volume claim (PVC), and the PVC consumes a PV.

For more details about storage class and PVs in Kubernetes, see the Kubernetes storage concepts page.

In Streaming Data Platform 1.1 we recommend the use of 4 disks per ESXi node for vSAN (1 NVMe cache drive and 3 SSD capacity drives organized into a single Disk Group). With this new model we can reserve the remaining disks for Bookkeeper on Bosh.

For example, if there are 5 NVMe local disks and 3 SSD local disks in the ESXi node, 4 disks (1 NVMe + 3 SSD) will be dedicated for vSAN and 3 NVMe disks will be used for Bookkeeper VMs. We also recommend to keep 1 NVMe disk as spare to handle disk failure.

Highlights and recommendations for vSAN configurations include the following:

- Initially configure the best **harddisk device controller** model available.
- Use a write-intensive I/O model with the best SSD models in terms of write performance (not read performance).

- Use stripes in the vSAN default storage policy.
- Use mirror-1 failure minimum protection.
- Enable auto balance. No fault domains are required in the stand-alone clusters.
- Monitor the health and the capacity of vSAN cluster periodically.
- Use an NFS or other shared-storage datastore for management VMs to keep vSAN available only for PKS.

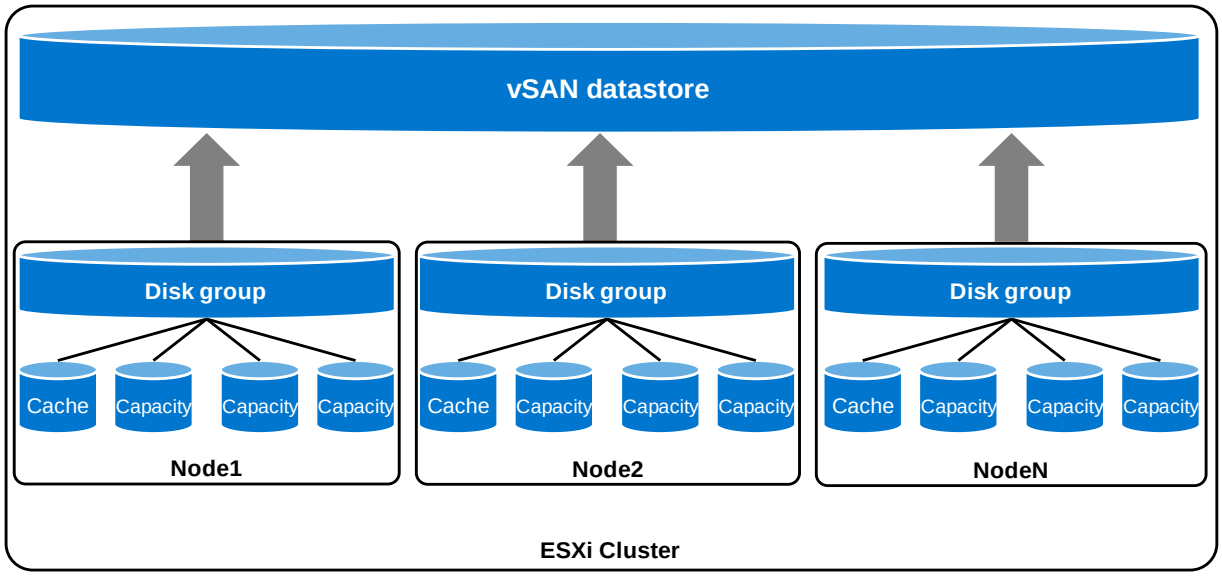


Figure 5 vSAN configuration

3.3 Logical network architecture

The following network-level configurations are available with the Streaming Data Platform architecture:

- vCenter distributed switch
- NSX-T software-defined network (SDN)

3.3.1 vCenter distributed switch configuration review

This section provides an example and best practices to follow when using four physical network interfaces per node. See Figure 6 for a diagram of this example.

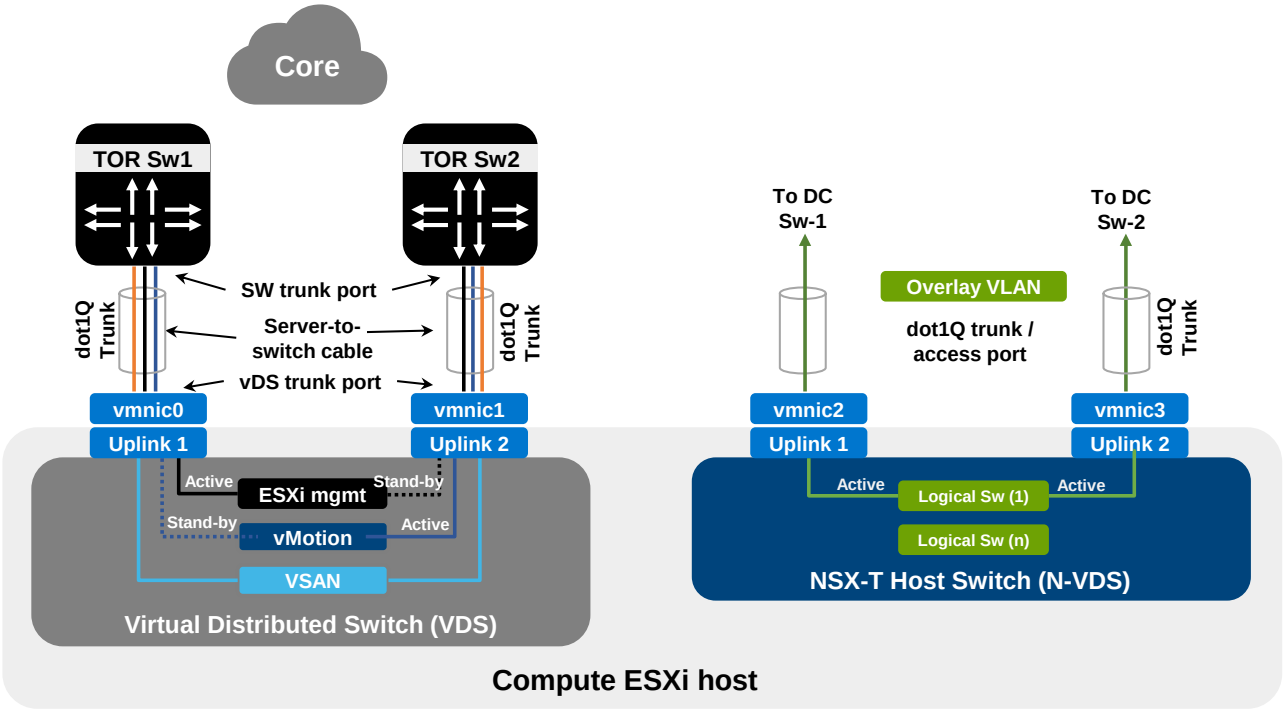
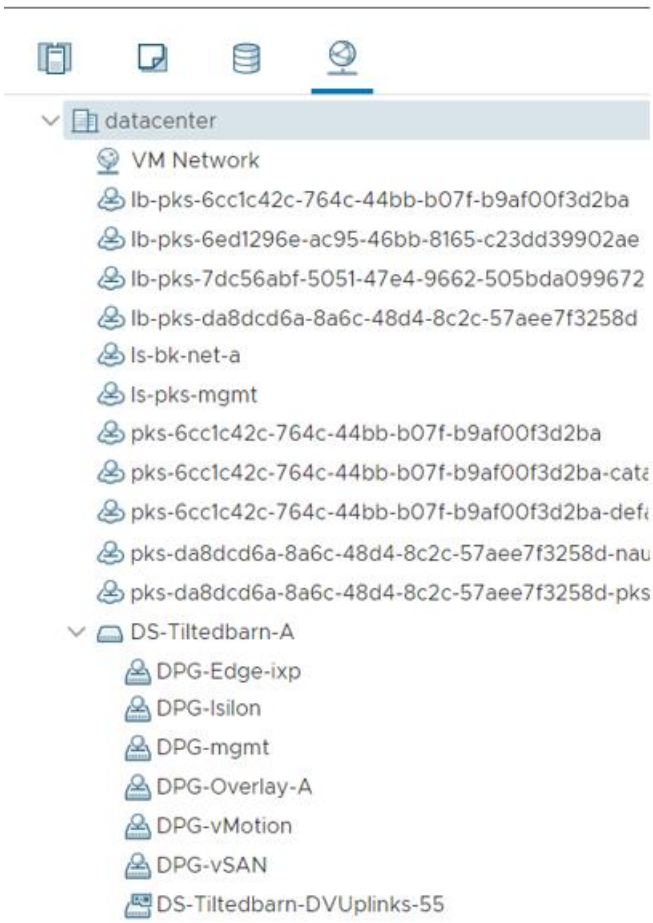


Figure 6 Example configuration with four physical network interfaces per node

Figure 7 shows an example of how to isolate and distribute different traffic types in vCenter for Streaming Data Platform.



The native distributed switch (DVS) in vCenter is DS-Titledbarn-DVUplinks-55**.**

Port groups for vmnic0 and vmnic1 are used as uplink NICS:

- DPG-mgmt (Native vLAN)
- DPG-vSAN (vSAN or VxFlex): **VLAN 103**
- DPG-vMotion: **VLAN 102**
- DPG-Overlay-A for edge VMs: **VLAN 104**
- DPG-Isilon for Isilon traffic: **vLAN 106** (could be named DPG-ECS for ECS traffic)
- DPG-Edge-ixp for edge VMs: **VLAN 105**
  - This port group routes all PKS external traffic.
  - NSX-T runs in active/passive mode. Only one NIC at 25 GbE is working.

- **ls-pks-mgmt** is the logical switch that is created by NSX-T for OpsMan/PKS/Harbor Mgmt VMs linked to NSX-T T1 router manually created for this purpose.
- **lb-pks-XXX** and **pks-XXX** are load-balancers and switches that NSX-T automatically creates for each PKS cluster.
- **ls-bk-net-a** is the logical switch that is created by NSX-T for Bookkeeper on BOSH linked to NSX-T T1 router manually created for this purpose. This only applies for Deployment using Bookkeeper on BOSH.

Figure 7 Isolating and distributing traffic types in vCenter

Highlights and best practices for distributed switches include the following

- Disable network I/O control in the DVS settings.
  - This action maximizes the vSAN throughput and avoids prelimited bandwidth in the port groups.
  - Management requires low bandwidth.
  - VMotion traffic is occasional and not continuous.
  - vSAN traffic is the most intensive.
- LACP is defined in physical switches, so this control is not required.
  - This attribute is configured as lag1 in DVS.
  - Network I/O control is not required with this configuration.
- Configure DVS advanced settings.
  - The Link Layer Discovery Protocol (LLDP) operation mode is set to **Both**.
  - Set the multicast filtering mode according to required standards.
- Configure the VLAN configuration and uplink teaming in each port group.
- Ensure that each physical server has a minimum of four 10/25 GbE network interfaces.
- Ensure redundancy with two pairs of the following:

  - One NIC pair for NSX-T overlay ESXI host network (vmnic2 and vmnic3)
  - One NIC pair for the other services: vMotion, vSAN, Edge, and overlay VM network traffic (vmnic0 and vmnic1)

- vSAN requires redundancy as a prerequisite.

3.3.2 NSX-T software-defined network

This section explores the concepts and configuration for the NSX-T software-defined network (SDN).

3.3.2.1 NSX-T concepts

NSX-T is a VMware product that replaces traditional NSX-V.

- It is based in the **Geneve** universal tunneling encapsulation protocol. It uses an encapsulating method of L2 by L3.
- The NSX-T current version is 2.5.1 (as of December 2019).
  - The minimum MTU value is **1600**. The recommended value is 9000 if Top-Of-Rack switch supports it.
  - The **Geneve** network is equivalent to an overlay network in NSX-T nomenclature.
- The edge VM cluster manages uplink traffic to the customer network external traffic.

3.3.2.2 PKS concepts

The following points apply to PKS:

- Layer 3 switches with BGP required
- T0 router:
  - Manages the physical switch routing communication
  - Requires a BGP configuration
  - Distribute the K8s public IP routes externally
- T1 routers:
  - Distributed across all ESXi hosts
  - PKS creates only T1 linked with the unique T0
- NSX-T requires subnet IP ranges (/24 subnet; floating IP pool) to publish Streaming Data Platform services
- Current T0 active-passive cluster configuration supported by PKS

3.3.2.3 NSX-T configuration for PKS

The following points apply to an NSX-T configuration for PKS:

- FLIPs (floating IP pool):
  - Required to expose Streaming Data Platform services externally (for example, Pravega Controller, ingress, Grafana, or Flink)
  - Scale-up and create more PKS clusters to get independent Streaming Data Platform instances. The number of PKS SDP clusters depends on the total number of physical nodes and the size needed for each PKS cluster
  - Scale out to add more workers (VMS) to PKS clusters to get more K8s nodes inside one Streaming Data Platform PKS cluster. For example: one Streaming Data Platform cluster can grow from three masters and five workers to 30 to 40 workers per cluster.
- IP pool (VTEPs, overlay NSX-T resource internal communication):

  - Example: 172.16.104.0/24 on VLAN 104

- IPAM IP pools (internal IPs for pods and PKS nodes)
  - IPAM range for nodes: 172.32.0.0/16
  - IPAM range for pods: 172.28.0.0/14

- Node overlay configuration:
  - vmnic2 and vmnic3 are dedicated for overlay protocol; NSX-T takes full control of these interfaces
  - Configured logically as load-balancing near soft LACP
  - Provides full internal communication for PKS/K8s Streaming Data Platform pods
  - Edge overlay communication is by vCenter DVS (they are VMs)

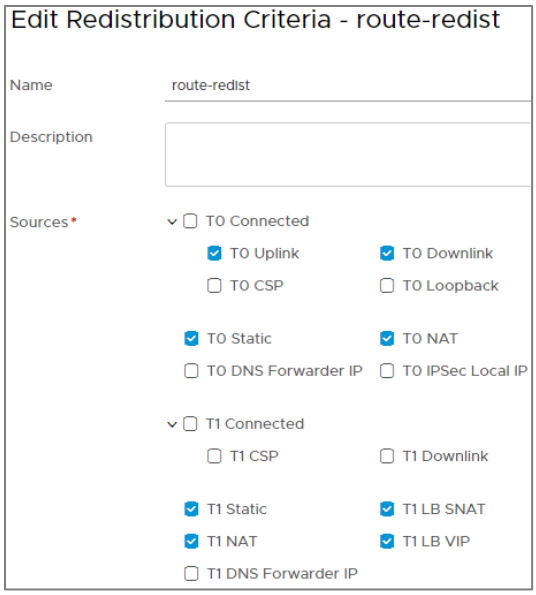
- Profiles:
  - Configuration definitions for uplinks and overlay assets
  - Good configuration key for edge-cluster-VM health

- vCenter registered to the following:
  - Communicate with all NSX-T components
  - Install kernel modules on each ESXi to manage NICs directly

- T0 router configuration considerations (only one required for PKS):
  - NAT: All management Pivotal IPs must be added manually:
    > DNAT and SNAT
    > ls-pks-mgmt switch created manually
    Reserve first seven IPs of the FLIPs range for Pivotal and other management VMs.
    Examples:

    - OpsMan: 172.16.0.2
    - Boshd: 172.16.0.3
    - PKS: 172.16.0.4
    - Harbor: 172.16.0.5
    - linux-Jumpserver: 172.16.0.6
    - DNS-Internal: 172.16.0.7

- BGP (switch configuration examples):
  - 172.16.105.20
  - 172.16.105.21
  - Neighbors: 172.16.105.2, 172.16.105.3 (physical switches)
    > Route distribution T0 described



> Disable firewall as prerequisite
> T0 NAT for internal OpsMan, PKS, and management IPs
> NAT hair-pinning
> T0 NAT and routing path distribution
> Hair-pinning: Source and destination are behind the NSX-T NAT

ID	Action	Match					Translated
		Protocol	Source IP	Source Ports	Destination IP	Destination Ports	IP
Priority: 102							
1027	SNAT	Any	172.16.0.2	Any	Any	Any	10. .37.2
1028	SNAT	Any	172.16.0.3	Any	Any	Any	10. 37.3
1029	SNAT	Any	172.16.0.4	Any	Any	Any	10. .37.4
1030	SNAT	Any	172.16.0.5	Any	Any	Any	10. .37.5
1031	DNAT	Any	Any	Any	10. .37.2	Any	172.16.0.2
1032	DNAT	Any	Any	Any	10. .37.3	Any	172.16.0.3
1033	DNAT	Any	Any	Any	10. 37.4	Any	172.16.0.4
1034	DNAT	Any	Any	Any	10. 37.5	Any	172.16.0.5
1144	SNAT	Any	172.16.0.6	Any	Any	Any	10. 37.6
1145	DNAT	Any	Any	Any	10. 37.6	Any	172.16.0.6

- T1 distributed router for management: **ls-pks-mgmt**
  - Manual operation: Only first seven IPs used by pivotal management VMs
  - Create route port: 172.16.0.1
  - No requirements for service router; association with edge cluster not required
  - Enabled route distribution

- T1 automatic routers linked to T0 created by PKS
  - Managed by PKS with API communication
  - All NSX-T objects handled by PKS
    - Highlight: PKS cluster deletion must be performed from PKS CLI to release all objects created in NSX-T; do not leave orphan objects.
    - https://code.vmware.com/apis/696/nsx-t
    - If one object must be manually deleted, use API calls.
    - Example: **DELETE /api/v1/logical-router-ports/<logical-router-port-id>**

```
curl -k -u admin:P@ssw0rd -X DELETE 'https://172.16.101.61/api/v1/logical-router-ports/e78a357e-274c-428a-9e4d-1d660b196804' -H "X-Allow-Overwrite: true"
```

- License. 60 days of evaluation
- Certificate generations required by OpsMan and PKS; generate and register the following in NSX-T:

  CA.crt and PKS-superuser certificates for OpsMan and PKS

  See the following for more information: https://docs.vmware.com/en/VMware-Enterprise-PKS/1.4/vmware-enterprise-pks-14/GUID-generate-nsx-ca-cert-24.html

3.4 Logical infrastructure overhead considerations

The virtualization layer, and in particular PKS, introduce non-negligible resources overhead. We need to have an accurate estimation of what these overheads are, as we cannot account on these resources for SDP.

Table 3 Minimal and Small deployments

VM	vCPU	Memory (GB)	Disk Space (GB)
vCenter Appliance (x2)	4	16	290
NSX-T Manager (x3)	6	24	200
Ops Manager	1	8	160
BOSH Director	2	8	103
PKS Control Plane	2	8	29
Harbor Registry	2	8	167
NSX-T Edge Node (x4)	8	32	120
TOTAL	**65**	**264**	**2119**

Table 4 Medium and Large deployments

VM	vCPU	Memory (GB)	Disk Space (GB)
vCenter Appliance (x2)	8	32	290
NSX-T Manager (x3)	8	32	200
Ops Manager	4	16	160

BOSH Director	4	16	103
PKS Control Plane	4	16	50
Harbor Registry	2	8	167
NSX-T Edge Node (x8)	8	32	120
TOTAL	**118**	**472**	**2703**

Note: One important point to consider is that for Medium/Large cluster sizes, we double the number of NSX-T Edge Nodes. The objective is to allow the cluster to double the inbound/outbound traffic capacity. We also assume that NSX-T Edge Nodes will be mapped to the 25GbE ports of physical servers where they run. It is also important to set anti-affinity policy to NSX-T Edge Nodes to spread them across physical servers.

Another thing to consider is that vSAN will consume a certain amount of physical memory on each physical server. Consider the following memory consumption for vSAN.

Table 5 vSAN overhead

Resource	Memory (GB)
RAM allocated to vSAN per physical server	64

Last considerations, it is important to set up internal DNS servers for SDP to avoid additional networking problems. Having VMs prepared for operations and troubleshooting is also recommended.

Table 6 DNS and Jumpbox overhead

VM	vCPU	Memory (GB)	Disk Space (GB)
Internal SDP DNS Server (x2)	4	8	100
Linux or Windows Jump Server (x2)	8	32	100
TOTAL	**24**	**80**	**400**

4 Physical infrastructure

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

4.1 Servers

The solution offers two physical architecture options:

- Traditional model: Recommendation is to use PowerEdge R640 series
- Dell EMC VxRail model: using VxRail Hyperconverged infrastructure. With this model, Bookkeeper on Bosh deployments are not supported.

Both models support four different deployment options (Minimal / Small / Medium / Large)

Table 7 Cluster Size

Size	Physical Servers
Minimal	4
Small	6
Medium	12
Large	24

4.1.1 Traditional model

Compute nodes are running on ESXi version 6.7.0u3 or higher. Each node is built using a Dell EMC PowerEdge™ R640 server. See Table 8 and Figure 8 for more details.

Table 8 Traditional model: compute nodes

Node type	Model	CPU	RAM	NICs	Disks
Compute	PowerEdge R640	2 Intel® Xeon® Gold 6230 CPU @ 2.10GHz, 20 cores, 40 threads Total of 80 vCPUs	384 GB DDR4-2400 or faster	2 x 25 GbE nics (for a total of 4 x 25 GbE ports) SFP28 recommended	2 x 240 GB BOSS controller, M2 for boot disk in RAID 1 PERC H330 RAID Controller, 5 x 1.6 TB NVMe Drives and 3 x 1.6 TB SSD Write-oriented performance.

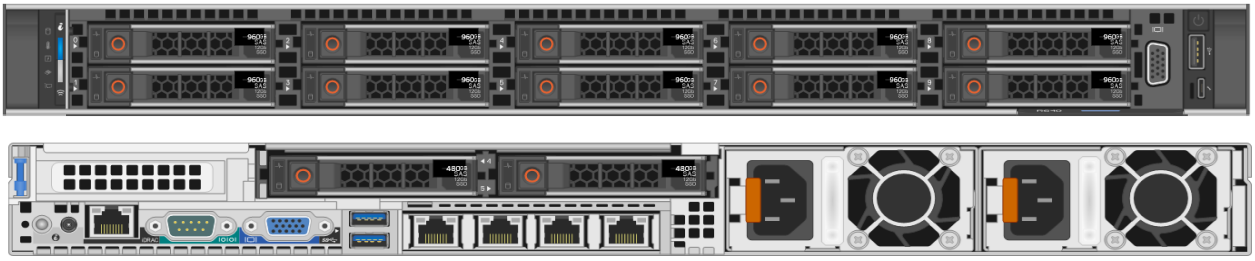


Figure 8 Traditional model: PowerEdge R640 compute nodes (front and back view)

4.2 Switches

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.

- Minimal and Small cluster with 4 or 6 servers with no growth expectation: 2 x PowerSwitch S5224-ON
- Medium cluster with 12 servers: 2 x PowerSwitch S5248-ON
- Large cluster with 24 servers: 2 x PowerSwitch S5296-ON

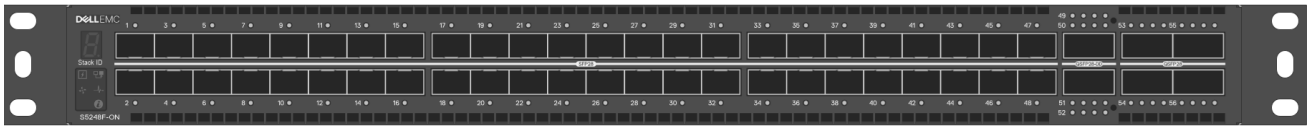


Figure 9 Dell EMC PowerSwitch S5200-ON series

Traffic can be spread over the two switches as follows:

- Internal traffic: Management and NSX-T overlay communication
- External traffic: Uplink network (NSX-T) and Long-Term storage traffic
- vCenter native traffic: vSAN, vMotion, and vCenter datastore on Isilon storage

Note: MTU for uplink ports must be set to 9216 on the switches (Internal Switches and Customer switches)

4.3 Long-Term Storage (LTS)

Streaming Data Platform 1.1 introduces a new storage option. Previous SDP release supported Isilon as LTS; SDP 1.1 introduces support for ECS as LTS in addition. This allows ECS S3 buckets to be used for Pravega long term storage and analytic project storage. 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 Isilon

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

H600, H500, H5600, H400, A200, or A2000 models are supported. Carefully select the appropriate Isilon model depending on the expected data growth over time.

Highlights and recommendations for the Isilon configuration include the following:

- NFSv4 is enabled on the Isilon system.
- Isilon storage can be shared with other data center resources and does not need to be dedicated to Streaming Data Platform.
- Isilon storage can be used to provide NFS datastores to the vCenter for management VMs, vCenter VM, and backups. Configure each node, and create a datastore cluster with DRS. This practice provides HA, redundancy, and increased throughput.
- The best option is to connect Isilon 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 Isilon network interfaces data ports, but it depends on the specific configuration.
- Each Streaming Data Platforms pod connects to Isilon storage through NSX-T edge VMs by a virtual T0 router using a vCenter DVS uplink port group.

4.3.2 ECS S3 Buckets

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 3.4.
- SDP 1.1 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 1.1.
- 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.

https://pravega.github.io/workshop-samples 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

A.1 Related resources

See the following additional resources:

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