



# How to store Denodo logs in Azure Blob Storage

Revision 20230324

## NOTE

This document is confidential and proprietary of **Denodo Technologies**.  
No part of this document may be reproduced in any form by any means without prior written authorization of **Denodo Technologies**.

Copyright © 2024  
Denodo Technologies Proprietary and Confidential

## 1 INTRODUCTION

---

By default, the Denodo components store the log files on the local filesystem. However, it is possible to configure them to store the log files on [Azure Blob Storage](#) as well. This is useful for Denodo deployments that run on Azure and that are switched off when they are no longer needed. With this feature, the logging information is saved even if the virtual machines are deleted.

This feature was added in Denodo Platform 8.0 update 20220126 and it has been added for the Solution Manager in Denodo 8.0 update 20220815.

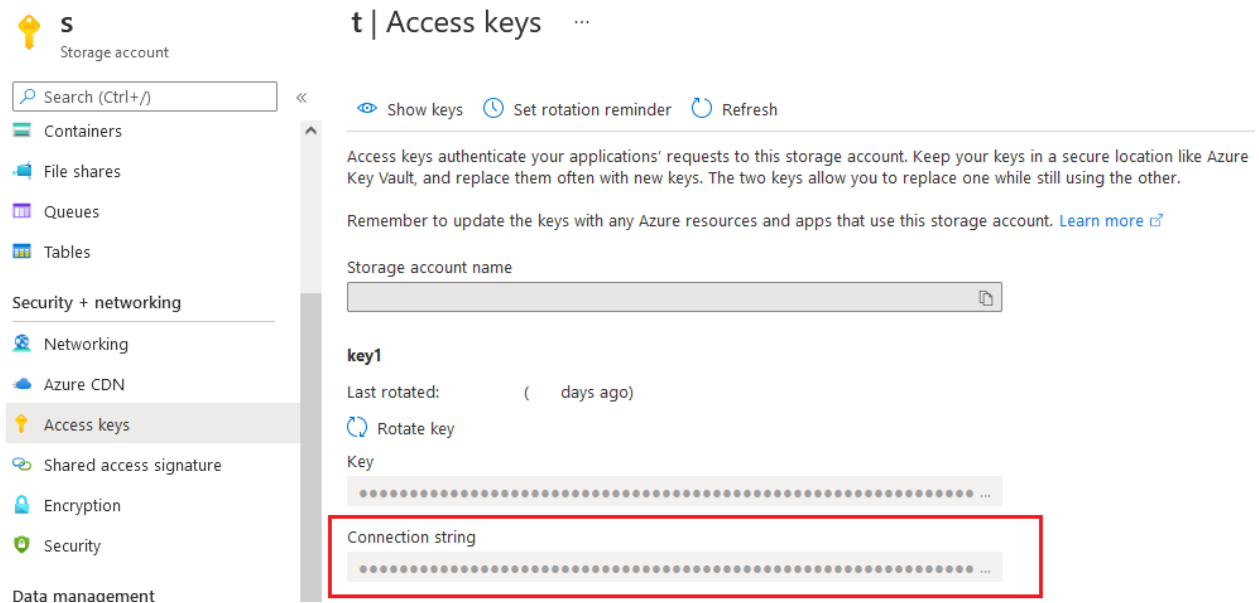
## 2 CONFIGURATION

### 2.1 AUTHENTICATION

To configure the access to Azure Blob Storage, we have to provide the Connection String of the storage account to be used, in order to authenticate the applications' requests to this storage account.

The Microsoft documentation about how to configure Azure Storage connection strings is available [here](#).

In the Azure portal, in the storage account detail page we can access the connection string value through the Access keys option of Security + networking.



The screenshot shows the 'Access keys' page in the Azure portal. On the left, there is a navigation pane with 'Access keys' selected under 'Security + networking'. The main content area shows the 'Access keys' page for a storage account. It includes a search bar, 'Show keys', 'Set rotation reminder', and 'Refresh' buttons. Below this, there is a description of access keys and a link to 'Learn more'. The 'Storage account name' is displayed in a text box. Under 'key1', the 'Last rotated' date is shown as '( days ago)' and there is a 'Rotate key' button. The 'Key' is displayed as a series of dots. The 'Connection string' is also displayed as a series of dots and is highlighted with a red rectangular box.

### 2.2 LOG4J2.XML FILE CONFIGURATION

To store the log files of a Denodo Virtual DataPort server on an S3 bucket you need to modify the file `<DENODO_HOME>/conf/vdp/log4j2.xml`.

Add the CloudAppender inside the "Appenders" section:

```
<CloudAppender name="CloudAppender">
  <PatternLayout pattern="%-4r [%t] %-5p %d{yyyy-MM-dd'T'HH:mm:ss.SSS}
  %C %X - %m %n"/>
  <stagingBufferSize>2500</stagingBufferSize>
  <stagingBufferAge>60</stagingBufferAge>
</CloudAppender>
```

```
<azureBlobContainer>[container_name]</azureBlobContainer>  
<azureBlobNamePrefix>[blob_name_prefix]</azureBlobNamePrefix>  
<azureStorageConnectionString>[access_key_connection_string]</azureStorageConnectionString>  
</CloudAppender>
```

Add the appender to the Root logger:

```
<Root level="error">  
    <AppenderRef ref="FILEOUT" />  
    <AppenderRef ref="CloudAppender" />  
</Root>
```

The following properties control how the logs are stored in Azure Blob Storage:

- **azureBlobContainer**: the container to use. The appender will attempt to create the container if it does not exist.
- **azureBlobNamePrefix**: the prefix where each log file will be stored inside the container.
- **azureStorageConnectionString**: the connection string used to authenticate the applications' requests to this storage account.

These properties control when the appender will send the logs:

- **stagingBufferSize**: number of log messages after which the content is written.
- **stagingBufferAge**: number of minutes after which the content is written. If this property is defined, stagingBufferSize property value is ignored.

NOTE: to log the server start / stop messages, add these log categories in the Loggers section:

```
<Logger name="server.start" level="info" />  
<Logger name="server.stop" level="info" />
```

The log4j2.xml files are available in the following paths in a Denodo Platform installation for each type of server:

- Virtual DataPort server: <DENODO\_HOME>/conf/vdp/log4j2.xml
- Scheduler server: <DENODO\_HOME>/conf/scheduler/log4j2.xml
- Scheduler Index: <DENODO\_HOME>/conf/arn-index/log4j2.xml
- Data Catalog: <DENODO\_HOME>/resources/apache-tomcat/webapps/denodo-data-catalog/WEB-INF/classes/log4j2.xml

In each case, you will need to stop the corresponding server, make the changes in the file and start the server with the new appender configured.