



Remote SAS® Data Agent: Deployment and Administration Guide

2026.09

This document might apply to additional versions of the software. Open this document in [SAS Help Center](#) and click on the version in the banner to see all available versions.

Quick Start for Remote SAS Data Agent	2
How to Use the Quick Start	2
First Steps	3
Register Remote SAS Data Agent with the SAS Viya Platform	5
Configure Connectivity and Authentication with the SAS Viya Platform	6
Run the Container	7
Download the dagentsrv Plug-in to the sas-viya CLI and Sign In	7
Example: Test Remote SAS Data Agent	7
Next Steps	8
Frequently Asked Questions	8
How does remote SAS Data Agent work?	8
What is SAS Container Manager?	9
What operating system is the container image based on?	9
Can I manage the container with Docker, Docker Compose, or Podman commands?	9
How is access to data sources configured?	10
What network security configurations allow incoming requests to remote SAS Data Agent from the SAS Viya platform?	10
What is required for authentication?	11
What options are available to store encrypted data?	11
How should Vault keys be read to start the container?	12
Can a customer-provided database server be used to store SYSCAT information?	13

How is TLS configured?	13
How is mTLS configured?	14
How is logging configured?	14
How do I determine if remote SAS Data Agent is running?	15
Requirements for Remote SAS Data Agent	15
Host Machine	15
Kubernetes	17
Connectivity Requirements	17
Deployment Example: Kubernetes	18
Before You Begin	18
Step 1 — Download the Required Files	19
Step 2 — Access the Container Image	20
Step 3 — Tag and Store the Container Image in a Local Docker Registry	21
Step 4 — Copy and Edit the Sample Deployment YAML File	21
Step 5 — Deploy the Container Image	22
Configuration	23
Configure Remote SAS Data Agent	23
Configure Data Sources	33
Software Updates	34
Run the Container with a New Release	34
Run the Container with a New License	35
Back Up and Restore	36
What Is Backed Up?	36
Back Up and Restore from the Storage Systems in the Container	37
Migrate Data	39
Before You Begin	39
Step 1: Export the SYSCAT Information to a Backup File	39
Step 2: Restore the SYSCAT Backup	40
Step 3: Migrate Credentials	41
Reference Topics	43
dagentsrv Plug-in to the SAS Viya CLI	43
SAS Container Manager Commands and Options	43

Quick Start for Remote SAS Data Agent

How to Use the Quick Start

Remote SAS Data Agent is an on-premises server that runs as a container on a host machine. After the container is started, the SAS Data Agent plug-in (dagentsrv plug-in) to the SAS Viya CLI (sas-viya CLI) is used to access, copy, and load data between on-premises data sources and data sources that are available to the SAS Viya platform.

The Quick Start provides the basics for configuring a connection to the SAS Viya platform deployment and for running the container. Follow the steps to gain an understanding about the

required files and software. At the end of the Quick Start, refer to the documentation in Next Steps to further customize remote SAS Data Agent for your environment.

First Steps

Before you begin, make sure that you have a supported version of Docker Desktop on Linux or Podman on Linux installed on the host machine where you plan to run the container. For more information, see [“Requirements for Remote SAS Data Agent” on page 15](#).

In this Quick Start, the following directories and files are on the host machine:

```
working-directory
├── container-manager
├── containermgr-linux.tgz
├── da-vars.env
├── sas-access.properties
├── mysasdata
│   ├── mydata
│   ├── list-of-trusted-certificate-authorities.pem
│   ├── tls-certificate-for-remote-data-agent-server.pem
│   └── tls-certificate-for-remote-data-agent-server-private-key.key
├── SAS-SAS-certificates-file-name.zip
└── SAS-SAS-license-file-name.jwt
```

Here are the steps to create the directory structure and to copy or download the files:

- 1 Create the working directory and the **mysasdata/mydata** directories:

```
mkdir deploy
cd deploy
mkdir -m 777 mysasdata
mkdir -m 777 mysasdata/mydata
```

Note: In the documentation, **deploy** is the name of the working directory. The name of your working directory can be different.

- 2 Copy the following files to the **mysasdata** directory:

- The file (.pem) that contains the certificate authority (CA) list
- The server certificate file (.pem) and the private key file (.key)

Note: For more information about the preceding files, see [“How is TLS configured?” on page 13](#)

- 3 Copy the SAS license and certificates files to the deploy directory. These files are available in the software order at **My SAS** (my.sas.com), and they are the same license and certificates files that were used to deploy the SAS Viya platform. Here are the steps for downloading the files:

- a [Log in to my.sas.com](#).
- b Select **My Orders**, and then click the software order to display the order details.

- c Click **Downloads**. Next, click the **License** and **Certificates** check boxes for the version that you want to deploy, and then click the **Download** button. A ZIP file that includes the license and certificates is downloaded.
 - d Extract the license and certificates from the downloaded ZIP file to the deploy directory. The license is a JWT file, and the certificates are in a ZIP file. The files are used as is — do not unzip the certificates file.
- 4 Download the **SAS Container Manager** deployment tool, which is used to start the container:
- a Return to the software order at my.sas.com and click **My Orders**.
 - b Click the **Deployment Tools** button. Next, in the SAS Container Manager tile, select **Linux**, and then click the **Download** button. A compressed file is downloaded, and you can close the **Deployment Tools** window.
 - c Extract the files from the SAS Container Manager compressed file to the deploy directory:


```
tar xvf containermgr-linux.tgz
```

The SAS Container Manager executable is created.

- 5 Install the container image with the `container-manager install` command from the deploy directory:

```
./container-manager install --cadence-name "stable-or-lts" --cadence-version "version-number" --deployment-data SAS-certificates-file-name.zip sas-data-agent-server-remote
```

Note: The command pulls the container image from a SAS repository. Because of the large size of the image, it might take a few minutes to pull it. Also, the `da-vars.env` and `sas-access.properties` files are downloaded.

Here are descriptions for the cadence values:

"stable-or-lts"

The release cadence for the certificates that you downloaded from your software order. Use "stable" for the Stable cadence, or use "lts" for the Long-Term Support cadence.

"version-number"

The software version for the certificates that you downloaded from your software order (for example, "2026.04").

- 6 Copy the `da-vars.env` and `sas-access.properties` files to the deploy directory. Run the following command from the deploy directory:

```
cp ~/.sas-container-manager/sas-data-agent-server-remote/stable/files/* .
```

TIP To configure remote SAS Data Agent for your environment and to access a data source, you can edit the `da-vars.env` and `sas-access.properties` files, respectively. Because entries in these files are evaluated as internal to the container, any paths that are specified must be container paths. In the Quick Start, only the `da-vars.env` file is used.

Note: As updates become available, you can download a more recent container image, and then run the container with it. When downloading the container with a more recent container image,

the `da-vars.env` and `sas-access.properties` files in `~/sas-container-manager` are overwritten. Make sure that you copy the files to the deploy directory before updating the software.

- 7 For Security Enhanced Linux (SELinux) only: Set the SELinux context for files on the local file system that are referenced from inside the container. Here is an example in which the **chcon** tool is used to label the files and directories to be read by all containers:

```
chcon -t container_file_t SAS-SAS-license-file-name.jwt
chcon -t container_file_t mysasdata
chcon -t container_file_t mysasdata/mydata
chcon -t container_file_t mysasdata/list-of-trusted-certificate-authorities.pem
chcon -t container_file_t mysasdata/tls-certificate-for-remote-data-agent-server.pem
chcon -t container_file_t mysasdata/tls-certificate-for-remote-data-agent-server-private-key.key
```

Register Remote SAS Data Agent with the SAS Viya Platform

Here is the command to register remote SAS Data Agent with the SAS Viya platform:

```
./sas-viya dagentsrv servers register --name name-of-remote-data-agent-instance --address https://url-remote-data-agent-instance:port --context-root root-part-of-url --oauth-client OAuth-client-ID --oauth-secret OAuth-secret
```

Argument	Description
name	The name of the remote SAS Data Agent instance. This name can be specified with the <code>--data-agent</code> option.
address	The URL that is used to access remote SAS Data Agent from the SAS Viya platform. ■ The URL of the host machine or virtual machine that is running the container might be used, or the URL of the firewall or reverse proxy that sends requests from the SAS Viya platform to remote SAS Data Agent. ■ The default port is 25141, and it can be changed with the <code>container-manager --port</code> option when the container is started.
context-root	The root part of the URL that is used when the SAS Viya platform communicates with remote SAS Data Agent. Typically, the context-root is used by an on-premises reverse proxy to determine which incoming HTTPS requests are sent to remote SAS Data Agent.

Argument	Description
oauth-client	An OAuth client ID that you create.
oauth-secret	The secret for your OAuth client ID.

Note: If the container is uninstalled at a later time, remote SAS Data Agent can be deregistered before or after the container is uninstalled:

```
./sas-viya dagentsrv servers deregister --name name-of-remote-data-agent-instance
```

Configure Connectivity and Authentication with the SAS Viya Platform

Open the da-vars.env file in a text editor and replace the following highlighted variables with values that are appropriate to your environment, and then save the file.

```
SAS_DA_SERVICES_URL=https://name-of-ingress-host:port
SAS_DA_NAMESPACE=name-of-namespace
SAS_DA_OAUTH_CLIENT=OAuth-client-ID
SAS_DA_OAUTH_SECRET=OAuth-secret
SAS_DA_SSLCALISTLOC=/sasdata/list-of-trusted-certificate-authorities.pem
SAS_DA_SSLCERTLOC=/sasdata/tls-certificate-for-remote-data-agent-server.pem
SAS_DA_SSLPVTKEYLOC=/sasdata/tls-certificate-for-remote-data-agent-server-private-key.key
```

To run the container with Podman on Linux, the following environment variable must be set:

```
SAS_DA_VAULT_STORAGE=FILE
```

Note:

- The /sasdata path is a container path. When the container is started with the container-manager start command (next task), the mysasdata/ directory (on the host) is mounted as /sasdata inside the container.
- The values for **SAS_DA_OAUTH_CLIENT** and **SAS_DA_OAUTH_SECRET** are the same values that were used to register remote SAS Data Agent with the SAS Viya platform.
- For more information about the CA list file, and the TLS certificate and private key files, see [“How is TLS configured?” on page 13](#)

Run the Container

Run the container with the `container-manager start` command from the deploy directory:

```
./container-manager start --cadence-name "stable-or-lts" --cadence-version "version-number"
sas-data-agent-server-remote --license-path SAS-license-file-name.jwt --sasdata mysasdata --
vars da-vars.env --data mysasdata/mydata
```

Note: Use the same cadence values that were used for the `container-manager install` command.

TIP View the logs to confirm that the container started successfully:

```
docker logs sas-data-agent-server-remote
```

Download the dagentsrv Plug-in to the sas-viya CLI and Sign In

After the container is running successfully, the dagentsrv plug-in to the sas-viya CLI is installed on any machine that is used to access remote SAS Data Agent. Here is an overview of the steps and links to related documentation:

- 1 Install the sas-viya CLI. For more information, see the [Support / Downloads and Hot Fixes](#) page.
- 2 Download the dagentsrv plug-in to the sas-viya CLI. For more information, see “[Command-Line Interface: Instructions when Downloading the CLI](#)” in *SAS Viya Platform: Using the Command-Line Interface*.
- 3 Use the sas-viya CLI to sign in to the SAS Viya platform with a user that is a member of the **Data Agent Administrators** or **Data Agent Power Users** groups. For more information about the login credentials and using the dagentsrv plug-in, see “[Command Line Interface: dagentsrv](#)” in *Cloud Data Exchange for the SAS Viya Platform: Administrator's Guide*.

Example: Test Remote SAS Data Agent

Use the following example to perform a simple test of the connectivity between remote SAS Data Agent and the SAS Viya platform. This example assumes that the remote SAS Data Agent container is running, the dagentsrv plug-in to the sas-viya CLI is installed, and the SAS Viya platform is operational.

Perform the following steps using the dagentsrv plug-in to the sas-viya CLI:

8

- 1 Create a logical catalog for BASE data sets named BASECAT:

```
./sas-viya dagentsrv catalogs create base --name BASECAT --data-agent name-of-remote-data-agent-instance
```

- 2 Create a schema for the **data** directory named BASEDATA:

```
./sas-viya dagentsrv schemas create base --name BASEDATA --primary-path /data --catalog BASECAT --data-agent name-of-remote-data-agent-instance
```

- 3 Create a table named TEST:

```
./sas-viya dagentsrv sql --sql "create table TEST(col CHAR(10))" --dsn BASE --data-agent name-of-remote-data-agent-instance
```

- 4 Insert a row:

```
./sas-viya dagentsrv sql --sql "insert into TEST values ('success')" --dsn BASE --data-agent name-of-remote-data-agent-instance
```

- 5 Select from the new table:

```
./sas-viya dagentsrv sql --sql "select * from TEST" --dsn BASE --data-agent name-of-remote-data-agent-instance
```

The following output is displayed:

```
COL  
===  
success
```

- 6 Confirm that the data set was created in the correct location:

```
sudo ls -l mysasdata/mydata
```

Next Steps

- [“Configure Remote SAS Data Agent” on page 23](#)
- [“Configure Data Sources” on page 33](#)
- [“Migrate Data” on page 39](#)

Frequently Asked Questions

How does remote SAS Data Agent work?

Remote SAS Data Agent is a server that runs as a Docker on Linux container. After the container is running, the SAS Data Agent plug-in (dagentsrv plug-in) to the SAS Viya CLI (sas-viya CLI) is used to

perform data transfers securely from remote SAS Data Agent to the SAS Viya platform where co-located SAS Data Agent is configured.

What is SAS Container Manager?

SAS Container Manager is a command-line tool that automates the retrieval of the required container images and running the container. The latest release of a product is started unless a prior version release is specified by the user or has been previously installed. You can download SAS Container Manager from My SAS (my.sas.com).

What operating system is the container image based on?

The container image is based on Red Hat Universal Base Image (UBI) 9. Any additional third-party binary files that are mounted to the container must be compatible with Red Hat UBI 9. For example, libraries that are required for data access must be compatible with Red Hat UBI 9. Any attempt to mount third-party binaries that are based on Windows or another Linux operating system might not execute without errors.

Note: Prior to 2025.11, the container image is based on Red Hat UBI 8; and any files, libraries, or binaries that are mounted to the container must be compatible with Red Hat UBI 8.

Can I manage the container with Docker, Docker Compose, or Podman commands?

In this guide, the SAS Container Manager command `container-manager` is used to download the container image and to install, start, and stop the container. As an alternative, the `docker` commands that are used by SAS Container Manager to manage the container can be used instead. To display the `docker` commands that SAS Container Manager uses, run the `container-manager` command with the `--dry-run` option. To use Docker Compose or Podman, the `docker` commands can be converted to `docker compose` or `podman` commands, respectively.

For more information about the `--dry-run` option, see [“Examples: Output Docker Commands with the --dry-run Option” on page 49](#).

Note: SAS does not provide a Docker Compose file. To use Docker Compose with remote SAS Data Agent, you must create your own script.

How is access to data sources configured?

The `sas-access.properties.properties` file is used to pass environment variables to enable access to a data source such as Oracle, Microsoft SQL Server, and MongoDB. The `sas-access.properties` file is included as part of the SAS Container Manager download.

The environment variables that are used are the same ones that are used to configure a data source for the SAS Viya platform. For more information, see the README file that is provided with the SAS Viya platform deployment located at `$deploy/sas-bases/examples/data-access/README.md` (for Markdown) or `$deploy/sas-bases/docs/configuring_sasaccess_and_data_connectors_for_sas_viya_4.htm` (for HTML).

What network security configurations allow incoming requests to remote SAS Data Agent from the SAS Viya platform?

Remote SAS Data Agent is designed to run as a server within a secure on-premises network and to communicate with the SAS Viya platform over the internet. Without additional configuration or software, on-premises network security can block incoming requests to the remote SAS Data Agent server. For example, the use of a firewall or a reverse proxy can block a required port for the remote SAS Data Agent server. To allow incoming requests over the internet to remote SAS Data Agent, here are two examples to consider:

- Security rules can be created to route incoming requests to the remote SAS Data Agent server. In this example, the address and the port of the firewall or reverse proxy are registered with the SAS Viya platform. Also, the context root is used to create rules that route traffic to the remote SAS Data Agent server and TLS certificates are used to secure the connection.
- A VPN can be set up between the on-premises network and the SAS Viya platform deployment. In this example, a VPN can allow HTTPS requests from the SAS Viya platform to be sent to the remote SAS Data Agent server as if they were on the same network. The use of a VPN requires no additional configuration that is specific for the remote SAS Data Agent server.

IMPORTANT Setting up a VPN and other network security options that are not specific to SAS Data Agent are beyond the scope of this documentation. For more information, see the IT security administrator for your company or organization.

What is required for authentication?

You must provide an OAuth client ID and secret when you register remote SAS Data Agent with the SAS Viya platform. Also, the same OAuth client ID and secret must be recognized when the container is started. Here are the requirements:

- The OAuth client ID must be specified as a value for the `SAS_DA_OAUTH_CLIENT` environment variable in the `da-vars.env` file.
- To recognize the OAuth secret, two methods are supported:
 - You can specify the OAuth secret as the value for the `SAS_DA_OAUTH_SECRET` environment variable in the `da-vars.env` file.
 - You specify the OAuth secret with the `--oauth-secret OAuth-secret` option when the `container-manager sas-data-agent-server-remote` command is used to start the container. Using this method overrides any value that is specified for the `SAS_DA_OAUTH_SECRET` environment variable.

What options are available to store encrypted data?

Encrypted data such as credentials that are used to access data sources can be stored in the following ways:

- **(Default) Vault in the container:** By default, Hashicorp Vault (Vault) is used to store the encrypted data. When remote SAS Data Agent is started for the first time, a Vault database is initialized, and a related file is created in the directory mounted in the container: `/sasdata/vault/.vault.init`. Vault contains the credentials to access databases, and the `.vault.init` file stores the Vault keys that are used to access Vault. When credentials are stored on-premises, the encryption keys must be read each time the container starts. For more information, see [“How should Vault keys be read to start the container?” on page 12](#)
- **Customer-provided Vault server:** Encrypted data can be managed by a customer-provided Vault server that is recognized by the container. The container requires the address and token of the customer-provided Vault server that is provided by the Vault server administrator. Here are the two ways to configure the container to recognize the customer-provided Vault server at runtime:
 - Provide values for the `SAS_DA_VAULT_ADDRESS`, `SAS_DA_VAULT_SECRET_PATH`, and `SAS_DA_VAULT_TOKEN` environment variables in the `da-vars.env` file. For more information see, [“Customer-provided Vault Server: Environment Variables” on page 29](#).
 - Specify the `--vault-address`, `--vault-secret-path`, and `--vault-token` options when running the `container-manager start` command to start the container. For more information, see [“Customer-provided Vault Server: Options” on page 46](#).

Multiple remote SAS Data Agent servers can share a single customer-provided Vault server.

Note: Be aware that when a customer-provided Vault server is used, the Vault keys, storage, and directory options are ignored. Also, customers are responsible for the backup and restore of any data in a customer-provided Vault server.

- **With the SAS Viya platform deployment:** Encrypted data can be managed by the security that is configured for the SAS Viya platform deployment. To store credentials with the SAS Viya platform deployment, `--type=password` is specified when the security domain is created with the `dagentsrv` plug-in to the `sas-viya` CLI.

How should Vault keys be read to start the container?

By default, when the Vault server in the container is initialized, five Vault keys are generated and stored in the `/sasdata/vault/.vault.init` file. To unlock Vault and to initialize remote SAS Data Agent, three or more of the Vault keys must be read. The following methods are used to read the Vault keys each time the container is started:

- (Default) Directly from the `/sasdata/vault/.vault.init` file
- Vault keys are specified for the `SAS_DA_VAULT_KEYn` environment variables in the `da-vars.env` file.
- Vault keys are specified for the `container-manager start --vault-keyn` option.

To secure the Vault keys, use a combination of the methods. Here is an example:

- Vault key 1 is read directly from the `/sasdata/vault/.vault.init` file.
- Vault key 3 is specified as the value for the `SAS_DA_VAULT_KEY3` environment variable in the `da-vars.env` file.
- Vault key 5 is specified for the `container-manager start --vault-key5` option.
- Vault keys 2, 3, 4, and 5 are removed from the `/sasdata/vault/.vault.init` file.

Here is an example of the `container-manager start` command with the vault key information:

```
./container-manager start --cadence-name "stable-or-lts" --cadence-version "version-number"
sas-data-agent-server-remote --license-path SAS-license-file-name.jwt --vars da-vars.env --
vault-key5=vault-key-identifier
```

CAUTION

If the keys are lost and there is no way to recover them, the contents of Vault are lost.

For more information, see [“Configure Remote SAS Data Agent” on page 23](#) and [“SAS Container Manager Commands and Options” on page 43](#).

Can a customer-provided database server be used to store SYSCAT information?

Yes. By default, remote SAS Data Agent stores SYSCAT information in a PostgreSQL server that is in the container and that runs automatically when the container is started. As an alternative, customers can set environment variables so that the container recognizes a PostgreSQL server or a Oracle server that is provided by the customer. For more information, see the following:

- [“Customer-provided PostgreSQL Server: Environment Variables” on page 30](#)
- [“Customer-provided Oracle Server: Environment Variables” on page 32](#)

How is TLS configured?

Transport layer security (TLS) encryption is required to encrypt data-in-motion between remote SAS Data Agent and the SAS Viya platform with co-located SAS Data Agent. Here are the requirements to configure TLS encryption:

- Specify a value for the `SAS_DA_SSLCALISTLOC` environment variable in the `da-vars.env` file. The value is the directory path and the name of the file that contains the certificate authority (CA) list. The CA list must include the CA that is used for the SAS Viya platform and the CA for any other server that remote SAS Data Agent connects to via https.
- Use one of the following methods to connect to the SAS Viya platform:
 - (Recommended) Generate your own server certificate files for remote SAS Data Agent to use. To do this, specify values for the `SAS_DA_SSLCERTLOC` and `SAS_DA_SSLPVTKEYLOC` environment variables in the `da-vars.env` file. The values are for the TLS server certificate and the private key that you generate for remote SAS Data Agent. Also, the certificate authority that is used to generate the TLS server certificate must be added to the SAS Viya platform CA trust list.
 - Allow remote SAS Data Agent to generate and use self-signed server certificate files. If no value is specified for either `SAS_DA_SSLCERTLOC` or `SAS_DA_SSLPVTKEYLOC`, the following certificate files are generated in the `/sasdata` directory when the container is first started: `rda-cert.crt`, `rda-cert.key`, and `rda-ca.pem`. The `rda-cert.crt` file is the self-signed certificate for remote SAS Data Agent, and the `rda-cert.key` file is the private key that is required to use the self-signed certificate. To enable TLS, the `rda-ca.pem` file (CA certificate) must be added to the SAS Viya platform CA trust list.

For more information about the environment variables, see [“Configure Remote SAS Data Agent” on page 23](#).

How is mTLS configured?

Mutual Transport Layer Security (mTLS) is a security protocol that requires both parties in a connection to authenticate using certificates. Unlike standard Transport Layer Security (TLS), in which only the client validates the server certificate, mTLS requires the server to also validate the client certificate.

To use mTLS, configure it in remote SAS Data Agent and in each client that connects to remote SAS Data Agent. Supported clients include a CDE LIBNAME, a CDE data connector (CASlib), the Data Agent service. Also, the CDE CLI is supported for the use of the `security credentials create` command.

To configure mTLS, perform the following steps, and then restart the container:

- 1 Add the following line in the `da-vars.env` file:

```
SAS_DA_SSLCLIENTAUTH=true
```

Note: The SAS Data Agent service must be configured to provide client certificates to remote SAS Data Agent. For more information, see [“SAS Data Agent Service for mTLS” in *Cloud Data Exchange for the SAS Viya Platform: Administrator's Guide*](#).

- 2 If you have a separate SSL truststore for validating the client certificates, add the following line in the `da-vars.env` file:

```
SAS_DA_SSLCLIENTCALISTLOC=location-of-public-certificates-for-client-trusted-certificate-authorities
```

Note: If this environment variable is not set, remote SAS Data Agent validates client certificates using the CA list that is specified for the `SAS_DA_SSLCALISTLOC` environment variable.

For more information about the environment variables, see [“Configure Remote SAS Data Agent” on page 23](#).

How is logging configured?

By default, remote SAS Data Agent provides basic tracing. Additional levels of trace output can be specified in the `da-vars.env` file by setting `SAS_DA_DEBUG_LOGTYPE` to `Prefetch`, `Performance`, `Trace`, or `TraceAll`.

For more information, see [“Configure Remote SAS Data Agent” on page 23](#).

How do I determine if remote SAS Data Agent is running?

The following curl command detects if remote SAS Data Agent is running, and if there is a network path between the container and the host where the curl command is running.

```
curl https://registered-rda-host:registered-rda-port/registered-context-root/heartbeat
```

UP is returned, if remote SAS Data Agent is running.

Note: HTTPS is required.

Requirements for Remote SAS Data Agent

Host Machine

Docker Desktop on Linux

Minimum Requirements

- Red Hat Enterprise Linux 8.10 and RHEL 9.x
- To install Docker Desktop, see [Install Docker Desktop on RHEL](#).

Here are additional requirements to run the container:

- Basic knowledge of Linux commands is required.
- The Docker process runs as a user with root privileges. As a result, the user who performs the deployment requires sudoers privileges in order to run many of the required commands.

Note: Instead of granting sudoers privileges to any user who executes Docker commands, you can create a (UNIX) Docker group on the Docker host machine. Any users that are added to this group have Read/Write/Execute ownership of the Docker process, and are able to execute Docker commands without using sudo. For more information about the Docker group, see <https://docs.docker.com/install/linux/linux-postinstall/>.

Requirement for Docker SELinux

If Security Enhanced Linux (SELinux) is in use, additional configuration is required to access the directories and files of the remote SAS Data Agent container image. All directories and files on the local file system that are referenced from inside the container must have a security context (or label) that is recognized by the SELinux security policy that is in use. Here are the typical directories and files to label:

- The `/sasdata` directory, the `/sasdata/data` subdirectory, and any other subdirectories such as `/sasdata/backup`, `/sasdata/vault`, and `/sasdata/pgdata`
- The license file: `SAS-SAS-license-file-name.jwt`
- The server certificate, private key, and certificate authority files: `list-of-trusted-certificate-authorities.pem`, `tls-certificate-for-remote-data-agent-server.pem`, `tls-certificate-for-remote-data-agent-server-private-key.key`
- All directories and files that are referenced in the `da-vars.env` file
- All directories and files that are referenced in the `sas-access.properties` file
- All directories mounted on the container via the `--mount`, `--sasdata`, or `--data` options

Note: An example of labeling directories and files is provided in [Quick Start for Remote SAS Data Agent: Step 7 on page 5](#). The method that you use to label the directories and files can be different and are specific to the security policies in use. For more information about SELinux, see [Docker SELinux Security Policy](#) and [SELinux Contexts – Labeling Files](#).

Podman on Linux

Minimum Requirements

- Podman version 4.9.4-rhel or later on Red Hat Enterprise Linux 8.10 and RHEL 9.x and with the `podman-docker` package installed and configured:

```
dnf install podman-docker
```

The `podman-docker` package installs a script that emulates `docker` commands by using `podman` commands.

- An empty file named `/etc/containers/nodocker` must be created.

Note: If `/etc/containers/nodocker` does not exist, the following message is emitted and interferes with the use of SAS Container Manager to download the container image from the SAS repository: `Emulate Docker CLI using podman. Create /etc/containers/nodocker to quiet msg.`

- Basic knowledge of Linux commands is required.

Rootless Podman and Running the Container

By default, remote SAS Data Agent runs as container user ID (UID) 1001 and rootless Podman maps that UID to a UID from `/etc/subuid`. The mapped UID in `/etc/subuid` requires Read privileges and, in some cases, Write privileges for the directories and files that are accessed by remote SAS Data Agent.

As an alternative, remote SAS Data Agent can run as the user that starts the container. In the following example, the `keep-id` mode is used to run the container as the current user and to map the current user inside the container to the current user outside the container:

```
./container-manager --user $(id -u):$(id -g) --userns keep-id
```

For more information, see [Understanding rootless Podman's user namespace modes](#).

Hardware Requirements

Resource	Minimum Requirement
VMs	1
vCPUs	4 cores
RAM	1–4 GB The amount of RAM required depends on the workload and the number of concurrent users.
Disk Space	12–15 GB This value assumes that data is not stored in the file system inside the running container.

Kubernetes

For requirements and instructions to deploy remote SAS Data Agent to a Kubernetes cluster, see [“Deployment Example: Kubernetes” on page 18](#).

Connectivity Requirements

Firewall in Use

To download and install the container image from SAS when a firewall is in use, configure the firewall to enable outbound connections to `cr.sas.com` on port 443.

Required Connectivity between SAS Viya Platform and Remote SAS Data Agent

Here is a summary of the required network connections between the SAS Viya platform deployment, the remote SAS Data Agent server, and the SAS Viya CLI (sas-viya CLI).

- The sas-viya CLI must be able to send HTTPS requests to the SAS Viya platform deployment.
- The services in the SAS Viya platform deployment must be able to send HTTPS requests to the remote SAS Data Agent server.
- The remote SAS Data Agent server must be able to send HTTPS requests to the SAS Viya platform deployment.
- The sas-viya CLI must be able to send HTTPS requests to the remote SAS Data Agent server. This is required for the following operations:
 - The sas-viya CLI `security credentials --bypass-microservice` command that manages the shadow domains credentials.
 - The sas-viya CLI `sql --sql` command that sends SQL commands to the remote SAS Data Agent server.
 - The sas-viya CLI `servers test` command that tests whether the remote SAS Data Agent server is running and is reachable by the sas-viya CLI.

All other sas-viya CLI operations are sent to the SAS Viya platform deployment and do not require a network connection between the sas-viya CLI and the remote SAS Data Agent server.

Deployment Example: Kubernetes

Before You Begin

- This deployment example applies to a remote SAS Data Agent server that is deployed to a Kubernetes cluster by using a sample deployment YAML file. The steps to edit and apply the sample deployment YAML file are provided, as well as the typical steps to download the required files and access the container image. The sample deployment YAML file includes configuration for load balancing and fault tolerance. Your environment or site requirements might differ from the sample configuration that is provided.
- Multiple instances of remote SAS Data Agent can be used in parallel as long as they all share the same customer-provided PostgreSQL and Hashicorp Vault servers. The PostgreSQL and Vault servers that are in the container cannot be shared with more than one instance of remote SAS Data Agent.
- Requirements for the Kubernetes cluster:

- ❑ A supported version of Kubernetes. See the table of supported Kubernetes versions in [“Upgrading Kubernetes” in System Requirements for the SAS Viya Platform](#).

Note: For each of the supported cloud platforms, only the corresponding managed Kubernetes services are supported. For example, in a Microsoft Azure environment, only Microsoft Azure Kubernetes Service is supported. In AWS, only Amazon Elastic Kubernetes Service is supported.

- ❑ An OCI-compliant container runtime is required to run the container.
- ❑ Each node: 64-bit x86_64 chipset
- A client machine with a supported version of kubectl is required. The kubectl version can be only one minor version later or earlier than the version of Kubernetes (kube-apiserver) that is used in the cluster. For more information, see the [Kubernetes version skew policy](#).
- 12–15 GB of disk space is required on the worker nodes where the remote SAS Data Agent container image is pulled.
- Make sure that you have access to your software order at My SAS (my.sas.com). Your software order includes the certificates and the license file that are required to run the container.

Step 1 – Download the Required Files

Note: Remote SAS Data Agent runs as a Docker or Podman container in a supported Red Hat Enterprise Linux environment. Although Docker commands are shown in this guide, Podman commands are also supported. Knowledge about basic Linux commands is required, and sudoers privileges are assumed for the required commands.

- 1 Create a directory to save the downloaded files in.

```
mkdir directory-name
```

Note: In the documentation, the directory is referred to as \$deploy. The directory should be local to the host machine and not mounted via NFS.

- 2 Download the license and certificates to the \$deploy directory:

- a [Log in to my.sas.com](#).
- b Select **My Orders**, and then click the software order to display the order details.

TIP Record the order number for future use.

- c Click **Downloads**. Next, click the **License** and **Certificates** check boxes for the version that you want to deploy, and then click the **Download** button. A ZIP file that includes the license and certificates is downloaded.

- d Extract the license and certificates from the downloaded ZIP file to the deploy directory. The license is a JWT file, and the certificates are in a ZIP file. The files are used as is — do not unzip the certificates file.
- 3 Download SAS Container Manager. SAS Container Manager is a command-line tool that automates the retrieval of the required container image and running the container.
 - a Click **My Orders** to return to the My Orders page.
 - b Click the **Deployment Tools** button. . Next, in the **SAS Container Manager** tile, select **Linux**, and then click the **Download** button. A compressed file is downloaded, and you can close the **Deployment Tools** window.
 - c Extract the files from the SAS Container Manager compressed file to the deploy directory:


```
tar xvf containermgr-linux.tgz
```

The SAS Container Manager executable is created.

TIP SAS Container Manager is used to pull the container image directly from the SAS Container Registry (cr.sas.com) to a directory on your host machine.

Step 2 — Access the Container Image

- 1 Change to the \$deploy directory:


```
cd deploy
```
- 2 Install the container image with the `container-manager install` command from the deploy directory:


```
./container-manager install --cadence-name "stable-or-lts" --cadence-version "version-number" --deployment-data SAS-certificates-file-name.zip sas-data-agent-server-remote
```

Note: The command pulls the container image from a SAS repository. Because of the large size of the image, it might take a few minutes to pull it.

Here are descriptions for the cadence values:

"stable-or-lts"

The release cadence for the certificates that you downloaded from your software order. Use "stable" for the Stable cadence, or use "lts" for the Long-Term Support cadence.

"version-number"

The software version for the certificates that you downloaded from your software order (for example, "2026.02").

- 3 (Optional) View the container image repository:

```
docker images
```

Here is an example of the output:

REPOSITORY	IMAGE ID
TAG	
CREATED	SIZE
cr.sas.com/viya-4-x64_oci_linux_2-docker/sas-data-agent-server-remote	
0.47.30-20260310.1773153047307_sas-data-agent-server-remote_stable-2026.02	72aff051415f 4 weeks ago 5.41GB

Step 3 — Tag and Store the Container Image in a Local Docker Registry

For Kubernetes deployments, the use of a local Docker registry to tag and store the container image is required, and the registry must be accessible to your Kubernetes cluster.

Here are examples of tagging the pulled container image and pushing it to Docker registry's address:

```
docker tag cr.sas.com/viya-4-x64_oci_linux_2-docker/sas-data-agent-server-remote:0.47.30-20260310.1773153047307_sas-data-agent-server-remote_stable-2026.02 registry:5000/viya-4-x64_oci_linux_2-docker/sas-data-agent-server-remote:0.13.26-20260227.1692094990150
docker push registry:5000/viya-4-x64_oci_linux_2-docker/sas-data-agent-server-remote:0.13.26-20260227.1692094990150
```

Note: Enter each command on a single line.

For more information about using a local Docker registry, see the [Docker Documentation](#).

Step 4 — Copy and Edit the Sample Deployment YAML File

- 1 Copy the sample deployment YAML file to the \$deploy directory:

```
cp ~/.sas-container-manager/sas-data-agent-server-remote/stable/files/sas-data-agent-server-remote.yaml $deploy/.
```

- 2 Edit the \$deploy/sas-data-agent-server-remote.yaml file and replace all values that are enclosed by angle brackets <...> with values for your environment.

IMPORTANT Maintain the indentation of the sas-data-agent-server-remote.yaml file.

Here are the rules and some guidance for editing the sample deployment YAML file:

- All files, directories, and paths must be mounted in the container.
- All directories mounted in the container must be accessible by the container UID 1001.

- The values for `rda.SAS_DA_OAUTH_CLIENT` and `rda.SAS_DA_OAUTH_SECRET` are the same OAuth client ID and secret that are used when you register remote SAS Data Agent with the SAS Viya platform in a later step.
- The value for `rda.SASCNTANRLICENSEDIR` must be set to a directory that contains the license file and that directory must be mounted in the container. For example, the `/sasdata` directory can be used.
- `rda.SASLICENSEFILE` must be specified to the file name of the license file stored in the `rda.SASCNTANRLICENSEDIR` directory.
- Because a customer-provided PostgreSQL server is used in this example, all values for `rda.SAS_DA_PG` must be specified.
- Because a customer-provided Vault server is used in this example, all values for `rda.SAS_DA_VAULT` must be specified.
- Add any additional environment variables from the `da-vars.env` file or the `sas-access.properties` file that might be required.
- For `image:`, use the Docker registry, the path to the container image, and the container image name and tag. Here is an example:

```
registry/viya-4-x64_oci_linux_2-docker/sas-data-agent-server-
remote:0.13.22-20230815.1692094990150
```

- For `externalIPs:`, use the IP address of an externally routable node to enable access to remote SAS Data Agent outside of the cluster. The value must be added inside the brackets, such as `[IP-of-externally-routable-node]`. For more information about the value of the node, contact your Kubernetes administrator.
- For `replicas:` specify the number of concurrent remote SAS Data Agent pods that are used to provide fault tolerance and load balancing.
- The default `memory:` values in the sample deployment YAML file are appropriate for a single connection that accesses SAS datasets. More concurrent users or database clients that consume additional resources might require larger values.
- Session affinity is required so that operations requiring multiple requests are routed to the same pod. The example in the sample deployment YAML file uses `sessionAffinity: ClientIP` to achieve this but additional configuration might be needed when using a cluster ingress.
- The sample YAML file shows one way to mount an external NFS directory in the container as `/sasdata`. Remote SAS Data Agent requires that a persistent `/sasdata` directory is mounted in the container, but it does not have to use NFS.
- The sample YAML file mounts a non-persistent volume as `/data` that can be used for simple validation tests.
- Additional mounts should be added to provide access to external data and database clients as needed.

Step 5 – Deploy the Container Image

- 1 Deploy the container image:

```
kubectl apply -f sas-data-agent-server-remote.yml
```

- 2 Download the dagentsrv plug-in to the sas-viya CLI to any machine that is used to access remote SAS Data Agent. Here is an overview of the steps and links to related documentation:
 - a Install the sas-viya CLI. For more information, see the [Support / Downloads and Hot Fixes page](#).
 - b Download the dagentsrv plug-in to the sas-viya CLI. For more information, see “[Command-Line Interface: Instructions when Downloading the CLI](#)” in *SAS Viya Platform: Using the Command-Line Interface*.
 - c Use the sas-viya CLI to sign in to the SAS Viya platform with a user that is a member of the **Data Agent Administrators** or **Data Agent Power Users** groups. For more information about the login credentials and using the dagentsrv plug-in, see “[Command Line Interface: dagentsrv](#)” in *Cloud Data Exchange for the SAS Viya Platform: Administrator’s Guide*.
- 3 Register remote SAS Data Agent with the SAS Viya platform deployment:

```
./sas-viya dagentsrv servers register --name name-of-remote-data-agent-instance --address https://externally-routable-node:port --context-root root-part-of-url --oauth-client OAuth-client-ID --oauth-secret OAuth-secret
```

TIP The externally routable name and port (default 25141) are the values specified in the deployment YAML file.

- 4 Test the connection to the remote SAS Data Agent using the sas-viya CLI:

```
./sas-viya dagentsrv servers test -data-agent name-of-remote-data-agent-instance
```

Note: Additional information and examples about the container deployment is provided in the Quick Start guide. For more information, see “[Quick Start for Remote SAS Data Agent](#)” on page 2.

Configuration

Configure Remote SAS Data Agent

Specify Environment Variables

Environment variables that are used to configure remote SAS Data Agent are specified in the da-vars.env file. The da-vars.env file is included in the SAS Container Manager download and must be copied to a location that is readable by the container. The da-vars.env file includes syntax that is recognized by Docker, and each time the container is run, the --vars option is used to invoke the file.

Note: When SAS Container Manager is downloaded, the `da-vars.env` file is located in the `~/sas-container-manager/sas-data-agent-server-remote/stable/files/` directory.

- 1 Stop the container if it is running:

```
./container-manager stop sas-data-agent-server-remote
```

- 2 Open the `da-vars.env` file in a text editor and specify values for the environment variables that are appropriate to your environment, and then save the file. For a list of the environment variables, see [“Reference: da-vars.env” on page 24](#).

Note: Use a text editor that saves files with only the Linux end of line (EOL) character.

- 3 Run the container with the `container-manager start` command. Here is an example:

```
./container-manager start --cadence-name "stable-or-lts" --cadence-version "version-number"
sas-data-agent-server-remote --license-path SAS-license-file-name.jwt --sasdata mysasdata --
vars da-vars.env
```

TIP View the logs to confirm that the container started successfully:

```
docker logs sas-data-agent-server-remote
podman logs sas-data-agent-server-remote
```

Note: For more information about the correct syntax for environment variables that are recognized by Docker, see [Set environment variables](#) in the Docker documentation. For more information about the SAS Container Manager download, see [“Quick Start for Remote SAS Data Agent” on page 2](#).

Reference: da-vars.env

TIP Consult with the SAS Viya platform administrator or Kubernetes administrator to obtain the values for environment variables. Also, entries in the `da-vars.env` file are evaluated inside the container. Any directories or files that are specified must be in terms of paths that are mounted on the container.

General Environment Variables

Environment Variable	Value	Notes
<code>SAS_DA_SERVICES_URL</code>	The base URL of the SAS Viya platform deployment with co-located SAS Data Agent configured.	Required

Environment Variable	Value	Notes
SAS_DA_NAMESPACE	The name of the namespace of the SAS Viya platform deployment.	Required
SAS_DA_OAUTH_CLIENT	An OAuth client ID that you create. The OAuth client ID must be the same one that is used to register remote SAS Data Agent with the SAS Viya platform.	Required When the container is started, the OAuth client ID and secret are used to authenticate remote SAS Data Agent with the SAS Viya platform.
SAS_DA_OAUTH_SECRET	The OAuth secret that is used with the OAuth client ID that you create. The OAuth client secret must be the same one that is used to register remote SAS Data Agent with the SAS Viya platform.	Optional You can specify the OAuth secret as the value for SAS_DA_OAUTH_SECRET , or you can use the <code>--oauth-secret OAuth-secret</code> option when the <code>container-manager sas-data-agent-server-remote</code> command is used to start the container. For more information, see “What is required for authentication?” on page 11
SAS_DA_SSLCALISTLOC	The directory path and the name of the file that contains the TLS certificate authority (CA) list.	Required The CA list includes the certificates of servers that remote SAS Data Agent allows a connection to.
SAS_DA_TENANTID	The assigned tenant ID for the SAS Viya platform deployment that is running as a multi-tenant environment.	Required if the SAS Viya platform is deployed as a tenant in a multi-tenant environment.
SAS_DA_SSLPVTKEYLOC	The directory path and the name of the file that contains the private key for the TLS server certificate.	Not required but recommended. If no value is specified, self-signed certificate files (<code>rda-ca.pem</code> , <code>rda-cert.crt</code> , <code>rda-cert.key</code>) are generated when the container is started.
SAS_DA_SSLCERTLOC	The directory path and the name of the file that contains the TLS server certificate that is used for remote SAS Data Agent.	Not required but recommended. If no value is specified, self-signed certificate files (<code>rda-ca.pem</code> , <code>rda-cert.crt</code> , <code>rda-cert.key</code>) are generated when the container is started.

Environment Variable	Value	Notes
<code>SAS_DA_SSLSELF CERTDIR</code>	The directory where the self-signed certificate files are written to if they are created.	The default value is <code>/data</code> . The directory must be writable by container uid 1001.
<code>SAS_DA_SSLREQCERT</code>	A value that indicates whether remote SAS Data Agent validates TLS certificates. Values can be <code>ALLOW</code> , <code>DEMAND</code> , <code>NEVER</code> , or <code>TRY</code> .	The default value is <code>DEMAND</code> .
<code>SAS_DA_SSLCLIENTAUTH</code>	A value that indicates if mTLS is in use. Set to <code>true</code> to configure mTLS for use with remote SAS Data Agent.	Required if mTLS is configured.
<code>SAS_DA_SSLCLIENTCALISTLOC</code>	The location of public certificates for client-trusted certificate authorities.	Required if mTLS is configured and a separate SSL truststore for validating the client certificates. If this environment variable is not set, remote SAS Data Agent validates client certificates using the truststore that is specified for the <code>SAS_DA_SSLCALISTLOC</code> environment variable.
<code>SAS_DA_CONTENT_ROOT</code>	The directory structure that file access drivers are allowed to access. Example: <code>SAS_DA_CONTENT_ROOT=/accounting/data</code>	The default value is <code>/data</code> . This value is ignored if <code>SAS_DA_RESTRICT_CONTENT_ROOT=FALSE</code> .
<code>SAS_DA_RESTRICT_CONTENT_ROOT</code>	A value that indicates the access to directories by the access drivers. Values can be <code>TRUE</code> or <code>FALSE</code> .	The default value is <code>TRUE</code> . If set to <code>TRUE</code> the file access drivers are able to access only the directory structure specified by <code>SAS_DA_CONTENT_ROOT</code> . If set to <code>FALSE</code> the file access drivers are able to access only the directories that are accessible from within the remote SAS Data Agent container.
<code>SAS_DA_SERVICES_PROXY_SERVER</code>	The name of the proxy server that enables remote SAS Data Agent to communicate with the SAS Viya platform.	

Environment Variable	Value	Notes
<code>SAS_DA_SERVICES_PROXY_PORT</code>	The name of the proxy server port that enables remote SAS Data Agent to communicate with the SAS Viya platform.	
<code>SAS_DA_SERVICES_PROXY_USER</code>	The user name for the proxy server that enables remote SAS Data Agent to communicate with the SAS Viya platform.	
<code>SAS_DA_SERVICES_PROXY_PASS</code>	The password for the proxy server that enables remote SAS Data Agent to communicate with the SAS Viya platform.	
<code>SAS_DA_SAS_APP_NAME</code>	The application name of the remote SAS Data Agent. This is used as the host name of the container.	Typically, the name is data-agent-server-remote and should not be changed.
<code>SAS_DA_INSTANCE_ID</code>	The server instance name used to register SAS Data Agent.	
<code>SAS_DA_LOGOUTTYPE</code>	The format of the log output. Values can be JSON or TEXT .	The default is JSON .
<code>SAS_DA_DEBUG_LOGTYPE</code>	The level of additional logging. Values can be one of the following: TRACEALL to enable trace level for all log items. TRACE to enable trace level for most log items. TRACEAPI to enable trace level for common SQL API calls. PERFORMANCE to enable trace and debug level items for performance debugging. PREFETCH to enable trace and debug level items for prefetch debugging. NONE to disable additional tracing.	The default is NONE .
<code>SAS_DA_BACKUP_DIR</code>	The directory used to store backup files that are created by dagentsrv CLI backup command.	The default is /sasdata/backup . If a different directory is specified, the directory must be accessible from inside the container by user UID 1001.
<code>SAS_DA_VAULT_ROOT_DIR</code>	The directory used to persist Vault data containing encrypted credentials.	The default is /sasdata/vault .

Environment Variable	Value	Notes
<code>SAS_DA_VAULT_STORAGE</code>	A value that indicates how Vault stores encrypted data on disk. Values can be one of the following: RAFT to enable Vault Integrated Storage for credentials data. For more information about Vault Integrated Storage, see Integrated storage (Raft) backend. FILE to enable Vault Filesystem Storage for credentials data. This option requires additional steps to back up and restore Vault data. For more information about Vault Filesystem Storage, see File system configuration for Vault backend storage.	The default value is RAFT. To run the container with Podman on Linux, FILE is required. This value is ignored if <code>SAS_DA_VAULT_ADDRESS</code> and <code>SAS_DA_VAULT_TOKEN</code> are set.
<code>SAS_DA_VAULT_INIT_DIR</code>	The directory that contains the .vault.init file.	The default is <code>/sasdata/vault</code> .
<code>SAS_DA_VAULT_KEY1</code> <code>SAS_DA_VAULT_KEY2</code> <code>SAS_DA_VAULT_KEY3</code> <code>SAS_DA_VAULT_KEY4</code> <code>SAS_DA_VAULT_KEY5</code>	Vault keys that were stored in the .vault.init file when Vault was initialized.	Vault keys can be provided in the .vault.init file, the da-vars.env file, or when running the container-manager start command. At least three of the Vault keys are required to start the container. For more information, see “How should Vault keys be read to start the container?” on page 12
<code>SAS_DA_HADOOP_JAR_PATH</code> <code>SAS_DA_HADOOP_CONFIG_PATH</code>	The path to Hadoop JAR and configuration files. Example: <pre>SAS_DA_HADOOP_CONFIG_PATH=/clients/hadoopconfig/prod SAS_DA_HADOOP_JAR_PATH=/clients/jdbc/spark/2.6.22</pre>	Required for SAS/ACCESS Interface to Spark. See the SAS/ACCESS Interface to Spark documentation for more information.
<code>SAS_DA_JAVA_HOME</code>	An alternative location for <code>JAVA_HOME</code> for use by remote SAS Data Agent. Example: <code>SAS_DA_JAVA_HOME=/java/lib/jvm/jre</code>	Optional
<code>SAS_DA_LD_LIBRARY</code>	Additional directories to add to the <code>LD_LIBRARY_PATH</code> environment variable in order to access additional shared libraries.	Optional

Environment Variable	Value	Notes
<code>SAS_DA_SESSION_CLEANUP</code>	A value for how often to check for and kill idle sessions.	The default value is 0. The default value 0 equals 60 seconds and specifies that the check is performed once every minute.
<code>SAS_DA_DEFAULT_SESSION_TIMEOUT</code>	A value for the time to wait before an unused session is considered idle and therefore eligible to be killed.	The default value is 3600. The default value 3600 equals one hour. This value is used only when the <code>SESSION_TIMEOUT</code> connection option is not specified when connecting to the remote SAS Data Agent server.
<code>SAS_DA_MAX_SESSION_TIMEOUT</code>	A value for the maximum time to wait before an unused session is considered idle and therefore eligible to be killed.	The default value 0 specifies that there is no time-out. This maximum time limit is enforced even if <code>SAS_DA_DEFAULT_SESSION_TIMEOUT</code> or <code>SESSION_TIMEOUT</code> are longer.
<code>SAS_DA_MAX_OBJECT_TIMEOUT</code>	A value for how long to wait for a database object to complete during idle cleanup. Examples of a database objects are Connection, Statement, or ResultSet.	The default value 0 specifies that there is no time-out.
<code>SAS_DA_UMASK</code>	An octal value that defines default permission restrictions for newly created files and directories. Example: <code>SAS_DA_UMASK=002</code>	Optional

Customer-provided Vault Server: Environment Variables

By default, the container uses the Vault server in the container to store encrypted data. To use a customer-provided Vault server instead, set the following environment variables.

Environment Variable	Value	Notes
<code>SAS_DA_VAULT_ADDRESS</code>	The fully qualified address and port of the customer-provided Vault server, such as <code>https://customer-vault-server-address:customer-vault-port</code> .	If no value is specified, the Vault server in the container is used. If a customer-provided Vault server is used, customers are responsible for the backup and restore of data.

Environment Variable	Value	Notes
	For more information, contact the Vault server administrator.	Multiple remote SAS Data Agent servers can share a single customer-provided Vault server.
<code>SAS_DA_VAULT_SECRET_PATH</code>	The path where secrets in the customer-provided Vault server are written. For more information, contact the Vault server administrator.	The default is <code>/secret</code> .
<code>SAS_DA_VAULT_TOKEN</code>	The authentication token for the customer-provided Vault server. For more information, contact the Vault server administrator.	

Note: As an alternative to setting the environment variables in the `da-vars.env` file, the address and the authentication token can be specified with the `container-manager` command and the `--vault-address`, `--vault-secret-path`, and `--vault-token` options.

Customer-provided PostgreSQL Server: Environment Variables

By default, the container uses the PostgreSQL server in the container to store SYSCAT information. To use a customer-provided PostgreSQL server instead, set the following environment variables.

Environment Variable	Value	Notes
<code>SAS_DA_PG_HOST</code>	The IP address or host name of a customer-provided PostgreSQL server that is used to store SYSCAT information. For more information, contact the PostgreSQL server administrator.	If no value is specified, the PostgreSQL server in the container is used. If a customer-provided PostgreSQL server is used, customers are responsible for the backup and restore of data.
<code>SAS_DA_PG_PORT</code>	The port of a customer-provided PostgreSQL server. For more information, contact the PostgreSQL server administrator.	If no value is specified, the default value <code>5432</code> is used.
<code>SAS_DA_PG_UID</code>	The user ID that is used to connect to a customer-provided PostgreSQL server. For more information, contact the PostgreSQL server administrator.	This environment variable is ignored if the PostgreSQL server in the container is used.
<code>SAS_DA_PG_PWD</code>	The password that is used to connect to a customer-provided PostgreSQL server. For more information, contact the PostgreSQL server administrator.	Required if a customer-provided PostgreSQL server is used to store SYSCAT data.

Environment Variable	Value	Notes
		This environment variable is ignored if the PostgreSQL server in the container is used.
<code>SAS_DA_PG_SYSCAT_DB</code>	The database that is used to store SYSCAT information in a customer-provided PostgreSQL server. For more information, contact the PostgreSQL server administrator.	If no value is specified, the default value <code>DA_SYSCAT_default</code> is used. If the database does not exist, it is created when the container starts as long as the user has sufficient privileges. For more information, contact the PostgreSQL server administrator.
<code>SAS_DA_PG_SYSCAT_SCHEMA</code>	The schema that is used to store SYSCAT information in a customer-provided PostgreSQL server. For more information, contact the PostgreSQL server administrator.	If no value is specified, the default value <code>FS_SYSCAT</code> is used. If the schema does not exist, it is created when the container starts, as long as the user has sufficient privileges. For more information, contact the PostgreSQL server administrator. Multiple remote SAS Data Agent servers can share a single customer-provided PostgreSQL server as long as separate schemas are specified.
<code>SAS_DA_FIPS_MODE</code>	A value to restrict encryption of SYSCAT data in the remote (co-located) SAS Data Agent to FIPS 140-3 compliant algorithms. Values can be TRUE or FALSE.	If no value is specified, the default value FALSE is used. To enable FIPS 140-3 compliant algorithms, set to TRUE.

Note:

- Multiple remote SAS Data Agent servers can share the same customer-provided PostgreSQL server database and schema as long as they share the same customer-provided Vault server.
 - As an alternative to setting the environment variables in the `da-vars.env` file, the values for the host name, port, user, and password can be specified with the `container-manager` command and the `--pg-host`, `--pg-port`, `--pg-uid`, and `--pg-pwd` options.
-

Customer-provided Oracle Server: Environment Variables

By default, the container uses the PostgreSQL server in the container to store SYSCAT information. To use a customer-provided Oracle server instead, set the following environment variables.

Environment Variable	Value	Notes
<code>SAS_DA_ORA_PATH</code>	The connection information that is used to connect to the customer-provided Oracle server. For more information, contact the Oracle server administrator.	Example: <code>SAS_DA_ORA_PATH=host-name:port/service-name</code> For more information, contact the Oracle server administrator.
<code>SAS_DA_ORA_UID</code>	The user ID that is used to connect to a customer-provided Oracle server. For more information, contact the Oracle server administrator.	Required if a customer-provided Oracle server is used to store SYSCAT data. The user ID is not created automatically and must exist before starting the data agent server remote container. Multiple remote SAS Data Agent servers can share a single customer-provided Oracle server as long as separate user IDs are specified. This environment variable is ignored if the PostgreSQL server in the container is used.
<code>SAS_DA_ORA_PWD</code>	The password that is used to connect to a customer-provided Oracle server. For more information, contact the Oracle server administrator.	Required if a customer-provided Oracle server is used to store SYSCAT data. This environment variable is ignored if the PostgreSQL server in the container is used.
<code>SAS_DA_FIPS_MODE</code>	A value to restrict encryption of SYSCAT data in the remote (co-located) SAS Data Agent to FIPS 140-3 compliant algorithms. Values can be TRUE or FALSE .	If no value is specified, the default value FALSE is used. To enable FIPS 140-3 compliant algorithms, set to TRUE .

Note:

- Multiple remote SAS Data Agent servers can share the same customer-provided Oracle server as long as they share the same customer-provided Vault server.

- As an alternative to setting the environment variables in the `da-vars.env` file, the values for the host name, port, user, and password can be specified with the `container-manager` command and the `--ora-path`, `--ora-uid`, and `--ora-pwd`.

Configure Data Sources

To configure some SAS/ACCESS products, environment variables must be specified in the `sas-access.properties` file. The `sas-access.properties` file is included in the SAS Container Manager download and must be copied to a location that is readable by the container. The `sas-access.properties` file includes syntax that is recognized by Docker, and each time the container is run, the `--access-vars` option is used to invoke the file.

Note: The values in the `sas-access.properties` file are examples of environment variables that can be set for some database clients. As needed, you can add other environment variables that are required for the database client that you want to use. Also, some supported data sources do not require the use of the `sas-access.properties` file. Contact your database administrator for more information.

Here is an example of using the `sas-access.properties` file:

- 1 Stop the container if it is running:

```
./container-manager stop sas-data-agent-server-remote
```

- 2 Open the `sas-access.properties` file in a text editor. Next, uncomment the lines for the data source that you want to use and provide values for the environment variables. Here is an example for Oracle:

Note: When SAS Container Manager is downloaded, the file is located in the `~/.sas-container-manager/sas-data-agent-server-remote/stable/files/` directory.

```
#####
# SAS/ACCESS to Oracle
#####
ORACLE=/dbclients/oracle/19c/lib
ORACLE_HOME=/dbclients/oracle/19c
ORACLE_BIN=/dbclients/oracle/19c/bin
TNS_ADMIN=/dbclients/oracle/19c/network/admin/tnsnames.ora
```

Note:

- Use a text editor that saves files with only the Linux end of line (EOL) character.
- All files and paths must use container-relative paths that are mounted on the container using the `--mount`, `--sasdata`, or `--data` options. For example, to mount the Oracle client directory on the container specify `--mount /net/installed/dbms/oracle:/oracle` and then reference the Oracle client using the mounted directory `ORACLE=/dbclients/oracle/19c/lib`.

- 3 Save the file. Then, run the container with the `container-manager start` command and make sure to include the `--access-vars` option. Here is an example:

```
./container-manager start --cadence-name "stable-or-lts" --cadence-version "version-number"
sas-data-agent-server-remote --license-path SAS-license-file-name.jwt --sasdata mysasdata --
vars da-vars.env --access-vars sas-access.properties
```

TIP View the logs to confirm that the container started successfully:

```
docker logs sas-data-agent-server-remote
```

Note:

- For more information about the correct syntax for environment variables that are recognized by Docker, see [Set environment variables](#) in the Docker documentation.
 - For more information about the SAS Container Manager download, see [“Quick Start for Remote SAS Data Agent” on page 2](#).
-

Software Updates

Run the Container with a New Release

As updates become available, you can download and run a newer container image. The latest software release includes the most current enhancements, such as new or updated features, security updates, and bug fixes. The certificates and license from a software order are required to run the container, and software orders can be found at My SAS (my.sas.com). To apply a new license only, see [“Run the Container with a New License” on page 35](#).

IMPORTANT If you update the SAS Viya platform to a new release, you must also update the remote SAS Data Agent container image to the same cadence version. Running different cadence versions of remote SAS Data Agent and the SAS Viya platform is not covered by Standard Support. For information about releases, see [“Release Schedule and Versions” in Getting Started with SAS Viya Platform Operations](#).

- 1 When downloading the container with a more recent container image, the `da-vars.env` and `sas-access.properties` files in `~/sas-container-manager` are overwritten. Make sure to copy the files to the deploy directory before updating the software.
- 2 Download the certificates for the release that you want to deploy. For more information, see the initial steps in [“Quick Start for Remote SAS Data Agent” on page 2](#).

- 3 Run the following container-manager install command from the deploy directory:

```
./container-manager install --cadence-name "stable-or-lts" --cadence-version "version-number" --deployment-data SAS-certificates-file-name.zip sas-data-agent-server-remote
```

Note: The command pulls the container image from a SAS repository. Because of the large size of the image, it might take a few minutes to pull it. Also, the da-vars.env and sas-access.properties files are downloaded.

Here are descriptions for the cadence values:

"stable-or-lts"

The release cadence for the certificates that you downloaded from your software order in [Step 2 on page 34](#). Use "stable" for the Stable cadence, or use "lts" for the Long-Term Support cadence.

"version-number"

The software version (for example, "2026.01") for the certificates that you downloaded from your software order in [Step 2 on page 34](#).

- 4 Stop the container and remove it.

```
./container-manager stop sas-data-agent-server-remote
./container-manager uninstall sas-data-agent-server-remote
```

- 5 Run the container with the new release:

```
./container-manager start --cadence-name "stable-or-lts" --cadence-version "version-number" sas-data-agent-server-remote --license-path SAS-license-file-name.jwt --sasdata mysasdata --vars da-vars.env --data mysasdata/mydata
```

Note: Use the same cadence values that were used for the container-manager install command.

Run the Container with a New License

These instructions are for running the container with a new license that is downloaded from your software order at My SAS (my.sas.com). As a best practice, save your current license file before you begin.

- 1 [Log in to my.sas.com](#).
- 2 Select **My Orders**, and then click the software order to display the order details.
- 3 Click **Downloads**. Next, click the **License** check box for the version that is currently deployed, and then click the **Download** button. A JWT file is downloaded.
- 4 Stop the container:

```
./container-manager stop sas-data-agent-server-remote
```

TIP Here is the command to see whether the container is running:

```
docker ps
```

- 5 Copy the license file to the **deploy** directory. Here is a command-line example:

```
cp new-SAS-license-file-name.jwt
```

- 6 Modify permissions for the license file:

```
chmod 0644 new-SAS-license-file-name.jwt
```

- 7 Run the container with the new license:

```
./container-manager start --cadence-name "stable-or-lts" --cadence-version "version-number"  
sas-data-agent-server-remote --license-path new-SAS-license-file-name.jwt
```

Back Up and Restore

What Is Backed Up?

- System catalog (SYSCAT) information about data services, privileges, catalogs, and schemas can be backed up. By default, SYSCAT information is stored in the PostgreSQL server that is in the container.
- Encrypted data, such as credentials that are used to access data sources, can be backed up. By default, encrypted data is stored in the version of Hashicorp Vault (Vault) that is in the container, and Integrated Storage (Raft) is used. Filesystem Storage (File) is another storage option. Customers are responsible for backing up and restoring credentials when File is used.

Note: File is required if Podman on Linux is used to run the container. To specify the type of storage, the **SAS_DA_VAULT_STORAGE** environment variable is set in the `da-vars.env` file. For more information, see [“Reference: da-vars.env” on page 24](#).

- Customers are responsible for backing up and restoring data if customer-provided storage is used. As an alternative to the storage systems in the container, a customer-provided database server and a customer-provided Vault server can be used to store SYSCAT information and Vault data, respectively. For more information, see the FAQs: [“Can a customer-provided database server be used to store SYSCAT information?” on page 13](#) and [“What options are available to store encrypted data?” on page 11](#)

TIP A best practice is to back up and restore SYSCAT data and Vault data at the same time.

Back Up and Restore from the Storage Systems in the Container

The following tasks assume that data is stored in the PostgreSQL server and in the Vault server that are in the container. Steps for using Raft and File for the Vault storage are provided. Vault storage is set by the `SAS_DA_VAULT_STORAGE` environment variable in the `da-vars.env` file, and `RAFT` is the default value.

Note: To perform the tasks, the remote SAS Data Agent container must be registered and operational. Because the `dagentsrv` CLI is used, the user must be signed in to the host machine that is used for remote SAS Data Agent.

To back up data:

- 1 Run the `dagentsrv` CLI backup command:

```
./sas-viya dagentsrv security credentials backup
```

The backup command creates two files or one file, depending on whether Raft or File is used.

- **Raft** Two files are created:

- The SYSCAT backup file is created: `/sasdata/backup/sas-da-syscat.timestamp`
- Vault backup file: `/sasdata/backup/sas-da-vault.timestamp`

- **File** The SYSCAT backup file is created: `/sasdata/backup/sas-da-syscat.timestamp`

Note: The backup files are created outside the container. The directory path `/sasdata/backup` that is shown is the default value for the `SAS_DA_BACKUP_DIR` environment variable in the `da-vars.env` file. Also, the timestamp extension that is appended to each file name indicates when the backup request was received.

- 2 Copy the SYSCAT backup file `/sasdata/backup/sas-da-syscat.timestamp` to a secure location.
- 3 Perform one of the following steps to copy the Vault backup to a secure location:
 - **Raft** Copy the Vault backup file `/sasdata/backup/sas-da-vault.timestamp` to a secure location.
 - **File** Copy the contents of the `/sasdata/vault` directory to a secure location.

IMPORTANT Users are responsible for securing the backup files. It is recommended that the container user UID 1001 be restricted to prevent unauthorized access. Backup files should be backed up and preserved in a durable location based on your company policies.

To restore the backed up data:

- 1 Stop the container if it is running:

```
./container-manager stop sas-data-agent-server-remote
```

- 2 Copy the contents of the SYSCAT backup file from the secure location to **/sasdata/backup/sas-da-syscat.restore**.

Here is an example command:

```
cp secure-location/sas-da-syscat.timestamp mysasdata/backup/sas-da-syscat.restore
```

Note: The **sas-da-syscat.restore** file must be readable and writable by the container user UID 1001.

- 3 Perform one of the following steps to restore with the Vault backup:

- **Raft** Copy the contents of the Vault backup file from the secure location to **/sasdata/backup/sas-da-vault.restore**.

Here is an example command:

```
cp secure-location/sas-da-vault.timestamp mysasdata/backup/sas-da-vault.restore
```

Note: The **sas-da-vault.restore** file must be readable and writable by the container user UID 1001.

- **File** Copy the previously backed up Vault files from the secure location to the **/sasdata/vault** directory.

- 4 Restart the container. When the container is started successfully, the SYSCAT data and Vault data are restored, and the **.restore** files are erased. Here is an example of restarting the container with the `container-manager start` command:

```
./container-manager start --cadence-name "stable-or-lts" --cadence-version "version-number"
sas-data-agent-server-remote --license-path SAS-license-file-name.jwt --sasdata mysasdata --
vars da-vars.env
```

TIP View the logs to confirm that the container started successfully and that the **.restore** files are deleted:

```
docker logs sas-data-agent-server-remote
```

Migrate Data

Before You Begin

- This topic provides the steps to migrate data from remote SAS Data Agent for SAS Viya 3.5 to remote SAS Data Agent for SAS Viya 4. As needed, the version numbers are used to indicate the different environments, such as version 3.5 and version 4.
- Data that can be migrated is stored in two places:
 - **SYSCAT:** The system catalog (SYSCAT) stores information about data services, privileges, catalogs, and schemas. By default, SYSCAT information is migrated to the PostgreSQL server that is in the container for remote SAS Data Agent for SAS Viya 4.
 - **Vault:** Hashicorp Vault is used to store shadow domains and encrypted credentials on-premises. By default, encrypted data is migrated to the version of Hashicorp Vault (Vault) that is in the container for remote SAS Data Agent for SAS Viya 4.
- Customers are responsible for migrating data if customer-provided storage is used for remote SAS Data Agent for SAS Viya 4. For more information, see the FAQs: [“Can a customer-provided database server be used to store SYSCAT information?” on page 13](#) and [“What options are available to store encrypted data?” on page 11](#)
- Migrating SYSCAT information from remote SAS Data Agent for SAS Viya 3.5 to a customer-provided Oracle server for remote SAS Data Agent for SAS Viya 4 is not supported.

Step 1: Export the SYSCAT Information to a Backup File

- 1 Sign in to the host machine that is running remote SAS Data Agent for SAS Viya 3.5, and change to a user ID that has Read access to the SAS Viya installation files.

Note: User ID `sas` is recommended.

- 2 Run the following `export` commands to get the PostgreSQL credentials, catalog, and schema.

Note: The values for `SASROOT` and `SASTENANT` are the default values for the SAS Viya deployment. Change these values to match your environment.

```
export SASROOT=/opt/sas/viya
export SASTENANT=
```

```
export CONSUL_HTTP_TOKEN=$(sudo cat ${SASROOT}/${SASTENANT}/config/etc/
SASSecurityCertificateFramework/tokens/consul/default/client.token)
export PGUSER=$((${SASROOT}/home/bin/consul kv get config/application/postgres/
username)
export PGPASSWORD=$((${SASROOT}/home/bin/consul kv get config/application/postgres/
password)
export PGCATALOG=$(grep cfg.pg.syscatdb ${SASROOT}/${SASTENANT}/config/etc/dagentsrv/
default/da_entities.dtd | cut -d '"' -f 2)
export PGSHEMA=$(grep cfg.pg.syscat.schema ${SASROOT}/${SASTENANT}/config/etc/
dagentsrv/default/da_entities.dtd | cut -d '"' -f 2)
```

- 3 Specify a location to write the SYSCAT output:

Note: The user ID must have Write access to the specified location.

```
export SYSCAT_BACKUP_FILE=/tmp/sas-da-syscat.restore
```

- 4 Write the contents of the SYSCAT to a backup file:

```
${SASROOT}/home/postgresql11/bin/pg_dump --dbname=$PGCATALOG --schema=\"$PGSCHEMA\" --file=
$SYSCAT_BACKUP_FILE
```

- 5 Replace the catalog and schema with generic identifiers:

Note: Because the database and schema might be different between the version 3.5 and version 4 environments, generic identifiers can enable the successful migration of data.

```
sed -i "s/\${PGCATALOG}/\"_SYSCAT_CATALOG_\"/g" $SYSCAT_BACKUP_FILE
sed -i "s/\${PGSCHEMA}/\"_SYSCAT_SCHEMA_\"/g" $SYSCAT_BACKUP_FILE
```

Step 2: Restore the SYSCAT Backup

- 1 Sign in to the host machine that is used for remote SAS Data Agent for SAS Viya 4.

- 2 Stop the container, if it is running:

```
./container-manager stop sas-data-agent-server-remote
```

- 3 Make sure that the `/sasdata/backup` subdirectory exists and that container UID 1001 has Read and Write access to it.

- 4 Copy the `sas-da-syscat.restore` file to the `/sasdata/backup` directory.

Note: The directory must be readable and writable by the container user ID (UID 1001).

- 5 Start the container to restore the SYSCAT backup:

```
./container-manager start --cadence-name "stable-or-lts" --cadence-version "version-number"
sas-data-agent-server-remote --license-path SAS-license-file-name.jwt --sasdata mysasdata --
vars da-vars.env
```

- Review the log output for the following lines to determine whether the restore was successful:

```
"message": "Attempting to restore SYSCAT."
...
"message": "Restore of SYSCAT successful."
```

Note: The `/sasdata/backup/sas-da-syscat.restore` file is deleted if the restore was successful.

Step 3: Migrate Credentials

- Return to the host machine that is running remote SAS Data Agent for SAS Viya 3.5. You must be signed in and using a user ID that has Read access to the SAS Viya installation files.

Note: User ID `sas` is recommended.

- Use the `dagentsrv` CLI to obtain an authorization token using a SAS Data Agent (3.5) administrator account:

IMPORTANT The most recent version of the `dagentsrv` CLI must be used.

```
./sas-viya prof set-endpoint https://da35.endpoint.host.company.com:443
./sas-viya auth login -u da35-administrator-username -p da35-administrator-password
./sas-viya dagentsrv server set-default rda35-server-name
```

- Migrate all the credentials from the version 3.5 Vault to the version 4 Vault:

```
./sas-viya dagentsrv security credential migrate \
--destination-endpoint https://sas-viya-4-name-of-ingress-host:port \
--destination-username sas-viya-4-data-agent-administrator-username \
--destination-password sas-viya-4-data-agent-administrator-password \
--destination-server sas-viya-4-remote-data-agent-instance-name
```

The `security credentials migrate` option uses the version 3.5 credentials and server that were set when the authorization token was obtained and uses the version 4 credentials that are specified with the following options: `--destination-server`, `--destination-endpoint`, `--destination-username`, and `--destination-password`.

Here are the options for `security credential migrate`:

Option	Description
<code>data-agent</code>	The name of the remote SAS Data Agent server (3.5) instance from which to migrate data.

Option	Description
	Not required if <code>set-default rda35-server-name</code> was specified when the authorization token was obtained.
<code>destination-endpoint</code>	Required. The complete URL of the SAS Viya 4 endpoint for the destination remote SAS Data Agent (4) environment.
<code>destination-username</code>	Required. The administrator user name for the destination remote SAS Data Agent (4) environment.
<code>destination-password</code>	Required. The administrator password for the destination remote SAS Data Agent (4) environment.
<code>destination-server</code>	Required. The server instance name of the destination remote SAS Data Agent (4) environment.
<code>domain</code>	Optional. The default is all domains. If used, only the specified domain is migrated.
<code>identity</code>	Optional. The default is all identities. If used, only the specified identity is migrated.
<code>tenant</code>	Required for multi-tenant deployments. The source tenant name in a multi-tenant deployment.

Reference Topics

dagentsrv Plug-in to the SAS Viya CLI

The dagentsrv plug-in to the SAS Viya (sas-viya) CLI is the user interface to SAS Data Agent. For additional information, see the following documentation:

- [SAS Viya Platform: Using the Command-Line Interface](#)
- [“Command Line Interface” in *Cloud Data Exchange for the SAS Viya Platform: Administrator’s Guide*](#)

SAS Container Manager Commands and Options

Commands

Here are some SAS Container Manager commands to consider when managing the container.

Command	Description	Notes
<code>./container-manager list started</code>	Lists information about the container.	The information listed includes the container name, the cadence name, the cadence version, and the container ID.
<code>./container-manager stop sas-data-agent-server-remote</code>	Stops the container.	
<code>./container-manager uninstall sas-data-agent-server-remote</code>	Uninstalls the container.	You must stop the container before you uninstall it.
<code>./container-manager help</code>	Displays help for SAS Container Manager.	

Options for container-manager sas-data-agent-server-remote

The following options are for the `container-manager sas-data-agent-server-remote` command.

General Options

Options	Description	Notes
<code>--hostname</code>	The internal host name that the container uses.	Required The default is <code>sas-data-agent-server-remote</code> . Generally, the value should not change from the default.
<code>--license-path</code>	The directory path and the name of the license file.	Required
<code>--vars</code>	The directory path and the name of the Docker configuration file to be used by remote SAS Data Agent.	Required In this guide, the file name is <code>da-vars.env</code> file.
<code>--access-vars</code>	The directory path and the name of the Docker configuration file that contains driver configuration for data sources.	Required In this guide, the file name is <code>sas-access.properties</code> .
<code>--sasdata</code>	The directory mounted in the container that contains the backup, pgdata, and Vault directories and the self-signed certificate files, if they are created.	Required This directory will be mounted in the container as <code>/sasdata</code> .
<code>--data</code>	The directory mounted in the container as <code>/data</code> .	Optional
<code>--mount</code>	The one or more external directories that are mounted in the container.	Optional The format is <i>local-directory:container-directory</i> . Multiple values can be specified by separating them with commas, such as <code>--mount /u/my/data:/mydata, /tmp:/vmtmp</code> .
<code>--port</code>	The port that the container listens to for incoming HTTPS requests.	Optional The default is 25141.

Options	Description	Notes
<code>--memory</code>	The maximum amount of memory to be used by the container.	Optional The value is a positive integer followed by one of the following: b, k, m, g to indicate bytes, kilobytes, megabytes, or gigabytes. For more information, see the memory runtime options in the Docker documentation. If this option is not used, all memory is available for use by the container.
<code>--user</code>	The user and optional group that runs the container.	Optional
<code>--userns</code>	The user namespace for the container.	Optional
<code>--oauth-secret</code>	The OAuth secret that is used to authenticate remote SAS Data Agent with the SAS Viya platform.	Optional When used with the <code>container-manager sas-data-agent-server-remote</code> command, the value overrides the SAS_DA_OAUTH_SECRET value from the <code>da-vars.env</code> file. For more information, see “What is required for authentication?” on page 11
<code>--vault-storage</code>	A value that indicates how Vault stores encrypted data on disk.	Optional When used with the <code>container-manager sas-data-agent-server-remote</code> command, the value overrides the SAS_DA_VAULT_STORAGE value from the <code>da-vars.env</code> file. For more information, see the SAS_DA_VAULT_STORAGE value in the Reference: da-vars.env table on page 24.
<code>--vault-key1</code> <code>--vault-key2</code> <code>--vault-key3</code> <code>--vault-key4</code>	The Vault keys that unlock Vault when the container is started.	Optional At least three of the Vault keys are required to start the container. For more information, see “How should Vault keys be

Options	Description	Notes
<code>--vault-key5</code>		read to start the container?" on page 12

Customer-provided Vault Server: Options

By default, the container uses the Vault server in the container to store encrypted data. To use a customer-provided Vault server instead, the following options can be used.

Options	Description	Notes
<code>--vault-address</code>	The fully qualified address and port of the Vault server provided by a customer.	Optional When used with the <code>container-manager sas-data-agent-server-remote</code> command, the value overrides the SAS_DA_VAULT_ADDRESS environment variable in the <code>da-vars.env</code> file.
<code>--vault-secret-path</code>	The path where secrets in the customer-provided Vault server are written.	Optional When used with the <code>container-manager sas-data-agent-server-remote</code> command, the value overrides the SAS_DA_VAULT_SECRET_PATH environment variable in the <code>da-vars.env</code> file.
<code>--vault-token</code>	The authentication token for the Vault server provided by a customer.	Optional When used with the <code>container-manager sas-data-agent-server-remote</code> command, the value overrides the SAS_DA_VAULT_TOKEN environment variable in the <code>da-vars.env</code> file.

Note: For information about the environment variables, see [“Customer-provided Vault Server: Environment Variables” on page 29](#).

Customer-provided PostgreSQL Server: Options

By default, the container uses the PostgreSQL server in the container to store SYSCAT information. To use a customer-provided PostgreSQL server instead, the following options can be used.

Options	Description	Notes
<code>--pg-host</code>	The IP address or hostname of the external PostgreSQL server that is used to store SYSCAT data. For more information, contact the PostgreSQL server administrator.	Optional When used with the <code>container-manager sas-data-agent-server-remote</code> command, the value overrides the SAS_DA_PG_HOST value from the <code>da-vars.env</code> file.
<code>--pg-port</code>	The port of the external PostgreSQL server that is used to store SYSCAT data. For more information, contact the PostgreSQL server administrator.	Optional When used with the <code>container-manager sas-data-agent-server-remote</code> command, the value overrides the SAS_DA_PG_PORT value from the <code>da-vars.env</code> file.
<code>--pg-uid</code>	The user ID that is used to connect to a customer-provided PostgreSQL server. For more information, contact the PostgreSQL server administrator.	Optional When used with the <code>container-manager sas-data-agent-server-remote</code> command, the value overrides the SAS_DA_PG_UID value from the <code>da-vars.env</code> file.
<code>--pg-pwd</code>	The password that is used to connect to a customer-provided PostgreSQL server. For more information, contact the PostgreSQL server administrator.	Optional When used with the <code>container-manager sas-data-agent-server-remote</code> command, the value overrides the SAS_DA_PG_PWD value from the <code>da-vars.env</code> file.
<code>--pg-db</code>	The database that is used to store SYSCAT data in a customer-provided PostgreSQL server. For more information, contact the PostgreSQL server administrator.	Optional When used with the <code>container-manager sas-data-agent-server-remote</code> command, the value overrides the SAS_DA_PG_SYSCAT_DB value from the <code>da-vars.env</code> file.
<code>--pg-schema</code>	The schema that is used to store SYSCAT data in a customer-provided PostgreSQL server. For more information, contact the PostgreSQL server administrator.	Optional When used with the <code>container-manager sas-data-agent-server-remote</code> command, the value overrides the

Options	Description	Notes
		SAS_DA_PG_SYSCAT_SCHEMA value from the da-vars.env file.

Note: For information about the environment variables, see [“Customer-provided PostgreSQL Server: Environment Variables” on page 30.](#)

Customer-provided Oracle Server: Options

By default, the container uses the PostgreSQL server in the container to store SYSCAT information. To use a customer-provided Oracle server instead, the following options can be used.

Options	Description	Notes
--ora-path	The connection information that is used to connect to a customer-provided Oracle server. For more information, contact the Oracle server administrator.	Optional When used with the <code>container-manager sas-data-agent-server-remote</code> command, the value overrides the SAS_DA_ORA_PATH value from the da-vars.env file.
--ora-uid	The user ID that is used to connect to a customer-provided Oracle server. For more information, contact the Oracle server administrator.	Optional When used with the <code>container-manager sas-data-agent-server-remote</code> command, the value overrides the SAS_DA_ORA_UID value from the da-vars.env file.
--ora-pwd	The password that is used to connect to a customer-provided Oracle server. For more information, contact the Oracle server administrator.	Optional When used with the <code>container-manager sas-data-agent-server-remote</code> command, the value overrides the SAS_DA_ORA_PWD value from the da-vars.env file.

Note: For information about the environment variables, see [“Customer-provided Oracle Server: Environment Variables” on page 32.](#)

Examples: Output Docker Commands with the --dry-run Option

The SAS Container Manager `--dry-run` option is used to display the Docker commands that SAS Container Manager uses to download, install, start, and stop the container. These Docker commands can be modified to work with Docker Compose and Podman or to add options that are not supported by SAS Container Manager.

Note: SAS does not provide a Docker Compose file. To use Docker Compose with remote SAS Data Agent, you must create your own script.

Here is an example of the `container-manager install` command with the `--dry-run` option:

```
./container-manager --dry-run install --cadence-name "stable-or-lts" --cadence-version
"version-number" --deployment-data SAS-certificates-file-name.zip sas-data-agent-server-remote
```

Here is example output that displays the docker commands:

```
docker images --quiet --filter label=PRODUCT-ID=sas-data-agent-server-remote_stable-2026.01
Uninstalling product 'sas-data-agent-server-remote_stable-2026.01'
Installing product 'sas-data-agent-server-remote_stable-2026.01'

docker login --username XXXXXX --password "XXXXXXXX" cr.sas.com

docker image build -t cr.sas.com/viya-4-x64_oci_linux_2-docker/sas-data-agent-server-
remote:0.46.30-20260315.1773612356892_sas-data-agent-server-remote_stable-2026.01 --label managed-
by=sas-container-manager --label CADENCE-NAME=stable --label CADENCE-VERSION=2026.01 --label
CADENCE-RELEASE=20260317.1773765562856 --label PRODUCT-NAME=sas-data-agent-server-remote --label
PRODUCT-ID=sas-data-agent-server-remote_stable-2026.01 /tmp/sas-container-manager583764206

docker logout cr.sas.com
```

Here is an example of the `container-manager start` command with the `--dry-run` option:

```
./container-manager --dry-run start --cadence-name "stable-or-lts" --cadence-version "version-
number" sas-data-agent-server-remote --license-path SAS-license-file-name.jwt --sasdata
mysasdata --vars da-vars.env --data mysasdata/mydata
```

Here is example output that displays the docker commands:

```

docker images --quiet cr.sas.com/viya-4-x64_oci_linux_2-docker/sas-data-agent-server-remote:0.46.30-20260315.1773612356892_sas-data-agent-server-remote_stable-2026.01
Starting product instance 'sas-data-agent-server-remote'

docker compose ls

docker compose ls --filter name=^sas-data-agent-server-remote$

docker run --label managed-by=sas-container-manager --label CADENCE-NAME=stable --label CADENCE-VERSION=2026.01 --label CADENCE-RELEASE=20260317.1773765562856 --label PRODUCT-NAME=sas-data-agent-server-remote --label PRODUCT-ID=sas-data-agent-server-remote_stable-2026.01 -it -d --restart always --cap-add NET_ADMIN --cap-add SYS_PTRACE --env SASCNTANRLICENSEDIR=/sasinside --env SASLICENSEFILE=SASViyaV4_9D31R8_0_stable_2025.12_license_1770818495628.jwt --env-file /home/enterprise-github.com/viya-container-manager/da-vars.env --hostname sas-data-agent-server-remote --volume /home/enterprise-github.com/viya-container-manager:/sasinside -p 25141:25141 --name sas-data-agent-server-remote --volume /home/enterprise-github.com/viya-container-manager/mysasdata:/sasdata --env-file /home/.sas-container-manager/sas-data-agent-server-remote/stable/2026.01/files/sas-access.properties cr.sas.com/viya-4-x64_oci_linux_2-docker/sas-data-agent-server-remote:0.46.30-20260315.1773612356892_sas-data-agent-server-remote_stable-2026.01

```

Note: To modify the Docker commands that are used by SAS Container Manager, you must use the following configuration:

- The environment variable **SASCNTANRLICENSEDIR** must be set to **/sasinside**.
- The environment variable **SASLICENSEFILE** must be set to the name (without path) of the license file.
- The directory that contains the license file must be mounted in the container as **/sasinside**.
- The directory that is used for sasdata must be mounted in the container as **/sasdata**.
- The environment variables listed in the **da-vars.env** and **sas-access.properties** files must be set by using the **--env-file** option or its equivalent.