



Spotfire® Data Virtualization

Installation Guide

Version 8.9.0 | April 2026

Contents

Contents	2
Preparing for Installation	7
Software Components for Installation	7
Overview of Installation	8
Tracking and Documenting Your Install or Upgrade	9
Installation Requirements and Support Information	10
Port Requirements	11
SDV Supported Platforms	13
SDV Server	14
SDV Business Directory	19
SDV Studio	21
SDV Drivers	23
Sizing Guidelines for SDV	28
Data Source Support	33
Supported Data Sources	33
Supported Add-On Adapters	40
Supported Advanced Data Source Adapters	40
Supported Cache Targets	45
Data Ship Source and Target Support	51
SDV DDL Feature Support	54
Data Sources Supported for Kerberos Token Pass-through	55
RHEL 9 - Required Locale Configuration for PostgreSQL (SDV Server)	56
Security Features	57
Support and Maintenance Policies for Spotfire Products	57
Limitations for SDV Discovery	59
Studio and Server Connectivity and Installer Limitations	59

Installing SDV Server, Studio, and Drivers	62
Overview of Installation Steps	62
Installation Overview for New SDV Software Customers	63
Installation Overview for Existing Customers Upgrading from a Previous Release	63
Preparing Your System for Installation	64
Preparing UNIX for SDV Installation	65
Preparing Microsoft Windows for SDV Installation	67
Installing on Windows	68
Running the SDV Server Installer	68
Running the Studio Installer	70
Installing the Drivers	71
Installing on UNIX	71
Installing SDV Server on UNIX	72
Installing Drivers on UNIX	74
Setting the SDV Server to Start Automatically on UNIX	75
Installing on Amazon Web Service	76
About SDV Software Patches	77
About the Installed SDV Services	78
Importing Metadata into the New SDV Instance	78
Tips from an Expert if the Server Does Not Start	78
Where to Go After Installation	79
Silent Mode Installation	81
SDV Docker Container	86
Prerequisites	86
Docker	88
SDV	89
Building SDV Docker Images	89
Using Quick Start Script	90
Using Docker Build	91
Publishing SDV Docker Images	92

Launching SDV Containers	93
Launching SDV Containers (Single Node) - Using Quick Start Script	94
Launching SDV Containers - Using Docker Compose	98
Runtime SDV Container Configuration - Common Examples	99
Change SDV Admin Password (while container is running)	100
Change SDV Base Port	100
Change SDV Server Memory Setting	102
Configure External Volume For Local Persisted File Data Sources	103
Configure Data Source With 3rd party JDBC Driver (type 4)	104
Configure Data Source With 3rd party JDBC Driver (type 3)	105
Configure JVM Settings from Docker	106
Configure LDAP Properties From Docker	108
Configure Truststore and Keystore Files From Docker	108
Limitations	109
Upgrading to a New Version of SDV Server	109
Useful Docker Commands for SDV Containers	110
SDV Container Orchestration Using Kubernetes	111
Introduction to Container Orchestration	111
SDV Container Orchestration Architecture	111
Prerequisites	113
Pre-Configuration of the Runtime Environment (On-Premises/Private Cloud only):	113
Building the Container Image	114
Configuring Kubernetes Using Quick Start Script	114
Deploying SDV	118
Installing the SDV HAProxy Service using the Quick Start Script	121
Removing a SDV application	123
Running SDV on a Public Cloud - Microsoft Azure	125
Creating a Container Registry	126
Creating a Cluster	127
Deploying the SDV Helm Chart	127

Running SDV on a Public Cloud - Amazon Web Service	129
Creating a Container Registry	130
Creating a Cluster	131
Deploying the SDV Helm Chart	132
Spotfire® Data Virtualization Container Distribution	134
Useful Kubernetes Commands	135
Generic Kubernetes Commands	136
Useful SDV-Specific Commands	137
SDV Deployment Examples	138
Installing Optional SDV Products	143
Version Support	143
Installation Requirements	143
Add-On Adapter Installation Requirements	143
Active Cluster Requirements	144
Installing an Optional SDV Product	145
Installing the Advanced Adapters	145
Auto Deployment	146
Manual Deployment	146
Installing the SDV Client Drivers that are Distributed with SDV	148
Importing Resources Defined in an Earlier Release	149
Manage Active Cluster Security	149
Updating the Digital Certificate to Secure Cluster Communication	150
Set Access Privileges	150
SDV and Business Directory Product Maintenance	151
About Hotfix Maintenance	151
Applying the Hotfix to SDV Server, Studio, and Business Directory	152
Downgrade/Rollback	153
About Service Pack Maintenance	159
Applying the Service Pack to SDV Server, Studio, and Business Directory	160
Applying the Service Pack or Hotfix to Active Cluster	161

Upgrading from an Earlier Release and Migrating The Metadata	162
Upgrade Path - Examples	167
Maintaining SDV-to-Client Application Connections	169
Updating an ODBC Client Application	169
Updating a JDBC Client Application	169
Uninstalling SDV	171
Uninstalling SDV on Windows	171
Uninstalling SDV on UNIX	171
Preparing for Uninstalling on UNIX	172
Uninstalling SDV On UNIX	172
SpotfireDocumentation and Support Services	173
Legal and Third-Party Notices	175

Preparing for Installation

Read the sections described in this chapter, prior to installing the Data Virtualization Platform:

- [Software Components for Installation](#)
- [Overview of Installation](#)
- [Tracking and Documenting Your Install or Upgrade](#)

Software Components for Installation

SDV provides the following installers for the Data Virtualization software components:

Installer	Included in the installer	
Server	Server Studio Deployment Manager Repository Java Monitor Discovery Web UI	Active Cluster Salesforce.com Adapter SAP Adapter SAPBW and BEx Adapters Oracle EBS Adapter Siebel Adapter Default cache database Advanced Data Sources Adapters
Studio	Studio	Java
Client	ODBC ADO.Net SSIS Power BI	JDBC
Business Directory Server	BD Server BD Repository	BD web application Java

Overview of Installation

Before you install Spotfire® Data Virtualization, review the new features and bug fixes as documented in the *SDV and Business Directory Release Notes*. You can view it at <https://docs.tibco.com/products/tibco-data-virtualization/>

Choose one of the options below:

1. For a new SDV installation, refer to the chapter [Installing SDV Server, Studio, and Drivers](#).

Note: After SDV 8.x is installed, you can proceed with installation of hotfix or service packs. Refer to the sections [About Hotfix Maintenance](#) and [About Service Pack Maintenance](#) for more information on how to do this.

2. If you are upgrading within the same major version (for example, 8.x to 8.(x+1)), refer to the section [About Service Pack Maintenance](#).
3. If you are upgrading from an earlier SDV major version (for example, 7.x to 8.x), and also want to migrate the metadata created in the earlier SDV version, refer to [Upgrading from an Earlier Release and Migrating The Metadata](#).

Note: If you do not need to migrate the metadata, follow the steps as outlined in the chapter [Installing SDV Server, Studio, and Drivers](#) to install a new version of SDV.

4. If you need to apply a hotfix, follow the steps as outlined in the section [About Hotfix Maintenance](#).

SMTP email settings

Ensuring correct email settings during the SDV installer prompts is crucial for the successful operation of the Deployment Report Trigger (located at `/lib/report/GenerateDeploymentReportTrigger`). The following email settings prompted during 8.9 SDV Server installation and are essential for optimal trigger functionality:

- **From Address** (required)
- **SMTP Host Name** (required)
- **SMTP Port** (optional; default: 25)

- **SMTP Authentication Required** (optional; default: false)
- **SMTP Authentication User Name** (optional)
- **SMTP Authentication User Password** (optional)

Tracking and Documenting Your Install or Upgrade

We recommend that you document the issues that you encounter during an upgrade and document all customizations made to your new SDV system, to help ensure that your next upgrade goes smoothly.

Before beginning your install, be aware that Java and PostgreSQL customizations are not preserved through the installation process. After install you must redo the customizations.

Installation Requirements and Support Information

This topic describes installation requirements and what Spotfire® Data Virtualization (SDV) supports. It includes the following topics:

- [Disk Space and Physical Memory Requirements](#)
- [Port Requirements](#)
- [Studio and Server Connectivity and Installer Limitations](#)
- [SDV Supported Platforms](#)
 - [SDV Server](#)
 - [SDV Business Directory](#)
 - [SDV Studio](#)
 - [SDV Drivers](#)
- [Data Source Support](#)
 - [Supported Data Sources](#)
 - [Supported Add-On Adapters](#)
 - [Supported Advanced Data Source Adapters](#)
 - [Supported Cache Targets](#)
 - [Data Ship Source and Target Support](#)
 - [Data Sources Supported for Kerberos Token Pass-through](#)
- [Security Features](#)
- [Support and Maintenance Policies for Spotfire Products](#)
- [Limitations for SDV Discovery](#)

Port Requirements

By default, SDV Server listens to port 9401 for ODBC connections. The ODBC port number is always one greater than the server's web services HTTP base port which by default, is 9400. So the ODBC default port number is 9401. If SSL is used (encrypt is set to true), the ODBC driver automatically adds 2 to the port value so that the 9403 port is used. To determine the actual ODBC port settings, refer to the Client Interfaces Guide, section "SDV Port Settings for Client Connections to SDV".

Changing the HTTP base port value also changes the value of all derived ports after the next SDV restart (with the exception of the Repository and Cache database ports, which will remain the same).

Port number availability for SDV and Business Directory:

SDV Server

SDV Ports Default	Description
9400	Web services HTTP port ← port needs to be exposed for non SSL SDV http access
9401	JDBC, ODBC, and ADO.NET ← port needs to be exposed for non SSL SDV client access
9402	Web services HTTP SSL ← port needs to be exposed for SSL SDV http access
9403	JDBC SSL, ODBC SSL, and ADO.NET SSL ← port needs to be exposed for SSL SDV client access
9404	Default caching database port
9405	JMX/RMI port for Monitor collector
9406	Monitor Daemon
9407	Active Cluster - JGroups (when enabled/when cluster is created)

SDV Ports Default	Description
9408	Repository
9409	Monitor RMI registry + JMX/RMI port for Monitor daemon

SDV Business Directory

Business Directory Ports Default	Description
9500	Web services HTTP port
9501	JDBC, ODBC, and ADO.NET
9502	Web services HTTP SSL
9503	JDBC SSL, ODBC SSL, and ADO.NET SSL
9504	[reserved]
9505	JMX/RMI port for Monitor collector
9506	Monitor Daemon
9507	[reserved]
9508	Repository
9509	Monitor RMI registry + JMX/RMI port for Monitor daemon

MPP Engine

SDV Ports Default	Description
9300	Zookeeper Quorum port
9301	Zookeeper Election port
9302	Zookeeper Client Port
9303	Drill HTTP (web console) Port
9304	Drill User Port
9305	Drill Server Bit Ports
9306	Drill Server Bit Ports

Note: The above ports are only active on operating systems that support MPP engine. (Refer to [MPP Engine OS Support](#))

SDV Supported Platforms

Studio can be installed and run on all Microsoft Windows platforms, but is not available for any UNIX platforms. Business Directory is supported on Windows and UNIX platforms only.

64-bit installers are provided for each of the Windows and UNIX platforms. In addition, separate java versions are provided for each platform Refer to the following topics for the support information for SDV components.

[SDV Server](#)

[SDV Business Directory](#)

[SDV Studio](#)

[SDV Drivers](#)

SDV Server

This page lists system requirements for Spotfire® Data Virtualization Server version 8.9.

Hardware

SDV performance depends on processor speeds, available memory, network bandwidth, data source response times, query join types, the complexity of views, and many other implementation factors. Fast response times and support for a large active user base and large loads are achieved with:

- Fast multi-core CPUs
- Large amounts of RAM
- Ample disk space
- GB Ethernet network connections on the same subnet as the most heavily trafficked data source

If hardware configurations are less than optimal, SDV functions equally well, although more slowly, for most development tasks. Given below are the minimum hardware requirements.

Processor	Minimum: 2 CPUs/cores Recommended: 8 CPUs/cores
RAM	Minimum: 8 GB memory Recommended: 32 GB memory
Hard Disk Space	Minimum: 8 GB read/write, persistent storage. • 2.0 GB for SDV installation • 1.0 GB for the SDV repository database • 5.0 GB for future growth (e.g. default SDV temp

	directory, SDV VCS data, logs, file cached data, and future SDV patch/Java/repository software updates) Recommended: 32 GB of readable/writable persistent storage
--	--

For details about different SDV deployments, please refer to [Sizing Guidelines for SDV](#).

Software

Operating System	architecture: PowerPC 64 • AIX 7.3 TL4 or higher • AIX 7.2 TL6 or higher Note: AIX is deprecated and will be removed in a future release. architecture: x86_64 • AlmaLinux v8.x (x>=10) • AlmaLinux v9.x (x >= 5) • Microsoft Windows 11 (Version 24H2 or higher) • Microsoft Windows Server 2022 Standard • Oracle Enterprise Linux 8.x or higher in Red Hat compatibility mode (x>=10) • Oracle Enterprise Linux 9.x in Red Hat compatibility mode (x >= 5) • Red Hat Enterprise Linux v8.x (x>=10) • Red Hat Enterprise Linux v9.x (x >= 5) • Rocky Linux v8.x (x>=10) • Rocky Linux v9.x (x >= 5) • SLES 15 (SP5 or higher)
SDV Server Database	• PostgreSQL 17.8 (Windows/Linux)

	• PostgreSQL 14.8 (AIX)
Java Support	• AIX - openjdk version "21.0.9" 2023-04-18 LTS • Linux - openjdk version "21.0.9" 2023-04-18 LTS • Windows - openjdk version "21.0.9" 2023-04-18 LTS
Network	IPv4
Authentication Methods	• Composite (Default) • NTLM • Kerberos (Not Supported in Business Directory) • OAuth (Not Supported in Deployment Manager and Business Directory)
Directory Services Support for LDAP	• Active Directory 2012 - LDAP, LDAPS, Kerberos • Azure Active Directory - LDAP, LDAPS • Novell eDirectory 9.2 - LDAP • Oracle Directory Server Enterprise Edition 11.1 - LDAP, LDAPS
Kerberos Support	Enterprise users can leverage Kerberos infrastructure to authenticate just once to secure access to SDV-defined resources. The duration of an authenticated session is set by the Kerberos administrator. SDV supports pass-through of the Kerberos tokens from the authenticated client through SDV to the Kerberos server and to the data sources. The Spotfire® Data Virtualization Server, data sources, and clients of the SDV Server must be configured to support Kerberos token pass-through and SSO. The KDC Kerberos v5 Server must already be installed and running in your environment before you install SDV Server and Studio. You then configure the Kerberos system to use

	with SDV, establishing a security context in which Kerberos and the SDV identify each other.
Directory Services Support for Kerberos	Active Directory 2019 - LDAP, LDAPS, Kerberos
Web Browser Support	Deployment Manager - • Mozilla Firefox 115.0.5790.102 on Windows 10. • Mozilla Firefox 115.0.5790.102 on macOS Monterey • Chrome 115.0.5790.102 on Windows 10. • Chrome 115.0.5790.102 on macOS Monterey. • Edge 115.0.1901.183 on Windows 10. • Safari is not supported. Web Manager - • Mozilla Firefox 115.0.5790.102 on Windows 10. • Mozilla Firefox 115.0.5790.102 on macOS Monterey • Chrome 115.0.5790.102 on Windows 10. • Chrome 115.0.5790.102 on macOS Monterey. • Edge 115.0.1901.183 on Windows 10. • Safari 16.5.2 on macOS Monterey Web UI - • Mozilla Firefox 115.0.5790.102 on Windows 10. • Mozilla Firefox 115.0.5790.102 on macOS Monterey • Chrome 115.0.5790.102 on Windows 10. • Chrome 115.0.5790.102 on macOS Monterey. • Edge 115.0.1901.183 on Windows 10. • Safari is not supported

	Note: Safari is not supported for web service API calls. Different browsers have different settings to enable Kerberos support. Spotfire recommends that you search the web to confirm the instructions to enable Kerberos SPNEGO authentication and credential delegation for your browser and operating system. For example in Firefox, add the url to both network.negotiate-auth.trusted-uris and network.negotiate-auth.delegation-uris and switch network.negotiate-auth.allow-non-fqdn to true.
Support for Virtualization Environments	The SDV Server is fully supported and can be hosted in operation systems run on virtualization platforms such as Red Hat Enterprise Virtualization, Hyper-V, VMware Fusion and Oracle VirtualBox as long as the Server is run on a supported OS platform and version that is listed above.
SNMP Support	The SDV system supports SNMP v3.
Web Service Protocols	• SOAP v1.1 • SOAP v1.2 • WSDL v1.1 • WSI-Basic Profile v1.0 • WSI-Basic Profile v1.1 • XPath v3.0 • XQuery v3.0 • XQuery v3.1 • XSLT v1.1 • XSLT v2.0
Enterprise Service Buses	• Sonic 7.5 • Spotfire EMS 4.4 • OpenMQ 4.4

SDV Business Directory

This page lists system requirements for Spotfire® Data Virtualization Server version 8.9.

Hardware

SDV performance depends on processor speeds, available memory, network bandwidth, data source response times, query join types, the complexity of views, and many other implementation factors. Fast response times and support for a large active user base and large loads are achieved with:

- Fast multi-core CPUs
- Large amounts of RAM
- Ample disk space
- GB Ethernet network connections on the same subnet as the most heavily trafficked data source

If hardware configurations are less than optimal, SDV functions equally well, although more slowly, for most development tasks. Given below are the minimum hardware requirements.

Processor	Minimum: 2 CPUs/cores Recommended: 8 CPUs/cores
RAM	Minimum: 8 GB memory Recommended: 32 GB memory
Hard Disk Space	Minimum: 8 GB read/write, persistent storage. • 2.0 GB for SDV installation • 1.0 GB for the SDV repository database • 5.0 GB for future growth (e.g. default SDV temp directory, SDV VCS data, logs, file cached data, and future SDV patch/Java/repository software updates) Recommended: 32 GB of readable/writable persistent storage

For details about different SDV deployments, please refer to [Sizing Guidelines for SDV](#).

Software

Operating System	• Microsoft Windows 11 (Version 24H2 or higher) • Microsoft Windows Server 2022 Standard • Oracle Enterprise Linux 8.x or higher in Red Hat compatibility mode (x>=10) • Oracle Enterprise Linux 9.x in Red Hat compatibility mode (x >= 5) • Red Hat Enterprise Linux v8.x (x>=10) • Red Hat Enterprise Linux v9.x (x >= 5) • Rocky Linux v8.x (x>=10) • Rocky Linux v9.x (x >= 5) • AlmaLinux v8.x (x>=10) • AlmaLinux v9.x (x >= 5) • SLES 15
SDV Server Database	PostgreSQL 17.8
Java Support	• Linux - java version "21.0.9" 2023-04-18 LTS • Windows - java version "21.0.9" 2023-04-18 LTS
Network	IPv4
Authentication Methods	Composite (Default) NTLM
Support for Virtualization Environments	The SDV Server is fully supported and can be hosted in operation systems run on virtualization platforms such as Red Hat Enterprise Virtualization, Hyper-V, VMware Fusion and Oracle VirtualBox as long as the Server is run on a supported OS platform and version that is listed above.

Web Browser Support	• Mozilla Firefox 115.0.5790.102 on Windows 10. • Mozilla Firefox 115.0.5790.102 on macOS Monterey • Chrome 115.0.5790.102 on Windows 10. • Chrome 115.0.5790.102 on macOS Monterey. • Edge 115.0.1901.183 on Windows 10. • Safari 16.5.2 on macOS Monterey Note: Different browsers have different settings to enable Kerberos support. Spotfire recommends that you search the web to confirm the instructions to enable Kerberos SPNEGO authentication and credential delegation for your browser and operating system. For example in Firefox, add the url to both <code>network.negotiate-auth.trusted-uris</code> and <code>network.negotiate-auth.delegation-uris</code> and switch <code>network.negotiate-auth.allow-non-fqdn</code> to true.
---------------------	--

SDV Studio

This page lists system requirements for Spotfire® Data Virtualization Server version 8.9.

Hardware

SDV performance depends on processor speeds, available memory, network bandwidth, data source response times, query join types, the complexity of views, and many other implementation factors. Fast response times and support for a large active user base and large loads are achieved with:

- Fast multi-core CPUs
- Large amounts of RAM
- Ample disk space
- GB Ethernet network connections on the same subnet as the most heavily trafficked data source

If hardware configurations are less than optimal, SDV functions equally well, although more slowly, for most development tasks. Given below are the minimum hardware requirements.

Processor	Minimum: 2 CPUs/cores Recommended: 8 CPUs/cores
RAM	Minimum: 8 GB memory Recommended: 32 GB memory
Hard Disk Space	Minimum: 8 GB read/write, persistent storage. • 2.0 GB for SDV installation • 1.0 GB for the SDV repository database • 5.0 GB for future growth (e.g. default SDV temp directory, SDV VCS data, logs, file cached data, and future SDV patch/Java/repository software updates) Recommended: 32 GB of readable/writable persistent storage

For details about different SDV deployments, please refer to [Sizing Guidelines for SDV](#).

Software

Operating System	• Microsoft Windows 11 (Version 24H2 or higher) • Microsoft Windows Server 2022 Standard
SDV Server Software	An older version of SDV Studio can connect to a newer version of SDV Server. For example: • Connecting a 8.6.0 Studio with a 8.7.0 or 8.8.0 Server is supported. • Connecting a 8.8.0 Studio with a 8.5.0 Server is not supported.

Java Support	Java version '21.0.9' 2023-04-18 LTS
Support for Virtualization Environments	The SDV Server is fully supported and can be hosted in operation systems run on virtualization platforms such as Red Hat Enterprise Virtualization, Hyper-V, VMware Fusion and Oracle VirtualBox as long as the Server is run on a supported OS platform and version that is listed above.
Network	IPv4
Authentication Methods	• Composite (Default) • NTLM • Kerberos • OAuth
Kerberos Support	Enterprise users can leverage Kerberos infrastructure to authenticate just once to secure access to SDV-defined resources. The duration of an authenticated session is set by the Kerberos administrator. SDV supports pass-through of the Kerberos tokens from the authenticated client through SDV to the Kerberos server and to the data sources. The Spotfire® Data Virtualization Server, data sources, and clients of the SDV Server must be configured to support Kerberos token pass-through and SSO. The KDC Kerberos v5 Server must already be installed and running in your environment before you install SDV Server and Studio. You then configure the Kerberos system to use with SDV, establishing a security context in which Kerberos and the SDV identify each other.

SDV Drivers

This page lists system requirements for Spotfire® Data Virtualization Server version 8.9.

Hardware

SDV performance depends on processor speeds, available memory, network bandwidth, data source response times, query join types, the complexity of views, and many other implementation factors. Fast response times and support for a large active user base and large loads are achieved with:

- Fast multi-core CPUs
- Large amounts of RAM
- Ample disk space
- GB Ethernet network connections on the same subnet as the most heavily trafficked data source

If hardware configurations are less than optimal, SDV functions equally well, although more slowly, for most development tasks. Given below are the minimum hardware requirements.

Processor	Minimum: 2 CPUs/cores Recommended: 8 CPUs/cores
RAM	Minimum: 8 GB memory Recommended: 32 GB memory
Hard Disk Space	Minimum: 8 GB read/write, persistent storage. • 2.0 GB for SDV installation • 1.0 GB for the SDV repository database • 5.0 GB for future growth (e.g. default SDV temp directory, SDV VCS data, logs, file cached data, and future SDV patch/Java/repository software updates) Recommended: 32 GB of readable/writable persistent storage

For details about different SDV deployments, please refer to [Sizing Guidelines for SDV](#).

SDV Server Software

Refer to [SDV Server](#) for the SDV Server Requirements.

Drivers

ODBC	• iODBC Driver Manager v3.52.12 for AIX and Linux • DataDirect Driver Manager v8.0 - DataDirect Driver Manager is supported on Linux and AIX platforms. There is no extra configuration needed for using SDV ODBC driver with DataDirect Driver Manager. • unixODBC Driver Manager(v2.3.1) - unixODBC Driver Manager is supported on Linux and AIX platforms. For using SDV ODBC driver with the unixODBC Driver Manager, set the environment variable SDV_ODBC_UODBC to TRUE for the applications that use SDV ODBC Driver. <pre>set SDV_ODBC_UODBC=TRUE</pre> • Windows Driver Manager - The Windows Driver Manager is part of your Windows Operating System. Refer to the section <i>Operating System Support for Server</i> in the <i>SDV Installation and Upgrade Guide</i> for a list of supported OS.
JDBC	• JRE v11 and conforms to JDBC API 4.0 • JRE v1.8 and conforms to JDBC API 4.0 • JRE v17 and conforms to JDBC API 4.0 Example: If you are running a client using JRE 8(also known as 1.8), you would include csjdbc.jar in the CLASSPATH as shown below: <pre>"C:\Program Files\Java\jdk1.8.0_211\bin\java" -classpath .;.\csjdbc.jar Test</pre>
ADO .Net	ADO .Net Driver Support • ADO.NET (32-bit and 64-bit) Client-Side ADO.NET Driver Support The SDV ADO.NET driver can be installed, uninstalled, or re-

	installed. It is delivered as a DLL and .NuGet package. It supports native ADO.NET driver functionality on the operating systems supported by .Net Standard 2.0. Refer the following link for a list of OS and .Net versions supported by .Net Standard 2.0: https://dotnet.microsoft.com/en-us/platform/dotnet-standard SDV supports communication and use with Visual Studio version 2019 that the tool extension supports. You will need to install the extension by executing the following file: <pre>SDV_INSTALL_DIR\ADO.NET Driver\Composite.VisualStudio.vsix</pre> The SKUs supported by the Visual Studio Extension are Community, Professional and Enterprise. The Extension essentially has a Provider for DDEX(Data Designer Extensibility). Refer the following link for further details: https://docs.microsoft.com/en-us/previous-versions/bb163760(v=vs.140) Client Application Target Framework .NET Standard 2.0 Refer to the following link for a list of .Net Framework versions supported by .Net Standard 2.0: https://dotnet.microsoft.com/en-us/platform/dotnet-standard
ADO.Net Data Provider with Entity Framework support	ADO.Net 2022 Data Provider - v25.0.9477
Power BI	The Power BI Connector for Spotfire(R) Data Virtualization is built on top of an ODBC driver. SDV supports version 25.0.9403 SDV Installation prompts you to install Microsoft Visual C++

	Redistributable for Visual Studio 2017, if it is not already installed. Proceed with the installation. This is needed to connect to the Powe BI Connector.
SSIS	The Spotfire SSIS Components for Spotfire(R) Data Virtualization supports Visual Studio versions 2012 and above. SDV Supports SSIS version 25.0.9403
Authentication Methods	• Composite (Default) • NTLM • Kerberos • OAuth
Client Applications	JDBC Driver: • Cognos 11 R3 • Cognos v10.2.2 fixpack 5 ODBC Driver: • MicroStrategy 9.0.2 - Data sources supported are : Oracle 10g or 11g, Netezza 5 or 6, SQL Server 2008, and for mixed data coming from Oracle 11g and Netezza 6. • MicroStrategy 9.2.1/9.2.1m on Windows I-Server - Because MicroStrategy can create and delete data directly, you must have used Studio and configured one of the following as a temporary tablespace to hold the created and deleted data: Oracle 10g and 11g, Netezza 5 and 6, SQL Server 2008, Teradata 13, MySQL 5, and DB2 v9. • Tableau Desktop Professional Edition Version 7.0.13 ADO.Net Driver: • Spotfire Spotfire Note: All other client applications are supported through the standard communication protocols that include JDBC and ODBC.

Sizing Guidelines for SDV

For your deployment, use the following SDV sizing guidelines. Keep in mind that these are starting point recommendations for each configuration. It is OK to exceed these recommendations for a given configuration.

On-Premises	Small configuration (poc/demo) 2 CPUs/cores, 8 GB of memory, 8 GB of readable/writable persistent storage Medium configuration (development) 4 CPUs/cores, 16 GB of memory, 16 GB of readable/writable persistent storage Large configuration (production) 8 CPUs/cores, 32 GB of memory, 32 GB of readable/writable persistent storage X-Large configuration (production with MPP Engine) 16 CPUs/cores, 64 GB of memory, 64 GB of readable/writable persistent storage Note: For x-large SDV instances running MPPE, the minimum requirement calls for 2 clustered 8-core instances.
Docker/OCI Compliant runtime	Small configuration (poc/demo) • SDV - 2 CPUs/cores, 8 GB of memory, 8 GB of readable/writable persistent storage • SDV Repository - 1 CPU/core, 2 GB of memory, 2 GB of readable/writeable persistent storage • SDV Cache- 1 CPU/core, 2 GB of memory, 2 GB of readable/writeable persistent storage Medium configuration (development) • SDV - 4 CPUs/cores, 16 GB of memory, 8 GB of readable/writable persistent storage

	• SDV Repository - 1 CPU/core, 2 GB of memory, 4 GB of readable/writeable persistent storage • SDV Cache- 1 CPU/core, 2 GB of memory, 4 GB of readable/writeable persistent storage Large configuration (production) • SDV - 8 CPUs/cores, 32 GB of memory, 16 GB of readable/writable persistent storage • SDV Repository - 1 CPU/core, 4 GB of memory, 8 GB of readable/writeable persistent storage • SDV Cache- 1 CPU/core, 4 GB of memory, 8 GB of readable/writeable persistent storage X-Large configuration (production with MPP Engine) • SDV - 16 CPUs/cores, 64 GB of memory, 32 GB of readable/writable persistent storage • SDV Repository - 1 CPU/core, 8 GB of memory, 16 GB of readable/writeable persistent storage • SDV Cache- 1 CPU/core, 8 GB of memory, 16 GB of readable/writeable persistent storage Note: For X-Large SDV instances running MPPE, the minimum requirement calls for 2 clustered 8-core instances.
Public Cloud	AWS and Azure provide a wide selection of instances optimized to fit different use cases. For SDV deployments (i.e. Docker and Kubernetes) in this environment, the minimum recommendation is to use the Medium resource configuration from the <i>Docker/OCI Compliant runtime</i> section. If your organizational and/or resource needs require more then we recommend looking at either the Large or X-Large configurations.

	Refer to the following links for the public cloud platform guidelines: https://aws.amazon.com/ec2/instance-types/ https://docs.microsoft.com/en-us/azure/cloud-services/cloud-services-sizes-specs
Kubernetes	The SDV Kubernetes distribution is packaged with example helm charts for various environments (e.g. on-premises, AWS, and Azure). For SDV deployments (i.e. Kubernetes) in this environment, the minimum recommendation for each SDV pod is the Medium resource configuration from the Docker section. Based on that, your Kubernetes worker nodes need more resources than that in order to comfortably run that SDV pod. Example: Deploy 3 SDV node cluster instances in a Kubernetes cluster which has 3 worker nodes. Each Kubernetes worker node is provisioned with: 5 cpu, 16 GB memory, and 22 GB of free storage. Kubernetes worker node (1 cpu, 4 GB memory, 10 GB of storage) SDV server pod (2 cpu, 8 GB memory, 8 GB of storage) SDV repo pod (1 cpu, 2 GB memory, 2 GB of storage) SDV cache pod (1 cpu, 2 GB memory, 2 GB of storage)

Tips From an Expert for Memory Capacity Planning

Following are some of the considerations when doing memory capacity planning. Note that these are only suggestions and the actual capacity planning depends on several factors including the query load, size of the result sets, etc.

- The section [Sizing Guidelines for SDV](#) will help you plan the CPU recommendations that you will need for your specific requirement. You need to monitor the CPU to make sure that you have sufficient capacity to handle the data.
- Allocate only up to a maximum of 80-85% of the available memory on the machine to the SDV Server, while no other application than SDV is running on the machine. Make sure to leave 15-20% of the available memory for other OS processes.
- Identifying the kind of query you are running and which engine it might fall into (Classic Query Engine or Massively Parallel Processing Engine) will help you allocate memory accordingly. The Classic Query Engine is used for operational queries where the data is retrieved for a particular set of entities and the MPP Engine is ideal for federated queries used for analytical purposes. You can tune the different memory configuration settings, based on which type of engine will be used for your query. Refer to the table given below for a list of the memory settings and how you can tune them.
- If you notice performance issues, it is advised to enable GC logging. These logs will help get details about the Garbage collection process running for the SDV Server. Knowing this will help you assess the memory profile and evaluate the processes that might be slowing the performance.
- The SDV installation provides a startup script for UNIX and WINDOWS that you can customize for your own purpose. You can use this script to enable Garbage collection(GC) logging for SDV. Refer to the sections *Customizing the SDV Server Startup Scripts on UNIX/WINDOWS* in the *Administration Guide* for instructions on how to do this.
- You can also use third party performance monitoring tools such as dynatrace, datadog, etc. to proactively manage your SDV implementation.
- SDV provides some configuration settings that can be tuned to assist the performance and efficiency of SDV. Some useful settings are given below. You can search for all *Memory* settings by using the Search option in the Configuration window which can be accessed by the menu item Administration->Configuration:

Configuration Parameter	Description
Total System Memory	The total physical memory calculated on the target installation machine. This can be set and tuned externally and cannot be changed within SDV.

Configuration Parameter	Description
Classic Engine:	
Total Available Memory	The total amount of Java heap (memory) that is available for the server's use. The minimum value is 50 MB. This is the Java heap memory used by the Classic Query Engine. Changing this value will have no effect until the next server restart. This value is locally defined. It will not be altered when restoring a backup and will not be replicated in a cluster.
Percent of Memory to Manage	This value is the percentage of the memory available for query processing (heap minus unmanaged/reserved memory). It is set as a percentage to accommodate the uncertainties associated with dynamic memory usage and a factor of safety. For example, if the Java VM heap size is 700 MBytes and the reserved memory is 60 MBytes and this value is 70%, the amount of memory available for query processing is 448 MBytes of memory. If the managed memory setting value is too large, then the server may experience out of memory errors. This value is locally defined. It will not be altered when restoring a backup and will not be replicated in a cluster.
Maximum Memory per Request	This value is the percentage of managed memory that a single request is limited to during execution. This can be used to prevent excessive memory use by a single request. This value is locally defined. It will not be altered when restoring a backup and will not be replicated in a cluster
MPP Engine:	
Maximum Heap Memory	This is a setting used for the MPP Engine. It is the upper limit of the heap memory allocated for parallel processing. For this value to take effect you will need to

Configuration Parameter	Description
	disable and enable parallel processing. When the JVM starts up, the heap memory is created and used by the application. Default value will be 10% to total system memory, and the value set must be greater than 0.
Maximum Direct Memory	This is the upper limit of the direct memory allocated for parallel processing. For this value to take effect you will need to disable and enable parallel processing. Default value will be 50% to total system memory, and the value set must be greater than 0.
Resource Quota per Request	This value is the percentage of system resources that each request is limited to during parallel execution. Setting this value to 100 would allow only 1 query at a time to run in the MPP Engine, using a value of 50 would allow 2 concurrent queries to be served by the MPP Engine and so on and so forth.

Data Source Support

The following topics lists all the Data sources supported by SDV.

- [Supported Data Sources](#)
- [Supported Add-On Adapters](#)
- [Supported Cache Targets](#)
- [Data Ship Source and Target Support](#)
- [SDV DDL Feature Support](#)
- [Data Sources Supported for Kerberos Token Pass-through](#)

Supported Data Sources

SDV supports these data sources.

SDV supports OAuth 2.0 for HTTP-based data sources: SOAP, REST, WSDL, and XML-HTTP. It is also available for several Advanced Adapter data sources.

For other supported data sources and applications, see these sections:

- [Supported Add-On Adapters](#)
- [Supported Advanced Data Source Adapters](#)

Refer to the Adapter Guides for more details about each of the adapters.

Data Source	SDV Support	Versions, Compatibility, and Notes
Apache Drill	Active	CAST AS functions are not supported.
Amazon AWS S3	Active	
Azure Data Lake Storage	Active	
Composite	Active	
ComputeDB	Active	Version 1.2
Custom Java Procedure	Active	
DB2 V10.5 (Type4)	Active	
DB2 z/OS Version 10 (Type 4)	Active	
DB2 z/OS Version 11	Active	
Data Direct Mainframe	Active	The Shadow RTE Server (version 6.1.4.7606 or later) must be installed on the DataDirect Mainframe computer and the Shadow RTE Client (version 6.1.1.1080 or later) must be installed locally on the computer hosting the SDV Server.
File (cache, delimited, and XML)	Active	For data sources that access a file share, the SDV service user account needs to have

Data Source	SDV Support	Versions, Compatibility, and Notes
		permission to read the file share.
Greenplum 3.3	Active	SDV capabilities with Greenplum have been developed and tested with a single node license.
Greenplum 4.1	Active	
Greenplum 4.3	Active	
HBase 0.98 (Apache Phoenix Driver)	Active	Introspection of HBase databases retrieves information from the system tables. User created tables are only introspected if they have been created using the Apache Phoenix shell. Requires installation of Apache Phoenix JDBC drivers, specifically those in phoenix-4.1.0-bin.tar.gz. For more information see the SDV Administration Guide.
HSQLDB 2.2.9	Active	
Hive 2.x	Active	Kerberos is supported. Trusted Delegation is not supported. For Hive data sources, SDV introspects tables and columns only.
Impala	Active	Supported versions are - Apache Impala 2.x (Connect using Hive jar files) and Cloudera Impala 2.7.
Informix 9.x	Active	
Local File Storage	Active	
LDAP	Active	v3

Data Source	SDV Support	Versions, Compatibility, and Notes
Microsoft Access	Active	Windows platforms only
Microsoft Access (non-ODBC)	Active	Windows platforms only
Microsoft Excel	Active	2000
Microsoft Excel (non-ODBC)	Active	2000
Microsoft SQL Server 2008	Active	Kerberos authentication is supported.
Microsoft SQL Server 2012	Active	Kerberos authentication is supported with the 2008 driver.
Microsoft SQL Server 2014	Active	Kerberos authentication is supported.
Microsoft SQL Server 2016	Active	Kerberos authentication is supported.
Microsoft SQL Server 2019	Active	Kerberos authentication is supported.
MySQL 5.1	Active	
MySQL 5.5	Active	
MySQL 8.0.23	Active	Use MySQL5.5 adapter to connect to Mysql8.0.X databases.
Neoview 2.3	Active	
Neoview 2.4	Active	

Data Source	SDV Support	Versions, Compatibility, and Notes
Netezza 6.0	Active	NPS
Netezza 7.0	Active	NPS
OData	Active	Provides for some limited access to SharePoint data.
Oracle 11g (OCI Driver)	Active	11g R1, 11g R2, Oracle RAC Kerberos authentication is supported with thin driver version 11.2.0.4.
Oracle 11g (Thin Driver)	Active	11g R1, 11g R2, Oracle RAC Kerberos authentication is supported with thin driver version 11.2.0.4.
Oracle 12c (OCI Driver)	Active	Oracle RAC
Oracle 12c (Thin Driver)	Active	Oracle RAC
Oracle 19c (OCI Driver)	Active	
Oracle 19c (Thin Driver)	Active	
PostgreSQL 12	Active	
PostgreSQL 9.0	Active	
PostgreSQL 9.1	Active	
PostgreSQL 9.2.3	Active	
REST	Active	Kerberos authentication is supported. NTLM authentication is supported. OAuth 2.0 authentication is supported.

Data Source	SDV Support	Versions, Compatibility, and Notes
RSS	Active	
Redshift	Read-only	New datasources cannot be created. The following trigonometric functions should not be pushed to Redshift data sources, because they might return incorrect results: SIN, COS, TAN, ASIN, ACOS, COT.
SOAP	Active	1.1, 1.2 Kerberos authentication is supported. NTLM authentication is supported. OAuth 2.0 authentication supported.
SAP HANA SPS 09	Active	Support is for on premise SAP HANA deployments.
Sybase 12	Active	12.5 ASE
Sybase 15	Active	15 and 15.5 ASE Kerberos authentication is supported.
Sybase 16		
Sybase IQ	Active	15
Sybase IQ (Type 2)	Active	15.2
Teradata 13	Active	13 and 13.10 Support for query band.
Teradata 14	Active	14.10 Might require installation of a Teradata 15

Data Source	SDV Support	Versions, Compatibility, and Notes
		driver. Support for query band.
Teradata 15	Active	FastExport is not supported. The JDBC driver does not support CLOB columns with NULL values when using SDV to cache data into a Teradata 15 target. Support for query band.
Teradata 16	Active	16.20 The JDBC driver does not support CLOB columns with NULL values when using SDV to cache data into a Teradata 16 target. Support for query band.
Spotfire Streaming	Active	11.0.0
Vertica 6.1	Active	
WSDL	Active	1.1 Kerberos authentication is supported. NTLM authentication is supported. OAuth 2.0 authentication supported.
XML/HTTP	Active	Flat files or over HTTP. Kerberos authentication is supported. NTLM authentication is supported. OAuth 2.0 authentication supported.

Note: ComputeDB is deprecated and will be removed in a future release.

Supported Add-On Adapters

Consult your vendor specific documentation for detailed documentation of the objects and fields that have changed from version to version. These SAP adapters require the SAP JCo driver. Configuration steps can be found in the SDV SAP BW Adapter Guide. OLAP Cube Support—With SDV 5.1.0.1 and later, you can create dimensional OLAP views in SDV.

SDV supports the following application data sources.

Adapter	Version Support
Oracle E-Business Suite Adapter	11.5.8, 11.5.10 , 12.1, 12.2 on Oracle 9i and 10g
Salesforce.com Adapter	Version 37 You can install and use the Salesforce.com Adapter on all platforms that SDV supports. See Installation Requirements and Support Information .
SAP Adapter	5.0, 6.0, and above SAP R/3 v4.7
SAP BW Adapter	3.5 and 7.4 SP 9
SAP BW BEx Adapter	3.5 and 7.4 SP 9
Siebel Adapter	7.7, 7.8, 8.0

For installation and licensing instructions, consult [Installing Optional SDV Products](#).

Supported Advanced Data Source Adapters

SDV supports the following application data sources.

Data Source Adapter	Versions, Compatibility, and Notes
Active Directory	LDAP v2 and v3 servers
Amazon DynamoDB	DynamoDB REST API Version 2012-08-10
Amazon Redshift	Version 1.0.7562
Cassandra	Versions 2.1.7 and 3.0.0
CosmosDB	2019
Couchbase	Version 4.0 of the API
DynamicsCRM	Windows server 2016, Windows server 2012 R2, Windows Server 2012. Supports OAuth.
DynamicsGP	Dynamics GP 2010, 2013, and 2015
DynamicsNAV	Dynamics NAV 2013, 2015, and 2016
Eloqua	Eloqua REST API and Bulk API version 2.0 Supports OAuth.
Elasticsearch	Version 2.0 and above
Email	Standard IMAP client as specified in RFC 1730 and RFC 2060
Sharepoint Excel Services	Excel data from Windows SharePoint Services 3.0, SharePoint Server 2007+ and SharePoint Online.
Facebook	Facebook Graph API 2.0, 2.1, 2.2, 2.3 Supports OAuth.

Data Source Adapter	Versions, Compatibility, and Notes
Google Ads	API v201809 Supports OAuth. Note: Google AdWords has been re-branded as GoogleAds. Google AdWords datasource created in SDV version 8.0 or earlier cannot be opened in SDV version 8.1 or later. You need to create a new datasource by choosing GoogleAds from the new datasource dialogue in SDV Studio and introspect it again.
Google Analytics	Google Analytics Management API v3.0, Google Analytics Core Reporting API v3.0. Supports OAuth.
Google Contacts	API v3.0 Supports OAuth.
Google Calendar	API v3.0 Supports OAuth.
Google Drive	API V3.0 Supports OAuth.
Google BigQuery	Google BigQuery API v2.0 Supports OAuth.
Google Sheets	Google Sheets API v3.0 Supports OAuth.
HubSpot	HubSpot REST API Supports OAuth.
JDBC-ODBC Bridge	ODBC 2.x and 3.x drivers

Data Source Adapter	Versions, Compatibility, and Notes
JSON	Standard JSON format as specified in RFC 7519
MarkLogic	2019
Marketo	Marketo REST API v1, Marketo SOAP API v2.6 Supports OAuth.
MongoDB	MongoDB 2.6 and 3.0
NetSuite	NetSuite SOAP APIs 2011-2015 Supports OAuth.
OData	OData 2.0, 3.0, and 4.0 Supports OAuth.
RSS	RSS 2.0 feeds
Salesforce with SSO	Supports OAuth.
SharePoint	Excel data from Windows SharePoint Services 3.0, SharePoint Server 2007+ and SharePoint Online.
Single Store	21.0.7810.0
Snowflake	Version 3.25.5. Supports OAuth.
SparkSQL	Version 1.0 and above
Splunk	2019
Twitter	Twitter REST API v1.1. Supports OAuth.

Limitations:

- Sometimes, instead of returning an empty value, “Select * from table where columnname = 'value'” may throw an exception, if there is no value in the column.
- Some adapters support ORDER BY, but sometimes there are only a few objects within that data source that support ORDER BY. SDV displays a message if the tables do not support ORDER BY.
- Tables might need to be filtered with mandatory inputs for querying the contents for table scans to work as expected. For example for google apps directions, the starting location and ending location might be needed to retrieve the results.
- Sharepoint adapters support direct Kerberos authentication.
- Bulk inserts are not supported.
- GoogleSheets does not support client side filtering.
- Table names or column names with the period character are not supported.
- Eloqua data sources where the password value is entered when creating the data source will persist the password and it cannot be changed.
- For the Sharepoint Excel Services adapter, during introspection all String data types are mapped to VARCHAR.
- The DynamicsCRM, DynamicsNAV, DynanoDB, GoogleBigQuery, and SharePoint adapters do not support "is not null" syntax.
- Queries that contain "LIMIT" are not supported.
- For MongoDB, updating schema files within a running instance of SDV is not supported.
- Deployment Manager is case sensitive when using it with these adapters.
- Deployment Manager attributes for these adapters can cause plans to fail.
- “Ignore case sensitivity mismatch between CIS and data source” and “Ignore trailing space mismatch between CIS and data source” override the server side setting for a data source.
By default these two overrides are enabled so that queries are always pushed. This is the case even when there is a mismatch and the query does not contain UPPER or RTRIM or similar options.
- Set these attributes to false or disable the push to get the consistent results as when the query is run with in the SDV.

For installation and licensing instructions, consult *Installing Optional SDV Products*, page 49.

For OAuth descriptions, see the *User Guide*, section *Configuring OAuth 2.0 for SDV Advanced Adapters*.

Supported Cache Targets

SDV supports the following as cache targets:

Cache Target	SDV Support	Parallel Cache Target Support	Native Cache Target Support	Notes
Amazon Redshift	Active	Active	Active	
Apache Hive 2.x	Active		Active	
Apache Impala 2.x	Active		Active	
File	Active	Active		Typically best for demonstrations or caching of a few hundred rows.
Google BigQuery	Active	Active	Active	
Greenplum 4.1	Active	Active	Active	
Greenplum 4.3	Active	Active	Active	
HSQLDB 2.2.9	Active	Active		
IBM DB2 LUW v10.5	Active	Active	Active	Native load with insert and select, and DB2

Cache Target	SDV Support	Parallel Cache Target Support	Native Cache Target Support	Notes
				Load are supported.
Microsoft SQL Server 2008	Active	Active	Active	The DBO schema must be selected and introspected as a resource prior to attempting to cache data.
Microsoft SQL Server 2012	Active	Active	Active	The DBO schema must be selected and introspected as a resource prior to attempting to cache data.
Microsoft SQL Server 2014	Active	Active	Active	The DBO schema must be selected and introspected as a resource prior to attempting to cache data.
Microsoft SQL Server 2016	Active	Active	Active	The DBO schema must be selected and introspected as a resource prior to attempting to cache data.
Microsoft SQL Server 2019	Active	Active	Active	The DBO schema must be selected and introspected as a resource prior to

Cache Target	SDV Support	Parallel Cache Target Support	Native Cache Target Support	Notes
				attempting to cache data.
MySQL 5.1	Active	Active	Active	
MySQL 5.5	Active	Active	Active	
Netezza 6.0	Active	Active	Active	Native load with insert and select is supported. Parallel cache processing is achieved using the native DISTRIBUTE syntax. Procedure caching is supported.
Netezza 7.0	Active	Active	Active	Native load with insert and select is supported. Parallel cache processing is achieved using the native DISTRIBUTE syntax. Procedure caching is supported.
Oracle 11g and 11g R2	Active	Active	Active	
Oracle 12c	Active	Active	Active	

Cache Target	SDV Support	Parallel Cache Target Support	Native Cache Target Support	Notes
Oracle 19c	Active	Active	Active	
PostgreSQL 9.1	Active	Active	Active	Bulk load is supported. Native loading is supported when the source and target are the same database. If not then Parallel loading is used.
PostgreSQL 9.2.3	Active	Active	Active	Bulk load is supported. Native loading is supported when the source and target are the same database. If not then Parallel loading is used.
SAP HANA SPS 09	Active	Active		
Singlestore	Active	Active	Active	
Snowflake	Active	Active	Active	
Sybase ASE 12.5	Active			
Sybase ASE 15.5	Active			

Cache Target	SDV Support	Parallel Cache Target Support	Native Cache Target Support	Notes
Sybase IQ 15.2	Active		Active	
Teradata 13	Active		Active	Supported, but with limitations. If source and target tables are co-located within the same Teradata instance native loading (using INSERT/SELECT statements) will be used, else bulk loading using Teradata FASTLOAD will be attempted.
Teradata 13.10	Active		Active	Supported, but with limitations. If source and target tables are co-located within the same Teradata instance native loading (using INSERT/SELECT statements) will be used, else bulk loading using Teradata FASTLOAD will be attempted.
Teradata 14.10	Active		Active	Supported, but with limitations. Might require Teradata 15

Cache Target	SDV Support	Parallel Cache Target Support	Native Cache Target Support	Notes
				driver. If source and target tables are co-located within the same Teradata instance native loading (using INSERT/SELECT statements) will be used, else bulk loading using Teradata FASTLOAD will be attempted.
Teradata 15	Active		Active	Choose tables For Caching is not supported. If source and target tables are co-located within the same Teradata instance native loading (using INSERT/SELECT statements) will be used, else bulk loading using Teradata FASTLOAD will be attempted.
Teradata 16.20	Active		Active	Choose tables For Caching is not supported. If source and target

Cache Target	SDV Support	Parallel Cache Target Support	Native Cache Target Support	Notes
				tables are co-located within the same Teradata instance native loading (using INSERT/SELECT statements) will be used, else bulk loading using Teradata FASTLOAD will be attempted.
ComputeDB	Active	Active	Active	
Vertica 6.1	Active	Active	Active	Supports the use of native load and parallel cache load together. Native load with INSERT AND SELECT is supported.

Data Ship Source and Target Support

Data ship optimization is supported for following data source types.

Data Source Type	Data Ship Source Support	Data Ship Target Support	Performance Option	Notes
DB2 v10.5	Active	Active	Bulk Load using the LOAD utility	LUW

Data Source Type	Data Ship Source Support	Data Ship Target Support	Performance Option	Notes
Greenplum 3.3	Active	Active		
Greenplum 4.1	Active	Active		
Greenplum 4.3	Active	Active		
Microsoft SQL Server 2008	Active	Active	Bulk import/export using BCP	
Microsoft SQL Server 2012	Active	Active	Bulk import/export using BCP	
Microsoft SQL Server 2014	Active	Active	Bulk import/export using BCP	
Microsoft SQL Server 2016	Active	Active	Bulk import/export using BCP	
Microsoft SQL Server 2019	Active	Active	Bulk import/export using BCP	
Netezza 6.0	Active	Active	external tables	
Netezza 7.0	Active	Active	external tables	
Oracle 11g	Active	Active	Database Links	To use an Oracle data source for data ship, the DBA must install the DBMS_XPLAN package in the database and create an area for temporary tables. For this data source to

Data Source Type	Data Ship Source Support	Data Ship Target Support	Performance Option	Notes
				participate in data ship, it must be specified as a data ship source. Participation as a data ship target is optional. If Oracle is both source and target, DB Link needs to be set up between the Oracle databases.
Oracle 12c	Active	Active	Database Links	
Oracle 19c	Active	Active	Database Links	
PostgreSQL 9.1	Active	Active	Database Links	
PostgreSQL 9.2.3	Active	Active	Database Links	
Snowflake				
Sybase IQ 15	Active	Active	Location: iAnywhere JDBC driver	For a Sybase IQ data source to participate in data ship, the QUERY_PLAN_TEXT_ACCESS database option must be set to ON. For this data source to participate in data ship, it must be specified as a data ship source. Participation as a data ship target is optional.

Data Source Type	Data Ship Source Support	Data Ship Target Support	Performance Option	Notes
Teradata 13.00	Active	Active	FastLoad/ FastExport	For this data source to participate in data ship, it must be specified as a data ship source. Participation as a data ship target is optional. Teradata Fastload mode doesn't work correctly using the 14.10 JDBC driver when Teradata is the Target Data Source. To workaround this issue, use a later version of the Teradata JDBC driver.
Teradata 13.10	Active	Active	FastLoad/ FastExport	
Teradata 14.10	Active	Active	FastLoad/ FastExport	
Teradata 15	Active	Active	FastLoad	
Teradata 16.20	Active	Active	FastLoad	
Vertica 5.0	Inactive	Inactive		
Vertica 6.1	Active	Active	Bulk load utility Export to another Vertica database	

SDV DDL Feature Support

SDV DDL (Data Definition Language) feature to CREATE and DROP tables directly is supported in the following data sources:

- Amazon S3
- Azure Data Lake Storage
- Composite
- ComputeDB
- DB2
- Greenplum
- HBase

- HSQLDB
- Apache Hive
- Local File Storage
- MSSQL
- MySQL
- SybaseIQ
- Netezza
- Oracle
- Postgres
- Sybase
- Teradata
- Vertica

Support for DDL Clauses

SDV supports certain DDL extension clauses for the following data sources. See the *SDV Reference Guide*, chapter *SDV SQL Keywords and Syntax* for a list of these extensions.

- ComputeDB
- Teradata
- Vertica

Data Sources Supported for Kerberos Token Pass-through

- IBM DB2 LUW version 9
- Oracle

With these Kerberos authentication modes:

- Microsoft memory-based
- Ticket cache file-based

- Specified data source name and password
- ...these Oracle data sources are supported for Kerberos:
- Database version 11gR2 with an Oracle 11g driver
- Database version 19c
- Microsoft SQL Server 2008, 2012, 2014, 2016 and 2019
- SOAP 1.1 and 1.2
- REST
- Sybase ASE v12 and v15
- WSDL 1.1
- XML over HTTP

SDV Operating Systems Support

- 64-bit Windows Server 2012, 2016 and 2019
- 64-bit RHEL AS 6.6 and 7.0

Communication Interfaces and Protocols

- ADO.NET
- JDBC
- OData
- ODBC
- Web Services

RHEL 9 - Required Locale Configuration for PostgreSQL (SDV Server)

SDV Server installation on RHEL 9 requires the en_US.UTF-8 locale to be present and configured for Linux.

By default, RHEL 9 minimal installations may not include this locale, which can cause PostgreSQL initialization failures during SDV installation.

Prerequisites

Ensure the following locale package is installed and configured before installing SDV Server.

Required Commands

```
sudo dnf install -y glibc-langpack-en
sudo localectl set-locale LANG=en_US.UTF-8
```

Verification

```
locale
```

Expected output:

```
LANG=en_US.UTF-8
```

Note: Failure to configure this locale may result in PostgreSQL installation or initialization errors during SDV setup.

Security Features

Security features are discussed throughout this guide:

- Kerberos can be used when connecting to several data sources ([Supported Data Sources](#)).
- Password protection is available for operations like installing and starting SDV and registering with data sources like SAP (see Registering with the SAP System Landscape Directory, in the SDV User Guide).

Support and Maintenance Policies for Spotfire Products

Spotfire provides support and maintenance for major/minor releases of SDV.

Support Policies for Third-Party Environments

All versions stated of an environment presume the initial release of a Third-party product without any need for patches, service packs or equivalent terms unless stated. Equally, unless stated, we presume that patches or service packs and minor version releases are upward compatible for our products. Whenever a new release of SDV requires deployment of a patch or service pack or is compatible only with a minor version of an environment, SDV will highlight these requirements in release notes and will require customers to install a patch or service pack or minor version to receive support and maintenance on the product.

The following classifications indicate the level of support for the current release.

Classification	Description
Active	All aspects (design/creation and runtime) are supported in Studio and Server.
Desupported Not Supported	Design/creation of platform version is no longer supported, runtime will persist until the next major or minor version. OR: This platform version has not been added to SDV yet.
Deprecated	Runtime removed from SDV. Old data sources will need to be upgraded to platform versions that are supported
Inactive	Design/creation and runtime are allowed in Studio and Server, no active testing or development of new features will be performed to the platform version

Support Policies for Third-Party Application Virtualization Environments

Customers deploying Spotfire's products in third-party application virtualization environments from VMWare, Xen, and others should first consult the list of native host environments supported by SDV to verify compatibility. Support issues arising from deploying SDV in any Third-party application virtualization environments will be reviewed and resolved only on the native host operating system to remove any incompatibilities that might be introduced by the application virtualization environment itself.

Limitations for SDV Discovery

Servlets are not supported and cannot be imported from previous SDV versions.

Data Sources Not Supported by Discovery

Discovery supports all data sources and SDV Adapters except the following:

Data Sources Not Supported by Discovery

- Custom Java procedures—Not supported because they are procedural.
- DB2 z/OS
- Hive
- HP Neoview
- IBM DB2 z/OS Version 8, Version 9, Version 10
- Impala
- Netezza
- PostgreSQL
- Relational data sources—Procedural objects in relational data sources are not supported.
- SAP BW
- SAP HANA
- Teradata
- Vertica
- WSDL
- XML (flat files or over HTTP)

Studio and Server Connectivity and Installer Limitations

You can sometimes mix versions of Studio and Server as follows within a major release.

Studio Version	Server Version	Support
older	newer	Active
newer	older	Not active

For example:

- Connecting a 8.0.0 Studio with a 8.2.1 or 8.2.2 Server is supported.
- Connecting a 8.2.1 Studio with a 8.2.0 Server is not supported.

Business Directory and Deployment Manager Limitations

You can sometimes mix versions of Business Directory, Deployment Manager, and SDV as follows.

BD/DM Web UI Version	SDV Version	Support
older	newer	Not Supported
newer	older	Supported

For example:

- Business Directory 8.4, 8.5 clients are not compatible with published resources from SDV 8.6.
- The use of Business Directory 8.6 clients with published resources from SDV 8.4 is supported.
- When trying to connect to 8.5/8.6 SDV servers from 8.3/8.4 DM to create a site, for example, the keystore and truststore files must match between the servers if default SDV ones are used. This is to ensure backward compatibility

Installer Limitations

- 64-bit installers are supported only on 64-bit platforms.
- Linux and Windows installers are available only on the x86 hardware platform..

Type of Client	Requirements
64-bit Studio client	• They are at the same SDV version and patch level. • The Server version is newer than the Studio version and they are both within the same major SDV version.

Installing SDV Server, Studio, and Drivers

This topic describes how to install SDV on both Windows and UNIX computers and then verify that the installation was successful.

Topics include:

- [Overview of Installation Steps](#)
- [Preparing Your System for Installation](#)
- [Installing on Windows](#)
- [Installing on UNIX](#)
- [About the Installed SDV Services](#)
- [Tips from an Expert if the Server Does Not Start](#)
- [Where to Go After Installation](#)

Refer to the following sections for the other SDV distribution platforms:

- [SDV Docker Container](#)
- [SDV Container Orchestration Using Kubernetes](#)

Overview of Installation Steps

This section includes the following topics:

- [Installation Overview for New SDV Software Customers](#)
- [Installation Overview for Existing Customers Upgrading from a Previous Release](#)

Installation Overview for New SDV Software Customers

If you are installing SDV Data Virtualization products for the first time, here is an overview of how you would proceed:

1. Review the new features and bug fixes as documented in the *SDV and Business Directory Release Notes*.
2. Review the information in the following topics:
 - [Installation Requirements and Support Information](#)
 - [About SDV Software Patches](#)
 - [Preparing Your System for Installation](#)
3. Install SDV as described in:
 - [Installing SDV Server, Studio, and Drivers](#)
 - [Silent Mode Installation](#)
4. If a SDV service pack release exists for your SDV installation, then it is recommended to apply that to your new installation. Instructions for how to install a patch or service pack are subject to change with each service pack. For instructions, see [SDV and Business Directory Product Maintenance](#).

Installation Overview for Existing Customers Upgrading from a Previous Release

To install a major upgrade for SDV

1. Review the new features and bug fixes as documented in the *SDV and Business Directory Release Notes*.
2. The *SDV and Business Directory Release Notes* also lists the steps you need to follow when doing an in-place upgrade from an existing SDV installation. It is advised to follow those steps when doing an Upgrade.
3. Review the information in the following topics:
 - [Installation Requirements and Support Information](#)
 - [About SDV Software Patches](#)

- [Preparing Your System for Installation](#)
- 4. Review and follow the steps in [Upgrading from an Earlier Release and Migrating The Metadata](#).

Preparing Your System for Installation

To prepare your systems for installation

1. Review the new features and bug fixes as documented in the *SDV and Business Directory Release Notes*.
2. Review any README file included in your installation, patch, or service pack bundle.
3. Review the following requirements:
 - [Installation Requirements and Support Information](#)
 - You must have administrator privileges on the host computer to install SDV Server.
 - You can have multiple JVMs running on the installation machine.
 - Server requires a block of nine ports for use by SDV and associated services. The port setting for Web services HTTP communication serves as the “base port”. By default, the base port is 9400, but you can change it after installation using configuration parameters.
4. Make sure that any LIBPATH or LD_LIBRARY_PATH environment variable that you might have does not begin with a "/" slash or end with a ":" colon. Those characters may keep the repository from starting successfully.
5. Review your firewall settings and verify that they allow access to the ports that SDV (Business Directory, Deployment Manager, and SDV) products need to use.
6. To see the current base port setting, choose Configuration from the Administration menu and navigate to Server > Web Services Interface > Communications > HTTP > Port (Current).

Note: Changing the HTTP base port value also changes the value of all derived ports after the next SDV restart. When the base port is changed, you must update all data sources with the new port information.

These example ports are reserved or are derived from the base port:

9400	Web services HTTP port
9401	JDBC, ODBC, and ADO.NET
9402	Web services HTTP SSL
9403	JDBC SSL, ODBC SSL, and ADO.NET SSL
9404	Default caching database port
9405	JMX/RMI port for Monitor collector
9406	Monitor Daemon
9407	Active Cluster - JGroups (when enabled/when cluster is created)
9408	Repository
9409	Monitor RMI registry + JMX/RMI port for Monitor daemon
9500	Business Directory
9502	Business Directory (reserved)
9508	Business Directory

7. Stop Server if an earlier version is running.
8. Restart databases, especially those used for your caches and repositories.
9. Shut down all other application programs running on the installation machine.
10. Make sure you know the hostname or the IP address of the installation machine.
11. If you are installing on a Linux operating system, see [Preparing UNIX for SDV Installation](#).
12. If you are installing on a Windows operating system, see [Preparing Microsoft Windows for SDV Installation](#).

Preparing UNIX for SDV Installation

This section applies only if you are installing SDV on a machine running a supported UNIX operating system. Examples of valid and invalid `/etc/hosts` file entries are shown in the following table.

Validity	/etc/hosts File Entry
Valid	127.0.0.1 localhost IP hostname.domain hostname
Valid	127.0.0.1 localhost localhost.localdomain IP hostname.domain hostname
Valid	127.0.0.1 localhost localhost.localdomain localhost IP hostname.domain hostname
Invalid	127.0.0.1 localhost.localdomain IP hostname.domain hostname
Invalid	127.0.0.1 localhost.localdomain localhost IP hostname.domain hostname

To prepare your UNIX machine for installation of SDV products

1. Review the new features and bug fixes as documented in the *SDV and Business Directory Release Notes*.
2. Run the following command to determine if localhost can be resolved on the target installation machine before attempting an installation:

```
ping localhost
```

3. If the ping results look like the following, localhost is being resolved and the machine is ready for SDV installation. You can continue with the instructions in other sections.

Linux Ping Example with Valid localhost

```
$ ping localhost
```

```
PING localhost (127.0.0.1) from 127.0.0.1 : 56(84) bytes of data.
64 bytes from localhost (127.0.0.1): icmp_seq=1 ttl=255 time=0.071 ms
64 bytes from localhost (127.0.0.1): icmp_seq=2 ttl=255 time=0.063 ms
64 bytes from localhost (127.0.0.1): icmp_seq=3 ttl=255 time=0.043 ms
--- localhost ping statistics ---
3 packets transmitted, 3 received, 0% loss, time 1999ms
rtt min/avg/max/mdev = 0.043/0.059/0.071/0.011 ms
```

4. If the ping results look like the following, localhost is not correct. You must edit your `/etc/hosts` file.

Linux Ping Example with Invalid localhost

This example of `/etc/hosts` files shows where Server is unable to connect to the repository database because of the `localhost.localdomain` entry preceding the `localhost` entry (assuming the `localhost` entry exists at all).

```
$ ping localhost

PING localhost.localdomain (127.0.0.1) from 127.0.0.1 : 56(84) bytes of data.
64 bytes from localhost.localdomain (127.0.0.1): icmp_seq=1 ttl=255 time=0.080 ms
64 bytes from localhost.localdomain (127.0.0.1): icmp_seq=2 ttl=255 time=0.071 ms
64 bytes from localhost.localdomain (127.0.0.1): icmp_seq=3 ttl=255 time=0.044 ms
--- localhost.localdomain ping statistics ---
3 packets transmitted, 3 received, 0% loss, time 1998ms
rtt min/avg/max/mdev = 0.044/0.065/0.080/0.015 ms
```

5. Edit the `/etc/hosts` file to add a `localhost` entry, directly after the `127.0.0.1` entry, with the following syntax:

```
127.0.0.1 localhost <optional host name>
```

6. Save your changes and rerun the ping for `localhost`.

Preparing Microsoft Windows for SDV Installation

If you are installing SDV on Windows Server 2012 R2, you must disable User Account Control before installing SDV Server. Optionally, if you run into permission issues when running the installer, you can use this procedure to attempt to solve the issue.

To disable the User Account Control

From the Windows Start menu, select `Control Panel > User Accounts > Change User Account Control Settings`.. Change the setting to `Never notify`.

Installing on Windows

This installation process is used to install one or more components of SDV. You install SDV for Windows using the InstallAnywhere installer wizard.

Note: If you installed SDV on Microsoft Windows Server 2012 R2, see [Preparing Microsoft Windows for SDV Installation](#).

- [Running the SDV Server Installer](#)
- [Running the Studio Installer](#)
- [Installing the Drivers](#)

Running the SDV Server Installer

This installer installs the following components:

• Server	• Deployment Manager	• Repository
• Java	• Monitor	• Discovery
• Active Cluster	• Salesforce.com Adapter	• SAP Adapter
• SAPBW and BEx Adapters	• Oracle EBS Adapter	• Siebel Adapter
• Studio	• Default caching database	• Advanced Data Sources Adapters

To install SDV on a Windows computer

1. Read the README files included with or associated with the download file.
2. Run the installer executable for your platform (for example, the installer executable for SDV 8.9 Windows installation is *SPOT_sdv_server_8.9.0_win_x86_64.exe*)

Note: Make sure your Windows OS has all the necessary updates installed, so that SDV installation can run smoothly. SDV installer and patch are packaged with VC++ redistributables which are needed for the postgres and svn connections to run smoothly.

- The installer installs Microsoft Visual VC++ version 2015-2019. This is required by postgres.
- SDV installer also installs Microsoft Visual VC++ version 2013. This is required for svn connections to be successful. When using a patch, this version of VC++ (2013) needs to be manually installed. Execute the file vc_redist_x64.exe which can be found in the directory <SDV_install_dir>\bin

3. Follow the prompts on the screen. Special characters and spaces are not supported for <SDV_install_dir>.

You can select the defaults for the <SDV_install_dir> and the SDV Server base port number.

You will be prompted for the following passwords.

SDV Password Type	Description
SDV Server Application Password	The SDV Server application password is used to login to web manager.
SDV Repository	This is the database that will be used to store all of the data and metadata about the items that you create within SDV. It also stores your configuration and other environment settings. Passwords with special characters that are supported by your operating system shell are fine to use.
Default Caching Database	This is the database that will be created for you to hold data that you want to cache using the default caching method. There are multiple caching options. You might want to note the password for future use of this database.

The installation process might take a few minutes, during which progress windows are displayed.

Note: SDV will generate a new encryption key when the installation is in a new destination. If it is an existing installation SDV uses the existing encryption key. In case of any errors encountered, the administrator may have to investigate if the encryption key file location and content are correct and then contact SDV support team for assistance.

4. Select a password for the default caching database.
5. Finish to exit the installer when the installation is completed.

The Server starts automatically at the completion of the installation process. You can also start and stop the services as described in the SDV Administration Guide.

Install and uninstall logs are called `bitrock_installer_<number>.log` while the installer is running. After installation is complete, the logs are named `<product>_install` or `<product>_uninstall.log`. The log files can be found in the following directories:

Platform	Default Location of Log Files
Unix	/tmp
Windows	C:\Users\<username>\AppData\Local\Temp

6. Optionally, download and install the latest SDV patch as described in SDV and Business Directory Release Notes.

Running the Studio Installer

This installer installs the following components:

- Studio
- Java

This installer can be run on each Windows machine that needs access to the SDV Server.

To install Studio on a Windows computer

1. Read any README files included with or associated with the download file.
2. Run the installer executable for Studio (for example, the installer executable for SDV 8.9 Windows installation is `SPOT_sdv_studio_8.9.0_win_x86_64.exe`).
3. Follow the prompts on the screen.
4. When the installation is complete, click Finish to exit the installation program.

Studio automatically runs and prompts you for login information.

Installing the Drivers

This client distribution (driver zip) file includes the following components:

- ODBC
- ADO.NET
- JDBC
- SSIS
- Power BI

This zip file can be unpacked on each machine that has client application that needs access to the SDV Server.

To install the drivers distributed with SDV

1. Read any README files included with or associated with the download file.
2. Locate and extract the drivers zip file.
3. When installing the ODBC Win 64-bit driver on Windows 10, make sure to select Run as Administrator. Select the client EXE file, right click and select Run as Administrator. When prompted, select Yes and allow the installation to run to completion.
4. Follow the instructions in the SDV Administration Guide for details on how to complete configuration of each driver.
5. When the installation is complete, click Done to exit the installation program.

Installing on UNIX

Your SDV Server can be installed on a UNIX machine. Studio is not available for UNIX and must be installed on a Windows machine. You can then connect the Studio client to the Server on the UNIX machine.

- [Installing SDV Server on UNIX](#)
- [Installing Drivers on UNIX](#)

- [Setting the SDV Server to Start Automatically on UNIX](#)

Installing SDV Server on UNIX

This installer installs the following components:

• Server	• Deployment Manager	• Repository
• Java	• Monitor	• Discovery
• Active Cluster	• Salesforce.com Adapter	• SAP Adapter
• SAPBW and BEx Adapters	• Oracle EBS Adapter	• Siebel Adapter
• Default caching database	• Advanced Data Source Adapters	

To install SDV on a UNIX computer

1. Make sure you have reviewed and completed any necessary preparation as discussed in [Installation Requirements and Support Information](#).
2. For CentOS, Red Hat Enterprise Linux, and Oracle Red Hat Enterprise Linux systems Security-Enhanced Linux (SELinux) must be enabled. Refer to the section *Configuring Security Enhanced Linux Environments* in the Security Features Guide.
3. If necessary, log into the installation machine as a non-root user. Change your working directory to the user's home directory.
4. Run the following command for your platform:

```
chmod 755 <installer file name>
```

For example, the <installer_file_name> for SDV 8.9 Unix installation is *SPOT_sdv_server_8.9.0_linux_x86_64.bin*

5. Make sure that the directory and path that you expect to use for SDV does not contain any spaces.

6. Make sure that you have READ and WRITE permissions on the installation directory.
7. Run the following command to start the installation:

```
./<installer file name>
```

8. Follow the prompts on the screen. Special characters are not supported for <SDV_install_dir>.

You can select the defaults for the <SDV_install_dir> and the SDV Server base port number. The value you use for <SDV_install_dir> cannot contain a space.

You will be prompted for the following passwords...

Password Type	Description
SDV Server Application Password	The SDV Server application password is used to login to web manager.
SDV Repository	This is the database that will be used to store all of the data and metadata about the items that you create within SDV. It also stored your configuration and other environment settings. Passwords with special characters that are supported by your operating system shell are fine to use.
Default Caching Database	This is the database that will be created for you to hold data that you want to cache using the default caching method. There are multiple caching options. You might want to note the password for future use of this database.

Note: SDV will generate a new encryption key when the installation is in a new destination. If it is an existing installation SDV uses the existing encryption key. In case of any errors encountered, the administrator may have to investigate if the encryption key file location and content are correct and then contact SDV support team for assistance.

9. Finish to exit the installer when the installation is completed.

The Server starts automatically at the completion of the installation process. For information about automatically restarting SDV, see [Setting the SDV Server to Start Automatically on UNIX](#). You can also start and stop Server as described in [About the Installed SDV Services](#) and the SDV Administration Guide.

Install and uninstall logs are called `bitrock_installer_<number>.log` while the installer is running. After installation is complete, the logs are named `<product>_install` or `<product>_uninstall.log`. The log files can be found in the following directories:

Platform	Default Location of Log Files
Unix	/tmp

10.If installing SDV on AIX, make sure that `MAX_MEMORY >1500MB` is in the `<SDV_install_dir>/conf/server/server.properties`.

The `server.properties` file is processed every time the server is restarted from `composite.sh monitor`.

11.Optionally, download and install the latest SDV patch as described in SDV and Business Directory Release Notes.

Installing Drivers on UNIX

These files contain the following driver components:

- ODBC
- ADO.NET
- JDBC
- SSIS
- Power BI

To install the drivers

1. Make sure you have reviewed and completed any necessary preparation as discussed in [Installation Requirements and Support Information](#).
2. If necessary, log into the installation machine as a non-root user. Change your working directory to the user's home directory.
3. Make sure that you have READ and WRITE permissions on the directory for which you want to unzip the contents of the file.
4. Locate and extract the drivers zip file.

5. Follow the instructions in the SDV Administration Guide for details on how to complete configuration of each driver.
6. When the installation is complete, click Done to exit the installation program.

Setting the SDV Server to Start Automatically on UNIX

If at any time after installing the software, you restart the UNIX installation machine, Server and the metadata repository do NOT start automatically (unlike when they start automatically after a successful installation of the software).

To configure the SDV service files `cis.repository` and `cis.server`

1. Log into the installation machine as root.
2. Change the working directory to `<SDV_install_dir>/bin`.
3. Run the following command as the root user:

```
cis_install_services.sh
```

This command prompts for a username, and other details to install and configure the service files `cis.repository` and `cis.server`.

4. Enter the name of the user to start SDV (not the root user) and the other information requested.

The script then installs `cis.repository` and `cis.server` into an appropriate location on the installation machine and configures them. The location will be printed on your screen when the configuration is successful, so make note of this location, because you need this to perform verification of the service files.

Note: Do not run the `cis.repository` or `cis.server` scripts in the `<SDV_install_dir>/bin/` directory. These are template files used by `cis_install_services.sh` only and are not meant to be run.

Running `cis_install_services.sh` does not interrupt any repository or server processes that are running, but prepares the machine for automatically starting those processes during restart of the UNIX-based computer.

5. Run the following commands as the root user:

```
cd <init_directory>

chmod 550 cis.repository

chmod 550 cis.server

chmod 550 cis.cache
```

The value of `init_directory` depends on the operating system:

- Linux: `/etc/rc.d/init.d` or `/etc/rc.d`
- AIX: `/etc/rc.d/init.d`

To verify the SDV service files configuration

6. Go to the location noted previously from running `cis_install_services.sh`.

Note: The console output of the script `cis_install_services.sh` displays the exact location. Choose the location for your operating system.

7. Enter these commands:

```
./cis.repository restart

./cis.server restart

./cis.cache restart
```

Now if the machine is rebooted, the monitor, server, and repository processes should automatically start once the machine is ready to go.

Installing on Amazon Web Service

The SDV Server is supported on Windows and UNIX. Studio requires a Windows-based OS to operate.

To install SDV on a Windows-based AWS

1. Install and configure a supported version of Windows for AWS.
2. Select and install the AMI for SDV.
3. Follow the install instructions in [Running the SDV Server Installer](#).

4. Follow the install instructions in [Running the Studio Installer](#).
5. Follow the instructions in the SDV Administration Guide to register you SDV licenses.

To install SDV Server on a UNIX-based AWS

1. Install and configure a supported version of Linux for AWS.
2. Select and install the AMI for SDV Server.
3. Follow the install instructions in [Installing on UNIX](#).
4. Locate the SDV Studio installer that came bundled with your AMI.
5. Move the installer file to a Windows-based AWS or another Windows machine.
6. Follow the install instructions in [Running the Studio Installer](#).
7. Connect to the SDV Server on your Linux AWS.
8. Follow the instructions in the SDV Administration Guide to register you SDV licenses.

About SDV Software Patches

SDV produces service pack patches as needed to update installed products. Patches are applied after the product has been installed. A patch is a zipped package of files that fixes known issues and which often provides enhanced functionality.

You must use the component-specific patch to get fixes for each component (For example, the Studio patch has fixes for Studio, the Client driver patch has fixes for client drivers such as ODBC, ADO.Net). Refer to [Software Components for Installation](#) for a list of components.

After installation of SDV, you might want to apply the latest SDV patch which might be a later version than what you just installed. It is recommended that you install a patch on all computers running SDV products to ensure complete compatibility and minimize unforeseen problems.

For information about how to obtain and install the latest patch, see [SDV and Business Directory Product Maintenance](#).

Note: Instructions for how to install a patch or service pack are subject to change with each service pack. For instructions, see the SDV and Business Directory Release Notes.

About the Installed SDV Services

The installation process installs the following services which are SDV processes that run in the background:

- server—the SDV Server process.
- repository—the database repository used by SDV.
- monitor—a process that monitors the SDV Server and ensures that it is always running.
- cache—a process that runs the default caching database.

All processes must be running for SDV to function properly.

For more information on configuring and starting SDV, see the SDV Administration Guide.

Importing Metadata into the New SDV Instance

If you are upgrading your version of SDV from an earlier version and you have completed the instructions in [Exporting Metadata from the Existing SDV Instance](#), follow the instructions in the section [Importing Metadata into the New SDV Instance](#). To verify the installation, follow the instructions in the section [Verifying a Successful Installation](#)

Tips from an Expert if the Server Does Not Start

If the server does not start and the log files indicate that the cause is not enough heap memory, you can modify the default max memory setting.

The server.properties file is processed every time the server is restarted from `composite.sh monitor`.

To modify the max memory setting

1. Stop the server.
2. Increase the MAX_MEMORY value in the one of the following locations depending on your server:
 - <SDV_install_dir>/conf/server/server.properties
 - <BD_install_dir>/bd/conf/server/server.properties
3. If adjusting the heap size with MAX_MEMORY is not enough to allow large CAR files to load, you can try setting the following Studio configuration parameters back to their default values:
 - Default Bytes to Fetch—Default value is 100.
 - Default Rows to Fetch—Default value is 1000.
4. From the process manager for your platform, shut down and restart all SDV processes (such as the SDV Server and monitor).

Where to Go After Installation

For your next steps, particularly if you are new to SDV products, see the information in the following PDFs or online help. You can access the PDFs at <SDV_install_dir>/docs, or from within Studio at Help > Online Help.

Book Title	Description
Getting Started Guide	Contains a simple example to get you familiar with the Studio application.
Administration Guide	Contains procedures for: • Completing and configuring your SDV installation • Licensing SDV software • Starting and stopping SDV • Finding and interpreting log files • Setting up security

Book Title	Description
	• Setting up JDBC, ODBC, and other drivers
User Guide	Explains Studio features and how to create and publish resources.
Client Interfaces Guide	Contains instructions, guidelines, and examples of how to access SDV resources through various client applications.

Silent Mode Installation

Installations can be run without manual interactive interfaces (i.e. graphical user interface or console based). There are two ways to run the installer in silent mode: 1) using a property file with key/value pairs or 2) command line with key/value pairs.

Topics covered in this chapter include:

- [Creating the Options File for a Silent Installation](#)
- [Running the Installer in Silent Mode](#)

Creating the Options File for a Silent Installation

Optionally, when running a silent mode installation you can use an options file that has specific key-value pairs.

To create the options file for a silent install

1. In a text editor, create a options file similar to the following:

Business
Directory

```
# Modify install directory and all port number references
# mode=unattended
install_directory=/opt/Spotfire/BD
server_port=9500
repository_admin_password=password
bd_admin_password=password
```

SDV
Server

```
# Modify install directory and all port number references
# mode=unattended
install_directory=/opt/Spotfire/SDV
server_port=9400
repository_admin_password=password
database_admin_password=password
server_admin_password=password
```

2. Edit the values within the file for your installation.

The following table describes the variables in the options file:

Variable	Description and Value
INSTALL_DIRECTORY	Directory in which to install the software referred to as <SDV_Installdir>. The value can be empty, or the directory can be non-existent. On UNIX, there can be no space in the directory name. Examples: <pre>install_directory=/opt/Spotfire/SDV install_directory=C:\Program Files\Spotfire\Studio install_directory=/opt/Spotfire/BD</pre>
REPOSITORY_ADMIN_PASSWORD	Password to access the repository database, which is automatically installed during the installation. PostgreSQL requires that the password you choose cannot contain a # or \$.
SERVER_PORT	Defaults to 9400 for SDV and 9500 for Business Directory.
DATABASE_ADMIN_PASSWORD	The password used to access the default caching database, which is automatically created during installation. PostgreSQL requires that the password you choose cannot contain a # or \$.
SERVER_ADMIN_PASSWORD	The password used to login to the web manager and the client applications.

3. Save the file as <installer.properties>.

Running the Installer in Silent Mode

Running the installer via command line options

Option file method:

- a. Create the options file. See [Creating the Options File for a Silent Installation](#).
- b. Run the installer with the following option: <instFile>.exe/bin --optionfile <OPTION_FILE>

1. Command line (no options file) method - See examples below:

Component	Command Options
SDV Server	- Windows Installation with all input parameters: <pre><instFILE>.exe --mode unattended --install_directory <SDV_Installdir> --server_port "6400" --server_admin_password "admin1" --repository_admin_password "password" --database_admin_password "password"</pre> Note: database_admin_password is only valid for SDV Server. BD does not use this variable. - Windows Installation with only the required parameters: <pre><instFILE>.exe --mode unattended --server_admin_password "admin1" --repository_admin_password "password" --database_admin_password "password"</pre> This command installs SDV Server in the default directory C:\Program Files\Spotfire\SDV Server <version> on default port 9400. - Linux/AIX Installation with all input parameters: <pre><instFILE>.bin --mode unattended --install_directory <SDV_Installdir> --server_port "6400" --server_admin_password "admin1" --repository_admin_password "password" --database_admin_password "password"</pre> - Linux/AIX Installation with only the required parameters: <pre><instFILE>.bin --mode unattended --server_admin_password "admin1" --repository_admin_password "password" --database_admin_password "password"</pre>

Component	Command Options
	This command installs SDV Server in the default directory /opt/Spotfire/SDV_Server_<version> on default port 9400. Note: User should have rwx permissions on /opt
Studio	Windows: <pre><instFILE>.exe --mode "unattended" --install_directory <SDV_Installdir></pre>
Business Directory	- Windows Installation with all input parameters: <pre><instFILE>.exe --mode unattended --install_directory "<SDV_Installdir>" --bd_admin_password "admin1" --repository_admin_password "password" --server_port 9500</pre> - Windows Installation with only the required parameters: <pre><instFILE>.exe --mode unattended --bd_admin_password "admin1" --repository_admin_password "password"</pre> This command installs BD in the default directory C:\Program Files\Spotfire\BD Server<version> on default port 9500. - Linux/AIX Installation with all input parameters: <pre><instFILE>.bin --mode unattended --install_directory "<SDV_Installdir>" --bd_admin_password "admin1" --repository_admin_password "password" --server_port 9500</pre> - Linux/AIX Installation with only the required parameters: <pre><instFILE>.bin --mode unattended --bd_admin_password "admin1" --repository_admin_password "password"</pre> The above command installs BD in the default directory: /opt/Spotfire/SDV_BD_Server_8.0 on port 9500

Component	Command Options
Note: User should have rwx permissions on /opt	

Note: All the available options can be viewed by executing the command
 <instFILE>.exe/bin --help

The variables used in the above table are as follows:

- <instFILE> is the file name. For example, SPOT_sdv_server_8.0.0_win_x86_64.exe for a Windows SDV Server.
 - <OPTION_FILE> is the name of the file where the input parameters are stored.
 - <SDV_Installdir> is the installation directory For example, /opt/Spotfire/SDV_BD_Server_8.0 for Linux/AIX BD
2. Verify that the installation was successful by looking for the SDV installation directory. You can also view success or failure messages in:
- %HOMEDRIVE%\BD_install.log (Windows) or /tmp/BD_install.log (UNIX)
 - %HOMEDRIVE%\SDV_install.log (Windows) or /tmp/SDV_install.log (UNIX)

SDV Docker Container

This section will cover the Spotfire® Data Virtualization (SDV) software for the Docker container distribution format. Users will be able to build a SDV Docker image and run it as a Docker container. The following sections are described in this chapter:

[Prerequisites](#)

[Building SDV Docker Images](#)

[Publishing SDV Docker Images](#)

[Launching SDV Containers](#)

[Runtime SDV Container Configuration - Common Examples](#)

[Limitations](#)

[Upgrading to a New Version of SDV Server](#)

[Tips from Expert](#)

[Useful Docker Commands for SDV Containers](#)

Prerequisites

The following section outlines what you need to prepare for building and running the Data Virtualization Docker image. SDV containers can run on any OCI compliant container runtime environment (<https://opencontainers.org/>).

Note: The SDV product does not provide a Docker image. You must build it explicitly.

Before building the Docker image, ensure that the following prerequisites are met.

Linux, macOS, and Windows platforms

Install the latest version of the Docker Engine 19.03.5 or higher (<https://docs.docker.com/get-docker/>). This is the build and runtime environment you will use to create the Data Virtualization Docker images and containers.

Use the following link to all non-root users to run Docker commands (<https://docs.docker.com/install/linux/linux-postinstall>) in your Linux Docker runtime environment.

Note: Linux based operating system are the most popular for hosting Docker runtime environments. If you are not using, that then look at the Windows OS or macOS alternatives below.

Windows platform

SDV supports Windows 11 Pro 64bit 21H2 or higher and Windows Server 2022 21H2 or higher. Some of the key considerations when using the docker desktop for windows are given below:

Hyper-V feature must be enabled on Windows (<https://learn.microsoft.com/en-us/windows-server/virtualization/hyper-v/system-requirements-for-hyper-v-on-windows>)

References:

- Windows 11 : <https://techcommunity.microsoft.com/t5/educator-developer-blog/step-by-step-enabling-hyper-v-for-use-on-windows-11/ba-p/3745905>
- Windows Server 2022: <https://learn.microsoft.com/en-us/windows-server/virtualization/hyper-v/hyper-v-technology-overview>

Use the Docker desktop for Windows to create the SDV docker images and containers. The latest version of Docker Desktop for Windows can be installed from here - <https://docs.docker.com/desktop/install/windows-install/>

Listed below are some of the important considerations, If you are using the docker desktop:

- WSL 2 feature must be enabled on Windows (<https://docs.docker.com/desktop/windows/wsl/>)
- During Docker Desktop for Windows installation, choose “Use WSL 2 instead of Hyper-V” option.
- Enable the “use Linux containers” configuration as outlined here: <https://docs.docker.com/desktop/get-started/#switch-between-windows-and-linux-containers>

If you choose to run the SDV docker build and run scripts in Docker Desktop for Windows, then you must open a WSL shell and run with bash.

Some examples of running the SDV script inside the WSL shell are given below:

Build example

```
$ bash build_sdv_image.sh
```

Run single instance example

```
$ bash run_sdv_container.sh -d --del-vol --env SDV_ADMIN_PASSWORD=123456
```

The SDV Docker container will require a minimum of 8GB of RAM and 8GB of read & write persistent disk storage. For additional resource requirements see [SDV](#)

macOS platform

Latest version of Docker Desktop for Mac (<https://docs.docker.com/desktop/install/mac-install/>)

Note : To allow non-root users to run Docker commands, follow the instructions in <https://docs.docker.com/install/linux/linux-postinstall>

Docker

Docker Engine 19.03.5 or higher is required for building/running a SDV Docker image/container.

Verifying Docker Installation

- In the command prompt, run the command:

```
$ docker run hello-world
```

- If you cannot run this default Docker container, then check your Docker installation. Being able to run this default Docker container ensures your Docker environment is ready for building and running SDV.

Note: When running SDV with a Docker container, Docker will manage the SDV Server process lifecycle, therefore the SDV Monitor Daemon will not be running inside the SDV container. You will notice that the `cs_csmonitor_server.log` displays errors. This is

because the monitor server is attempting to connect to a monitor daemon which does not exist in a docker environment. It is safe to ignore this error message.

SDV

Resource Requirements for SDV

Storage, CPU and memory resources should to be specified when building a SDV Docker image and when running a SDV Docker container.

- **Storage (runtime)** - SDV requires read/write persistent disk storage that is maintained outside of the Docker container. See [Sizing Guidelines for SDV](#) for recommendations.
- **CPU (runtime)** - SDV requires a minimum of 2 CPUs/cores per SDV Docker container. See [Sizing Guidelines for SDV](#) for recommendations.
- **Memory (build time)** - SDV requires a minimum of 8 GB memory per SDV Docker container. See [Sizing Guidelines for SDV](#) for recommendations.

Support

- The Java version supported is openjdk 21.0.10 LTS
- The Postgres version supported is 17.9
- SDV Docker distribution is supported on Ubuntu 24.04.4 LTS (SDV Server) and Debian 13.4 (SDV Repository and SDV Cache)

Building SDV Docker Images

SDV is packaged with a quick start container build script “build_sdv_image.sh”. You can download this script from the edelivery site:

<https://www.spotfire.com/downloads>

Alternatively, you can also use build without that script by just directly using the SDV Dockerfiles and corresponding tar.gz file that are provided on - <https://www.spotfire.com/downloads> alongside the other SDV distributions (e.g. installer and patch). Follow the steps given in the sections below, to build the SDV Docker Images.

Once you build the SDV Docker image, you can use private clouds (i.e. on-premises) or public clouds such as AWS and Azure to run the SDV application as a container. For information on deploying docker containers refer to the following documentation:

Manage SDV Image on AWS or Azure:

<https://docs.aws.amazon.com/AmazonECR/latest/userguide/what-is-ecr.html>

<https://docs.microsoft.com/en-us/azure/container-registry/container-registry-get-started-docker-cli?tabs=azure-cli>

Deploying SDV Container on AWS:

<https://docs.aws.amazon.com/AmazonECS/latest/developerguide/docker-basics.html>

<https://docs.docker.com/cloud/ecs-integration/>:

Deploying docker containers on Azure:

<https://docs.docker.com/cloud/aci-integration/>

Using Quick Start Script

Follow these steps, to build the SDV Docker Images using the quick start script:

1. Download the build_sdv_image.sh on your instance for building Docker images.
2. Run the following command:

```
$ ./build_sdv_image.sh [-hv]  
[-c] [-f <file>] [-n <name>:<tag>]
```

The table below gives a description of the different parameters used with the script:

Parameter	Description	Comments
-h	Help for using the script.	
-v	Enable verbose mode output.	
-c	Enable build caching.	default is "docker build --no-cache"

Parameter	Description	Comments
-f	Location of Dockerfile.sdv	
-n	Set name:tag for image.	default value: Server - "Spotfire/sdv:<version>" Cache - "Spotfire/sdvcache:<version>" Repo - "Spotfire/sdvrepo:<version>"

Using Docker Build

Follow these steps, to build the SDV Docker Images using the docker build command:

1. Download the docker container files Dockerfile.sdv, Dockerfile.sdv.repo, Dockerfile.sdv.cache and SPOT_sdv_<SDV_VERSION>_docker.zip on your instance for building Docker images. **The edelivery has separate archive files for each distribution. This archive is only for the Docker distribution.**
2. You should also download the md5 checksum files in order to validate the integrity of these files.

Example:

```
md5sum -c Dockerfile.sdv.md5
```

```
md5sum -c SPOT_sdv_<SDV_VERSION>_docker.zip.md5
```

3. Goto to the location where you have downloaded the SDV Docker files (Dockerfile.sdv, Dockerfile.sdv.repo, Dockerfile.sdv.cache) and tar.gz file and run the Docker build command as shown below.

Generic example:

```
$ docker build -t <your-docker-repo-name>/<image_name>[:<image-tag>] -f <image physical file name>
```

SDV default build example:

```
$ docker build -t myrepo/sdvrepo:<SDV_VERSION> -f Dockerfile.sdv.repo
```

```
$ docker build -t myrepo/sdvcache:<SDV_VERSION> -f Dockerfile.sdv.cache
```

```
$ docker build -t myrepo/sdv:<SDV_VERSION> -f Dockerfile.sdv #(default SDV base port 9400 and default SDV server memory 7 GB.
```

Publishing SDV Docker Images

If you want to publish your SDV Docker image to a Docker registry then log in to your Docker registry and follow the commands below.

For more information on Docker login, visit <https://docs.docker.com/engine/reference/commandline/login>

After a successful login, you can publish the Docker image using the docker push command.

Note: The example below shows the publishing command for one image. You will have to run the “docker tag” and “docker push” commands for each image (the repo, cache and server).

Generic Command Usage:

```
$ docker login --username <user-name> <remote-repo-name>
```

```
$ docker tag <local-repo-name>/<image-name>:<image-tag> <remote-repo-name>/<image-name>[:<image-tag>]
```

```
$ docker push <remote-repo-name>/<image_name>[:<image-tag>]
```

```
$ docker logout <remote-repo-name>
```

Launching SDV Containers

SDV is pre-packaged with the following deployment tools to help users to launch SDV Containers:

- Quick start container deploy script “run_sdv_container.sh” (for single node)
- Docker-Compose SDV example "run_sdv_docker_compose.sh" (you can use this script to run single/cluster modes)

You can download the SDV Docker distribution files from:

<https://www.spotfire.com/downloads>

Alternately, you can also use the SDV Docker files and the corresponding zip file that are available in the site - <https://www.spotfire.com/downloads> alongside the other SDV distributions (e.g. installer and patch). Follow the steps given in the sections below, to deploy the SDV Docker Images.

This section will explain how to start a SDV Docker container. If you need to review SDV container sizing guidelines refer [Sizing Guidelines for SDV](#)

Prerequisites

- To use docker compose you need the template engine Jinja. To install it,
 - Download the jinja 2 command line tool from <https://github.com/kolypto/j2cli>
 - Run the following commands:

```
pip install j2cli
pip install j2cli[yaml]
```

- If you encounter issues when running above commands, then you may also need to download and install pip:<https://pypi.org/project/j2cli/>

SDV Admin Password

There is no default SDV Admin password for the docker container and specifying an Admin Password is required when launching a SDV container. You can specify the password in one of the following ways:

1. By setting an environment variable SDV_ADMIN_PASSWORD prior to running the quick script or the “docker run” command:

```
$ export SDV_ADMIN_PASSWORD=<VALUE>;
```

2. By using the option “-e”:

```
$ docker run -e SDV_ADMIN_PASSWORD
```

Note: Refer to the sections [Launching SDV Containers \(Single Node\) - Using Quick Start Script](#) and [Launching SDV Containers \(Cluster Nodes\) - Using Quick Start Script](#) for the usage of the “-e” option.

3. By storing the password in a file and using that file while launching the container:

```
$ docker run -env SDV_ADMIN_PASSWORD_FILE=<FILE> # The file should have the
SDV_ADMIN_PASSWORD=<VALUE> set.
```

Note: You can also store all the docker run parameters in an environment file and use the file while launching the container:

```
$ docker run --env-file <FILE>. # <FILE> The file should have the SDV_ADMIN_
PASSWORD=<VALUE> set along with other required parameters.
```

Launching SDV Containers (Single Node) - Using Quick Start Script

Follow these steps to run the script and launch the SDV container using the quick start script:

1. Download the run_sdv_container.sh on your instance for launching the Docker container.

Note: For Docker Desktop Window users, you must run “bash run_sdv_container.sh” from a WSL shell if you want to run this script.

2. Run the following command:

```
$ ./run_sdv_container.sh [-hv] [--dry-run][--skip-wait]
[-d] [--del-vol] # deletion settings (container and volumes)
[-c <cpus>] [-i <name>:<tag>] [-m <memory>] [-n <network>] [--name <name>] [-p <port>]
[--vol <volume>] #container settings
```

```
[ -e <SDV_ARG> ... ] [ --env <SDV_ARG>=<SDV_VALUE> ... ] [ --env-file <file> ] # SDV
runtime configuration
```

The table below gives a description of the different parameters used with the script.

Parameter	Description	Comments
--dry-run	Show output of execution without actually executing the script.	
--skip-wait	Skip wait check for SDV container.	
-h	Help for using the script.	
-v	Enable verbose mode output.	
Deletion Settings (Container and Volume)		
-d	Delete and stop SDV container if it already exists with same <name>:<tag>.	
--del-vol	Delete and stop SDV container if it already exists with same <name>:<tag>.	
Container Settings		
-c	Set the number of CPUs for container.	Use decimal or positive whole number form. Default is "2.0". "1.0" CPU is the minimum. Any value below that will result in an error.
-i	Set <name>:<tag> for image to use for container.	default is "Spotfire/sdv:version"

Parameter	Description	Comments
-m (memory)	Set the amount of memory for container.	Default is 7GB. 4GB is the minimum. Any value below that will result in an error. Note: If this is set and -e SDV_MAX_MEMORY is not, then SDV_MAX_MEMORY defaults to the value mentioned in this option.
-mem-repo-memory	Set the amount of memory for sdv repo and cache containers.	If you need to override the default memory of the cache and repo containers, you can use this option.
-n <network>	Set network for container.	Default is "sdv-bridge". If <network> doesn't exist then an error will result.
--name	Set <name> for container.	Default is "sdv<version>".
-p <port>	Set base host port for container	Default is "9400".
--vol	Set <volume> for container	default is "sdv<version>-vol"
SDV Runtime Configuration		
--env <SDV_ARG>=<SDV_VALUE>	Pass SDV parameters to container for runtime configuration.	Required Settings: --env SDV_ADMIN_PASSWORD=<PASSWORD> # Set SDV admin password for all containers (i.e. sdv, cache and repo). or --env SDV_ADMIN_PASSWORD_FILE=<clear text password in file> # (optional) Use file to store SDV admin

Parameter	Description	Comments
		password. Specified file remapped to /run/secrets/sdv-admin-password in container. Note: only SDV_ADMIN_PASSWORD or SDV_ADMIN_PASSWORD_FILE can be specified. Setting both is not allowed. Optional Settings: --env SDV_BASE_PORT=9400 # (optional) Change SDV Server base port. --env SDV_MAX_MEMORY=7 # (optional) Change SDV Server memory value (GB). If this is set and -m <memory> is not, -m <memory> defaults to SDV_MAX_MEMORY + 1 GB.
-e <SDV_ARG>	Pass SDV parameters to container for runtime configuration, using defined environment variables.	The variables used in the argument of this option are environment variables defined and assigned values ahead of the usage.
--env-file <file>	Pass SDV parameters file to container for runtime configuration.	

Note: By default, the script runs the docker command with the “---restart” option set as “unless-stopped”. It means “Restart the container if it stops, except that when the container is stopped (manually or otherwise), it is not restarted even after Docker daemon restarts.”

Launching SDV Containers - Using Docker Compose

Docker Compose is a tool used for running multi-container Docker applications. With this tool, you make use of a YAML file to configure your application's services. SDV requires the version of Docker Compose to be 1.24 or higher. SDV also requires the Docker Compose File Format version to be 3.5 or higher.

To run the docker compose, download the `SPOT_sdv_<VERSION>_docker.zip` file from the [edelivery site](#) and extract the contents of the file. The docker compose and related files are extracted in a separate folder called `docker_compose`:

- `docker-compose.yml.j2`
- `haproxy.cfg.j2`
- `run_sdv_docker_compose.sh`
- `sdv-k8s.properties.j2`
- `values.yml`

Note: All the files with extension ".j2" are template files and will be regenerated when the script is run.

To use docker compose you need the template engine j2cli. It is a command-line tool that can be used as a template in shell-scripts. To install it,

- Download the jinja 2 command line tool from <https://github.com/kolypto/j2cli>
- Run the following commands:

```
pip install j2cli
pip install j2cli[yaml]
```

- If you encounter issues when running above commands, then you may also need to download and install pip: <https://pypi.org/project/j2cli/>

Before you run the docker-compose script, make sure that the `values.yml` file has the correct values for the sdv image name, sdv version, ports and the number of nodes (a value greater than 1 indicates that its a cluster node). For example, for SDV 8.9, the following are the default values set for the images. Please change it, if your sdv image name is different:

- sdvserver image: Spotfire/sdv:8.9.0
- sdvcache image: Spotfire/sdvcache:8.9.0
- sdvrepo image: Spotfire/sdvrepo:8.9.0

After verifying the values.yml and making appropriate changes, simple run the following script:

```
run_sdv_docker_compose.sh
```

Note: You need root permission to execute the script due to a sudo call to create local storage directories for your containers. Contact your organization's administrator to get the permission needed.

Runtime SDV Container Configuration - Common Examples

The following topics are explained in this section:

[Change SDV Admin Password \(while container is running\)](#)

[Change SDV Base Port](#)

[Change SDV Server Memory Setting](#)

[Configure External Volume For Local Persisted File Data Sources](#)

[Configure Data Source With 3rd party JDBC Driver \(type 4\)](#)

[Configure Data Source With 3rd party JDBC Driver \(type 3\)](#)

[Configure JVM Settings from Docker](#)

[Configure LDAP Properties From Docker](#)

[Configure Truststore and Keystore Files From Docker](#)

Text Editor

SDV Docker distribution is packaged with the text editor Nano. For a list of editor commands refer to <https://www.nano-editor.org/dist/latest/cheatsheet.html>

Change SDV Admin Password (while container is running)

To change the SDV Docker container's admin password you will need to use the SDV Studio client.

Note: Your default SDV Docker admin user "admin" password is defined in your Dockerfile.sdv for your given SDV release.

Follow these steps to change the password:

1. Use SDV Studio (same version of SDV as your SDV Docker container) to connect to your SDV Docker container. Login to the SDV Studio using the SDV admin user "admin" and your SDV admin password along with the appropriate SDV base port.
2. Select the "File" tab in the SDV Studio client menu options and choose "Change Password".
3. In the new pop-up dialog window, provide the current SDV admin user "admin" user password and the new password. When completed, click OK.
4. SDV Studio client will not logout of the client and a new login dialog will be displayed. Proceed to login again with your new SDV admin user "admin" password to verify the update.

Change SDV Base Port

To change the SDV Docker container's base port you will need to reconfigure SDV and create a new SDV Container.

Example (change base port from 9400 to 10400):

a. Before starting the SDV Container:

To map HOST_PORT to internal SDV container port SDV_BASE_PORT (default 9400)

```
-p HOST_PORT
```

To change internal SDV container port to 10400 instead of default 9400. Host port is also 10400 if "-p" is not set.

```
--env SDV_BASE_PORT=10400
```

b. If the SDV Container is already running:

1) Use SDV Studio (same version of SDV as your SDV Docker container) to connect to your SDV Docker container. Login to the SDV Studio using the SDV admin user "admin" and your SDV admin password along with the appropriate SDV base port.

2) Goto Administration > Configuration

3) Search for "Port"

4) From the search result, select "Port (On Server Restart): 9400".

5) Change your base port value to "10400". Apply your Changes.

Note: "Port (Current)" shows your current SDV Server base port configuration.

"Port (On Server Restart)" shows your future SDV Server base port configuration. This only takes affect when you restart the SDV Server though.

6) Now click on the "Ok" button to close the pop-up window.

7) Logout of the SDV Studio client.

8) Stop the SDV Docker container where your SDV Server is running.

```
docker stop <sdv-container-name>
```

9) Start a new SDV Docker container with the new base ports:

```
./run_sdv_container.sh -p 10400 --env SDV_ADMIN_PASSWORD=admin1
```

Note: Running the run_sdv_container.sh script in verbose mode (-v) will display the low level docker run commands which will be executed to achieve the result. Refer to [Launching SDV Containers \(Cluster Nodes\) - Using Quick Start Script](#) for instructions on how to use the script.

Note: The above example expects a valid sdv-vol, default docker network bridge works on your Docker host, and that you already have a valid SDV Docker image that exists.

It is also expected that you reuse your SDV Container volume, otherwise your base port and any other SDV metadata changes will be lost. Basically, if you specify a new SDV volume, then the SDV Container will create a brand new, default SDV Container based on your SDV Docker image defaults (i.e. default base port and server settings).

Change SDV Server Memory Setting

To change the SDV Docker container's memory setting you will need to reconfigure SDV and create a new SDV Container.

Example (change server memory from 8192 Mbytes (8 GB) to 16,384 MBytes (16 GB)):

1) Use SDV Studio (same version of SDV as your SDV Docker container) to connect to your SDV Docker container. Login to the SDV Studio using the SDV admin user “admin” and your SDV admin password along with the appropriate SDV base port.

2) Go to Administration > Configuration

3) Search for “Total Available Memory (“

4) In the search results, select “Total Available Memory (On Server Restart): 4096 Mbytes”.

5) Change your base port value to “8192”. Apply your changes.

Note: “Total Available Memory (Current)” shows your current SDV Server memory configuration.

“Total Available Memory (On Server Restart)” shows your future SDV Server memory configuration. This only takes affect when you restart the SDV Server though.

6) Click on the OK button to close the pop-up window.

7) Logout of the SDV Studio client.

8) Stop the SDV Docker container where your SDV Server is running

```
docker stop <sdv-container-name>
```

9) Now start a new SDV Docker container with the new server memory value (for example, to change from 8GB to 10GB):

```
./run_sdv_container.sh -m 10 --env SDV_ADMIN_PASSWORD=admin1
```

Note: Running the run_sdv_container.sh script in verbose mode (-v) will display the low level docker run commands which will be executed to achieve the result. Refer to [Launching SDV Containers \(Cluster Nodes\) - Using Quick Start Script](#) for instructions on how to use the script.

Note: The above example expects a valid sdv-vol, default docker network bridge works on your Docker host, and that you already have a valid SDV Docker image that exists.

It is also expected that you reuse your SDV Container volume, otherwise your server memory and any other SDV metadata changes will be lost. Basically, if you specify a new SDV volume, then the SDV Container will create a brand new, default SDV Container based on your SDV Docker image defaults (i.e. default "admin" user password, base port and server settings).

Configure External Volume For Local Persisted File Data Sources

To allow your SDV Docker container to introspect and query data from a locally persisted flat file data sources (For example .csv, .xml, .txt files), you will need to transfer those files into your SDV Docker container's volume.

Example (introspect a flat file csv file stored on the SDV Container's volume):

```
sudo cp <flat-file-csv> /var/lib/docker/volumes/<sdv-container-volume-name>/_data/SDV_Server_<sdv-version>/tmp
```

Note: This example expects a valid sdv-vol, default docker network bridge that works on your Docker host, and that you already have a valid SDV Docker Server container that exists and is running.

See [References](#): for more details regarding <sdv-container-volume-name>.

1. `docker exec -it <sdv-server-container-name> ls -al SDV*/tmp/* .csv # validate SDV container can see the new file.`
2. Use SDV Studio Client to introspect and query new csv file.
 - Go to File -> New -> Data Source -> File-Delimited
 - Provide "name", select "Local File System" with "Root Path" /opt/Spotfire/SDV_Server_<sdv-version>
 - Leave all other settings with the default values.
 - Click "Create & Introspect" button.
 - Open "name" data source. Click on "Show Contents" to query data in csv file data source.

Configure Data Source With 3rd party JDBC Driver (type 4)

SDV Data sources may require 3rd party JDBC drivers (type 4).

This section is to cover how to install such drivers in your SDV Docker container.

Example (install Oracle 3rd party JDBC type 4 driver for Oracle 11g):

1. Find the latest Oracle 12g JDBC driver (type 4) drivers (e.g. ojdbc10.jar and xdb.jar).

See the Oracle Adapter Guide for details on where to get this driver and how to configure it for your SDV Container.

2. Stop the SDV Docker container where your SDV Server is running

```
docker stop <sdv-container-name>
```

3. Install Oracle JDBC type 4 drivers in your SDV Server Container

```
sudo cp ojdbc10.jar /var/lib/docker/volumes/<sdv-container-volume-name>/_data/SDV_
Server_<sdv-version>/conf/adapters/system/oracle_19c_thin_driver

sudo cp xdb.jar /var/lib/docker/volumes/<sdv-container-volume-name>/_data/SDV_
Server_<sdv-version>/conf/adapters/system/oracle_19c_thin_driver
```

Note: The above example expects a valid sdv-vol, default docker network bridge works on your Docker host, and that you already have a valid SDV Docker Server container that exists and is not running.

See [References](#): for more details regarding <sdv-container-volume-name>.

4. Start the SDV Docker container where your SDV Server is running

```
docker start <sdv-container-name>
```

5. Validate that the SDV Docker container has the new file.

```
docker exec -it <sdv-container-name> ls -al SDV*/conf/adapters/system/oracle_19c_thin_
driver/
```

6. Check your SDV Docker container server log for acknowledgement that you have installed the JDBC driver for your “Oracle 19c (Thin Driver)” DV adapter.


```
docker exec -it <sdv-container-name> /bin/bash

$ cd SDV*/logs

$ grep -i "Oracle 19c" cs_server.log
```

7. The output of step 6 will show before and after loading of your "Oracle 19c" DV adapter. If the installation was successful, then the DV adapter will have a "loaded" message instead of the following "has not been installed" message (shown below) that was displayed before 3rd party drivers were installed for "Oracle 19c" DV Adapter

```
INFO [main] 2020-03-30 22:07:51.134 +0000 DbUtil - The adapter for 'Oracle 19c (Thin Driver)' has not been installed. For details on adapter installation, see the Installation Guide.
```

A sample message of successful installation:

```
INFO [main] 2020-03-30 17:11:08.222 -0700 JdbcDriverClassLoaderUtil - Adapter: Oracle 19c (Thin Driver) loaded from /opt/Spotfire/SDV_Server_<sdv-version>/conf/adapters/system/oracle_19c_thin_driver
```

8. Once the "loaded" message is seen you can create, introspect and load data from your Oracle 19c DV Adapter.

Configure Data Source With 3rd party JDBC Driver (type 3)

SDV Data sources may require 3rd party JDBC drivers (type 3). This section describes how to install such drivers in your SDV Docker container.

Example (install SAP JCo 3rd party JDBC type 3 driver for linux x64 platforms:

1. Find latest SAP JCo JDBC type 3 driver download from SAP.

See the SDV User Guide Chapter "Configuring Advanced Adapters" section "Installing the SAP Java Connector Library" for more details on where to download SAP JCo connection library and how to install it.

2. Install SAP JCo JDBC type 3 driver (linux x64) from SAP in your SDV Container. Refer to the SDV User Guide "Installing SAP JCo on UNIX" section for more details on how to install the linux x64 version of this driver.

```
sudo cp <sap-cjo-tgz-file> /var/lib/docker/volumes/<sdv-container-volume-name>/_
data/SDV_Server_<sdv-version>/tmp
```

Note: The above example expects a valid sdv-vol, default docker network bridge works on your Docker host, and that you already have a valid SDV Docker Server container that exists and is running.

See [References](#): for more details regarding <sdv-container-volume-name>.

3. Stop your SDV container.
4. Run the following commands:

```
$ cd SDV_Server_<Version>

$ cp sapjco<version>.jar apps/dlm/app_ds_[sap|sapbw|sapbwbox]/lib

$ cp libsapjco<version>.so apps/server/lib/svn/lib64
```

5. Restart your SDV container
6. Run the following commands:

```
$ docker exec -it <SDV container name> /bin/bash

$ cd SDV_Server_<version>

$ export LD_LIBRARY_PATH=apps/server/lib/svn/lib64

$ ./jdk/bin/java -jar apps/dlm/app_ds_[sap|sapbw|sapbwbox]/lib/sapjco<version>.jar
```

This command should have no errors.

7. Now you are ready to introspect your SAP datasource.

Configure JVM Settings from Docker

This section describes how to review existing JVM settings and add custom settings for SDV Server in the docker environment.

To Review Existing Settings

1. List the contents of the files “script_env.sh”. This file exists if you added custom JVM options to your SDV Server.

```
cat conf/script_env.sh
```

2. Search the default Linux JVM arguments set for the SDV Server in the “server.properties” file.

```
cat conf/server/server.properties | grep "linux.vargs"
```

To Add Custom JVM Settings for SDV Server

Follow these steps to add custom JVM Settings::

Navigate to<SDV_INSTALLDir>/conf

```
$ cd ./<SDV_INSTALL_DIR>/conf
```

Copy script_env.sh.sample to script_env.sh

```
$ cp script_env.sh.sample script_env.sh
```

Open script_env.sh with a text editor (nano) and uncomment the last two lines:

```
$ nano script_env.sh
```

```
# CIS_SERVER_VM_ARGS=
```

```
# export CIS_SERVER_VM_ARGS
```

Modify the script_env_sh file to have a content like this at the end of the file:

```
CIS_SERVER_VM_ARGS="-Djavax.net.debug=all -Ddrill.java.home=/data/opt/Spotfire/<SDV_INSTALL_DIR>/jdk -server -XX:+HeapDumpOnOutOfMemoryError -XX:HeapDumpPath=/data/opt/Spotfire/<SDV_INSTALL_DIR>/logs -XX:+UseG1GC -XX:MaxGCPauseMillis=200 -Xmx2048m -Djava.library.path=/data/opt/Spotfire/<SDV_INSTALL_DIR>/apps/server/lib/svn/lin64"
```

```
export CIS_SERVER_VM_ARGS
```

Note: You can get the content for `CIS_SERVER_VM_ARGS` from the file `<SDV_INSTALL_DIR>/conf/server/ server.properties`.

Configure LDAP Properties From Docker

Query searches for retrieving user and group information are controlled by the `ldap` properties file. If you add LDAP domains to SDV Server, you should configure the `ldap.properties` file after installation and prior to adding and configuring the LDAP domain on the Studio Manager Domain Management page. In the docker environment the properties file can be accessed from the following location:

```
$ docker exec -it <CONTAINER_NAME> /bin/bash
```

```
$ cd $SDV_INSTALL_DIR/conf/server
```

Use the Nano text editor for editing the `ldap.properties` file:

```
$ nano ldap.properties
```

For more information on the structure of the properties file, refer to the *Administration Guide*, Chapter *LDAP Domain Administration*.

Configure Truststore and Keystore Files From Docker

Keystore and truststore files are the places where keys for secure communications are stored. Each system component participating in SSL communication requires:

- A keystore file (`cis_server_keystore.jks`) for its own key, which it furnishes to any other component that requests that it authenticate itself.
- A truststore file (`cis_server_truststore.jks`) for the keys of each other component that it trusts and needs to authenticate

The files can be accessed from the following location. Use the Nano text editor for editing the truststore and keystore files:

```
$ docker exec -it <CONTAINER_NAME> /bin/bash
```

```
$ cd $SDV_INSTALL_DIR/conf/server/security
```

```
$ nano <filename>
```

For more information on the SSL Protocol and the Keystore and Truststore files, refer to the *Administration Guide*, Chapter *SDV and SSL Authentication*.

Limitations

While running the SDV Server using the Docker, the monitor daemon is not active. Therefore you cannot use SDV Studio or the command line tool to restart SDV Server. You will have to stop the container and start it again to do so.

Upgrading to a New Version of SDV Server

In order to upgrade your SDV Server to a newer version, follow these steps:

1. Perform a Backup Export of Metadata from the previous version of SDV Server. Refer to the Administration Guide for the instructions on how to perform this using the command line utility.
2. Create a New container using the newer version provided. Refer to the sections in this chapter to build and launch SDV containers.
3. Do a Backup Import of the Metadata into your newly built SDV container instance. Refer to the Administration Guide for the instructions on how to perform this using the command line utility.

Useful Docker Commands for SDV Containers

To check details of all the Docker containers.	<code>\$ docker ps</code>
To check the Docker SDV container system logs.	<code>\$ docker logs <sdv-container-name></code>
Installing the SAP Java Connector Library	<code>\$ docker exec -it <sdv-container-name> /bin/bash</code>
To review the SDV Server logs in an interactive shell.	<code>\$ docker exec -it <sdv-container-name> ls -al SDV*/logs</code>
To stop the container.	<code>\$ docker stop <sdv-container-name></code>
To start the container.	<code>\$ docker start <sdv-container-name></code>
To see a detailed information on Docker objects. such as docker images, containers, networks, volumes, plugins, etc.	<code>docker inspect [OPTIONS] NAME ID [NAME ID...]</code> Refer https://docs.docker.com/engine/reference/commandline/inspect/ for more information.
A quick reference for using Nano text editor.	https://www.nano-editor.org/dist/latest/cheatsheet.html

SDV Container Orchestration Using Kubernetes

Introduction to Container Orchestration

Container orchestration is a procedure used to manage the lifecycles of containers, especially in large, dynamic environments. Many organizations use container orchestration to control and automate tasks such as:

- Provisioning and deployment of containers
- Allocation of resources between containers
- Load balancing of service discovery between containers
- Configuration of an application in relation to the containers running it
- Scaling up or removing containers to spread application load evenly across host infrastructure

Spotfire® Data Virtualization Container distribution is based on an orchestration framework and is scalable, lightweight, and supports widely accepted industry standards. You can deploy, monitor, and manage the application using Kubernetes-based orchestration framework provided by SDV.

SDV Container Orchestration Architecture

The following image shows the SDV Server deployment in a Kubernetes environment:

Prerequisites

The Spotfire® Data Virtualization Container distribution comes with a Helm chart to simplify the deployment of SDV. This section describes all the prerequisites you need before deploying a SDV application.

1. Ensure that your Kubernetes cluster version (i.e. server version) is 1.21 (or later), helm client is version 3.4 (or later), and any OCI compliant container runtime supported by Kubernetes:
<https://kubernetes.io/docs/setup/production-environment/container-runtimes/>
 SDV uses Docker for development/testing of the SDV container images.
2. SDV Container and Docker distribution is supported on Ubuntu 24.04.4 LTS (SDV Server) and Debian 13.4 (SDV Repository and SDV Cache)
3. The Java version supported is 21.0.10 LTS.
4. The Postgres version supported is 17.9.
5. Refer to [Sizing Guidelines for SDV](#) for sizing recommendations and accordingly choose your environment.
6. You will need a container registry url, username and password that can be accessible by your Kubernetes cluster to retrieve the SDV container image. It is required to pre-build the SDV container image for this distribution before you deploy any SDV applications.

Pre-Configuration of the Runtime Environment (On-Premises/Private Cloud only):

Follow these pre-configuration steps to setup your runtime environment.

1. Download the following files from edelivery and unzip the contents to your instance where you intend to build the image:
 - SPOT_sdv_<version>_kubernetes.zip - contains SDV helm charts (SPOT_sdv_<version>_helmcharts.tar.gz) needed for the Spotfire® Data Virtualization Container distribution
 - SPOT_sdv_<version>_docker.zip - contains the scripts, YAML files and templates that are needed to build and launch the images.

- SPOT_sdv_<version>_kubernetes.zip.md5 and SPOT_sdv_<version>_docker.zip.md5 files.
2. Run the md5 checksum in order to validate the integrity of the downloaded files:

```
For example:
$ md5sum SPOT_sdv_8.9.0_kubernetes.zip 072c7444a4bbf858cd627f2ccc00aa91
SPOT_sdv_8.9.0_kubernetes.zip

$ grep SPOT_sdv_8.9.0_kubernetes.zip SPOT_sdv_8.9.0.md5
072c7444a4bbf858cd627f2ccc00aa91 SPOT_sdv_8.9.0_kubernetes.zip
```

Building the Container Image

Refer to the section [Building SDV Docker Images](#) for instruction on how to build a SDV image.

Configuring Kubernetes Using Quick Start Script

The Quick Start script shipped with SDV can be used for configuring Kubernetes for Private/Public cloud.

Follow these steps to run the SDV script to launch a SDV application (i.e. Single or Cluster mode) using the quick start script:

1. SDV is packaged with quick start shell scripts and YAML files that you will need to configure Kubernetes. Extract the SPOT_sdv_<version>_kubernetes.zip file which contains the quick start scripts and SDV Helm charts. For more details refer to the section [Spotfire® Data Virtualization Container Distribution](#).
2. Run the script run_sdv_helmchart.sh (which is part of the SPOT_sdv_<version>_kubernetes.zip file) on your Kubernetes master node. Run the following command:

```
./run_sdv_helmchart.sh [-hv] [--dry-run] [--skip-wait] [-d] [--del-pvc] [--del-reg-secret] [--deployment-environment < aks | eks | private-loadbalancer | private-nodeport >] [--docker-server <host or url>] [--docker-username <user>] [--docker-password <password>] [--docker-email <email>] [--name <name>] [--set <key1>=<val1>,<key2>=<val2>,...]
```

The table below gives a description of the different parameters used with the script.

Parameter	Description	Comments
--dry-run	Show output of execution without actually executing the script.	
--skip-wait	Skip wait check for SDV pods.	
-h	Help for using the script.	
-v	Enable verbose mode output.	
Deletion Settings		
-d	Delete and uninstall SDV helmchart if already exists with same --name <name>.	
--del-pvc	Delete SDV Dynamic PVC storage.	
--del-reg-secret	Delete SDV registry secret (default name "sdv-registry-secret")	
Deployment Settings		
--deployment-environment	Values can be aks, eks, private-loadbalancer or private-nodeport	
Docker Registry Settings		
--docker-server	Server location for Docker registry.	This is required. If not provided, an error will be displayed.

Parameter	Description	Comments
--docker-username	Username for Docker registry authentication.	This is required. If not provided, an error will be displayed.
--docker-password	Password for Docker registry authentication.	This is required. If not provided, an error will be displayed.
--docker-email	Email for Docker registry.	This is required. If not provided, an error will be displayed.
Helm Chart Settings		
--name	Set <name> for helmchart (default is "sdv").	
	Pass SDV helm chart parameters from a yaml file.	
--set <key1>=<val1>,<key2>=<val2>,...	Pass SDV helm chart parameters to helm chart for runtime configuration.	This option is used to configure several runtime settings such as setting the sdv image, specifying list of external IP source ranges, etc. It is recommended that you run "run_sdv_helmchart.sh -h" for a complete list of the various configuration settings that can be tuned using this option.

Required SDV runtime settings for all deploymentEnvironment values:

1) `--set sdvRepo.image=<image path>:<image tag> # sdv repo image location`

`sdvCache.image=<image path>:<image tag> # sdv cache image location`

`sdvServer.image=<image path>:<image tag> # sdv server image location`

Required SDV runtime settings only for deploymentEnvironment="private-loadbalancer":

1. `--set service.externalIPs="{<value1>},{<value2>},...,{<value n>}"` - To specify a list of externalIP values to route requests the the SDV haproxy service. Example:
`service.externalIPs="{100.20.30.20}"`.
2. `--set service.loadBalancerSourceRanges="{<value1>},{<value2>},...,{<value n>}"` - To specify a list of external IP source ranges to limit SDV haproxy service access. List source ranges must match the externalIP list subnets. Example:
`service.loadBalancerSourceRanges="{100.20.30.0/12}"`.

Optional SDV runtime settings:

1. `--set replicas=<value>` - This is an optional parameter to change SDV pod replicas number (positive number > 0). Default value for replicas is 3.
2. `--set sdvAdminPassword=<password>` - To set the SDV admin password

Sample Values YAML file

SDV is pre-packaged with a yaml file that contains the parameters that is needed to install a helm chart successfully. When running the quick start script the default values defined in this file are used. To override the default values, you can use the `--set` flag when the running the script `run_sdv_helmchart.sh`. To help you create an appropriate Values YAML file for different deployment environments, there are 4 sample files provided by SDV (eks-sample, aks-sample, private-loadbalancer-sample, private-nodeport-sample).

Deployment Environment Setting

The deployment-environment parameter file indicates whether the deployment is for a private (on-premises) Kubernetes installation or for a public cloud environment (example: "aks" for Azure AKS). When deployment-environment is set as "private-loadbalancer", the following values must be set:

1. `--set service.externalIPs="{<value1>},{<value2>},...,{value n>}"` # Specify a list of externalIP values to route requests the the SDV haproxy service. Example:
`service.externalIPs="{100.20.30.20}"`.
2. `--set service.loadBalancerSourceRanges="{<value1>},{<value2>},...,{value n>}"` # Specify a list of external IP source ranges to limit SDV haproxy service access. List source ranges must match the externalIP list subnets. Example:
`service.loadBalancerSourceRanges="{100.20.30.0/12}"`.

Replicas

When running the script, set this option to a number greater than 0. This is the number of SDV pod replicas. The default value is 3. To run the container on a Single node, set this value to 1. To run it on a cluster node set this value to a number greater than 1. (for example, `--set replicas=3`)

Application Memory Setting

You can change the resource request memory setting for the SDV Pod by tuning the following options:

Use the `sdvServer.resources.requests.memory` setting to set the memory for the SDV Pod container. Use the `sdvServer.sdvMaxMemory` to set the memory for SDV processes inside the pod. By default, 8GB is the memory set for the SDV Pod container and 7000 (7GB) for the SDV processes inside the Pod.

Deploying SDV

This section describes how to deploy SDV as a cluster mode using the quick start script.

Refer to the section [SDV Container Orchestration Architecture](#) to understand the implementation of SDV Container Orchestration.

Example to run a SDV instance as a Cluster Mode

Run the following command to run SDV as a Cluster Mode:

```
./run_sdv_helmchart.sh -d --del-pvc --deployment-environment private-loadbalancer --set
sdvRepo.image=Spotfire/sdvrepo:8.9.0,sdvCache.image=Spotfire/sdvcache:8.9.0,
sdvServer.image=Spotfire/sdv:8.9.0, service.externalIPs="
{100.20.30.20}",service.loadBalancerSourceRanges="{100.20.30.0/12}",replicas=3
```

You will see a log like the one below, indicating the number of pods that will be starting, the number of worker nodes in the cluster, the name of the worker node, and a series of elapsed times. When the pod is ready you will see a message indicating the which pod has started to run.

```
[sdcv02@na2devdvsjk12 20230711]$ ./run_sdv_helmchart.sh --deployment-environment private-loadbalancer --set sdvRepo.image=redocker.tibco.com/tdv/tdv/qa/release-8.8.0/tdvrepo:8.8.0rc1,tdvCache.image=redocker.tibco.com/tdv/engr/tdv/qa/release-8.8.0/tdvcache:8.8.0rc1,tdvServer.image=redocker.tibco.com/tdv/engr/tdv/qa/release-8.8.0/tdv:8.8.0rc1,service.externalIPs="{10.69.34.22}",service.loadBalancerSourceRanges="{10.69.34.0/12}"
WARNING: version difference between client (1.26) and server (1.24) exceeds the supported minor version skew of +/-1
WARNING: version difference between client (1.26) and server (1.24) exceeds the supported minor version skew of +/-1
NAME: tdv
LAST DEPLOYED: Mon Jul 31 19:53:42 2023
NAMESPACE: tdv
STATUS: deployed
REVISION: 1
TEST SUITE: None

Request to start "3" TDV server pod(s).
Available worker nodes: "3"
..... (30s elapsed)
..... (60s elapsed)
..... (90s elapsed)
.....
TDV pod "tdvserver-1" is now running after 95s.

Waiting for TDV Server pod "tdvserver-2" to start.
..... (30s elapsed)
..... (60s elapsed)
.....
TDV pod "tdvserver-2" is now running after 85s.

TDV pod information
tdvcache-0 na2devdvsjk83 <none>
tdvcache-1 na2devdvsjk82 <none>
tdvcache-2 na2devdvsjk83 <none>
tdvrepo-0 na2devdvsjk83 <none>
tdvrepo-1 na2devdvsjk82 <none>
tdvrepo-2 na2devdvsjk83 <none>
tdvserver-0 na2devdvsjk83 <none>
tdvserver-1 na2devdvsjk82 <none>
tdvserver-2 na2devdvsjk83 <none>

TDV service information
tdvcache ClusterIP None <none> 9484/TCP 3652s cluster-tdv,service-tdvcache,version=8.8.0
tdvrepo ClusterIP None <none> 9488/TCP 3652s cluster-tdv,service-tdvrepo,version=8.8.0
tdvserver ClusterIP None <none> 9389/TCP,9381/TCP,9382/TCP,9383/TCP,9384/TCP,9385/TCP,9386/TCP,9489/TCP,9481/TCP,9482/TCP,9483/TCP,9485/TCP,9486/TCP,9487/TCP,9489/TCP 3652s cluster-tdv,service-tdvserver,version=8.8.0
```

Run the following command to check the status of a specific pod:

```
$ kubectl describe pod <pod name> -n sdv
```

After you see the output of `run_sdv_helmchart.sh` indicating that the pods and services are ready, then you can use the external IP to access your SDV service. If you want to setup a monitor external IP for one of the SDV pods in your SDV cluster then you have to run `run_sdv_haproxy.sh` to create a service that exposes your SDV pod for access with an external IP. Note that you can only setup one SDV HAProxy service at a time because the service is tied to the SDV pod and requires a unique external IP for its service. Refer to section [Installing the SDV HAProxy Service using the Quick Start Script](#) for instructions on using this script `run_sdv_haproxy.sh`.

Note: To deploy SDV as a Single node instance set the parameter `replicas=1` when running the `run_sdv_helmchart.sh`

```
./run_sdv_helmchart.sh -d --del-pvc --deployment-environment private-loadbalancer --set
sdvRepo.image=Spotfire/sdvrepo:8.9.0,sdvCache.image=Spotfire/sdvcache:8.9.0,
```

```
sdvServer.image=Spotfire/sdv:8.9.0, service.externalIPs="
{100.20.30.20}",service.loadBalancerSourceRanges="{100.20.30.0/12}",replicas=1
```

Scaling SDV Pods

Many organizations use container orchestration to control and automate tasks such as scaling up or removing containers to spread application load evenly across host infrastructure. You can do that in SDV by editing the Values YAML file.

Example to Scale Up From 2 to 3 nodes

1. Run the following command to start sdv cluster with replicas=2 with run_sdv_helmchart.sh (private loadbalancer):

```
./run_sdv_helmchart.sh -d --del-pvc --deployment-environment private-loadbalancer --set
sdvRepo.image=Spotfire/sdvrepo:8.9.0,sdvCache.image=Spotfire/sdvcache:8.9.0,
sdvServer.image=Spotfire/sdv:8.9.0, service.externalIPs="
{100.20.30.20}",service.loadBalancerSourceRanges="{100.20.30.0/12}",replicas=2
```

2. Run the following command to get the current helm chart values:

```
helm get values -n sdv sdv > current_values.yaml
```

3. Open the yaml file and edit the replicas number to 3.

```
vi current_values.yaml
```

4. Run the following command to upgrade the helm chart to the new replica:

```
helm upgrade -n sdv -f current_values.yaml sdv ./sdv
```

5. After waiting for a few minutes, you will have 3 SDV nodes in your SDV cluster.

```
kubectl get pods -n sdv
```

Example to Scale Down from 3 to 2 nodes (i.e. stop one SDV node without deleting it)

1. You already have a 3 SDV node cluster running so ensure all 3 are running and

that your system is in a quiet period (i.e. no queries running, no scheduled cache refreshes, no data movement operations are going on).

2. Run the following command to get the current helm chart values:

```
helm get values -n sdv sdv > current_values.yaml
```

3. Open the yaml file and edit the replicas number to 2.

```
vi current_values.yaml
```

4. Run the following command to upgrade the helm chart to the new replica:

```
helm upgrade -n sdv -f current_values.yaml sdv ./sdv
```

5. After waiting for a few minutes you will have 2 SDV nodes in your cluster. You will see that the 3rd SDV node is terminated, but the volume still exists. The 3rd SDV node's storage (i.e. persistent volume) will still exist because it was not deleted.

```
kubectl get pod -n sdv
```

```
kubectl get pvc -n sdv
```

Note: Scaling down SDV pods could impact outstanding SDV workloads and user requests during this operation. Please take care in running this operation during a quiet period on your system. You can tune the system wait time for gracefully terminating the pod using the option `terminationGracePeriodSeconds` in the values.yml file (as well as the sample files provided for other distributions).

Installing the SDV HAProxy Service using the Quick Start Script

The Quick Start script shipped with the SDV Helm Charts archive can be used for installing a SDV HAProxy service.

Run the script `run_sdv_haproxy.sh` (which is part of the `SPOT_sdv_<version>_kubernetes.zip` file) on your Kubernetes master node. Refer to the section [Spotfire® Data Virtualization Container Distribution](#) for a list of files that are pre-packaged with SDV.

Run the following command:

```
./run_sdv_haproxy.sh [-hv] [--dry-run] [--skip-wait] [-d] [--del-secret] [--name <name>] [--set <key1>=<val1>,<key2>=<val2>,...]
```

The table below gives a description of the different parameters used with the script.

Parameter	Description	Comments
--dry-run	Show output of execution without actually executing the script.	
--skip-wait	Skip wait check for SDV pods.	
-h	Help for using the script.	
-v	Enable verbose mode output.	
Deletion Settings		
-d	Delete and uninstall SDV HAProxy helmchart if already exists with same --name <name>.	
--del-secret	Delete gateway secret for SDV HAPRoxy service (default name sdv-registry-secret)	
Helm Chart Settings		
--name	Set <name> for helmchart (default is "sdv-haproxy").	
	Pass SDV helm chart parameters from a yaml	

Parameter	Description	Comments
	file.	
--set <key1>=<val1>,<key2>=<val2>,...	Pass SDV helm chart parameters to helm chart for runtime configuration.	It is recommended that you run "run_sdv_haproxy.sh -h" for a complete list of the various configuration settings that can be tuned using this option.

Note: You can only run one SDV HAProxy service at a time.

Removing a SDV application

To remove a SDV application completely, you must do the following:

1. Uninstall the HAProxy Service
2. Uninstall SDV application
3. Uninstall Persistent Volume Claims
4. Delete the SDV registry Secret

To Uninstall HAProxy Service

Run the following command to lists all of the releases in sdv namespace:

```
helm ls -n sdv
```

Run the following command to uninstall the SDV HAProxy service. Run this command only if you deployed the SDV HAProxy application via run_sdv_haproxy.sh.

```
helm uninstall -n sdv <sdv release name> #for example, helm uninstall -n sdv sdv-haproxy-sdvserver-1
```

Uninstall Application

To uninstall the SDV applications you have deployed:

1. Run the following command to list all the SDV applications:

```
helm ls -n sdv
```

An output similar to the following is displayed:

NAME CHART	NAMESPACE APP VERSION	REVISION	UPDATED	STATUS
sdv	sdv 1	2023-08-01 20:32:38.661939539	-0700 PDT	deployed
sdv-helm-1.1.0	8.9.0			
sdv-haproxy-sdvserver-1	sdv 1	2023-08-01 20:37:55.529521313	-0700 PDT	deployed
sdv-haproxy-1.0.0				

2. Run the following command to remove the SDV application(s) you have deployed:

```
helm uninstall -n sdv sdv-haproxy-sdvserver-1
```

An output similar to the following is displayed:

```
release "sdv-haproxy-sdvserver-1" uninstalled
```

Uninstall the Persistent Volume Claim

The dynamical volumes will not be removed automatically. You will need to manually remove the dynamically provisioned volumes. To do this follow the steps below:

1. Run the following command to list all persistent volume claims in sdv namespace:

```
kubectl get pvc -n sdv
```

An output similar to the following is displayed:

NAME	STATUS	VOLUME	CAPACITY	ACCESS MODES
sdvserver-vol-sdvcache-0	Bound	pvc-26f478d6-8a08-49db-90ab-29f54cedd5af	5Gi	
sdvserver-vol-sdvcache-1	Bound	pvc-6606860a-273a-49dc-b68e-c9327a6a6433	5Gi	

```

RWO      local-path  13m
sdvserver-vol-sdvrepo-0  Bound  pvc-17c7d00c-2933-43c8-8116-8257dfee6061  5Gi
RWO      local-path  13m
sdvserver-vol-sdvrepo-1  Bound  pvc-15cc6c8e-3294-4fa5-ac36-e97cb5a08a62  5Gi
RWO      local-path  13m
sdvserver-vol-sdvserver-0  Bound  pvc-fa32e417-8e44-4a62-9dec-d53a22f85fce  20Gi
RWO      local-path  13m
sdvserver-vol-sdvserver-1  Bound  pvc-99fedcc0-e541-4095-9301-558f73d39b59  20Gi
RWO      local-path  12m

```

2. Run the following command(s) to remove the SDV application(s) you have deployed:

```
kubectl delete pvc -n sdv sdvserver-vol-sdvcache-0
```

```
kubectl delete pvc -n sdv sdvserver-vol-sdvrepo-0
```

```
kubectl delete pvc -n sdv sdvserver-vol-sdvserver-0
```

Note: You must stop the pod that mounts the pvc before deleting it

Kubernetes cleans up unused images on the Kubernetes nodes so there is no need to manually clean up the image. Refer to the following documentation on this topic:

<https://kubernetes.io/docs/concepts/architecture/garbage-collection/#containers-images>

Delete the SDV Registry Secret

You will finally need to also clean up the SDV Registry Secret. Run the following command to do this:

```
kubectl delete secret sdv-haproxy-gateway-certs-secret -n sdv # delete the SDV registry secret
```

Running SDV on a Public Cloud - Microsoft Azure

The Spotfire® Data Virtualization Container distribution provides a distributed platform for containerized applications. In this section you will find detailed instructions on how to

run SDV on the Microsoft Azure Public Cloud platform.

For a basic understanding of the Azure Kubernetes Service (AKS) see:

<https://docs.microsoft.com/en-us/azure/aks/concepts-clusters-workloads>

In order to interact with Azure APIs an AKS cluster requires either an Azure Active Directory service principal or a managed identity. You will need this to manage Azure resources such as the load balancer, container registry, etc. You can access the Azure portal using your account information at

<https://portal.azure.com>

Creating a Container Registry

After you login to the Azure portal, follow these steps to set up a Resource Group. A resource group is an area for all your AKS resources such as Network Security Group, Virtual Network, Storage account (used to deploy helm charts), Container Registry (where the SDV image is stored) and a Kubernetes Cluster.

To create a Container Registry,

1. Login to the Azure portal using your account credentials.
2. Choose your Resource Group.
3. Click on the "Create" button from your Resource Group Page.
4. Choose Container Registry.
5. In the New Container Registry dialog, choose the default settings for Basics, Networking, encryption and Tags. Finally. review the chosen options and click on Create to create your Container Registry.
6. Once the Container Registry is created, push your SDV images to this location. Refer to the chapter [SDV Docker Container](#) for more information on how to build your SDV image and push it to the registry. You may also refer to the following Azure documentation for more details about how to push an image to a container registry.

<https://docs.microsoft.com/en-us/azure/aks/tutorial-kubernetes-prepare-acr?tabs=azure-cli>

Creating a Cluster

Follow these steps to create a Kubernetes cluster in your resource group:

1. From your Resource Group, click on Create and choose the option Kubernetes Service to create a Kubernetes Cluster.
2. Choose your subscription and your resource group if it is not selected already.
3. Provide a name for your Kubernetes cluster.
4. Choose a region that is appropriate for your organization.
5. Choose a Kubernetes version. Currently SDV supports version 1.21.0 or a higher version in 1.21.X.
6. Use the defaults for the worker node settings such as Node size, Scale method and Node count range.
7. Use the default settings for the rest of the Kubernetes Create Cluster dialogue - Node pools, Authentication, Tags.
8. In the Networking tab, choose your Container Registry. Refer to the section [Creating a Container Registry](#) for more information on creating a registry.
9. Finally click on Create to create the cluster.

Deploying the SDV Helm Chart

The SDV Helm chart provisions the necessary storage for the SDV application on the Kubernetes Cluster. Spotfire® Data Virtualization Container distribution includes the helm chart with certain default settings. You can run the helm chart using a pre-built script provided.

You can use an Azure CLI or a Cloud-shell to connect to Azure and execute administrative commands on the Azure resources. These tools allow the execution of commands through interactive command-line prompts or a script. Refer to the following documentation on the Azure CLI and Azure Cloud Shell for more information about using these tools:

<https://docs.microsoft.com/en-us/cli/azure/install-azure-cli>

<https://docs.microsoft.com/en-us/azure/cloud-shell/overview>

To edit/run the helm chart and scripts using one of these tools:

1. Open the Cloud Shell interface or the Azure CLI.
2. Connect to your cluster:
 - Click on the option Connect in the top menu bar of the Cluster. A window is displayed with all the connection details and the associated kubernetes commands to use in your shell.
 - Use the commands in your shell interface to connect to the cluster. (for example, run these commands:

```
$ az account set --subscription <subscription id>
```

```
$ az aks get credentials --resource-group <resource group name> --name <user id>
```

- Once the connection to cluster is established, you can then issue any kubernetes command.

For example, to get information on all the pods running, type the following command:

```
$ kubectl get pods --all-namespaces
```

7. Upload your SDV Helm charts archive file (i.e. SPOT_sdv_<version>_helmchart.tar.gz) to Azure using the Upload option in your Cloud shell. If you are using the Azure CLI, make sure that has access to the files extracted from the SPOT_sdv_<version>_helmchart.tar.gz.
8. Once you have connected to the cluster and have all the files you need, you can use the SDV scripts to install your helm chart. Refer to the section [Configuring Kubernetes Using Quick Start Script](#).
9. Azure AKS uses a load balancer service to route all client requests across all your SDV applications that are ready to fulfill those requests. Azure AKS will automatically assign an external IP for your SDV service endpoint. To get the external IP of that load balancer service run:

```
$ kubectl get svc -o wide -n sdv
```

Note: If you also deployed the SDV HAProxy application, then you will see two SDV services with external IPs. One is for the Client External IP and one for the Monitor External IP (Refer to the illustration [SDV Container Orchestration Architecture](#) for the Cluster SDV Deployment Mode architecture).

The screen below is a sample output of the Azure Cloud Shell displaying the load balancer service and the IP to access.

```
kubect1 get service -o wide -n tdv | grep tdv
tdv-svc          LoadBalancer  10.0.124.134  20.69.84.64  9400:31400/TCP,9401:31401/TCP,9402:31402/TCP,9403:31403/TCP,
9303:31303/TCP  20m  app=tdv
tdv-svc-headless ClusterIP      None         <none>        9300/TCP,9301/TCP,9302/TCP,9303/TCP,9304/TCP,9305/TCP,9306/T
CP,9400/TCP,9401/TCP,9402/TCP,9403/TCP,9407/TCP  20m  app=tdv
tdv-svc-http     ClusterIP      10.0.207.49  <none>        9400/TCP,9402/TCP,9303/TCP
20m  app=tdv
tdv-svc-tcp      ClusterIP      10.0.95.109  <none>        9401/TCP,9403/TCP
20m  app=tdv
stephen@Azure:~/85/k8s$
```

Running SDV on a Public Cloud - Amazon Web Service

The Spotfire® Data Virtualization Container distribution provides a distributed platform for containerized applications.

In this section you will find detailed instructions on how to run SDV on the Amazon Web Service (AWS) Public Cloud platform.

For a basic understanding of the AWS Elastic Kubernetes Service (EKS) see:

<https://aws.amazon.com/eks/>

In order to interact with AWS APIs, your EKS cluster will need the appropriate Identity and Access Management (IAM) roles/permissions to manage AWS resources such as the load balancer, container registry, etc. You can access the AWS console using your account information at:

<https://aws.amazon.com/>

In versions 2.5 and newer, the AWS Load Balancer Controller is the default controller for Kubernetes services. When installing SDV on EKS, you will need to install the AWS Load Balancer Controller. For more information, refer to:

<https://docs.aws.amazon.com/eks/latest/userguide/aws-load-balancer-controller.html>

You will also need the Amazon Elastic Block Store Container Storage Interface Driver to manage the life cycle of the Persistent Volumes. For more information, refer to:

<https://docs.aws.amazon.com/eks/latest/userguide/ebs-csi.html>

Creating a Container Registry

The following link shows how to setup an AWS ECR registry:

<https://docs.aws.amazon.com/AmazonECR/latest/userguide/repository-create.html>

Once the container registry is created, push your SDV images to that location. Refer to the chapter [SDV Docker Container](#) for more information on how to build your SDV image and push it to the registry. You may also refer to the following AWS ECR documentation for more details about how to push an image to a container registry:

<https://docs.aws.amazon.com/AmazonECR/latest/userguide/docker-push-ecr-image.html>

Example:

1. Login to AWS management console with appropriate IAM role/permissions to create an AWS ECR resource.
2. Create a private registry `<ID>.dkr.ecr.<REGION>.amazonaws.com` in region us-west-2 with repository name "Spotfire/sdv".
3. Push SDV images (Spotfire/sdv:8.9.0, Spotfire/sdvrepo:8.9.0, Spotfire/sdvcache:8.9.0) to private registry.

Login to the console on the Docker Server that has the "Spotfire/sdv:8.9.0" image locally.

a. Configure local AWS CLI config files (i.e. `$HOME/.aws/config` and `$HOME/.aws/credentials`) with AWS Access Key ID and AWS Secret Access Key.

b. Run the following commands:

```
$docker login --username AWS -p $(aws ecr get-login-password --region us-west-2)
<ID>.dkr.ecr.<REGION>.amazonaws.com
```

```
$docker tag Spotfire/sdv:8.9.0 <ID>.dkr.ecr.<REGION>.amazonaws.com/Spotfire/sdv:8.9.0
```

```
$docker push <ID>.dkr.ecr.<REGION>.amazonaws.com/Spotfire/sdv:8.9.0
```

4. Validate "Spotfire/sdv:8.9.0" image now exists on

```
https://<REGION>.console.aws.amazon.com/ecr/repositories/private/<ID>/Spotfire/sdv?r
egion=<REGION>
```

Creating a Cluster

Follow these steps to create a Kubernetes cluster in your AWS account.

<https://docs.aws.amazon.com/eks/latest/userguide/getting-started.html>

Example:

1. Login to AWS management console with appropriate IAM role/permissions to create an AWS EKS resource.
2. Select "region".
3. Goto:
<https://<REGION>.console.aws.amazon.com/eks/home?region=<REGION>#/clusters>
4. Click "Add Cluster" -> "Create"
5. Configure the "Configure cluster" info. Provide the required fields: EKS cluster name, Kubernetes version, and Cluster service role.
Note: SDV 8.5+ requires Kubernetes 1.21 or higher.
6. Configure the "Specify networking" info. Provide the required fields: VPC, subnets, security group.
7. Configure the "Configure logging" info (optional).
8. Review and click "create" button.
9. Wait for new cluster to have Status="Active".
10. Cluster name you specified in step 5 will now be "<NAME>_eks_cluster". Click on "<NAME>_eks_cluster".
11. Use the same Cluster service role you provided in step 5.
12. Select "Compute" tab.
13. Click on "Add Node Group".

Node group compute configuration

Select the "Instance type" and Disk size appropriate for each node to run one or more SDV pods. See [Sizing Guidelines for SDV](#) for sizing guidelines for a SDV deployment (In this case a SDV pod that contains a SDV container). For example, Suppose you have 2 nodes in the node group where each node is a 4 cpu, 16GB type with 16GB of storage. This SDV Medium configuration will allow you to run 2 SDV pods per node.

14.Wait for new node group to be in the operational status

15.Add internet gateway to "Route table" for each subnet in VPC.

Note: Without this you won't be able to access your Kubernetes cluster from outside of AWS.

```
0.0.0.0/0  igw-<NUMBER> igw-<NUMBER>
```

16.Add two tags to each subnets in the VPC

```
tag name "kubernetes.io/role/elb", tag value "1"
```

```
tag name "kubernetes.io/cluster/<EKS cluster name>" "shared"
```

Deploying the SDV Helm Chart

To deploy the SDV Helm chart on the AWS Public cloud, follow these steps:

1. Console login to an instance that has AWS CLI, Helm, and Kubernetes clients.
See [Prerequisites](#) for details on the versions of Helm and Kubernetes needed for these clients.
2. Configure local AWS CLI config files (i.e. \$HOME/.aws/config and \$HOME/.aws/credentials) with AWS Access Key ID and AWS Secret Access Key.

```
$ aws ecr get-login-password --region <REGION>
```

```
$ aws eks update-kubeconfig --region <REGION> --name "<NAME>_eks_cluster"
```

```
$ kubectl config current-context arn:aws:eks:<REGION>:<ID>:cluster/<NAME>_eks_
cluster
```

```
$ kubectl get nodes -o wide
```

NAME	STATUS	ROLES	AGE	VERSION	INTERNAL-IP	EXTERNAL-IP	OS-IMAGE
			KERNEL-VERSION	CONTAINER-RUNTIME			
ip-172-31-24-119.<REGION>.compute.internal	Ready	<none>	47m	v1.21.5-eks-9017834			
172.31.24.119	34.221.117.5	Amazon Linux 2	5.4.181-99.354.amzn2.x86_64	docker://20.10.7			
ip-172-31-55-142.<REGION>.compute.internal	Ready	<none>	47m	v1.21.5-eks-9017834			

```
172.31.55.142 35.81.76.37 Amazon Linux 2 5.4.181-99.354.amzn2.x86_64 docker://20.10.7
```

```
$ tar xvpfz SPOT_sdv_<sdv_VERSION>_helmchart.tar.gz . # This comes from eDelivery (SPOT_sdv_<sdv_version>_kubernetes.zip).
```

Example to run 3 node SDV cluster in EKS with default cpu, memory, and storage specified by eks-sample.yaml for SDV 8.9.0.

(sample bash script):

```
#!/bin/bash
```

```
SDV_VERSION=8.9.0
```

```
REPLICAS=3
```

```
REPO="<ID>.dkr.ecr.<REGION>.amazonaws.com/Spotfire/sdv"
```

```
SDV_IMAGE_TAG=${SDV_VERSION}
```

```
DOCKER_SERVER=<ID>.dkr.ecr.<REGION>.amazonaws.com
```

```
DOCKER_USER=aws
```

```
DOCKER_PASSWORD=`aws ecr get-login-password --region us-west-2`
```

```
DOCKER_EMAIL=xyz@abc.com
```

```
./run_sdv_helmchart.sh -d --del-pvc --deployment-environment eks --set  
sdvRepo.image=spotcosdvrepo:8.9.0,sdvCache.image=spotcosdvcache:8.9.0,  
sdvServer.image=spotcosdvrepo:8.9.0,replicas=$REPLICAS
```

Example to run 3 SDV pod cluster in EKS to override resources like cpu or memory in SDV 8.9.0.

(sample bash script):

```
#!/bin/bash
```

```
SDV_VERSION=8.9.0
```

```
REPLICAS=3
```

```
REPO="<ID>.dkr.ecr.<REGION>.amazonaws.com/Spotfire/sdv"
```

```
SDV_IMAGE_TAG=${SDV_VERSION}
```

```
DOCKER_SERVER=<ID>.dkr.ecr.<REGION>.amazonaws.com
```

```
DOCKER_USER=aws
```

```
DOCKER_PASSWORD=`aws ecr get-login-password --region us-west-2`
```

```
DOCKER_EMAIL=xyz@abc.com
```

```
/run_sdv_helmchart.sh -d --del-pvc --deployment-environment eks --set  
sdvRepo.image=Spotfire/sdvRepo:8.9.0,sdvCache.image=Spotfire/sdvCache:8.9.0,  
sdvServer.image=Spotfire/sdv:8.9.0,replicas=$REPLICAS,sdvServer.resource.limits.memory="10  
Gi",  
sdvServer.resource.limits.cpu="2000m"
```

Spotfire® Data Virtualization Container Distribution

This following table lists the files that are part of the Spotfire® Data Virtualization Container distribution.

File Name	Description
SPOT_sdv_<version>_helmchart.tar.gz	The compressed file consists of: csi-sc-standard- This folder contains sample configuration of an Azure storage class. local-path-storage - This folder contains sample configuration for a local storage class that utilizes local storage on your Kubernetes nodes. sdv - This folder contains the Helm chart for the SDV application. There are 4 separate sample yaml files for the different deployment environments (aks, eks, private-loadbalancer and private-endpoint. Copy the contents from one of the deployment environment sample files into the values.yml file). sdv-haproxy - This folder contains the Helm chart for SDV HAProxy application. run_sdv_helmchart.sh - Quick start script for the SDV deployment in Kubernetes. run_sdv_haproxy.sh - Quick start script for the SDV HAProxy deployment in Kubernetes.
SPOT_sdv_<version>_docker.zip	This compressed file contains: SPOT_sdv_<version>_docker.tar.gz, Dockerfile.sdv, Dockerfile.cache, Dockerfile.repo- SDV docker container files. build_sdv_image.sh - Quick start script to build the SDV image. run_sdv_container.sh - Quick start script to launch SDV container as a Single node. docker_compose: This folder contains the templates (<i>docker-compose.yml.j2, haproxy.cfg, sdv-k8s.properties</i>) and scripts (<i>run_sdv_docker_compose, values.yml</i>) needed to run the docker_compose tool to launch the sdv containers.

Useful Kubernetes Commands

The table below lists some commonly used Kubernetes commands

Generic Kubernetes Commands

To list all services running	<pre>\$ kubectl get services</pre>
To list all nodes with more details such as Internal IP.	<pre>\$ kubectl get nodes -o wide</pre>
To list images	<pre>\$ kubectl get pods --all-namespaces -o jsonpath="{.items[*].spec.containers[*].image}" tr -s ' '[:space:]]' '\n' sort uniq -c</pre>
To list all Pods running	<pre>\$ kubectl get pods</pre>
View details about a particular node	<pre>\$ kubectl describe node [node-name]</pre>
View details about a particular pod	<pre>\$ kubectl describe pod [pod-name]</pre>
To print logs from containers in a pod	<pre>\$ kubectl logs [pod-name]</pre>
To install/uninstall helm chart	<pre>helm install/uninstall <chart></pre>
A list of all installed helm charts	<pre>helm list</pre>

Useful SDV-Specific Commands

To list the SDV Pods	<pre>\$ kubectl get pods -n sdv</pre>
To describe SDV Pod.	<pre>\$ kubectl describe pod sdvserver-0 -n sdv</pre>
To list SDV volumes	<pre>\$ kubectl get pvc -n sdv</pre>
To display the log of the init container in the specified pod.	<pre>\$ kubectl logs sdvserver-0 init -n sdv</pre>
To invoke bash command prompt for the specified pod.	<pre>\$ kubectl exec -it sdvserver-0 -c sdv -n sdv -- /bin/bash</pre>
To install Docker Registry Secret	<pre>\$ kubectl create secret sdv-registry-secret -n sdv --docker-server=<DOCKER_SERVER> - -docker-username=<DOCKER_USERNAME> --docker- password=<DOCKER_PASSWORD> -- docker-email=<DOCKER_EMAIL></pre> Note: Name of the secret must be "sdv-registry-secret" otherwise SDV deployment in Kubernetes will not work with the SDV Helm chart.
To Manually Deploy the Local Path Storage class	<pre>\$ kubectl apply -f local-path-storage/local-path-storage.yaml</pre> <pre>\$ kubectl get deployment -n local-path-storage --no-headers</pre>

To Manually Deploy the Azure csi-standard-cs Storage class

```
$ kubectl apply -f csi-sc-standard/sc-azure-disk.yaml
```

```
$ kubectl get storageclass
```

Note: This storage class is only used when the deployment environment value is aks.

To install HAProxy Certificate Secret

```
openssl req -x509 -nodes -days 365 -newkey rsa:2048 -out gateway.crt -keyout gateway.key -subj "/CN=sdvServer.haproxy/O=sdv-haproxy-gateway-certs-secret"
```

```
cat gateway.crt gateway.key | tee gateway.pem
```

```
kubectl create secret generic $sdv-haproxy-gateway-certs-secret --from-file=gateway.pem=./gateway.pem -n sdv
```

Note: Name of the secret must be "sdv-haproxy-gateway-certs-secret" otherwise SDV deployment in Kubernetes will not work with the SDV Helm chart.

SDV Deployment Examples

Private Cloud (On-Premises) Deployment

SDV Single example with default replicas=3 and default memory and cpu

```
./run_sdv_helmchart.sh --set deployment-environment private-loadbalancer,
sdvRepo.image=<REPO_URL>,sdvCache.image=<CACHE_URL>,sdvServer.image=<SERVER_URL>,service.externalIPs="{100.20.30.20}",service.loadBalancerSourceRanges="{100.20.30.20/12}",replicas=1
```

SDV Cluster example with 3 replicas, custom memory, cpu, and storage settings

```

$. /run_sdv_helmchart.sh --set sdvServer.cluster.externalIPs="{100.20.30.20}",
sdvServer.cluster.loadBalancerSourceRanges="{100.20.30.0/12}",
sdvRepo.image=<REPO_URL>,sdvCache.image=<CACHE_URL>,sdvServer.image=<SERVER_
URL>,
sdvServer.resources.requests.cpu="2000m",
sdvServer.resources.requests.memory="16Gi",
sdvServer.volumeSize="16Gi",replicas=3

```

SDV Single example with default replicas=3 and default memory and cpu - Using Helm command

```

helm install --create-namespace -n sdv sdv ./sdv --set replicas=1,sdvRepo.image=<REPO_
URL>,sdvCache.image=<CACHE_URL>,sdvServer.image=<SERVER_URL>

```

SDV Cluster example with default values and loadBalancer source range settings

```

./run_sdv_helmchart.sh --set sdvServer.externalIPs="{100.20.30.20}",
sdvServer.loadBalancerSourceRanges="{100.20.30.0/12}",
sdvRepo.image=<REPO_URL>,sdvCache.image=<CACHE_URL>,sdvServer.image=<SERVER_
URL>,replicas=3

```

SDV Cluster example with default values - Using Helm command

```

helm install --create-namespace -n sdv sdv ./sdv --set sdvRepo.image=<REPO_
URL>,sdvCache.image=<CACHE_URL>,sdvServer.image=<SERVER_URL>

```

SDV HAProxy example (sdvserver-0 default pod monitoring)

```

$. /run_sdv_haproxy.sh --set deployment-environment aks

```

```

./run_sdv_haproxy.sh -d --set sdvServer.externalIPs="{100.20.30.20}",
sdvServer.loadBalancerSourceRanges="{100.20.30.0/12}"

```

SDV HAProxy example (sdvserver-1 pod monitoring)

```

$. /run_sdv_haproxy.sh -d --set sdvServer.podName="sdvServer-1",sdvServer.externalIPs="
{100.20.30.20}",sdvServer.loadBalancerSourceRanges="{100.20.30.0/12}"

```

Public Cloud Deployment

Public cloud (Azure)

SDV Cluster example with 3 replicas, custom memory, cpu, and storage settings

```
$ ./run_sdv_helmchart.sh --set deployment-environment aks,
storage.storageClassName="csi-sc-standard",sdvRepo.image=<REPO_
URL>,sdvCache.image=<CACHE_URL>,sdvServer.image=<SERVER_URL>,
sdvServer.resources.requests.cpu="2000m",
sdvServer.resources.requests.memory="16Gi",
sdvServer.volumeSize="16Gi",sdvServer.sdvMaxMemory="15000",replicas=3
```

SDV Single example with default replicas=3 and default memory and cpu

```
./run_sdv_helmchart.sh --set deployment-environment aks,
storage.storageClassName="csi-sc-standard",sdvRepo.image=<REPO_URL>,
sdvCache.image=<CACHE_URL>,sdvServer.image=<SERVER_URL>,replicas=1
```

SDV Single example with default replicas=3 and default memory and cpu - Using Helm command

```
helm install --create-namespace -n sdv sdv ./sdv --set deployment-environment aks,
storage.storageClassName="csi-sc-standard",sdvRepo.image=<REPO_
URL>,sdvCache.image=<CACHE_URL>,sdvServer.image=<SERVER_URL>,replicas=1
```

SDV Cluster example with default replicas=3 and default memory and cpu

```
./run_sdv_helmchart.sh --set deployment-environment aks,
storage.storageClassName="csi-sc-standard",sdvRepo.image=<REPO_
URL>,sdvCache.image=<CACHE_URL>,sdvServer.image=<SERVER_URL>,replicas=1
```

SDV Cluster example with default replicas=3 and default memory and cpu - Using Helm command

```
helm install --create-namespace -n sdv sdv ./sdv --set
deployment-environment aks,storage.storageClassName="csi-sc-
standard",sdvRepo.image=<REPO_URL>,sdvCache.image=<CACHE_URL>,
sdvServer.image=<SERVER_URL>,replicas=1
```

SDV HAProxy example (sdvserver-0 default pod monitoring)

```
$.run_sdv_haproxy.sh --set deployment-environment aks
```

SDV HAProxy example (sdvserver-1 pod monitoring)

```
$.run_sdv_haproxy.sh --set sdvServer.podName="sdvserver-1",deployment-environment aks
```

Public cloud (Amazon Web Service)

SDV Cluster example with 3 replicas, custom memory, cpu, and storage settings

```
$ ./run_sdv_helmchart.sh --set deployment-environment eks,
storage.storageClassName="csi-sc-standard",
sdvRepo.image=<REPO_URL>,sdvCache.image=<CACHE_URL>,sdvServer.image=<SERVER_
URL>,
sdvServer.resources.requests.cpu="2000m",
sdvServer.resources.requests.memory="16Gi",
sdvServer.volumeSize="16Gi",sdvServer.sdvMaxMemory="15000",replicas=3
```

SDV Single example with default replicas=3 and default memory and cpu

```
./run_sdv_helmchart.sh --set deployment-environment eks,
storage.storageClassName="csi-sc-standard",sdvRepo.image=<REPO_
URL>,sdvCache.image=<CACHE_URL>,sdvServer.image=<SERVER_URL>,replicas=1
```

SDV Single example with default replicas=3 and default memory and cpu - Using Helm command

```
helm install --create-namespace -n sdv sdv ./sdv --set deployment-environment eks,
storage.storageClassName="csi-sc-standard",
sdvRepo.image=<REPO_URL>,sdvCache.image=<CACHE_URL>,sdvServer.image=<SERVER_
URL>,replicas=1
```

SDV Cluster example with default replicas=3 and default memory and cpu

```
./run_sdv_helmchart.sh --set deployment-environment eks,
storage.storageClassName="csi-sc-standard",
sdvRepo.image=<REPO_URL>,sdvCache.image=<CACHE_URL>,sdvServer.image=<SERVER_
URL>
```

SDV HAProxy example (sdvserver-0 default pod monitoring)

```
$/run_sdv_haproxy.sh --set deployment-environment eks
```

SDV HAProxy example (sdvserver-1 pod monitoring)

```
$/run_sdv_haproxy.sh --set sdvServer.podName="sdvserver-1",deployment-environment eks
```

Installing Optional SDV Products

This topic describes the installation of optional SDV products. These topics are covered:

- [Version Support](#)
- [Installation Requirements](#)
- [Installing an Optional SDV Product](#)
- [Installing the SDV Client Drivers that are Distributed with SDV](#)
- [Importing Resources Defined in an Earlier Release](#)
- [Manage Active Cluster Security](#)

Version Support

SDV supports the versions listed in [Supported Add-On Adapters](#) and [Supported Advanced Data Source Adapters](#).

Installation Requirements

- [Add-On Adapter Installation Requirements](#)
- [Active Cluster Requirements](#)
- [Installing the Advanced Adapters](#)

Add-On Adapter Installation Requirements

Individual adapters have these requirements:

- SAP BW BEx Adapter should be installed on a separate machine from the SAP GUI, to avoid possible conflict between JCo versions. See [Installing the SAP Java Connector Library](#), in the SDV SAP BW Adapter Guide.
- For Salesforce.com and SAP adapter installations, disable User Account Control.

- SAP BW can cause SDV errors similar to:

```
com.compositesw.cdms.webapi.WebapiException: Error [sapbw-2900000]: BAPI_
ODSO_READ_DATA_UC failed: Key figure 0CMPYPTAMT unknown in InfoProvider
0BP_REL
```

To avoid this error, locate and install the patch listed in SAP “Note 1243987 - Extraction from DataStore object fails.” Install this patch, and use the program SAP_RSADMIN_MAINTAIN to set the parameter RSDRI_DS_NEW in the table RSADMIN to ' ' (empty or space).

Note: To upgrade from an earlier version of an adapter, install the new version and then see [Importing Resources Defined in an Earlier Release](#) .

Active Cluster Requirements

This section lists the software and hardware requirements for Active Cluster. All data sources and databases that are supported with this release of SDV are supported by Active Cluster.

- [SDV File Customizations](#)
- [Digital Certificates](#)
- [Supported Platforms](#)
- [Disk Space and Physical Memory](#)
- [Load Balancer Requirements](#)

SDV File Customizations

The data source capability files and LDAP properties file are not automatically synchronized with other machines in the cluster. Therefore, if you customized the ldap.properties file or data source capability files on a SDV Server that will be in a cluster, you need to copy these files manually to all computers that are members of the cluster.

For example, if you modified the external domain configuration file and the data source capability file for DB2, you would need to copy the following files to all computers that are or will become members of the cluster:

```
<SDV_install_dir>/conf/server/ldap.properties
```



```
<SDV_install_dir>/apps/server/apps/dlm/cis_ds_db2/conf/db2.capabilities
```

Digital Certificates

A digital certificate ensures the identity of a particular computer and the data it transmits to another computer. Every server in an Active Cluster must have a digital certificate set up on the computer. A trial digital certificate is shipped with SDV Server but must be changed to ensure full security. See [Updating the Digital Certificate to Secure Cluster Communication](#) for how to do this.

Supported Platforms

See [Installation Requirements and Support Information](#) for a list of the platforms and protocols supported by Active Cluster.

Disk Space and Physical Memory

Active Cluster requires an additional 4 MB of disk space.

Load Balancer Requirements

Although a load balancer is not required to be used with Active Cluster, it is highly recommended to achieve the maximum benefits of using Active Cluster.

Installing an Optional SDV Product

All optional SDV products are installed for you when you perform the SDV Server install.

Before you create or join a cluster, make sure that all cluster members use the same level of encryption.

Installing the Advanced Adapters

To deploy a new adapter, you will need to have a running SDV Server and the new adapter jar file.

Auto Deployment

Follow these steps to deploy the adapter automatically when SDV is restarted:

1. Obtain new adapter (e.g. sdv.<adapter_name>.zip)
2. Copy sdv.<adapter_name>.zip to <SDV_install_dir>/tmp
3. Unzip the sdv.<adapter_name>.zip under <SDV_install_dir>/tmp
4. Copy the <adapter name>.jar file to the folder <SDV_Install_Dir>/packages/autodeploy_ds_adapters
5. Restart SDV and the adapter is deployed automatically
6. You can verify the deployment using Studio. Navigate to host/packages to verify that the adapter is deployed successfully.

Manual Deployment

Follow these instructions to install the advanced adapters:

1. Obtain new adapter (e.g. sdv.<adapter_name>.zip)
2. Copy sdv.<adapter_name>.zip to <SDV_install_dir>/tmp
3. Unzip the sdv.<adapter_name>.zip under <SDV_install_dir>/tmp
4. Open a shell window and go to the <SDV_Install_dir>

UNIX

```
/bin/sh
cd <SDV_Install_Dir>
```

Windows

```
- cmd.exe with "Admin Privileges"
cd <SDV_Install_Dir>
```

5. Check if you have already deployed sdv.<adapter-name>

UNIX:

```
ls -al ./packages
```

Windows:

```
dir .\packages
```

Example: `sdv.googlebigquery.jar` is represented as `<SDV_install_dir>/packages/GoogleBigQuery_1.jar` when SDV Server has already deployed it.

6. If you find your `<adapter-name>` in the previous step, you must undeploy it first. You can undeploy the adapter using the command below:

UNIX:

```
./bin/server_util.sh -server <hostname> [-port <port>] -user <user> -password  
<password> -undeploy -name <adapter-name> -version 1
```

Windows:

```
.\bin\server_util.bat -server <hostname> [-port <port>] -user <user> -password  
<password> -undeploy -name <adapter-name> -version 1
```

Note: `<adapter-name>` for undeploy must match the adapter name under `<SDV_install_dir>/packages` that you are trying to undeploy.

Example:

- **Unix:**

```
./bin/server_util.sh -server <hostname> [-port <port>] -user <user> -password  
<password> -undeploy -name GoogleBigQuery -version 1
```

- **Windows:**

```
.\bin\server_util.bat -server <hostname> [-port <port>] -user <user> -password  
<password> -undeploy -name GoogleBigQuery -version 1
```

7. To deploy the adapter using the command below:

UNIX:

```
./bin/server_util.sh -server <hostname> [-port <port>] -user <user> -password  
<password> -deploy -package ./tmp/sdv.<adapter-name>/sdv.<adapter-name>.jar
```

Windows:

```
.\bin\server_util.bat -server <hostname> [-port <port>] -user <user> -password
<password> -deploy -package .\tmp\sdv.<adapter-name>\sdv.<adapter-name>.jar
```

Example:**Unix:**

```
./bin/server_util.sh -server <hostname> [-port <port>] -user <user> -password
<password> -deploy -package ./tmp/sdv.googlebigquery/sdv.googlebigquery.jar
```

Windows:

```
.\bin\server_util.bat -server <hostname> [-port <port>] -user <user> -password
<password> -deploy -package .\tmp\sdv.googlebigquery\sdv.googlebigquery.jar
```

8. To verify the new adapter was undeployed and deployed, check <SDV_install_dir>/logs/cs_server.log.

You will see messages about undeployment and deployment of your adapter.

Example: log snippet from <SDV_install_dir>/logs/cs_server.log when undeploying and deploying sdv.googlebigquery.jar (aka name=GoogleBigQuery, version=1)

```
INFO [jetty thread pool-413] 2020-04-29 19:17:30.749 -0700 ExtensionManager -
Undeployment of Extension Package GoogleBigQuery:1 is successful!
```

```
INFO [jetty thread pool-420] 2020-04-29 19:19:16.672 -0700 ExtensionManager -
Deployment of Extension Package GoogleBigQuery:1 is successful!
```

Installing the SDV Client Drivers that are Distributed with SDV

This client distribution (driver zip) file includes the following components:

- ODBC
- ADO.NET
- JDBC
- SSIS

- Power BI

This zip file can be unpacked on each machine that has client application that needs access to the SDV Server.

To install the drivers distributed with SDV

1. Read any README files included with or associated with the download file.
2. Locate and extract the drivers zip file.
3. When installing the ODBC Win 64-bit driver on Windows 10, make sure to select Run as Administrator. Select the client EXE file, right click and select Run as Administrator. When prompted, select Yes and allow the installation to run to completion.
4. Follow the instructions in the SDV Administration Guide for details on how to complete configuration of each driver.
5. When the installation is complete, click Done to exit the installation program.

Importing Resources Defined in an Earlier Release

If you used a previous release of the adapter and defined resources, you can use them with this release. Follow the instructions for exporting and importing the resources in the *SDV User Guide*.

Manage Active Cluster Security

Users who create and manage an Active Cluster must have administrative privileges. SSL is used for inter-node communications and each server in an Active Cluster must have a valid digital certificate for authentication.

All cluster members must use the same level of encryption.

- [Updating the Digital Certificate to Secure Cluster Communication](#)
- [Set Access Privileges](#)

Updating the Digital Certificate to Secure Cluster Communication

Every SDV Server ships with a trial digital certificate so SSL works right out of the box. However, the security is poor. To secure cluster communication, you must update the digital certificate on each SDV Server node in the cluster.

Updating the digital certificate entails getting a signed digital certificate from a Certificate Authority (CA) and installing it in the keystore on each SDV Server. CAs are independent vendors (such as VeriSign) that have instructions on their websites for how to generate public key/private key pairs that accompany certificate requests. The CA then returns the digital certificate back to you. After you have this information, you need to install the digital certificate on the SDV Server.

To install a digital certificate on SDV Server

1. Open Studio, and select Administration > Launch Manager (Web) from the menu to open the Manager Web interface.
2. Click the CONFIGURATION tab and select SSL.
Manager displays the SSL Management page.
3. Enter new values as appropriate for your digital certificate, and click APPLY.

Set Access Privileges

You must have administrative privileges for Active Cluster management. Refer to the *Active Cluster Guide* for the specific rights needed for various cluster operations. Refer to the *Administration Guide* for more information about setting SDV access rights.

SDV and Business Directory Product Maintenance

This chapter explains the procedures involved in installing the latest service pack and the precautions that must be taken. If you are installing a base version of the products, you can skip this chapter.

- [About Hotfix Maintenance](#)
- [Applying the Service Pack to SDV Server, Studio, and Business Directory](#)
- [Applying the Service Pack to SDV Server, Studio, and Business Directory](#)
- [Applying the Service Pack or Hotfix to Active Cluster](#)
- [Upgrading from an Earlier Release and Migrating The Metadata](#)
- [Downgrade/Rollback](#)
- [Maintaining SDV-to-Client Application Connections](#)

About Hotfix Maintenance

If you are sent a patch to address a specific urgent issue, it is referred to as a hotfix patch. Hotfix patches have been quickly created by our engineering group for timeliness and are not for general use.

Hotfix patches are not automatically carried over through export and import processes during an upgrade or migration. They need to be reapplied if you migrate to another instance that is the same version. Hotfixes are also built for specific versions of SDV, so they should not be used for later versions of SDV than the version for which they were built.

For hotfixes, run the update process for SDV and Business Directory.

Applying the Hotfix to SDV Server, Studio, and Business Directory

Hot fixes include bug fixes to any SDV major, minor or service pack version.

The latest hotfix, must be applied to the same SDV major, minor and service pack level. For example,

- hotfix SDV 8.3.0.005 can only be applied to SDV 8.3.0
- hotfix SDV 8.3.1.002 can only be applied to SDV 8.3.1

The latest hotfix, requires a clean update of your SDV Server and Business Directory environment. The optimal way to achieve the clean environment is to apply the hotfix as if it was a Service Pack.

To apply a hotfix

1. Optionally, perform a full SDV and Business Directory backup. For details about backup export, see the SDV Administration Guide or the Business Directory Guide.
2. If installing on Windows, be sure to use Run as Administrator.
3. For Windows, make sure to close any open windows to the repository/jdk folder.
4. Stop the SDV Server.
5. Copy the hotfix zip file to <SDV_install_dir>.
6. Unzip the file.

To Rollback a hotfix, refer to [To rollback a hotfix for SDV Server, Studio, or Business Directory](#).

Note:

All SDV hotfixes and patches will ship the log4j2.properties.cfgt and log4j2.properties file. Application of a hotfix/patch will automatically trigger server regeneration of log4j2.properties which means any custom changes a customer has in that file or the corresponding cfgt file will definitely be lost.

Downgrade/Rollback

Here are the steps to Upgrade/Downgrade to 8.9.0.

Note 8.9.0 has repo version change. So we need to account for this during the downgrade process for both SDV server and BD. As of now only upgrade works. Downgrade works on windows only.

Upgrade steps for SDV Server

1. Install the base install(8.5.6, 8.8.0, 8.6.0, 8.7.0, 8.8.1)
2. Stop the monitor, repo and cache (make sure all the java process and postgres process are off. We can see those in Task manager). One way to check is in cs_server.log. It should say 'Bye', if successfully stopped. Also checked for any background stale processes).
3. Unzip the latest patch to the install directory.
(aix/linux, cd to installDir and execute the command "unzip -o ~/SPOT_SDV_server_8.9.0_all.zip -d .", make sure you have copied the patch to home directory)
4. chmod 733 bin/cis_upgrade_patch.sh
5. execute bin/cis-upgrade_patch.sh/bat (Enter the database admin password for Linux AIX. On windows it is not necessary)
6. The server will start automatically.
7. Check the cs-server.log for 'Ready' message and check errors.

Upgrade steps for BD Server

1. Install the base install(8.5.6, 8.8.0, 8.6.0, 8.7.0, 8.8.1)
2. Stop the monitor, repo(make sure all the java process and postgres process are off). One way to check is in cs_bd_server.log. It should say 'Bye', if successfully stopped. Also checked for any background stale processes)
3. Unzip the latest patch to the install directory.
4. chmod 733 bin/bd_upgrade_patch.sh
5. execute bin/bd_upgrade_patch.sh/bat(On linux, it will prompt for database admin password, enter the password. on windows it won't)The server will start

automatically.

6. Check the cs-server.log for 'Ready' message and keep an eye for any errors.

Upgrade steps for SDV Studio

1. Install the base install(8.8.1, 8.8.0, 8.6.0, 8.7.0, 8.5.6)
2. Stop the studio processes
3. Unzip the latest patch to the install directory.
4. execute bin/studio_upgrade_patch.sh(bat)

Downgrade steps for SDV Server

1. Stop monitor daemon (bin/composite.<bat|sh> stop)
2. Start server process without monitor daemon using composite_server.sh(bat) run
 <- this is a foreground process, it will keep running until the next command is successful. So open another cmd prompt for next command. Also before you execute next command in another cmd prompt, open cs-server.log and see that it says 'Ready'.
3. Run server_util.sh(bat) -server <id> -port <port> -user <uid> -password <password> -rollbackRepository -toVersion 8.8.1 (or 8.7.0 Or 8.8.0)
 Example: bin/server_util.bat -server localhost -port 9400 -user admin -password admin1 -rollbackRepository -to Version 8.5.x (for 8.5.X)

 bin/server_util.bat -server localhost -port 9400 -user admin -password admin1 -rollbackRepository -toVersion 8.8.1

 Make sure to check the repo version after rollback: bin\psql.exe -Uroot -hlocalhost -p9408 -d cisrepo -c "select * from cisrepo.metadata_version"(This is optional)
4. Kill all java processes.
5. Stop the repo and cache.
6. Run <install_dir>/install/rollback_patch.sh(bat) script for CIS.
7. chmod 733 for the /bin/cis_downgrade_patch.sh
8. Run cis_downgrade_path.sh(bat)
9. Apply 8.8.X or 8.x.x patch (downgraded version)
10. For Linux/AIX, start the repo bin/composite.sh repo start

set LIBPATH(AIX) and LD_LIBRARY_PATH(Linux) to installDir/repository/lib,

Ex. export LIBPATH=/opt/SPOTCO/SDV_Server_8.9.0/repository/lib

Run the following command:

```
./repository/bin/psql -U root -p 9408 -h localhost -d cisrepo -c "update
cisrepo.metadata_version set version =37;"
```

Number 37 is the metadata version for 8.8.1, it will be different for other SDV versions. refer to the metadata table at the end

Stop the repo.

11. See the Note 2

12. Execute bin/cis_upgrade_patch.sh

After the downgrade process, we need to manually undeploy and deploy the 3 adapters mentioned below. Here are all the downgrade steps:

1. cd to installDir
2. Undeploy the ComputeDB, ApacheDrill and TibcoStreaming adapter using the following commands:

```
bin/server_util.sh -server localhost -port 9400 -user admin -password admin1 -domain composite -
undeploy -name ComputeDB -version 1 -verbose
```

```
bin/server_util.sh -server localhost -port 9400 -user admin -password admin1 -domain composite -
undeploy -name ApacheDrill -version 1 -verbose
```

```
bin/server_util.sh -server localhost -port 9400 -user admin -password admin1 -domain composite -
undeploy -name TibcoStreaming -version 1 -verbose
```

13. Redploy ComputeDB, ApacheDrill and TibcoStreaming adapter using following commands.

```
bin/server_util.sh -server localhost -port 9400 -user admin -password admin1 -domain composite -
deploy -package packages/archived_ds_adapters/dst-drill-8.X.Y-080X0Y001.jar -verbose
```

```
bin/server_util.sh -server localhost -port 9400 -user admin -password admin1 -domain composite -
deploy -package packages/archived_ds_adapters/dst-computedb-8.X.Y-080X0Y001.jar -verbose
```

```
bin/server_util.sh -server localhost -port 9400 -user admin -password admin1 -domain composite -
deploy -package packages/archived_ds_adapters/dst-streambase-8.X.Y-080X0Y001.jar -verbose
```

14. Restart the server.

Note1: Windows server also has Studio, so follow the upgrade downgrade process for studio here too for windows.

Note2: When downgrading to 8.7.0/8.5.6. You would need to copy below files from a 8.5.6 install to downgraded environment and restart the server.

apps/common/lib/saxon9-unpack-9.4.jar

apps/common/lib/saxon9-sql-9.4.jar

apps/common/lib/saxon9ee-9.4.jar

Downgrade Steps for BD Server

Before starting the repo rollback, you would need to enable the admin interface.

1. Edit /opt/SPOTCO/SDV_BD_Server_8.8/bd/conf/server/bd_server_values.xml (To edit this go to windows explorer and follow the given path. The file should look like below image in the path, open it in notepad and edit).

2. Add the below tag between <common:attributes xmlns:common="http://www.compositesw.com/services/system/util/common"> and </common:attributes>

Tip: in place of and add the below code

```
<common:attribute>
<common:name>/bd/config/enableAdminInterface</common:name>
<common:type>BOOLEAN</common:type>
<common:value>true</common:value>
</common:attribute>
```

3. Restart the server command:
bin\bd.bat monitor restart. Wait until you see ready in cs_bd_server.log

Start the downgrade steps:

1. Stop monitor daemon(bin/bd.<bat|sh> stop) use this command:(bin\bd.bat monitor stop)
2. Start server process without monitor daemon using bd_server.sh(bat) run <- this is

a foreground process, it will keep running until the next command is successful. So open another cmd prompt for next command. Also before you execute next command in another cmd prompt, open cs-server.log and check that it says 'Ready?'

3. Run `server_util.sh(bat) -server <id> -port <port> -user <uid> -password <password> -rollbackRepository -toVersion 8.5.0 (or 8.7.0 or 8.8.0)` Ex: `bin/bd_server_util.bat -server localhost -port 9500 -user admin -password admin1 -rollbackRepository -toVersion 8.8.0 /8.5.6` When you run this command, it displays the below message error message

```
[cgopara1@guwddv004 IDV_BD_Server_8.8]$ ./bin/bd_server_util.sh -server localhost -port 9500 -user admin -password admin1 -rollbackRepository -toVersion 8.8.0
No KeyStore password strings in init_server_keystore_files file and exiting to restore defaults in the script. Please re-run the program.
```

4. Rerun the command until you don't see the error. Make sure to check the repo version after rollback: `bin\psql.exe -Uroot -hlocalhost -p9508 -d bdrepo -c "select * from bdrepo.metadata_version"` (This is optional).
5. Make sure to check `cs_bd_server.log` to make sure the repo roll back is done. It should have a message similar to this.


```
INFO [jetty thread pool-75] 2024-10-29 12:57:13.163 -0400 RollbackRepositoryAction - Requesting repository rollback
INFO [jetty thread pool-75] 2024-10-29 12:57:13.166 -0400 RepositoryImpl - Rolled back repository from version 31 to version 30.(validate this )
INFO [jetty thread pool-75] 2024-10-29 12:57:13.167 -0400 RepositoryImpl - Repository version rollback complete.
INFO [jetty thread pool-75] 2024-10-29 12:57:13.167 -0400 RollbackRepositoryAction - Shutting down server
```
6. Kill all java processes <- when doing repo rollback for 8.5.6, (If you notice a running task named Java(TM) Platform SE binary, check its properties to identify which server it's associated with—either SDV Server or BD Server. If it is related to the SDV Server, you can safely ignore it.) Stop the repo and cache.
7. Run `<install_dir>/bd//install/rollback_patch.sh(bat)` script for CIS
8. `chmod 733` for the `/bin/bd_downgrade_patch.sh(bat)`
9. Run `bd_downgrade_path.sh(bat)`
10. Apply 8.5.6 or 8.8.0 patch (downgraded version)

Process:

Copy the patch file `SPOT_SDV_bd_8.8.0_all` from the shared location

`Q:\release\cis\8.8.0\patch` to your BD server installation directory. Once copied, unzip the patch file into the same installation directory

See Note 2

11. Execute `bin/bd_upgrade_patch.sh`

Note on steps 3. if you are doing repo rollback for 8.5.6, toVersion will still be 8.5.0

As of now, when you do repo rollback on 8.5.6 as below

`bin/server_util.sh -server localhost -port 9400 -user admin -password admin1 -rollbackRepository -toVersion 8.5.0`, even though the command is successful.

Note 2: When downgrading to 8.7.0/8.5.6. You would need to copy below files from a 8.5.6 install to downgraded env and restart the server:

`apps/common/lib/saxon9-unpack-9.4.jar`

`apps/common/lib/saxon9-sql-9.4.jar`

`apps/common/lib/saxon9ee-9.4.jar` for that issue.

Downgrade steps for SDV Studio

1. Stop the studio processes.
2. cd to `installDir/install` folder
3. Run `<install_dir>/install/rollback_*.patch*.bat`
4. Execute `./bin/studio_downgrade_patch.bat`
5. Apply unzip the 8.7.0/8.5.6, 8.8.0 or 8.8.1 this patch in to the install dir
6. Execute this command `./bin/studio_upgrade_patch.bat`
7. When downgrading to 8.7.0/8.5.6. You would need to copy below files from a 8.5.6 install to downgraded env and restart the server
 - `apps/common/lib/saxon9-unpack-9.4.jar`
 - `apps/common/lib/saxon9-sql-9.4.jar`
 - `apps/common/lib/saxon9ee-9.4.jar`

Repository Versions of SDV Server/BD

Release Version	SDV Server	BD
8.3	32	30
8.4	33	30
8.5	34	30
8.6	34	30
8.7	35	30
8.8.0	36	30
8.8.1	37	31
8.9.0	38	31

About Service Pack Maintenance

A service pack is a zipped package of files that fixes known issues and often provides enhanced functionality. All SDV Server, Studio and Business Directory software service packs are cumulative and supersede previously released service packs. A service pack should be applied on all computers where SDV products are installed, keeping them all at the same revision level. Installation of a service pack does not change configuration settings and custom functionality.

It is recommended that you keep your SDV Server and Business Directory Server instances at the same service pack level.

Applying the Service Pack to SDV Server, Studio, and Business Directory

Important bug fixes and additional functionality are added to SDV with each service pack.

Service packs typically involve careful ordering of procedural steps to make sure appropriate scripts are generated and available, and backup files saved.

Notes:

- Customers using advanced data source adapters that require OAuth need to first run SDV as a stand-alone server (no Monitor Daemon), set up OAuth once, and then run SDV with the Monitor Daemon.
- Sometimes when running Manager after a service pack install an error message pops up (Unable to finish loading...). You can safely ignore this message, close, and reopen Manager.

To apply a service pack

1. Optionally, perform a full SDV and Business Directory backup. For details about backup export, see the SDV Administration Guide or the Business Directory Guide.
2. If installing on Windows, be sure to use Run as Administrator.
3. For Windows, make sure to close any open windows to the repository/jdk folder.
4. Stop all SDV processes, instances, and databases.
5. Copy the service pack zip file to <SDV_install_dir>.
6. Unzip the file.

7. When asked whether to replace the existing files, reply Yes.

If you are not asked whether to replace files, the service pack is being extracted to some directory other than <SDV_install_dir>. The service pack must be extracted to the appropriate installation directory, and must overwrite files in that directory.

8. Run one or more of these scripts to upgrade your products.

Note: When running on a Linux/AIX environment, you will have to grant execute permission on the upgrade scripts. Run the following command to do this:


```
chmod 755 *.sh
```

9. Enter the repo and cache passwords when prompted.

Product	Directory	Script
SDV Studio	bin	studio_upgrade_patch.bat
SDV Server	bin	cis_upgrade_patch.<sh bat>
SDV Business Directory	bin	bd_upgrade_patch.<sh bat>

10. After you run the scripts successfully, restart the SDV Server to fully activate any new libraries included in the upgrade.

To Rollback a service pack, refer to [Downgrade/rollback a service pack for SDV Server, Studio, or Business Directory](#).

Applying the Service Pack or Hotfix to Active Cluster

When updating Active Cluster, it is critical that clusterid remain unchanged. Changing it can adversely affect cached data.

These instructions are guidelines developed from testing with a two node Active Cluster environment.

To upgrade with scheduled system downtime

1. Determine and note the clusterid, so that it can remain unchanged.
2. Make sure that the cluster is in sync. That is, make sure both nodes are in the cluster and are functional.
3. Disable any cache refreshes on both of the nodes by enabling the Studio configuration *Disable Cache Refreshes* on both nodes.
4. Configure the load balancer to stop sending traffic to node 1.
Shutting down node 1 would cause in-flight requests to fail.
5. Remove node 1 from the cluster.

6. If your SDV products are running, stop them, including all processes and databases used as repositories or caches.
7. Apply the service pack or hot fix to node1.
8. Start up node 1, but do not re-join the cluster.
9. Configure the load balancer to send traffic to node 1 instead of node 2.
10. Remove node 2 from the cluster.
11. Apply the service pack or hot fix to node 2.
12. Join node 1 and node 2 to the cluster.
13. Enable cache refreshes on both of the nodes.
14. Verify the cluster status.

Upgrading from an Earlier Release and Migrating The Metadata

This section is a guide for customers who are upgrading to SDV from a previous version and want to migrate metadata from that version to the new version.

SDV is released either as a Mainstream Support Release (MS) or a Long Term Support Release (LTS). More details about these can be read at https://docs.tibco.com/pub/sdv/general/lts/sdv_lts_releases.htm

The metadata upgrade process requires installing a new SDV instance in parallel with the existing SDV instance, exporting the metadata from the old instance, and importing the old instance's backup CAR file into the new SDV instance.

SDV recommends that you keep the older SDV instance until you are sure the new installation is stable. However, be aware that:

- If you are running two versions of SDV simultaneously in the same instance, then their port numbers must be different.
- If you are using Active Cluster, all servers in the cluster must be running at the same version and patch level.
- Change the older SDV version's Server services to a "manual" start. In the Windows OS this is set to "automatic" by default.

Make sure that you have administrator privileges and perform all of the steps below as that user. In addition, it is advised that the installation and the upgrade steps be performed by the same user to avoid any permission issues that may arise.

To upgrade and migrate your existing installation, follow the steps (in the given order) in these sections

1. [Documenting the Existing SDV Instance](#)
2. [Considerations for Upgrading to SDV 8.X](#)
3. [Exporting Metadata from the Existing SDV Instance](#)
4. [Installing the New Version of SDV](#)
5. [Importing Metadata into the New SDV Instance](#)
6. [Verifying a Successful Installation](#)

Documenting the Existing SDV Instance

Before making a backup of the existing SDV instance, document the key features of the instance. These settings are later applied to the new SDV instance to ensure the consistency of results returned from published resources.

Note: If you cannot upgrade directly from the existing SDV instance to the new version, multiple versions of SDV and multiple export and import processes might be required, so that database schemas remain compatible.

Make note of the settings in the following table.

Setting	What to Record
Ports	The port numbers for the existing instance, because after the installation of the new server is complete, the port numbers of the new instance might need to be changed.
Authentication mechanism	The authentication configuration. If LDAP based authentication is used and you have modified <code>ldap.properties</code> in the SDV Server, then the same settings need to be applied to the new server. This configuration determines various authentication mechanisms enabled within the server.

Setting	What to Record
Users/groups	The groups created in Server and the users that belong to these groups. If LDAP authentication is used, note the LDAP groups that were imported into Server.
Metadata repository	The full path of the repository location and the administrator user ID and password.
Custom data sources	Custom data sources that were introspected and any custom drivers that were used to introspect these sources.
External libraries	Any external libraries that were referenced from the instance.
Customized settings, including Java flags, managed and unmanaged memory	Configuration parameter settings for the existing Server instance. From the Administration menu, select Configuration and check all relevant parameter settings. The new SDV instance's settings should match the old instance settings if you want similar results and performance.

Considerations for Upgrading to SDV 8.X

- In order to keep your database schemas compatible, you will need to preserve your metadata. You may need to perform an export and import of metadata.
 - During the export you perform in [Exporting Metadata from the Existing SDV Instance](#), all of the relevant information from your old repository is captured.
 - During the import you perform in [Importing Metadata into the New SDV Instance](#), all of the relevant information from your old repository is transferred into the new SDV PostgreSQL repository database.
- Long-Term Support (LTS) releases are typically supported with Hotfixes and Service Packs. This approach provides an option more suited for strictly regulated environments, or for customers who prefer a fully supported environment for an extended period of time, without the need for upgrades.
- On the windows platform only, if you are upgrading from a pre-8.5 SDV Server installation to SDV Server 8.5 or higher, note that you need to upgrade your SDV Studio to the same version you upgraded your SDV Server with.

Refer to [Upgrade Path - Examples](#) for a few common examples to understand the various Upgrade paths you may have to take to Upgrade your current SDV version to the version of your choice..

Exporting Metadata from the Existing SDV Instance

The first step for upgrade or migration is to export the existing metadata information from the repository. This process writes out a CAR file that includes six files containing metadata, scheduling, settings, and user information.

To run the export

1. Verify that you have administrator privileges.
2. Open a command prompt window.
3. Navigate to <SDV_install_dir>/bin.
4. Perform a full backup with the options that you need:
 - Using Studio. See “Using Studio for a Full Server Backup” in the *SDV User Guide*.
 - Using the SDV backup_export utility. For more information, see the *SDV Administration Guide*.
5. Locate and copy the resulting CAR file to a safe and easily accessible location for use later.

Installing the New Version of SDV

To install a new SDV when you are upgrading from an earlier release

1. Review the new features and bug fixes as documented in the *SDV and Business Directory Release Notes*.
2. Install the new version of Server as described in [Installing SDV Server, Studio, and Drivers](#) or [Silent Mode Installation](#).
3. Install the latest versions of all other SDV software that you use.
4. Set up any external libraries, including JDBC drivers, and then shut down and restart the server.

5. Deliver the upgraded drivers (the SDV ODBC driver and the SDV JDBC driver) to the dependent clients.
6. To ensure consistency in results and performance, make the configuration of the new server instance similar to the old instance.
7. Perform the instructions in [Importing Metadata into the New SDV Instance](#) .

Perform the instructions in [Verifying a Successful Installation](#) .

Importing Metadata into the New SDV Instance

If you are upgrading your version of SDV from an earlier version and you have completed the instructions in [Exporting Metadata from the Existing SDV Instance](#) , then follow the instructions in this section. If you are performing a new installation, you can skip these instructions.

After the new SDV instance is successfully installed, the metadata from old SDV instance needs to be imported into the new instance. After the import is successfully completed, settings such as Java configurations, managed memory setting and ports can be updated on the new instance.

To run the import

1. Verify that you have administrator privileges.
2. Locate the CAR file that you produced from [Exporting Metadata from the Existing SDV Instance](#) .
3. Perform a full backup import with the options that you need:
 - Using the Studio Import dialog window. For more information, see the *SDV User Guide*.
 - Using the SDV backup_import utility. For more information, see the *SDV Administration Guide*.
4. Validate that the SDV resources or other settings are as you expect in the new version of Studio.

Verifying a Successful Installation

To verify that your installation of the SDV software was successful, follow the steps in this section.

To verify a successful installation

1. Start Studio. Follow instructions in the SDV Getting Started Guide or in the SDV User Guide.
2. If you have just completed an upgrade from one version of SDV to a new one, then we suggest that you complete these instructions:

- a. Determine a set of tests that will touch all published resources and all introspected data sources, and then apply the tests against:

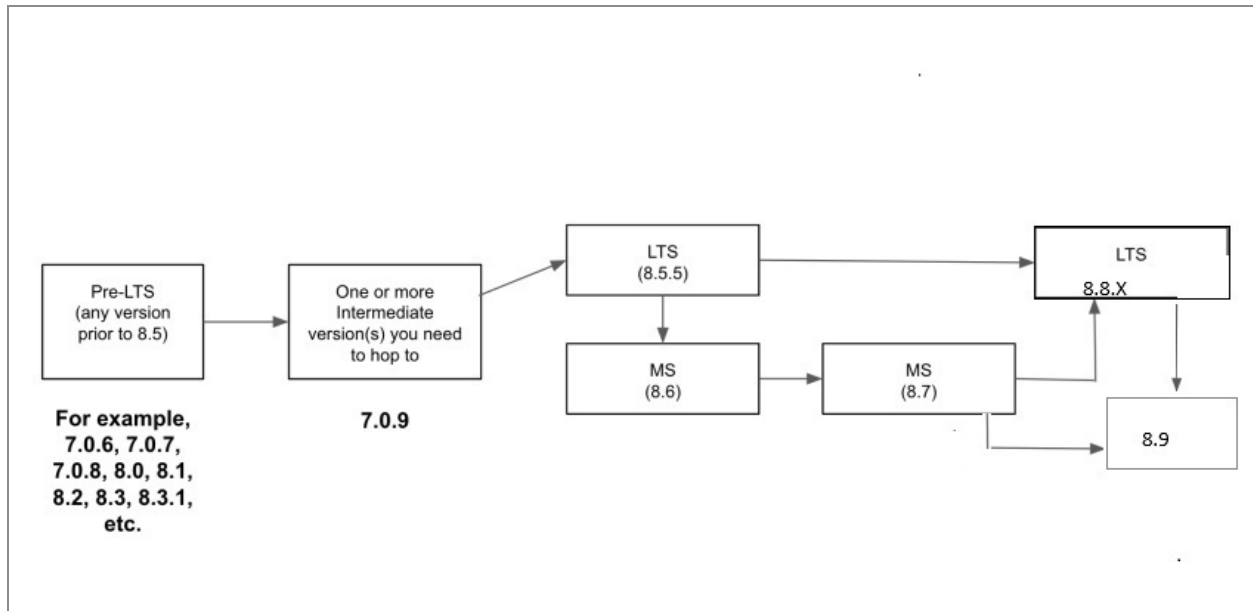
- The existing instance of SDV, as a sanity check.
- The new instance of SDV, to ensure the same results are produced.

- b. Configure and use the PubTest tool to test all your published resources.

The PubTest program can be configured to test all published resources using JDBC, ODBC, and Web services. Additional configuration might be required to test the ODBC and Web services. Starting with SDV 4.0, an end-to-end testing program referred to as PubTest (pubtest.java) is included with the SDV installation. This program is located in the <SDV_install_dir>\apps\jdbc directory. A PubTest.doc file in this directory provides additional documentation about using this tool.

Upgrade Path - Examples

Upgrading from an earlier version of SDV to the latest version may require that you need to upgrade to one or more intermediate versions before upgrading to the latest. An illustration of your upgrade path is given below:



The table below gives some examples to help you understand the upgrade process for your version of SDV.

Current Version to Target Version	Steps
LTS to LTS (8.5.X to 8.9.0)	1. Export Full Server Backup of TDV 8.5.X. 2. Install new SDV 8.9.0. 3. Import Full Server Backup into the new SDV 8.9.0 install.
LTS to LTS (8.8.0 to 8.9.0)	Run SDV 8.9.0 Upgrade process.
LTS to LTS (8.8.1 to 8.9.0)	Run SDV 8.9.0 Upgrade process.
MS to LTS (8.7 to 8.9.0)	1. Export Full Server Backup of TDV 8.7.0. 2. Install new SDV 8.9.0. 3. Import Full Server Backup into the new SDV 8.9.0 install.

Maintaining SDV-to-Client Application Connections

This section includes instructions for how to update connections between SDV Server and your client applications in the following sections:

- [Updating an ODBC Client Application](#)
- [Updating a JDBC Client Application](#)

Updating an ODBC Client Application

To patch ODBC client applications

1. Install the ODBC patch as you would for a Studio installation, but apply the ODBC patch only to where you installed a SDV ODBC client:
 - 32-bit driver: <SDV_install_dir>/apps/odbc
 - 64-bit driver: <SDV_install_dir>/apps/odbc64
2. See the *SDV Client Interfaces Guide* for more information.

To patch ODBC client users on Windows 7

1. Locate your Windows 7 DSN entry file.
2. Recreate any custom system DSNs, using DSN tools.
3. See the *SDV Client Interfaces Guide* for more information.

Updating a JDBC Client Application

To patch JDBC client applications

1. If you only need to obtain the updated SDV JDBC driver for your JDBC client, you can do one of the following:
 - Obtain the updated csjdbc.jar file from the server's <SDV_install_dir>/apps/jdbc/lib directory after the Studio patch is installed.

- Extract any one of the csjdbc.jar files directly from the patch zip file.
2. See the *SDV Client Interfaces Guide* for more information.

Uninstalling SDV

This topic describes the process of uninstalling Spotfire® Data Virtualization (SDV) and related products for Windows and UNIX. The uninstall process is similar, regardless of whether you performed a silent or interactive installation.

- [Uninstalling SDV on Windows](#)
- [Uninstalling SDV on UNIX](#)

Uninstalling SDV on Windows

When you uninstall SDV, everything stored in the metadata repository is deleted along with the SDV software.

To uninstall SDV on Windows

1. Stop the Server and Repository if they are running.
2. Start the uninstallation process:
... > Uninstall SDV
For a silent uninstall, SDV is uninstalled without further interaction. For an interactive uninstall, go to step 3.
3. Click OK to confirm the uninstall.
4. Click Done when the uninstallation process is completed.

Uninstalling SDV on UNIX

The following tasks are described here:

- [Preparing for Uninstalling on UNIX](#)
- [Uninstalling SDV On UNIX](#)

Preparing for Uninstalling on UNIX

Before you uninstall SDV, remove the SDV service files from the installation machine, because the uninstaller does not remove these files automatically.

To remove the SDV service files `cis.repository` and `cis.server`

1. Log into the installation machine as root.
2. Change the working directory to `<SDV_install_dir>/bin`.
3. Run the following command:

```
cis_remove_services.sh
```

Uninstalling SDV On UNIX

During the uninstallation process, all the components from the previous installation are removed. You cannot uninstall the components individually.

To uninstall SDV on UNIX

1. Log into the installation machine as the user that installed the software.
2. Run the following command:

```
<SDV_install_dir>/uninstall
```

For a silent uninstall, run

```
./uninstall --mode "unattended"
```

For an interactive uninstall, go to step 3.

3. Press the Enter key.

You will see a warning about loss of data.

4. Press the Enter key to complete the uninstallation process and leave the uninstaller.

Spotfire Documentation and Support Services

For information about this product, you can read the documentation, contact Spotfire Support, and join Spotfire Community.

How to Access Spotfire Documentation

Documentation for Spotfire products is available on the [Product Documentation website](#), mainly in HTML and PDF formats.

The [Product Documentation website](#) is updated frequently and is more current than any other documentation included with the product.

Product-Specific Documentation

The following documentation for this product is available on the [Spotfire® Data Virtualization](#) page.

Users

- Spotfire Getting Started Guide
- Spotfire User Guide
- Spotfire Web UI User Guide
- Spotfire Client Interfaces Guide
- Spotfire Tutorial Guide
- Spotfire Northbay Example

Administration

- Spotfire Installation and Upgrade Guide
- Spotfire Administration Guide
- Spotfire Active Cluster Guide
- Spotfire Security Features Guide

Data Sources

- Spotfire Adapter Guides

Spotfire Data Source Toolkit Guide (Formerly Extensibility Guide)

References

Spotfire Reference Guide

Spotfire Application Programming Interface Guide

Other

Spotfire Business Directory Guide

Spotfire Discovery Guide

Spotfire and Business Directory Release Notes - Read the release notes for a list of new and changed features. This document also contains lists of known issues and closed issues for this release.

Release Version Support

Spotfire 8.9 is designated as Long Term Support (LTS) versions. Some release versions of Spotfire® Data Virtualization products are selected to be long-term support (LTS) versions. Defect corrections will typically be delivered in a new release version and as hotfixes or service packs to one or more LTS versions. See also [Long Term Support](#).

How to Contact Support for Spotfire Products

You can contact the Support team in the following ways:

- To access the Support Knowledge Base and getting personalized content about products you are interested in, visit our [product Support website](#).
- To create a Support case, you must have a valid maintenance or support contract with a Cloud Software Group entity. You also need a username and password to log in to the [product Support website](#). If you do not have a username, you can request one by clicking **Register** on the website.

How to Join Spotfire Community

Spotfire Community is the official channel for Spotfire customers, partners, and employee subject matter experts to share and access their collective experience. Spotfire Community offers access to Q&A forums, product wikis, and best practices. It also offers access to extensions, adapters, solution accelerators, and tools that extend and enable customers to gain full value from Spotfire products. In addition, users can submit and vote on feature requests from within the [Spotfire Ideas Portal](#). For a free registration, go to [Spotfire Community](#).

Legal and Third-Party Notices

SOME CLOUD SOFTWARE GROUP, INC. ("CLOUD SG") SOFTWARE AND CLOUD SERVICES EMBED, BUNDLE, OR OTHERWISE INCLUDE OTHER SOFTWARE, INCLUDING OTHER CLOUD SG SOFTWARE (COLLECTIVELY, "INCLUDED SOFTWARE"). USE OF INCLUDED SOFTWARE IS SOLELY TO ENABLE THE FUNCTIONALITY (OR PROVIDE LIMITED ADD-ON FUNCTIONALITY) OF THE LICENSED CLOUD SG SOFTWARE AND/OR CLOUD SERVICES. THE INCLUDED SOFTWARE IS NOT LICENSED TO BE USED OR ACCESSED BY ANY OTHER CLOUD SG SOFTWARE AND/OR CLOUD SERVICES OR FOR ANY OTHER PURPOSE.

USE OF CLOUD SG SOFTWARE AND CLOUD SERVICES IS SUBJECT TO THE TERMS AND CONDITIONS OF AN AGREEMENT FOUND IN EITHER A SEPARATELY EXECUTED AGREEMENT, OR, IF THERE IS NO SUCH SEPARATE AGREEMENT, THE CLICKWRAP END USER AGREEMENT WHICH IS DISPLAYED WHEN ACCESSING, DOWNLOADING, OR INSTALLING THE SOFTWARE OR CLOUD SERVICES (AND WHICH IS DUPLICATED IN THE LICENSE FILE) OR IF THERE IS NO SUCH LICENSE AGREEMENT OR CLICKWRAP END USER AGREEMENT, THE LICENSE(S) LOCATED IN THE "LICENSE" FILE(S) OF THE SOFTWARE. USE OF THIS DOCUMENT IS SUBJECT TO THOSE SAME TERMS AND CONDITIONS, AND YOUR USE HEREOF SHALL CONSTITUTE ACCEPTANCE OF AND AN AGREEMENT TO BE BOUND BY THE SAME.

This document is subject to U.S. and international copyright laws and treaties. No part of this document may be reproduced in any form without the written authorization of Cloud Software Group, Inc.

Spotfire, the Spotfire logo, the Spotfire O logo, ActiveSpaces, Enterprise Messaging Service, Spotfire, TERR, S-PLUS, and S+ are either registered trademarks or trademarks of Cloud Software Group, Inc. and/or its subsidiaries Spotfire Software Inc. in the United States and/or other countries.

Java and all Java based trademarks and logos are trademarks or registered trademarks of Oracle and/or its affiliates.

All other product and company names and marks mentioned in this document are the property of their respective owners and are mentioned for identification purposes only. You acknowledge that all rights to these third party marks are the exclusive property of their respective owners. Please refer to Cloud SG's Third Party Trademark Notices (<https://www.cloud.com/legal>) for more information.

This document includes fonts that are licensed under the SIL Open Font License, Version 1.1, which is available at: <https://scripts.sil.org/OFL>

Copyright (c) Paul D. Hunt, with Reserved Font Name Source Sans Pro and Source Code Pro.

Cloud SG software may be available on multiple operating systems. However, not all operating system platforms for a specific software version are released at the same time. See the “readme” file for the availability of a specific version of Cloud SG software on a specific operating system platform.

THIS DOCUMENT IS PROVIDED “AS IS” WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NON-INFRINGEMENT.

THIS DOCUMENT COULD INCLUDE TECHNICAL INACCURACIES OR TYPOGRAPHICAL ERRORS. CHANGES ARE PERIODICALLY ADDED TO THE INFORMATION HEREIN; THESE CHANGES WILL BE INCORPORATED IN NEW EDITIONS OF THIS DOCUMENT. CLOUD SG MAY MAKE IMPROVEMENTS AND/OR CHANGES IN THE PRODUCT(S), THE PROGRAM(S), AND/OR THE SERVICES DESCRIBED IN THIS DOCUMENT AT ANY TIME WITHOUT NOTICE.

THE CONTENTS OF THIS DOCUMENT MAY BE MODIFIED AND/OR QUALIFIED, DIRECTLY OR INDIRECTLY, BY OTHER DOCUMENTATION WHICH ACCOMPANIES THIS SOFTWARE, INCLUDING BUT NOT LIMITED TO ANY RELEASE NOTES AND “README” FILES.

This and other products of Cloud SG may be covered by registered patents. For details, please refer to the Virtual Patent Marking document located at <https://www.tibco.com/patents>.

Copyright © 2002-2026. Cloud Software Group, Inc. All Rights Reserved.