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

February 2020
Revisions

<table>
<thead>
<tr>
<th>Date</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>February 2020</td>
<td>Initial release</td>
</tr>
</tbody>
</table>

Acknowledgments

Author: Damien Mas
# Table of contents

Revisions .............................................................................................................................................. 2
Acknowledgments .................................................................................................................................... 2
Table of contents .................................................................................................................................... 3
Executive summary ............................................................................................................................... 5
1 Introduction .......................................................................................................................................... 6
  1.1 Product overview .......................................................................................................................... 6
  1.2 Architecture .................................................................................................................................. 7
  1.3 Stream definition and scope ......................................................................................................... 8
2 Streaming Data Platform ................................................................................................................... 9
  2.1 Pravega ........................................................................................................................................ 9
  2.1.1 Pravega Operator ..................................................................................................................... 9
  2.1.2 Pravega service broker ........................................................................................................... 9
  2.1.3 Pravega Controller .................................................................................................................. 9
  2.1.4 Pravega Segment Store ......................................................................................................... 10
  2.1.5 Pravega Zookeeper ............................................................................................................... 10
  2.1.6 Pravega InfluxDB .................................................................................................................. 10
  2.1.7 Pravega Grafana ................................................................................................................... 10
  2.1.8 Pravega Bookkeeper ............................................................................................................. 10
  2.1.9 Pravega data flow .................................................................................................................. 12
  2.2 Flink .......................................................................................................................................... 13
  2.3 Pivotal Container Service (Kubernetes) ....................................................................................... 14
3 Logical infrastructure .......................................................................................................................... 15
  3.1 Pivotal components ...................................................................................................................... 15
  3.1.1 Operations Manager ............................................................................................................. 15
  3.1.2 Pivotal Container Service .................................................................................................... 15
  3.1.3 BOSH Director for vSphere ................................................................................................. 16
  3.1.4 Harbor .................................................................................................................................. 16
  3.2 vSAN .......................................................................................................................................... 16
  3.3 Logical network architecture ...................................................................................................... 17
    3.3.1 vCenter distributed switch configuration review ............................................................... 18
    3.3.2 NSX-T software-defined network ...................................................................................... 20
4 Physical infrastructure ....................................................................................................................... 24
  4.1 Servers ....................................................................................................................................... 24
    4.1.1 Traditional model ................................................................................................................. 24
Table of contents

4.1.2 Consolidated model.........................................................................................................................26
4.2 Switches ..............................................................................................................................................27
4.3 Isilon tier 2 storage ..............................................................................................................................27
A Technical support and resources ........................................................................................................28
A.1 Related resources .................................................................................................................................28
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.
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 data 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 and simplify 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 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 it 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 and 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: The initial release of Streaming Data Platform supports only Dell EMC Isilon™ systems for persistent storage and Apache Flink for the streaming analytics engine. Since Pravega is an open-source project, it supports different technologies which are not necessarily supported by Dell Technologies.
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).

In 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.

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 Pravega service broker

The Pravega service broker creates and deletes Pravega Scopes.

2.1.3 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 all 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.

2.1.4 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 is 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.
- Tier 2: This tier provides long-term storage for Stream data. Streaming Data Platform only supports Dell EMC Isilon to implement tier 2 storage. Tier 2 storage 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.

2.1.5 Pravega Zookeeper

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

2.1.6 Pravega InfluxDB

The Pravega influxDB is used to store Pravega metrics.

2.1.7 Pravega Grafana

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

2.1.8 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.

Table 1  Bookkeeper parameters

<table>
<thead>
<tr>
<th>Parameter name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>bookkeeper replicas</td>
<td>The number of bookies needed in the cluster</td>
</tr>
<tr>
<td>bkEnsembleSize</td>
<td>bkEnsembleSize = bookkeeper replicas - F</td>
</tr>
<tr>
<td></td>
<td>F represents the number of bookie failures tolerated. For instance, wanting</td>
</tr>
<tr>
<td></td>
<td>to tolerate two failures, at least three copies of the data are needed</td>
</tr>
<tr>
<td></td>
<td>(bkEnsembleSize = 3). To enable two faulty bookies to be replaced,</td>
</tr>
<tr>
<td></td>
<td>instantiate two additional bookies, with a total of five bookkeeper replicas.</td>
</tr>
<tr>
<td>bkWriteQuorumSize</td>
<td>This parameter corresponds to the number of replicas of the data to ensure</td>
</tr>
<tr>
<td></td>
<td>durability. The default value is 3, which means that the data is replicated</td>
</tr>
<tr>
<td></td>
<td>three times on three different bookies.</td>
</tr>
<tr>
<td>bkAckQuorumSize</td>
<td>By default, the following is true:</td>
</tr>
<tr>
<td></td>
<td>bkWriteQuorumSize == bkAckQuorumSize</td>
</tr>
<tr>
<td></td>
<td>The platform waits for the acknowledgment of all bookies on a write to go to</td>
</tr>
<tr>
<td></td>
<td>the next write.</td>
</tr>
</tbody>
</table>
2.1.9 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 Store where to write the data.
3. The client writes to the Segment Store.
4. The Segment Store processes the data into its cache volume.
5. The data is written to tier 1 in Apache Bookkeeper. The client receives an acknowledgment from Pravega confirming that the data has been written.
6. Asynchronously, the data is copied to tier 2 long-term storage.

![Write Data Flow Diagram]

Read data flow:

1. The client contacts the Controller to identify where to perform the read.
2. The Controller returns the Segment Store where to read the data.
3. Data is read from the Segment Store.
4. The Segment Store reads from cache or tier 2 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.

![Read Data Flow Diagram]

Note: Apache Bookkeeper is not used in this scenario. The data that is stored in Apache Bookkeeper is only used for recovery purposes.
2.2 **Flink**

Flink is the embedded analytics engine in Streaming Data Platform. It processes streams in real time. Flink is an open-source project from the Apache Software Foundation. Streaming Data Platform supports Flink version 1.7.2, 1.8.1, 1.9.0, and also custom Flink images (such as images for machine learning workloads). GPU scheduling is not supported in Streaming Data Platform 1.0.

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
- NFS project file storage (as a ReadWriteMany Kubernetes Persistent Volume in the Kubernetes namespace)
- 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 the needs. The 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. Flink clusters are automatically injected with the Pravega credentials, allowing applications to access the Pravega Scope and process the data. See Figure 3 for a diagram of an analytics project.

![Analytics project diagram](image-url)
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 three plans:

**Small:**
- Name: small
- Master/ETCD Node instances: 1
- Master/ETCD VM Type: medium.disk (CPU: 2, RAM: 4 GB, disk: 32 GB)
- Master persistent disk type: 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 type: 50 GB
- Worker Availability Zone: az1

**Medium:**
- Name: medium
- Master/ETCD Node instances: 3
- Master/ETCD VM Type: medium.disk (CPU: 2, RAM: 4 GB, disk: 32 GB)
- Master persistent disk type: 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 type: 50 GB
- Worker Availability Zone: az1

**Large:**
- Name: large
- Master/ETCD Node instances: 3
- Master/ETCD VM Type: medium.disk (CPU: 2, RAM: 4 GB, disk: 32 GB)
- Master persistent disk type: 30 GB
- Master/ETCD Availability Zone: az1
- Maximum number of workers on a cluster: 50
- Worker Node instances: 5
- Worker VM Type: 2xlarge.cpu (CPU: 16, RAM: 16 GB, disk: 64 GB)
- Worker persistent disk type: 50 GB
- Worker Availability Zone: az1
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. It forms two separate VMware clusters, one cluster for management and one cluster for compute.

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

SDP supports PKS version 1.5.1 and higher.

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.

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.

For the Ops Manager VM, allocate four vCPUs, 8 GB RAM, and two disks (64 GB and 75 GB), changing the default resources assigned by OVA.

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.
For the PKS tile VM, allocate four vCPUs, 8 GB or 16 GB RAM, and 30 GB to 50 GB of persistent disk at minimum.

### 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.

For the Bosh VM tile, allocate four vCPUs, 8 GB or 16 GB RAM, and 64 GB of persistent disk in the resource configuration chapter.

### 3.1.4 Harbor

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

For the Harbor VM tile, it is recommended to allocate default resources but increase the persistent disk to 150 GB to 300 GB.

### 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.

Streaming Data Platform requires the highest-performance disks that the solution hardware allows. It requires a complete SSD disk infrastructure with write-performance-oriented SSDs preferred. One improvement to consider is to use NVMe disks as vSAN cache disks for use cases where high streaming process performance is required. Traditional HDDs are not supported for Streaming Data Platform.

The best practice to configure a vSAN datastore with SSDs only is to create multiple disk groups where two disks are added to each disk group. One disk is used for cache, and one disk is used for capacity.

For example, if there are 10 SSD local disks in the ESXi node, create five disk groups, and add two disks in each of them. Repeat this step for all ESXi servers located in the vCenter cluster.

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).
- NVMe disks are highly recommended in some use cases.
- Use stripes in the vSAN default storage policy.
- Maximize the number of server and disks groups.
- 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.

![Diagram of vCenter distributed switch configuration](image)

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 `eaglemong-dvs`.

Port groups for vmnic0 and vmnic1 are used as uplink NICS:
- Management (Native VLAN)
- Storage (vSAN or VxFlex): VLAN 103
- vMotion: VLAN 102
- Overlay for edge VMs: VLAN 104
- Uplink2 for edge VMs: VLAN 105
  - This port group routes all PKS external traffic.
  - NSX-T runs in active/passive mode. Only one NIC at 10 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.

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 MTU is **1600**.
  - 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. For example: One compute cluster of 10 nodes can get 20 to 30 PKS Streaming Data Platform clusters.
  - 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 node.
• 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-Jumpserven: 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

- **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](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-1d60b196804' -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

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**: Computing nodes are separated from the edge nodes to the host vCenter and NSX-T VMs.
  - Advantages: The computing nodes are dedicated for PKS and streaming I/O, maintaining management I/O cycles outside.
  - Disadvantages: The quantity of nodes and switch ports are not ideal, and cabling is more complex than with the consolidated model.

- **Dell EMC VxRail™ consolidated model**: All ESXi hosts or nodes contain PKS and management VMs.
  - Advantages: This model offers simplicity. It requires lower numbers of nodes and switch ports, and requires less cabling.
  - Disadvantages: Each node requires increased in RAM capacity, and likely more powerful CPUs.

4.1.1 Traditional model
The traditional model of deployment includes the following nodes:

- Four compute nodes (runs Kubernetes within PKS cluster)
- Three edge nodes (NSX-T, vCenter)

Compute and edge nodes are running on ESXi version 6.7.0u3 or higher. Each node is built using a Dell EMC PowerEdge™ R640 server. See Table 2 and Figure 8 for more details.

Table 2 Traditional model: compute nodes

<table>
<thead>
<tr>
<th>Node type</th>
<th>Model</th>
<th>CPU</th>
<th>RAM</th>
<th>NICs</th>
<th>Disks</th>
</tr>
</thead>
<tbody>
<tr>
<td>Compute</td>
<td>PowerEdge R640</td>
<td>2 Intel® Xeon® Silver/Gold Processor, 12 cores, 24 threads Total of 48 vCPUs or more</td>
<td>256 GB DDR4-2400 or faster</td>
<td>4 x 10 GbE or 4 x 25 GbE SFP+ or SFP28 recommended</td>
<td>2 x 240 GB BOSS controller, M2 for boot disk in RAID 1 PERC H740P RAID controller Controller: 6 x 1.6 TB SSDs Write-oriented performance as minimum number (three groups or two disks) 1 or 2 NVMe disks optional as cache vSAN disks</td>
</tr>
</tbody>
</table>
Physical infrastructure

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

Table 3  Traditional model: edge nodes

<table>
<thead>
<tr>
<th>Node type</th>
<th>Model</th>
<th>CPU</th>
<th>RAM</th>
<th>NICs</th>
<th>Disks</th>
</tr>
</thead>
<tbody>
<tr>
<td>Edge</td>
<td>PowerEdge R640</td>
<td>1 Intel® Xeon® Silver Processor, 8 cores, 16 threads. Total of 16 vCPUs or more</td>
<td>128 GB</td>
<td>4 x 10 GbE or 4 x 25 GbE SFP+ or SFP28 recommended</td>
<td>2 x 480 GB SSDs Booting 1 and cache other 480 GB disk 3 x 960 GB SSDs or more for vSAN capacity</td>
</tr>
</tbody>
</table>

Figure 9  Traditional model: PowerEdge R640 edge nodes (front and back view)
4.1.2 Consolidated model

The consolidated model of deployment has four compute nodes (running Kubernetes within PKS cluster, NSX-T, vCenter) at a minimum.

Compute and edge nodes are running ESXi version 6.7.0u3 or higher. Each node is built using a PowerEdge R640 server.

Table 4 Consolidated model: compute nodes

<table>
<thead>
<tr>
<th>Node type</th>
<th>Model</th>
<th>CPU</th>
<th>RAM</th>
<th>NICs</th>
<th>Disks</th>
</tr>
</thead>
<tbody>
<tr>
<td>Compute</td>
<td>PowerEdge R640</td>
<td>2 x Intel® Xeon® Silver/Gold Processor, 16 cores, 32 threads</td>
<td>512 GB DDR4-2400 or faster</td>
<td>4 x 10 GbE or 4 x 25 GbE SFP+ or SFP28 recommended</td>
<td>2 x 240 GB BOSS controller M2 for boot disk in R1</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Total of 64 vCPUs or more</td>
<td></td>
<td></td>
<td>PERC H740P RAID controller</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Controller: 6 x 1.6 TB SSDs</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Write-oriented performance as minimum number (three groups or two disks)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1 or 2 NVMe disks optional as cache vSAN disks</td>
</tr>
</tbody>
</table>

Figure 10 Consolidated 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.

- Small consolidated environments with few servers with no growth expectation: 2 x PowerSwitch S5212-ON
- Traditional model or consolidated model with growth expectation: 2 x PowerSwitch S5224-ON
- Any model with more than 15 servers, now or in the future: 2 x PowerSwitch S5248-ON

![Dell EMC PowerSwitch S5200-ON series](image)

4.3 **Isilon tier 2 storage**

Streaming Data Platform requires an Isilon system with NFSv4/v3 as tier 2 storage 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.

---

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

A.1 Related resources

See the following additional resources:

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