Get started with SQL database auditing

06/07/2017 • 7 minutes to read • Contributors all

In this article

Azure SQL database auditing overview
Define server-level vs. database-level auditing policy
Set up auditing for your database
Analyze audit logs and reports
Production practices
Manage SQL database auditing using Azure PowerShell
Manage SQL database auditing using REST API

Azure SQL database auditing tracks database events and writes them to an audit log in your Azure storage account. Auditing also:

- Helps you maintain regulatory compliance, understand database activity, and gain insight into discrepancies and anomalies that could indicate business concerns or suspected security violations.
- Enables and facilitates adherence to compliance standards, although it doesn't guarantee compliance. For more information about Azure programs that support standards compliance, see the Azure Trust Center.

Azure SQL database auditing overview

You can use SQL database auditing to:

- **Retain** an audit trail of selected events. You can define categories of database actions to be audited.
- **Report** on database activity. You can use preconfigured reports and a dashboard to get started quickly with activity and event reporting.
- **Analyze** reports. You can find suspicious events, unusual activity, and trends.

You can configure auditing for different types of event categories, as explained in the Set up auditing for your database section.

> **Important**
>
> Audit logs are written to **Append Blobs** in an Azure Blob storage on your Azure subscription.
>
> - **Premium Storage** is currently **not supported** by Append Blobs.
> - **Storage in VNet** is currently **not supported**.

Define server-level vs. database-level auditing policy

An auditing policy can be defined for a specific database or as a default server policy:

- A server policy applies to all existing and newly created databases on the server.
- If *server blob auditing is enabled*, it *always applies to the database*. The database will be audited, regardless of the database auditing settings.
- Enabling blob auditing on the database, in addition to enabling it on the server, does *not* override or change any of the settings of the server blob auditing. Both audits will exist side by side. In other words, the database is audited twice in parallel; once by the server policy and once by the database policy.

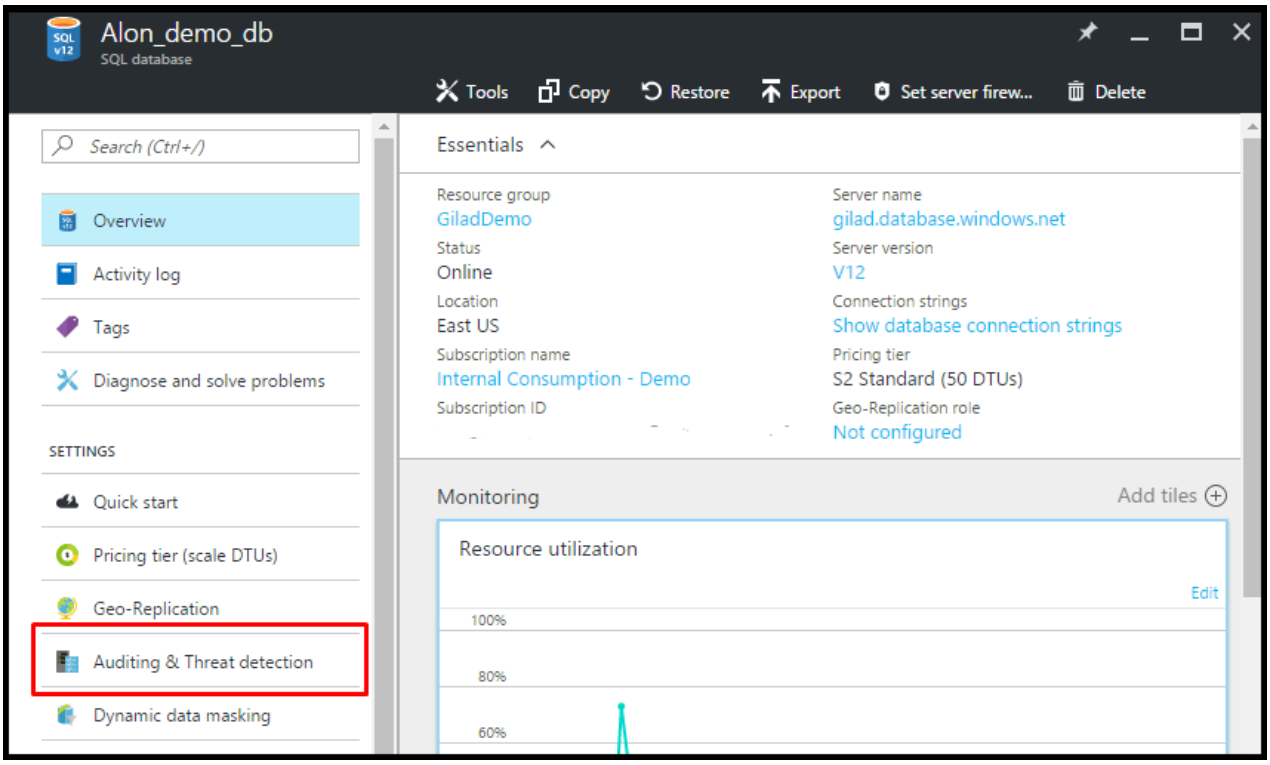
  > **Note**
  >
  > You should avoid enabling both server blob auditing and database blob auditing together, unless:
  >
  > - You want to use a different *storage account* or *retention period* for a specific database.
  > - You want to audit event types or categories for a specific database that differ from the rest of the databases on the server. For example, you might have table inserts that need to be audited only for a specific database.
  >
  > Otherwise, we recommended that you enable only server-level blob auditing and leave the database-level auditing disabled for all databases.

Set up auditing for your database

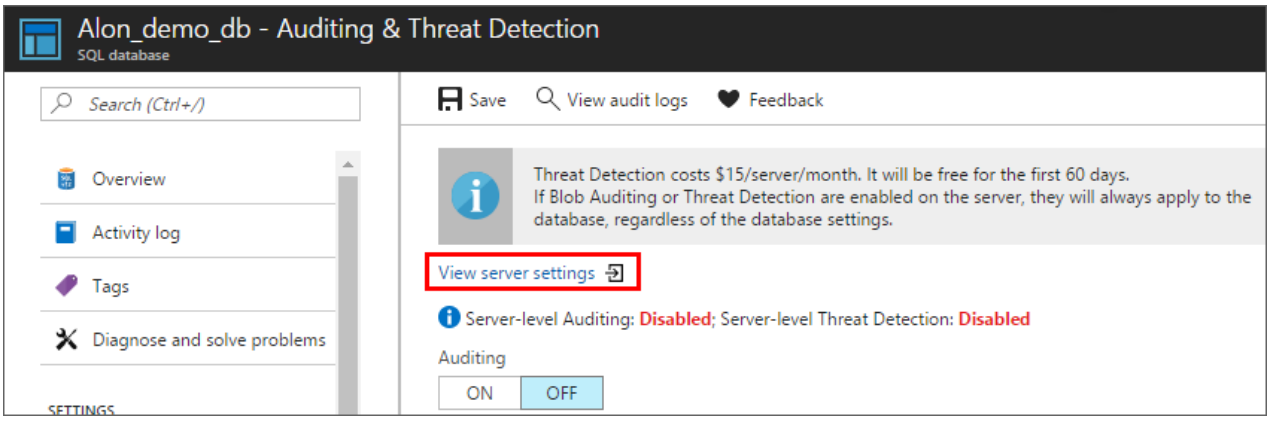
The following section describes the configuration of auditing using the Azure portal.

1. Go to the Azure portal.

2. Go to the **Settings** blade of the SQL database/SQL server you want to audit. In the **Settings** blade, select **Auditing & Threat detection**.

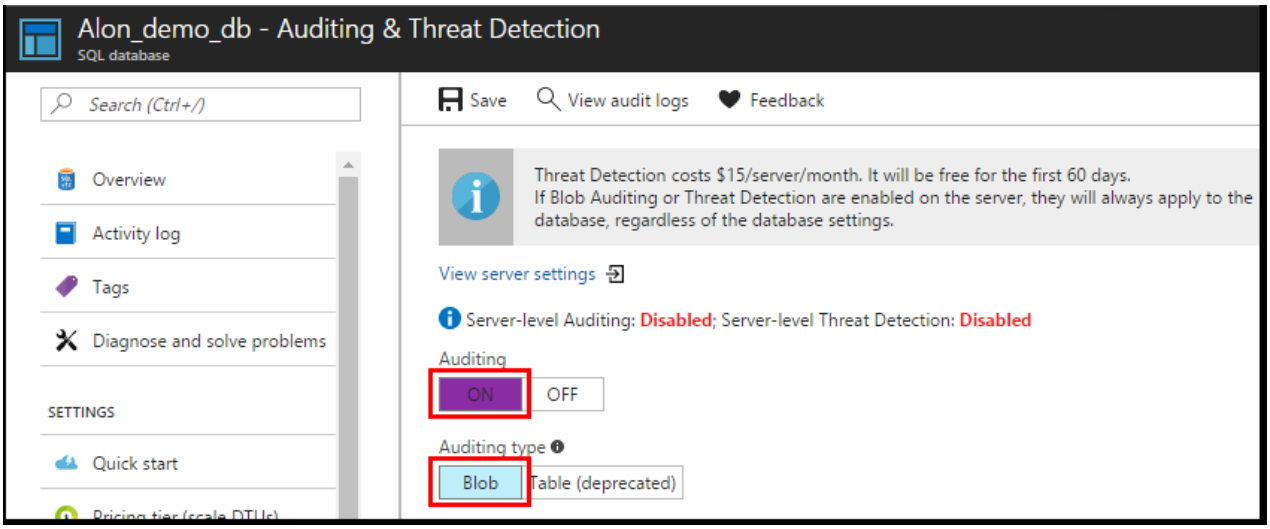


3. If you prefer to set up a server auditing policy, you can select the **View server settings** link in the database auditing blade. You can then view or modify the server auditing settings. Server auditing policies apply to all existing and newly created databases on this server.



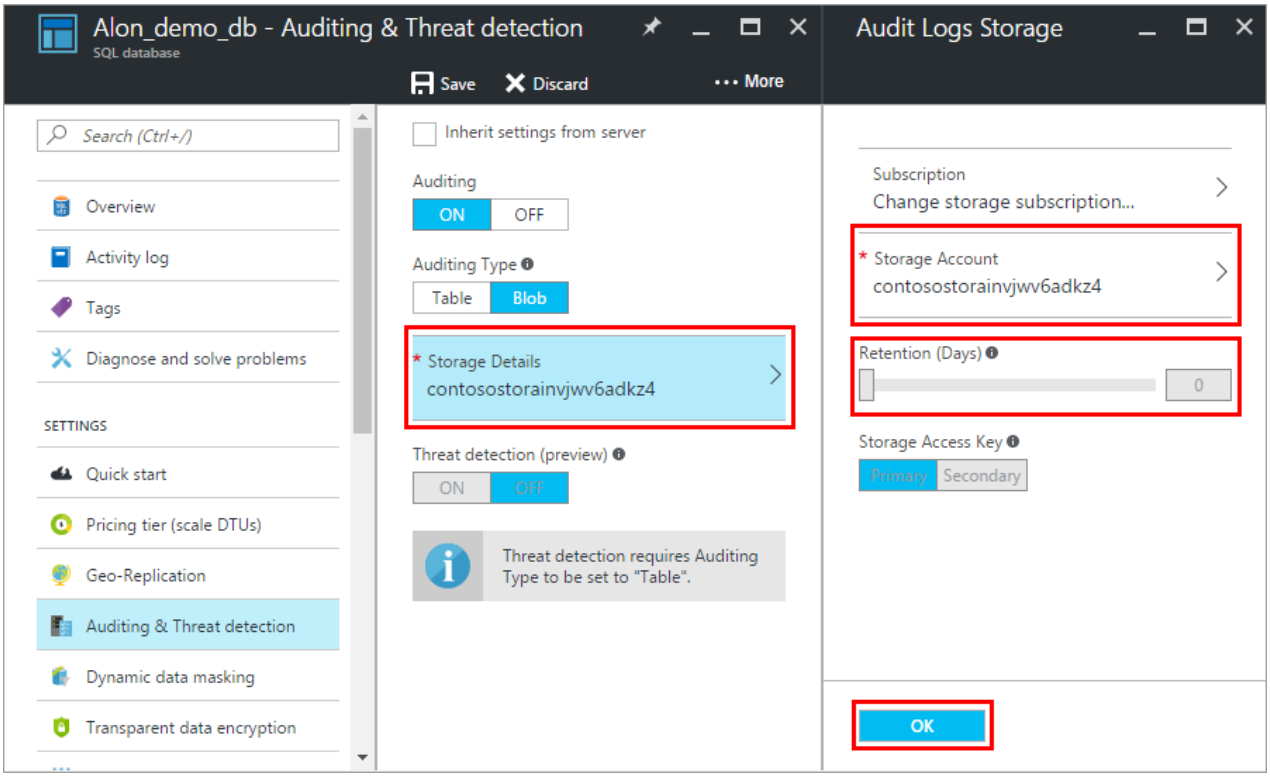
4. If you prefer to enable blob auditing on the database level, for **Auditing**, select **ON**, and for **Auditing type**, select **Blob**.

   If server blob auditing is enabled, the database-configured audit will exist side by side with the server blob audit.



5. To open the **Audit Logs Storage** blade, select **Storage Details**. Select the Azure storage account where logs will be saved, and then select the retention period. The old logs will be deleted. Then click **OK**.

> **Tip**
>
> To get the most out of the auditing reports templates, use the same storage account for all audited databases.



6. If you want to customize the audited events, you can do this via PowerShell or the REST API.
7. After you've configured your auditing settings, you can turn on the new threat detection feature and configure emails to receive security alerts. When you use threat detection, you receive proactive

alerts on anomalous database activities that can indicate potential security threats. For more information, see Getting started with threat detection.

8. Click **Save**.

Analyze audit logs and reports

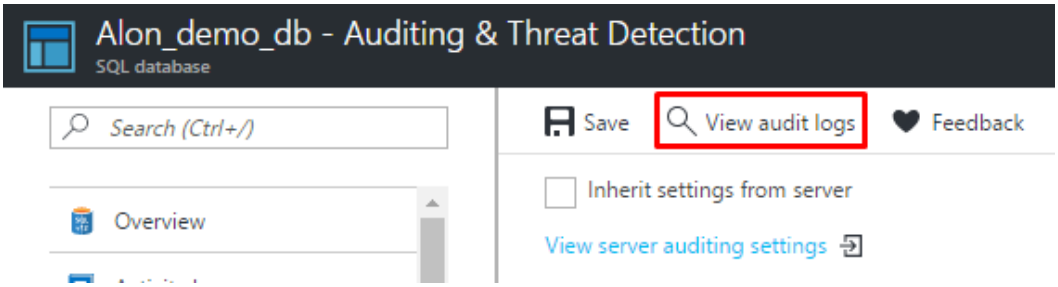
Audit logs are aggregated in the Azure storage account you chose during setup. You can explore audit logs by using a tool such as Azure Storage Explorer.

Blob auditing logs are saved as a collection of blob files within a container named **sqldbauditlogs**.

For further details about the hierarchy of the storage folder, naming conventions, and log format, see the Blob Audit Log Format Reference.

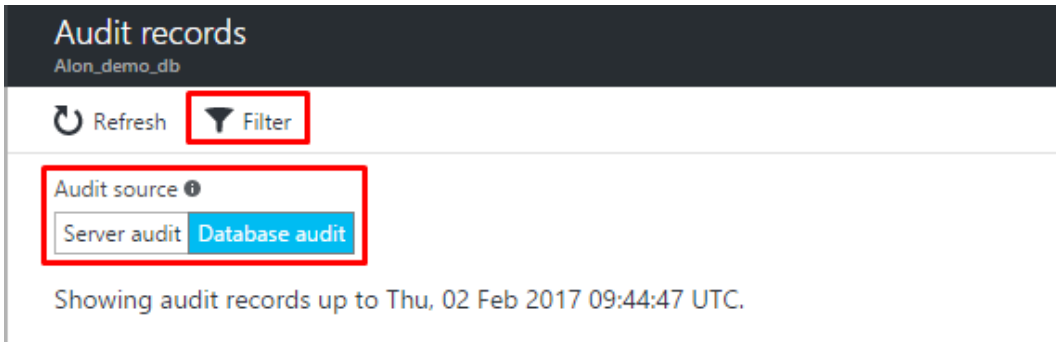
There are several methods you can use to view blob auditing logs:

- Use the Azure portal. Open the relevant database. At the top of the database's **Auditing & Threat detection** blade, click **View audit logs**.



  An **Audit records** blade opens, from which you'll be able to view the logs.

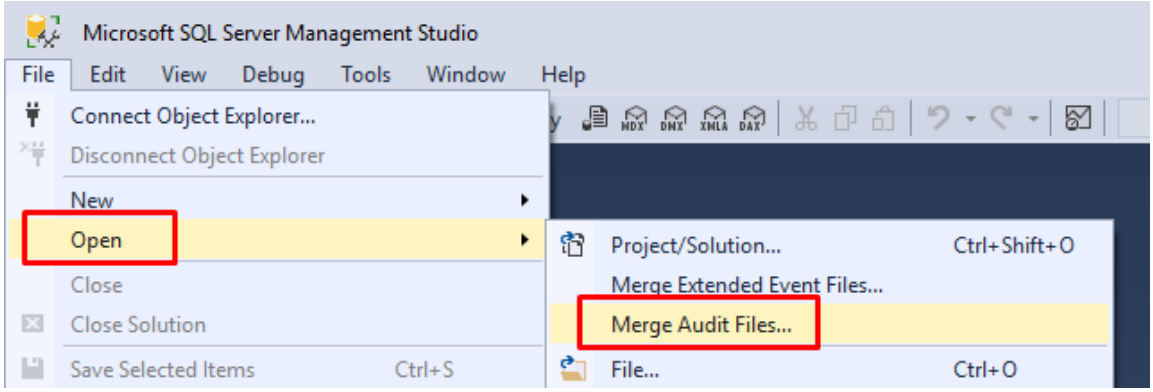
  - You can view specific dates by clicking **Filter** at the top of the **Audit records** blade.
  - You can switch between audit records that were created by a server policy or database policy audit.



- Use the system function **sys.fn_get_audit_file** (T-SQL) to return the audit log data in tabular format. For more information on using this function, see the sys.fn_get_audit_file documentation.

- Use **Merge Audit Files** in SQL Server Management Studio (starting with SSMS 17):

  1. From the SSMS menu, select **File** > **Open** > **Merge Audit Files**.



  2. The **Add Audit Files** dialog box opens. Select one of the **Add** options to choose whether to merge audit files from a local disk or import them from Azure Storage. You are required to provide your Azure Storage details and account key.

  3. After all files to merge have been added, click **OK** to complete the merge operation.

  4. The merged file opens in SSMS, where you can view and analyze it, as well as export it to an XEL or CSV file or to a table.

- Use the sync application that we have created. It runs in Azure and utilizes Operations Management Suite (OMS) Log Analytics public APIs to push SQL audit logs into OMS. The sync application pushes SQL audit logs into OMS Log Analytics for consumption via the OMS Log Analytics dashboard.

- Use Power BI. You can view and analyze audit log data in Power BI. Learn more about Power BI, and access a downloadable template.

- Download log files from your Azure Storage blob container via the portal or by using a tool such as Azure Storage Explorer.

  - After you have downloaded a log file locally, you can double-click the file to open, view, and analyze the logs in SSMS.
  - You can also download multiple files simultaneously via Azure Storage Explorer. Right-click a specific subfolder and select **Save as** to save in a local folder.

- Additional methods:

  - After downloading several files or a subfolder that contains log files, you can merge them locally as described in the SSMS Merge Audit Files instructions described earlier.

  - View blob auditing logs programmatically:

    - Use the Extended Events Reader C# library.

- Query Extended Events Files by using PowerShell.

Production practices

Auditing geo-replicated databases

With geo-replicated databases, when you enable auditing on the primary database the secondary database will have an identical auditing policy. It is also possible to set up auditing on the secondary database by enabling auditing on the **secondary server**, independently from the primary database.

- Server-level (**recommended**): Turn on auditing on both the **primary server** as well as the **secondary server** - the primary and secondary databases will each be audited independently based on their respective server-level policy.
- Database-level: Database-level auditing for secondary databases can only be configured from Primary database auditing settings.
  - Blob auditing must be enabled on the *primary database itself*, not the server.
  - After blob auditing is enabled on the primary database, it will also become enabled on the secondary database.

> **Important**
>
> With database-level auditing, the storage settings for the secondary database will be identical to those of the primary database, causing cross-regional traffic. We recommend that you enable only server-level auditing, and leave the database-level auditing disabled for all databases.

Storage key regeneration

In production, you are likely to refresh your storage keys periodically. When refreshing your keys, you need to resave the auditing policy. The process is as follows:

1. Open the **Storage Details** blade. In the **Storage Access Key** box, select **Secondary**, and click **OK**. Then click **Save** at the top of the auditing configuration blade.