



SAS[®] Viya[®] Platform: Backup and Restore

2025.06 - 2026.07

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.

<i>Backup and Restore: Overview</i>	3
Introduction to Backup and Restore	3
Backup and Restore Utilities Description	3
How to Use This Document	4
What Is Backed Up?	4
What Is Not Backed Up?	6
Multi-Tenancy Support	6
<i>Backup and Restore: Concepts</i>	7
Backup and Restore Terms	7
<i>Backup and Recovery: Workflows</i>	8
Backup and Recovery Using Provider-Specific Processes for PostgreSQL and Using the SAS Viya Platform for Remaining Data	8
<i>Backup and Restore: Perform an Ad Hoc Backup</i>	11
What Backup Scenarios Are Not Supported?	12
Best Practices for Performing Backups	13
Perform an Ad Hoc Backup	16
Perform an Incremental Ad Hoc Backup	17
Perform an Ad Hoc Backup of All Content with the SAS Viya Platform (Including PostgreSQL)	18
List All Executed Backups and Their Status	18
List Backups Where PostgreSQL Is Included	18
List Backups Where PostgreSQL Is Not Included	18
Get Status of a Backup	18
Get Details of a Backup	19

View Backup Logs	20
Backup and Restore: Schedule a Backup	20
Overview	20
Enable Suspended CronJob to Back Up All Content with the SAS Viya Platform (Including PostgreSQL)	21
Change the Default Backup Schedule	21
Schedule a Backup	22
Schedule an Incremental Backup	23
Suspend an Incremental Backup CronJob	23
List the Schedules for Automated Backups	24
List the Schedules for Automated Incremental Backups	24
Unschedule a Backup Schedule	24
Change a Backup Schedule	25
Backup and Restore: Perform a Restore	25
What Restore Scenarios Are Not Supported?	26
Best Practices for Performing Restores	27
Perform a Restore Using Only the SAS Viya Platform Processes for All Content	29
Perform a Restore Using Provider-Specific Processes for PostgreSQL	35
Post-Restore Validation	45
List Restores	45
Show Details of a Restore	45
View Restore Logs	46
Backup and Restore: Purge Backups	46
Purge Backups	46
Backup and Restore: Recover from a Disaster	48
Overview	48
Disaster Recovery Prerequisites and Planning on the Source Environment	49
Perform a Disaster Recovery	51
Post-Restore Validation	58
Backup and Restore: Administrator Tasks	59
Backup and Restore: Reference	60
Backup Details	60
How to Find the Physical Location of the Persistent Volume (PV) in an Azure Environment	62
Examples of Transferring Backup Data between the SAS Viya Platform Environment and the Local File System	63
Copy Backup Data from Azure Environment to Azure Environment	70
Determine the Backup ID of the Last Successful Backup	72
Set the INCLUDE_POSTGRES configMap Parameter	74
Backup Progress	75
Adjust CPU and Memory Resources for Restore Job and PostgreSQL	77
Resource Requirements for Restore	78
Sizing Recommendations for Restore	79
Comparison of Backup and Restore Approaches	79
LZ4 Compression Details	81
Backup and Restore: Troubleshooting	83
General Troubleshooting Strategies	83
Logs	83
Backup Fails for fileSystem Source When Using Azure Premium Storage	84
Backup Fails When Services Are Not Completely Started	84
Backup Fails If Available Disk Space Cannot Be Calculated	85

Restore Fails If Resource Validation Fails	86
Purge Job Fails Requiring Manual Deletion of Backup	87
Purge Job Fails Due to sas-backup-agent Time-Out	88
Backup and Restore Processes are Slow	88
The sas-backup-pv-copy-cleanup Script Fails to Clean Up the CAS PVs during a Restore on OpenShift	89
CAS Server Data Fails to Backup	90
Restore Job or Migration Job Fails Due to Active Connection Termination in External PostgreSQL	91

Backup and Restore: Overview

Note: During deployment, you created a deployment directory that SAS recommended that you name `deploy` or an alternative name that is meaningful to you. The directory is referred to as `$deploy` in this guide. Replace `$deploy` with the directory name that you choose

Introduction to Backup and Restore

The SAS Viya platform can be deployed with other SAS software at your site to gain insight from analytics and business intelligence. The SAS Viya platform consists of many software components, servers, and services (including SAS Cloud Analytic Services (CAS) Server, SAS Compute Server, SAS Launcher Server, Identities service, and Audit service). You should have a strong backup and restore strategy and validate your backup and restore processes regularly. A backup strategy should include tasks for backing up all software components and data in addition to running the backup job.

Backup and Restore Utilities Description

You can manage the backup and restore of the SAS Viya platform as follows:

- Use only the SAS Viya platform processes to back up all data including all registered instances of PostgreSQL. This is the default method. This method uses the native utilities like `pg_dump` and `pg_restore` to perform the backup and restore. This method can be slower, but provides more flexibility during the migration.
- Use provider-specific PostgreSQL processes to back up PostgreSQL, and SAS Viya platform processes to back up the remaining data. There could be a variety of cloud providers for external PostgreSQL. For internal Crunchy Data PostgreSQL, use `pgBackRest`. To use this option, the provider must support Point-In-Time-Recovery (PITR) of the external PostgreSQL server to a custom restore point based on date and time. It might also create a new host for the external PostgreSQL instance. For improved performance, this is the recommended approach. See [“Backup and Recovery: Workflows” on page 8](#) for more information.

This document uses the following terms to refer to the different backup and restore utilities that are provided with the SAS Viya platform:

SAS Viya backup and restore utility

Uses source-specific methods for backing up and restoring multiple SAS Viya content sources.

When this utility is used for the PostgreSQL content, it is managed with `pg_dump` and `pg_restore`.

SAS internal PostgreSQL pgBackRest utility

Uses Crunchy Data's `pgBackRest` utility for backing up and restoring a binary backup of the internal Crunchy data PostgreSQL.

See [“Comparison of Backup and Restore Approaches” on page 79](#) for a summary of the different approaches to backup and restore in SAS Viya 4.

How to Use This Document

Here is a strategy for using this document:

See [“Best Practices for Performing Backups” on page 13](#) for information about when to perform backups, backup configuration properties, and other backup settings and properties.

See [“Best Practices for Performing Restores” on page 27](#) for information about when to perform restores, restore configuration properties, and other restore settings.

See [“Incremental Backups” on page 60](#) to perform faster and more frequent backups.

See [“Backup and Restore Utilities Description” on page 3](#) for information about different methods to manage backup and restore.

See [“Perform a Restore Using Provider-Specific Processes for PostgreSQL” on page 35](#) for information about how to back up and restore PostgreSQL using provider-specific PostgreSQL processes.

See [“What Is Not Backed Up?” on page 6](#) for your awareness about content that is not backed up.

See [“Disaster Recovery Prerequisites and Planning on the Source Environment” on page 49](#) for recommended ongoing tasks to prepare for a disaster recovery.

See [“Backup and Restore: Administrator Tasks” on page 59](#) for additional administrative tasks.

What Is Backed Up?

- Content in the SAS Infrastructure Data Server (platform PostgreSQL) is backed up when either internal or external PostgreSQL is configured.

- Several offerings on the SAS Viya platform include a second instance of PostgreSQL referred to as CDS PostgreSQL. Content in the SAS Common Data Server (CDS PostgreSQL) is backed up when either internal or external PostgreSQL is configured.

Note:

- Multiple methods are available to back up and restore content for the SAS Infrastructure Data Server and the SAS Common Data Server. The same backup and restore method is used for both whether internal PostgreSQL or external PostgreSQL is configured. It is not possible to use a different method for each database server.
 - The backup and restore in this document uses the SAS Viya backup and restore utility . This is different from the SAS internal PostgreSQL pgBackRest utility used by internal Crunchy Data PostgreSQL. Though the SAS Viya backup and restore utility can back up internal Crunchy Data PostgreSQL, it has its own format and location that are completely different from those of the SAS internal PostgreSQL pgBackRest utility. They are not related and are independently operated. For the pgBackRest backup and restore operations, see [“Take an Ad Hoc pgBackRest Backup” in SAS Viya Platform: Infrastructure Servers](#) and [“Restore PostgreSQL Cluster from pgBackRest Backup” in SAS Viya Platform: Infrastructure Servers](#).
 - When the INCLUDE_POSTGRES parameter is set to **true**, the SAS Viya backup and restore utility is used to back up content whether using internal PostgreSQL or external PostgreSQL.
 - When the INCLUDE_POSTGRES parameter is set to **false**, the method varies. SAS internal PostgreSQL pgBackRest utility must be used to back up and restore internal PostgreSQL. For external PostgreSQL, tools from the cloud-provider must be used.
-
- Content such as reports (including SAS Visual Analytics reports), comments, authorization rules, attachments, audit records, user preferences, and data source definitions is backed up.
 - Configuration settings and definitions for infrastructure and applications that you defined based on your business requirements are backed up. These settings are saved in SAS Configuration Server (Consul). Application configuration settings include user interface (UI) themes (for example, the SAS Visual Analytics UI theme) and batch run date (for example, batch date of running SAS Visual Analytics reports).
 - If CAS is configured for data management and analytics, CAS access controls and global caslibs are backed up. Caslibs help with accessing data from a data source. A caslib consists of a data source definition and can include access controls. Access controls help with controlling user access to the data.

In a clustered environment, caslibs and access controls of the CAS primary node are backed up.

Note: CAS Servers must be running to backup CAS server data.

- From the CAS data persistent volume (PV), the subdirectory `/cas/data/caslibs` is backed up. In a non-multi-tenant environment, there is one CAS data PV named cas-default-data.
-

Note:

Multiple persistent volumes (PVs) are used for backup.

Run this command to list the PVCs where backup is stored:

```
kubectl -n namespace get pvc -l "sas.com/backup-role=storage"
```

What Is Not Backed Up?

Backup does not support the backup of several items. You must take alternative steps to back up these items.

- Audit archive files. If you store audit files in an audit archive file system, back up that file system separately, as part of your site security policy. To learn more about audit archive backup, restore, and migration, see [“Backup, Restore, and Migrate the Audit Archive” in SAS Viya: Auditing](#).
- Kubernetes manifest and related items.
- User home directories.
- Other data sources, such as:
 - Third-party databases.
 - Data that is stored outside of the SAS Infrastructure Data Server or PostgreSQL server.
 - SAS Infrastructure Data Server or PostgreSQL server metadata such as database user information, roles, and permissions that are stored in the server.
 - Data that is loaded to CAS. This is in-memory data that is not saved to a persistent volume (PV).
 - Data that is stored on remote file systems (such as path-based data sources including PATH and DNFS).
- Any PVs mounted that are not already explicitly included in the list of backed up items [here on page 4](#).
- If the `INCLUDE_POSTGRES` parameter is set to `false` in the `sas-backup-job-parameters` configMap, external PostgreSQL is not backed up.

Multi-Tenancy Support

Customers who support use cases that require separate tenants are encouraged to deploy the SAS Viya platform into more than one namespace on the same Kubernetes cluster.

Backup and Restore: Concepts

Backup and Restore Terms

Common Terms

Here is a list of common backup and restore terms:

ad hoc backup

the backup that can be initiated by a user with batch or jobs permissions on an as-needed basis.

backup

a copy of several SAS Viya platform content and configuration files in a time-stamped directory that are distributed among multiple persistent volumes (PV). A backup can be initiated by a user with batch or jobs permissions. For more information, see [“Backup Details” on page 60](#).

Consul

also known as SAS Configuration Server. An open-source server by Hashicorp. It is used for service discovery and configuration, and is characterized by its high availability and scalability.

incremental backup

a backup that contains only the portion of data that has changed since the preceding backup copy was made. For more information, see [“Incremental Backups” on page 60](#).

job time-out

the number of minutes before a job times out.

persistent volume (PV)

a cluster-wide resource that you can use to store data in a way that it persists beyond the lifetime of a pod.

persistent volume claim (PVC)

a request for storage, which is met by binding the PVC to a persistent volume (PV).

pod

represents a unit of deployment—a single instance of an application in Kubernetes—that can consist of either a single container or a few containers that are tightly coupled and share resources.

PostgreSQL server

a relational database management system (RDBMS) that stores user content such as reports, comments, authorization rules, attachments, audit records, user preferences, and some data source definitions.

pre-restore validation

validation done by the restore job before performing a restore. A pre-restore validation checks the following:

- Does the provided backup exist?

- Is the backup completed?
- Is the backup purged?

recovery point objective (RPO)

the interval of time that might pass during a disruption before the quantity of data lost during that period exceeds the organization's maximum allowable threshold or "tolerance."

restore

the process of restoring a backup.

retention period

the number of days that backups are stored before they are eligible for purge. Default is 30 days.

scan

the process of performing an inventory of a SAS Viya platform deployment. For more information, see [SAS Viya Platform: Inventory](#).

SAS Infrastructure Data Server

an instance of *PostgreSQL* server that is provided with the SAS Viya platform.

scheduled backup

the backup that can be periodically scheduled by a user with batch or jobs permissions based on business requirements.

status

status of a backup or restore. The possible values are:

- *Completed* indicates that the backup or restore operation has completed successfully.
- *CompletedWithWarning* indicates that the backup or restore operation has completed with warnings.
- *Failed* indicates that a backup or restore for one or more data sources failed.

Backup and Recovery: Workflows

Backup and Recovery Using Provider-Specific Processes for PostgreSQL and Using the SAS Viya Platform for Remaining Data

The default method of performing backup and recovery uses the SAS Viya backup and restore utility to back up and recover all data. An additional method of performing backup and recovery uses cloud-provider processes to back up and recover external PostgreSQL instances or [the SAS internal PostgreSQL pgBackRest utility](#) for internal Crunchy Data PostgreSQL. For better performance during backup and recovery, this is the recommended approach.

Note: You can back up SAS Viya platform content using an incremental backup. Incremental backups capture only changed or differential data between successive backups after the last full backup. If a full backup does not exist, executing an incremental backup will create a full backup.

Incremental backups are supported only when performing backup and recovery of PostgreSQL using the SAS internal PostgreSQL pgBackRest utility for internal Crunchy Data PostgreSQL or cloud-provider processes for external PostgreSQL.

For more information, see [“Incremental Backups” on page 60](#).

Step 1: Exclude PostgreSQL from the SAS Viya Platform Backup

To use your PostgreSQL provider tools for backup and restore of PostgreSQL, you must configure the SAS Viya backup and restore utility to exclude PostgreSQL by setting the `INCLUDE_POSTGRES` configMap parameter to `false`. The restore of PostgreSQL is also skipped for these backups. Then use the tools included with your PostgreSQL provider to manage the backup and restore of PostgreSQL instances. See [“Set the INCLUDE_POSTGRES configMap Parameter” on page 74](#) for more information about the `INCLUDE_POSTGRES` configMap parameter.

IMPORTANT If you are using the SAS internal PostgreSQL pgBackRest utility for the backup and recovery of internal Crunchy Data PostgreSQL, SAS recommends that you continue to schedule periodic backups that include PostgreSQL. Backups that use the SAS Viya backup and restore utility for all data are required for disaster recovery and SAS Viya platform to SAS Viya platform migrations. Create these backups by setting the `INCLUDE_POSTGRES` configMap parameter to `true`. Alternatively, you can use the new CronJob designed for this feature. See [“Enable Suspended CronJob to Back Up All Content with the SAS Viya Platform \(Including PostgreSQL\)” on page 21](#).

Note:

- If you set `INCLUDE_POSTGRES` to `false`, then you must restore PostgreSQL using PostgreSQL-provider processes.
 - Incremental backups are supported only when `INCLUDE_POSTGRES` is set to `false`.
-

Step 2: Check the Retention Policy

- If you are using the SAS internal PostgreSQL pgBackRest utility to back up and restore internal Crunchy Data PostgreSQL, update the pgBackRest backup settings according to your backup or archive retention policy. See [“Change the Default pgBackRest Backup Settings” in SAS Viya Platform Operations: Servers and Services](#) and the related files for details about how to modify the settings.

- If you are using cloud-provider processes to back up and restore external PostgreSQL, follow any specific instructions from the cloud-provider to update the backup settings according to your backup or archive retention policy.

Step 3: Check for Incremental Backup

If you use the SAS internal PostgreSQL pgBackRest utility to back up and restore internal Crunchy Data PostgreSQL, incremental backups are enabled by default. To enable incremental backups for the rest of SAS Viya, see [“Schedule an Incremental Backup” on page 23](#).

Step 4: Perform Ad Hoc Backup

- 1 For external PostgreSQL, back up using any specific instructions from your PostgreSQL provider. For internal Crunchy Data PostgreSQL, see [“Take an Ad Hoc pgBackRest Backup” in SAS Viya Platform: Infrastructure Servers](#) for backup instructions.

Note: As a best practice, perform a backup of the PostgreSQL content before performing the backup of SAS Viya content. This ensures that the PostgreSQL content is available for a PITR at the time of the SAS Viya backup. Similarly, if regular backups are scheduled, ensure that the backup for PostgreSQL has time to complete before the backup of SAS Viya begins.

- 2 Perform an ad hoc backup of the rest of the SAS Viya platform. See the instructions [here on page 11](#) for a full backup. See the instructions [here on page 60](#) for an incremental backup.
- 3 Make sure to note and record the backup ID of the job.

```
kubectl get jobs -n name-of-namespace -l "sas.com/backup-job-type=scheduled-backup" -L
"sas.com/sas-backup-id"
```

- 4 Check the status of the job. See [“Get Status of a Backup” on page 18](#), and [“Get Details of a Backup” on page 19](#) for more information.

Step 5: Perform a Restore Using Provider-Specific Processes for PostgreSQL and Using the SAS Viya Platform for the Remaining Data

If your PostgreSQL provider supports Point-In-Time-Recovery (PITR) of the PostgreSQL instance to a custom restore point based on date and time, then you can use the following approach to restore your PostgreSQL instance and the rest of the data. If your PostgreSQL provider creates a new host for the PostgreSQL instance, additional options are available to specify the new host during recovery.

Note: When PITR is performed of external PostgreSQL, a new PostgreSQL server might be created. If so, the timeline of the new PostgreSQL server starts from the point the server is created. As a result, once the old server is deleted, it is not possible to restore to any point in time before the new

server was created. This implies that after the restore the older backups cannot be used if the old server is no longer available.

To complete the restore, you must do both of the following:

1 Restore PostgreSQL.

- For internal Crunchy Data PostgreSQL and external PostgreSQL, do the following:
 Use the same time stamp that you retrieved from the backup ID of the ad hoc backup when you specify a time stamp for restoring external PostgreSQL or internal Crunchy Data PostgreSQL. Both of the time stamps are in Universal Coordinated Time (UTC) format.
 For example, if the backup ID of your ad hoc backup is `20240104-102233F`, then the time stamp of the ad hoc backup is January 4, 2024 at 10:22:33am.
 If this is an incremental backup and your backup ID is `2024104-102233F_20240105-112059I`, then the time stamp considered is the value after the “_”. Accordingly, the time stamp of `20240105-112059I` is January 05, 2024 at 11:20:59 am).
- For internal Crunchy Data PostgreSQL, see [“Restore PostgreSQL Cluster from pgBackRest Backup” in SAS Viya Platform: Infrastructure Servers](#), and follow the instructions to perform a restore using the PITR method.
- For external PostgreSQL, use instructions from your PostgreSQL provider. Specify a host name for the new PostgreSQL cluster if the restore takes place on a new PostgreSQL cluster.

2 Restore the remaining data using the SAS Viya backup and restore utility .

If your provider-specific processes for PostgreSQL restore the PostgreSQL instance to a new host and support Point-In-Time-Recovery (PITR), then some additional parameters must be set during the restore. You must specify the `AUTO_SWITCH_POSTGRES_HOST` and `DATASERVER_HOST_MAP` parameters during the restore.

Note: The restore of internal Crunchy Data PostgreSQL is performed to the same PostgreSQL cluster with the same host name. Do not specify the `AUTO_SWITCH_POSTGRES_HOST`, `DATASERVER_HOST_MAP`, or `SQL_PROXY_POSTGRES_CONNECTION_MAP` parameters during the restore if using internal Crunchy Data PostgreSQL.

See [“Perform a Restore Using Provider-Specific Processes for PostgreSQL” on page 35](#) for instructions.

Backup and Restore: Perform an Ad Hoc Backup

Note:

- The backup and restore in this document uses the SAS Viya backup and restore utility . This is different from the SAS internal PostgreSQL pgBackRest utility used by internal Crunchy Data

PostgreSQL. Though the SAS Viya backup and restore utility can back up internal Crunchy Data PostgreSQL, it has its own format and location that are completely different from those of the SAS internal PostgreSQL pgBackRest utility. They are not related and are independently operated. For the pgBackRest backup and restore operations, see [“Take an Ad Hoc pgBackRest Backup” in SAS Viya Platform: Infrastructure Servers](#) and [“Restore PostgreSQL Cluster from pgBackRest Backup” in SAS Viya Platform: Infrastructure Servers](#).

- During deployment, you created a deployment directory that SAS recommended that you name deploy or an alternative name that is meaningful to you. The directory is referred to as \$deploy in this guide. Replace \$deploy with the directory name that you choose

What Backup Scenarios Are Not Supported?

- Restoring a backup from a different cadence is not supported. For example, you can restore to a SAS Viya platform Stable environment from a backup of a SAS Viya platform Stable environment. You can restore to a SAS Viya platform Long-Term Support environment from a backup of a SAS Viya platform Long-Term Support environment. But you cannot restore to a SAS Viya platform Long-Term Support environment from a backup of a SAS Viya platform Stable environment. See [“Software Releases and Versions” in Getting Started with SAS Viya Platform Operations](#) for more information.
- Restoring a backup from a different cadence version is not supported. For example, you can restore to a SAS Viya platform 2026.02 environment from a backup of a 2026.02 environment. But you cannot restore to a 2026.02 environment from a backup of a 2026.01 or earlier environment. Starting in 2026.02, when you restore from a backup of a different cadence, the restore job is marked as failed. See [“Software Releases and Versions” in Getting Started with SAS Viya Platform Operations](#) for more information.

Note:

The cadence and cadence version of the deployment are reported in the status.json file for each backup. The path location of the status.json file is `backupID/status.json` in the sas-common-backup-data PVC. Here is an example of the cadence and cadence version that is included in the status.json file:

```
CADENCE_RELEASE: 20250123.1737607638742
```

```
CADENCE_DISPLAY_NAME: Long-Term Support 2026.05
```

Unless you have access to the sas-common-backup data PVC outside of the SAS Viya platform environment, you must use the sas-backup-copy-job to get a pod running that has the sas-common-backup data PVC mounted in it. To do so, make sure that the copy pod is running. If the copy pod is not running, do step 3a [here on page 64](#).

Once the copy pod is running, do the following steps:

- Get the name of the running copy pod.

```
kubectl -n name-of-namespace get pods -l sas.com/backup-job-type=sas-backup-pv-copy-cleanup
```

- Get the cadence information.

```
kubectl -n name-of-namespace exec copy-pod-name -- sh -c "jq '{CadenceRelease: .CadenceRelease,
```

```
CadenceDisplayName: .CadenceDisplayName}' /sasviyabackup/backup-id/status.json"
```

Best Practices for Performing Backups

- During deployment, you created a deployment directory that SAS recommended that you name deploy or an alternative name that is meaningful to you. The directory is referred to as \$deploy in this guide. Replace \$deploy with the directory name that you choose
- For deployments in OpenShift environments, a Kubernetes administrator must create and configure a Security Context Constraint (SCC). When CAS is deployed, administrators can choose and associate appropriate SCCs with the sas-cas-server service account, based on the CAS server configuration such as host launch, SSSD, or others. To perform the restore process, a Kubernetes administrator must create and associate a corresponding SCC with the sas-viya-backuprunner service account that aligns with the CAS server configuration. After the SCC is created and set up, namespace administrators can perform the restore operation.
- Restoring a backup from a different cadence is not supported. For example, you can restore to a SAS Viya platform Stable environment from a backup of a SAS Viya platform Stable environment. You can restore to a SAS Viya platform Long-Term Support environment from a backup of a SAS Viya platform Long-Term Support environment. But you cannot restore to a SAS Viya platform Long-Term Support environment from a backup of a SAS Viya platform Stable environment. See [“Software Releases and Versions” in Getting Started with SAS Viya Platform Operations](#) for more information.
- Restoring a backup from a different cadence version is not supported. For example, you can restore to a SAS Viya platform 2026.02 environment from a backup of a 2026.02 environment. But you cannot restore to a 2026.02 environment from a backup of a 2026.01 or earlier environment. Starting in 2026.02, when you restore from a backup of a different cadence, the restore job is marked as failed. See [“Software Releases and Versions” in Getting Started with SAS Viya Platform Operations](#) for more information.

Note:

The cadence and cadence version of the deployment are reported in the status.json file for each backup. The path location of the status.json file is **backupID/status.json** in the sas-common-backup-data PVC. Here is an example of the cadence and cadence version that is included in the status.json file:

```
CADENCE_RELEASE: 20250123.1737607638742
```

```
CADENCE_DISPLAY_NAME: Long-Term Support 2026.05
```

Unless you have access to the sas-common-backup data PVC outside of the SAS Viya platform environment, you must use the sas-backup-copy-job to get a pod running that has the sas-common-backup data PVC mounted in it. To do so, make sure that the copy pod is running. If the copy pod is not running, do step 3a [here on page 64](#).

Once the copy pod is running, do the following steps:

- Get the name of the running copy pod.

```
kubectl -n name-of-namespace get pods -l sas.com/backup-job-type=sas-backup-pv-copy-cleanup
```

- Get the cadence information.

```
kubectl -n name-of-namespace exec copy-pod-name -- sh -c "jq '{CadenceRelease: .CadenceRelease,
CadenceDisplayName: .CadenceDisplayName}' /sasviyabackup/backup-id/status.json"
```

- SAS recommends that you perform backups during off-peak hours. Generally, backups are not disruptive to the environment, but problems can still occur. Here are some examples:
 - It is possible but unlikely that content might not be from a single point in time since different content is backed up with different methodologies.
 - The content might not be as expected based on other ongoing activities.
- After your backup is complete, if you want to copy your backup package to a safe location until you can restore it, see [“Examples of Transferring Backup Data between the SAS Viya Platform Environment and the Local File System” on page 63.](#)
- For information about using PostgreSQL database options to improve the performance of the backup, see [“Tuning SAS Backup for PostgreSQL Database Backup Operations” in SAS Viya Platform Administration: Tuning.](#)
- Make sure that the source Kubernetes environment is functioning as expected, and that the frequency of performing [backups on page 7](#) is as specified in the [recovery point objective \(RPO\)](#) set by your organization.
- Perform an [inventory scan](#) of your source Kubernetes environment before each backup.
- Make sure that the scheduled backups are successful.
- After each successful backup, see [“Determine the Backup ID of the Last Successful Backup” on page 72](#) to obtain and save the backup ID.
- Use the backup job to perform a backup of SAS Viya platform content, configuration, and services. It backs up this data at the same point in time, which is required for a same point-in-time restore.
- If the `$deploy` directory is not stored within a version control system, be sure to back up the `$deploy` directory in addition to performing regular backups.
- For the following scenarios, always run a full backup with PostgreSQL. See [“Perform an Ad Hoc Backup of All Content with the SAS Viya Platform \(Including PostgreSQL\)” on page 18](#) for more information.
 - Before any infrastructure change, such as a Kubernetes upgrade, perform an ad hoc backup immediately. Do not wait for the next scheduled backup.
 - After a configuration change, perform an ad hoc backup immediately. Do not wait for the next scheduled backup.
 - After a migration, perform an ad hoc backup immediately. Do not wait for the next scheduled backup to run. For a future restore, use this backup or any backup performed after this backup. Do not use a backup performed before the migration.
 - If you are using incremental backups, always perform a full backup immediately after there is a change in the backup configuration, the deployment is modified (such as adding a CAS server), or the system is restored or migrated. Do not wait for the next scheduled backup.
- For information about how to change configuration settings for backups, see one of the following files:

File Format	Path
Markdown	<code>\$deploy/sas-bases/examples/ backup/configure/README.md</code>
HTML	<code>\$deploy/sas-bases/docs/ configuration_settings_for_backup _using_the_sas_viya_backup_and_re store_utility.htm</code>

- Storage resources such as the space available on the PVC are pre-validated by default to help ensure that there is adequate space on the PVC to store data for a backup job. If the pre-validation fails, the backup job fails. If you want to disable the resource validation, see one of the following files:

File Format	Path
Markdown	<code>\$deploy/sas-bases/examples/ backup/configure/README.md</code>
HTML	<code>\$deploy/sas-bases/docs/ configuration_settings_for_backup _using_the_sas_viya_backup_and_re store_utility.htm</code>

- Edit the backup configuration properties immediately after a new installation or any modifications to your SAS Viya platform deployment. Examples include, but are not limited to, the following:
 - deploying the SAS Viya platform
 - installing software updates
 - changing the topology
 - modifying the SAS Viya platform configuration
 - changing configuration properties
- Backups are purged after the retention period (default is 30 days). If you do not want any backups to be deleted after the retention period, you must manually archive the backups before they are purged. If you are using incremental backups, you must manually archive the related incremental backups as well.
- If a Kubernetes job entry for a purged backup needs to be retained, set the property `RETAIN_BACKUP_JOB_HISTORY_AFTER_PURGE` to `TRUE`. If this property is not set to `true`, all Kubernetes job entries are deleted by default.
- If the default storage class policy is not set to `retain`, the `sas-common-backup-data-storage-class-transformer.yaml` patch transformer must be used during deployment. As a result, backup PVs are created with a storage class that has the `retain` reclaim policy set. This ensures that the associated PVC is retained if the namespace is deleted.

For more information, see one of the following files:

File Format	Path
Markdown	<code>\$deploy/sas-bases/examples/ backup/configure/README.md</code>
HTML	<code>\$deploy/sas-bases/docs/ configuration_settings_for_backup _using_the_sas_viya_backup_and_re store_utility.htm</code>

- If the environment being backed up or restored was deployed by the SAS Viya Platform Deployment Operator, then note the following:
 - The SAS Viya Platform Deployment Operator does not participate in a backup and restore.
 - An ad hoc backup or restore should be performed manually without changing the custom resource (CR) file used by the SAS Viya Platform Deployment Operator.

See Also

For more information, see [“Backup Details” on page 60](#).

Perform an Ad Hoc Backup

See one of the following files for configuration options:

File Format	Path
Markdown	<code>\$deploy/sas-bases/examples/backup/configure/README.md</code>
HTML	<code>\$deploy/sas-bases/docs/ configuration_settings_for_backup_using_the_sas_viya_backup_and_restore_utility.htm</code>

If you are using cloud-provider processes to back up and restore external PostgreSQL or [the SAS internal PostgreSQL pgBackRest utility](#) for internal Crunchy Data PostgreSQL, then you must set the **INCLUDE_POSTGRES** configMap parameter to **false** to exclude PostgreSQL servers from being included in the SAS Viya platform default backup. See [“Set the INCLUDE_POSTGRES configMap Parameter” on page 74](#) for more information.

IMPORTANT If you are using the SAS internal PostgreSQL pgBackRest utility for the backup and recovery of internal Crunchy Data PostgreSQL, SAS recommends that you continue to schedule periodic backups that include PostgreSQL. Backups that use the SAS Viya backup and restore utility for all data are required for disaster recovery and SAS Viya platform to SAS

Viya platform migrations. Create these backups by setting the **INCLUDE_POSTGRES** configMap parameter to **true**. Alternatively, you can use the new CronJob designed for this feature. See [“Enable Suspended CronJob to Back Up All Content with the SAS Viya Platform \(Including PostgreSQL\)”](#) on page 21.

To trigger an ad hoc backup from the default backup schedule CronJob:

```
kubectl create job -n name-of-namespace --from=cronjob/sas-scheduled-backup-job backup-job-name
```

To trigger an ad hoc backup from the default backup, schedule a CronJob with DEBUG enabled:

```
kubectl create job -n name-of-namespace --from=cronjob/sas-scheduled-backup-job backup-job-name
--dry-run=client -o yaml | kubectl set env --dry-run=client --local=true -c "sas-backup-job"
SAS_LOG_LEVEL="DEBUG" -o=yaml -f - | kubectl set env --dry-run=client --local=true -c "sas-
backup-agent" SAS_LOG_LEVEL="DEBUG" -o=yaml -f - | kubectl apply -f -
```

If you are using cloud-provider processes to back up and restore external PostgreSQL or [the SAS internal PostgreSQL pgBackRest utility](#) for internal Crunchy Data PostgreSQL, then the **INCLUDE_POSTGRES** configMap parameter is set to **false**. If you want the backup to include PostgreSQL, then you can revert the **INCLUDE_POSTGRES** to **true** for this backup only. Here is the command:

```
kubectl create job -n name-of-namespace --from=cronjob/sas-scheduled-backup-job test-adhoc-
backup --dry-run=client -o yaml | kubectl set env --dry-run=client --local=true -c "sas-backup-
job" INCLUDE_POSTGRES="TRUE" -o=yaml -f - | kubectl apply -f -
```

Here is the above command with DEBUG enabled:

```
kubectl create job -n name-of-namespace --from=cronjob/sas-scheduled-backup-job test-adhoc-
backup --dry-run=client -o yaml | kubectl set env --dry-run=client --local=true -c "sas-backup-
job" INCLUDE_POSTGRES="TRUE" SAS_LOG_LEVEL="DEBUG" -o=yaml -f - | kubectl set env --dry-
run=client --local=true -c "sas-backup-agent" SAS_LOG_LEVEL="DEBUG" -o=yaml -f - | kubectl
apply -f -
```

Perform an Incremental Ad Hoc Backup

- 1 Make sure that a full backup has already been completed successfully with no warnings. If a full backup is not present, the incremental backup initiates a full backup with the current backup configurations.
- 2 Make sure that the **INCLUDE_POSTGRES** configMap parameter is set to *false*. See [“Set the INCLUDE_POSTGRES configMap Parameter”](#) on page 74.
- 3 Make sure to use the same backup configurations that were used when creating the last full backup.
- 4 To initiate an ad hoc incremental backup using the default incremental backup schedule CronJob, use the following command:

```
kubectl create job -n name-of-namespace --from=cronjob/sas-scheduled-backup-incr-job backup-
job-name
```

Note: Only one backup job can run at a time. The incremental backup job fails if any other backup job is running.

Perform an Ad Hoc Backup of All Content with the SAS Viya Platform (Including PostgreSQL)

To initiate a backup of all content including PostgreSQL even if the `INCLUDE_POSTGRES` configMap parameter is set to *false*, use the following command:

```
kubectl create job -n name-of-namespace --from=cronjob/sas-scheduled-backup-all-sources backup-job-name
```

List All Executed Backups and Their Status

To list all executed backups and their status:

```
kubectl get jobs -n name-of-namespace -l "sas.com/backup-job-type in (scheduled-backup, scheduled-backup-incremental)" -L "sas.com/sas-backup-id,sas.com/backup-job-type,sas.com/sas-backup-job-status,sas.com/sas-backup-persistence-status,sas.com/sas-backup-include-postgres" --sort-by=.status.startTime
```

List Backups Where PostgreSQL Is Included

To list all backups where PostgreSQL is included:

```
kubectl get jobs -n name-of-namespace -l "sas.com/backup-job-type in (scheduled-backup, scheduled-backup-incremental)","sas.com/sas-backup-include-postgres=true" -L "sas.com/sas-backup-id,sas.com/backup-job-type,sas.com/sas-backup-job-status,sas.com/sas-backup-persistence-status,sas.com/sas-backup-include-postgres" --sort-by=.status.startTime
```

List Backups Where PostgreSQL Is Not Included

To list all backups where PostgreSQL is not included:

```
kubectl get jobs -n name-of-namespace -l "sas.com/backup-job-type in (scheduled-backup, scheduled-backup-incremental)","sas.com/sas-backup-include-postgres=false" -L "sas.com/sas-backup-id,sas.com/backup-job-type,sas.com/sas-backup-job-status,sas.com/sas-backup-persistence-status,sas.com/sas-backup-include-postgres" --sort-by=.status.startTime
```

Get Status of a Backup

To obtain the status of a backup, get the ID of the applicable backup :

```
kubectl get jobs -n name-of-namespace -l "sas.com/backup-job-type in (jobtype)" -L "sas.com/sas-backup-id,sas.com/backup-job-type" --sort-by=.status.startTime
```

Note: In the above command, replace *(jobtype)* with one of the following values: (scheduled-backup), (scheduled-backup-incremental), (scheduled-backup,scheduled-backup-incremental). Be sure to include the parentheses around the value.

Then to get the status for the backup job, specify the backup ID from the previous command:

```
kubectl get jobs -n name-of-namespace -l "sas.com/sas-backup-id=backupID" -L "sas.com/sas-backup-id,sas.com/backup-job-type,sas.com/sas-backup-job-status,sas.com/sas-backup-persistence-status"
```

For example, suppose the namespace is *viyans* and the backup ID is 20200401-102233F:

```
kubectl get jobs -n viyans -l "sas.com/sas-backup-id=20200401-102233F" -L "sas.com/sas-backup-id,sas.com/backup-job-type,sas.com/sas-backup-job-status,sas.com/sas-backup-persistence-status"
```

Or suppose the incremental backup ID is 20200401-102233F_20200402-102233I:

```
kubectl get jobs -n viyans -l "sas.com/sas-backup-id=20200401-102233F_20200402-102233I" -L "sas.com/sas-backup-id,sas.com/backup-job-type,sas.com/sas-backup-job-status,sas.com/sas-backup-persistence-status"
```

Repeat these commands until you get a status of **Completed**, **Failed**, or **CompletedWithWarning**.

Pod status and job status are not synonymous. A backup job can fail while the pod status is complete.

Note: You can include the `sas.com/backup-remaining-time` label to get the estimated time for the backup to complete:

```
kubectl get jobs -n name-of-namespace -l "sas.com/sas-backup-id=backupID" -L "sas.com/sas-backup-id,sas.com/backup-job-type,sas.com/sas-backup-job-status,sas.com/sas-backup-persistence-status,sas.com/backup-remaining-time"
```

See “Backup Progress” on page 75 for more information.

IMPORTANT If a data source that contains a volume to be backed up is not running, the Failed status is quickly returned.

Get Details of a Backup

To obtain details for a backup, get the ID of the applicable backup:

```
kubectl get jobs -n name-of-namespace -l "sas.com/backup-job-type in (jobtype)" -L "sas.com/sas-backup-id,sas.com/backup-job-type" --sort-by=.status.startTime
```

Note: In the above command, replace *(jobtype)* with one of the following values: (scheduled-backup), (scheduled-backup-incremental), (scheduled-backup,scheduled-backup-incremental). Be sure to include the parentheses around the value.

Then to get the details for the backup job, specify the backup ID from the previous command:

```
kubectl describe jobs -n name-of-namespace -l "sas.com/sas-backup-id=backupID"
```

For example, suppose the backup ID is 20200401-102233F:

```
kubectl describe jobs -n viyans -l "sas.com/sas-backup-id=20200401-102233F"
```

Or suppose the backup ID is 20200401-102233F_20200402-102233I:

```
kubectl describe jobs -n viyans -l "sas.com/sas-backup-id=20200401-102233F_20200402-102233I"
```

To get the details for the backup data sources in a given job, specify the backup ID from the previous command:

```
kubectl get jobs -n name-of-namespace -l "sas.com/sas-backup-id=backupID", "sas.com/backup-job-type=scheduled-backup" -L "sas.com/sas-backup-datasource-types"
```

View Backup Logs

To view the log of a specific backup job, find its pod name:

```
kubectl get pods -n name-of-namespace | grep backup-job-name
```

To see the log:

```
kubectl logs pod-name -n name-of-namespace -c sas-backup-job
```

Backup and Restore: Schedule a Backup

Note: During deployment, you created a deployment directory that SAS recommended that you name deploy or an alternative name that is meaningful to you. The directory is referred to as \$deploy in this guide. Replace \$deploy with the directory name that you choose

Overview

In the SAS Viya platform, a default backup is scheduled on Sunday at 1:00 a.m. out of the box. You can [change the default scheduled backup](#), or you can [add additional scheduled backups](#) as needed.

Note: The time zone of scheduled jobs in the SAS Viya platform is determined by the time zone of the Kubernetes controller manager. See the [Kubernetes Cronjob documentation](#) for more information.

Here are the other backup tasks:

- [list scheduled backups](#)
- [unschedule a backup schedule](#)
- [change a backup schedule.](#)

Enable Suspended CronJob to Back Up All Content with the SAS Viya Platform (Including PostgreSQL)

The `sas-scheduled-backup-all-sources` CronJob triggers backup jobs that use the SAS Viya platform to back up all data (including PostgreSQL) even if the `INCLUDE_POSTGRES` configMap parameter is set to `false`. Backups that use the SAS Viya platform processes for all data are required for disaster recovery and SAS Viya platform to SAS Viya platform migrations.

See the "Enable a Suspended Job to Back Up All the Sources" section of the README file in one of the following locations for information about enabling the CronJob:

File Format	Path
Markdown	<code>\$deploy/sas-bases/overlays/backup/README.md</code>
HTML	<code>\$deploy/sas-bases/docs/optional_configurations_for_backup_jobs.htm</code>

The default scheduled time for the backup jobs scheduled by the CronJob is 1:00 am on Sunday. See the "Change the Default Schedule to Back Up All Sources to a Custom Schedule" section in the README file in one of the following locations for information about changing the scheduled time:

File Format	Path
Markdown	<code>\$deploy/sas-bases/examples/backup/configure/README.md</code>
HTML	<code>\$deploy/sas-bases/docs/configuration_settings_for_backup_using_the_sas_viya_backup_and_restore_utility.htm</code>

Change the Default Backup Schedule

See one of the following files for information about changing the default backup schedule:

File Format	Path
Markdown	<code>\$deploy/sas-bases/examples/backup/configure/README.md</code>
HTML	<code>\$deploy/sas-bases/docs/configuration_settings_for_backup_using_the_sas_viya_backup_and_restore_utility.htm</code>

Schedule a Backup

Note: For deployments in OpenShift environments, a Kubernetes administrator must create and configure a Security Context Constraint (SCC). When CAS is deployed, administrators can choose and associate appropriate SCCs with the sas-cas-server service account, based on the CAS server configuration such as host launch, SSSD, or others. To perform the restore process, a Kubernetes administrator must create and associate a corresponding SCC with the sas-viya-backuprunner service account that aligns with the CAS server configuration. After the SCC is created and set up, namespace administrators can perform the restore operation.

You can schedule a backup through a CronJob. By default, a backup is scheduled weekly on Sunday at 1:00 a.m. A maximum of 100 backup jobs per day can be scheduled. The CronJob has a backup manifest and creates an instance of a backup using the KUBECTL CREATE command.

To create a custom scheduled backup:

- 1 If it does not exist, create the directory `$deploy/site-config/backup`.¹
See [“Retrieve Required Files” in SAS Viya Platform: Deployment Guide](#) for more information about the `$deploy` directory.
- 2 Navigate to the `$deploy/sas-bases/examples/backup` folder.
- 3 Copy the `$deploy/sas-bases/examples/backup` directory to the `$deploy/site-config/backup` directory.
- 4 Navigate to the `$deploy/site-config/backup` directory.
- 5 Make sure that the Execute permission is set on the `generate-backup-job` script.

```
chmod +x generate-backup-job.sh
```

Make sure that Write permission is set on the `kustomization.yaml` file.

```
chmod +w kustomization.yaml
```

- 6 Execute the `generate-backup-job.sh` script to generate the manifest file for the scheduled backup job.

```
./generate-backup-job.sh "<cron expression>" name-of-namespace
```

For example:

```
./generate-backup-job.sh "0 1 * * 6" viya-ns
```

This example creates a schedule that runs the job weekly on Saturday at 1:00 a.m. This is a Kubernetes crontab expression that schedules the backup job at the time that the user specifies.

This script uses a `kubectrl` utility to generate in the same folder a Kubernetes manifest file with the prefix `sas-scheduled-backup-job-`, followed by a unique identifier. The `kubeconfig` environment variable must be set to allow `kubectrl` commands on this machine.

1. During deployment, you created a deployment directory that SAS recommended that you name `deploy` or an alternative name that is meaningful to you. The directory is referred to as `$deploy` in this guide. Replace `$deploy` with the directory name that you choose.

- 7 Edit the `kustomization.yaml` file. Add the name of the manifest file generated in step 6 to the **resources** section. For example:

```
resources: - sas-scheduled-backup-xxx.yaml
```

There should be only one scheduled backup manifest file in the **resources** section.

- 8 Navigate to the `$deploy` directory.
 - a Make a copy of the `kustomization.yaml` file to recover after temporary changes are made.

```
cp $deploy/kustomization.yaml $deploy/kustomization.yaml.save
```

- b Edit the `kustomization.yaml` and add the following to the **resources** section.

```
resources: - site-config/backup
```

- 9 As an administrator with cluster permissions, apply the edited files to your deployment by performing the steps described in [Modifying the Deployment Configuration](#).

To change the default backup schedule, see [“Change the Default Backup Schedule” on page 21](#).

Schedule an Incremental Backup

You can schedule an incremental backup using a CronJob. By default, this backup is set to run at 06:00 a.m. on every day of the week from Monday through Saturday but it is initially suspended. Ensure that no other backup job is scheduled to run at the same time.

To activate the suspended incremental backup CronJob, follow the steps below:

- 1 Make sure that the **INCLUDE_POSTGRES** configMap parameter is set to 'false'. See [“Set the INCLUDE_POSTGRES configMap Parameter” on page 74](#).
- 2 Navigate to the `$deploy` directory.
- 3 Create a backup of `$deploy/kustomization.yaml`.


```
cp $deploy/kustomization.yaml $deploy/kustomization.yaml.save
```
- 4 Edit the `$deploy/kustomization.yaml` and add the following to the transformers section:


```
transformers: - /sas-bases/overlays/backup/sas-scheduled-backup-incr-job-enable.yaml
```
- 5 As an administrator with cluster permissions, apply the edited files to your deployment by performing the steps described in [Modifying the Deployment Configuration](#).

Suspend an Incremental Backup CronJob

To suspend an incremental backup CronJob:

- 1 Navigate to the `$deploy` directory.
- 2 Create a backup of `$deploy/kustomization.yaml`.

```
cp $deploy/kustomization.yaml $deploy/kustomization.yaml.save
```

- 3 Edit the `$deploy/kustomization.yaml` and remove the following from the transformers section:

```
transformers: - /sas-bases/overlays/backup/sas-scheduled-backup-incr-job-enable.yaml
```

- 4 As an administrator with cluster permissions, apply the edited files to your deployment by performing the steps described in [Modifying the Deployment Configuration](#).

List the Schedules for Automated Backups

To list the schedules for automated backups:

```
kubectl get cronjobs -n name-of-namespace -l "sas.com/backup-job-type=scheduled-backup"
```

List the Schedules for Automated Incremental Backups

To list the schedules for automated incremental backups:

```
kubectl get cronjobs -n name-of-namespace -l "sas.com/backup-job-type=scheduled-backup-incremental"
```

Unschedule a Backup Schedule

To permanently unschedule a backup schedule, delete the CronJob.

```
kubectl delete cronjob -n name-of-namespace name-of-cron-job
```

To temporarily unschedule or suspend a backup schedule:

- 1 Navigate to the `$deploy/site-config/backup` directory.
- 2 Edit the YAML file for the scheduled job that you want to unschedule. Update the value of the `SUSPEND` field to `true`.
- 3 Save the file.
- 4 Edit the `kustomization.yaml` file, and add the name of the YAML file from step 2 to the resources section:

```
resources: - sas-scheduled-job-xxxx.yaml
```

- 5 Navigate to the `$deploy` directory. Edit the `kustomization.yaml` file and add the following line to the resources section:

```
resources: - site-config/backup
```

- 6 Build the manifest:

```
kustomize build -o site.yaml
```

7 Apply the manifest:

```
kubectl apply -f site.yaml --server-side --force-conflicts
```

Note: See “[Managing a SAS Viya Platform Deployment](#)” in *SAS Viya Platform: General Management of Servers and Services* for more information about suspending and resuming jobs.

Change a Backup Schedule

To change a backup schedule:

- 1 Navigate to the `$deploy/site-config/backup` directory.
- 2 Edit the YAML file for the scheduled job that you want to change. Update the value of the SCHEDULE field to the cron expression representing the new schedule.
- 3 Save the file.
- 4 Edit the `kustomization.yaml` file, and add the name of the YAML file from step 2 to the resources section:

```
resources: - sas-scheduled-job-xxxx.yaml
```

- 5 Navigate to the `$deploy` directory. Verify that `- site-config/backup` exists in the resources section of the `kustomization.yaml` file. If it does not, add the following line to the resources section:

```
resources: - site-config/backup
```

- 6 Build the manifest:

```
kustomize build -o site.yaml
```

- 7 Apply the manifest:

```
kubectl apply -f site.yaml --server-side --force-conflicts
```

Backup and Restore: Perform a Restore

Note:

- The backup and restore in this document uses the SAS Viya backup and restore utility. This is different from the SAS internal PostgreSQL `pgBackRest` utility used by internal Crunchy Data PostgreSQL. Though the SAS Viya backup and restore utility can back up internal Crunchy Data PostgreSQL, it has its own format and location that are completely different from those of the SAS internal PostgreSQL `pgBackRest` utility. They are not related and are independently operated. For the `pgBackRest` backup and restore operations, see “[Take an Ad Hoc pgBackRest](#)”

Backup ” in [SAS Viya Platform: Infrastructure Servers](#) and “Restore PostgreSQL Cluster from pgBackRest Backup ” in [SAS Viya Platform: Infrastructure Servers](#).

- During deployment, you created a deployment directory that SAS recommended that you name deploy or an alternative name that is meaningful to you. The directory is referred to as \$deploy in this guide. Replace \$deploy with the directory name that you choose

What Restore Scenarios Are Not Supported?

- The backup and the restore environments must reference the same Kubernetes namespace.
- The restore environment must include the same SAS products as the backup environment. When you add or remove SAS products it is important to take new backups.
- Restoring a backup from a different cadence is not supported. For example, you can restore to a SAS Viya platform Stable environment from a backup of a SAS Viya platform Stable environment. You can restore to a SAS Viya platform Long-Term Support environment from a backup of a SAS Viya platform Long-Term Support environment. But you cannot restore to a SAS Viya platform Long-Term Support environment from a backup of a SAS Viya platform Stable environment. See [“Software Releases and Versions” in Getting Started with SAS Viya Platform Operations](#) for more information.
- Restoring a backup from a different cadence version is not supported. For example, you can restore to a SAS Viya platform 2026.02 environment from a backup of a 2026.02 environment. But you cannot restore to a 2026.02 environment from a backup of a 2026.01 or earlier environment. Starting in 2026.02, when you restore from a backup of a different cadence, the restore job is marked as failed. See [“Software Releases and Versions” in Getting Started with SAS Viya Platform Operations](#) for more information.

Note:

The cadence and cadence version of the deployment are reported in the status.json file for each backup. The path location of the status.json file is `backupID/status.json` in the sas-common-backup-data PVC. Here is an example of the cadence and cadence version that is included in the status.json file:

```
CADENCE_RELEASE: 20250123.1737607638742
```

```
CADENCE_DISPLAY_NAME: Long-Term Support 2026.05
```

Unless you have access to the sas-common-backup data PVC outside of the SAS Viya platform environment, you must use the sas-backup-copy-job to get a pod running that has the sas-common-backup data PVC mounted in it. To do so, make sure that the copy pod is running. If the copy pod is not running, do step 3a [here on page 64](#).

Once the copy pod is running, do the following steps:

- Get the name of the running copy pod.

```
kubectl -n name-of-namespace get pods -l sas.com/backup-job-type=sas-backup-pv-copy-cleanup
```

- Get the cadence information.

```
kubectl -n name-of-namespace exec copy-pod-name -- sh -c "jq '{CadenceRelease: .CadenceRelease,
```

```
CadenceDisplayName: .CadenceDisplayName}' /sasviyabackup/backup-id/status.json"
```

Best Practices for Performing Restores

- For deployments in OpenShift environments, a Kubernetes administrator must create and configure a Security Context Constraint (SCC). When CAS is deployed, administrators can choose and associate appropriate SCCs with the sas-cas-server service account, based on the CAS server configuration such as host launch, SSSD, or others. To perform the restore process, a Kubernetes administrator must create and associate a corresponding SCC with the sas-viya-backuprunner service account that aligns with the CAS server configuration. After the SCC is created and set up, namespace administrators can perform the restore operation.
- Restoring a backup from a different cadence is not supported. For example, you can restore to a SAS Viya platform Stable environment from a backup of a SAS Viya platform Stable environment. You can restore to a SAS Viya platform Long-Term Support environment from a backup of a SAS Viya platform Long-Term Support environment. But you cannot restore to a SAS Viya platform Long-Term Support environment from a backup of a SAS Viya platform Stable environment. See [“Software Releases and Versions” in Getting Started with SAS Viya Platform Operations](#) for more information.
- Restoring a backup from a different cadence version is not supported. For example, you can restore to a SAS Viya platform 2026.02 environment from a backup of a 2026.02 environment. But you cannot restore to a 2026.02 environment from a backup of a 2026.01 or earlier environment. Starting in 2026.02, when you restore from a backup of a different cadence, the restore job is marked as failed. See [“Software Releases and Versions” in Getting Started with SAS Viya Platform Operations](#) for more information.

Note:

The cadence and cadence version of the deployment are reported in the status.json file for each backup. The path location of the status.json file is **backupID/status.json** in the sas-common-backup-data PVC. Here is an example of the cadence and cadence version that is included in the status.json file:

```
CADENCE_RELEASE: 20250123.1737607638742
```

```
CADENCE_DISPLAY_NAME: Long-Term Support 2026.05
```

Unless you have access to the sas-common-backup data PVC outside of the SAS Viya platform environment, you must use the sas-backup-copy-job to get a pod running that has the sas-common-backup data PVC mounted in it. To do so, make sure that the copy pod is running. If the copy pod is not running, do step 3a [here on page 64](#).

Once the copy pod is running, do the following steps:

- Get the name of the running copy pod.

```
kubectl -n name-of-namespace get pods -l sas.com/backup-job-type=sas-backup-pv-copy-cleanup
```

- Get the cadence information.

```
kubectl -n name-of-namespace exec copy-pod-name -- sh -c "jq '{CadenceRelease: .CadenceRelease,
```

```
CadenceDisplayName: .CadenceDisplayName}' /sasviyabackup/backup-id/status.json"
```

- While performing a restore, ensure that the system is not being actively used.
- Always choose the most recent and successful backup to perform a restore.
- Obtain the deployment resources for the target SAS Viya platform environment. SAS recommends that you use the same deployment assets as from the source SAS Viya platform environment. The version should be the same for the same cadence. To obtain the same deployment assets as you deployed on the source, you can download them again by selecting **Download History** from the order in the my.sas.com portal.
- You can use your PostgreSQL provider to restore PostgreSQL. The available methods of restoring the SAS Viya platform are as follows:
 - Use the SAS Viya backup and restore utility to restore all data including all registered instances of PostgreSQL. This is the default method. See [“Perform a Restore Using Only the SAS Viya Platform Processes for All Content” on page 29](#).
 - Use PostgreSQL provider processes to back up PostgreSQL, and the SAS Viya backup and restore utility to back up the remaining data. For improved performance, this is the recommended approach. See [“Perform a Restore Using Provider-Specific Processes for PostgreSQL” on page 35](#).
- If you are planning to restore to the same machine, you must perform an [inventory scan](#) of the source machine before you create the backup that you restore from. That way, after the restore, you can run another inventory scan and then compare the current environment to the restored content.
- If you are using incremental backups, perform a full backup immediately if the system has been restored or migrated.
- If you need to copy your backup package from another location so that it is accessible from the target environment, see [“Examples of Transferring Backup Data between the SAS Viya Platform Environment and the Local File System” on page 63](#).
- See [“Sizing Recommendations for Restore”](#) for information about PostgreSQL database size in the restore space.
- For information about using PostgreSQL database options to improve the performance of the restore, see [“Tuning SAS Backup for PostgreSQL Database Restore Operations” in SAS Viya Platform Administration: Tuning](#).
- Resources such as CPU and memory are pre-validated by default to help ensure that the restore job completes successfully. If the pre-validation fails, the restore job will fail. If you want to disable the resource validation, see one of the following files:

File Format	Path
Markdown	<code>\$deploy/sas-bases/examples/ restore/configure/README.md</code>
HTML	<code>\$deploy/sas-bases/docs/ configuration_settings_for_restor e_using_the_sas_viya_backup_and_r estore_utility.htm</code>

For more information about the resource requirements for restore, see [“Resource Requirements for Restore” on page 78](#).

- For less common configuration changes to restores, such as providing a custom database name or enabling parallel execution, see one of the following files:

File Format	Path
Markdown	<code>\$deploy/sas-bases/examples/ restore/postgresql/README.md</code>
HTML	<code>\$deploy/sas-bases/docs/ uncommon_restore_customizations.h tm</code>

- Restoring to a different cloud provider is supported as long as the backup SAS Viya platform environment and the target SAS Viya platform environment use an internal instance of PostgreSQL.
- See [“Support for Migration from SAS Viya 4 to SAS Viya 4 for Offerings” in SAS Viya Platform: System Migration](#) for links to additional steps that might be required for some offerings.
- If the environment being backed up or restored was deployed by the SAS Viya Platform Deployment Operator, then note the following:
 - The SAS Viya Platform Deployment Operator does not participate in a backup and restore.
 - An ad hoc backup or restore should be performed manually without changing the custom resource (CR) file used by the SAS Viya Platform Deployment Operator.

Perform a Restore Using Only the SAS Viya Platform Processes for All Content

If you use this method of restoring, make sure to choose a backup that does include PostgreSQL. See [“List Backups Where PostgreSQL Is Included” on page 18](#).

Note:

- Check for active or scheduled jobs in the environment before performing a restore.
 - Suspend any scheduled jobs in the environment to prevent them from automatically starting after the restore. To check for and disable scheduled jobs, you can do one of the following:

Use the Jobs page in SAS Environment Manager.	See “job flows unschedule” in SAS Viya Platform: Jobs and Flows
Use the job plug-in to the SAS Viya platform CLI.	See “job schedulers pause” in SAS Viya Platform: Jobs and Flows

- If your deployment contains SAS Workload Orchestrator, ensure that there are no running or pending SAS Workload Orchestrator jobs in the environment before performing a restore. To check the status of these jobs, you can do one of the following:

Use the SAS Workload Orchestrator graphical user interface (GUI) in SAS Environment Manager.

See [“SAS Workload Management Overview” in SAS Viya Platform: Workload Management](#).

Use the workload_orchestrator plug-in to the SAS Viya platform CLI.

Sign in to the sas-viya command-line interface (CLI) and download the workload-orchestrator plug-in. See [“Command-Line Interface: Instructions when Downloading the CLI” in SAS Viya Platform: Using the Command-Line Interface](#) for more information.

Use this command to identify active workload-orchestrator jobs: `./sas-viya workload-orchestrator jobs list`

Do not proceed with the restore if there are any running or pending SAS Workload Orchestrator jobs.

Here are the steps to perform a restore:

Note: Incremental backups are not supported when using the SAS Viya backup and restore utility to back up and recover all data. Incremental backups are supported only when the `INCLUDE_POSTGRES` configMap parameter is set to `false`, meaning that PostgreSQL is not included in SAS Viya platform backup.

- 1 If you want to use the new restore steps in your release of the SAS Viya platform, continue with the instructions in the next paragraph and each of the steps below:

Identify the `sas-restore-job-parameters` configMap that needs to be modified.

```
kubectl -n name-of-namespace describe cronjob sas-restore-job | grep -i sas-restore-job-parameters | awk '{print $1}' | head -n 1
```

- 2 Edit the configMap `sas-restore-job-parameters`. Include the parameters and values in a comma-separated list at the end of the command. For each parameter, specify an operation, path, and value. The `SAS_BACKUP_ID` and `SAS_DEPLOYMENT_START_MODE` parameters are always required.

```
kubectl patch cm sas-restore-job-parameters-name -n name-of-namespace --type json -p '[{"op": "operation", "path": "/data/parameter-name", "value": "value"}, {"op": "operation", "path": "/data/parameter-name", "value": "value"} ]'
```

Here is an example with sample values.

```
kubectl patch cm sas-restore-job-parameters-name -n name-of-namespace --type json -p '[{"op": "replace", "path": "/data/SAS_BACKUP_ID", "value": "2020-04-01T10_22_33_633_0700"}, {"op": "replace", "path": "/data/SAS_DEPLOYMENT_START_MODE", "value": "RESTORE"} ]'
```

Here is an example with sample values and DEBUG enabled.

```
kubectl patch cm sas-restore-job-parameters-name -n name-of-namespace --type json -p
' [ {"op": "replace", "path": "/data/SAS_BACKUP_ID", "value": "2020-04-01T10_22_33_633_0700"},
{"op": "replace", "path": "/data/SAS_DEPLOYMENT_START_MODE", "value": "RESTORE" }, {"op":
"replace", "path": "/data/SAS_LOG_LEVEL", "value": "DEBUG" } ] '
```

Here are more details about the previous code.

- Replace the value for SAS-BACKUP-ID with the ID of the backup that is selected for restore.
 - If you are performing a backup that uses the SAS Viya backup and restore utility to back up and recover all data, then select a backup for which the INCLUDE_POSTGRES configMap parameter is set to *true*. See [“List Backups Where PostgreSQL Is Included” on page 18](#).
 - If you are performing a backup that uses provider-specific processes for PostgreSQL, then select a backup for which the INCLUDE_POSTGRES configMap parameter is set to *false*. See [“List Backups Where PostgreSQL Is Not Included” on page 18](#).
- To increase the logging levels, add the SAS_LOG_LEVEL parameter to the list with the value DEBUG. SAS Technical Support recommends increasing log levels to DEBUG before performing a restore.

IMPORTANT Replace each variable with the value that is appropriate for your deployment. For example, *SAS-BACKUP-ID* should be replaced with a backup ID, such as 20200104-102233F.

If you want to exclude a deprecated configuration from being restored to the target SAS Viya platform system, see one of the following files:

File Format	Path
Markdown	<code>\$deploy/sas-bases/examples/ restore/configuration/README.md</code>
HTML	<code>\$deploy/sas-bases/docs/ configuration_settings_for_restore_using_the_sas_viya_backup_and_restore_utility.htm</code>

- 3 If you are migrating from a SAS Viya 3.5 system with the 2.87.1 version of the sas.files package and your target SAS Viya 4 platform is a 2026.01 or 2026.02 cadence, add the following parameters to your sas-restore-job-parameters:

Note: Each parameter must be on a single line.

```
-
RESTORE_DEFINITION_FILTER='{ "files.sas.files.6": "*", "files.sas.files.7": "*", "files.sas
.files.8": "*" }'
- RESTORE_CONFIGURATION_FILTER='{ "*.application/
vnd.sas.configuration.config.sas.files+json;version=6": "*", "*.application/
```

```
vnd.sas.configuration.config.sas.files+json;version=7": "*", "*.application/
vnd.sas.configuration.config.sas.files+json;version=8": "*"}
```

For more information about the 2.87.1 version of the sas.files package, see [“Configuration Values Are Lost during Migration with sas-files Version 2.87.1” in SAS Viya Platform: Full-System Migration from SAS Viya 3.5](#).

4 Check if the restore job already exists.

```
kubectl get job sas-restore-job -n name-of-namespace
```

If it does exist, delete the job.

```
kubectl delete job sas-restore-job -n name-of-namespace
```

5 Create job from the restore CronJob.

```
kubectl create job -n name-of-namespace --from=cronjob/sas-restore-job sas-restore-job
```

6 The restore job was started in the previous step. Inspect the status of the job:

```
kubectl get jobs -n name-of-namespace -l "sas.com/backup-job-type=restore" -L "sas.com/sas-backup-id,sas.com/backup-job-type,sas.com/sas-restore-status"
```

Submit the command until you get the Completed status. Possible statuses are Completed, CompletedWithWarnings, and Failed.

Note: A restore job might be marked as CompletedWithWarnings when there are configuration definition mismatches between SAS Viya 3.5 and SAS Viya 4 that prevent some configurations from being applied during the migration process. In these cases, RequestValidationFailed warnings are recorded in the restore logs. The restore process still completes successfully, and the target environment remains usable. However, some configuration definitions might be skipped due to version incompatibility.

- If you get an Error status or need to rerun the job for any reason, delete the job:

```
kubectl delete job sas-restore-job -n name-of-namespace
```

And then perform this previous step again:

- If a resource is not available to the pod, the restore job might appear to be running with "0/1" completions, whereas it actually is in a waiting state. You might want to describe the restore pod to confirm that the job has been able to start.

Get the name of the restore pod:

```
kubectl get pods -n name-of-namespace | grep sas-restore-job
```

Describe the pod, substituting the restore pod name that you identified in the last step:

```
kubectl -n name-of-namespace describe pod sas-restore-job-pod-name
```

7 Check the log of the restore job to confirm that the restore was successful.

To view the log of the restore job, find its pod name:

```
kubectl get pods -n name-of-namespace | grep sas-restore-job
```

To see the log:

```
kubectl -n name-of-namespace logs sas-restore-job-pod-name -c sas-restore-job
```

- 8 While the restore job is running, all pods get restarted with the exception of the CAS controller pods and CAS worker pods, which are scaled down. Before proceeding to the next step, do the following:

- Make sure that the `sas-logon-app` and `sas-configuration` are running.

```
kubectl get pods -n name-of-namespace | grep sas-logon
```

```
kubectl get pods -n name-of-namespace | grep sas-configuration
```

Note: At this point, while CAS-dependent pods might be in a **Running** state, requests to those pods might fail until CAS is successfully restarted in a later step.

- Make sure that the CAS controller pods and CAS worker pods have been scaled down.

```
kubectl get pods --selector="app.kubernetes.io/managed-by==sas-cas-operator" -n name-of-namespace
```

- 9 The file system restore requires a clean volume. Otherwise, the restore for the volume is skipped.

Note: If you are restoring to an OpenShift cluster, see the "Granting Security Context Constraints for Copy and Cleanup Job on an OpenShift Cluster" section in one of the following files for information about granting security context constraints before proceeding:

File Format	Path
Markdown	<code>\$deploy/sas-bases/examples/restore/scripts/README.md</code>
HTML	<code>\$deploy/sas-bases/docs/restore_scripts.htm</code>

- a Navigate to the `$deploy/sas-bases/examples/restore/scripts` folder.

- b Add Execute permission to the scripts used in the following steps.

```
chmod +x ./sas-backup-pv-copy-cleanup.sh ./scale-up-cas.sh
```

- c Run the `sas-backup-pv-copy-cleanup` script to clean up the CAS PVs. The script requires the following three arguments:

<i>namespace</i>	Defines the namespace.
<i>operation</i>	Defines the operation that the script performs. Valid values are <code>remove</code> and <code>copy</code> .
<i>tenant-list</i>	In a non-multi-tenancy environment, specify <code>default</code> .

Here is the command:

```
sas-backup-pv-copy-cleanup.sh namespace operation "tenant-list"
```

Here is an example of a command to remove the CAS PV in a non-multi-tenancy environment:

```
sas-backup-pv-copy-cleanup.sh viyans remove "default"
```

If you get the following error and this is the first attempt at a restore after a successful SAS Viya 3.x to SAS Viya platform migration, you might need to perform additional steps to clean the CAS PVs:

```
./sas-backup-pv-copy-cleanup.sh $NS remove shared
./sas-backup-pv-copy-cleanup.sh: line 100: [: : integer expression expected
```

To resolve, see one of the following files:

File Format	Path
Markdown	<code>\$deploy/sas-bases/examples/restore/scripts/README.md</code>
HTML	<code>\$deploy/sas-bases/docs/restore_scripts.htm</code>

10 Scale up the CAS deployment (restore the CAS data).

- a Navigate to the `$deploy/sas-bases/examples/restore/scripts` folder.
- b Run the scale-up-cas script to scale up the CAS deployment. This step restores the CAS data. The script requires the following two arguments:

namespace	Defines the namespace.
tenant-list	In a non-multi-tenancy environment, specify <i>default</i> .

Here is the command:

```
scale-up-cas.sh namespace "tenant-list"
```

Here is an example of a command to scale up the CAS deployment in a non-multi-tenancy environment:

```
scale-up-cas.sh viyans "default"
```

11 Confirm that the deployment is running. Verify that all pods are running successfully:

```
kubectl get pods -n name-of-namespace
```

Make sure that you are able to sign on to the system.

12 Check the log of the cas-controller pod for each participating CAS server for these messages to confirm that the restore was successful:

- "message": "Restore of CAS permstore volume content completed successfully."
- "message": "Restore of CAS data volume content completed successfully."
- "message": "Restore of CAS data and permstore volumes completed successfully."

If all of these messages are present in the log, then the CAS restore was successful. If neither or only one of these messages are present in the log along with an error message, then the CAS restore was not successful. Check the CAS pod logs for details.

Here is a sample command to check the CAS pod logs:

```
kubectl -n name-of-namespace logs sas-cas-server-default-controller -c sas-cas-server | grep "Restore of CAS"
```

13 Reset all configMap sas-restore-job-parameters.

Here is an example:

```
kubectl patch cm sas-restore-job-parameters-name -n name-of-namespace --type json -p ' [{ "op": "remove", "path": "/data/SAS_BACKUP_ID" }, {"op": "remove", "path": "/data/SAS_DEPLOYMENT_START_MODE"}, { "op": "remove", "path": "/data/SAS_LOG_LEVEL" } ] '
```

14 If you are using SCIM for identity management, you must re-create the access token. See [“How to Configure SCIM” in SAS Viya Platform: Identity Management](#).

15 If your software order includes offerings that include Process Orchestration, you must do the following to make sure that flows run after the restore:

- a Obtain a new access token and set the **ACCESS_TOKEN** environment variable. See [“Obtain an Access Token” in SAS Viya Platform: Authentication](#).
- b Run the following command:

```
curl -k -x POST "${INGRESS_URL}/processOrchestration/restore" \ -H "Authorization: Bearer $ACCESS_TOKEN"
```

Perform a Restore Using Provider-Specific Processes for PostgreSQL

To use this method of restoring, the PostgreSQL provider must support Point-In-Time-Recovery (PITR) of the PostgreSQL server to a custom restore point based on date and time. Also, it might create a new host for the external PostgreSQL instance. Follow the steps in [“Backup and Recovery Using Provider-Specific Processes for PostgreSQL and Using the SAS Viya Platform for Remaining Data” on page 8](#) to determine whether this method will work for you and then continue with the restore steps below.

If you use this method of restoring, make sure to choose a backup that does not include PostgreSQL. See [“List Backups Where PostgreSQL Is Not Included” on page 18](#).

Note:

- Check for active or scheduled jobs in the environment before performing a restore.
 - Suspend any scheduled jobs in the environment to prevent them from automatically starting after the restore. To check for and disable scheduled jobs, you can do one of the following:

Use the **Jobs** page in SAS Environment Manager.

See [“job flows unschedule” in SAS Viya Platform: Jobs and Flows](#)

Use the job plug-in to the SAS Viya platform CLI.

See [“job schedulers pause” in SAS Viya Platform: Jobs and Flows](#)

- If your deployment contains SAS Workload Orchestrator, ensure that there are no running or pending SAS Workload Orchestrator jobs in the environment before performing a restore. To check the status of these jobs, you can do one of the following:

Use the SAS Workload Orchestrator graphical user interface (GUI) in SAS Environment Manager.

See [“SAS Workload Management Overview” in SAS Viya Platform: Workload Management](#).

Use the workload_orchestrator plug-in to the SAS Viya platform CLI.

Sign in to the sas-viya command-line interface (CLI) and download the workload-orchestrator plug-in. See [“Command-Line Interface: Instructions when Downloading the CLI” in SAS Viya Platform: Using the Command-Line Interface](#) for more information.

Use this command to identify active workload-orchestrator jobs: `./sas-viya workload-orchestrator jobs list`

Do not proceed with the restore if there are any running or pending SAS Workload Orchestrator jobs.

Here are the steps to perform a restore:

1 Restore PostgreSQL as follows:

- For internal Crunchy Data PostgreSQL and external PostgreSQL, do the following:

Use the same time stamp that you retrieved from the backup ID of the ad hoc backup when you specify a time stamp for restoring external PostgreSQL or internal Crunchy Data PostgreSQL. Both of the time stamps are in Universal Coordinated Time (UTC) format.

For example, if the backup ID of your ad hoc backup is `20240104-102233F`, then the time stamp of the ad hoc backup is January 4, 2024 at 10:22:33am.

If this is an incremental backup and your backup ID is `20240104-102233F_20240105-112059I`, then the time stamp considered is the value after the “_”. Accordingly, the time stamp of `20240105-112059I` is January 05, 2024 at 11:20:59 am).

- For internal Crunchy Data PostgreSQL, see [“Restore PostgreSQL Cluster from pgBackRest Backup” in SAS Viya Platform: Infrastructure Servers](#), and follow the instructions to perform a restore using the PITR method.
- For external PostgreSQL, use instructions from your PostgreSQL provider. Specify a host name for the new PostgreSQL cluster if the restore takes place on a new PostgreSQL cluster.

2 Identify the sas-restore-job-parameters configMap that needs to be modified.

```
kubectl -n name-of-namespace describe cronjob sas-restore-job | grep -i sas-restore-job-parameters | awk '{print $1}' | head -n 1
```

- 3 Edit the configMap `sas-restore-job-parameters`. Include the parameters and values in a comma-separated list at the end of the command. For each parameter, specify an operation, path, and value. The `SAS_BACKUP_ID` and `SAS_DEPLOYMENT_START_MODE` parameters are always required.

```
kubectl patch cm sas-restore-job-parameters-name -n name-of-namespace --type json -p '[{"op": "operation", "path": "/data/parameter-name", "value": "value"}, {"op": "operation", "path": "/data/parameter-name", "value": "value"} ]'
```

Here is an example with sample values.

```
kubectl patch cm sas-restore-job-parameters-name -n name-of-namespace --type json -p '[{"op": "replace", "path": "/data/SAS_BACKUP_ID", "value": "2020-04-01T10_22_33_633_0700"}, {"op": "replace", "path": "/data/SAS_DEPLOYMENT_START_MODE", "value": "RESTORE"} ]'
```

Here is an example with sample values and DEBUG enabled.

```
kubectl patch cm sas-restore-job-parameters-name -n name-of-namespace --type json -p '[{"op": "replace", "path": "/data/SAS_BACKUP_ID", "value": "2020-04-01T10_22_33_633_0700"}, {"op": "replace", "path": "/data/SAS_DEPLOYMENT_START_MODE", "value": "RESTORE"}, {"op": "replace", "path": "/data/SAS_LOG_LEVEL", "value": "DEBUG"} ]'
```

Here are more details about the previous code.

- Replace the value for `SAS-BACKUP-ID` with the ID of the backup that is selected for restore.
 - If you are performing a backup that uses the SAS Viya backup and restore utility to back up and recover all data, then select a backup for which the `INCLUDE_POSTGRES` configMap parameter is set to `true`. See [“List Backups Where PostgreSQL Is Included” on page 18](#).
 - If you are performing a backup that uses provider-specific processes for PostgreSQL, then select a backup for which the `INCLUDE_POSTGRES` configMap parameter is set to `false`. See [“List Backups Where PostgreSQL Is Not Included” on page 18](#).
- To increase the logging levels, add the `SAS_LOG_LEVEL` parameter to the list with the value `DEBUG`. SAS Technical Support recommends increasing log levels to `DEBUG` before performing a restore.

IMPORTANT Replace each variable with the value that is appropriate for your deployment. For example, `SAS-BACKUP-ID` should be replaced with a backup ID, such as `20200104-102233F`.

When a provider that is specific for PostgreSQL manages PostgreSQL restore by using Point-In-Time-Recovery (PITR), the restore might be performed to a new PostgreSQL cluster with a different host name. In those cases, the new cluster needs to be registered with the SAS Viya deployments before it can be used. The parameters that you add to the list to register the new cluster differ if a SQL proxy is used to connect to the new PostgreSQL instance.

Note: The restore of internal Crunchy Data PostgreSQL is performed to the same PostgreSQL cluster with the same host name. Do not specify the `AUTO_SWITCH_POSTGRES_HOST`, `DATASERVER_HOST_MAP`, or `SQL_PROXY_POSTGRES_CONNECTION_MAP` parameters during the restore if using internal Crunchy Data PostgreSQL.

- To register and switch to the new PostgreSQL instance without a SQL proxy, add the `AUTO_SWITCH_POSTGRES_HOST` and `DATASERVER_HOST_MAP` parameters to the list.

Note: If the restore of external PostgreSQL is not to a new host, you do not need to include the `AUTO_SWITCH_POSTGRES_HOST` parameter in the commands. The examples below include the `AUTO_SWITCH_POSTGRES_HOST` parameter.

Here is an example:

```
{ "op": "replace", "path": "/data/AUTO_SWITCH_POSTGRES_HOST", "value": "TRUE" }, { "op": "replace", "path": "/data/DATASERVER_HOST_MAP", "value": "value" }
```

The value for the `DATASERVER_HOST_MAP` parameter is a comma-separated list of key value (KV) pairs indicating the mapping of data server custom resources (CR) to updated host names. The key and value within each KV pair are separated by a colon (:), where the key is the data server custom resource (CR) name and the value is the new host name for the recovered Postgres cluster.

There is one data server CR for each Postgres cluster that is configured with the SAS Viya platform environment. The names of the data servers are `sas-platform-postgres` for the default Postgres instances and the `sas-cds-postgres` for the CDS Postgres instances. To retrieve the list of all the data servers, use this command:

```
kubectl -n name-of-namespace get dataservers
```

Here is an example of the `DATASERVER_HOST_MAP` parameter with sample KV pairs:

```
{ "op": "replace", "path": "/data/DATASERVER_HOST_MAP", "value": "sas-platform-postgres:postgres-server-restored-name.postgres.azure.com,sas-cds-postgres:restored-cds-postgres.postgres.azure.com" }
```

Here is an example of the full command with `DEBUG` enabled and the `DATASERVER_HOST_MAP` parameter with sample KV pairs :

```
kubectl patch cm sas-restore-job-parameters-name -n name-of-namespace --type json -p '[ { "op": "replace", "path": "/data/SAS_BACKUP_ID", "value": "2020-04-01T10_22_33_633_0700" }, { "op": "replace", "path": "/data/SAS_DEPLOYMENT_START_MODE", "value": "RESTORE" }, { "op": "replace", "path": "/data/SAS_LOG_LEVEL", "value": "DEBUG" }, { "op": "replace", "path": "/data/DATASERVER_HOST_MAP", "value": "sas-platform-postgres:postgres-server-restored-name.postgres.azure.com,sas-cds-postgres:restored-cds-postgres.postgres.azure.com" } ]'
```

- To register and switch to the new PostgreSQL instance with a SQL proxy, add the `AUTO_SWITCH_POSTGRES_HOST` and `SQL_PROXY_POSTGRES_CONNECTION_MAP` parameters to the list.

Note: This is used If you are using Google Cloud SQL for PostgreSQL

Note: If the restore of external PostgreSQL is not to a new host, you do not need to include the `AUTO_SWITCH_POSTGRES_HOST` parameter in the commands. The examples below include the `AUTO_SWITCH_POSTGRES_HOST` parameter.

Here is an example:

```
{ "op": "replace", "path": "/data/AUTO_SWITCH_POSTGRES_HOST", "value": "TRUE" }, { "op": "replace", "path": "/data/SQL_PROXY_POSTGRES_CONNECTION_MAP", "value": "value" }
```

The value for the `SQL_PROXY_POSTGRES_CONNECTION_MAP` parameter is a comma-separated list of key-value pairs that describes the mapping of the SQL proxy Kubernetes deployment name to the new PostgreSQL connection string. The key and value within each KV pair is separated

by a colon (:), where the key is the SQL proxy Kubernetes deployment name for the PostgreSQL cluster and the value is the connection string for the recovered PostgreSQL cluster

There is one SQL proxy deployment for each PostgreSQL cluster that is configured with the SAS Viya platform environment. The names of the SQL proxy deployments are **platform-postgres-sql-proxy** for the default Postgres instances and the **cds-proxy-sql-postgres** for the CDS PostgreSQL instances. To retrieve the list of all the SQL proxy deployments, use this command:

```
kubectl -n name-of-namespace get deployments | grep -i "sql-proxy"
```

Here is an example of the **SQL_PROXY_POSTGRES_CONNECTION_MAP** parameter with sample KV pairs:

```
{ "op": "replace", "path": "/data/SQL_PROXY_POSTGRES_CONNECTION_MAP", "value": "platform-postgres-sql-proxy:sub7:us-east1:restored-postgres-default-pgsql-clone,cds-postgres-sql-proxy:sub7:us-east1:restored-cds-postgres-default-pgsql-clone" }
```

Here is an example of the full command with DEBUG enabled, the **AUTO_SWITCH_POSTGRES_HOST** parameter, and the **SQL_PROXY_POSTGRES_CONNECTION_MAP** parameter with sample KV pairs :

```
kubectl patch cm sas-restore-job-parameters-name -n name-of-namespace --type json -p '[ {"op": "replace", "path": "/data/SAS_BACKUP_ID", "value": "2020-04-01T10_22_33_633_0700"}, {"op": "replace", "path": "/data/SAS_DEPLOYMENT_START_MODE", "value": "RESTORE" }, {"op": "replace", "path": "/data/SAS_LOG_LEVEL", "value": "DEBUG" }, {"op": "replace", "path": "/data/AUTO_SWITCH_POSTGRES_HOST", "value": "TRUE" }, {"op": "replace", "path": "/data/SQL_PROXY_POSTGRES_CONNECTION_MAP", "value": "platform-postgres-sql-proxy:sub7:us-east1:restored-postgres-default-pgsql-clone,cds-postgres-sql-proxy:sub7:us-east1:restored-cds-postgres-default-pgsql-clone"} ] '
```

If you want to exclude a deprecated configuration from being restored to the target SAS Viya platform system, see one of the following files:

File Format	Path
Markdown	<code>\$deploy/sas-bases/examples/restore/configure/README.md</code>
HTML	<code>\$deploy/sas-bases/docs/configuration_settings_for_restore_using_the_sas_viya_backup_and_restore_utility.htm</code>

- If you are migrating from a SAS Viya 3.5 system with the 2.87.1 version of the sas.files package and your target SAS Viya 4 platform is a 2026.01 or 2026.02 cadence, add the following parameters to your sas-restore-job-parameters:

Note: Each parameter must be on a single line.

```
RESTORE_DEFINITION_FILTER='{ "files.sas.files.6": "*", "files.sas.files.7": "*", "files.sas.files.8": "*" }'
```

```
- RESTORE_CONFIGURATION_FILTER='{ "*.application/
vnd.sas.configuration.config.sas.files+json;version=6": "*", "*.application/
vnd.sas.configuration.config.sas.files+json;version=7": "*", "*.application/
vnd.sas.configuration.config.sas.files+json;version=8": "*" }'
```

For more information about the 2.87.1 version of the sas.files package, see [“Configuration Values Are Lost during Migration with sas-files Version 2.87.1” in SAS Viya Platform: Full-System Migration from SAS Viya 3.5](#).

5 Check if the restore job already exists.

```
kubectl get job sas-restore-job -n name-of-namespace
```

If it does exist, delete the job.

```
kubectl delete job sas-restore-job -n name-of-namespace
```

6 Create job from the restore CronJob.

```
kubectl create job -n name-of-namespace --from=cronjob/sas-restore-job sas-restore-job
```

7 The restore job was started in the previous step. Inspect the status of the job:

```
kubectl get jobs -n name-of-namespace -l "sas.com/backup-job-type=restore" -L "sas.com/sas-backup-id,sas.com/backup-job-type,sas.com/sas-restore-status"
```

Submit the command until you get the Completed status. Possible statuses are Completed, CompletedWithWarnings, and Failed.

Note: A restore job might be marked as CompletedWithWarnings when there are configuration definition mismatches between SAS Viya 3.5 and SAS Viya 4 that prevent some configurations from being applied during the migration process. In these cases, RequestValidationFailed warnings are recorded in the restore logs. The restore process still completes successfully, and the target environment remains usable. However, some configuration definitions might be skipped due to version incompatibility.

- If you get an Error status or need to rerun the job for any reason, delete the job:

```
kubectl delete job sas-restore-job -n name-of-namespace
```

And then perform this previous step again:

- If a resource is not available to the pod, the restore job might appear to be running with "0/1" completions, whereas it actually is in a waiting state. You might want to describe the restore pod to confirm that the job has been able to start.

Get the name of the restore pod:

```
kubectl get pods -n name-of-namespace | grep sas-restore-job
```

Describe the pod, substituting the restore pod name that you identified in the last step:

```
kubectl -n name-of-namespace describe pod sas-restore-job-pod-name
```

8 Check the log of the restore job to confirm that the restore was successful.

To view the log of the restore job, find its pod name:

```
kubectl get pods -n name-of-namespace | grep sas-restore-job
```

To see the log:

```
kubectl -n name-of-namespace logs sas-restore-job-pod-name -c sas-restore-job
```

- 9 While the restore job is running, the CAS controller pods and CAS worker pods are scaled down. Before proceeding to the next step, do the following:

Note: At this point, while CAS-dependent pods might be in a **Running** state, requests to those pods might fail until CAS is successfully restarted in a later step.

- Make sure that the CAS controller pods and CAS worker pods have been scaled down.

```
kubectl get pods --selector="app.kubernetes.io/managed-by==sas-cas-operator" -n name-of-namespace
```

- 10 The file system restore requires a clean volume. Otherwise, the restore for the volume is skipped.

Note: If you are restoring to an OpenShift cluster, see the "Granting Security Context Constraints for Copy and Cleanup Job on an OpenShift Cluster" section in one of the following files for information about granting security context constraints before proceeding:

File Format	Path
Markdown	<code>\$deploy/sas-bases/examples/restore/scripts/README.md</code>
HTML	<code>\$deploy/sas-bases/docs/restore_scripts.htm</code>

- a Navigate to the `$deploy/sas-bases/examples/restore/scripts` folder.

- b Add Execute permission to the scripts used in the following steps.

```
chmod +x ./sas-backup-pv-copy-cleanup.sh ./scale-up-cas.sh
```

- c Run the `sas-backup-pv-copy-cleanup` script to clean up the CAS PVs. The script requires the following three arguments:

<i>namespace</i>	Defines the namespace.
<i>operation</i>	Defines the operation that the script performs. Valid values are <code>remove</code> and <code>copy</code> .
<i>tenant-list</i>	In a non-multi-tenancy environment, specify <code>default</code> .

Here is the command:

```
sas-backup-pv-copy-cleanup.sh namespace operation "tenant-list"
```

Here is an example of a command to remove the CAS PV in a non-multi-tenancy environment:

```
sas-backup-pv-copy-cleanup.sh viyans remove "default"
```

If you get the following error and this is the first attempt at a restore after a successful SAS Viya 3.x to SAS Viya platform migration, you might need to perform additional steps to clean the CAS PVs:

```
./sas-backup-pv-copy-cleanup.sh $NS remove shared
./sas-backup-pv-copy-cleanup.sh: line 100: [: : integer expression expected
```

To resolve, see one of the following files:

File Format	Path
Markdown	<code>\$deploy/sas-bases/examples/restore/scripts/README.md</code>
HTML	<code>\$deploy/sas-bases/docs/restore_scripts.htm</code>

11 Scale up the CAS deployment (restore the CAS data).

- Navigate to the `$deploy/sas-bases/examples/restore/scripts` folder.
- Run the `scale-up-cas` script to scale up the CAS deployment. This step restores the CAS data. The script requires the following two arguments:

<code>namespace</code>	Defines the namespace.
<code>tenant-list</code>	In a non-multi-tenancy environment, specify <i>default</i> .

Here is the command:

```
scale-up-cas.sh namespace "tenant-list"
```

Here is an example of a command to scale up the CAS deployment in a non-multi-tenancy environment:

```
scale-up-cas.sh viyans "default"
```

12 Confirm that the deployment is running. Verify that all pods are running successfully:

```
kubectl get pods -n name-of-namespace
```

Make sure that you are able to sign on to the system.

13 Check the log of the `cas-controller` pod for each participating CAS server for these messages to confirm that the restore was successful:

- `"message": "Restore of CAS permstore volume content completed successfully."`
- `"message": "Restore of CAS data volume content completed successfully."`
- `"message": "Restore of CAS data and permstore volumes completed successfully."`

If all of these messages are present in the log, then the CAS restore was successful. If neither or only one of these messages are present in the log along with an error message, then the CAS restore was not successful. Check the CAS pod logs for details.

Here is a sample command to check the CAS pod logs:

```
kubectl -n name-of-namespace logs sas-cas-server-default-controller -c sas-cas-server | grep "Restore of CAS"
```

- 14 To complete the switch over to the newly restored Postgres instance, restart all the services with the sas-stop-all and sas-start-all CronJobs.

Note: You can skip this step if you restored internal Crunchy Data PostgreSQL using pgBackRest.

```
kubectl -n name-of-namespace create job --from cronjobs/sas-stop-all job-name
```

```
kubectl -n name-of-namespace create job --from cronjobs/sas-start-all job-name
```

- 15 Ensure that SAS Viya platform is now referencing the restored PostgreSQL instance. The steps differ if a SQL proxy is used to connect to the new PostgreSQL instance.

Note: You can skip this step if you restored internal Crunchy Data PostgreSQL using pgBackRest.

- If no SQL proxy is used, execute the following command for each PostgreSQL instance in the cluster:

```
kubectl -n name-of-namespace exec -it sas-consul-server-0 -- /bin/bash -c "export
CONSUL_HTTP_ADDR=https://localhost:8500;/opt/sas/viya/home/bin/sas-bootstrap-config
catalog service serviceID"
```

Note:

- In the above command, if you are using full Transport Layer Security (TLS) use HTTPS as the protocol. If you are using Front-door TLS or no TLS at all, use HTTP as the protocol.
 - In the above command, the serviceID for the default Postgres instance is *sas-postgres*. The serviceID for the CDS Postgres instance is *sas-cdspostgres*.
-

- If a SQL proxy is used, execute the following command for each PostgreSQL instance in the cluster:

```
kubectl -n name-of-namespace get deployments | grep -i "sql-proxy"
```

For each result returned from the above command, use the following command to verify that it is pointing to the connection string of the restored PostgreSQL:

```
kubectl -n name-of-namespace describe deployment deployment-name | grep -i
"POSTGRES_CONNECTION_NAME:"
```

- 16 Reset all configMap sas-restore-job-parameters.

Here is an example:

```
kubectl patch cm sas-restore-job-parameters-name -n name-of-namespace --type json -p
'[{ "op": "remove", "path": "/data/SQL_PROXY_POSTGRES_CONNECTION_MAP" }, { "op": "remove",
"path": "/data/AUTO_SWITCH_POSTGRES_HOST" }, { "op": "remove", "path": "/data/
DATASERVER_HOST_MAP" }, { "op": "remove", "path": "/data/SAS_BACKUP_ID" }, { "op": "remove",
"path": "/data/SAS_DEPLOYMENT_START_MODE" }, { "op": "remove", "path": "/data/
SAS_LOG_LEVEL" } ]'
```

- 17 Update the manifests with the new host names. The steps differ if a SQL proxy is used to connect to the new PostgreSQL instance.

Note: You can skip this step if you restored internal Crunchy Data PostgreSQL using pgBackRest.

- Update the charts with the new host names after restore without a SQL proxy.

External PostgreSQL hosts must have been set as per the instructions in steps 6 through 9 in the "External PostgreSQL Configuration" section in the one of the following files:

File Format	Path
Markdown	<code>\$deploy/sas-bases/examples/postgres/README.md</code>
HTML	<code>\$deploy/sas-bases/docs/external_postgresql_configuration.htm</code>

See steps 6 through 9 where `{{ POSTGRES-SERVER-NAME }}-dataserver-transformer.yaml` is modified and referenced in the base `kustomization.yaml` file. Ensure that `{{ POSTGRES-SERVER-NAME }}-dataserver-transformer.yaml` is updated to point to the new hosts for every PostgreSQL server. Generally, these files are present in the `$deploy/site-config/postgres/` directory.

- Update the charts with the new host names after restore with a SQL proxy.

Note: This is used If you are using Google Cloud SQL for PostgreSQL

External PostgreSQL hosts must have been set as per the instructions in steps 3 and 4 in the "Google Cloud Platform Cloud SQL for PostgreSQL Prerequisites" section in the one of the following files:

File Format	Path
Markdown	<code>\$deploy/sas-bases/examples/postgres/README.md</code>
HTML	<code>\$deploy/sas-bases/docs/external_postgresql_configuration.htm</code>

See steps 3 and 4 where `{{ POSTGRES-SERVER-NAME }}-cloud-sql-proxy.yaml` is modified and referenced in the base `kustomization.yaml` file. Ensure that `{{ POSTGRES-SERVER-NAME }}-cloud-sql-proxy.yaml` is modified to update the value of the `POSTGRES_CONNECTION_NAME` variable with the connection string for every PostgreSQL server that is restored.. Generally, these files are present in the `$deploy/site-config/postgres/` directory.

These changes will be implemented in the manifest files the next time they are built.

Note: If you use the SAS Deployment Operator to manage the deployment and you have configured the operator to automatically apply any cadenceRelease updates, you should rebuild the custom-resource (CR) file now. See [“Manage Updates” in SAS Viya Platform: Deployment Guide](#) for information about this feature. See [“Run the create sas-deployment-cr Command” in SAS Viya Platform: Deployment Guide](#) for the steps to rebuild the CR file.

- 18 If you are using SCIM for identity management, you must re-create the access token. See [“How to Configure SCIM” in SAS Viya Platform: Identity Management](#).
- 19 If your software order includes offerings that include Process Orchestration, you must do the following to make sure that flows run after the restore:
 - a Obtain a new access token and set the **ACCESS_TOKEN** environment variable. See [“Obtain an Access Token” in SAS Viya Platform: Authentication](#).
 - b Run the following command:

```
curl -k -x POST "${INGRESS_URL}/processOrchestration/restore" \
  -H "Authorization: Bearer $ACCESS_TOKEN"
```

Post-Restore Validation

SAS Viya platform provides the SAS Viya platform Inventory and SAS Viya platform Comparison reports that you can use to validate the target SAS Viya platform environments. These reports, and the associated SASVIYATYPES table are accessible by SAS Environment Manager. To use the reports, you must first generate additional required data by performing an inventory scan. To run an inventory scan, you must create a Kubernetes job from the CronJob. See [SAS Viya Platform: Inventory](#) for more information.

List Restores

To list restores:

```
kubectrl get jobs -n name-of-namespace -l "sas.com/backup-job-type=restore" -L "sas.com/sas-backup-id,sas.com/backup-job-type,sas.com/sas-restore-status"
```

Show Details of a Restore

To show details for a restore:

```
kubectrl -n name-of-namespace describe jobs sas-restore-job
```

View Restore Logs

To view the log of a specific restore job, find its pod name:

```
kubectl get pods -n name-of-namespace | grep sas-restore-job
```

To see the log:

```
kubectl -n name-of-namespace logs sas-restore-job-pod-name -c sas-restore-job
```

Backup and Restore: Purge Backups

Note: During deployment, you created a deployment directory that SAS recommended that you name `deploy` or an alternative name that is meaningful to you. The directory is referred to as `$deploy` in this guide. Replace `$deploy` with the directory name that you choose

Purge Backups

Note: For deployments in OpenShift environments, a Kubernetes administrator must create and configure a Security Context Constraint (SCC). When CAS is deployed, administrators can choose and associate appropriate SCCs with the `sas-cas-server` service account, based on the CAS server configuration such as host launch, SSSD, or others. To perform the restore process, a Kubernetes administrator must create and associate a corresponding SCC with the `sas-viya-backuprunner` service account that aligns with the CAS server configuration. After the SCC is created and set up, namespace administrators can perform the restore operation.

Backups are purged through a CronJob. The CronJob finds backups past the retention period (the default is 30 days) and purges them from all backup data sources. Here is the workflow:

- 1 Purging is performed through a CronJob that executes daily at 12:15 a.m. To get details about the CronJob, submit the following command:

```
kubectl describe cronjob sas-backup-purge-job
```

- 2 The purge operation lists any Kubernetes resource of type JOBS that has a label related to back up. For example:

```
sas.com/sas-backup-id=sas.com/sas-backup-id=2020-05-05T07_34_12_612_0700
```

```
sas.com/sas-backup-persistence-status=available
```

```
sas.com/sas-backup-job-status=completed
```

- 3 Purging a backup job is based on the following conditions:

- The backup job is past the retention period.
- The backup job is not already purged.

IMPORTANT During a purge, all successful backups are considered. The most recent successful backup is set aside. This backup is not purged, even if it is past the retention period.

- 4 If there are backup jobs that need to be purged, then an event is sent to the backup agent.
- 5 The backup agent deletes the backup directory from the PV.
- 6 If a successful status is received from the backup agent, the backup job is deleted.
- 7 After the backup job is purged storage resources such as the space available on the PVC are validated proactively to help ensure that there is adequate space on the PVC to store data for the next scheduled backup job. This feature is enabled by default. To disable this proactive resource validation, see one of the following files:

File Format	Path
Markdown	<code>\$deploy/sas-bases/examples/ backup/configure/README.md</code>
HTML	<code>\$deploy/sas-bases/docs/ configuration_settings_for_backup _using_the_sas_viya_backup_and_re store_utility.htm</code>

Note: If you are purging full backups that have associated incremental backups, the following rules apply:

- A full backup is purged if it has passed the retention period, and if the last of any associated incremental backups has also passed the retention period. In such cases, all associated incremental backups are purged along with the full backup.
- To qualify for purging, an incremental backup must have an associated parent full backup present in the backup directory.

By default, when a backup is purged, the job that created the backup is also purged. If you want to save the backup job when the backup is purged, set the property `RETAIN_BACKUP_JOB_HISTORY_AFTER_PURGE` to `TRUE` as follows:

- 1 Navigate to the `$deploy` directory.
- 2 Edit the `kustomization.yaml` file, and add `RETAIN_BACKUP_JOB_HISTORY_AFTER_PURGE=TRUE` in the `literals` section as follows:

```
configMapGenerator:
- name: sas-backup-job-parameters
  behavior: merge
```

```

literals:
- RETAIN_BACKUP_JOB_HISTORY_AFTER_PURGE=TRUE

```

3 Build the manifest to save the job for the purged backup.

```
kustomize build -o site.yaml
```

4 Apply the manifest.

```

kubectl apply --selector="sas.com/admin in (cluster-api,cluster-wide,cluster-
local,namespace)" -f site.yaml --server-side --force-conflicts

```

If the `RETAIN_BACKUP_JOB_HISTORY_AFTER_PURGE` property is `TRUE`, then the status of the jobs for which the backups are purged is set to `purged`. To view the jobs with a status of `purged`, issue this command:

```

kubectl -n name-of-namespace get jobs -l "sas.com/sas-backup-persistence-status=purged" -L
"sas.com/sas-backup-persistence-status"

```

Backup and Restore: Recover from a Disaster

Note:

- During deployment, you created a deployment directory that SAS recommended that you name `deploy` or an alternative name that is meaningful to you. The directory is referred to as `$deploy` in this guide. Replace `$deploy` with the directory name that you choose
 - For deployments in OpenShift environments, a Kubernetes administrator must create and configure a Security Context Constraint (SCC). When CAS is deployed, administrators can choose and associate appropriate SCCs with the `sas-cas-server` service account, based on the CAS server configuration such as `host launch`, `SSSD`, or others. To perform the restore process, a Kubernetes administrator must create and associate a corresponding SCC with the `sas-viya-backuprunner` service account that aligns with the CAS server configuration. After the SCC is created and set up, namespace administrators can perform the restore operation.
-

Overview

Disaster recovery in the SAS Viya platform is the process for recovering in the event of a disaster. Recovery in this context refers to regaining access to data, hardware, and software.

Note: The disaster recovery steps are the same as the restore steps from a backup. However, there are additional restrictions in place for disaster recovery. The assumption is that you are trying to re-create the system in its pre-disaster state. Consequently, you cannot change cloud provider during a

disaster recovery, an option that is supported when following the standard [“Backup and Restore: Perform a Restore”](#) steps.

Here are the steps.

- Prepare the target environment by installing the SAS Viya platform using the deployment files from the source environment.
- Make sure that the persistent volumes (PVs) that contain backup data of the source environment are accessible from the target environment.
- Perform a restore using the most recent successful backup of the source environment.

The steps above restore your source environment to a SAS Viya platform target environment that is the same as the source environment with respect to the following items:

- the name of the namespace
- the cloud infrastructure
- the set of SAS Viya platform products and the version

Disaster Recovery Prerequisites and Planning on the Source Environment

- Develop a plan for disaster recovery specific for your organization. This disaster recovery plan should contain a documented process for preparing your organization to recover in the shortest amount of time in the event of a disaster.
- Make sure to set the Kubernetes reclaim policy for PVs to Retain instead of Delete, in order to prevent accidentally losing backups stored in backup PVs if the namespace is deleted.
- Restoring a backup from a different cadence is not supported. For example, you can restore to a SAS Viya platform Stable environment from a backup of a SAS Viya platform Stable environment. You can restore to a SAS Viya platform Long-Term Support environment from a backup of a SAS Viya platform Long-Term Support environment. But you cannot restore to a SAS Viya platform Long-Term Support environment from a backup of a SAS Viya platform Stable environment. See [“Software Releases and Versions” in *Getting Started with SAS Viya Platform Operations*](#) for more information.
- Restoring a backup from a different cadence version is not supported. For example, you can restore to a SAS Viya platform 2026.02 environment from a backup of a 2026.02 environment. But you cannot restore to a 2026.02 environment from a backup of a 2026.01 or earlier environment. Starting in 2026.02, when you restore from a backup of a different cadence, the restore job is marked as failed. See [“Software Releases and Versions” in *Getting Started with SAS Viya Platform Operations*](#) for more information.

Note:

The cadence and cadence version of the deployment are reported in the status.json file for each backup. The path location of the status.json file is `backupID/status.json` in the sas-common-backup-data PVC. Here is an example of the cadence and cadence version that is included in the status.json file:

CADENCE_RELEASE: 20250123.1737607638742

CADENCE_DISPLAY_NAME: Long-Term Support 2026.05

Unless you have access to the sas-common-backup data PVC outside of the SAS Viya platform environment, you must use the sas-backup-copy-job to get a pod running that has the sas-common-backup data PVC mounted in it. To do so, make sure that the copy pod is running. If the copy pod is not running, do step 3a [here on page 64](#).

Once the copy pod is running, do the following steps:

- Get the name of the running copy pod.

```
kubectl -n name-of-namespace get pods -l sas.com/backup-job-type=sas-backup-pv-copy-cleanup
```

- Get the cadence information.

```
kubectl -n name-of-namespace exec copy-pod-name -- sh -c "jq '{CadenceRelease: .CadenceRelease, CadenceDisplayName: .CadenceDisplayName}' /sasviyabackup/backup-id/status.json"
```

- For information about using PostgreSQL database options to improve the performance of the restore, see [“Tuning SAS Backup for PostgreSQL Database Restore Operations” in SAS Viya Platform Administration: Tuning](#).
- Resources such as CPU and memory are pre-validated by default to help ensure that the restore job completes successfully. If the pre-validation fails, the restore job will fail. If you want to disable the resource validation, see one of the following files:

File Format	Path
Markdown	<code>\$deploy/sas-bases/examples/restore/configure/README.md</code>
HTML	<code>\$deploy/sas-bases/docs/configuration_settings_for_restore_using_the_sas_viya_backup_and_restore_utility.htm</code>

For more information about the resource requirements for restore, see [“Resource Requirements for Restore” on page 78](#).

- In order to maximize your ability to recover from a disaster, you should do the following on an ongoing basis:
 - If you are using the SAS internal PostgreSQL pgBackRest utility for the backup and recovery of internal Crunchy Data PostgreSQL, SAS recommends that you continue to schedule periodic backups that include PostgreSQL. Backups that use the SAS Viya backup and restore utility for all data are required for disaster recovery and SAS Viya platform to SAS Viya platform migrations. Create these backups by setting the `INCLUDE_POSTGRES` configMap parameter to `true`. Alternatively, you can use the new CronJob designed for this feature. See [“Enable Suspended CronJob to Back Up All Content with the SAS Viya Platform \(Including PostgreSQL\)” on page 21](#).

- ❑ Make sure that the source Kubernetes environment is functioning as expected, and that the frequency of performing [backups](#) is as specified in the [recovery point objective \(RPO\)](#) set by your organization. Make sure that the scheduled backups are successful.
- ❑ Perform a [scan](#) of the source Kubernetes environment before each backup.
- ❑ Make sure that the data is backed up correctly by the backup job, and that the status of the most recent backup is **Completed**.
- ❑ Make sure that the persistent volumes (PVs) that contain backup data are secure by verifying that the necessary replication is configured.
- ❑ Make sure that the backups are stored safely.
- ❑ Make sure that you know the physical location of the PVs that contain backup data. See [“How to Find the Physical Location of the Persistent Volume \(PV\) in an Azure Environment”](#) on page 62 for examples of finding the physical location of PVs. Make a note of the PV location information and save it in a file. It is recommended that you save this file in the same location where the `$deploy` directory is stored.
- ❑ Practice the restore process using a successful backup of the source system.
- If you store audit files in an audit archive file system on your source Kubernetes environment and you want the ability to recover these audit archive files in the event of a disaster, you must back up that file system separately as part of your site security policy. To learn more about audit archive backup, restore, and migration, see [“Backup, Restore, and Migrate the Audit Archive”](#) in *SAS Viya: Auditing*.
- If the number of CAS servers is reduced on the target environment, the data from the dropped server is not restored.

Perform a Disaster Recovery

- 1 See [“Determine the Backup ID of the Last Successful Backup”](#) on page 72 to get the backup ID.
- 2 Prepare the target environment.

Note:

- In 2022.12 and later releases of the SAS Viya platform, the term “manual deployment” is replaced by “deployment using Kubernetes commands”.
-

In the following steps, deploy using the Kubernetes commands deployment method of the SAS Viya platform.

- a See [“Support for Migration from SAS Viya 4 to SAS Viya 4 for Offerings”](#) in *SAS Viya Platform: System Migration* for links to additional steps that might be required for some offerings.
- b A virtual infrastructure and certain hardware and resources are required for the deployment. Review [“Virtual Infrastructure Requirements”](#) in *System Requirements for the SAS Viya Platform* and [“Storage Requirements”](#) in *System Requirements for the SAS Viya Platform* to ensure that all requirements are met.

For example, one requirement is that you must create a dedicated namespace for the deployment. Create the namespace that is mentioned in the root `kustomization.yaml` file in the `$deploy` directory.

```
kubectl create -n name-of-namespace
```

- c See [“Sizing Recommendations for Restore”](#) for information about the PostgreSQL database size in the restore space.
- d Copy the `$deploy` directory saved from the source environment to the `$deploy` directory on the target environment.
- e Perform any necessary [pre-installation tasks](#).
- f If you need to make any changes to your manifest, see [“Required Customizations” in SAS Viya Platform: Deployment Guide](#), or [“Optional Customizations” in SAS Viya Platform: Deployment Guide](#) and the README files that this section describes.

IMPORTANT The SAS Viya platform database (commonly named SharedServices) is not created automatically for external PostgreSQL during the initial deployment of the SAS Viya platform. If the external PostgreSQL database is included in the SAS Viya platform backup that is being used for deployment, then this database must be created before deploying the SAS Viya platform. Ensure that the username and password configured for the SharedServices database in the target environment are the same as in the source environment. See [“External PostgreSQL Requirements” in System Requirements for the SAS Viya Platform](#) for more information.

- g See [“Deployment Using Kubernetes Commands” in SAS Viya Platform: Deployment Guide](#) to deploy the software.

Note: If the backup that you are using for the source SAS Viya platform environment was created by a migration or has undergone a backup or restore, make sure that the `kustomization.yaml` file does not contain any references to migration or backup jobs from the source environment

- h Perform any necessary post-installation steps. See [“Post-Installation Tasks” in SAS Viya Platform: Deployment Guide](#) for more information.
- i After completing the installation of SAS Viya platform, ensure that the software is deployed correctly and works as expected by logging in to the SAS Logon application.
- j Run an [ad hoc backup on page 11](#) to ensure that deployment is working correctly. This backup can be helpful if you want to roll back to the original state for the target environment prior to restore.

- 3 Ensure that the backup contents are accessible from the target environment.

Note:

Multiple persistent volumes (PVs) are used for backup.

Run this command to list the PVCs where backup is stored:

```
kubectl -n namespace get pvc -l "sas.com/backup-role=storage"
```

- If you are reusing the PVs from the source environment (static provisioning), then verify that the backup contents are already available.
- If you are not reusing the PVs from the source environment (dynamic provisioning), then you need to copy the backup content from the source environment PV to the respective PV on the target environment. Make sure to preserve the permissions and date timestamp while copying.

Copy the backup data from the source environment to the target environment. See [“Examples of Transferring Backup Data between the SAS Viya Platform Environment and the Local File System”](#) for a generic example for copying data. See [“Copy Backup Data from Azure Environment to Azure Environment” on page 70](#) for examples of copying backup PV data from one Azure environment to another Azure environment.

- 4 Perform a restore of the last successful backup using the backup ID identified in the planning step.

If you are restoring using provider-specific processes for PostgreSQL, see [“Perform a Restore Using Provider-Specific Processes for PostgreSQL” on page 35](#) for instructions about how to restore. If you are restoring using only SAS Viya processes for content, continue to the steps below.

Use the same steps as if you are doing a normal restore.

Note: Incremental backups are not supported when using the SAS Viya backup and restore utility to back up and recover all data. Incremental backups are supported only when the INCLUDE_POSTGRES configMap parameter is set to *false*, meaning that PostgreSQL is not included in SAS Viya platform backup.

- a If you want to use the new restore steps in your release of the SAS Viya platform, continue with the instructions in the next paragraph and each of the steps below:

Identify the sas-restore-job-parameters configMap that needs to be modified.

```
kubectl -n name-of-namespace describe cronjob sas-restore-job | grep -i sas-restore-job-parameters | awk '{print $1}' | head -n 1
```

- b Edit the configMap sas-restore-job-parameters. Include the parameters and values in a comma-separated list at the end of the command. For each parameter, specify an operation, path, and value. The SAS_BACKUP_ID and SAS_DEPLOYMENT_START_MODE parameters are always required.

```
kubectl patch cm sas-restore-job-parameters-name -n name-of-namespace --type json -p '[{"op": "operation", "path": "/data/parameter-name", "value": "value"}, {"op": "operation", "path": "/data/parameter-name", "value": "value"}]'
```

Here is an example with sample values.

```
kubectl patch cm sas-restore-job-parameters-name -n name-of-namespace --type json -p '[{"op": "replace", "path": "/data/SAS_BACKUP_ID", "value": "2020-04-01T10_22_33_633_0700"}, {"op": "replace", "path": "/data/SAS_DEPLOYMENT_START_MODE", "value": "RESTORE"}]'
```

Here is an example with sample values and DEBUG enabled.

```
kubectl patch cm sas-restore-job-parameters-name -n name-of-namespace --type json -p '[{"op": "replace", "path": "/data/SAS_BACKUP_ID",
```

```
"value":"2020-04-01T10_22_33_633_0700"}, {"op": "replace", "path": "/data/
SAS_DEPLOYMENT_START_MODE", "value":"RESTORE" }, {"op": "replace", "path": "/data/
SAS_LOG_LEVEL", "value":"DEBUG" }]]'
```

Here are more details about the previous code.

- Replace the value for SAS-BACKUP-ID with the ID of the backup that is selected for restore.
 - If you are performing a backup that uses the SAS Viya backup and restore utility to back up and recover all data, then select a backup for which the INCLUDE_POSTGRES configMap parameter is set to *true*. See [“List Backups Where PostgreSQL Is Included” on page 18](#).
 - If you are performing a backup that uses provider-specific processes for PostgreSQL, then select a backup for which the INCLUDE_POSTGRES configMap parameter is set to *false*. See [“List Backups Where PostgreSQL Is Not Included” on page 18](#).
- To increase the logging levels, add the SAS_LOG_LEVEL parameter to the list with the value `DEBUG`. SAS Technical Support recommends increasing log levels to `DEBUG` before performing a restore.

IMPORTANT Replace each variable with the value that is appropriate for your deployment. For example, *SAS-BACKUP-ID* should be replaced with a backup ID, such as 20200104-102233F.

If you want to exclude a deprecated configuration from being restored to the target SAS Viya platform system, see one of the following files:

File Format	Path
Markdown	<code>\$deploy/sas-bases/examples/ restore/configuration/README.md</code>
HTML	<code>\$deploy/sas-bases/docs/ configuration_settings_for_resto re_using_the_sas_viya_backup_and _restore_utility.htm</code>

- c If you are migrating from a SAS Viya 3.5 system with the 2.87.1 version of the `sas.files` package and your target SAS Viya 4 platform is a 2026.01 or 2026.02 cadence, add the following parameters to your `sas-restore-job-parameters`:

Note: Each parameter must be on a single line.

```
RESTORE_DEFINITION_FILTER='{ "files.sas.files.6": "*", "files.sas.files.7": "*", "files.
sas.files.8": "*" }'
```

```
- RESTORE_CONFIGURATION_FILTER='{ "*.application/
vnd.sas.configuration.config.sas.files+json;version=6": "*", "*.application/
```

```
vnd.sas.configuration.config.sas.files+json;version=7":"*", "*.application/
vnd.sas.configuration.config.sas.files+json;version=8":"*"}'
```

For more information about the 2.87.1 version of the sas.files package, see [“Configuration Values Are Lost during Migration with sas-files Version 2.87.1” in SAS Viya Platform: Full-System Migration from SAS Viya 3.5.](#)

- d Check if the restore job already exists.

```
kubectl get job sas-restore-job -n name-of-namespace
```

If it does exist, delete the job.

```
kubectl delete job sas-restore-job -n name-of-namespace
```

- e Create job from the restore CronJob.

```
kubectl create job -n name-of-namespace --from=cronjob/sas-restore-job sas-restore-job
```

- f The restore job was started in the previous step. Inspect the status of the job:

```
kubectl get jobs -n name-of-namespace -l "sas.com/backup-job-type=restore" -L "sas.com/
sas-backup-id,sas.com/backup-job-type,sas.com/sas-restore-status"
```

Submit the command until you get the Completed status. Possible statuses are Completed, CompletedWithWarnings, and Failed.

Note: A restore job might be marked as CompletedWithWarnings when there are configuration definition mismatches between SAS Viya 3.5 and SAS Viya 4 that prevent some configurations from being applied during the migration process. In these cases, RequestValidationFailed warnings are recorded in the restore logs. The restore process still completes successfully, and the target environment remains usable. However, some configuration definitions might be skipped due to version incompatibility.

- If you get an Error status or need to rerun the job for any reason, delete the job:

```
kubectl delete job sas-restore-job -n name-of-namespace
```

And then perform this previous step again:

- If a resource is not available to the pod, the restore job might appear to be running with "0/1" completions, whereas it actually is in a waiting state. You might want to describe the restore pod to confirm that the job has been able to start.

Get the name of the restore pod:

```
kubectl get pods -n name-of-namespace | grep sas-restore-job
```

Describe the pod, substituting the restore pod name that you identified in the last step:

```
kubectl -n name-of-namespace describe pod sas-restore-job-pod-name
```

- g Check the log of the restore job to confirm that the restore was successful.

To view the log of the restore job, find its pod name:

```
kubectl get pods -n name-of-namespace | grep sas-restore-job
```

To see the log:

```
kubectl -n name-of-namespace logs sas-restore-job-pod-name -c sas-restore-job
```

- h While the restore job is running, all pods get restarted with the exception of the CAS controller pods and CAS worker pods, which are scaled down. Before proceeding to the next step, do the following:

- Make sure that the sas-logon-app and sas-configuration are running.

```
kubectl get pods -n name-of-namespace | grep sas-logon
```

```
kubectl get pods -n name-of-namespace | grep sas-configuration
```

Note: At this point, while CAS-dependent pods might be in a **Running** state, requests to those pods might fail until CAS is successfully restarted in a later step.

- Make sure that the CAS controller pods and CAS worker pods have been scaled down.

```
kubectl get pods --selector="app.kubernetes.io/managed-by==sas-cas-operator" -n name-of-namespace
```

- i The file system restore requires a clean volume. Otherwise, the restore for the volume is skipped.

Note: If you are restoring to an OpenShift cluster, see the "Granting Security Context Constraints for Copy and Cleanup Job on an OpenShift Cluster" section in one of the following files for information about granting security context constraints before proceeding:

File Format	Path
Markdown	<code>\$deploy/sas-bases/examples/restore/scripts/README.md</code>
HTML	<code>\$deploy/sas-bases/docs/restore_scripts.htm</code>

- 1 Navigate to the `$deploy/sas-bases/examples/restore/scripts` folder.

- 2 Add Execute permission to the scripts used in the following steps.

```
chmod +x ./sas-backup-pv-copy-cleanup.sh ./scale-up-cas.sh
```

- 3 Run the sas-backup-pv-copy-cleanup script to clean up the CAS PVs. The script requires the following three arguments:

<i>namespace</i>	Defines the namespace.
<i>operation</i>	Defines the operation that the script performs. Valid values are <code>remove</code> and <code>copy</code> .
<i>tenant-list</i>	In a non-multi-tenancy environment, specify <code>default</code> .

Here is the command:

```
sas-backup-pv-copy-cleanup.sh namespace operation "tenant-list"
```

Here is an example of a command to remove the CAS PV in a non-multi-tenancy environment:

```
sas-backup-pv-copy-cleanup.sh viyans remove "default"
```

If you get the following error and this is the first attempt at a restore after a successful SAS Viya 3.x to SAS Viya platform migration, you might need to perform additional steps to clean the CAS PVs:

```
./sas-backup-pv-copy-cleanup.sh $NS remove shared
./sas-backup-pv-copy-cleanup.sh: line 100: [: : integer expression expected
```

To resolve, see one of the following files:

File Format	Path
Markdown	<code>\$deploy/sas-bases/examples/restore/scripts/README.md</code>
HTML	<code>\$deploy/sas-bases/docs/restore_scripts.htm</code>

j Scale up the CAS deployment (restore the CAS data).

- 1 Navigate to the `$deploy/sas-bases/examples/restore/scripts` folder.
- 2 Run the scale-up-cas script to scale up the CAS deployment. This step restores the CAS data. The script requires the following two arguments:

namespace	Defines the namespace.
tenant-list	In a non-multi-tenancy environment, specify <i>default</i> .

Here is the command:

```
scale-up-cas.sh namespace "tenant-list"
```

Here is an example of a command to scale up the CAS deployment in a non-multi-tenancy environment:

```
scale-up-cas.sh viyans "default"
```

k Confirm that the deployment is running. Verify that all pods are running successfully:

```
kubectl get pods -n name-of-namespace
```

Make sure that you are able to sign on to the system.

- l Check the log of the cas-controller pod for each participating CAS server for these messages to confirm that the restore was successful:
 - "message": "Restore of CAS permstore volume content completed successfully."

- "message": "Restore of CAS data volume content completed successfully."
- "message": "Restore of CAS data and permstore volumes completed successfully."

If all of these messages are present in the log, then the CAS restore was successful. If neither or only one of these messages are present in the log along with an error message, then the CAS restore was not successful. Check the CAS pod logs for details.

Here is a sample command to check the CAS pod logs:

```
kubectl -n name-of-namespace logs sas-cas-server-default-controller -c sas-cas-server |
grep "Restore of CAS"
```

- m Reset all configMap sas-restore-job-parameters.

Here is an example:

```
kubectl patch cm sas-restore-job-parameters-name -n name-of-namespace --type json -p
'[{ "op": "remove", "path": "/data/SAS_BACKUP_ID" }, {"op": "remove", "path": "/data/
SAS_DEPLOYMENT_START_MODE"}, { "op": "remove", "path": "/data/SAS_LOG_LEVEL" }]'
```

- n If you are using SCIM for identity management, you must re-create the access token. See [“How to Configure SCIM” in SAS Viya Platform: Identity Management](#).
- o If your software order includes offerings that include Process Orchestration, you must do the following to make sure that flows run after the restore:
 - 1 Obtain a new access token and set the **ACCESS_TOKEN** environment variable. See [“Obtain an Access Token” in SAS Viya Platform: Authentication](#).
 - 2 Run the following command:

```
curl -k -x POST "${INGRESS_URL}/processOrchestration/restore" \ -H "Authorization: Bearer
$ACCESS_TOKEN"
```

Post-Restore Validation

The SAS Viya platform provides the SAS Viya platform Inventory and the SAS Viya platform Comparison reports that you can use to validate the target SAS Viya 4 environments. These reports, and the associated SASVIYATYPES table are accessible by SAS Environment Manager. To use the reports, you must first generate additional required data by performing an inventory scan. To run an inventory scan, you create a Kubernetes job from the Cronjob. See [SAS Viya Platform: Inventory](#) for more information.

Backup and Restore: Administrator Tasks

Here are tasks that an administrator can perform as needed:

- Configure a backup. You can change a backup configuration through ConfigMap. Configuration settings include:
 - backup purge information, such as retention period (default is 30 days)
 - backup job time-out value
 - additional SAS Configuration Server (Consul) properties to back up

For information about how to change the configuration settings for backups, see one of the following files:

File Format	Path
Markdown	<code>\$deploy/sas-bases/examples/backup/configure/README.md</code>
HTML	<code>\$deploy/sas-bases/docs/configuration_settings_for_backup_using_the_sas_viya_backup_and_restore_utility.htm</code>

- List the persistent volume claims that store the backup content for products that participate in backup and restore. For example:


```
kubectl -n name-of-namespace get pvc -l "sas.com/backup-role=provider"
```
- Determine the persistent volume claim (PVC) that each backup source is stored on. For example:


```
kubectl -n name-of-namespace get pvc -l "sas.com/backup-role=storage"
```
- Determine the persistent volume claim (PVC) that corresponds to the persistent volume that each backup source is stored on. See [“Backup Details” on page 60](#) for more information.

Backup and Restore: Reference

Note: During deployment, you created a deployment directory that SAS recommended that you name deploy or an alternative name that is meaningful to you. The directory is referred to as \$deploy in this guide. Replace \$deploy with the directory name that you choose

Backup Details

Overview

A backup consists of several files contained in a time-stamped directory and distributed among multiple persistent volumes (PV). The timestamp is the backup ID. Different PVs are used for different backup sources, such as CAS and PostgreSQL. Identical timestamp directories on different PVs indicate that the files belong to the same backup. The following table shows the persistent volume claim (PVC) that corresponds to the PV that each backup source is stored on.

Backup Source	PVC
SAS Infrastructure Data Server or PostgreSQL server	sas-common-backup-data
Configuration	sas-common-backup-data
CAS	sas-cas-backup-data

IMPORTANT The PVs that contain backups are part of the cluster that they are backing up. For best practices for backups and disaster recovery, see [“Best Practices for Performing Backups”](#) on page 13 and [“Disaster Recovery Prerequisites and Planning on the Source Environment”](#) on page 49.

Incremental Backups

The default method of performing backups is a full backup that captures all of the data. You can also perform an incremental backup. Incremental backups optimize backup times by capturing only changed or differential data between successive backups.

You can schedule incremental backups more frequently than the more time-consuming full backups. For incremental backups, you must always exclude PostgreSQL from the backup by setting the `INCLUDE_POSTGRES` configMap parameter to 'false'. See [“Set the INCLUDE_POSTGRES configMap Parameter” on page 74](#) for more information. Incremental backups include full backups for all data sources except for the file system and PostgreSQL. To perform incremental backups, see [“Incremental Backups”](#), and [“Schedule an Incremental Backup” on page 23](#).

Backup ID Naming Convention for Full Backups and Incremental Backups

The backup ID for backups consists of the timestamp that the backup is taken. The format of the backup ID for full backups and incremental backups is as follows:

Table 1 Full and Incremental Backup ID Format

Type	Format ¹	Meaning
Full backup ID	YYYYMMDD-HHMMSSF	The format is a timestamp down to the nearest second (YYYYMMDD-HHMMSS). The "F" at the end indicates that it is a full backup.
Incremental backup ID	YYYYMMDD-HHMMSSF_YYYYMMDD-HHMMSSI	■ The format of the first part (YYYYMMDD-HHMMSSF) is the timestamp of the full backup from which the incremental backup originates. ■ The format of the second part (YYYYMMDD-HHMMSSI) is a timestamp for the incremental backup. The "I" at the end indicates that it is an incremental backup.

¹ In YYYYMMDD, YYYY is an integer that represents the year, MM is an integer that represents the month of the year, and DD is an integer that represents the day of the month. In HHMMSS, HH is an integer that represents the hour, MM is an integer that represents the minute, and SS is an integer that represents the second.

How to Determine the Persistent Volume Claim (PVC) That Each Backup Source Is Stored On

There might be additional persistent volumes. To list all backup persistent volume claims corresponding to these persistent volumes, use this command:

```
kubectl -n name-of-namespace get pvc -l "sas.com/backup-role=storage"
```

Here is an example of the output:

NAME	STATUS	VOLUME	CAPACITY
ACCESS MODES STORAGECLASS AGE			
sas-cas-backup-data	Bound	pvc-ce04c7d5-0dd6-4dad-99c6-bbc6e0dd00e1	8Gi
nfs-client 34d			RWX
sas-common-backup-data	Bound	pvc-724d74f1-9d79-4639-9c77-399b3c3ad10a	25Gi
nfs-client 34d			RWX

Backup Directory Structure

- Here is the backup directory structure:

directory-mount-for-backup-PVC/backup ID/__default__/source/file name

For example, suppose you backed up the PostgreSQL server in a non-multi-tenancy environment. The backup PVC is sas-common-backup-data. The backup ID is 20191028-053347F, which is the timestamp. In this scenario, the backup creates the following files:

Note: The directory mount for sas-common-backup-data is typically /sasviyabackup.

- /sasviyabackup/20191028-053347F/__default__/postgres/blobs
- /sasviyabackup/20191028-053347F/__default__/postgres/SharedServices_pg_dump.dmp
- /sasviyabackup/20191028-053347F/__default__/postgres/SharedServices_pg_dump.log
- /sasviyabackup/20191028-053347F/__default__/postgres/status.json

Where to Copy Backup Data for Recovery after a Disaster Recovery

To restore data from a backup in a disaster recovery scenario, you must copy the contents of the desired backup onto the appropriate volumes in the target environment. This content should be stored inside the directory whose name contains the backup ID of the desired backup.

For example, suppose that the backup ID of the desired backup is 20191028-053347F. You are backing up PostgreSQL data, so the backup PVC is sas-common-backup-data. In this scenario, the directory that contains the backup data that you must copy to the appropriate volumes in the target environment is **sas-common-backup-data-pvc/20191028-053347F/__default__/postgres**.

How to Find the Physical Location of the Persistent Volume (PV) in an Azure Environment

Note: The following example uses Azure. The steps might be different for you depending on your specific deployment or configuration

In this example, the storage class is azurefile.

- 1 Run the following command on the Kubernetes controller node:

```
kubect1 -n name-of-namespace get pv $(kubect1 get pvc -n name-of-namespace | grep
persistent_volume_claim | awk '{print $3}') -o json | jq '.spec'
```

Here is an example of the resulting output when running the command for the sas-common-backup-data PVC:

```
{
  "accessModes": [
    "ReadWriteMany"
  ],
  "azureFile": {
    "secretName": "azure-storage-account-fbc3e3ecf1289451c913303-secret",
    "secretNamespace": "d25366",
    "shareName": "kubernetes-dynamic-pvc-23a787c9-9488-4c80-a86e-f9b263d623ce"
  },
  "capacity": {
    "storage": "25Gi"
  },
  "claimRef": {
    "apiVersion": "v1",
    "kind": "PersistentVolumeClaim",
    "name": "sas-common-backup-data",
    "namespace": "d25366",
    "resourceVersion": "4666",
    "uid": "23a787c9-9488-4c80-a86e-f9b263d623ce"
  },
  "persistentVolumeReclaimPolicy": "Delete",
  "storageClassName": "azurefile",
  "volumeMode": "Filesystem"
}
```

- 2 Locate the azureFile section in the above JSON file to find the secretName and shareName to use for mounting the volume.

Here are these values using the sas-common-backup-data PVC output from the prior step:

secretName	azure-storage-account-fbc3e3ecf1289451c913303-secret
shareName	kubernetes-dynamic-pvc-23a787c9-9488-4c80-a86e-f9b263d623ce

- 3 Get the Azure storage account name from the value of secretName. Here is an example using the sample values from the prior step:

```
STORAGEACCT=fbc3e3ecf1289451c913303
```

Examples of Transferring Backup Data between the SAS Viya Platform Environment and the Local File System

The following examples use a generic method (`tar` and `kubectl exec`) and a customized pod to copy data between environments. The steps in this generic method can be used for different cloud

providers and are not specific to any one cloud provider. You might follow a slightly different set of steps depending on what type of storage you are using for your data.

Note: Make sure that the user that issues the tar command to copy files has the appropriate permissions (root level) to copy backup data.

How to Copy Common Backup Data or CAS Backup Data between the SAS Viya Platform Environment and the Local File System Using tar and kubectl exec

- 1 Determine the location of the data that you want to copy from or to.
- 2 Navigate to the machine where kubeconfig is set to run kubectl commands.
- 3 To copy common backup data from the sas-common-backup-data PVC to the local file system or from the local file system to the sas-common-backup-data PVC, you must create a Kubernetes job. Follow the steps below:
 - a Create the copy pod.

See the "Append the Execute Permissions to Scripts" and "Copy Backup Data to and from Backup Persistent Volume Claims" sections of the README file in one of the following locations for information about creating the copy pod:

File Format	Path
Markdown	<code>\$deploy/sas-bases/examples/ restore/scripts/README.md</code>
HTML	<code>\$deploy/sas-bases/docs/ restore_scripts.htm</code>

If you have multiple instances of a CAS server, you will see multiple copy pods for each CAS instance. Make a note of the names of the copy pods that you created in the steps above.

This step creates the sas-backup-pv-copy(s). For example, if you have a single CAS Server in your deployment, then you will see that a single pod is created and two PVC(s), sas-common-backup-data and sas-cas-backup-data, are mounted.

The job creates the following mount paths for the target environment

Pod	PVC	Mount path
sas-backup-pv-copy-xxxxx	sas-common-backup-data	/sasviyabackup
	sas-cas-backup-data	/cas

- b Get the name of the running pod.

Note: The pod name that results from the following command is the pod name of the sas-backup-pv-copy-cleanup job. Replace *sas-pod-name* in the following commands in this topic with this pod name.

```
kubectl -n name-of-namespace get pods -l sas.com/backup-job-type=sas-backup-pv-copy-cleanup
```

- c You can copy the data to the target location in uncompressed format (as it is currently) or you can compress the backup data directly to the target location to reduce the space requirements. If you need to compress or decompress the data, see the step [here on page 66](#). If you do not need to compress or decompress the data, do the following steps:

Copy the backup data to the target location using the tar and kubectl exec command.

Note: See “List All Executed Backups and Their Status” on page 18 for the command to get the backup ID that is needed in the following steps.

- 1 If you are not restoring to an alternate namespace on OpenShift, here are examples of the command to copy sas-common backup data or sas-cas-backup data from the SAS Viya platform environment to the local file system:

Here is the example command for sas-common backup data:

```
kubectl -n name-of-namespace exec sas-pod-name -c sas-backup-pv-copy-cleanup-job -- sh -c 'tar cpf - -C /sasviyabackup $(find /sasviyabackup -maxdepth 1 -type d -regex ".*\/backup_id.*" -printf "%P\n")' > backup_id_common.tar
```

Here is the example command for sas-cas backup data:

```
kubectl -n name-of-namespace exec sas-pod-name -c sas-backup-pv-copy-cleanup-job -- sh -c 'tar cpf - -C /cas $(find /cas -maxdepth 1 -type d -regex ".*\/backup_id.*" -printf "%P\n")' > backup_id_cas.tar
```

If you are restoring to an alternate namespace on OpenShift, here are examples of the command to copy sas-common backup data or sas-cas-backup data from the SAS Viya platform environment to the local file system:

Here is the example command for sas-common backup data:

```
kubectl -n name-of-namespace exec sas-pod-name -c sas-backup-pv-copy-cleanup-job -- sh -c 'tar cpf - --mode=777 -C /sasviyabackup $(find /sasviyabackup -maxdepth 1 -type d -regex ".*\/backup_id.*" -printf "%P\n")' > backup_id_common.tar
```

Here is the example command for sas-cas backup data:

```
kubectl -n name-of-namespace exec sas-pod-name -c sas-backup-pv-copy-cleanup-job -- sh -c 'tar cpf - --mode=777 -C /cas $(find /cas -maxdepth 1 -type d -regex ".*\/backup_id.*" -printf "%P\n")' > backup_id_cas.tar
```

Note:

- In the command above, BACKUP_ID is always a full backup ID. The parameters on the "tar" command ensure that any related incremental backups are copied along with the full backup that was specified.

- In a SAS Viya platform environment, each backup data directory contains subdirectories with the same name, such as "consul". You must keep each backup data directory in a different location to avoid overwriting existing content. Make a record of that location.

- 2 Here is an example of the commands to copy sas-common backup data and sas-cas backup data from the local file system to the SAS Viya platform environment:

```
cd parent-directory-of-backup-package-directory
```

Here is the example command for sas-common backup data:

```
kubectl -n name-of-namespace exec -i sas-pod-name -c sas-backup-pv-copy-cleanup-job -- tar xf - -C /sasviyabackup < /path/to/localfilesystem/backup_id_common.tar
```

Here is the example command for sas-cas backup data:

```
kubectl -n name-of-namespace exec -i sas-pod-name -c sas-backup-pv-copy-cleanup-job -- tar xf - -C /cas < /path/to/localfilesystem/backup_id_cas.tar
```

Note: In the command above, BACKUP_ID is always a full backup ID. The parameters on the "tar" command ensure that any related incremental backups are copied along with the full backup that was specified.

Note: The time required to copy the backup data contents varies based on the size of the backup directory, the disk I/O rate of the system, and the type of file system that you are using.

- d If you need to compress or decompress the data to the target location, use the tar, lz4 and kubectl exec commands as follows:

Note: See [“List All Executed Backups and Their Status” on page 18](#) for the command to get the backup ID that is needed in the following steps.

- 1 If you are not restoring to an alternate namespace on OpenShift, here is an example of the command to compress sas-common backup data and sas-cas backup data from the SAS Viya platform environment to the local file system:

Here is the example command for sas-common backup data:

```
kubectl -n name-of-namespace exec sas-pod-name -c sas-backup-pv-copy-cleanup-job -- sh -c 'tar cpf - -C /sasviyabackup $(find /sasviyabackup -maxdepth 1 -type d -regex ".*\/backup_id.*" -printf "%P\n") | lz4' > backup_id_common.tar.lz4
```

Here is the example command for sas-cas backup data:

```
kubectl -n name-of-namespace exec sas-pod-name -c sas-backup-pv-copy-cleanup-job -- sh -c 'tar cpf - -C /cas $(find /cas -maxdepth 1 -type d -regex ".*\/backup_id.*" -printf "%P\n") | lz4' > backup_id_cas.tar.lz4
```

If you are restoring to an alternate namespace on OpenShift, you must change the permissions of the target backup folder while copying the sas-common backup data. Here is an example of the command:

Here is the example command for sas-common backup data:

```
kubectl -n name-of-namespace exec sas-pod-name -c sas-backup-pv-copy-cleanup-job
-- sh -c 'tar cpf - --mode=777 -C /sasviyabackup $(find /sasviyabackup -maxdepth
1 -type d -regex ".*backup_id.*" -printf "%P\n") | lz4' >
backup_id_common.tar.lz4
```

Here is the example command for sas-cas backup data:

```
kubectl -n name-of-namespace exec sas-pod-name -c sas-backup-pv-copy-cleanup-job
-- sh -c 'tar cpf - --mode=777 -C /cas $(find /cas -maxdepth 1 -type d -regex ".*back
up_id.*" -printf "%P\n") | lz4' > backup_id_cas.tar.lz4
```

Note:

- In the command above, BACKUP_ID is always a full backup ID. The parameters on the "tar" command ensure that any related incremental backups are copied along with the full backup that was specified.
 - LZ4 is the compression method used on the given backup_id folder. The output of the compression is saved with the .tar.lz4 extension. See [Table 6 on page 81](#) for more information.
 - In a SAS Viya platform environment, multiple backup data directories exist with the same name. You must keep each backup data directory in a different directory and make a record of the location.
-

- 2 Here is an example of the commands to decompress sas-common backup data and sas-cas backup data from the local file system to the SAS Viya platform environment:

Here is the example command for sas-common backup data:

```
kubectl -n name-of-namespace exec -i sas-pod-name -c sas-backup-pv-copy-cleanup-
job -- sh -c 'lz4 -dc - | tar xkf - -C /sasviyabackup/' < backup_id.tar.lz4
```

Here is the example command for sas-cas backup data:

```
kubectl -n name-of-namespace exec -i sas-pod-name -c sas-backup-pv-copy-cleanup-
job -- sh -c 'lz4 -dc - | tar xkf - -C /cas/' < backup_id.tar.lz4
```

Note:

- If you need to perform an integrity check on the compressed LZ4 backup, run the following command before decompressing the backup:

```
kubectl -n name-of-namespace exec -i sas-pod-name -c sas-backup-pv-copy-cleanup-
job -- sh -c 'lz4 -tc' < /path/to/localfilesytem/backup_id.tar.lz4
```

- The time that is required to compress or decompress the backup data contents varies based on the size of the backup directory, the disk I/O rate of the system, and the type of file system that you are using. See "[Compression Benchmarks](#)" on [page 82](#) for more information.
-

- e Remove the sas-backup-pv-copy job from the target environment if required.

```
kubectl -n name-of-namespace delete job -l sas.com/backup-job-type=sas-backup-pv-copy-
cleanup
```

- f Repeat steps c through e to copy sas-cas-backup data from the /cas location in the sas-backup-pv-copy pod.

How to Copy Other Backup Data between the SAS Viya Platform Environment and the Local File System Using tar and kubectl exec

Note: Use these steps to copy data to and from PVCs other than the sas-common-backup-data PVC and sas-cas-backup-data PVC.

- 1 Determine the location of the data that you want to copy from or to.
- 2 Navigate to the machine where kubeconfig is set to run kubectl commands.
- 3 Use the following command to get a list of running pods using a backup PVC from the SAS Viya platform environment:

```
kubectl -n name-of-namespace get pods -l 'sas.com/backup-job-type!=purge-backup,sas.com/backup-job-type!=sas-backup-pv-copy-cleanup' --field-selector=status.phase=Running -o=jsonpath='{range .items[*]}{@.metadata.name}{ "\t"}{@.spec.containers[*].name}{ "\n"}{end}' | grep sas-backup-agent | awk '{print $1}'
```

Here is an example of the list of running pods that might result.

```
sas-cas-server-default-controller
sas-risk-cirrus-core-c9dd8fbc6-2bgg8
```

If you have a CAS backup controller configured, then the resulting pod name might be sas-cas-server-default-backup which can also be used as the name of the running pod in the next step.

Get a list of backup PVCs that are associated with one of the pods above:

```
kubectl -n name-of-namespace get pod sas-pod-name -o=json | jq -r '.spec.volumes[] | select(.persistentVolumeClaim.claimName) | .persistentVolumeClaim.claimName' | grep -i backup
```

For example, use the following command to get the backup PVCs that are associated with the sas-risk-cirrus-core-c9dd8fbc6-2bgg8 pod:

```
kubectl -n name-of-namespace get pod sas-risk-cirrus-core-c9dd8fbc6-2bgg8 -o=json | jq -r '.spec.volumes[] | select(.persistentVolumeClaim.claimName) | .persistentVolumeClaim.claimName' | grep -i backup
```

Here is an example of the output that might result from the command above:

```
sas-risk-cirrus-core-backup-data
```

Identify the pods that you are going to copy to and from. Use these steps to copy data to and from PVCs other than the sas-common-backup-data PVC and sas-cas-backup-data PVC. For example, you might want to copy backup data for the sas-risk-cirrus-core PVC.

The pods and mount paths from backup PVCs that result from the command above might be as follows:

Pod	PVC	Mount path
sas-risk-cirrus-core-xxxx	sas-risk-cirrus-core-backup-data	/sasviyabackup

Note: The list of resulting PVCs above is just an example. Additional PVCs might be present for which the mount path is also `/sasviyabackup`.

- 4 Repeat this step or the next step for each running pod in the list from the previous step. You can copy the data to the target location in uncompressed format (as it is currently) or you can compress the backup data directly to the target location to reduce the space requirements. If you need to compress or decompress the data, see the step [here on page 69](#). If you do not need to compress or decompress the data, perform the following steps:

Copy the backup data to the target location using the tar and kubectl exec command.

Note: See “[List All Executed Backups and Their Status](#)” on page 18 for the command to get the backup ID that is needed in the following steps.

- a Here is an example of the command to copy backup data from the SAS Viya platform environment to the local file system:

```
kubectl -n name-of-namespace exec sas-pod-name -c sas-backup-agent -- sh -c 'tar cpf - -C /sasviyabackup $(find /sasviyabackup -maxdepth 1 -type d -regex ".*\/backup_id.*" -printf "%P\n")' > backup_id.tar
```

Note:

- In the command above, BACKUP_ID is always a full backup ID. The parameters on the “tar” command ensure that any related incremental backups are copied along with the full backup that was specified.
 - In a SAS Viya platform environment, multiple backup data directories exist with the same name. You must keep each backup data directory in a different directory and make a record of the location.
-

- b Here is an example of the commands to copy backup data from the local file system to the SAS Viya platform environment:

```
cd parent-directory-of-backup-package-directory

kubectl -n name-of-namespace exec -i sas-pod-name -c sas-backup-agent -- tar xf - -C /sasviyabackup/ < backup_id.tar
```

Note: In the command above, BACKUP_ID is always a full backup ID. The parameters on the “tar” command ensure that any related incremental backups are copied along with the full backup that was specified.

Note: The time required to copy the backup data contents varies based on the size of the backup directory, the disk I/O rate of the system, and the type of file system that you are using.

- 5 Repeat this step or the previous step for each running pod in the list from step 3. If you need to compress or decompress the data to the target location, use the tar, lz4 and kubectl exec commands as follows:

Note: See [“List All Executed Backups and Their Status” on page 18](#) for the command to get the backup ID that is needed in the following steps.

- a Here is an example of the command to compress backup data from the SAS Viya platform environment to the local file system:

```
kubectl -n name-of-namespace exec sas-pod-name -c sas-backup-agent -- sh -c 'tar cpf - -C /sasviyabackup $(find /sasviyabackup -maxdepth 1 -type d -regex ".*backup_id.*" -printf "%P\n") | lz4' > backup_id.tar.lz4
```

Note:

- In the command above, BACKUP_ID is always a full backup ID. The parameters on the "tar" command ensure that any related incremental backups are copied along with the full backup that was specified.
 - LZ4 is the compression method used on the given backup_id folder. The output of the compression is saved with the .tar.lz4 extension. See [Table 6 on page 81](#) for more information.
 - In a SAS Viya platform environment, multiple backup data directories exist with the same name. You must keep each backup data directory in a different directory and make a record of the location.
-

- b Here is an example of the commands to decompress backup data from the local file system to the SAS Viya platform environment:

```
kubectl -n name-of-namespace exec -i sas-pod-name -c sas-backup-agent -- sh -c 'lz4 -dc - | tar xkf - -C /sasviyabackup/' < backup_id.tar.lz4
```

Note:

- If you need to perform an integrity check on the compressed LZ4 backup, run the following command before decompressing the backup:
- ```
kubectl -n name-of-namespace exec -i sas-pod-name -c sas-backup-agent -- sh -c 'lz4 -tc' < /path/to/localfilesytem/backup_id.tar.lz4
```
- The time that is required to compress or decompress the backup data contents varies based on the size of the backup directory, the disk I/O rate of the system, and the type of file system that you are using. See [“Compression Benchmarks” on page 82](#) for more information.
- 

---

## Copy Backup Data from Azure Environment to Azure Environment

---

**Note:** The following example uses Azure file share. The steps might be different for you depending on your specific deployment or configuration

---

- 1 Determine the location of the PVs used in the source environment as specified in “[Disaster Recovery Prerequisites and Planning on the Source Environment](#)” on page 49.
- 2 Go to the target environment machine where the target persistent volumes are mounted or from where they can be accessed. Mount the source environment path identified in the previous step or go to the actual location if it is already available. If target persistent volumes are not mounted, you can use the same steps as used for mounting the source environment persistent volumes.

If you mount the path locally, make sure that you have the appropriate permissions to mount the file system.

Mount the Azure file share/volume using the standard steps contained in the Azure documentation [here](#).

Here is the command to mount the Azure file share to the local directory.

```
sudo mount -t cifs //STORAGEACCT.file.core.windows.net/shareName /mnt/MyAzureFileShare -o vers=3.0,username=STORAGEACCT,password=STORAGEKEY,dir_mode=0777,file_mode=0777,serverino
```

Here is an example of the command when using the sas-common-backup-data PVC output identified in this [output on page 62](#):

```
sudo mount -t cifs //fbc3e3ecf1289451c913303.file.core.windows.net/kubernetes-dynamic-pvc-23a787c9-9488-4c80-a86e-f9b263d623ce /mnt/MyAzureFileShare -o vers=3.0,username=fbc3e3ecf1289451c913303,password=71He1K2Z5j5Bv8hCb5tb+m4zGgfJFXsBSiUejsV/+5x6eWHrFEUPREzHaQpMKF26IjB59qnKWN7Ex8cPP8954Q==,dir_mode=0777,file_mode=0777,serverino
```

- 3 Copy the required backup ID directory contents using the backup ID identified in [step 1 on page 51](#).

```
cp -rp $(find source_mounted_directory -maxdepth 1 -type d -regex ".*\/backup_id.*") target_environment_pv_path
```

Here is an example of the command when using the sas-common-backup-data PVC output from the source environment in the previous step:

```
cp -rp $(find /mnt/MyAzureFileShare -maxdepth 1 -type d -regex ".*\/backup_id.*") /mnt/TargetAzureFileShare
```

---

**Note:** In the commands above, BACKUP\_ID is always a full backup ID. The parameters on the “tar” command ensure that any related incremental backups are copied along with the full backup that was specified.

---

In the above example, `/mnt/TargetAzureFileShare` is the target environment PV location.

---

**Note:** The time required to copy the backup data contents varies based on the size of the backup directory, the disk I/O rate of the system, and the type of file system that you are using.

---

- 4 Repeat steps 1 through 3 to copy the backup ID directory contents for all PVs.
- 5 Make sure that the directory and file permissions are the same as on the source environment.

## Determine the Backup ID of the Last Successful Backup

To determine the backup ID of the last successful backup, do the following:

### 1 Create the copy pod.

See the "Append the Execute Permissions to Scripts" and "Copy Backup Data to and from Backup Persistent Volume Claims" sections of the README file in one of the following locations for information about creating the copy pod:

| File Format | Path                                                                    |
|-------------|-------------------------------------------------------------------------|
| Markdown    | <code>\$deploy/sas-bases/examples/<br/>restore/scripts/README.md</code> |
| HTML        | <code>\$deploy/sas-bases/docs/<br/>restore_scripts.htm</code>           |

If you have multiple instances of a CAS server, you will see multiple copy pods for each CAS instance. Make a note of the names of the copy pods that you created in the steps above.

This step creates the `sas-backup-pv-copy(s)`. For example, if you have a single CAS Server in your deployment, then you will see that a single pod is created and two PVC(s), `sas-common-backup-data` and `sas-cas-backup-data`, are mounted.

The job creates the following mount paths for the target environment

| Pod                                   | PVC                                 | Mount path                  |
|---------------------------------------|-------------------------------------|-----------------------------|
| <code>sas-backup-pv-copy-xxxxx</code> | <code>sas-common-backup-data</code> | <code>/sasviyabackup</code> |
|                                       | <code>sas-cas-backup-data</code>    | <code>/cas</code>           |

### 2 Get the name of the running pod.

```
kubectl -n name-of-namespace get pods -l sas.com/backup-job-type=sas-backup-pv-copy-cleanup
--field-selector status.phase=Running
```

Substitute the pod name resulting from the above command for *sas-pod-name* in the following steps.

### 3 To get the backup ID of the last successful backup with PostgreSQL, do the following steps:

- a Get the file name that contains the backup ID of the last successful backup with PostgreSQL from the running pod.

```
kubectl -n name-of-namespace exec sas-pod-name -c sas-backup-pv-copy-cleanup-job --
find /sasviyabackup/LastCompletedFullBackups -type f -name '*_postgres'
```

For example, use the following command to get the file name that contains the backup ID of the last successful backup with PostgreSQL from the sas-backup-pv-copy-default-default-260e6ea-vgz6q pod:

```
kubectl -n name-of-namespace exec sas-backup-pv-copy-default-default-260e6ea-vgz6q -c
sas-backup-pv-copy-cleanup-job -- find /sasviyabackup/LastCompletedFullBackups -type f -
name '*_postgres'
```

Here is an example of the file name that might result from the command above:

```
/sasviyabackup/LastCompletedFullBackups/3fa746b9f419285865e9cef3bc4843dc6a57abb2_postgres
```

- b** Use the file name from the previous step as input in the command below to get the backup ID:

```
kubectl -n name-of-namespace exec sas-pod-name -c sas-backup-pv-copy-cleanup-job -- cat
filename | grep -I backupID
```

For example, use the following command to retrieve the backup ID from the file name resulting from the previous step from the sas-backup-pv-copy-default-default-260e6ea-vgz6q pod: :

```
kubectl -n name-of-namespace exec sas-backup-pv-copy-default-default-260e6ea-vgz6q -c
sas-backup-pv-copy-cleanup-job -- cat /sasviyabackup/LastCompletedFullBackups/
3fa746b9f419285865e9cef3bc4843dc6a57abb2_postgres | grep -i backupID
```

Here is an example of the backup ID that might result:

```
"backupID": "20240712-191651F"
```

- 4** To get the backup ID of the last successful backup without PostgreSQL, do the following steps:

- a** Get the file name that contains the backup ID of the last successful backup without PostgreSQL.

```
kubectl -n name-of-namespace exec sas-pod-name -c sas-backup-pv-copy-cleanup-job --
find /sasviyabackup/LastCompletedFullBackups -type f ! -name '*_postgres'
```

For example, use the following command to get the file name that contains the backup ID of the last successful backup without PostgreSQL from the sas-backup-pv-copy-default-default-260e6ea-vgz6q pod:

```
kubectl -n name-of-namespace exec sas-backup-pv-copy-default-default-260e6ea-vgz6q -c
sas-backup-pv-copy-cleanup-job -- find /sasviyabackup/LastCompletedFullBackups -type f !
-name '*_postgres'
```

Here is an example of the file name that might result from the command above:

```
/sasviyabackup/LastCompletedFullBackups/ 3fa746b9f419285865e9cef3bc4843dc6a57abb2
```

- b** Use the file name from the previous step as input in the command below to get the backup ID:

```
kubectl -n name-of-namespace exec sas-pod-name -c sas-backup-pv-copy-cleanup-job -- cat
filename | grep -I backupID
```

For example, use the following command to retrieve the backup ID from the file name resulting from the previous step from the sas-backup-pv-copy-default-default-260e6ea-vgz6q pod: :

```
kubectl -n name-of-namespace exec sas-backup-pv-copy-default-default-260e6ea-vgz6q -c
sas-backup-pv-copy-cleanup-job -- cat /sasviyabackup/LastCompletedFullBackups/
3fa746b9f419285865e9cef3bc4843dc6a57abb2 | grep -i backupID
```

Here is an example of the backup ID that might result:

```
"backupID": "20240712-181661F"
```

- 5 Alternatively, you can list all of the successful backups by running the following command:

```
kubect1 -n name-of-namespace exec sas-pod-name -c sas-backup-pv-copy-cleanup-job -- sh -c
"cd /sasviyabackup ; grep -r -n -h --include 'status.json' -A 3 -i 'BackupID' | grep -A 4
'2:[[:space:]]' | grep -w -B 2 -A 2 -i 'completed'"
```

For example, use the following command to list all of the successful backups from the sas-backup-pv-copy-default-default-260e6ea-vgz6q pod:

```
kubect1 -n name-of-namespace exec sas-backup-pv-copy-default-default-260e6ea-vgz6q -c sas-
backup-pv-copy-cleanup-job -- sh -c "cd /sasviyabackup ; grep -r -n -h --include
'status.json' -A 3 -i 'BackupID' | grep -A 4 '2:[[:space:]]' | grep -w -B 2 -A 2 -i
'completed'"
```

Here is an example of the output from the command above:

```
2: "BackupID": "20240712-061344F",
3: "LastCompletedFullBackupID": "20240712-061344F",
4- "Status": "Completed",
5- "StartTime": "2024-07-12T06:13:44.578273662Z",
6- "EndTime": "2024-07-12T06:15:54.615447714Z",
--
2: "BackupID": "20240712-070841F",
3: "LastCompletedFullBackupID": "20240712-070841F",
4- "Status": "Completed",
5- "StartTime": "2024-07-12T07:08:41.692609738Z",
6- "EndTime": "2024-07-12T07:10:51.739766932Z",
--
2: "BackupID": "20240712-200310F",
3: "LastCompletedFullBackupID": "20240712-200310F",
4- "Status": "Completed",
5- "StartTime": "2024-07-12T20:03:10.762290795Z",
6- "EndTime": "2024-07-12T20:05:20.802512616Z",
```

**Note:** Each backup in the output above is separated by a line containing " -- ". Select the backup ID of the backup with the most recent timestamp from the above list.

- 6 Make a note of the backup ID and save it in a file in a known location.

---

## Set the INCLUDE\_POSTGRES configMap Parameter

To exclude all PostgreSQL servers that are registered with SAS Viya platform from the default backup, add the `INCLUDE_POSTGRES` configMap parameter to the `sas-backup-job-parameters` configMap with a value of `'false'`. If the variable is set to `'false'`, then no PostgreSQL servers are backed up by the SAS backup job.

---

**Note:** Make sure that you understand the steps that are required to create and retain backups for your deployment of PostgreSQL before making this change. To coordinate their contents, the PostgreSQL backup should be taken around the same time that the SAS Viya backup is created.

---

To modify the value of the **INCLUDE\_POSTGRES** configMap parameter, follow the instructions below:

- 1 Identify the **sas-backup-job-parameters** configMap that needs to be modified.

```
kubectl -n name-of-namespace describe cronjob sas-scheduled-backup-job | grep -i sas-backup-
job-parameters | awk '{print $1}'|head -n 1
```

- 2 You can add the **INCLUDE\_POSTGRES** configMap parameter in one of the following ways:

- Edit the base kustomization.yaml file directly, followed by a `kustomize build` and `kubectl apply` to update the charts. This is a more permanent method in that the changes persist during subsequent applications of the manifest.

For instructions about how to set the **INCLUDE\_POSTGRES** configMap parameter by editing the kustomization.yaml file, see the "Include or Exclude All Registered PostgreSQL Servers from Backup" section of the README located in one of the following files:

| File Format | Path                                                                                                                                    |
|-------------|-----------------------------------------------------------------------------------------------------------------------------------------|
| Markdown    | <code>\$deploy/sas-bases/examples/<br/>backup/configure/README.md</code>                                                                |
| HTML        | <code>\$deploy/sas-bases/docs/<br/>configuration_settings_for_backup<br/>_using_the_sas_viya_backup_and_re<br/>store_utility.htm</code> |

- Issue a `kubectl patch` command that includes the **INCLUDE\_POSTGRES** in the list of parameters. This is more of a temporary method in that it can be overwritten by the settings in the manifest files when they are next applied, or by another patch. To use this method, continue with the instructions below:

Here is the `kubectl patch` command to set the **INCLUDE\_POSTGRES** configMap parameters:

```
kubectl patch cm sas-backup-job-parameters-name -n name-of-namespace --type json -p
'[{"op": "replace", "path": "/data/INCLUDE_POSTGRES", "value":"false"}]'
```

---

## Backup Progress

---

**Note:** Neither the validation nor the progress feature work when the option to explicitly exclude schemas is set in the PostgreSQL database.

---

The backup progress feature provides real-time information about the status of an ongoing backup operation. Specifically, it provides users with the estimated time to complete the overall backup.

This feature is enabled by default. At this time, the backup progress is not calculated for incremental backups and blob storage.

The estimated time is calculated based on the overall time to complete the backup of the datasources, PostgreSQL, and the file system. The estimated time is displayed in the **BACKUP-REMAINING-TIME** column that results from the `kubectl get jobs` command. You must include the `sas.com/backup-remaining-time` label on the command to display the estimated time. Here is an example of the command and output:

```
kubectl get jobs -l "sas.com/backup-job-type in (scheduled-backup)" -L "sas.com/backup-remaining-time"
--sort-by=.status.startTime
```

| NAME          | COMPLETIONS | DURATION | AGE    | BACKUP-REMAINING-TIME |
|---------------|-------------|----------|--------|-----------------------|
| backupjobtest | 1/1         | 18m24s   | 18m24s | 1h04m05s              |

The **BACKUP-REMAINING-TIME** column contains the estimated time to complete the backup. The estimated time in this example is 1h04m05s where 1 hour, 04 minutes, and 05 seconds is the remaining time for the backup.

- The calculation of the remaining backup time is an approximation meaning that there might be a deviation of a few minutes based on the size of the data that is being backed up. Also, deviations can be observed on new deployments for which there are no prior backups.
- PostgreSQL compresses data by default, and any changes to its default compression levels might affect the accuracy of estimates.
- When the calculation of estimated time starts:
  - the value **"initializing-3min"** is displayed under the **BACKUP-REMAINING-TIME** column. This is the initialization time.
  - the value in the **BACKUP-REMAINING-TIME** column is updated with an estimated completion time, based on the polling frequency.
- After the backup completes, the value in the **BACKUP-REMAINING-TIME** column is updated to **SHUTTING-DOWN** until the job terminates, regardless of whether the backup was successful or failed. When the job terminates, the label is updated to **ENDED**.

**Note:** If the backups are slow due to environmental factors such as I/O or network performance, the backup progress updates might display **Estimation-Timed-Out** in the **BACKUP-REMAINING-TIME** column. At this point, the backup estimation updates are disabled, but the backup operation will continue.

If you want to disable the backup progress feature, see the "Backup Progress" section of the README file in one of the following locations:

| File Format | Path                                                                                                                     |
|-------------|--------------------------------------------------------------------------------------------------------------------------|
| Markdown    | <code>\$deploy/sas-bases/examples/backup/configure/README.md</code>                                                      |
| HTML        | <code>\$deploy/sas-bases/docs/configuration_settings_for_backup_using_the_sas_viya_backup_and_restore_utility.htm</code> |

# Adjust CPU and Memory Resources for Restore Job and PostgreSQL

To adjust the CPU and memory resources for a restore based on the backup size, do the following:

- To increase the restore CronJob resources, do the following:

- 1 Determine the existing resource specifications. Make a note of the values.

```
kubectl get -n name-of-namespace cronjob sas-restore-job -o
jsonpath='{.spec.jobTemplate.spec.template.spec.containers[0].resources.limits}'
```

- 2 See the table [here on page 78](#) to determine the adequate CPU and memory allocations. Increase the resources for the restore CronJob.

**Note:** The command below uses sample values for the number of CPUs and amount of memory.

```
kubectl patch cronjob sas-restore-job -n name-of-namespace --type='json' -p='[{"op":
"replace", "path": "/spec/jobTemplate/spec/template/spec/containers/0/resources/limits/cpu",
"value": "2"}, {"op": "replace", "path": "/spec/jobTemplate/spec/template/spec/containers/0/
resources/limits/memory", "value": "6Gi"}]'
```

- To increase the PostgreSQL pod resources, do the following:

- 1 Determine the existing resource specifications. Make a note of the values.

Here is the command for sas-crunchy-platform-postgres:

```
kubectl get -n name-of-namespace postgrescluster sas-crunchy-platform-postgres -o
jsonpath='{.spec.instances[0].resources.limits}'
```

Here is the command for sas-crunchy-cds-postgres:

```
kubectl get -n name-of-namespace postgrescluster sas-crunchy-cds-postgres -o
jsonpath='{.spec.instances[0].resources.limits}'
```

- 2 See the table [here on page 78](#) to determine the adequate CPU and memory allocations. Increase the resources for the PostgreSQL pods.

**Note:** The commands below use sample values for the number of CPUs and amount of memory.

Here is the command for sas-crunchy-platform-postgres:

```
kubectl patch -n name-of-namespace postgrescluster sas-crunchy-platform-postgres --
type='json' -p='[{"op": "replace", "path": "/spec/instances/0/resources/limits/
cpu", "value": "8"}, {"op": "replace", "path": "/spec/instances/0/resources/limits/
memory", "value": "16Gi"}]'
```

Here is the command for sas-crunchy-cds-postgres:

```
kubect1 patch -n name-of-namespace postgrescluster sas-crunchy-cds-postgres --type='json' -
p='[{"op":"replace","path":"/spec/instances/0/resources/limits/cpu","value":"8"},
{"op":"replace","path":"/spec/instances/0/resources/limits/memory","value":"16Gi"}]'
```

**Note:** After the resources are updated, wait until the pods come up before proceeding with the restore.

When the restore process is complete, revert to the original values of CPU and memory that were noted in the first step.

## Resource Requirements for Restore

### Resource Requirements for the Restore Job

To restore SAS Viya data successfully using the restore CronJob, the following specific CPU and memory allocations are recommended based on the size of the backup:

**Table 2** *Recommended CPU and Memory Allocations*

| Backup Size | CPU (vCPU) | Memory |
|-------------|------------|--------|
| <= 1 TB     | 2          | 4 GiB  |
| > 1 TB      | 2          | 6 GiB  |

### Resource Requirements for the PostgreSQL Pod during the Restore Process

For the PostgreSQL pod running in a non multi-tenant environment, the following resource allocations are recommended based on the size of the backup:

**Table 3** *Recommended Resource Allocations*

| Backup Size            | CPU (vCPU) | Memory |
|------------------------|------------|--------|
| <= 250 GB              | 8          | 8 GiB  |
| > 250 GB and <= 500 GB | 8          | 10 GiB |
| > 500 GB and <= 1 TB   | 8          | 12 GiB |
| > 1 TB                 | 8          | 16 GiB |

## Sizing Recommendations for Restore

The restore of a PostgreSQL database requires more space than the size of the database in the backup. SAS recommends that you allot 2.5 times the size of the source PostgreSQL database in the space that you are restoring to. For example, if your source PostgreSQL database is 50GB then you should allot 125GB of space for the restore.

The status.json file for each backup contains a sourceSize field for each instance of a PostgreSQL database in the source SAS Viya platform environment. The sourceSize field contains the size of the source PostgreSQL database. For example, if you are restoring a SAS Viya platform deployment that contains only the default instance of the PostgreSQL database, the path location of the status.json file and sourceSize field contents are as follows:

**Table 4** Path Location of status.json File and sourceSize Field Contents

| SAS Viya Platform Environment | Path Location of status.json File                      | Contents of sourceSize field        |
|-------------------------------|--------------------------------------------------------|-------------------------------------|
| non-multi-tenant environment  | <code>backupID/__default__/postgres/status.json</code> | size of the SharedServices database |

Also, you can view the sourceSize for all of the instances of a PostgreSQL database in these path locations:

**Table 5** Path Location for sourceSize of All PostgreSQL Databases

| head                         | head                                          |
|------------------------------|-----------------------------------------------|
| non-multi-tenant environment | <code>backupID/__default__/status.json</code> |

## Comparison of Backup and Restore Approaches

See the following table for a comparison of the different approaches to backup and restore in the SAS Viya platform:

| Category                        | SAS Viya Backup and Restore of All Data | Provider-Specific Processes for PostgreSQL and Full SAS Viya Backup and Restore of Remaining Data | Provider-Specific Processes for PostgreSQL and Incremental SAS Viya Backup and Restore of Remaining Data |
|---------------------------------|-----------------------------------------|---------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------|
| Tool to Back Up and Restore SAS | SAS Viya Backup and Restore             | SAS Viya Backup and Restore                                                                       | SAS Viya Backup and Restore                                                                              |

| Category                                       | SAS Viya Backup and Restore of All Data                                                                                                                            | Provider-Specific Processes for PostgreSQL and Full SAS Viya Backup and Restore of Remaining Data                                                                 |                          | Provider-Specific Processes for PostgreSQL and Incremental SAS Viya Backup and Restore of Remaining Data                                                                                       |                            |
|------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------|
| Viya platform Content                          |                                                                                                                                                                    |                                                                                                                                                                   |                          |                                                                                                                                                                                                |                            |
| Tool to Back Up and Restore PostgreSQL Content |                                                                                                                                                                    | PostgreSQL-specific provider                                                                                                                                      |                          | PostgreSQL-specific provider                                                                                                                                                                   |                            |
| SAS Viya Backup Setting for PostgreSQL         | INCLUDE_POSTGRES=true                                                                                                                                              | INCLUDE_POSTGRES=false                                                                                                                                            |                          | INCLUDE_POSTGRES=false                                                                                                                                                                         |                            |
| Prerequisites                                  |                                                                                                                                                                    | <p>PostgreSQL provider supports Point-In-Time-Recovery (PITR).</p> <p>Backup retention periods between PostgreSQL and the SAS Viya platform are synchronized.</p> |                          | <p>PostgreSQL provider supports Point-In-Time-Recovery (PITR).</p> <p>Backup retention periods between PostgreSQL and the SAS Viya platform are synchronized.</p>                              |                            |
| Advantages                                     | Ease of use                                                                                                                                                        | <p>Faster restore speeds with large PostgreSQL databases.</p> <p>Less disk space required by backup PVs.</p>                                                      |                          | <p>Faster restore speeds with large PostgreSQL databases.</p> <p>Less disk space required by backup PVs.</p> <p>Saves time in performing backups.</p> <p>Can be scheduled more frequently.</p> |                            |
| Disadvantages                                  | <p>Might result in slower restore speeds if PostgreSQL databases or CAS data volume are large.</p> <p>Backup package can consume a large amount of disk space.</p> | <p>Harder to manage</p> <p>Might result in higher cost of backup storage on cloud-provider</p>                                                                    |                          | <p>Harder to manage</p> <p>Might result in higher cost of backup storage on cloud-provider</p>                                                                                                 |                            |
|                                                |                                                                                                                                                                    | External PG                                                                                                                                                       | Internal PG <sup>1</sup> | External PG                                                                                                                                                                                    | Internal PG <sup>1,2</sup> |
| In-place restore                               | ✓                                                                                                                                                                  | ✓                                                                                                                                                                 | ✓                        | ✓                                                                                                                                                                                              | ✓                          |

| Category                                         | SAS Viya Backup and Restore of All Data | Provider-Specific Processes for PostgreSQL and Full SAS Viya Backup and Restore of Remaining Data |  | Provider-Specific Processes for PostgreSQL and Incremental SAS Viya Backup and Restore of Remaining Data |  |
|--------------------------------------------------|-----------------------------------------|---------------------------------------------------------------------------------------------------|--|----------------------------------------------------------------------------------------------------------|--|
|                                                  |                                         |                                                                                                   |  |                                                                                                          |  |
| Disaster Recovery                                | ✓                                       | ✓                                                                                                 |  | ✓                                                                                                        |  |
| SAS Viya platform to SAS Viya platform migration | ✓                                       | ✓                                                                                                 |  | ✓                                                                                                        |  |

- 1 You can use the `sas-scheduled-backup-all-sources` CronJob to easily generate a backup that includes PostgreSQL even if the `INCLUDE_POSTGRES` configMap parameter is set to `false`. Backups that use the SAS Viya platform processes for all data are required for disaster recovery and SAS Viya platform to SAS Viya platform migrations. See [“Enable Suspended CronJob to Back Up All Content with the SAS Viya Platform \(Including PostgreSQL\)”](#) on page 21.
- 2 When you restore from an incremental backup, the most recent full backup that it is associated with is also restored.

## LZ4 Compression Details

If you decide to compress or decompress data in the backup ID folder that you are copying between the SAS Viya platform environment and the local file system, LZ4 is the compression method that is used.

## Compression Options

### ■ Compression levels

You can specify compression levels from 1 to 9 with 1 using the least amount of compression and 9 using the maximum amount of compression.

**Table 6** LZ4 Compression Levels

| Compression Level | Amount of Compression                  | Speed                            |
|-------------------|----------------------------------------|----------------------------------|
| 1                 | Least compression                      | Fastest speed                    |
| 2 - 8             | Increasing compression with each level | Decreasing speed with each level |
| 9                 | Maximum compression                    | Slowest speed                    |

Here is example of LZ4 compression using level 1:

```
kubect1 -n name-of-namespace exec sas-pod-name -c sas-backup-agent -- sh -c 'tar cpf - -C /
sasviyabackup $(find /sasviyabackup -maxdepth 1 -type d -regex ".*backup_id.*" -printf "%P
\n") | lz4 -l' > backup_id.tar.lz4
```

#### ■ Checksums

The **-BX** option can be added to LZ4 compression to add block level checksums. Here is an example:

```
kubect1 -n name-of-namespace exec sas-pod-name -c sas-backup-agent -- sh -c 'tar cpf - -C /
sasviyabackup $(find /sasviyabackup -maxdepth 1 -type d -regex ".*backup_id.*" -printf "%P
\n") | lz4 -BX' > backup_id.tar.lz4
```

## Compression Benchmarks

**Table 7** LZ4 Compression Benchmarks

| Compression Type | Compression Time (seconds) <sup>1</sup> | Decompression Time (seconds) <sup>1</sup> | Uncompressed Size (GB) | Compressed Size (GB) | Approximate Ratio |
|------------------|-----------------------------------------|-------------------------------------------|------------------------|----------------------|-------------------|
| None             | 150                                     | 200                                       | 13                     | 13                   | 1:1               |
| LZ4(Level 1)     | 420                                     | 400                                       | 13                     | 7.7                  | 1.6:1             |
| LZ4(Level 1)     | 1556                                    | 1560                                      | 40                     | 20                   | 2:1               |
| LZ4(Level 1)     | 5396                                    | 3568                                      | 121                    | 61                   | 1.98:1            |

<sup>1</sup> You might improve the compression or decompression time by allocating more CPU resources to the sas-backup-agent container that is used while compressing data. Try increasing the Kubernetes limit that is allocated to the sas-backup-agent container for faster compression and decompression times

For information about increasing the CPU resources in the backup agent container in the CAS controller pod or in the backup copy job, see one of the following files:

| File Format | Path                                                                                                                     |
|-------------|--------------------------------------------------------------------------------------------------------------------------|
| Markdown    | <code>\$deploy/sas-bases/examples/backup/configure/README.md</code>                                                      |
| HTML        | <code>\$deploy/sas-bases/docs/configuration_settings_for_backup_using_the_sas_viya_backup_and_restore_utility.htm</code> |

---

# Backup and Restore: Troubleshooting

---

**Note:** During deployment, you created a deployment directory that SAS recommended that you name `deploy` or an alternative name that is meaningful to you. The directory is referred to as `$deploy` in this guide. Replace `$deploy` with the directory name that you choose

---



---

## General Troubleshooting Strategies

A general troubleshooting topic is available for the SAS Viya platform. See [Troubleshooting Strategies](#) for general information about how to approach issues.

---

## Logs

### Overview

Logs for a data source backup are stored with the backup data in the same folder. To view logs in the backup PV for the data source, go to the file system of the backup PV, and navigate to the *backup ID/restore/restore ID* folder where the logs are stored.

### Find the Pod Name for a Backup Job

List all of the backups to find the name of the backup job:

```
kubectl get jobs -n name-of-namespace -l "sas.com/backup-job-type in (backup,scheduled-backup,scheduled-backup-incremental)" -L "sas.com/sas-backup-id,sas.com/sas-backup-job-status,sas.com/sas-backup-persistence-status"
```

To find the pod name for the backup job, substituting the backup job name from the last step:

```
kubectl get pods -n name-of-namespace | grep backup-job-name
```

### View the Backup Job Log for the Pod

To view the backup job log for the pod:

```
kubectl logs backup-job-pod-name -n name-of-namespace -c sas-backup-job
```

## View a Backup Job Log by Backup ID

To view a backup job log by backup ID:

```
kubectl logs -l backup-ID -n name-of-namespace
```

## View a Backup Agent's Log

To view a backup agent's log:

```
kubectl logs data-source-pod-name -n name-of-namespace -c sas-backup-agent
```

The data source pod name can be the name of any pod where the backup agent is added as a side car. For example, it might be the backup-job-pod-name or the name of any CAS controller pod.

---

## Backup Fails for fileSystem Source When Using Azure Premium Storage

### Problem

The following error results when backing up the fileSystem source and you are using Azure premium storage:

```
08:54:15 {"level":"error","version":1,"source":"sas-backupjob",
"messageKey":"backupjob-log-icu.backup.occurred.error.log","messageParameters":
{"err":"The job failed for data sources: (DataSource: sas-planning SourceType:
fileSystem), (DataSource: cas-shared-default SourceType: fileSystem)"},"properties":
{"logger":"backupjob","caller":"backupjob/main.go:231"},"timeStamp":
"2021-09-06T12:50:49.501779+00:00","message":"An error occurred. Details:
The job failed for data sources: (DataSource: sas-planning SourceType: fileSystem),
(DataSource: cas-shared-default SourceType: fileSystem)"} }
```

### Explanation

The settings for Azure premium storage might not be set appropriately.

### Resolution

Ensure that the Azure premium storage settings are correct. See [“File System Permissions” in System Requirements for the SAS Viya Platform](#) for more information.

---

## Backup Fails When Services Are Not Completely Started

### Problem

The following errors result after performing a backup:

- You might see the following error in the status.json file located in the backup directory here:  
*backup PV/backup ID/status.json*:

```
{
```

```

 "ServiceID": "sas-crunchy-platform-postgres",
 "Status": "Failed",
 "StatusCode": 3,
 "Size": "",
 "Logs": [
 "pg_dump --format directory --host sas-crunchy-data-postgres
--lock-wait-timeout 300000
--no-blobs --verbose -U dbmsowner -d SharedServices -f
/sasviyabackup/2022-02-22T07_09_07_607_0700/___default___/postgres/SharedServices
_pg_dump.dmp -p 5432 ",
 "exit status 1"
],
 "StartTime": "2022-02-22T07:09:19.066361261Z",
 "EndTime": "2022-02-22T07:09:28.185899152Z"
 }
}

```

- You might see content such as the following in the PostgreSQL login the backup directory here: **backup PV/backup ID/\_\_\_default\_\_\_/postgres/SharedServices.log**

```

pg_dump: last built-in OID is 16383
pg_dump: reading extensions
pg_dump: identifying extension members
pg_dump: reading schemas
pg_dump: reading user-defined tables
pg_dump: error: query failed: ERROR: relation "risketl.databaschangelock"
does not exist
pg_dump: error: query was: LOCK TABLE risketl.databaschangelock
IN ACCESS SHARE MODE

```

### Explanation

The services might not be started or might still be in the process of starting.

### Resolution

Check the backup logs to ensure that all of the services are completely started before performing a backup. See [“Overview” on page 83](#) for information about the backup logs.

---

## Backup Fails If Available Disk Space Cannot Be Calculated

### Problem

An error such as the following results when performing a backup:

```
The Filesystem source size is 17Gi whereas remaining space on sas-cas-backup-data is -2Gi.
```

or

```
The postgres source size is 356.9M whereas remaining space on sas-common-backup-data
is -9223372036854775808.
```

### Explanation

Resource validation is performed before performing a backup. The backup job fails if the disk space cannot be calculated.

**Resolution**

Disable the resource validation. See the "Disable Resource Validation" section in one of the following files for more information:

| File Format | Path                                                                                                   |
|-------------|--------------------------------------------------------------------------------------------------------|
| Markdown    | <code>\$deploy/sas-bases/examples/<br/>backup/configure/README.md</code>                               |
| HTML        | <code>\$deploy/sas-bases/docs/<br/>configuration_settings_for_sas_vi<br/>ya_platform_backup.htm</code> |

---

## Restore Fails If Resource Validation Fails

**Problem**

The restore job results in resource validation errors. The error indicates that the available memory or CPU resources are less than the minimum required amount.

**Explanation**

Resource validation (such as CPU and memory) is performed before performing a restore. The restore job fails if there are not adequate resources.

**Resolution**

- Adjust the CPU and memory resources. See ["Adjust CPU and Memory Resources for Restore Job and PostgreSQL" on page 77](#).
- Disable the resource validation. See the "Disable Resource Validation" section in one of the following files for more information:

| File Format | Path                                                                                                    |
|-------------|---------------------------------------------------------------------------------------------------------|
| Markdown    | <code>\$deploy/sas-bases/examples/<br/>restore/configure/README.md</code>                               |
| HTML        | <code>\$deploy/sas-bases/docs/<br/>configuration_settings_for_sas_vi<br/>ya_platform_restore.htm</code> |

# Purge Job Fails Requiring Manual Deletion of Backup

## Problem

Failed backup needs to be purged immediately.

## Explanation

Backups are purged based on a set of conditions described in [“Purge Backups” on page 46](#).

Failed backups are included in the purge cycle once out of the retention policy. They are not purged immediately.

## Resolution

Manually delete the backup. This approach can also be used as a workaround while troubleshooting purge issues

To manually purge backups from the common backup data on the sas-common-backup-data PVC and the CAS backup data on the sas-cas-backup-data PVC, follow the steps below:

- 1 Create the copy pod.

See the “Append the Execute Permissions to Scripts” and “Copy Backup Data to and from Backup Persistent Volume Claims” sections of the README file in one of the following locations for information about creating the copy pod:

| File Format | Path                                                                    |
|-------------|-------------------------------------------------------------------------|
| Markdown    | <code>\$deploy/sas-bases/examples/<br/>restore/scripts/README.md</code> |
| HTML        | <code>\$deploy/sas-bases/docs/<br/>restore_scripts.htm</code>           |

If you have multiple instances of a CAS server, you will see multiple copy pods for each CAS instance. Make a note of the names of the copy pods that you created in the steps above.

- 2 Delete the content from the sas-common-backup-data PVC.

```
kubectl -n name-of-namespace exec sas-copy-pod-name -c sas-backup-pv-copy-cleanup-job -- sh
-c 'cd /sasviyabackup/ ; rm -rf backupID1 backupID2'
```

**Note:** In the command above, *backupID1* and *backupID2* are the names of the backups whose contents you need to delete from the sas-common-backup-data PVC.

- 3 Delete the content from the sas-cas-backup-data PVC. If there are more than one CAS controller in the SAS Viya platform deployment, repeat this step for each additional CAS backup data PVC.

```
kubectl -n name-of-namespace exec sas-copy-pod-name -c sas-backup-pv-copy-cleanup-job -- sh
-c 'cd /cas/ ; rm -rf backupID1 backupID2'
```

---

**Note:** In the command above, *backupID1* and *backupID2* are the names of the backups whose contents you need to delete from the sas-cas-backup-data PVC.

---

**IMPORTANT** Backups that are removed with the `rm -r backupID1` command are permanently removed, and are not recoverable.

## Purge Job Fails Due to sas-backup-agent Time-Out

### Problem

A purge job fails.

### Explanation

Backups are purged based on a set of conditions described in [“Purge Backups” on page 46](#).

A purge job can fail if the sas-backup-agent does not receive a response within the time-out limit.

### Resolution

Set a different time-out limit as follows:

- 1 Determine the sas-backup-agent-parameters name that is associated with the sas-backup-purge-job CronJob.

```
kubectl get cronjobs -n name-of-namespace sas-backup-purge-job -o yaml | grep sas-backup-agent-parameters | tail -1 | awk '{print $2}'
```

- 2 Set the BACKUP\_JOB\_ACK\_TIMEOUT configMap parameter.

```
kubectl patch cm sas-backup-agent-parameters -n name-of-namespace -p '{"data" : {"BACKUP_JOB_ACK_TIMEOUT" : "60"}}'
```

---

**Note:** The unit of time specified is in minutes. If 60 minutes is not sufficient, it is safe to increase it.

---

## Backup and Restore Processes are Slow

The performance of your backup and restore can be related to database size. Job instances that do not expire are a known problem. See [Job Objects and Database Growth](#) for more information.

# The sas-backup-pv-copy-cleanup Script Fails to Clean Up the CAS PVs during a Restore on OpenShift

## Problem

The following error results when cleaning up the file system PVs and you are using an OpenShift cluster:

```
SAS_BACKUP_AGENT_START_MODE is set to REMOVE
cleaning up the cas pv's
rsync: chgrp "/cas/data/." failed: Operation not permitted (1)
rsync: delete_file: rmdir(caslibs/vamodels) failed: Permission denied (13)
rsync: delete_file: unlink(caslibs/sysData/SASVIYATYPES.sashdat) failed:
Permission denied (13)
rsync: delete_file: unlink(caslibs/sysData/SASVIYAINVENTORY4_PRIOR.sashdat) failed:
Permission denied (13)
rsync: delete_file: unlink(caslibs/sysData/SASVIYAINVENTORY4_CASSVRDETAILS.sashdat)
failed: Permission denied (13)
rsync: delete_file: unlink(caslibs/sysData/SASVIYAINVENTORY4.sashdat) failed:
Permission denied (13)
rsync: delete_file: unlink(caslibs/sysData/SASVIYAINV4_COMPARISON.sashdat) failed:
Permission denied (13)
rsync: delete_file: unlink(caslibs/sysData/IMPORTEDITEMS.sashdat) failed:
Permission denied (13)
rsync: delete_file: unlink(caslibs/sysData/AUDIT_ACTIVITIES_URIS.sashdat) failed:
Permission denied (13)
rsync: delete_file: unlink(caslibs/sysData/AUDIT_ACTIVITIES_PROPERTIES.sashdat) failed:
Permission denied (13)
rsync: delete_file: unlink(caslibs/sysData/AUDIT_ACTIVITIES.sashdat) failed:
Permission denied (13)
rsync: delete_file: unlink(caslibs/sysData/AUDIT.sashdat) failed: Permission denied (13)
rsync: delete_file: rmdir(caslibs/sysData) failed: Permission denied (13)cannot delete
non-empty directory: caslibs/sysData
cannot delete non-empty directory: caslibs/search
cannot delete non-empty directory: caslibs/samples
rsync error: some files/attrs were not transferred (see previous errors) (code 23)
at main.c(1189) [sender=3.1.3]
```

## Explanation

The settings for fsGroup might not be set appropriately either for CAS or for the CronJob sas-backup-pv-copy-cleanup-job.

## Resolution

Add the fsGroup setting to the CronJob sas-backup-pv-copy-cleanup-job as follows:

- 1 The `$deploy/sas-bases/examples/restore/scripts/openshift` directory contains a file to grant security context constraints (SCCs) for the sas-backup-pv-copy-cleanup-job pod on an OpenShift cluster.

Use one of the following commands to apply the SCCs:

- To use kubectl, use this command:

```
kubectl apply -f sas-backup-pv-copy-cleanup-job-scc-fsgroup.yaml
```

- To use the OpenShift command-line interface (CLI), use this command:

```
oc create -f sas-backup-pv-copy-cleanup-job-scc-fsgroup.yaml
```

- 2 Use the following command to link the SCCs to the appropriate Kubernetes service account:

```
oc -n name-of-namespace adm policy add-scc-to-user sas-backup-pv-copy-cleanup-job -z sas-viya-backuprunner
```

- 3 Use the following command to add the fsGroup setting to the CronJob sas-backup-pv-copy-cleanup-job:

```
kubectl patch cronjob sas-backup-pv-copy-cleanup-job -n name-of-namespace --type json -p ' [{"op": "replace", "path": "/spec/jobTemplate/spec/template/spec/securityContext/fsGroup", "value": FSGROUP_VALUE}] '
```

Here is an example:

```
kubectl patch cronjob sas-backup-pv-copy-cleanup-job -n viyans --type json -p ' [{"op": "replace", "path": "/spec/jobTemplate/spec/template/spec/securityContext/fsGroup", "value": 1001}] '
```

- 4 Rerun the sas-backup-pv-copy-cleanup script.

Here is an example:

```
./sas-backup-pv-copy-cleanup.sh viyans remove "default"
```

---

## CAS Server Data Fails to Backup

### Problem

A backup of CAS server data fails. The backup status shows "Failed." The following error results when backing up CAS server data when CAS servers are not running:

```
{ "level": "error", "version": 1, "source": "sas-backupjob", "messageKey": "backupjob-log-icu.ack.timeout.for.source.error.log", "messageParameters": { "actionType": "backup", "sourceType": "cas-shared-default:cas" }, "properties": { "logger": "backupjob/handler", "caller": "handler/backup.go:683", "timeStamp": "2025-09-11T02:28:16.956733+00:00", "message": "Cannot perform the \"backup\" action for the \"cas-shared-default:cas\" data source as the acknowledgement not received from the backup agent." } }
```

```
{ "level": "error", "version": 1, "source": "sas-backupjob", "messageKey": "backupjob-log-icu.backup.occurred.error.log", "messageParameters": { "err": "The job failed for data sources: (DataSource: cas-shared-default SourceType: cas), (DataSource: cas-shared-default SourceType: fileSystem)" }, "properties": { "logger": "backupjob", "caller": "backupjob/main.go:615", "timeStamp": "2025-09-11T02:28:16.963753+00:00", "message": "An error occurred. Details: The job failed for data sources: (DataSource: cas-shared-default SourceType: cas), (DataSource: cas-shared-default SourceType: fileSystem)" } }
```

```
{ "level": "error", "version": 1, "source": "sas-backupjob", "messageKey": "backupjob-log-icu.backup.backup.status.info.log", "messageParameters": { "jobStatus": "Failed" }, "properties": { }
```

```
{ "logger": "backupjob", "caller": "backupjob/main.go:625" },
 "timeStamp": "2025-09-11T02:28:16.963930+00:00",
 "message": "The backup job execution finished with status \"Failed\"." }
```

### Explanation

CAS Servers must be running to back up CAS server data. CAS server data includes CAS access controls, global caslibs, and CAS Data.

### Resolution

Ensure that CAS Servers are running. See [SAS Cloud Analytic Services](#) for more information about managing CAS servers.

---

## Restore Job or Migration Job Fails Due to Active Connection Termination in External PostgreSQL

### Problem

You receive an error that a restore job or migration job failed due to connection termination issues in an external PostgreSQL setup.

### Explanation

The restore job or migration job failed because of one of the following reasons:

- The identity that connects to the database is not able to terminate autovacuum workers due to insufficient permissions.
- The identity that connects to the database has insufficient permissions.
- The restore job or migration job failed due to active sessions.

### Resolution

To resolve the error, perform the following steps:

---

**Note:** You must perform the following steps for all databases that report a connection termination failure in the logs.

---

- 1 Check the migration pod or restore pod logs. Run the following command to identify the error:

```
kubectl logs <podname> -n <namespace> | grep -F "Error while terminating connections"
```

- 2 Review the error logs to determine the type of error.

An error log related to autovacuum might look like this:

```
ERROR: permission denied to terminate process
DETAIL: Only roles with privileges of the "pg_signal_autovacuum_worker" role may
terminate autovacuum workers.
```

Other error logs typically mention that permission has been denied or that active sessions caused a migration failure or restore failure. The error log might look like this:

```
{ "level": "error", "version": 1, "source": "sas-restorejob", "messageKey": "backupjob-log-
icu.restore.job.failed.err.log", "messageParameters": { "error": "For ServiceID : sas-
postgres error occurred : RESTORE FAILED : Failed to terminate active connections for
database SharedServices.\n , exit status 1", "status": "Failed" }, "properties":
{ "logger": "restorejob", "caller": "restorejob/
main.go:556" }, "timeStamp": "2026-04-08T15:03:07.824444+00:00", "message": "The restore
```

job Failed. For ServiceID : sas-postgres error occurred : RESTORE FAILED : Failed to terminate active connections for database SharedServices.\n , exit status 1"}

### 3 Resolve the migration error or restore error and perform the process again:

If your error refers to autovacuum, perform the following steps:

---

**Note:** For managed services such as Google Cloud, Azure, or AWS, these actions are completed through parameter groups or server configuration settings.

---

- a Disable autovacuum on the PostgreSQL instance (server) level. For more information about autovacuum, see [“Perform Vacuum” in SAS Viya Platform: Infrastructure Servers](#).
- b Perform the restore job or migration job again.
- c Ensure that autovacuum is activated after the migration is complete.

If your error refers to denied permissions or active sessions, perform the following steps:

- a Terminate all active connections to the target database using a Superuser account.

---

**Note:** For managed PostgreSQL services such as Google Cloud, Azure, or AWS, connect using the Query Editor in the portal or a psql client with admin credentials, and run the query below:

---

```
-- Replace <db_name> with the database name
SELECT pg_terminate_backend(pid)
FROM pg_stat_activity
WHERE pid <> pg_backend_pid()
 AND datname = '<db_name>';
```

- b Perform the restore job or migration job again.