

NetBackup™ Self Service Installation Guide

11.2

Document version: 1

NetBackup™ Self Service Installation Guide

Last updated: 2026-09-15

Legal Notice

Copyright © 2026 Cohesity Inc All rights reserved.

© 2026 Cohesity Inc All Rights Reserved. Cohesity, the Cohesity Logo and other Cohesity Marks are trademarks of Cohesity Inc in the US and/or internationally. The information supplied herein is the confidential and proprietary information of Cohesity and may only be used (a) by the intended recipients and (b) in conjunction with validly licensed Cohesity software and services. Find the terms of Cohesity licenses at www.cohesity.com/agreements.

THE DOCUMENTATION IS PROVIDED "AS IS" AND ALL EXPRESS OR IMPLIED CONDITIONS, REPRESENTATIONS AND WARRANTIES, INCLUDING ANY IMPLIED WARRANTY OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT, ARE DISCLAIMED, EXCEPT TO THE EXTENT THAT SUCH DISCLAIMERS ARE HELD TO BE LEGALLY INVALID. COHESITY SHALL NOT BE LIABLE FOR INCIDENTAL OR CONSEQUENTIAL DAMAGES IN CONNECTION WITH THE FURNISHING, PERFORMANCE, OR USE OF THIS DOCUMENTATION. THE INFORMATION CONTAINED IN THIS DOCUMENTATION IS SUBJECT TO CHANGE WITHOUT NOTICE.

Cohesity Support

Reach Cohesity Support

There are several ways to create a Cohesity support case.

- Go to [Cohesity Support](#), to search in our knowledge base; or contact us by phone - United States and Canada: 1-855-9CO-HESI (926-4374), option 2.
- Log in to the [Cohesity Support Portal](#) to create a new case.
- Click the (?) icon on the Cohesity UI and select Support Portal.

Support/Service Assistance

First, contact the Service Provider that you have contracted for service and support. If you work directly with Cohesity and have a product warranty/entitlement, repair pricing, or technical support-related question, see your options below:

- To find solutions to your product issues or for suggestions or best practices, visit the [Cohesity Knowledge Base](#).
- Log in to the [Cohesity Support Portal](#) to create a new case.
- To monitor your open cases, log in to the portal and click the **Cases** tab on the home page. This page should have all the case statuses and updates. You can also view individual case status.

Cohesity Software Running on Partner Hardware

For Cohesity software running on qualified third-party hardware, the following support workflow applies:

1. The customer may contact Cohesity Support first if the issue cannot be determined as a hardware issue.

Note: Cohesity cannot process hardware replacement requests for partner hardware.

2. Cohesity Support triages the issue. If it is a software issue, Cohesity Support continues to work on it.
3. If it is a hardware/firmware issue or is suspected to be a hardware/firmware issue, Cohesity provides information about the issue to the customer and requests that the customer open a support ticket with the appropriate partner.
4. If needed, Cohesity Support can join a three-way call with the partner and the customer.
5. The customer informs Cohesity Support on the progress of the partner's case.

Contents

Chapter 1	Introduction	6
	About Self Service components	6
Chapter 2	Prerequisites	7
	About prerequisites	7
Chapter 3	Installation	9
	Installation overview	9
	Graphical user interface install	10
	Install modes	11
	Validation	12
	Installed components	12
	Uninstall and reinstall	14
Chapter 4	Upgrade	15
	Review current environment configuration	15
	Upgrade preparation	16
	Back up the ApplicationKey and central configuration	18
	Graphical user interface (GUI) upgrade	18
	Upgrade to a new installation directory	19
	Post upgrade resynchronization	19
	Post upgrade steps	19
	Rollback	20
Chapter 5	Post-installation validation	21
	About post-installation validation	21
	Visual Check	21
	IIS configuration check	22
	Windows Service	22
Chapter 6	Uninstallation	23
	Uninstalling NetBackup Self Service	23

Appendix A	Software requirements	24
	NetBackup software requirements for Self Service	24
Appendix B	Checklist for manually editing the Install.ini	26
	Checklist for manually editing the Install.ini	26
Appendix C	Troubleshooting	28
	About PowerShell execution policy	28
	Recovering the REST API signing key	30
	Recovering a lost application key	31
	Common installation errors and fixes	33
Appendix D	Default HTTPS configuration	35
	About the default HTTPS configuration	35
Appendix E	Load-balanced installation and upgrade	36
	Load-balanced installation and upgrade	36
Appendix F	Customizing image upload	46
	About Customizing Image Upload	46
Appendix G	Reduced Database Permissions for Database Upgrade	47
	Reduced Database Permissions for Database Upgrade	47

Introduction

This chapter includes the following topics:

- [About Self Service components](#)

About Self Service components

The installer consists of the `setup.exe` file.

The installation process installs the following components:

- Panels
- Tasks
- Database

The focus of this guide is the two-server install. A web server that hosts the website and Windows Service, and a database server that hosts the databases.

You can extend your NetBackup Self Service solution by using one of the additional add-ons. You can find more information, as well as download details, on the Veritas Open Exchange (VOX). A link takes you to the specific post.

- Cohesity NetBackup Self Service plug-in for VMware vRealize Automation
<https://tinyurl.com/ybldbpcx>
- Cohesity NetBackup Self Service plug-in for VMware vCloud Director
<https://tinyurl.com/y77f68jv>

Prerequisites

This chapter includes the following topics:

- [About prerequisites](#)

About prerequisites

The person who installs NetBackup Self Service needs a working knowledge of SQL Server, Windows Services, and Internet Information Services (IIS).

NetBackup Self Service can be installed on the following Windows platforms:

- Windows Server 2016
- Windows Server 2019
- Windows Server 2022
- Windows Server 2025

Note: Apply the latest service packs to the operating system.

The prerequisites for each component are:

Table 2-1

Component	Requirement
Database	■ Microsoft SQL Server 2016, 2017, 2019, or 2022, allowing SQL authentication method ■ Azure SQL database or Microsoft SQL Server on Amazon RDS ■ At least 5 GB free disk space for data and 2 GB for logs

Table 2-1 (continued)

Component	Requirement
Website and Windows Service	■ Microsoft .NET Framework version 10.0.5 ■ .NET Desktop Runtime 10.0.5 ■ IIS run-time support for .NET 10.0 (ASP .NET core V2) ■ Microsoft PowerShell version greater than 7.2 (part of standard Windows installation) ■ Access to an SMTP server ■ At least 1 GB free disk space ■ Internet Information Services (IIS) ■ Microsoft ODBC driver 17 for SQL Server ■ Microsoft Command Line Utilities 15 for SQL Server Note: You can install the prerequisites that are shown with the installer if the internet is available: Microsoft .NET Framework, .NET Desktop run-time, IIS run-time support, IIS, ODBC driver, Command-Line Utilities. If the internet is not available, two batch files are available that can download all the prerequisites and install them once they are copied to the computer without internet connectivity. The batch files are <code>Download Prerequisites.bat</code> and <code>Install Prerequisites.bat</code> which are located in the Silent Files folder in the setup package. Note: The installer installs version 10.0.5 of the .NET hosting bundle, and it is recommended that you upgrade the version on the server to the latest available version.

Installation

This chapter includes the following topics:

- [Installation overview](#)
- [Graphical user interface install](#)
- [Install modes](#)
- [Validation](#)
- [Installed components](#)
- [Uninstall and reinstall](#)

Installation overview

You can install Self Service either with the `.msi` file. [Table 3-1](#) provides an overview of the process. Additionally, this chapter provides details on where the various Self Service components are installed.

Note: In version 11.2 onwards, silent install is deprecated and the GUI Configurator is the recommended installation method.

Table 3-1 Installation overview

Step	Additional information
Https	See “About the default HTTPS configuration” on page 35.
Graphical user interface (GUI) install	See “Graphical user interface install” on page 10.

Table 3-1 Installation overview (*continued*)

Step	Additional information
Validation	See "Validation" on page 12.

Graphical user interface install

This section guides you through the installation of NetBackup Self Service.

To install NetBackup Self Service

- 1 Run `install_path\setup.exe`, which launches the InstallShield Wizard. The installer runs and copies the installation files onto disk. When it completes, the configurator launches.
- 2 The configurator reviews your local environment for the installation status of NetBackup Self Service.
- 3 If the configurator doesn't find a NetBackup Self Service installation, it shows you a configuration page with the default values that you can change. Select **Start Installation** to start the installation process.
- 4 The **Portal Name** field defines the name of the site. Self Service uses it to create the names of the IIS Applications and Windows service the installer creates. You cannot change the portal name once the installer runs. Choose the **Portal Name** carefully.

(Optional) Use existing database. This mode requires the ApplicationKey from the previous installation. You can retrieve the key with the command `nsscmd -getconfig`.

On the **Database to be created** dialog, enter the information about the database you want created. The default database name is the same as the **Portal Name**. Cohesity recommends that you keep the default database name.

System Base Currency defines the currency type that Self Service uses.

The **System Base Language** defines the language that is displayed in the user interface.

- 5 The final page of the configurator contains the URL for the website. The credentials for the initial logon are:
 - **User ID:** `Admin`.
The user ID is not case-sensitive.
 - **Password:** `password`.
The password is case-sensitive.

You are required to change the password at first logon. Keep a copy of the URL from this screen. Use this URL to connect to the system.

Note: When Create new databases is selected, Main and Adapter database username/password fields are hidden — credentials are auto-generated.

Install modes

The Configurator writes `Install Files\Install.ini` for you during a GUI-driven fresh install, so you do not normally edit the file manually.

Table 3-2 Install modes

Scenario	Edit Install.ini manually?	ApplicationKey
GUI fresh install — new database	No	Generated automatically; do not set ApplicationKey=
GUI fresh install — use existing DB	No (11.2+)	Enter in Configurator; written to <code>Install.ini</code>
GUI upgrade	No	Read from installed central Config; not prompted in UI

Do not set ApplicationKey= for a new database. The installer generates a new key and removes any leftover Config\ folder from a previous uninstall.

Attach to an existing database (UseExistingDatabase=1)

- Set UseExistingDatabase=1, then supply the ApplicationKey by using one of the following methods:
 - ApplicationKey=<hex key from `nsscmd -getconfig` on source server>
 - LegacyProgramFolder=C:\path\to\old ProgramFolder
- If you use LegacyProgramFolder, it must contain a readable `Config\machine.secrets.enc` file from the source install. Alternatively, if you uninstalled a previous install to the same target ProgramFolder, the installer reads the key from the leftover Config\ folder there.
- The installer resolves the ApplicationKey in this order: ApplicationKey=, ApplicationKeyFile=, LegacyProgramFolder\Config, then {ProgramFolder}\Config.
- Also set DatabaseName, AdapterDatabaseName, and the SQL credentials (MainDBUserName, MainDBPassword, AdapterDBUserName,

AdapterDBPassword) to match the existing databases. When possible, the installer verifies the ApplicationKey against SYS_APP_KEY_CHECK. For an upgraded database that has a stale check value, a warning may appear; the installer refreshes the value at the end of the install.

Note: When you select **Create new databases**, the Configurator hides the Main and Adapter database username and password fields and generates the credentials automatically.

Validation

Verify the installation

To verify the installation

- 1 Run the following commands from an elevated command prompt:

```
cd "{ProgramFolder}\Install Files"
```

```
nsscmd.exe -validateconfig -configDir "{ProgramFolder}\Config"
```

```
nsscmd.exe -readconfig -configDir "{ProgramFolder}\Config"
```

- 2 Confirm portal access.

Open the portal URL listed in the install log and confirm that the portal loads successfully.

More information on how to validate the installation and perform the initial setup is available.

See [“About post-installation validation”](#) on page 21.

Installed components

This section shows the results of a default installation of NetBackup Self Service. It shows the components that are installed and where they are installed.

File System

The portal and the adapter are installed under `C:\Program Files\Cohesity\NetBackup Self Service version_number`.

IIS

In this configuration, an IIS public website is created to host the web pages.

Windows Service

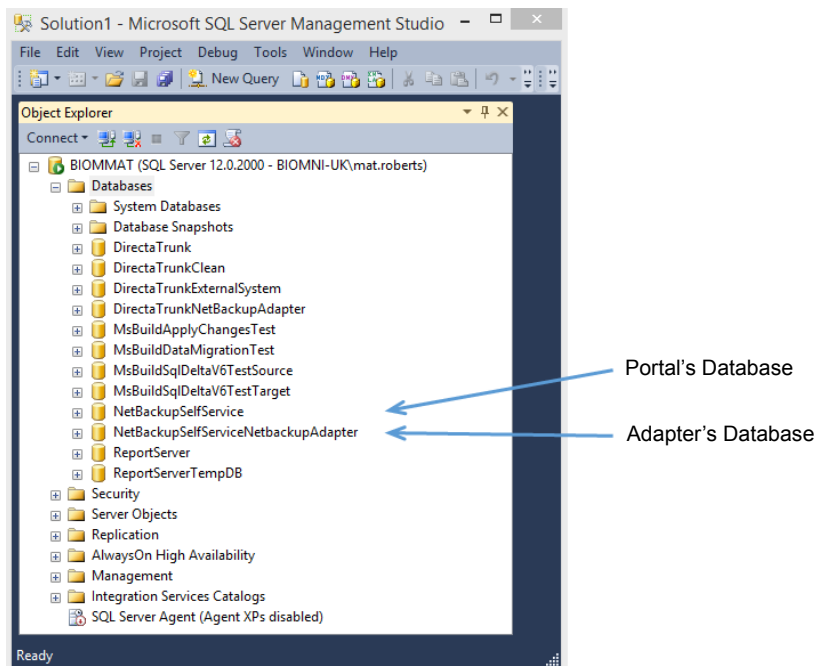
The portal installs a Windows Service.

Database

Two databases are created:

- Portal's database: **NetBackup Self Service**
- Adapter's database: **NetBackupSelfServiceNetBackupAdapter**

Figure 3-1 Self Service Databases



Soap service

The Soap Service URL has changed to

`https://domain:httpsport/NetBackupSelfServicePublicWebService`. The endpoint for automation against the soap service is available in the `wsdl` file at the able location at the very end of the document and appears as follows:

```
<?xml version='1.0' encoding='utf-8'>
<wsdl:service name="DirectaApiWcf">
  <wsdl:port name="BasicHttpBinding_IDirectaApi" binding="tns:BasicHttpBinding_IDirectaApi">
    <soap:address location="http://localhost:8080/NETBACKUPSELFSERVICEPUBLICWEBSERVICE/DirectaApi"/>
  </wsdl:port>
  <wsdl:port name="BasicHttpBinding_IDirectaApi1" binding="tns:BasicHttpBinding_IDirectaApi1">
    <soap:address location="https://localhost:444/NETBACKUPSELFSERVICEPUBLICWEBSERVICE/DirectaApi"/>
  </wsdl:port>
</wsdl:service>
```

The **Location** value is your endpoint for automation purposes. Note that the URL ends with `DirectaApi`.

Uninstall and reinstall

`UnInstall.bat` removes IIS applications, application pools, and the Directa Windows service. It preserves the `{ProgramFolder}\Config` folder to support disaster recovery.

After uninstall, use the appropriate reinstall path:

- New databases, same path: The installer removes leftover Config automatically and generates a new ApplicationKey.
- Existing databases, same path: The installer reuses leftover Config when the ApplicationKey matches. If Config is unreadable, provide the key in the Configurator or `Install.ini`.
- Different path: Set `LegacyProgramFolder=`, `ApplicationKey=`, or `ApplicationKeyFile=` in `Install.ini`.

Upgrade

This chapter includes the following topics:

- [Review current environment configuration](#)
- [Upgrade preparation](#)
- [Back up the ApplicationKey and central configuration](#)
- [Graphical user interface \(GUI\) upgrade](#)
- [Upgrade to a new installation directory](#)
- [Post upgrade resynchronization](#)
- [Post upgrade steps](#)
- [Rollback](#)

Review current environment configuration

Before you begin the upgrade, review your existing installation. Self Service has six components that are typically distributed across two servers.

Table 4-1 Typical Self Service configuration

Location	Component
IIS server	■ NetBackupSelfService ■ NetBackupSelfServicePublicWebService
Windows services	■ Portal Windows service ■ Adapter Tasks (Self Service 9.1 and later)

Table 4-1 Typical Self Service configuration (*continued*)

Location	Component
SQL server	■ Portal database ■ Adapter database

You can identify the components in your environment from within NetBackup Self Service.

- Identify the IIS components.
Log on to the web server and open **Internet Information Services (IIS) Manager**.
Browse the sites and identify the two IIS components that are listed in [Table 4-1](#).
- Identify the Windows service.
Log on to the server with the Windows Service. In a default installation of Self Service, the service is located on the web server.
Open **Services** and locate the **Portal Windows Service**.
- Identify the databases.
Open Microsoft SQL Server Management Studio, and connect to the database server.
Identify the two databases that are listed in [Table 4-1](#).
See “[Installed components](#)” on page 12.

Upgrade preparation

You must perform several steps to prepare for an upgrade.

To prepare for an upgrade

- 1 Before you upgrade, back up the Application key, central configuration, databases, and any customized email templates. Use these backups to restore configuration information if the upgrade is unsuccessful.
 - Export the central configuration
 - On the web server, open Command Prompt as an administrator.
 - Go to the installation files folder:
`cd "{ProgramFolder}\Install Files"`
 - Run the following command to export the central configuration:
`nsscmd.exe -getconfig`
 - Save the command output in a secure location.

- Back up databases and email templates
 - Back up both Self Service databases if database backup is part of your upgrade process.
 - If you customized email templates in version 11.1 or earlier, export the XML files from `Website\EmailTemplates`, or record the customizations before the upgrade.
 - After the upgrade, re-create the email templates in **Settings > Email Templates**, and then click **Save**. In version 11.2 or later, the email templates are stored in the database.

2 Back up the databases

You should back up both Self Service databases before you start the upgrade. The default names for the databases are **NetBackupSelfService** and **NetBackupSelfServiceNetBackupAdapter**. Perform these steps in **SQL Server Management Studio**.

- Make a note of the **NetBackupSelfService** database recovery model.
- Set the database recovery model to **Simple**.
- Back up the database.
- Make a note of the **NetBackupSelfServiceNetBackupAdapter** database recovery model.
- Set the database recovery model to **Simple**.
- Back up the database.

3 Take the portal offline.

Cohesity recommends that you prevent user logon and user activity while the upgrade is active. The best way to prevent user logon and user activity is to use **Internet Information Services (IIS) Manager** to stop the application pool for the portal website.

If a user attempts to connect to the website when the application pool is stopped, they receive an HTTP Error 503. The service is unavailable error in their web browser.

Do not stop the other application pools during the upgrade. If you stop the application pools with the suffix **PublicWebServiceAppPool** then the upgrade fails. The public web service is used for the upgrade.

Back up the ApplicationKey and central configuration

Before you upgrade, save the ApplicationKey and central configuration so you can recover or reconnect to existing databases if needed.

To back up the ApplicationKey and central configuration

- 1 On the web server, open an elevated command prompt.
- 2 Change to the `Install Files` folder:

```
cd "{ProgramFolder}\Install Files"
```
- 3 Export the current configuration:

```
nsscmd.exe -getconfig
```
- 4 Save the full command output in a secure location.
- 5 Back up both Self Service databases.
- 6 If you customized email templates in 11.1 or earlier, export or record those changes before you upgrade. In 11.2, re-create the templates in **Settings > Email Templates**, and then click **Save**. Version 11.2 stores email templates in the database.

Graphical user interface (GUI) upgrade

To upgrade NetBackup Self Service

- 1 Run the installer `install_directory\setup.exe`.
The installer runs and copies the installation onto the computer. When the installation completes, a configurator launches.
- 2 The original configuration parameters are filled automatically. Select **Start Upgrade** to start the upgrade.
Database credentials for upgrade are read from installed central Config (Configurator SQL sign-in is hidden during upgrade).
- 3 After successful completion of the upgrade, verify that your machine currently uses the latest versions of Microsoft .NET Runtime 10.0 and Microsoft Windows Desktop Runtime 10.0. If it uses earlier versions of those applications, go to the Microsoft download site to obtain the latest version of each one, and then install them.

Upgrade to a new installation directory

When the physical paths point to a different program folder, `Upgrade.bat` copies the central Config files from the old folder to the new folder before it starts the database steps.

For example, this applies when you move the installation from `C:\Program Files\...\11.2` to `D:\NSS\11.2`. The copied files include `machine.secrets.enc` and `connectionStrings.json`.

After a successful upgrade, `Upgrade.bat` removes the central Config files from the old folder only after the destination Config validates successfully.

If the destination Config already exists, the `ApplicationKey` values in the source and destination folders must match. If they do not match, the upgrade stops.

Post upgrade resynchronization

After the upgrade, manually resynchronize the data between NetBackup and vCloud Director, if Cloud is used. Two scheduled tasks run once per day: System Sync and Asset Import. System Sync imports any new backup images from all backup servers, expires old backup images, and calculates usage. Until these tasks are run, the data that is displayed to the user may be incomplete.

To resynchronize the data

- 1 Log in to the website as an administrator, and navigate to the **Monitoring** tab. The left side of the screen shows scheduled tasks.
- 2 Click the cog next to the **System Sync** task and select **Run Now**.
- 3 (Conditional) If you use vCloud Director, click the cog next to the **Asset Import** task, and select **Run Now**.
- 4 The **Activity** section on the right side of the screen monitors the progress of these tasks.

If you do not perform these steps manually, the tasks run automatically overnight.

Post upgrade steps

After the upgrade finishes, complete a connectivity check for all primary servers. Cohesity also recommends that you re-synchronize all data.

As part of the upgrade, the two databases were backed up and the recovery model was set to **Simple**. Revert the database's recovery model to its initial value.

To revert the database to its initial value:

- 1 Shrink both databases.
- 2 Set the recovery model of the databases back to its original value.

When an upgrade is performed a new set of code is placed in a new location on the server. Once the upgrade is complete, remove the old installation.

To remove old installation code

- 1 Go to **Add/Remove Programs**.
- 2 Uninstall any previous versions of:
 - **NetBackup Self Service Portal**
 - **NetBackup Self Service Adapter**

Rollback

To revert back to the previous version, a restore of the two NetBackup Self Service databases is required. Additionally, you must reinstall the previous portal and adapters or restore their web server from a backup).

If you reinstall the portal and the adapters, ensure **Database** is not selected on the **Select Components** dialog box during install. In both cases the database is restored and does not need to be reinstalled.

During the portal reinstallation, when prompted for an application key, enter the application key from the previous installation. This application key is the key used to encrypt third party passwords in the restored databases and was recorded when the previous version was deployed.

Post-installation validation

This chapter includes the following topics:

- [About post-installation validation](#)
- [Visual Check](#)
- [IIS configuration check](#)
- [Windows Service](#)

About post-installation validation

When you complete the installation, you can validate the installation with a series of checks.

Table 5-1 NetBackup Self Service validation checklist

Validation	Additional details
Perform a visual check of the website main screen.	See " Visual Check " on page 21.
Confirm the Windows service is configured correctly.	See " Windows Service " on page 22.

Visual Check

After installation it is important to check that the system has installed correctly. Log on to the portal website. The main screen of the website should display correctly.

IIS configuration check

After installation, check that IIS is configured correctly. Search for **Internet Information Service** in the **Start** menu.

- Under **Internet Information Service > Sites > Default website (or existing NSS website) > Bindings**, confirm that the installer created an SSL certificate.
- Under **Internet Information Service > Sites > Default website (or existing NSS website) > Bindings**, confirm the binding for port 443 is created and the certificate is assigned to that port.

Windows Service

After an install, it is advisable to check that the Windows service is running correctly. On the server where the Windows service is installed:

- Open Event Viewer, and navigate to the Application Log.
- Find messages with a source of **DirectaService11.2\$NetBackupSelfService**. The name may vary slightly - the naming convention is **DirectaService11.2\$SiteName**, where *SiteName* is the name of the website.
- If the Windows service has logged any errors then it is possible there is a configuration problem. Examine the detail of the error.

A common configuration problem is the Windows service cannot connect to the database. The Windows service checks to confirm that connectivity to the database is defined in the configuration file. If the service cannot connect to the database it logs an error in the Windows Event Log.

Uninstallation

This chapter includes the following topics:

- [Uninstalling NetBackup Self Service](#)

Uninstalling NetBackup Self Service

The uninstallation process removes the Windows service that is connected to the installation location. It then deletes the software on the hard disk.

The uninstallation does not delete the two databases that were created. The databases must be deleted manually.

To uninstall a NetBackup Self Service

- 1 Determine the version of NetBackup Self Service you want to uninstall.
- 2 Locate and stop the service **Cohesity Front Office Service 11.2 (NetBackupSelfService)**
- 3 In Windows open **Programs and Features**.
- 4 Locate **NetBackup Self Service version**, and select uninstall.

When the uninstall process finishes, delete the databases from within SQL Server Management studio. From **Object Explorer**, expand the **Databases** node. Right-click on each of the relevant databases and select **Delete**.

Software requirements

This appendix includes the following topics:

- [NetBackup software requirements for Self Service](#)

NetBackup software requirements for Self Service

NetBackup 8.0 or later with the latest service pack is required. With a Windows primary server, only US English operating system and code page installations of NetBackup are supported. NetBackup language packs are not supported.

With a UNIX primary server, the character encoding of the primary server operating system must be UTF-8. Multiple locales are supported. NetBackup language packs are supported.

NetBackup appliances are supported.

Software requirements for Self Service

The Self Service software requirements are

- If using a vCloud Director Integrated configuration, check the Software Compatibility List for supported API versions.
<http://www.netbackup.com/compatibility>
- NetBackup Self Service works on any virtual platform, such as Hyper-V or vSphere, provided one of the supported operating systems is installed.

The lists that are shown define the supported operating systems, SQL servers, and web browsers. The latest service pack should always be used.

Note: Any version of operating system, SQL server, and web browser that is not listed as **Supported** is considered unsupported.

Supported operating systems:

- Windows Server 2016
- Windows Server 2019
- Windows Server 2022
- Windows Server 2025

Supported SQL server:

- SQL Server 2016
- SQL Server 2017
- SQL Server 2019
- SQL Server 2022
- Azure SQL database
- SQL Server on Amazon RDS

Supported browsers:

- Internet Explorer
- Edge
- Firefox
- Chrome
- Safari (supported, but not recommended)

Checklist for manually editing the Install.ini

This appendix includes the following topics:

- [Checklist for manually editing the Install.ini](#)

Checklist for manually editing the Install.ini

Use this checklist when you run `Install.bat` without the Configurator.

Required settings

- Specify **WebsiteName**, **ApplicationName**, and **DatabaseServer**.
- Set **UseWindowsAuthentication**. If you do not use Windows authentication, provide the SQL administrator credentials in **UserNameForSA** and **PasswordForSA**.
- Specify **DatabaseName**. If the adapter database uses a **nondefault** name, also specify **AdapterDatabaseName**.
- Set **SystemBaseCurrency** and **SystemBaseLanguage**.
- For an existing database, set **UseExistingDatabase=1**. Provide one application key source: **ApplicationKey**, **ApplicationKeyFile**, or **LegacyProgramFolder**.
- For an existing database that uses SQL authentication, provide **MainDBUserName**, **MainDBPassword**, **AdapterDBUserName**, and **AdapterDBPassword**.
- For a new database installation, do not set **ApplicationKey**.
- After installation succeeds, remove or redact secrets from `Install.ini`.

Application key resolution order

When **UseExistingDatabase=1**, the installer resolves the application key in the following order: **ApplicationKey** → **ApplicationKeyFile** →

`LegacyProgramFolder\Config` → `ProgramFolder\Config`.

Post-installation verification

After installation, validate the configuration by running:

```
nsscmd.exe -validateconfig -configDir "{ProgramFolder}\Config".
```

Troubleshooting

This appendix includes the following topics:

- [About PowerShell execution policy](#)
- [Recovering the REST API signing key](#)
- [Recovering a lost application key](#)
- [Common installation errors and fixes](#)

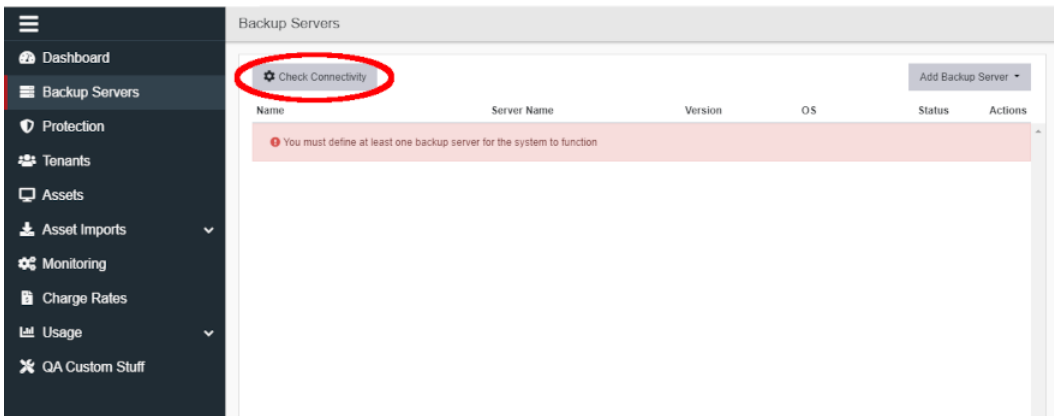
About PowerShell execution policy

The PowerShell execution policy determines if PowerShell can run scripts. The installer sets the execution policy to **Remote Signed** which allows scripts to run. Problems are encountered if this step of the installer fails or the execution policy is changed after install. This appendix describes diagnosing and solving execution policy issues.

Diagnosis

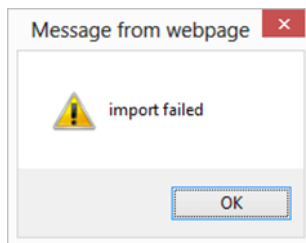
- Log on to the website
- Click the **Backup Servers** tab.
- Click the **Check Connectivity** icon

Figure C-1 Check connectivity



If you receive the error message shown, there may be an execution policy issue. If **Check Connectivity** does not generate an error, the execution policy is set correctly.

Figure C-2 Import failed pop-up box



To confirm there is an execution policy issue, navigate to the error log. Select `%ProgramData%\Cohesity\NetBackupSelfService` and examine the errors. An example of an execution policy issue is shown.

```
"CreateRequest failed with error:
File C:\Temp\NetBackupAdapter\NetBackupAdapterServices\PowerShellScripts\
ValidationHook\Initial.ps1 cannot be loaded because running scripts is
disabled on this system. For more information, see about_Execution_Policies
at http://go.microsoft.com/fwlink/?LinkID=135170. File C:\Temp
\NetBackupAdapter\NetBackupAdapterServices\PowerShellScripts\ValidationHook\
Initial.ps1 cannot be loaded because running scripts is disabled on this
system. For more information, see about_Execution_Policies at
http://go.microsoft.com/fwlink/?LinkID=135170."
```

Solution

- 1 Log on to the web server
- 2 Open a PowerShell command prompt as administrator.
- 3 Type: `Get-ExecutionPolicy -List`
The list of the current execution policies is shown
- 4 If the **Local Machine Scope** is not set to **Remote Signed**, type the command:

```
Set-ExecutionPolicy -Scope LocalMachine -ExecutionPolicy
RemoteSigned
```

Execution policy scope treats items higher up the list as higher priority, overriding those lower in the list. If the scope **MachinePolicy** is set to **Restricted**, then even though **LocalMachine** is set to **RemoteSigned** you are still unable to run scripts. This Stack Overflow post describes how to solve such problems.

<http://stackoverflow.com/a/27755459>

Recovering the REST API signing key

Use this procedure if the REST API JSON Web Token (JWT) signing key is lost or the `machine.secrets.enc` file is corrupted, but the application encryption key and the databases are intact.

To recover the REST API signing key

- 1 Generate or choose a new signing key of at least 32 characters:
`nsscmd -genkey.`
- 2 Set the key:
`nsscmd -setsecret -configDir "{NSS installation folder}\Config"
-name JWT:IssuerSigningKey -value "{new-key}"`
- 3 Recycle the IIS application pools, and then restart the Directa Windows service.
- 4 Inform API users that their existing bearer tokens are no longer valid. Clients must obtain new tokens through the authentication endpoint.

Rebuilding machine.secrets.enc

If you are rebuilding the `machine.secrets.enc` file and you know both the application encryption key and the JWT signing key that you want to use, create a file named `machine.secrets.json` in the Config folder with the following content:

```
{
  "ApplicationKey": "your-known-application-key",
```

```
"JWT": { "IssuerSigningKey":
"your-jwt-signing-key-at-least-32-chars" }
}
```

Then run the following commands:

- `nsscmd -protectsecrets -configDir "{Config}"`
- `nsscmd -validateconfig -configDir "{Config}"`

Recovering a lost application key

Important: Always back up the application key before an upgrade or immediately after an installation. You cannot recover the application key from the database. NetBackup Self Service stores the key only in the `Config` folder in a DPAPI-protected file, which can be read only on the machine where it was created. To support disaster recovery and future installations, keep a secure backup copy of the application key.

Overview

The `ApplicationKey` encrypts third-party integration passwords in the database. If the key is lost, users might still be able to sign in, but encrypted adapter and integration passwords cannot be decrypted. You must re-enter those passwords in the NetBackup Self Service UI.

Where the application key is stored in version 11.2 or later versions

- **Primary location:** `{ProgramFolder}\Config\machine.secrets.enc`. This file is DPAPI-protected and can be read on the same machine by running the command `nsscmd -readconfig`.
- **Backup reference:** Save the output from `nsscmd -getconfig` after every installation or upgrade.
- **Database connection strings:** The `connectionStrings.json` file in the same `Config` folder contains database connection strings and is also required for disaster recovery.

Common recovery scenarios

- **Web server failure:** Restore the web server or `{ProgramFolder}` from backup. Alternatively, restore the `Config` folder from backup to a replacement server.
- **Move to a new web server:** Before uninstalling the old site, obtain the `ApplicationKey` from the old server or from the saved `nsscmd -getconfig` output. On the new server, install with `UseExistingDatabase=1` and the same key.

Recovery steps

1. Restore the web server, {ProgramFolder}, or at minimum the Config folder from backup to the target server.
2. Confirm the ApplicationKey on that server by running one of the following commands from {ProgramFolder}\Install Files.
 - `nsscmd.exe -readconfig -programFolder "{ProgramFolder}"`
 - `nsscmd.exe -readconfig -configDir "{ProgramFolder}\Config"`

Note: The `machine.secrets.enc` file is DPAPI-protected. If you copy the Config folder to a different machine, DPAPI might prevent the new host from reading the secrets. In that case, use the saved `nsscmd -getconfig` output, or provide the known ApplicationKey by using `-initconfig` or `-setconfig` on the new server.

Disaster recovery on a new web server

When you install NetBackup Self Service on a replacement web server, provide the existing ApplicationKey by using one of the following methods:

- **Configurator:** Enter the application key in the **Use existing database** workflow.
- **Install.ini:** Set ApplicationKey=<hex key from `nsscmd -getconfig` on the source server>.
- **Install.ini file reference:** Set ApplicationKeyFile=C:\secure\nss-getconfig.txt.
- **Legacy program folder:** Set LegacyProgramFolder=C:\path\to\copied\old ProgramFolder. The copied folder must contain a readable `Config\machine.secrets.enc` file from the source installation.

Before the installation continues, the installer verifies the ApplicationKey against SYS_APP_KEY_CHECK.

Load-balanced nodes: Copy the configuration from the primary node by running `nsscmd -getconfig` on the source server and `nsscmd -setconfig` on each additional node. For more information, see Appendix D.

If the application key and Config backup are lost

The ApplicationKey cannot be recovered from the SQL database alone. Encrypted passwords stored in the database, such as integration settings and adapter credentials, cannot be decrypted. After you restore database connectivity and establish a new or known ApplicationKey, re-enter those values in the NetBackup Self Service administration UI.

Note: Important: You cannot change the ApplicationKey by using `nsscmd -setsecret`. Do not attempt to rotate the key after initial configuration because doing so breaks access to existing encrypted data.

Rebuild the Config folder when the application key is known

Run the following commands from `{ProgramFolder}\Install Files`.

- If the `Config` folder does not contain secrets:


```
nsscmd.exe -initconfig -programFolder "{ProgramFolder}"
      -applicationKey "{known-hex-key}"
```
- If `connectionStrings.json` exists and you have a full saved `getConfig` string:


```
nsscmd.exe -setconfig "ApplicationKey=...;MainDBServer=...;..."
```
- Validate the configuration:


```
nsscmd.exe -validateconfig -configDir "{ProgramFolder}\Config"
```

As part of the standard NetBackup Self Service backup process, back up the entire `Config` folder and the saved `nsscmd -getConfig` output.

Common installation errors and fixes

Table C-1 Common installation errors and fixes

Symptom	Cause / fix
UseExistingDatabase=1 needs an ApplicationKey	No key was found in <code>Install.ini</code> , the key file, the legacy folder, or <code>Config</code> . Restore the key from the <code>-getConfig</code> backup and set <code>ApplicationKey=</code> .
<code>initconfig</code> or <code>validateconfig</code> reports a key mismatch	The <code>Config</code> folder may be stale after uninstall. For existing databases, set the correct key and let the installer remove the stale <code>Config</code> folder. For new databases, cleanup is automatic.
ApplicationKey does not match existing database	The key or database selection is incorrect. Use the key from the original installation.
Warning: SYS_APP_KEY_CHECK could not be verified	This warning is common for upgraded databases. Continue only if the key is trusted; the installer refreshes the check value at the end.

Table C-1 Common installation errors and fixes (*continued*)

Symptom	Cause / fix
Database already exists (new-database install)	Use existing-database mode, or drop or rename the databases if it is safe to do so.
Empty <code>Config</code> folder after upgrading to new folder.	Make sure the old program folder is accessible. If needed, copy <code>Config</code> manually or set <code>LegacyProgramFolder=</code> .
InitData / Front Office initialization fails	Review the install log. In 11.2, failures include <code>sqlcmd</code> output. Verify SQL permissions and available disk space.
Dapi password still ChangeMe in <code>connectionStrings</code>	This is a known limitation. Configure Dapi after installation.
Second node <code>initconfig</code> fails (load balanced)	Use the same <code>ApplicationKey</code> hex value. Do not copy <code>.enc</code> files between servers.

Default HTTPS configuration

This appendix includes the following topics:

- [About the default HTTPS configuration](#)

About the default HTTPS configuration

The installation configures the site to use HTTPS by default. The installation adds the self-signed certificate by default and creates an HTTPS binding.

Load-balanced installation and upgrade

This appendix includes the following topics:

- [Load-balanced installation and upgrade](#)

Load-balanced installation and upgrade

This appendix provides the supported **procedure** for deploying NetBackup Self Service 11.2 or later, on multiple web servers that share one SQL Server instance containing the Main and Adapter databases.

Note: Applies to NSS 11.2 or later with central `{ProgramFolder}\Config\` created or migrated after installation.

Overview

A load-balanced NSS deployment consists of multiple IIS web servers running NSS and DirectaService behind a load balancer. All web servers use the same SQL Server instance for the Main and Adapter databases. Each web server retains its own local ProgramFolder and machine-specific DPAPI-protected secrets.

Figure E-1 Load-balanced installation example

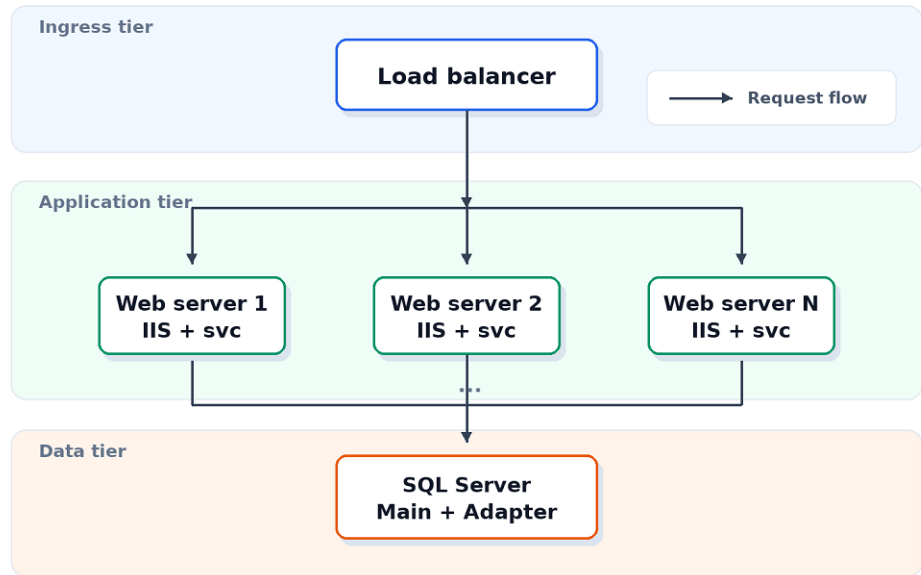


Table E-1 Architecture requirements

Component	Requirement	Implementation
SQL Server	One instance	Hosts both the Main and Adapter databases.
IIS + DirectaService	One per web server	Each node has its own {ProgramFolder}.
ApplicationKey + connection strings	Must match on every node	Stored in {ProgramFolder}\Config\ after installation.
Uploaded images	Shared storage	Use the same UNC path on every node when users upload files.

Note: The Main database and Adapter database cannot be placed on different SQL Server instances. `Install.ini` and `nsscmod -getconfig` use a single `DatabaseServer` for both connection strings.

Prerequisites

Complete the following requirements on every web server before installation.

- Windows Server 2016 or later.
- IIS and the .NET 10 hosting bundle. The installer can install prerequisites.
- Network connectivity from each web server to the SQL Server instance over TCP, with required firewall rules.
- SQL authentication or Windows authentication that works from each web server to the remote SQL Server.
- A load balancer or DNS round-robin in front of the IIS sites. Configure it after the nodes are healthy.
- A UNC share for uploaded images when users upload files. A shared image location is required for deployments with more than one web node.

Deployment planning

- Portal name (IIS application name). Use the same portal name on every node.
- Database names on the shared SQL Server instance.
- The ApplicationKey. All nodes must use the same ApplicationKey because they share the same SQL databases.

Note: Each node has its own `{ProgramFolder}\Config\machine.secrets.enc`. This file is protected with Windows DPAPI on the local machine. Do not copy `machine.secrets.enc` between servers; DPAPI decryption fails on another machine.

Configuration tools

Run the commands from `{ProgramFolder}\Install Files` on each node. The installed version folder is used in the path; for example:

```
C:\Program Files\Cohesity\NetBackup Self Service 11.2\Install Files
```

Table E-2

Command	Purpose
<code>nsscmd.exe -getconfig</code>	Exports the ApplicationKey and connection fields as one semicolon-delimited string. Use it to back up the configuration and align other nodes. Central Config\ must exist.

Table E-2 (continued)

Command	Purpose
<code>nsscmd.exe -setconfig "<string from getconfig>"</code>	Applies the same secrets and connection strings to the local central Config\ . The command restarts IIS and the local DirectaService. Config\ must already exist.
<code>nsscmd.exe -readconfig -programFolder "{ProgramFolder}"</code>	Reads configuration from a specified ProgramFolder for diagnostics.

Note: Save the complete `nsscmd -getconfig` output from the primary node in a secure vault after the primary node is installed or upgraded. You need this output before uninstalling the primary node.

Perform a load-balanced fresh install

Use the primary-node plus **Use existing database** workflow. This is the recommended NSS 11.2 deployment method because it avoids creating throwaway databases on secondary servers.

Install the primary web server (node 1)

1. Run `setup.exe` and open the Configurator.
2. Select **Create new database**.
3. Enter the **Database Server**. Specify the remote SQL host name, host\instance, or host,port.
4. Complete the installation. The installer generates the Main and Adapter database credentials.
5. Verify that you can log on to the portal.
6. Open an elevated command prompt and change to `{ProgramFolder}\Install Files`.

```
cd "{ProgramFolder}\Install Files"
```

```
nsscmd.exe -getconfig
```

Save the complete command output securely in a password vault or controlled runbook.

Install each additional web server (nodes 2 through N)

1. Run `setup.exe` on the additional web server. NSS must not already be installed on this machine.

2. In the Configurator, use the same portal name and IIS site choices as node 1.
3. Enter the same **Database Server** used by node 1.
4. Select **Use existing database**.
5. Select **Load existing** and choose the same Main and Adapter database names used by node 1.
6. Enter the ApplicationKey from node 1. Use the hex value from `nsscmd -getconfig` or only the ApplicationKey= value.
7. Complete the installation. The `-initconfig` phase creates the local `{ProgramFolder}\Config\` and attaches the node to the shared databases.

Note: If the ApplicationKey is unavailable supply it by using `Install.ini ApplicationKeyFile=` or `LegacyProgramFolder=` as described in the Installation Guide.

Align configuration across all nodes

Even when secondary nodes use the same databases, run `-setconfig` on nodes 2 through N. This ensures that DAPI credentials and all connection fields match that of node 1.

1. On node 1, capture the configuration if it has not already been saved:

```
cd "{ProgramFolder}\Install Files"

nsscmd.exe -getconfig > \\secure-share\nss-node1-getconfig.txt
```
2. On each secondary node, run the following command from an elevated prompt. Replace the example string with the complete, single-line output from node 1.

```
nsscmd.exe -setconfig "ApplicationKey=...;MainDBServer=...;
MainDBName=...;MainDBUserName=...;MainDBPassword=...;
AdapterDBServer=...;AdapterDBName=...;AdapterDBUserName=...;
AdapterDBPassword=...;DapiUserName=...;DapiPassword=..."
```

Note: Paste the entire single-line string returned by `-getconfig`. The `-setconfig` command validates that every required key is present. It also restarts IIS and the local DirectaService.

Configure shared uploaded images

Configure the same UNC path for uploaded images on every node. The uploaded-image path is not centralized in all configuration flows; confirm the path in the applicable administrator settings or integration configuration.

```
\\fileserver\NSSImages
```

Note: The default local path `%SYSTEMDRIVE%\inetpub\Cohesity\Images` is local to each server. Change it to a shared UNC path for load-balanced deployments. See [“Reduced Database Permissions for Database Upgrade”](#) on page 47.

Configure the load balancer and validate the farm

1. Point the load balancer to each node's HTTPS binding. Use the same host-header and certificate strategy required by your environment.
2. Log on through the load-balancer URL.
3. Confirm that sessions work and that uploaded images are available.
4. On each node, confirm that DirectaService is running and that the Windows Event Log is clean.
5. Optionally run `-getconfig` on each node and compare the ApplicationKey. The ApplicationKey must match on every node.

Legacy installation method: Temporary secondary databases

Note: Use this workflow only when **Use existing database** is unavailable.

1. Install node 1 with new databases using the normal installation workflow.
2. On nodes 2 through N, install using new database names with a node-specific suffix, for example `NetBackupSelfService2`. The installer creates temporary databases.
3. On node 1, run `nsscmd.exe -getconfig`.
4. On nodes 2 through N, run `nsscmd.exe -setconfig "<full string>"` after installation has completed the `-initconfig` phase and `Config\` exists.
5. In SQL Server Management Studio, drop the unused secondary-node databases.

For NSS 11.2, prefer **Use existing database** to avoid the temporary databases and cleanup step.

Command-line installation

For scripted secondary-node installations, edit `Install Files\Install.ini` before running `Install.bat`.

1. The following settings are from the source procedure; replace environment-specific values with your deployment values.

```
WebsiteName=Default Web Site
ApplicationName=/NetBackupSelfService
DatabaseServer=sql01.contoso.com
UseWindowsAuthentication=0
UserNameForSA=sa
PasswordForSA=
DatabaseName=NetBackupSelfService
AdapterDatabaseName=NetBackupSelfServiceNetBackupAdapter
UseExistingDatabase=1
ApplicationKey=<hex key from node 1 getconfig>
MainDBUserName=...
MainDBPassword=...
AdapterDBUserName=...
AdapterDBPassword=...
SystemBaseCurrency=USD
SystemBaseLanguage=en-US
```

2. Run `Install Files\Install.bat` as Administrator.
3. After installation, apply `-setconfig` from node 1 if you need an exact clone of DAPI settings.

Perform a load-balanced upgrade

There is no separate farm-upgrade wizard. Upgrade each web server that runs NSS.

Prepare for the upgrade

1. Back up both databases on the shared SQL Server instance.
2. On a reference node, run `nsscmd.exe -getconfig` and save the output.
3. Plan maintenance. Drain or stop the portal application pool on one node at a time while keeping the Public Web Service available according to standard upgrade guidance.

Upgrade the web servers

1. Upgrade node 1 using Configurator → **Start Upgrade**. `Upgrade.bat` runs database scripts against the shared database and `-migrateconfig` on the local ProgramFolder.
2. Verify the portal on node 1 and run `-getconfig`.
3. Upgrade nodes 2 through N one at a time using the same procedure. The Configurator detects the existing NSS installation on each machine.
4. Validate the database version after the first node before continuing. Although the database upgrade scripts are designed to be re-runnable, upgrade one node first and validate before proceeding.
5. After all nodes are on NSS 11.2, optionally run `-setconfig` from node 1 on any node where the `-getconfig` output differs.
6. Run `-getconfig` on the primary node again and update the secure vault copy.

Validate after the upgrade

- Confirm that all nodes report the same ApplicationKey in `-getconfig`.
- Confirm that the shared image UNC path is valid on every node.
- Test failover through the load balancer.

Troubleshooting

Table E-3 Troubleshooting scenarios

Symptom	Likely cause	Action
<code>-getconfig</code> fails: "Central Node not yet on Phase 2 Complete install or upgrade" or configuration files were not found.	Central Config\ is missing.	Run the applicable <code>-initconfig</code> or <code>-migrateconfig</code> phase first.
<code>-setconfig</code> fails: "Invalid values".	The configuration string is incomplete, for example DapiUserName or DapiPassword is missing.	Use the complete string from <code>-getconfig</code> on node 1.
<code>-setconfig</code> fails: central config not found.	Installation did not finish on the node.	Complete the installation through <code>-initconfig</code> before running <code>-setconfig</code> .
Secondary node cannot load existing databases.	The web server cannot reach SQL Server or authentication is incorrect.	From the web server, test the SQL connection with <code>sqlcmd</code> using the applicable server and credentials.

Table E-3 Troubleshooting scenarios (*continued*)

Symptom	Likely cause	Action
Images are missing on some nodes.	PathForUploadedImages points to a local path instead of a shared UNC path.	Set the same UNC path on all nodes.
Different ApplicationKey on two nodes.	A secondary node used the new-database path or the wrong ApplicationKey.	Run -setconfig from node 1, or reinstall the secondary node with UseExistingDatabase=1 and the correct key.
ApplicationKey mismatch or validateconfig failed on node 2 or later.	A leftover {ProgramFolder}\Config\ from a previous installation contains a different ApplicationKey.	Confirm node 1's ApplicationKey. Reinstall using Use existing database and the correct key. The installer removes stale Config\ when the supplied key does not match. As a last resort, delete {ProgramFolder}\Config\ on that node only, then reinstall using node 1's ApplicationKey.

Uninstall or reinstall a node

Uninstall a single web server

In a load-balanced farm, running `UnInstall.bat` on one node removes that node's IIS applications and DirectaService. It preserves `{ProgramFolder}\Config\` on that node and does not remove the shared SQL data.

- Before uninstalling the primary node, save the complete `nsscmd -getconfig` output.
- Uninstalling one web server does not remove shared SQL data.

Reinstall a secondary web server

- **Use existing database.**
- Use the same ApplicationKey as node 1.
- Do not select **Create new databases** on nodes 2 through N.

Related documentation

- NetBackup™ Self Service Installation Guide — prerequisites, GUI installation, upgrade, and Appendix E (customizing image upload).
- `ReadMe.txt` — ApplicationKey backup, central Config\, and manual `Install.ini` configuration.
- Two-server install — remote DatabaseServer rules for a deployment that uses one SQL Server instance for both databases.

Deployment validation checklist

- All web servers can reach the shared SQL Server instance.
- The same portal name is configured on every node.
- The Main and Adapter databases reside on the same SQL Server instance.
- All nodes use the same ApplicationKey.
- Each node retains its own DPAPI-protected `machine.secrets.enc`.
- Secondary nodes use **Use existing database** for the shared Main and Adapter databases.
- DAPI credentials and connection fields are aligned by using the complete `-getconfig` output and `-setconfig` where required.
- All nodes use the same UNC path for uploaded images.
- DirectaService is running on every node.
- The portal works through the load-balancer URL.
- Failover has been tested through the load balancer.
- The latest primary-node `-getconfig` output is stored securely.

See [“Recovering a lost application key”](#) on page 31.

Customizing image upload

This appendix includes the following topics:

- [About Customizing Image Upload](#)

About Customizing Image Upload

Image upload is configured automatically. The uploaded images are stored in `%SYSTEMDRIVE%\inetpub\Cohesity\Images` by default. In a load-balanced installation, all of the web servers need to share any images that users may upload to the system. You must configure the uploaded images to reside on a common network storage area. This section describes how to change the storage location.

To change the storage location

- 1 Go to `install_location\Website\appsettings.json`
- 2 In the **PathForUploadedImages** text box enter the path where any uploaded images are stored. The path can either be a path on the local server, such as `C:\uploadedimages`, or a UNC share, such as `\\myshare\uploadedimages`.

To verify that the image upload works correctly

- 1 Log on to the website as Admin.
- 2 Go to **Admin > Settings > Notice**, then select the **New** icon, and then the **Upload** icon.
- 3 Browse to an image file and upload. If the image is successfully uploaded, it should appear in the image manager dialog box.

Reduced Database Permissions for Database Upgrade

This appendix includes the following topics:

- [Reduced Database Permissions for Database Upgrade](#)

Reduced Database Permissions for Database Upgrade

When you upgrade the database it is necessary to choose a database logon to perform the database upgrade. The simplest choice is to use a user that has the 'sysadmin' role.

If your database administrator (DBA) is unwilling to grant the sysadmin role to you, you can do a database upgrade with a reduced permission set. This appendix describes the upgrade process with reduced permissions.

The following SQL script creates a logon **UpgradeUser** which is suitable for upgrading the database.

To create a reduced permissions user for upgrade

- 1 Run this script in SQL Management Studio, to create a logon and user suitable for upgrading the database
- 2 When you run the configurator and select the database to upgrade, choose:
 - Authentication Mode: **Sql**
 - DB User: **UpgradeUser**

- DB Password: ***password***

3 Once install is complete you can disable or delete the **UpgradeUser**, since it is only used during the upgrade process.

```
-- Create a login for upgrading the database
use master
Create Login UpgradeUser WITH PASSWORD = 'password', Check_Policy = OFF
GO

-- Make a database user for the login
-- and give them db_owner role on the target database
USE NetBackupSelfService
CREATE USER UpgradeUser FOR LOGIN UpgradeUser
GO
ALTER ROLE db_owner ADD MEMBER UpgradeUser
GO

-- Allow ownership of database to be transferred to sa.
-- The sa login can be disabled as per good dba practice,
-- and everything will still work ok.
use master
GRANT IMPERSONATE ON LOGIN::sa to UpgradeUser
```