Use Azure App Service Certificate with Azure Front Door

07/21/2025

Microsoft Azure Front Door (Standard and Premium) is a modern global load balancer and application delivery network that supports custom TLS certificates through Azure Key Vault. This article discusses how to use an Azure App Service Certificate securely together with Microsoft Azure Front Door by using managed identities and Bring Your Own Certificate (BYOC) support. This integration enables you to deliver encrypted traffic that has automatic renewal, enterprise-grade performance, and global scale.

Overview

Azure App Service Certificates provide a simple, integrated way to purchase, provision, and manage SSL/TLS certificates. These certificates are issued by trusted Certificate Authorities (such as DigiCert) and work together with App Services. They can also be extended to secure traffic that's routed through Azure Front Door.

To purchase a certificate, see Buy and configure an App Service Certificate.

> **Important**
>
> After you purchase a certificate, you must manually complete the **Store** step in the **Certificate Configuration** blade to import the certificate into Azure Key Vault. This step is required before the certificate can be used together with other Azure services.

Step 1: Enable managed identity on Azure Front Door

A managed identity enables Azure Front Door to securely retrieve the certificate from Azure Key Vault:

1. Navigate to your Azure Front Door profile.
2. Under **Security**, select **Identity**, and then enable a managed identity:
   - **System-assigned** (Recommended): Tied to the Front Door lifecycle
   - **User-assigned** (Optional): For reuse across multiple services
3. Select **Save**.

For more information, see Use managed identities to access Azure Key Vault certificates.

Step 2: Configure Key Vault Access for Front Door

Grant permission to Azure Front Door to access the certificate by using one of the following methods:

Method A: Azure RBAC (recommended)

1. Open **Key Vault** > **Access control (IAM)** > **+ Add** > **Add role assignment**.
2. Assign the **Key Vault Secrets User** role.
3. Select **Managed identity**, then select the system-assigned identity of Azure Front Door.
4. Select **Review + assign**.

```bash
    az role assignment create \
      --assignee-object-id <frontdoor-identity-object-id> \
      --role "Key Vault Secrets User" \
      --scope "/subscriptions/<sub-id>/resourceGroups/<rg>/providers/Microsoft.KeyVault/vaults/<vault-name>"
```

To retrieve the identity object ID:

```bash
az front-door show \
  --name <frontdoor-name> \
  --resource-group <rg> \
  --query identity.principalId -o tsv
```

> **Note**
>
> Make sure that the Key Vault firewall allows trusted services or specific Front Door IP ranges.

Method B: Key Vault Access Policy

1. Navigate to your key vault > **Access policies**.
2. Select **+ Add Access Policy**.
3. Grant **Get** and **List** permissions for **Secrets** and **Certificates**.
4. Assign the policy to the managed identity for Azure Front Door.

5. Save the access policy.

> ⓘ **Note**
>
> This method is suitable for legacy scenarios or if RBAC isn't enabled.

Step 3: Add certificate as a secret in Azure Front Door

Before you do this step, make sure that the App Service Certificate is successfully stored in Azure Key Vault through the App Service Certificate blade. For more information, see Buy and configure an App Service Certificate.

To add the certificate:

1. Go to your Azure Front Door (Standard/Premium) profile.
2. Under **Security**, select **Secrets** > **+ Add**.
3. Select your key vault, and then select the stored App Service Certificate.
4. Select the version. (Use `Latest` to enable automatic certificate rotation.)
5. Select **Add**.

> ⓘ **Note**
>
> Azure Front Door supports automatic certificate renewal when you reference the `Latest` version. Updates in Key Vault are reflected in Front Door within 72 hours. For more information, see Renew customer-managed TLS certificates.

> ⓘ **Important**
>
> Certificates must be stored in a Key Vault within the same subscription and must include a complete certificate chain that uses supported algorithms. For more information, see Use your own certificate with Azure Front Door.

Step 4: Configure a custom domain with BYOC

1. In your Front Door profile, go to **Domains** > **+ Add**.
2. Provide the domain details:

   - **Custom domain**: for example, `www.contoso.com`
   - **DNS zone**: Choose Azure DNS, if applicable.
   - **DNS management**: Azure-managed (recommended) or external

3. Verify domain ownership:

    - Use **TXT record** if you use custom DNS provider

4. Under **HTTPS Configuration**:

    - **Certificate type**: `Bring Your Own Certificate (BYOC)`
    - **Secret**: Select the secret that you added in Step 3 (for example, `certname-latest`).
    - **TLS policy**: Select a supported policy (for example, `TLS 1.2_2023`)

5. Select **Add** to finish the setup.

After verification is made, Front Door serves traffic securely by using the certificate from Azure Key Vault. For more information, see Add a custom domain in Azure Front Door.

Summary

Expand table

Task	Tool	Notes
Enable identity	Azure portal or CLI	System-assigned identity is recommended
Grant access	IAM Role or Access Policy	Use `Key Vault Secrets User` or equivalent
Add secret	Azure portal	Reference `-latest` to enable autorotation
Bind domain	Azure portal	Validate domain and configure HTTPS

References

- Configure HTTPS custom domain (Front Door)
- Add custom domain in Front Door
- Azure Front Door managed identity access
- Buy and configure an App Service Certificate

Use Azure App Service Certificate with Application Gateway

07/21/2025

Microsoft Azure provides various tools and services to secure your web applications by using SSL/TLS certificates. One such offering, the **Azure App Service Certificate**, is tightly integrated with Azure App Services. However, many organizations use **Azure Application Gateway** as a reverse proxy, load balancer, and Web Application Firewall (WAF). Understandably, such organizations want to use the same certificate across all services.

This article provides a comprehensive guide for using App Service Certificates in Application Gateway, including usage steps, restrictions, and best practices. By understanding the limitations and using the Azure Key Vault service effectively, you can build a robust certificate management workflow across both App Services and Application Gateway.

About App Service Certificate

Azure App Service Certificate is a first-party SSL certificate that's issued by DigiCert or GoDaddy and is designed for use together with Azure App Services. The certificate is stored securely in Azure Key Vault and supports autorenewal if it's integrated correctly.

Key Characteristics:

- Domain-validated SSL certificate
- Designed primarily for Azure App Services
- Stored in a key vault for secure usage
- Autorenewal supported if linked correctly

However, App Service Certificates aren't directly usable in Application Gateway unless you take additional steps.

How to use App Service Certificate in Application Gateway

You can use App Service Certificate in Azure Application Gateway, but not directly. Application Gateway requires a certificate in `.pfx` format (having a private key) to configure HTTPS listeners. App Service Certificates aren't exposed as downloadable PFXs by default. Therefore, you have to follow specific steps to extract and configure them.

Option 1: Manual export and upload

1. **Purchase and configure the certificate**: Buy and verify an App Service Certificate through Azure App Service.
2. **Import into Key Vault**: Navigate to the App Service Certificate resource. Then, use the **Key Vault** blade to store the certificate in a key vault of your choice.
3. **Export as .pfx from Key Vault**: Use Azure PowerShell or Azure CLI to download the certificate as a `.pfx` file that has a private key.
    - Example that uses Azure CLI:

```
Bash

    az keyvault secret download \
    --vault-name `YourKeyVaultName` \
    --name `CertificateName` \
    --file cert.pfx \
    --encoding base64
```

4. **Upload to Application Gateway**: Go to Application Gateway > Listeners > + Add Listener. Select **HTTPS**, upload the `.pfx` file, and then enter the password.
5. **Associate with a rule**: Create a routing rule, and link it to the HTTPS listener. For detailed steps, see Create a routing rule in Application Gateway

Option 2: Use Key Vault reference (recommended)

1. **Store App Service Certificate in Key Vault**: Navigate to the App Service Certificate resource. Then, use the **Key Vault** blade to store the certificate in a key vault of your choice.
2. **Enable Managed Identity for Application Gateway**: Enable user-assigned or system-assigned managed identity.
3. **Grant Access to key vault**: In the key vault, go to **Access Policies**, and add a policy for Application Gateway identity that has `get`, `list`, and `secret management` permissions.
4. **Reference Certificate from Key Vault**: Go to **Application Gateway** > **Listeners** > **+ Add Listener**, select **HTTPS**, and then select **Key Vault certificate**.

> ⓘ Note

Currently, Key Vault integration supports only certificates that have the private key in `.pfx` format.

Limitations and considerations

1. **Direct use not supported:**
    - You can't bind an App Service Certificate to Application Gateway directly in the same manner as you can for App Services.
2. **Export required for manual use:**
    - You must extract the `.pfx` format from Key Vault before you can use it in Application Gateway (if you're not using a Key Vault reference).
3. **Autorenewal challenges:**
    - App Service Certificates support autorenewal only for App Services.
    - When used in Application Gateway, autorenewal doesn't automatically propagate.
    - You must manually update the certificate in Application Gateway after you renew it.
    - We recommend that you use **Azure Automation** or **Logic App** to automate this update process. See Renew certificates in Application Gateway.
4. **Certificate format restrictions:**
    - Application Gateway accepts only `.pfx` files.
    - Application Gateway rejects `.cer` and `.pem` files.
    - Self-signed certificates are supported but must be uploaded as `.pfx`.
    - See Self-signed certificates for Application Gateway.

Best practices

- Use Key Vault-based integration for better security and easier management.
- Automate certificate renewal by using scripts or Azure Automation.
- Regularly audit access policies in Key Vault.
- Keep secure backup copies of your exported `.pfx` files.

Summary

Expand table

Feature	App Service	Application Gateway
Certificate Format	Managed by platform	Requires `.pfx`
Autorenewal	Supported	Manual (requires automation)
Key Vault Integration	Built in	Supported (requires setup)
Direct Use of App Service Certificate	✅ App Service only	❌ Not supported

Useful links

- Renew certificates in Application Gateway
- SSL certificates overview - Application Gateway
- Use self-signed certificates in Application Gateway
- Configure App Service Certificate
- Create a routing rule in Application Gateway

Third-party information disclaimer

The third-party products that this article discusses are manufactured by companies that are independent of Microsoft. Microsoft makes no warranty, implied or otherwise, about the performance or reliability of these products.

Azure App Service troubleshooting documentation

Welcome to Azure App Service troubleshooting. These articles explain how to determine, diagnose, and fix issues that you might encounter when you use Azure Monitor. In the navigation pane on the left, browse through the article list or use the search box to find issues and solutions.

Troubleshooting web apps

HOW-TO GUIDE

FAQ about creating or deleting web apps

Capture memory dumps on the Azure App Service platform

Article • 06/03/2024

This article provides guidance about Microsoft Azure App Service debugging features for capturing memory dumps. The capture method that you use is dictated by the scenario in which you capture a memory dump for troubleshooting a performance or availability issue. For example, capturing a memory dump is different for a process that's experiencing excessive memory consumption than for a process that's throwing exceptions or responding slowly. The process in this context is the Internet Information Services (IIS) worker process (W3WP, which runs as *w3wp.exe*).

Mapping memory dump scenarios to Azure App Service debugging features

The following table provides recommendations about the commands that each App Service feature runs to generate a memory dump. There are so many approaches to capturing a memory dump that the process might be confusing. If you're already proficient in capturing a W3WP memory dump, this information isn't intended to change your approach. Instead, we hope to provide guidance for inexperienced users who haven't yet developed a preference.

Expand table

Scenario	Azure App Service debugging feature	Command
Unresponsive or slow	Auto-heal (request duration)	`procdump -accepteula -r -dc "Message" -ma <PID> <PATH>`
Crash (process termination)	Crash monitoring	Uses DbgHost to capture a memory dump
Crash (handled exceptions)	Traces in Application Insights/Log Analytics	None
Crash (unhandled exceptions)	Application Insights Snapshot Debugger	None
Excessive CPU usage	Proactive CPU monitoring	`procdump -accepteula -dc "Message" -ma <PID> <PATH>`

Scenario	Azure App Service debugging feature	Command
Excessive memory consumption	Auto-heal (memory limit)	`procdump -accepteula -r -dc "Message" -ma <PID> <PATH>`

> ⓘ Note
>
> We have a secondary recommendation for capturing a W3WP process memory dump in the unresponsive or slow scenario. If that scenario is reproducible, and you want to capture the dump immediately, you can use the Collect a Memory dump diagnostic tool. This tool is located in the **Diagnose and solve problems** toolset page for the given App Service Web App in the Azure portal. Another location to check for general exceptions and poor performance is on the **Application Event Logs** page. (You can access Application logs also from the **Diagnose and solve problems** page.) We discuss all the available methods in the "Expanded Azure App Service debugging feature descriptions" section.

Expanded process scenario descriptions

This section contains detailed descriptions of the six scenarios that are shown in the previous table.

Unresponsive or slow scenario

When a request is made to a web server, some code must usually be run. The code execution occurs within the *w3wp.exe* process on threads. Each thread has a stack that shows what's currently running.

An unresponsive scenario can be either permanent (and likely to time out) or slow. Therefore, the unresponsive scenario is one in which a request takes longer than expected to run. What you might consider being slow depends on what the code is doing. For some people, a three-second delay is slow. For others, a 15-second delay is acceptable. Basically, if you see performance metrics that indicate slowness, or a super user states that the server is responding slower than normal, then you have an unresponsive or slow scenario.

Crash (process termination) scenario

Over many years, Microsoft .NET Framework has improved the handling of exceptions. In the current version of .NET, the exception handling experience is even better.

Historically, if a developer didn't place code snippets within a try-catch block, and an exception was thrown, the process terminated. In that case, an unhandled exception in the developer's code terminated the process. More modern versions of .NET handle some of these “unhandled” exceptions so that the process that's running the code doesn't crash. However, not all unhandled exceptions are thrown directly from the custom code. For example, access violations (such as 0xC0000005 and 0x80070005) or a stack overflow can terminate the process.

Crash (handled exceptions) scenario

Although a software developer takes special care to determine all possible scenarios under which the code runs, something unexpected can occur. The following errors can trigger an exception:

- Unexpected null values
- Invalid casting
- A missing instantiated object

It's a best practice to put code execution into try-catch code blocks. If a developer uses these blocks, the code has an opportunity to fail gracefully by specifically managing what follows the unexpected event. A handled exception is an exception that is thrown inside a try block and is caught in the corresponding catch block. In this case, the developer anticipated that an exception could occur and coded an appropriate try-catch block around that section of code.

In the catch block, it's useful to capture enough information into a logging source so that the issue can be reproduced and, ultimately, resolved. Exceptions are expensive code paths in terms of performance. Therefore, having many exceptions affects performance.

Crash (unhandled exceptions) scenario

Unhandled exceptions occur when code tries to take an action that it doesn't expect to encounter. As in the Crash (process termination) scenario, that code isn't contained within a try-catch code block. In this case, the developer didn't anticipate that an exception could occur in that section of code.

This scenario differs from the previous two exception scenarios. In the Crash (unhandled exceptions) scenario, the code in question is the code that the developer wrote. It isn't

the framework code that's throwing the exception, and it isn't one of the unhandled exceptions that kills the *w3wp.exe* process. Also, because the code that's throwing an exception isn't within a try-catch block, there's no opportunity to handle the exception gracefully. Troubleshooting the code is initially a bit more complex. Your goal would be to find the exception text, type, and stack that identifies the method that's throwing this unhandled exception. That information enables you to identify where you have to add the try-catch code block. Then, the developer can add the similar logic to log exception details that should exist in the Crash (unhandled exceptions) scenario.

Excessive CPU usage scenario

What's excessive CPU usage? This situation is dependent on what the code does. In general, if the CPU usage from the *w3wp.exe* process is 80 percent, then your application is in a critical situation that can cause various symptoms. Some possible symptoms are:

- Slowness
- Errors
- Other undefined behavior

Even a 20-percent CPU usage can be considered excessive if the web site is just delivering static HTML files. Post-mortem troubleshooting of an excessive CPU spike by generating a memory dump probably won't help you to determine the specific method that's using it. The best that you can do is to determine which requests were likely taking the longest time, and then try to reproduce the issue by testing the identified method. That procedure assumes that you don't run performance monitors on the performance systems that captured that burst. In many cases, you can cause performance issues by having monitors constantly run in real time.

Excessive memory consumption scenario

If an application is running in a 32-bit process, excessive memory consumption can be a problem. Even a small amount of activity can consume the 2-3 GB of allocated virtual address space. A 32-bit process can never exceed a total of 4 GB, regardless of the amount of physical memory that's available.

A 64-bit process is allocated more memory than a 32-bit process. It's more likely that the 64-bit process will consume the amount of physical memory on the server than that the process will consume its allocated virtual address space.

Therefore, what constitutes an excessive memory consumption issue depends on the following factors:

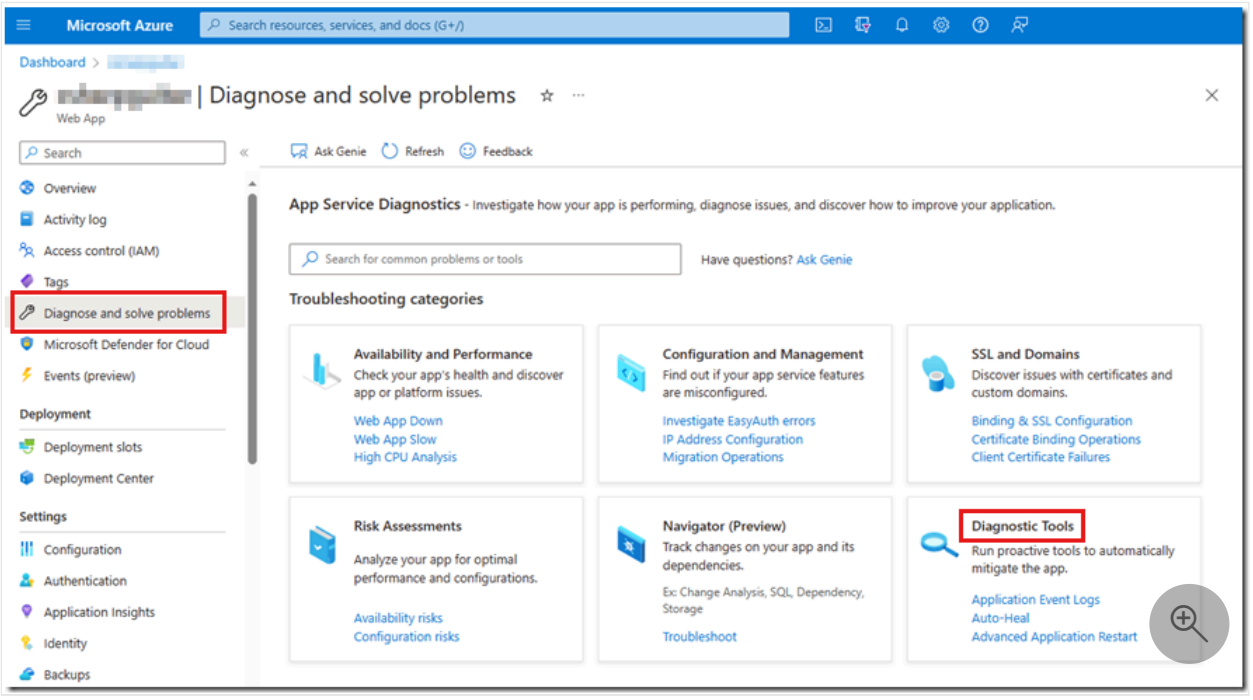
- Process bitness (32-bit or 64-bit)
- The amount of memory usage that's considered to be "normal."

If your process is consuming more memory than expected, collect a memory dump for analysis to determine what is consuming memory resources. For more information, see Create a memory dump of your App Service when it consumes too much memory.

Now that you have a bit more context about the different process scenarios that a memory dump can help you to troubleshoot, we'll discuss the recommended tool for capturing memory dumps on the Azure App Service platform.

Expanded Azure App Service debugging feature descriptions

In the table in the "Mapping memory dump scenarios to Azure App Service debugging features" section, we identified six debugging features that are targeted at collecting memory dumps. Each of these features is accessible from within the Azure portal on the **Diagnose and solve problems** page when you select the **Diagnostic Tools** tile.



In the following sections, we discuss each of these debugging features in more detail.

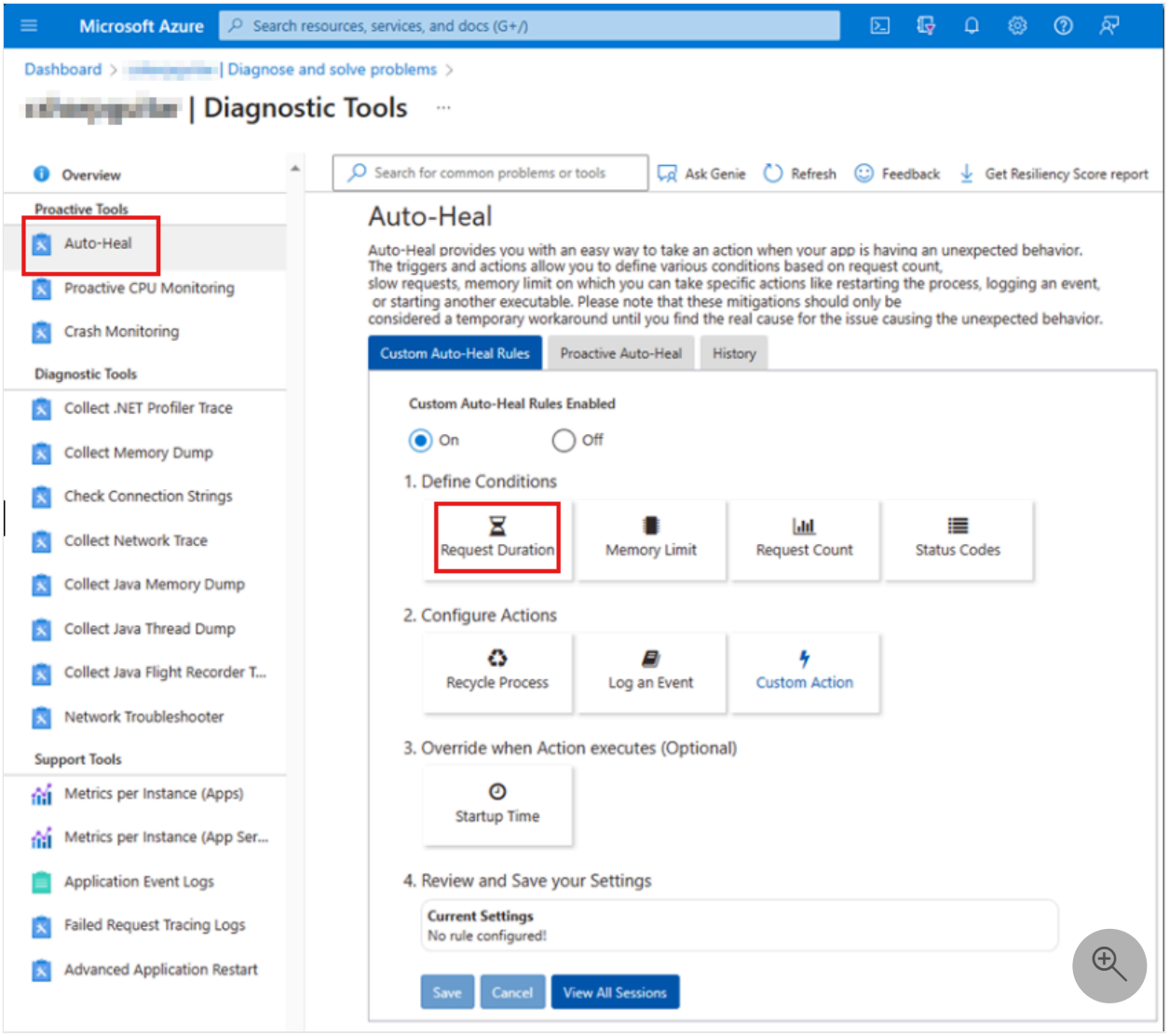
Auto-heal (request duration) feature

The auto-heal (request duration) feature is useful for capturing a memory dump if the response is taking longer than expected to finish. You can see the link to **Auto-Heal** in the **Diagnostic Tools** tile in the previous screenshot. Select that link to go directly to the

feature, or select the **Diagnostic Tools** tile to review all the available tools on the **Diagnostic Tools** page. For information about how to configure this feature, see the following articles:

- Announcing the New Auto Healing Experience in App Service Diagnostics
- Announcing Auto Heal for Linux
- Collect and automate diagnostic actions with Azure App Services

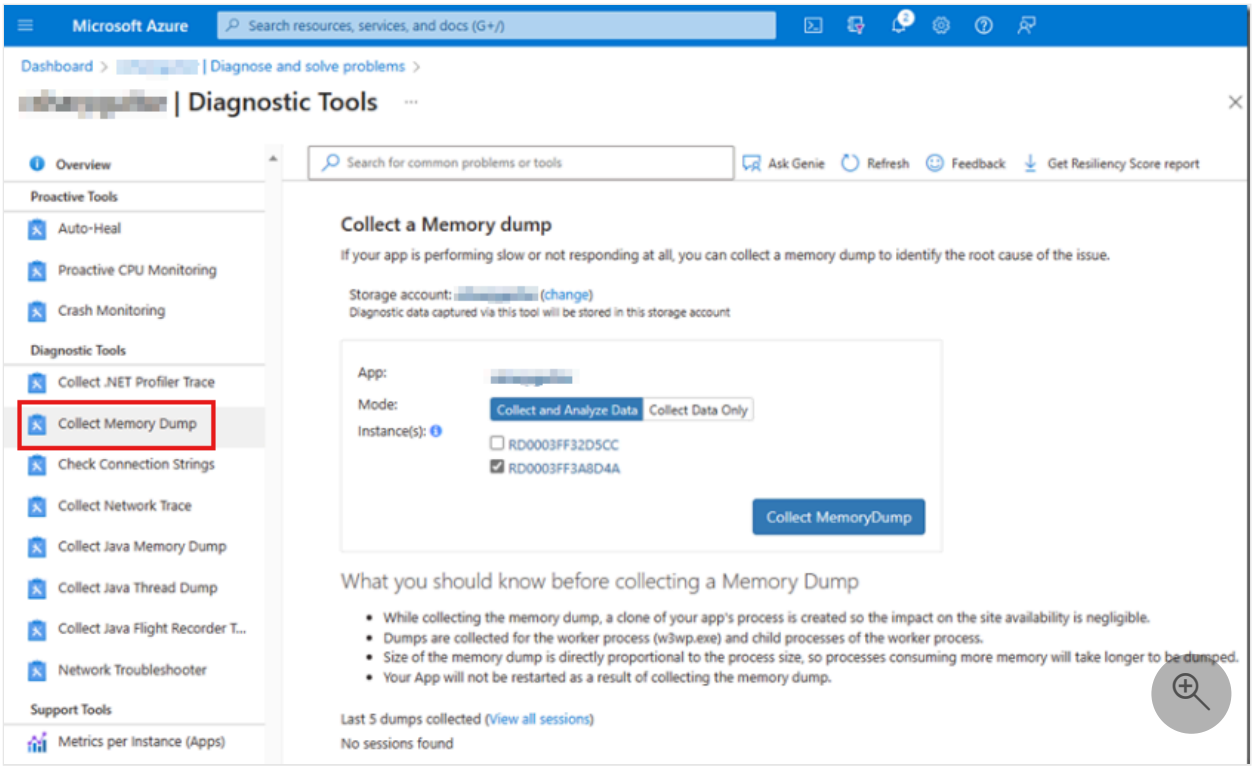
The auto-heal feature is shown in the following screenshot.



Another feature that's named "Collect a Memory dump" is useful in this scenario when the issue is currently occurring or reproducible. This feature quickly collects a memory dump on manual demand.

Collect a memory dump feature

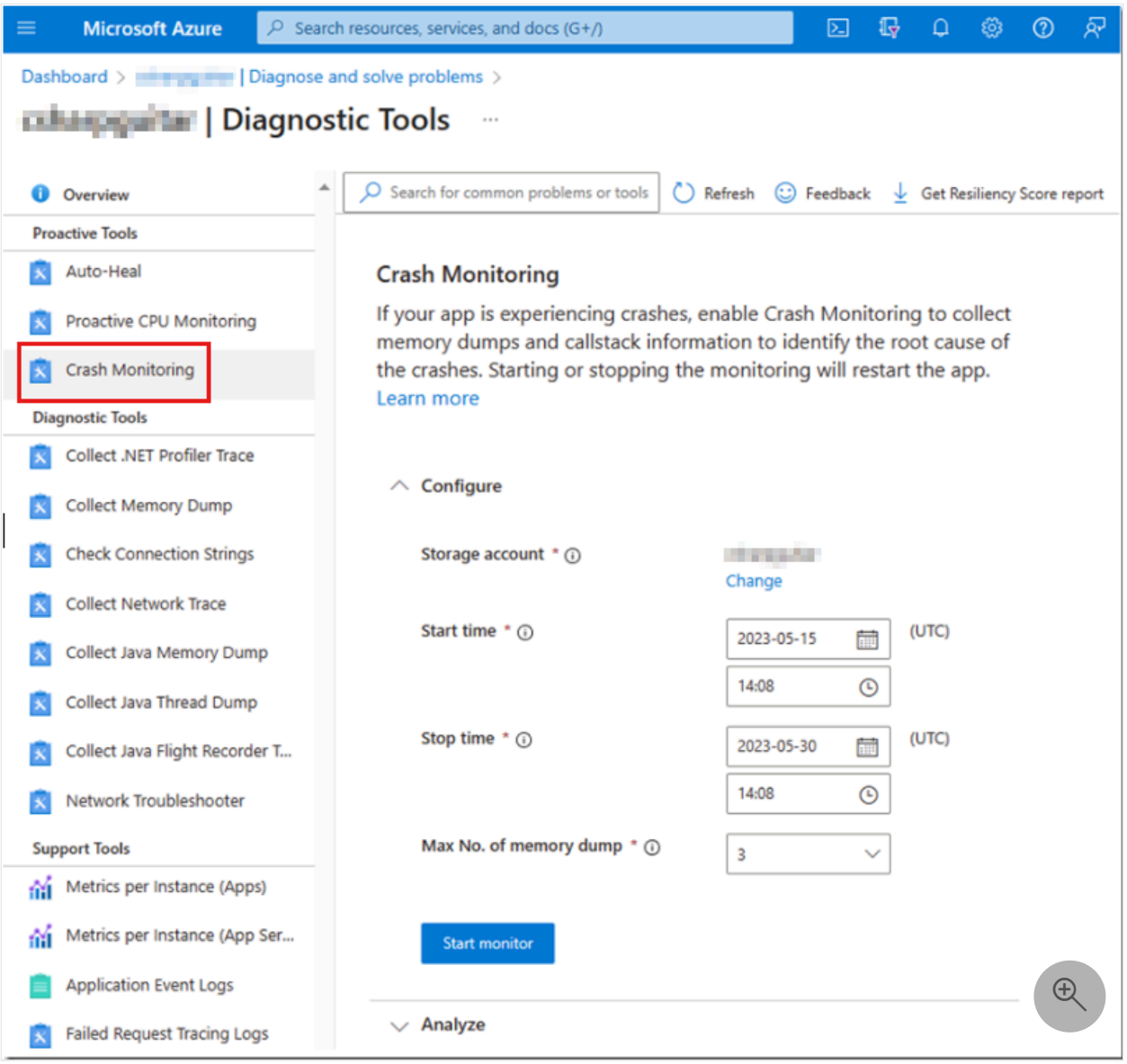
This approach requires manual intervention. The following screenshot shows the **Collect a Memory dump** page.



To use the feature, select a storage account in which to store the memory dump. Then, select which server instance you want to collect the memory dump from. If you have more than a single instance, make sure that the issue that you're debugging is occurring on that instance. Notice that a restart might not be optimal on a production application that's in operation.

Crash Monitoring feature

The Crash Monitoring feature is useful for capturing a memory dump if an unhandled exception causes the W3WP process to terminate. The following screenshot shows the **Crash Monitoring** page in **Diagnostic Tools**:



To view a guided walk-through about how to configure the crash monitoring feature in Azure App Service, see Crash monitoring in Azure App Service .

Traces in Application Insights/Log Analytics feature

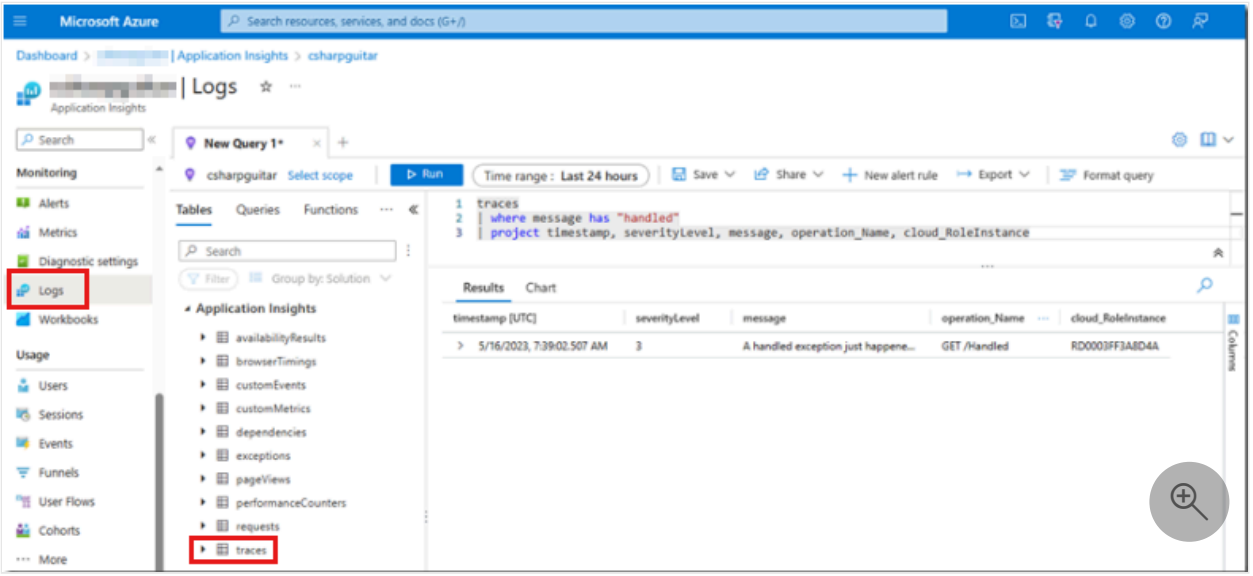
A handled exception is a scenario in which the code that's contained within a try-catch block tries to take an action that's unexpected or unsupported. For example, the following code snippet tries to divide a number by zero even though this is an illegal operation:

C#

```
decimal percentage = 0, number = 1000, total = 0;
try
{
  percentage = number / total;
}
catch (DivideByZeroException divEx)
{
```

```
  _logger.LogError("A handled exception just happened: -> {divEx.Message}",
divEx.Message);
}
```

This code snippet causes a divide-by-zero exception that's handled because the unsupported mathematical operation is placed within a try-catch block. Application Insights doesn't log handled exceptions unless you intentionally include the Microsoft.ApplicationInsights NuGet package in your application code, and then add the code to log the information. If the exception occurs after you add the code, you can view the entry in Log Analytics, as shown in the following screenshot.



The following Kusto code contains the query that's used to extract the data from Log Analytics:

Kusto

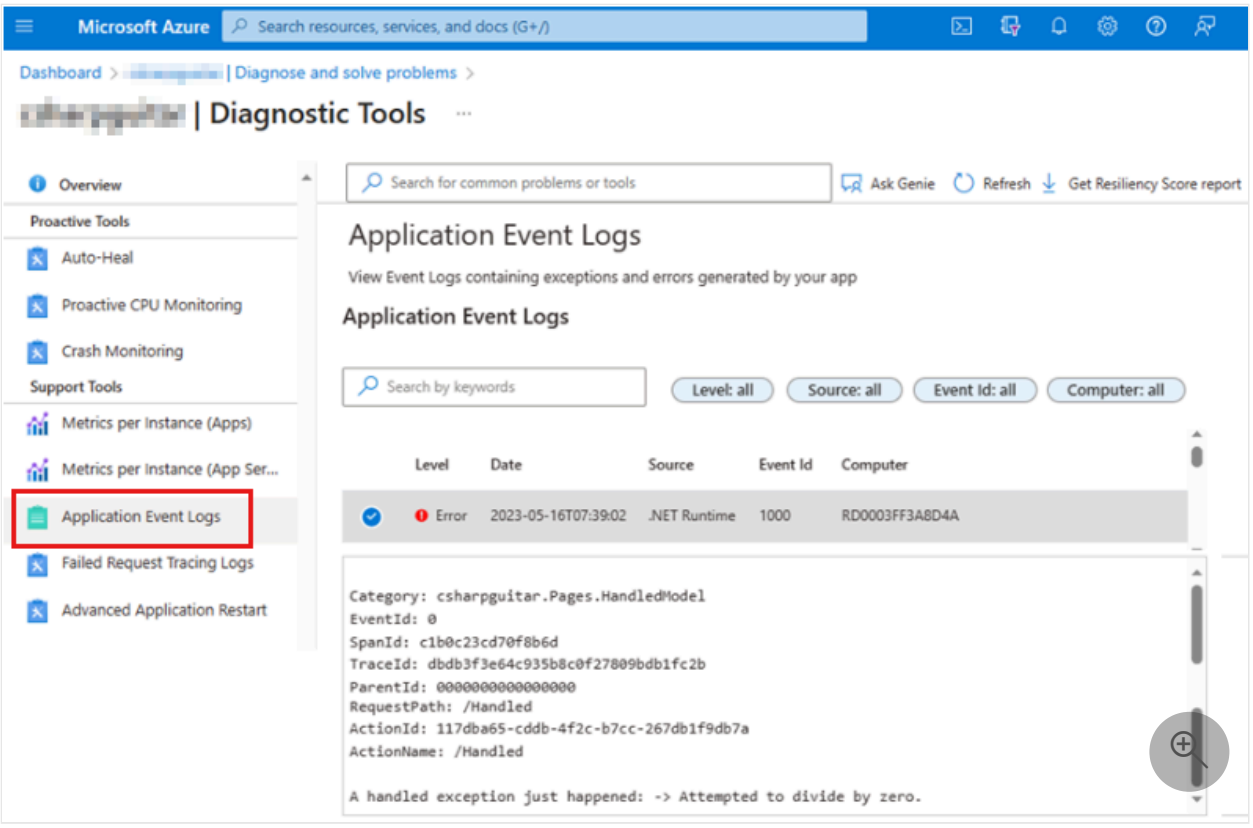
```
traces
| where message has "handled"
 | project timestamp, severityLevel, message, operation_Name,
cloud_RoleInstance
```

The `message` column is the location in which you can store the details that are required to find the root cause of the exception. The code that's used to write this query is in the division-by-zero code snippet. The software developer who wrote this code is the best person to ask about these kinds of exceptions and the attributes that are necessary to capture for analyzing root causes.

The best approach to add this functionality to your application code depends on the application code stack and version that you have (for example, ASP.NET, ASP.NET Core, MVC, Razor, and so on). To determine the best approach for your scenario, review Application Insights logging with .NET.

Application event logs (handled exceptions) feature

You also can find unhandled exceptions in the handled exception in the **Application Event Logs** page of **Diagnostic Tools** in the Azure portal, as shown in the following screenshot.



In this situation, you receive the same error message that you logged through your code. However, you lose some flexibility in how you can customize the queries on Application Insights trace logs.

Application Insights Snapshot Debugger feature

Unhandled exceptions are also logged on the **Application Event Logs** page, as shown in the output text in the next section. However, you can also enable the Application Insights Snapshot Debugger. This approach doesn't require that you add any code to your application.

Application event logs (unhandled exceptions) feature

The following output is from the **Application Event Logs** page of **Diagnostic Tools** in the Azure portal. It shows some example text of an unhandled application exception:

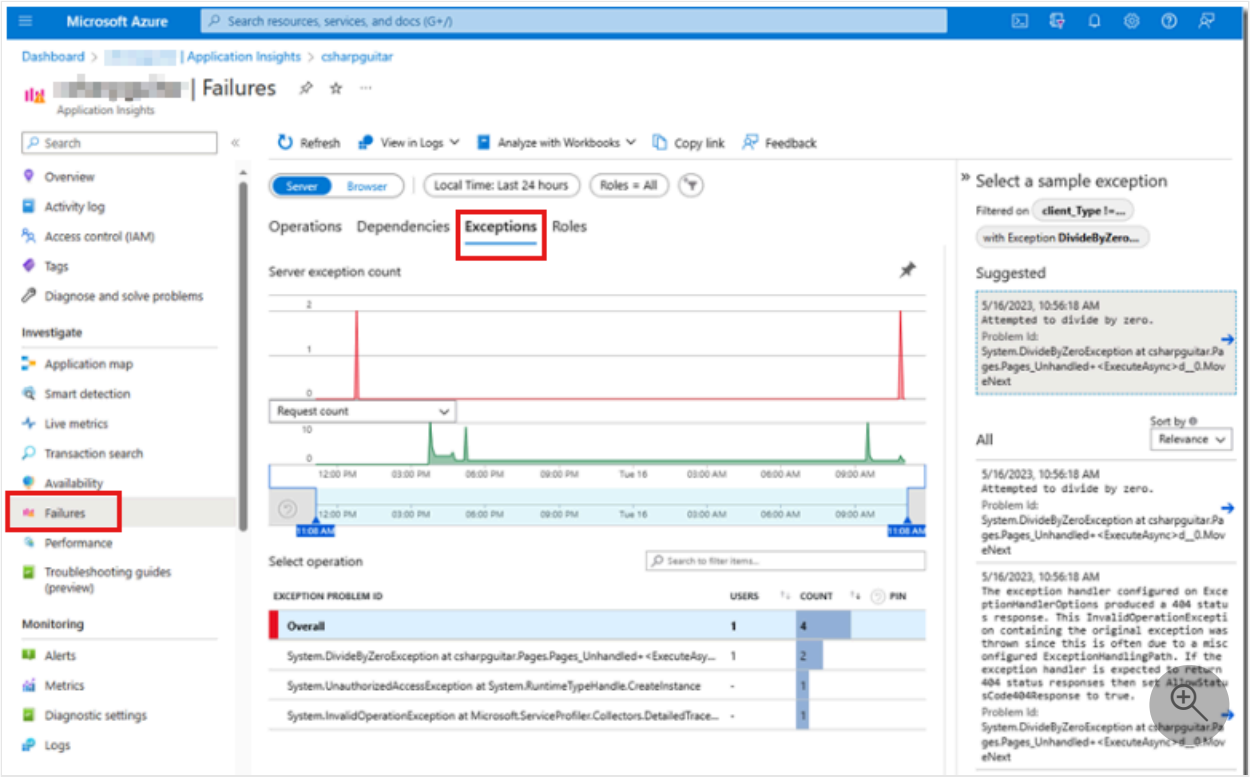
Output

```
Category: Microsoft.AspNetCore.Diagnostics.ExceptionHandlerMiddleware
EventId: 1
SpanId: 311d8cb5d10b1a6e
TraceId: 041929768411c12f1c3f1ccbc91f6751
ParentId: 0000000000000000
RequestId: 8000006d-0001-bf00-b63f-84710c7967bb
RequestPath: /Unhandled

An unhandled exception has occurred while executing the request.

Exception:
System.DivideByZeroException: Attempted to divide by zero.
   at System.Decimal.DecCalc.VarDecDiv(DecCalc& d1, DecCalc& d2)
   at System.Decimal.op_Division(Decimal d1, Decimal d2)
   at contosotest.Pages.Pages_Unhandled.ExecuteAsync()
     in
C:\Users\contoso\source\repos\contosorepo\contosorepo\Pages\Unhandled.cshtml
:line 12
```

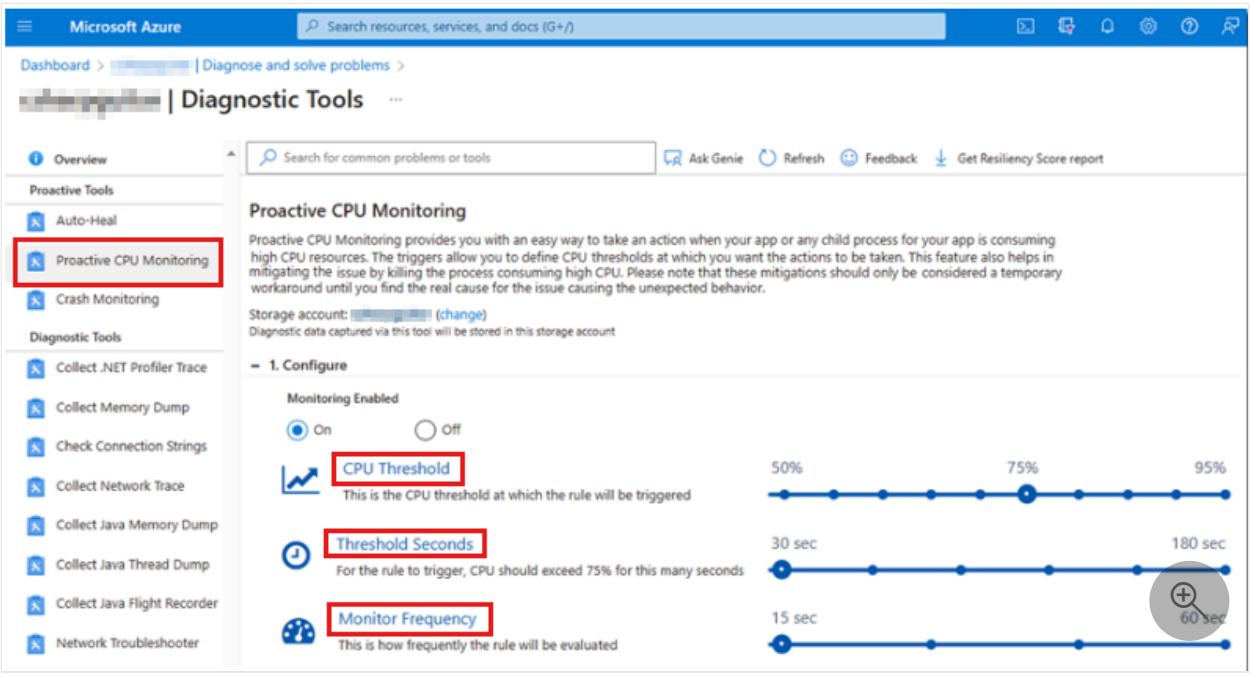
One difference here from the handled exception in the Application log is the existence of the stack that identifies the method and the line from which the exception is thrown. Also, you can safely assume that the Microsoft.AspNetCore.Diagnostics.ExceptionHandlerMiddleware functionality contains code to catch this unhandled exception so that termination of the process is avoided. The exception is shown in Application Insights on the **Exceptions** tab of the **Failures** page, as shown in the following screenshot.



In this view, you see all **Exceptions**, not only the one that you're searching for. The graphical representation of all exceptions that occur in your application is helpful to gain an overview of the health of your system. The Application Insights dashboard is more helpful visually in comparison to the Application event logs.

Proactive CPU monitoring feature

During excessive CPU usage scenarios, you can use the proactive CPU monitoring tool. For information about this tool, see Mitigate your CPU problems before they happen. The following image shows the **Proactive CPU Monitoring** page in **Diagnostic Tools**.



You should consider CPU usage of 80 percent or more as a critical situation that requires immediate investigation. In the **Proactive CPU Monitoring** page, you can set the scenario for which you want to capture a memory dump based on the following data monitoring categories:

- **CPU Threshold**
- **Threshold Seconds**
- **Monitor Frequency**

CPU Threshold identifies how much computer CPU the targeted process uses (W3WP in this case). **Threshold Seconds** is the amount of time that the CPU was used at the **CPU Threshold**. For example, if there's 75-percent CPU usage for a total of 30 seconds, the memory dump would be captured. As configured on the **Proactive CPU Monitoring** page, the process would be checked for threshold breaches every 15 seconds.

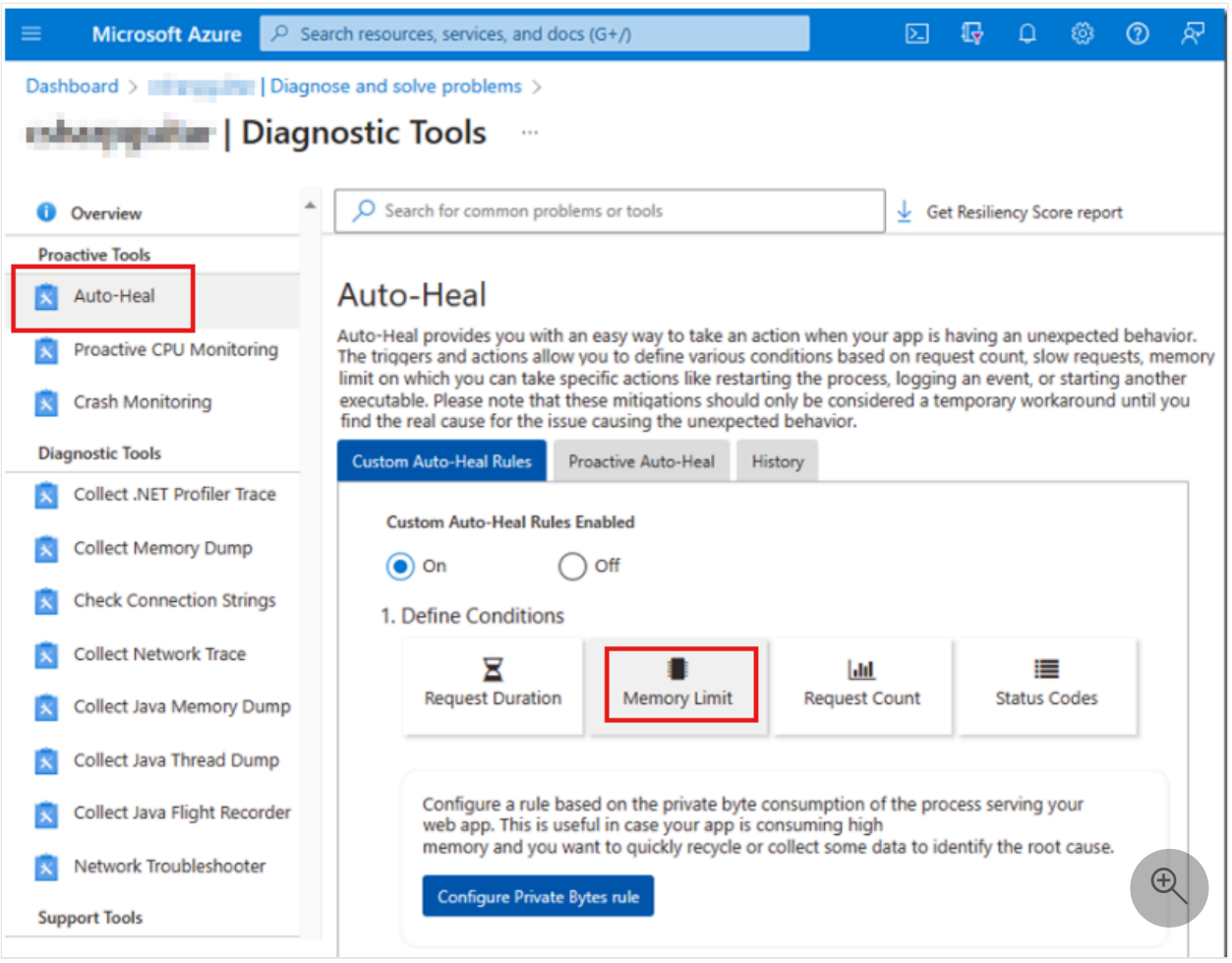
Auto-heal (memory limit) feature

The auto-heal (memory limit) feature is useful for capturing a memory dump if the process is consuming more memory than expected. Again, pay attention to the bitness (32 or 64). If you experience memory pressure in the 32-bit process context, and the memory consumption is expected, you might consider changing the bitness to 64. Typically, if you change the bitness, you have to also recompile the application.

Changing the bitness doesn't reduce the amount of memory that's used. It does allow the process to use more than 4 GB of total memory. However, if the memory consumption isn't as expected, you can use this feature to determine what's consuming the memory. Then, you can take an action to control the memory consumption.

In the "Expanded Azure App Service debugging feature descriptions" section, you can see the link to **Auto-Heal** in the **Diagnostic Tools** tile in the first screenshot. Select that link to go directly to the feature, or select the tile and review all the available tools in the **Diagnostic Tools** page. For more information, go to the "Auto-healing" section of Azure App Service diagnostics overview.

The auto-heal feature is shown in the following screenshot.



When you select the **Memory Limit** tile, you have the option to enter a memory value that triggers the capture of a memory dump when that memory limit is breached. For

example, if you enter *6291456* as the value, a memory dump of the W3WP process is taken when 6 GB of memory is consumed.

The Collect a Memory dump feature is useful in this scenario if the issue is currently occurring or reproducible. This feature quickly collects a memory dump on manual demand. For more information, see the "Collect a memory dump" section.

Expanded command descriptions

The art of memory dump collection takes some time to study, experience, and perfect. As you have learned, different procedures are based on the symptoms that the process is showing, as listed in the table in the "Expanded process scenario descriptions" section. By contrast, the following table compares the Azure App Service's memory dump capture command to the procdump command that you run manually from the Kudu console.

Expand table

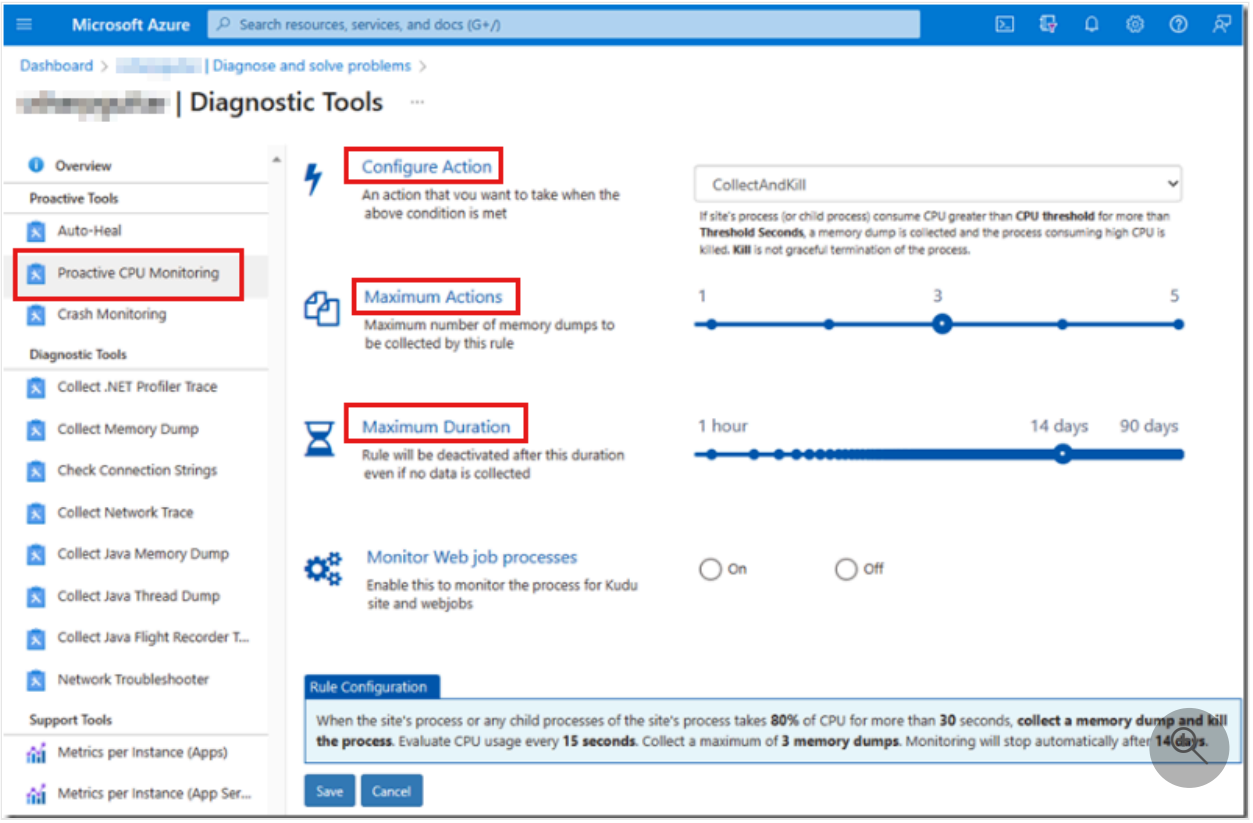
Scenario	Azure App Service command	General procdump command
Unresponsive or slow	`procdump -accepteula -r -dc "Message" -ma <PID> <PATH>`	`procdump -accepteula -ma -n 3 -s # <PID>`
Crash (process termination)	Uses DbgHost to capture memory dump	`procdump -accepteula -ma -t <PID>`
Crash (handled exceptions)	None (Application Insights)	`procdump -accepteula -ma -e 1 -f <filter> <PID>`
Crash (unhandled exceptions)	None (Application Insights Snapshot Debugger)	`procdump -accepteula -ma -e <PID>`
Excessive CPU usage	`procdump -accepteula -dc "Message" -ma <PID> <PATH>`	`procdump -accepteula -ma -n 3 -s # -c 80 <PID>`
Excessive memory consumption	`procdump -accepteula -r -dc "Message" -ma <PID> <PATH>`	`procdump -accepteula -ma -m 2000 <PID>`

The commands that you use in the memory dump capturing features in Azure App Service differ from the procdump commands that you would use if you captured dumps manually. If you review the previous section, you should notice that the memory dump collection portal feature in Azure App Service exposes the configuration. For example, in the excessive memory consumption scenario in the table, the command that the platform runs doesn't contain a memory threshold. However, the command that's shown in the general procdump command column does specify a memory threshold.

A tool that's named DaaS (Diagnostics as a service) is responsible for managing and monitoring the configuration that's specified in the Azure App Service debugging portal. This tool runs as a web job on the virtual machines (VMs) that run your web app. A benefit of this tool is that you can target a specific VM in your web farm. If you try to capture a memory dump by using procdump directly, it can be challenging to identify, target, access, and run that command on a specific instance. For more information about DaaS, see DaaS – Diagnostics as a service for Azure web sites.

Excessive CPU usage is another reason why the platform manages the memory dump collection so that they match the recommended procdump patterns. The procdump command, as shown in the previous table, collects three (`-n 3`) full memory dumps (`-ma`) 30 seconds apart (`-s #`, in which `#` is 30) when the CPU usage is greater than or equal to 80 percent (`-c 80`). Finally, you provide the process ID (`<PID>`) to the command: `procdump -accepteula -ma -n 3 -s # -c 80 <PID>`.

You can see the portal configuration in the "Proactive CPU monitoring" section. For brevity, that section showed only the first three configuration options: **CPU Threshold** (`-c`), **Threshold Seconds** (`-s`), and **Monitor Frequency**. The following screenshot illustrates that **Configure Action**, **Maximum Actions** (`-n`), and **Maximum Duration** are extra available features.



After you study the different approaches for capturing memory dumps, the next step is to practice making captures. You can use code examples on GitHub in conjunction with IIS debugging labs and Azure Functions to simulate each of the scenarios that are

listed in the two tables. After you deploy the code to the Azure App Service platform, you can use these tools to capture the memory dump under each given scenario. Over time and after practice, you can perfect your approach for capturing memory dumps by using the Azure App Service debugging features. The following list contains a few suggestions to consider as you continue to learn about memory dump collection:

- Capturing a memory dump consumes significant system resources and disrupts performance even further.
- Capturing memory dumps on the first chance isn't optimal because you will probably capture too many. Those first-chance memory dumps are most likely irrelevant.
- We recommend that you disable Application Insights before you capture a W3WP memory dump.

After the memory dump is collected, the next step is to analyze the memory dump to determine the cause of the problem, and then correct that problem.

Next steps (analyzing the memory dump)

Discussing how to analyze memory dumps is outside the scope of this article. However, there are many resources for that subject, such as the Defrag Tools training series and a list of must-know WinDbg commands.

You might have noticed the **Configure Action** option in the previous screenshot. The default setting for this option is **CollectAndKill**. This setting means that the process is killed after the memory dump is collected. A setting that's named **CollectKillAndAnalyze** analyzes the memory dump that's collected. In that scenario, the platform analysis might find the issue so that you don't have to open the memory dump in WinDbg and analyze it.

There are other options for troubleshooting and diagnosing performance issues on the Azure App Service platform. This article focuses on memory dump collection and makes some recommendations for approaching the diagnosis by using these methods. If you have already studied, experienced, and perfected your collection procedures, and they work well for you, you should continue to use those procedures.

Contact us for help

If you have questions or need help, create a support request, or ask Azure community support. You can also submit product feedback to Azure feedback community.

Feedback

Was this page helpful? Yes No

Provide product feedback

Frequently asked questions about creating or deleting resources in Azure App Service

This article answers common questions about creating and deleting web apps in Azure App Service.

If your Azure issue is not addressed in this article, visit the Azure forums on MSDN and Stack Overflow. You can post your issue in these forums, or post to @AzureSupport on Twitter. You also can submit an Azure support request. To submit a support request, on the Azure support page, select **Get support**.

Create a web app

What if I create a web app with the same name as another one?

We cannot create a web app with a name that already exists in the Azure. The web app name is part of the web app's URL, so it must be unique among all Azure App Service web apps.

How can I create a web app in a region that's unavailable in the subscription?

Certain Azure regions require customers to go through a request process to gain access. For more information, see the Azure region access request process and Azure App Service available by region.

What if the requested SKU of the App Service Plan isn't available in the Resource Group?

Your App Service deployments may not have specific SKUs available due to constraints (for example, the SKU isn't available in the region or country).

If the requested SKU isn't available in the Resource Group, you must create a new App Service Plan in a new Resource Group in the same region or in the same Resource Group using a different region.

If you want to reuse the same Resource Group and region, you must delete all the App Services, App Service Plans, and `Microsoft.Web/Certificates` resources in this Resource Group,

then create the desired SKU in this Resource Group. Note that migrating App Services originating in this Resource Group to another Resource Group won't unblock the creation.

Delete or restore a web app

If I delete all my web apps, will I still be charged?

Yes, charges will still apply unless you delete the App Service plan that the web app runs on and its all web apps.

To stop all billing associated with your App Service, you need to delete the App Service Plan or scale the App Service Plan to the free tier. For more information, see How much does my App Service Plan cost? and Plan and manage costs for Azure App Service.

I cannot create or delete a web app due to a permission error. What the permissions do I need to create or delete a web app?

You would need minimum Contributor access on the Resource Group to deploy App Services. If you have Contributor access only on App Service Plan and web app, it won't allow you to create the app service in the Resource Group.

For more information, see Azure built-in roles.

I'm trying to delete my App Service Plan, but I'm getting the following error "Storage usage quota exceeded. Can't update or delete a server farm. Please make sure your file system storage is below the limit of the target pricing tier". How and where do I check the file system storage limit?

During the deletion process, we calculate the usage of the remaining App Service Plans. If they are above the remaining limit, then this error appears.

The storage limit is the total content size across all apps in the same App Service Plan. The total content size of all apps across all App Service Plans in a single Resource Group and region can't exceed 500 GB. The file system quota for App Service hosted apps is determined by the aggregate of App Service Plans created in a region and Resource Group.

For more information, see App Service limits - Storage and App Service pricing.

Is there a way to list the deleted web apps for my Subscription?

You can run `Get-AzDeletedWebApp` to get the list of the web apps that were deleted within the last 30 days in your Subscription ID. Deleted apps are purged from the system 30 days after the initial deletion. After an app is purged, it can't be recovered. For more information, see List deleted apps.

How do I restore a deleted web app or a deleted App Service Plan?

If the web app was deleted within the last 30 days, you can restore it using `Restore-AzDeletedWebApp`. For more information, see Restore deleted app and Restore Deleted web apps.

Contact us for help

If you have questions or need help, create a support request, or ask Azure community support. You can also submit product feedback to Azure feedback community.

Frequently asked questions about App Service security

Article • 01/23/2025

This article provides answers to common questions about Azure App Service security.

FAQs

How do I know whether a specific CVE (Common Vulnerabilities and Exposures) or known security issue applies to my web app?

Microsoft Security Response Center (MSRC) investigates all reports of security vulnerabilities that affect Microsoft products and services. MSRC provides this information in the Security Update Guide as part of an ongoing effort to help you manage security risks and keep your systems protected.

If your question isn't answered and you still need help, submit a support request that includes the number of the CVE.

To report a vulnerability, see Report an issue.

How do I know when a particular specific version of software or security patch will arrive at the Azure platform runtime?

App Service is a platform that has various underlying technologies, such as Windows, Linux, and web application frameworks. Updates are applied at a routine cadence for OS, host runtime, and Microsoft image repo.

- Check this article to understand OS and runtime updating in Azure App Service regarding the OS or software in App Service.
- Check Guest OS update details to understand the updates that are applied to the Azure Guest OS.

If you still need help, gather the following information before you submit a request to Azure support:

- Specify the security update that you're inquiring about.

- Verify the security update version of the software that's deployed on Azure.
- Determine whether the update is already applied in Azure.

Is TLS 1.3 supported on Azure App Service?

For incoming requests to your web app, App Service supports TLS versions 1.0, 1.1, 1.2, and 1.3. See Azure App Service TLS overview for more information.

How do I disable weak ciphers on Azure App Service?

A cipher suite is a set of instructions that contains algorithms and protocols to help secure network connections between clients and servers. A client makes a request to the server that includes a list of cipher suites that it supports, and the server (front-end of the web app) picks the most secure suite that's supported by both client and server. For a more comprehensive discussion of cipher suites, see Demystifying Cipher Suites on Azure App Services.

For Azure App Service Environment (ASE), you can set your own ciphers through Azure Resource Explorer. For detailed steps, see Change TLS cipher suite order.

To disable Weak TLS cipher suites for web apps on multitenant setups, see Disabling weaker TLS ciphers suites for web apps on multitenant Premium App Service plans.

For more information, see FAQ on App Service cipher suites.

How do I enable protection against DDoS attacks or suspicious activity for my app service?

By default, Distributed Denial of Service (DDoS) protection is not enabled for App Service plans and their app services.

You can use Azure DDoS Protection to protect your Azure resources from attacks. Azure DDoS Protection, combined with application design best practices, provides enhanced DDoS mitigation features to defend against DDoS attacks.

Notice that Azure Traffic Manager is a DNS-based traffic load balancer that enables you to distribute traffic optimally to services across global Azure regions while providing high availability and responsiveness. However, Traffic Manager does not provide protection against DDoS attacks.

I suspect that my website is being hacked. What should I do?

Microsoft secures and frequently updates the hosting environment and infrastructure. If a website was hacked or defaced, this usually indicates an exploited vulnerability that's caused by an outdated app package.

Azure App Service does not block insecure apps from running. If the website is vulnerable, you must fix the vulnerabilities in the website code, and then redeploy it to Azure App Service.

Azure support can help you review the web app's HTTP logs and deployment history to identify when the unknown file was first accessed or whether suspicious patterns appear in the logs. We can also offer guidance about how to configure security services such as Web Application Firewall and Microsoft Defender for App Service. However, we can't take direct action because the permanent fix might involve implementing a Web Application Firewall or updating the existing codes.

You can restore a backup or redeploy the site, but this is not a long-term fix if the security issue is not resolved.

My site has been added to the blocklist. What should I do?

If the IP address is frequently blocklisted, it's important to investigate the root cause. The blockage might be caused by sending spam email messages, hosting malicious content, or other security vulnerabilities that should be resolved.

- **Inbound IP blocklisted**: To address an inbound IP blocklisting issue, request a static inbound IP address by using an IP-based SSL to secure your domain. Alternatively, you can use Azure services such as Azure Application Gateway or App Service Environment (ASE) to gain a dedicated inbound IP address.

- **Outbound IP blocklisted**: The only way to request dedicated outbound IP addresses is to use an App Service Environment. Apps that run in Azure share outbound addresses from a common pool.
  - You can deploy your app in a different (resource group + location) to host the application in a new scale unit. Scaling your app between pricing tiers will also trigger a change in outbound IP addresses.
  - Alternatively, use Azure's NAT Gateway to assign dedicated outbound IP addresses to your resources.

    - For more information, see How to fix outbound IPs for App Service using NAT Gateway.

- **SMTP blocklisted**: Port 25 is mainly used for unauthenticated email delivery. Outbound connections from App Services to the public internet by using port 25 are not restricted. However, using this design could result in outbound IP addresses being flagged as spam and, therefore, blocklisted.
    - We recommend that you use authenticated SMTP relay services to send email or implement App Service VNet Integration.
    - Alternatively, host the App Service in an App Service Environment (ASE) to route outbound SMTP connections over a private network.
    - For details, refer to Troubleshoot outbound SMTP connectivity problems in Azure.

Why am I receiving warnings or alerts for my web app in security scan reports?

Security scans are typically run against a web app URL. Make sure that the tested URL resolves to the intended web app. If it resolves elsewhere, such as an application gateway, you can expect to receive inaccurate scan results.

Some scan results could be false positives even as others indicate a genuine security issue that might require a consultation with Azure support. Certain changes are within your control, such as networking or website configuration. Other changes at the platform level can be controlled only by Microsoft.

Azure support can assist you by reviewing the full scan results, confirming the results, and providing security feature options to you.

Feedback

Application performance FAQs for Web Apps in Azure

Article • 04/15/2025

> **Note**
>
> Some of the following guidelines might only work on Windows or Linux App Services. For example, Linux App Services run in 64-bit mode by default.

This article has answers to frequently asked questions (FAQs) about application performance issues for the Web Apps feature of Azure App Service.

If your Azure issue is not addressed in this article, visit the Azure forums on MSDN and Stack Overflow. You can post your issue in these forums, or post to @AzureSupport on Twitter. You also can submit an Azure support request. To submit a support request, on the Azure support page, select **Get support**.

Why is my App Service Plan displaying CPU/Memory usage even when all Web Apps are stopped?

Azure App Service requires continuous system processes that handle several platform operations and features, such as security updates, availability of the SCM console, application monitoring, authentication, and many other vital features of your Web App.

System processes will run on App Service Plans even if there are no Web Apps running or if the App Service Plan contains no Web Apps.

The platform processes will consume a minimum amount of resources (such as CPU, Memory and Disk space), and the same should be accounted during the capacity planning, monitoring, and auto-scaling trigger configuration of an App Service Plan.

Why is my app slow?

Multiple factors might contribute to slow app performance. For detailed troubleshooting steps, see Troubleshoot slow web app performance.

> **Tip**

- Enable the **Always On** setting under **Configuration** > **General settings** to keep your app warm and avoid cold starts. This helps reduce delay after idle time, especially in Basic and higher plans.
- Configure a Health check path to monitor app health and automatically replace unresponsive instances. This helps maintain availability and performance. For more information, see **Monitor App Service instances by using Health check**.

How do I troubleshoot a high CPU-consumption scenario?

In some high CPU-consumption scenarios, your app might truly require more computing resources. In that case, consider scaling to a higher service tier so the application gets all the resources it needs. Other times, high CPU consumption might be caused by a bad loop or by a coding practice. Getting insight into what's triggering increased CPU consumption is a two-part process. First, create a process dump, and then analyze the process dump. For more information, see Capture and analyze a dump file for high CPU consumption for Web Apps.

How do I troubleshoot a high memory-consumption scenario?

In some high memory-consumption scenarios, your app might truly require more computing resources. In that case, consider scaling to a higher service tier so the application gets all the resources it needs. Other times, a bug in the code might cause a memory leak. A coding practice might also increase memory consumption. Getting insight into what's triggering high memory consumption is a two-part process. First, create a process dump, and then analyze the process dump. Crash Diagnoser from the Azure Site Extension Gallery can efficiently perform both these steps. For more information, see Capture and analyze a dump file for intermittent high memory for Web Apps.

How do I automate App Service web apps by using PowerShell?

You can use PowerShell cmdlets to manage and maintain App Service web apps. In our blog post Automate web apps hosted in Azure App Service by using PowerShell, we describe how to use Azure Resource Manager-based PowerShell cmdlets to automate common tasks.

> ⓘ **Note**
>
> For current automation scripts, use the latest Az.Websites module. The older `AzureRM` module is deprecated.

How do I view my web app's event logs?

To view your web app's event logs:

1. Sign in to your **Kudu website** (`https://*yourwebsitename*.scm.azurewebsites.net`).
2. In the menu, select **Debug Console** > **CMD**.
3. Select the **LogFiles** folder.
4. To view event logs, select the pencil icon next to **eventlog.xml**.
5. To download the logs, run the PowerShell cmdlet `Save-AzureWebSiteLog -Name webappname`.

How do I capture a user-mode memory dump of my web app?

To capture a user-mode memory dump of your web app:

1. Sign in to your **Kudu website** (`https://*yourwebsitename*.scm.azurewebsites.net`).
2. Select the **Process Explorer** menu.
3. Right-click the **w3wp.exe** process or your WebJob process.
4. Select **Download Memory Dump** > **Full Dump**.

How do I view process-level info for my web app?

You have two options for viewing process-level information for your web app:

- In the Azure portal:

    1. Open the **Process Explorer** for the web app.
    2. To see the details, select the **w3wp.exe** process.

- In the Kudu console:

    1. Sign in to your **Kudu website** (`https://*yourwebsitename*.scm.azurewebsites.net`).
    2. Select the **Process Explorer** menu.
    3. For the **w3wp.exe** process, select **Properties**.

When I browse to my app, I see "Error 403 - This web app is stopped." How do I resolve this?

Three conditions can cause this error:

- The web app has reached a billing limit and your site has been disabled.
- The web app has been stopped in the portal.
- The web app has reached a resource quota limit that might apply to a Free or Shared scale service plan.

To see what is causing the error and to resolve the issue, follow the steps in Web Apps: "Error 403 – This web app is stopped".

Where can I learn more about quotas and limits for various App Service plans?

For information about quotas and limits, see App Service limits.

How do I turn on failed request tracing?

To turn on failed request tracing, follow these steps:

1. In the Azure portal, go to your web app.
2. Select **All Settings** > **Diagnostics Logs**.
3. For **Failed Request Tracing**, select **On**.
4. Select **Save**.
5. On the web app blade, select **Tools**.
6. Select **Visual Studio Online**.
7. If the setting isn't **On**, select **On**.
8. Select **Go**.
9. Select **Web.config**.
10. In system.webServer, add the following configuration (to capture a specific URL):

```
XML
```

```
<system.webServer>
<tracing> <traceFailedRequests>
<remove path="*api*" />
<add path="*api*">
<traceAreas>
<add provider="ASP" verbosity="Verbose" />
<add provider="ASPNET" areas="Infrastructure,Module,Page,AppServices"
verbosity="Verbose" />
<add provider="ISAPI Extension" verbosity="Verbose" />
<add provider="WWW Server"
areas="Authentication,Security,Filter,StaticFile,CGI,Compression,
Cache,RequestNotifications,Module,FastCGI" verbosity="Verbose" />
</traceAreas>
<failureDefinitions statusCodes="200-999" />
</add> </traceFailedRequests>
</tracing>
```

11. To troubleshoot slow-performance issues, add this configuration (if the capturing request is taking more than 30 seconds):

XML

```
<system.webServer>
<tracing> <traceFailedRequests>
<remove path="*" />
<add path="*">
<traceAreas> <add provider="ASP" verbosity="Verbose" />
<add provider="ASPNET" areas="Infrastructure,Module,Page,AppServices"
verbosity="Verbose" />
<add provider="ISAPI Extension" verbosity="Verbose" />
<add provider="WWW Server"
areas="Authentication,Security,Filter,StaticFile,CGI,Compression,
Cache,RequestNotifications,Module,FastCGI" verbosity="Verbose" />
</traceAreas>
<failureDefinitions timeTaken="00:00:30" statusCodes="200-999" />
</add> </traceFailedRequests>
</tracing>
```

12. To download the failed request traces, in the portal, go to your website.

13. Select **Tools** > **Kudu** > **Go**.

14. In the menu, select **Debug Console** > **CMD**.

15. Select the **LogFiles** folder, and then select the folder with a name that starts with **W3SVC**.

16. To see the XML file, select the pencil icon.

I see the message "Worker Process requested recycle due to 'Percent Memory' limit." How do I address this issue?

The maximum available amount of memory for a 32-bit process (even on a 64-bit operating system) is 2 GB. By default, the worker process is set to 32-bit in App Service (for compatibility with legacy web applications).

Consider switching to 64-bit processes so you can take advantage of the additional memory available in your Web Worker role. This action triggers a web app restart, so schedule accordingly.

Also note that a 64-bit environment requires a Basic or Standard service plan. Free and Shared plans always run in a 32-bit environment.

For more information, see Configure web apps in App Service.

Why does my request time out after 230 seconds?

Azure Load Balancer has a default idle timeout setting of four minutes. This setting is generally a reasonable response time limit for a web request. Therefore, App Service returns a timeout to the client if your application does not return a response within approximately 240 seconds (230 seconds on Windows app, 240 seconds on Linux app). If your web app requires background processing, we recommend using Azure WebJobs. The Azure web app can call WebJobs and be notified when background processing is finished. You can choose from multiple methods for using WebJobs, including queues and triggers.

WebJobs is designed for background processing. You can do as much background processing as you want in a WebJob. For more information about WebJobs, see Run background tasks with WebJobs.

ASP.NET Core applications that are hosted in App Service sometimes stop responding. How do I fix this issue?

A known issue with an earlier Kestrel version might cause an ASP.NET Core 1.0 app that's hosted in App Service to intermittently stop responding. You might also see this message: "The specified CGI Application encountered an error and the server terminated the process."

This issue is fixed in Kestrel version 1.0.2. This version is included in the ASP.NET Core 1.0.3 update. To resolve this issue, make sure you update your app dependencies to use Kestrel 1.0.2. Alternatively, you can use one of two workarounds that are described in the blog post ASP.NET Core 1.0 slow perf issues in App Service web apps.

I can't find my log files in the file structure of my web app. How can I find them?

If you use the Local Cache feature of App Service, the folder structure of the LogFiles and Data folders for your App Service instance are affected. When Local Cache is used, subfolders are created in the storage LogFiles and Data folders. The subfolders use the naming pattern "unique identifier" + time stamp. Each subfolder corresponds to a VM instance in which the web app is running or has run.

To determine whether you're using Local Cache, check your App Service **Application settings** tab. If Local Cache is being used, the app setting `WEBSITE_LOCAL_CACHE_OPTION` is set to `Always`.

If you aren't using Local Cache and are experiencing this issue, submit a support request.

I see the message "An attempt was made to access a socket in a way forbidden by its access permissions." How do I resolve this error?

This error typically occurs if the outbound TCP connections on the VM instance are exhausted. In App Service, limits are enforced for the maximum number of outbound connections that can be made for each VM instance. For more information, see Cross-VM numerical limits.

This error might also occur if you try to access a local address from your application. For more information, see Local address requests.

For more information about outbound connections in your web app, see the blog post about outgoing connections to Azure websites.

How do I use Visual Studio to remote debug my App Service web app?

For a detailed walkthrough that shows you how to debug your web app by using Visual Studio, see Remote debug your App Service web app.

Additional recommendations for performance and resiliency

- Use Application Insights and Azure Monitor for full-stack observability of your App Service app, including telemetry, dependency tracing, and live metrics.
- If you're deploying in regions that support availability zones, consider enabling zone redundancy to enhance resiliency during regional outages. For more information, see Reliability in Azure App Service.
- App Service undergoes routine maintenance to ensure platform reliability. For more control over update behavior, especially in App Service Environment v3, configure upgrade preference. For more information, see Routine (planned) maintenance for Azure App Service.

Contact us for help

If you have questions or need help, create a support request, or ask Azure community support. You can also submit product feedback to Azure feedback community.

Azure App Service Compliance with PCI Standards 3.0 and 3.1

Article • 06/23/2022

Original product version: Web App (Windows)
Original KB number: 3124528

The Azure App Service is currently in compliance with PCI DSS version 3.0 Level 1. We have also noted customer requests that make reference to PCI DSS version 3.1, and specifically the change from version 3.0 to 3.1, which states that SSL and "early TLS versions " will no longer be considered valid security options from June 30, 2018. From June 30th, 2018, the multi-tenant hosting model for Azure App Service, will be accepting TLS 1.2 with an option for customers to select their required TLS encryption level.

What this means

PCI DSS version 3.1 certification requires disabling TLS 1.0. If you are using App Service Environments or are willing to migrate your workload to App Service Environments, you can get greater control of your environment including disabling TLS 1.0, for more information, see Custom configuration settings for App Service Environments.

More information

Microsoft regularly reviews standards compliance procedures and will periodically update compliance baselines as standards bodies update and change their requirements. As part of Microsoft's Fiscal 2017 compliance planning, PCI standards will again be re-reviewed and technical determinations will be made. To view the current certifications, technical determinations will be made. To view the current certifications, visit the Microsoft Azure Trust Center: Compliance site.

Contact us for help

If you have questions or need help, create a support request, or ask Azure community support. You can also submit product feedback to Azure community support.

Feedback

Backing up, restoring, and cloning Microsoft Azure App Service

Article • 06/23/2022

In this video, you will learn about different kinds of backup and restore procedures (traditional versus snapshot), the limitations of each, the requirements to complete the action (snapshot), and how to clone the app service to another ASP/ region.
https://www.youtube-nocookie.com/embed/P2BRPuQYPSo

Related content

For more troubleshooting tips, see My backups are failing - Azure App Service

Contact us for help

If you have questions or need help, create a support request, or ask Azure community support. You can also submit product feedback to Azure community support.

Feedback

Was this page helpful? Yes No

Provide product feedback | Get help at Microsoft Q&A

Azure App Service on Linux FAQ

> **Note**
>
> Was this article helpful? Your input is important to us. Please use the **Feedback** button on this page to let us know how well this article worked for you or how we can improve it.

With the release of App Service on Linux, we're working on adding features and making improvements to our platform. This article provides answers to questions that our customers have been asking us recently.

If you have a question, comment on this article.

Built-in images

What are the expected values for the Startup File section when I configure the runtime stack?

Expand table

Stack	Expected Value
Java SE	the command to start your JAR app (for example, `java -jar /home/site/wwwroot/app.jar --server.port=80`)
Tomcat	the location of a script to perform any necessary configurations (for example, `/home/site/deployments/tools/startup_script.sh`)
Node.js	the PM2 configuration file or your script file
.NET Core	the compiled DLL name as `dotnet <myapp>.dll`
PHP	optional custom startup
Python	optional startup script

Stack	Expected Value
Ruby	the Ruby script that you want to initialize your app with

These commands or scripts are executed after the built-in Docker container is started, but before your application code is started.

Management

What happens when I press the restart button in the Azure portal?

This action is the same as a Docker restart.

Can I use Secure Shell (SSH) to connect to the app container virtual machine (VM)?

Yes, you can do that through the source control management (SCM) site.

> **Note**
>
> You can also connect to the app container directly from your local development machine using SSH, SFTP, or Visual Studio Code (for live debugging Node.js apps). For more information, see Remote debugging and SSH in App Service on Linux.

How can I create a Linux App Service plan through an SDK or an Azure Resource Manager template?

Set the **reserved** field of the app service to *true*.

Continuous integration and deployment

My web app still uses an old Docker container image after I've updated the image on Docker Hub. Do you

support continuous integration and deployment of custom containers?

Yes, to set up continuous integration/deployment for Azure Container Registry or DockerHub, by following Continuous Deployment with Web App for Containers. For private registries, you can refresh the container by stopping and then starting your web app. Or you can change or add a dummy application setting to force a refresh of your container.

Do you support staging environments?

Yes.

Can I use 'WebDeploy/MSDeploy' to deploy my web app?

Yes, you need to set an app setting called `WEBSITE_WEBDEPLOY_USE_SCM` to *false*.

Git deployment of my application fails when using Linux web app. How can I work around the issue?

If Git deployment fails to your Linux web app, choose one of the following options to deploy your application code:

- Use the Continuous Delivery (Preview) feature: You can store your app's source code in an Azure DevOps Git repo or GitHub repo to use Azure Continuous Delivery. For more information, see How to configure Continuous Delivery for Linux web app .
- Use the ZIP deploy API : To use this API, SSH into your web app and go to the folder where you want to deploy your code. Run the following code:

  Bash

  ```
  curl -X POST -u <user> --data-binary @<zipfile> https://{your-sitename}.scm.azurewebsites.net/api/zipdeploy
  ```

  If you get an error that the `curl` command isn't found, make sure you install curl by using `apt-get install curl` before you run the previous `curl` command.

Language support

I want to use web sockets in my Node.js application, any special settings, or configurations to set?

Yes, disable `perMessageDeflate` in your server-side Node.js code. For example, if you're using socket.io, use the following code:

Node.js

```
const io = require('socket.io')(server,{
  perMessageDeflate :false
});
```

Do you support uncompiled .NET Core apps?

Yes.

Do you support Composer as a dependency manager for PHP apps?

Yes, during a Git deployment, Kudu should detect that you're deploying a PHP application (thanks to the presence of a composer.lock file), and Kudu will then trigger a composer install.

Custom containers

Can I use Managed Identities with App Service when pulling images form ACR?

Yes, this functionality is available from the Azure CLI. You can use system-assigned or user-assigned identities. This functionality isn't currently supported in the Azure portal.

I'm using my own custom container. I want the platform to mount an SMB share to the `/home/` directory.

If `WEBSITES_ENABLE_APP_SERVICE_STORAGE` setting is **unspecified** or set to *false*, the `/home/` directory **won't be shared** across scale instances, and files written **won't persist** across restarts. Explicitly setting `WEBSITES_ENABLE_APP_SERVICE_STORAGE` to *true* enables the mount. Once this is

set to true, if you wish to disable the mount, you need to explicitly set `WEBSITES_ENABLE_APP_SERVICE_STORAGE` to *false*.

My container fails to start with "no space left on device". What does this error mean?

App Service on Linux uses two different types of storage:

- File system storage: The file system storage is included in the App Service plan quota. It's used when files are saved to the persistent storage that's rooted in the `/home` directory.
- Host disk space: The host disk space is used to store container images. It's managed by the platform through the docker storage driver.

The host disk space is separate from the file system storage quota. It's not expandable and there is a 15-GB limit for each instance. It's used to store any custom images on the worker. You might be able to use larger than 15 GBs depending on the exact availability of host disk space, but this isn't guaranteed.

If the container's writable layer saves data outside of the `/home` directory or a mounted azure storage path, the host disk space will also be consumed.

The platform routinely cleans the host disk space to remove unused containers. If the container writes a large quantity of data outside of the `/home` directory or Bring Your Own Storage (BYOS), it results in startup failures or runtime exceptions once the host disk space limit is exceeded.

We recommend that you keep your container images as small as possible and write data to the persistent storage or BYOS when running on Linux App Service. If not possible, you have to split the App Service plan because the host disk space is fixed and shared between all containers in the App Service Plan.

My custom container takes a long time to start, and the platform restarts the container before it finishes starting up.

You can configure the amount of time the platform waits before restarting the container. To do so, set the `WEBSITES_CONTAINER_START_TIME_LIMIT` app setting to the value you want. The default value is 230 seconds, and the maximum value is 1800 seconds.

What is the format for the private registry server URL?

Provide the full registry URL, including `https://`.

What is the format for the image name in the private registry option?

Add the full image name, including the private registry URL (for example, myacr.azurecr.io/dotnet:latest). Image names that use a custom port can't be entered through the portal. To set `docker-custom-image-name`, use the az command-line tool.

Can I expose more than one port on my custom container image?

We don't support exposing more than one port.

Can I bring my own storage?

Yes, bring your own storage is in preview.

Why can't I browse my custom container's file system or running processes from the SCM site?

The SCM site runs in a separate container. You can't check the file system or running processes of the app container.

Do I need to implement HTTPS in my custom container?

No, the platform handles HTTPS termination at the shared front ends.

Do I need to use WEBSITES_PORT for custom containers?

Yes, this is required for custom containers. To manually configure a custom port, use the EXPOSE instruction in the Dockerfile and the app setting, WEBSITES_PORT, with a port value to bind on the container.

Can I use ASPNETCORE_URLS in the Docker image?

Yes, overwrite the environmental variable before .NET core app starts. E.g., In the init.sh script: export ASPNETCORE_URLS={Your value}

Multi-container with Docker Compose

How do I configure Azure Container Registry (ACR) to use with multi-container?

In order to use ACR with multi-container, **all container images** need to be hosted on the same ACR registry server. Once they are on the same registry server, you'll need to create application settings and then update the Docker Compose configuration file to include the ACR image name.

Create the following application settings:

- DOCKER_REGISTRY_SERVER_USERNAME
- DOCKER_REGISTRY_SERVER_URL (full URL, ex: `https://<server-name>.azurecr.io`)
- DOCKER_REGISTRY_SERVER_PASSWORD (enable admin access in ACR settings)

Within the configuration file, reference your ACR image like the following example:

YAML

```yaml
image: <server-name>.azurecr.io/<image-name>:<tag>
```

How do I know which container is internet accessible?

- Only one container can be open for access
- Only port 80 and 8080 is accessible (exposed ports)

Here are the rules for determining which container is accessible - in the order of precedence:

- Application setting `WEBSITES_WEB_CONTAINER_NAME` set to the container name
- The first container to define port 80 or 8080
- If neither of the above is true, the first container defined in the file will be accessible (exposed)

How do I use depends_on?

The `depends_on` option is *unsupported* on App Service and is ignored. Just as the control startup and shutdown recommendation from Docker, App Service Multi-container apps should check dependencies through application code - both at startup and disconnection. The example code below shows a Python app checking to see if a Redis container is running.

Python

```
import time
import redis
from flask import Flask
app = Flask(__name__)
cache = redis.Redis(host='redis', port=6379)
def get_hit_count():
    retries = 5
    while True:
        try:
            return cache.incr('hits')
        except redis.exceptions.ConnectionError as exc:
            if retries == 0:
                raise exc
            retries -= 1
            time.sleep(0.5)
@app.route('/')
def hello():
    count = get_hit_count()
    return 'Hello from Azure App Service team! I have been seen {}
times.\n'.format(count)
if __name__ == "__main__":
    app.run(host="0.0.0.0", port=80, debug=True)
```

Web Sockets

Web Sockets are supported on Linux apps. The `webSocketsEnabled` ARM setting doesn't apply to Linux apps since Web Sockets are always enabled for Linux.

> **Important**
>
> Web Sockets are now supported for Linux apps on Free App Service plans. We support up to five web socket connections on Free App Service plans. Exceeding this limit results in an HTTP 429 (Too Many Requests) response.

Pricing and SLA

What is the pricing, now that the service is generally available?

Pricing varies by SKU and region but you can see more details at our pricing page: App Service Pricing.

Other questions

How does the container warmup request work?

When Azure App Services starts your container, the warmup request sends an HTTP request to the /robots933456.txt endpoint of your application. This is simply a dummy endpoint, but your application must reply by returning any status code (including 5*xx*). If your application logic doesn't reply by sending an HTTP status code to nonexistent endpoints, the warmup request can't receive a response. Therefore, it perpetually restarts your container.

To change from the default behavior, you can customize the warmup endpoint path and status codes that consider the site to be warmed up. To do this, set the WEBSITE_WARMUP_PATH and WEBSITE_WARMUP_STATUSES application settings.

The warmup request also might fail because of port misconfiguration.

To make sure that the port is correctly configured on Azure App Services, see the question, *How do I specify the port in my Linux container?*

Is it possible to increase the container warmup request time-out?

The warmup request by default fails after waiting 240 seconds for a reply from the container. You may increase the container warmup request time-out by adding the application setting `WEBSITES_CONTAINER_START_TIME_LIMIT` with a value between 240 and 1800 seconds.

How do I specify the port in my Linux container?

Expand table

Container type	Description	How to set/use port
Built-in containers	If you select a language/framework version for a Linux app, a predefined container is selected for you.	To point your app code to the right port, use the PORT environment variable.
Custom containers	You have full control over the container.	App Service has no control about which port your container listens on. What it does need is to know which port to forward requests to. If your container listens to port 80 or 8080, App Service is able to automatically detect it. If it listens to any other port, you need to set the WEBSITES_PORT app setting to the port number, and App Service forwards requests to that port in the container. The WEBSITES_PORT app setting doesn't have any effect within the container, and you can't access it as an environment variable within the container.

Can I use a file based database (like SQLite) with my Linux Webapp?

The file system of your application is a mounted network share. This enables scale out scenarios where your code needs to be executed across multiple hosts. Unfortunately this blocks the use of file-based database providers like SQLite since it's not possible to acquire exclusive locks on the database file. We recommend a managed database service: Azure SQL, Azure Database for MySQL or Azure Database for PostgreSQL

What are the supported characters in application settings names?

You can use only letters (A-Z, a-z), numbers (0-9), and the underscore character (_) for application settings.

Where can I request new features?

You can submit your idea at the Web Apps feedback forum. Add "[Linux]" to the title of your idea.

Next steps

- What is Azure App Service on Linux?
- Set up staging environments in Azure App Service
- Continuous Deployment with Web App for Containers
- Things You Should Know: Web Apps and Linux
- Environment variables and app settings reference

Contact us for help

If you have questions or need help, create a support request, or ask Azure community support. You can also submit product feedback to Azure feedback community.

Deployment FAQs for Web Apps in Azure

Article • 06/23/2022

This article has answers to frequently asked questions (FAQs) about deployment issues for the Web Apps feature of Azure App Service.

If your Azure issue is not addressed in this article, visit the Azure forums on MSDN and Stack Overflow. You can post your issue in these forums, or post to @AzureSupport on Twitter. You also can submit an Azure support request. To submit a support request, on the Azure support page, select **Get support**.

I'm just getting started with App Service web apps. How do I publish my code?

Here are some options for publishing your web app code:

- Deploy by using Visual Studio. If you have the Visual Studio solution, right-click the web application project, and then select **Publish**.
- Deploy by using an FTP client. In the Azure portal, download the publish profile for the web app that you want to deploy your code to. Then, upload the files to \site\wwwroot by using the same publish profile FTP credentials.

For more information, see Deploy your app to App Service.

I see an error message when I try to deploy from Visual Studio. How do I resolve this error?

If you see the following message, you might be using an older version of the SDK:

> Error during deployment for resource 'YourResourceName' in resource group 'YourResourceGroup': MissingRegistrationForLocation: The subscription is not registered for the resource type 'components' in the location 'Central US'. Re-register for this provider in order to have access to this location.

To resolve this error, upgrade to the latest SDK. If you see this message and you have the latest SDK, submit a support request.

How do I deploy an ASP.NET application from Visual Studio to App Service?

The tutorial Create your first ASP.NET web app in Azure in five minutes shows you how to deploy an ASP.NET web application to a web app in App Service by using Visual Studio.

What are the different types of deployment credentials?

App Service supports two types of credentials for local Git deployment and FTP/S deployment. For more information about how to configure deployment credentials, see Configure deployment credentials for App Service.

What is the file or directory structure of my App Service web app?

For information about the file structure of your App Service app, see File structure in Azure.

How do I resolve "FTP Error 550 - There isn't enough space on the disk" when I try to FTP my files?

If you see this message, it's likely that you're running into a disk quota in the service plan for your web app. You might need to scale up to a higher service tier based on your disk space needs. For more information about pricing plans and resource limits, see App Service pricing.

How do I set up continuous deployment for my App Service web app?

You can set up continuous deployment from several resources, including Azure DevOps, OneDrive, GitHub, Bitbucket, Dropbox, and other Git repositories. These options are available in the portal. Continuous deployment to App Service is a helpful tutorial that explains how to set up continuous deployment.

How do I troubleshoot issues with continuous deployment from GitHub and Bitbucket?

For help investigating issues with continuous deployment from GitHub or Bitbucket, see Investigating continuous deployment.

I can't FTP to my site and publish my code. How do I resolve this issue?

To resolve FTP issues, follow these steps:

1. Verify that you're entering the correct host name and credentials. For detailed information about different types of credentials and how to use them, see Deployment credentials.
2. Verify that the FTP ports aren't blocked by a firewall. The ports should have these settings:

    - FTP control connection port: 21
    - FTP data connection port: 989, 10001-10300

How do I publish my code to App Service?

The Azure Quickstart is designed to help you deploy your app by using the deployment stack and method of your choice. To use the Quickstart, in the Azure portal, go to your app service, under **Deployment**, select **Quickstart**.

Why does my app sometimes restart after deployment to App Service?

To learn about the circumstances under which an application deployment might result in a restart, see Deployment vs. runtime issues. As the article describes, App Service deploys files to the wwwroot folder. It never directly restarts your app.

How do I integrate Azure DevOps code with App Service?

You have two options for using continuous deployment with Azure DevOps:

- Use a Git project. Connect via App Service by using the Deployment Center.
- Use a Team Foundation Version Control (TFVC) project. Deploy by using the build agent for App Service.

Continuous code deployment for both these options depends on existing developer workflows and check-in procedures. For more information, see these articles:

- Implement continuous deployment of your app to an Azure website
- Set up an Azure DevOps organization so it can deploy to a web app

How do I use FTP or FTPS to deploy my app to App Service?

For information about using FTP or FTPS to deploy your web app to App Service, see Deploy your app to App Service by using FTP/S.

Contact us for help

If you have questions or need help, create a support request, or ask Azure community support. You can also submit product feedback to Azure community support.

Feedback

Frequently asked questions about App Service Environment migration

FAQ

This article answers common questions about App Service Environment (ASE) migration from ASE v1/v2 to ASE v3, and provides solutions to common errors that might occur during the migration pre-check process.

If your Azure issue is not addressed in this article, visit the Azure forums on MSDN and Stack Overflow. You can post your issue in these forums, or post to @AzureSupport on Twitter. You also can submit an Azure support request. To submit a support request, on the Azure support page, select **Get support**.

Common questions

How do I know which migration option is right for me?

Review the migration path decision tree to decide which option is best for your use case.

For customers who want to migrate to ASE v3 with minimal changes to their networking configurations, the in-place migration feature is best. It can support about one hour of application downtime. The in-place migration feature creates your ASE v3 in the same subnet as your existing environment and uses the same networking infrastructure. You might have to consider the inbound and outbound IP address changes if you have any dependencies on these specific IPs. The in-place migration feature will also shut down and delete the old environment, and all your apps will be migrated to the new environment. Your old environment is no longer accessible. A rollback to the old environment isn't possible.

If you can't support downtime, see the side-by-side migration feature. The side-by-side feature creates a new ASE v3 with all your apps in a different subnet. Your existing ASE isn't deleted until you initiate its deletion at the end of the migration process. Because of this process, there's a rollback option if you need to cancel your migration. This migration option is best for customers who want to migrate to ASE v3 with zero downtime, and it can support using a different subnet for their new environments. For manual migration options that allow you to migrate at your own pace, see manual migration options.

Will there be any downtime during the migration?

It depends on the migration tool you use. With the in-place migration feature, you'll experience about one hour of downtime during the 4 to 6-hour service window required for this migration process, so plan accordingly. When you migrate using the in-place migration feature, if you have a different ASE that you can direct traffic to, you can eliminate application downtime.

With the side-by-side migration feature, you won't experience any downtime. Your apps continue to run on your existing ASE until you complete the final step of the migration where DNS changes are effective immediately. Once you complete the final step, your old ASE is shut down and deleted.

Are there any prerequisites or limitations before migrating?

It depends on the migration tool you use. Check the documentation for the two migration tools, the in-place migration feature or the side-by-side migration feature, to determine if there are prerequisites for the migration tool or feature. If your ASE doesn't support migration with the migration tool, you must use one of the manual methods to migrate to ASE v3.

Do I need to do anything to my apps after the migration so that they can run on the new ASE?

No, all your apps running on the old environment are automatically migrated to the new environment and run like before. No user input is needed.

What properties of my ASE will change?

You're on ASE v3, so be sure to review the features and how they differ from earlier versions.

For ILB ASE, you keep the same ILB IP address. For internet-facing ASE, the public and outbound IP addresses change. For ELB ASE, there was previously a single IP for both inbound and outbound. For ASE v3, they're separate. For more information, see ASE v3 networking. For a full comparison of the ASE version, see ASE version comparison.

Will my IPs change?

The platform will create the new inbound IP (if you're migrating an ELB ASE) and outbound IP addresses. When these IPs are created, the activity with your existing ASE isn't interrupted. However, you can't scale or change your existing environment. This process takes about 15 minutes to complete.

When completed, you'll get the new IPs that your future ASE v3 uses. These new IPs don't affect your existing environment. The IPs used by your existing environment continue to be used up until your existing environment is shut down by the in-place migration feature or by you when your ASE v1/v2 is no longer needed.

Is there any cost to migrate my ASE?

There's no cost to migrate your ASE. You no longer have to pay for your previous ASE as soon as it's shut down during the migration process, and you start paying for your new ASE v3 as soon as it's deployed. Migrating ASE from an earlier version to v3 can help reduce your monthly cost. For more information about ASE v3 pricing, see the pricing details.

What if my ASE has a custom domain suffix?

The migration tool supports this scenario. For more information, see the in-place migration feature and side-by-side migration feature.

What if my ASE is zone-pinned?

Zone-pinned ASE v2 is a supported scenario for migration using the in-place migration feature. ASE v3 doesn't support zone pinning. When migrating to ASE v3, you can choose whether to configure zone redundancy. The side-by-side migration feature currently doesn't support this migration scenario.

What if my ASE has IP SSL addresses?

ASE v3 doesn't support IP SSL. You must remove all IP SSL bindings before migrating using the migration feature or one of the manual options. If you intend to use the migration tool, once you remove all IP SSL bindings, you pass the validation check and can proceed with the automated migration.

Is backup and restore supported for moving apps from ASE v2 to v3?

Yes. The backup and restore feature allows you to keep your app configuration, file content, and database connected to your app when migrating to your new environment. Make sure you review the details of this feature.

You can select a custom backup and restore it to an App Service in your ASE v3. You must create the App Service you'll restore to before restoring the app. You can choose to restore the backup to the production slot, an existing slot, or a newly created slot that you can create during the restoration process. For more information, see Migration alternatives, backup, and restore.

What will happen to my ASE v1/v2 resources after August 31, 2024?

After August 31, 2024, if you haven't migrated, your ASE v1/v2 and the apps deployed on them will no longer be available. ASE v1/v2 is hosted on App Service scale units running on Cloud Services (classic) architecture that will be retired on August 31, 2024. Because of this, ASE v1/v2 will no longer be available after that date. Migrate to ASE v3 to keep your apps running or to save or back up any resources or data that you need to maintain.

What should I expect during the migration?

Migration consists of a series of steps that must be followed in order. Key points are given for a subset of the steps. It's important to understand what happens during these steps and how your environment and apps are impacted. These steps differ slightly depending on which migration feature you execute. For more information, see Overview of the in-place migration process or Overview of the side-by-side migration process. After reviewing the process overview information, when you're ready to migrate, follow the guide for the migration process you're executing, either the in-place migration guide or the side-by-side migration guide.

Migration requires a 4 to 6-hour service window for ASE v2 to v3 migrations. Depending on the size of the environment migrating from v1 to v3, a service window up to six hours is required during migration, scaling, and environment configurations. When the migration is complete, the apps on the old ASE will run on the new ASE v3.

What if the migration fails or an unexpected issue occurs during the migration?

If an unexpected issue occurs, support teams are on hand. You should migrate development environments before touching any production environments to learn about the migration process and see how it impacts your workloads.

Common errors

ASEv3 Migration isn't yet ready.

This error occurs because the underlying infrastructure isn't ready to support ASE v3. For more information about the migration tools and to review the migration path decision tree, see What tooling is available to help with the upgrade to App Service Environment v3? If the migration tools aren't available for your situation and you want to migrate immediately, we recommend using one of the manual migration options.

Migration is invalid. Your ASE needs to be upgraded to the latest build to ensure successful migration.

This error occurs because your ASE isn't on the minimum build required for migration. An upgrade will be started, and your ASE won't be impacted, but you won't be able to scale or change your ASE during the upgrade progress. You won't be able to migrate until the upgrade finishes. Build upgrades may take some time to complete. We recommend waiting until the upgrade finishes and before migrating.

Migrate can't be called on this ASE until the active upgrade has finished.

This error occurs because ASEs can't be migrated during platform upgrades. You can set your upgrade preference from the Azure portal. In some cases, an upgrade is initiated when accessing the migration page if your ASE isn't on the current build.

ASE management operation in progress.

This error occurs because your ASE is undergoing a management operation. These operations can include activities such as deployments or upgrades. Migration is blocked until these operations are complete. Management operations may take some time to complete. We recommend waiting until these operations are complete before migrating.

Migrate can't be called if IP SSL is enabled on any of the sites.

This error occurs because your ASE has IP SSL-enabled sites that can't be migrated. ASE v3 doesn't support IP SSL. You must remove all IP SSL bindings before migrating using the migration features or one of the manual options. If you intend to use the migration tool, once you remove all IP SSL bindings and pass the validation check, you can proceed with the automated migration.

Your InternalLoadBalancingMode isn't currently supported.

This error occurs because ASEs that have `InternalLoadBalancingMode` set to specific values can't currently be migrated using the migration feature. We recommend migrating using one of the manual migration options if you want to migrate immediately.

<ZoneRedundant><DedicatedHosts><ASEv3.ASE> isn't available in this location.

This error occurs if you try to migrate an ASE in a region that doesn't support one of your requested features. We recommend migrating using one of the manual migration options if you want to migrate immediately. Otherwise, wait for the migration feature to support this ASE configuration.

Subscription has too many ASEs. Please remove some before trying to create more.

This error occurs because the ASE quota for your subscription has been reached. We recommend removing unneeded environments or contacting support to review your options.

Migrate isn't available for this subscription.

We recommend opening a support case to resolve your issue.

Full migration can't be called before IP addresses are generated.

This error occurs if you try to migrate before finishing the pre-migration steps. We recommend that you ensure you have completed all pre-migration steps before you try to migrate. For more information, see the step-by-step guide for migrating.

Contact us for help

If you have questions or need help, create a support request, or ask Azure community support. You can also submit product feedback to Azure feedback community.

Feedback

Was this page helpful? Yes No

Frequently asked questions about scaling web apps in Azure App Service

This article answers common questions about scaling web apps in Azure App Service.

If your Azure issue is not addressed in this article, visit the Azure forums on MSDN and Stack Overflow. You can post your issue in these forums, or post to @AzureSupport on Twitter. You also can submit an Azure support request. To submit a support request, on the Azure support page, select **Get support**.

How do I scale up a Web App?

You can scale up a Web App by using the Azure portal. On your Web App page, select **Scale Up (App Service Plan)** from the left menu. For more information, see Scale up an app in Azure App Service.

Is there any further action I need to perform before scaling up a Free App Service Plan?

Before you switch an App Service Plan from the Free tier, you must remove the spending limits in place for your Azure subscription. To view or change options for your Microsoft Azure App Service subscription, see Microsoft Azure Subscriptions - Remove spending limits.

Is there any instance limitation when scaling a Web App?

Yes, the limitation depends on the tier of the App Service Plan. For more information, see App Service limits.

Can I scale a Standard App Service Plan for more than 10 instances?

The Standard App Service Plan tier doesn't support more than 10 instances. You can move to a Premium App Service Plan and get the benefit of having 30 instances (in selected regions) per App Service Plan. To check the instance limit per pricing tier, see App Service limits.

Can I configure App Services on the same App Service Plan with a different number of instances?

You can configure it by enabling Per-App Scaling, and changing App Service's **numberOfWorkers** property to the desired instance count. For more information, see Per-App Scaling.

Why does scale up for my Web App also trigger scale up for another Web App?

Scaling happens on the entire App Service Plan. If you host multiple App Services in the same App Service Plan, all App Services in this App Service Plan will be scaled up.

Why isn't autoscale working as expected?

You might be running into a scenario in which we intentionally choose not to scale to avoid an infinite loop due to "flapping." This behavior usually happens when there isn't an adequate margin between the scale-out and scale-in thresholds. For more information about how to avoid "flapping" and other autoscale best practices, see Autoscale best practices.

How do I determine when an autoscale rule triggered scaling?

You can retrieve scale history from the activity log. Whenever your resource is scaled up or down, an event is logged in the activity log. You can view the scale history of your resource for the past 24 hours by switching to the **Run history** tab. To view the complete scale history (for up to 90 days), select **Click here to see more details**. For more information, see View the scale history of your resource.

Why does autoscale sometimes scale only partially?

Autoscale is triggered when metrics exceed preconfigured boundaries. Sometimes, you might notice that the capacity is only partially filled compared to what you expected. This behavior

might occur when the number of instances you want isn't available. In that scenario, autoscale partially fills in with the available number of instances. Autoscale then runs the rebalance logic to get more capacity. It allocates the remaining instances, and this allocation might take a few minutes. If you don't see the expected number of instances after a few minutes, it might be because the partial refill was enough to bring the metrics within the boundaries. Or, autoscale might have scaled down because it reached the lower metrics boundary.

When I scale up an App Service Plan to a Premium V3 tier, the "Premium V3 isn't supported for this scale unit. Please consider redeploying or cloning your app." error occurs. What should I do?

The Premium V3 feature requires the site to run on the newest hardware infrastructure. To scale up an App Service Plan to Premium V3, the Web App must be running in an App Service deployment that supports PremiumV3. For more information, see Scale up from an unsupported resource group and region combination.

I'm unable to scale up/scale down the App Service Plan due to the "You have exceeded the maximum amount of scale changes within the past hour (XX changes and limit is XX)" error. What should I do?

To avoid this issue, don't perform scaling operations that release more than XX instances in an hour. Every time you release an instance during a scale-down operation, the instance is rebooted to ensure the next App Service Plan can get a clean instance. When you perform too many scaling operations in quick succession, instance reboots can cause performance issues for other App Services. Therefore we intentionally put a throttling mechanism for scaling that prevents you from executing scaling operations more than the acceptable limit in quick succession.

My Web App is using the Diagnostic setting "AppServiceFileAuditLogs" and I'm unable to scale the App Service Plan from Premium V2 to the Basic tier. What should I do?

The file change audit logs "AppServiceFileAuditLogs" are only available for App Services in Premium, PremiumV2, and Isolated App Service Plans. If you need "AppServiceFileAuditLogs," you won't be able to scale down to the Basic tier. To have these audit logs available, configure your App Service Plan for Premium or higher tier.

I'm getting the "App Service Plans with fewer than 3 workers aren't allowed for zone redundancy. Requested number of workers: number" error. What should I do?

Availability zone support requires at least three instances. Verify if the App Service Plan has zone redundancy enabled and if you have autoscale on the App Service Plan. If so, correct the autoscale rule to not set the number of instances to a value less than three.

Contact us for help

If you have questions or need help, create a support request, or ask Azure community support. You can also submit product feedback to Azure feedback community.

Set up an existing custom domain in Azure App Service

Article • 02/14/2025

> **Note**
>
> Starting June 1, 2024, newly created App Service apps can generate a unique default hostname that uses the naming convention `<app-name>-<random-hash>.<region>.azurewebsites.net`. Existing app names remain unchanged. For example:
>
> `myapp-ds27dh7271aah175.westus-01.azurewebsites.net`
>
> For more information, see Unique Default Hostname for App Service Resource.

Azure App Service provides a highly scalable, self-patching web hosting service. This guide shows you how to map an existing custom Domain Name System (DNS) name to App Service. To migrate a live site and its DNS domain name to App Service with no downtime, see Migrate an active DNS name to Azure App Service.

The DNS record type you need to add with your domain provider depends on the domain you want to add to App Service.

Expand table

Scenario	Example	Recommended DNS record
Root domain	contoso.com	A record. Don't use the CNAME record for the root record. (For information, see RFC 1912, section 2.4.)
Subdomain	www.contoso.com, my.contoso.com	CNAME record. You can map a subdomain to the app's IP address directly with an A record, but it's possible for the IP address to change. The CNAME maps to the app's default hostname instead, which is less susceptible to change.
Wildcard	*.contoso.com	CNAME record.

> **Note**

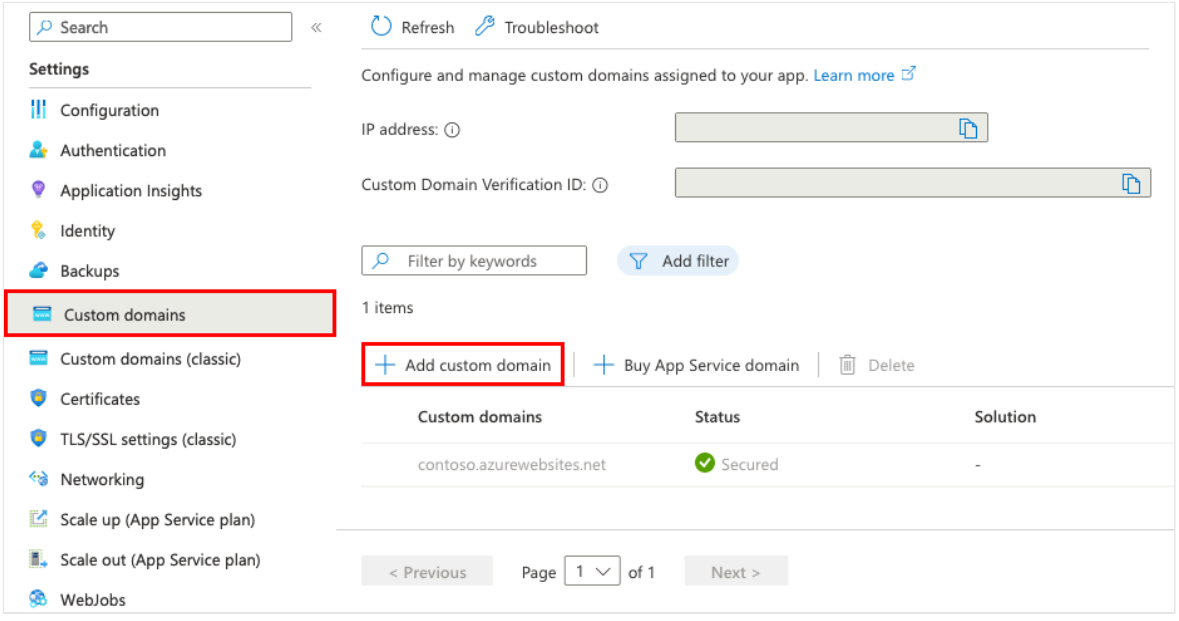
For an end-to-end tutorial that shows you how to configure a `www` subdomain and a managed certificate, see Tutorial: Secure your Azure App Service app with a custom domain and a managed certificate.

Prerequisites

- Create an App Service app, or use an app that you created for another tutorial. The web app's App Service plan must be a paid tier, not the Free (F1) tier. See Scale up an app to update the tier.
- Make sure you can edit the DNS records for your custom domain. To edit DNS records, you need access to the DNS registry for your domain provider, such as GoDaddy. For example, to add DNS entries for `contoso.com` and `www.contoso.com`, you must be able to configure the DNS settings for the `contoso.com` root domain. Your custom domains must be in a public DNS zone; private DNS zones are not supported.
- If you don't have a custom domain yet, you can purchase an App Service domain instead.

Configure a custom domain

1. In the Azure portal, navigate to your app's management page.
2. In the left menu for your app, select **Custom domains**.
3. Select **Add custom domain**.



4. For **Domain provider**, select **All other domain services** to configure a third-party domain.

   > **Note**
   >
   > To configure an App Service domain, see Buy a custom domain name for Azure App Service.

5. For **TLS/SSL certificate**, select **App Service Managed Certificate** if your app is in the Basic tier or higher. If you want to remain in the Shared tier, or if you want to use your own certificate, select **Add certificate later**.

6. For **TLS/SSL type**, select the binding type you want.

   Expand table

Setting	Description
Custom domain	The domain name to add the TLS/SSL binding for.
Private Certificate Thumbprint	The certificate to bind.
TLS/SSL Type	- **SNI SSL**: Multiple SNI SSL bindings may be added. This option allows multiple TLS/SSL certificates to secure multiple domains on the same IP address. Most modern browsers (including Internet Explorer, Chrome, Firefox, and Opera) support SNI (for more information, see Server Name Indication). - **IP SSL**: Only one IP SSL binding may be added. This option allows only one TLS/SSL certificate to secure a dedicated public IP address. After you configure the binding, follow the steps in Remap records for IP based SSL. IP SSL is supported only in **Standard** tier or above.

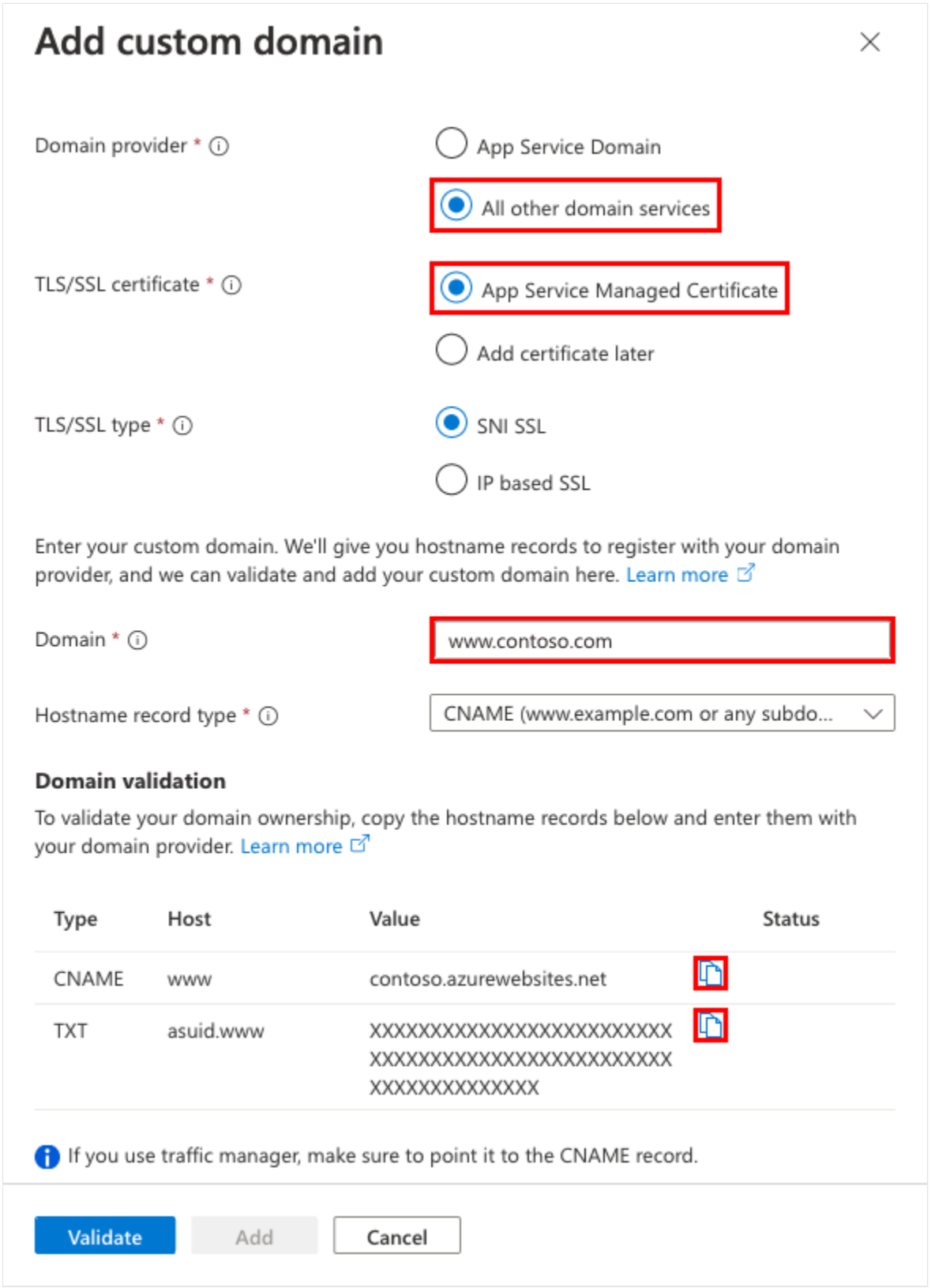
7. For **Domain**, specify a fully qualified domain name you want based on the domain you own. The **Hostname record type** box defaults to the recommended DNS record to use, depending on whether the domain is a root domain (like `contoso.com`), a subdomain (like `www.contoso.com`), or a wildcard domain (like `*.contoso.com`).

8. Don't select **Validate** yet.

9. For each custom domain in App Service, you need two DNS records with your domain provider. The **Domain validation** section shows you two DNS records that

you must add with your domain provider. You can use the copy buttons to copy the value or values that you need in the next section.

The following screenshot shows the default selections for a `www.contoso.com` domain. It shows a CNAME record and a TXT record to add.



> ⚠ **Warning**

While it's not absolutely required to add the TXT record, it's highly recommended for security. The TXT record is a *domain verification ID* that helps avoid subdomain takeovers from other App Service apps. For custom domains you previously configured without this verification ID, you should protect them from the same risk by adding the verification ID (the TXT record) to your DNS configuration. For more information on this common high-severity threat, see **Subdomain takeover**.

Create the DNS records

1. Sign in to the website of your domain provider.

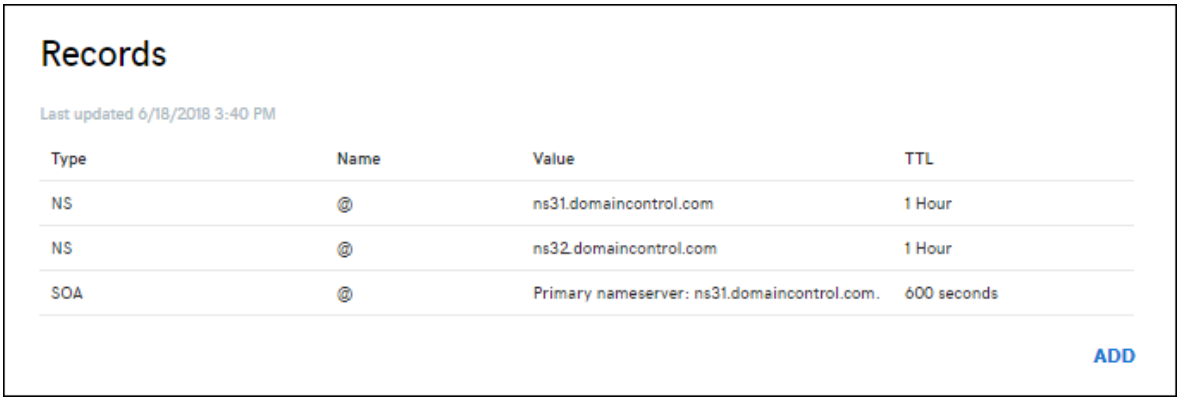
   You can use Azure DNS to manage DNS records for your domain and configure a custom DNS name for Azure App Service. For more information, see Tutorial: Host your domain in Azure DNS.

2. Find the page for managing DNS records.

   Every domain provider has its own DNS records interface, so consult the provider's documentation. Look for areas of the site labeled **Domain Name**, **DNS**, or **Name Server Management**.

   Often, you can find the DNS records page by viewing your account information and then looking for a link such as **My domains**. Go to that page, and then look for a link that's named something like **Zone file**, **DNS Records**, or **Advanced configuration**.

   The following screenshot is an example of a DNS records page:

3. Select **Add** or the appropriate widget to create a record.

> ⓘ Note

For certain providers, such as GoDaddy, changes to DNS records don't become effective until you select a separate **Save Changes** link.

Select the type of record to create and follow the instructions. You can use either a CNAME record or an A record to map a custom DNS name to App Service. When your function app is hosted in a Consumption plan, only the CNAME option is supported.

Root domain (for example, contoso.com)

Create two records, as described in the following table:

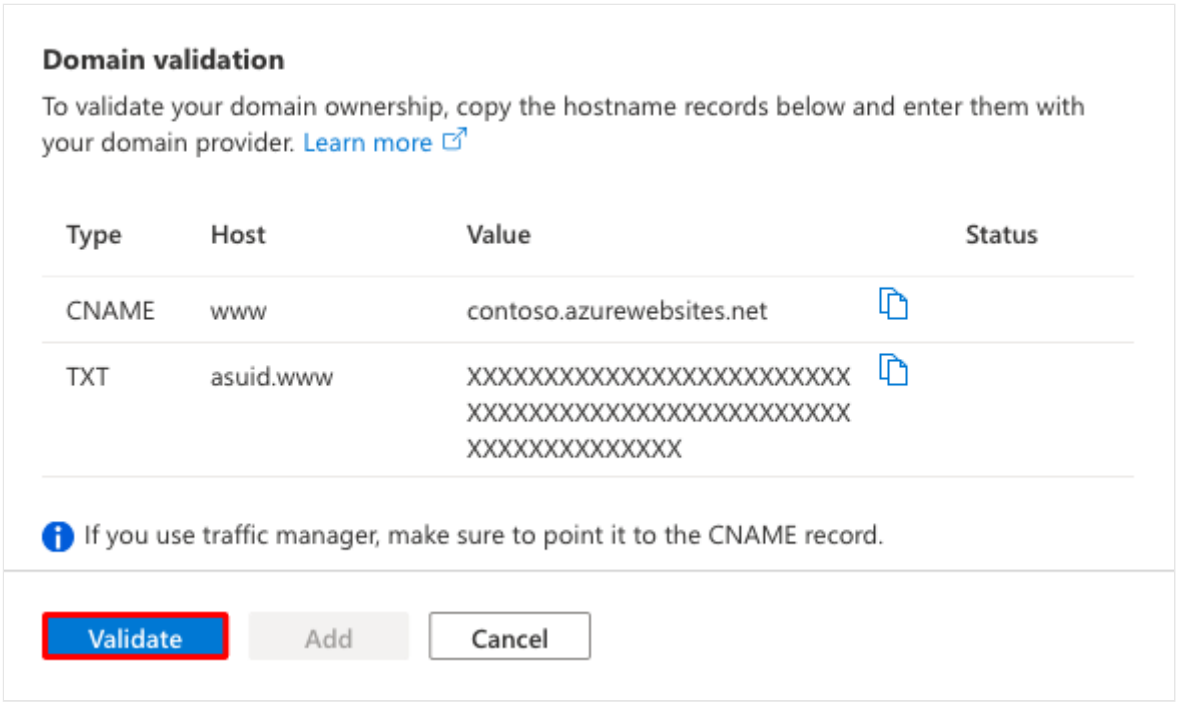
Expand table

Record type	Host	Value	Comments
A	`@`	The app's IP address shown in the **Add custom domain** dialog.	The domain mapping itself. (`@` typically represents the root domain.)
TXT	`asuid`	The domain verification ID shown in the **Add custom domain** dialog.	For the root domain, App Service accesses the `asuid` TXT record to verify your ownership of the custom domain.

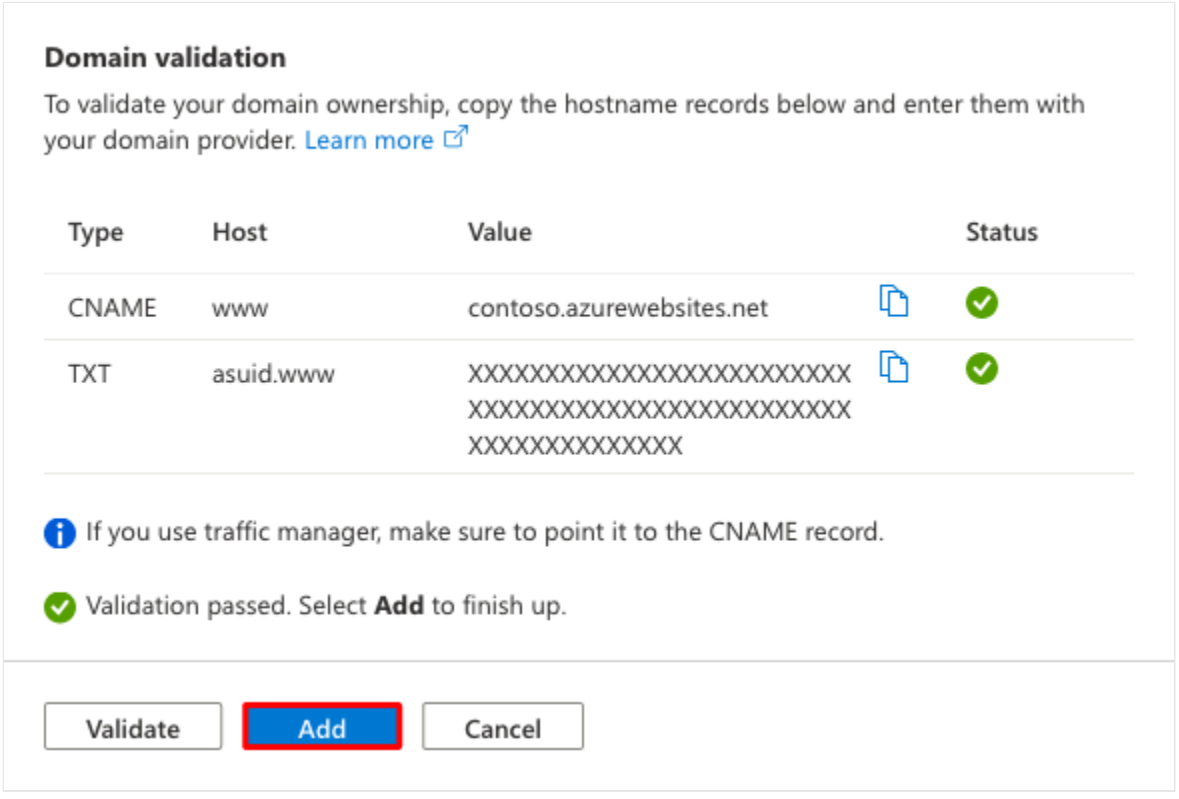
Name	Type	TTL	Value
@	A	3600	192.0.2.18
@	NS	172800	ns1-05.azure-dns.com. ns2-05.azure-dns.net. ns3-05.azure-dns.org. ns4-05.azure-dns.info.
@	SOA	3600	Email: contoso-hostmaster.microsoft.com Host: ns1-05.azure-dns.com. Refresh: 3600 Retry: 300 Expire: 2419200 Minimum TTL: 300 Serial number: 1
asuid	TXT	3600	<domain-verification-id-from-your-app>

Validate domain ownership and complete the mapping

1. Back in the **Add custom domain** dialog in the Azure portal, select **Validate**.



2. If the **Domain validation** section shows green check marks next to both domain records, you've configured them correctly. Select **Add**. If you see any errors or warnings, resolve them in the DNS record settings on your domain provider's website.

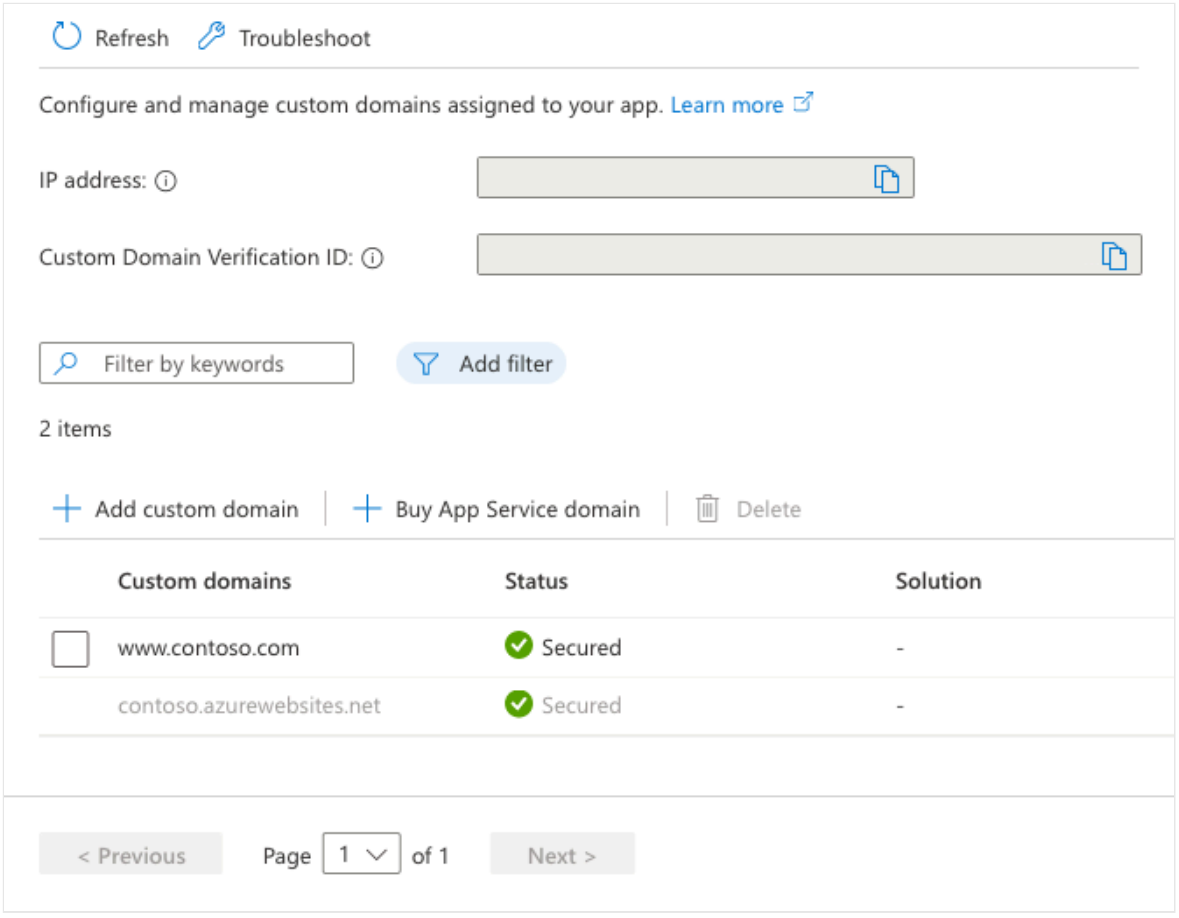


> ⓘ Note

If you configured the TXT record but not the A or CNAME record, App Service treats the change as a domain migration scenario and allows the validation to succeed, but you won't see green check marks next to the records.

3. You should see the custom domain added to the list. You might also see a red X and the text **No binding**.

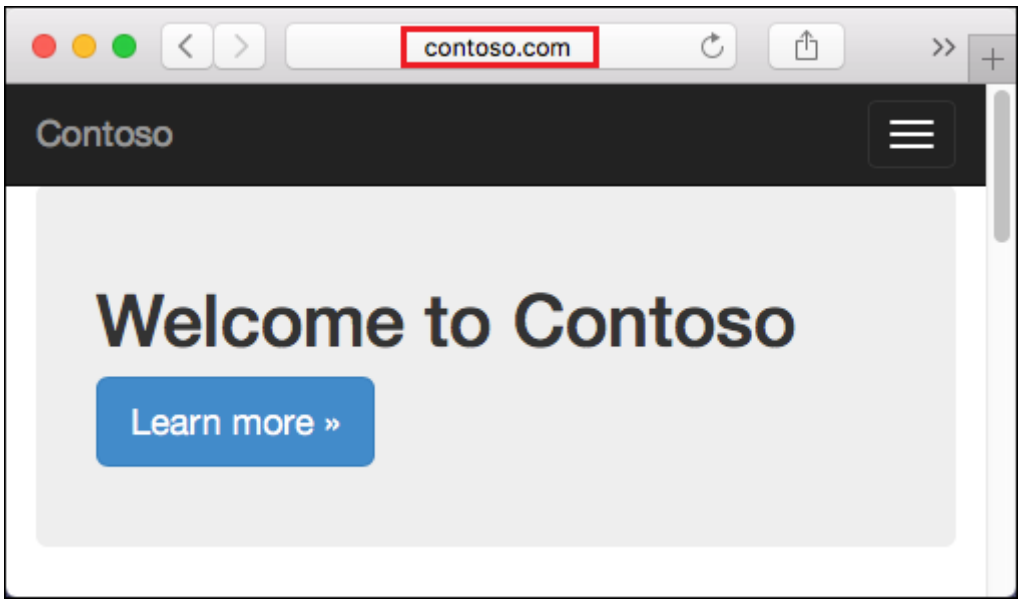
   If you selected **App Service Managed Certificate** earlier, wait a few minutes for App Service to create the managed certificate for your custom domain. When the process is complete, the red X becomes a green check mark and you see the word **Secured**. If you selected **Add certificate later**, the red X will remain until you add a private certificate for the domain and configure the binding.



> **Note**
>
> Unless you configure a certificate binding for your custom domain, any HTTPS request from a browser to the domain will receive an error or warning, depending on the browser.

Test the DNS resolution

Browse to the DNS names that you configured.



If you receive an HTTP 404 (Not Found) error when you browse to the URL of your custom domain, the two most likely causes are:

- The browser client has cached the old IP address of your domain. Clear the cache and test the DNS resolution again. On a Windows machine, you can clear the cache with `ipconfig /flushdns`.
- You configured an IP-based certificate binding, and the app's IP address has changed because of it. Remap the A record in your DNS entries to the new IP address.

If you receive a `Page not secure` warning or error, it's because your domain doesn't have a certificate binding yet. Add a private certificate for the domain and configure the binding.

(Optional) Automate with scripts

You can automate management of custom domains with scripts by using the Azure CLI or Azure PowerShell.

Azure CLI

The following command adds a configured custom DNS name to an App Service app.

Azure CLI

```
az webapp config hostname add \
    --webapp-name <app-name> \
```

```
    --resource-group <resource_group_name> \
    --hostname <fully_qualified_domain_name>
```

For more information, see Map a custom domain to a web app.

Next steps

Purchase an App Service domain

Secure a custom DNS name with a TLS/SSL binding in Azure App Service

Feedback

Was this page helpful? Yes No

Provide product feedback | Get help at Microsoft Q&A

Open-source technologies FAQs for Web Apps in Azure

Article • 03/19/2024

This article has answers to frequently asked questions (FAQs) about issues with open-source technologies for the Web Apps feature of Azure App Service.

If your Azure issue is not addressed in this article, visit the Azure forums on MSDN and Stack Overflow. You can post your issue in these forums, or post to @AzureSupport on Twitter. You also can submit an Azure support request. To submit a support request, on the Azure support page, select **Get support**.

How do I turn on PHP logging to troubleshoot PHP issues?

To turn on PHP logging, follow these steps:

1. Sign in to your **Kudu website** (`https://*yourwebsitename*.scm.azurewebsites.net`).
2. In the top menu, select **Debug Console** > **CMD**.
3. Select the **Site** folder.
4. Select the **wwwroot** folder.
5. Select the + icon, and then select **New File**.
6. Set the file name to **.user.ini**.
7. Select the pencil icon next to **.user.ini**.
8. In the file, add this code: `log_errors=on`
9. Select **Save**.
10. Select the pencil icon next to **wp-config.php**.
11. Change the text to the following code:

```php
//Enable WP_DEBUG modedefine('WP_DEBUG', true);//Enable debug logging
to /wp-content/debug.logdefine('WP_DEBUG_LOG', true);
```

```
//Suppress errors and warnings to screendefine('WP_DEBUG_DISPLAY',
false);//Suppress PHP errors to screenini_set('display_errors', 0);
```

12. In the Azure portal, in the web app menu, restart your web app.

For more information, see Enable WordPress error logs.

How do I log Python application errors in apps that are hosted in App Service?

If Python encounters an error while starting your application, only a simple error page will be returned. For example,

> The page cannot be displayed because an internal server error has occurred.

To capture Python application errors, follow these steps:

1. In the Azure portal, in your web app, select **Settings**.
2. On the **Settings** tab, select **Application settings**.
3. Under **App settings**, enter the following key/value pair:
    - Key: *WSGI_LOG*
    - Value: *D:\home\site\wwwroot\logs.txt* (enter your choice of file name)

You should now see errors in the *logs.txt* file in the *wwwroot* folder.

How do I change the version of the Node.js application that is hosted in App Service?

To change the version of the Node.js application, you can use one of the following options:

- In the Azure portal, use **App settings**.
    1. In the Azure portal, go to your web app.
    2. On the **Settings** blade, select **Application settings**.
    3. In **App settings**, you can include WEBSITE_NODE_DEFAULT_VERSION as the key, and the version of Node.js you want as the value.

4. Go to your **Kudu console** (`https://*yourwebsitename*.scm.azurewebsites.net`).

5. To check the Node.js version, enter the following command:

   Node.js

   ```
   node -v
   ```

- Modify the *iisnode.yml* file. Changing the Node.js version in the *iisnode.yml* file only sets the runtime environment that iisnode uses. Your Kudu cmd and others still use the Node.js version that is set in **App settings** in the Azure portal.

  To set the *iisnode.yml* manually, create an *iisnode.yml* file in your app root folder. In the file, include the following line:

  yml

  ```
  nodeProcessCommandLine: "D:\Program Files (x86)\nodejs\5.9.1\node.exe"
  ```

- Set the *iisnode.yml* file by using *package.json* during source control deployment. The Azure source control deployment process involves the following steps:

  1. Moves content to the Azure web app.

  2. Creates a default deployment script, if there isn't one (*deploy.cmd*, *.deployment* files) in the web app root folder.

  3. Runs a deployment script in which it creates an *iisnode.yml* file if you mention the Node.js version in the *package.json* file > engine `"engines": {"node": "5.9.1","npm": "3.7.3"}`

  4. The *iisnode.yml* file has the following line of code:

     yml

     ```
     nodeProcessCommandLine: "D:\Program Files
     (x86)\nodejs\5.9.1\node.exe"
     ```

I see the message "Error establishing a database connection" in my WordPress app

that's hosted in App Service. How do I troubleshoot this error?

If you see this error in your Azure WordPress app, to enable *php_errors.log* and *debug.log*, complete the steps detailed in Enable WordPress error logs.

When the logs are enabled, reproduce the error, and then check the logs to see if you're running out of connections:

Output

```
[09-Oct-2015 00:03:13 UTC] PHP Warning: mysqli_real_connect(): (HY000/1226):
User 'abcdefghijk79' has exceeded the 'max_user_connections' resource
(current value: 4) in D:\home\site\wwwroot\wp-includes\wp-db.php on line
1454
```

If you see this error in your *debug.log* or *php_errors.log* files, your app is exceeding the number of connections. If you're hosting on ClearDB, verify the number of connections that are available in your service plan.

How do I debug a Node.js app that's hosted in App Service?

1. Go to your **Kudu console** (`https://*yourwebsitename*.scm.azurewebsites.net/DebugConsole`).
2. Go to your application logs folder (*D:\home\LogFiles\Application*).
3. In the *logging_errors.txt* file, check for content.

How do I install native Python modules in an App Service web app or API app?

Some packages might not install by using pip in Azure. The package might not be available on the Python Package Index, or a compiler might be required (a compiler isn't available on the computer that is running the web app in App Service). For information about installing native modules in App Service web apps and API apps, see Install Python modules in App Service.

How do I deploy a Django app to App Service by using Git and the new version of Python?

For information about installing Django, see Deploying a Django app to App Service.

Where are the Tomcat log files located?

For Azure Marketplace and custom deployments:

- Folder location: *D:\home\site\wwwroot\bin\apache-tomcat-8.0.33\logs*
- Files of interest:
  - *catalina.<yyyy-mm-dd>.log*
  - *host-manager.<yyyy-mm-dd>.log*
  - *localhost.<yyyy-mm-dd>.log*
  - *manager.<yyyy-mm-dd>.log*
  - *site_access_log.<yyyy-mm-dd>.log*

For portal **App settings** deployments:

- Folder location: *D:\home\LogFiles*
- Files of interest:
  - *catalina.<yyyy-mm-dd>.log*
  - *host-manager.<yyyy-mm-dd>.log*
  - *localhost.<yyyy-mm-dd>.log*
  - *manager.<yyyy-mm-dd>.log*
  - *site_access_log.<yyyy-mm-dd>.log*

How do I troubleshoot JDBC driver connection errors?

You might see the following message in your Tomcat logs:

> The web application[ROOT] registered the JDBC driver [com.mysql.jdbc.Driver] but failed to unregister it when the web application was stopped. To prevent a memory leak,the JDBC Driver has been forcibly unregistered

To resolve the error, follow these steps:

1. Remove the *sqljdbc*.jar* file from your *app/lib* folder.
2. If you're using the custom Tomcat or Azure Marketplace Tomcat web server, copy this .jar file to the Tomcat lib folder.
3. If you're enabling Java from the Azure portal (select **Java 1.8** > **Tomcat server**), copy the *sqljdbc*.jar* file in the folder that's parallel to your app. Then, add the

following classpath setting to the *web.config* file:

```
XML
```

```
<httpPlatform>
    <environmentVariables>
        <environmentVariablename ="JAVA_OPTS" value=" -
Djava.net.preferIPv4Stack=true
-Xms128M -classpath %CLASSPATH%;[Path to the sqljdbc*.jarfile]" />
    </environmentVariables>
</httpPlatform>
```

Why do I see errors when I attempt to copy live log files?

If you try to copy live log files for a Java app (for example, Tomcat), you might see this FTP error:

> Error transferring file [filename] Copying files from remote side failed.
>
> The process cannot access the file because it is being used by another process.

The error message might vary, depending on the FTP client.

All Java apps have this locking issue. Only Kudu supports downloading this file while the app is running.

Stopping the app allows FTP access to these files.

Another workaround is to write a WebJob that runs on a schedule and copies these files to a different directory. For a sample project, see the CopyLogsJob project.

Where do I find the log files for Jetty?

For Marketplace and custom deployments, the log file is in the *D:\home\site\wwwroot\bin\jetty-distribution-9.1.2.v20140210\logs* folder. The folder location depends on the version of Jetty you're using. For example, the path provided here is for Jetty 9.1.2. Look for *jetty_<YYYY_MM_DD>.stderrout.log*.

For portal App Setting deployments, the log file is in *D:\home\LogFiles*. Look for *jetty_<YYYY_MM_DD>.stderrout.log*.

Can I send email from my Azure web app?

App Service doesn't have a built-in email feature. For some good alternatives for sending email from your app, see this Stack Overflow discussion.

Why does my WordPress site redirect to another URL?

If you have recently migrated to Azure, WordPress might redirect to the old domain URL. This issue is caused by a setting in the MySQL database.

WordPress Buddy+ is an Azure Site Extension that you can use to update the redirection URL directly in the database.

Alternatively, if you prefer to manually update the redirection URL by using SQL queries or PHPMyAdmin, see WordPress: Redirecting to wrong URL.

How do I change my WordPress sign-in password?

If you have forgotten your WordPress sign-in password, you can use WordPress Buddy+ to update it.

I can't sign in to WordPress. How do I resolve this issue?

If you find yourself locked out of WordPress after recently installing a plugin, you might have a faulty plugin. WordPress Buddy+ is an Azure Site Extension that can help you disable plugins in WordPress.

How do I migrate my WordPress database?

You have multiple options for migrating the MySQL database that's connected to your WordPress website:

- Developers: Use the command prompt or PHPMyAdmin

How do I help make WordPress more secure?

To learn about security best practices for WordPress, see Best practices for WordPress security in Azure.

I'm trying to use PHPMyAdmin, and I see the message "Access denied." How do I resolve this issue?

You might experience this issue if the MySQL in-app feature isn't running yet in this App Service instance. To resolve the issue, try to access your website. This starts the required processes, including the MySQL in-app process. To verify that MySQL in-app is running, in Process Explorer, ensure that *mysqld.exe* is listed in the processes.

After you ensure that MySQL in-app is running, try to use PHPMyAdmin.

I get an HTTP 403 error when I try to import or export my MySQL in-app database by using PHPMyadmin. How do I resolve this issue?

If you're using an older version of Chrome, you might be experiencing a known bug. To resolve the issue, upgrade to a newer version of Chrome. Also try using a different browser, like Internet Explorer or Microsoft Edge, where the issue doesn't occur.

Contact us for help

If you have questions or need help, create a support request, or ask Azure community support. You can also submit product feedback to Azure feedback community.

Feedback

How to troubleshoot temporary storage on Azure App Service

Article • 06/23/2022

If an application is reporting high resource consumption, the source of the problem might not be the site content. Slow performance can also be caused by excessive use of temporary storage.

This video demonstrates how to narrow the scope of your troubleshooting to temporary storage, and the next steps for investigating and troubleshooting performance issues.

> [!VIDEO https://www.youtube.com/embed/bk8h-VYaIXs]

Related content

 For more information, see: Understanding the Azure App Service file system - projectkudu/kudu Wiki (github.com)

Contact us for help

If you have questions or need help, create a support request, or ask Azure community support. You can also submit product feedback to Azure community support.

Feedback

Was this page helpful? Yes No

Provide product feedback | Get help at Microsoft Q&A

How to troubleshoot instance-related issues on Azure App Service

Article • 02/01/2023

App services are hosted on Azure App Service plans that define the compute resources that you need to be able to run a web app. The compute resources, also known as instances, can occasionally become unavailable. This video helps you to identify such issues, and explains how to resolve them. From improving load balancing to moving to an entirely new set of instances, you'll be well equipped to resolve instance-related issues by applying these techniques.
https://www.youtube-nocookie.com/embed/Cg_muggc2m0

Related content

For more information about the methods that are described in this video, plus additional methods to maintain high availability for Azure App Services, see The Ultimate Guide to Running Healthy Apps in the Cloud - Azure App Service.

Contact us for help

If you have questions or need help, create a support request, or ask Azure community support. You can also submit product feedback to Azure community support.

Feedback

Was this page helpful? Yes No

Provide product feedback | Get help at Microsoft Q&A

Troubleshoot virtual network integration with Azure App Service

07/28/2025

This article describes tools you can use to troubleshoot connection issues in Azure App Service that integrate with a virtual network.

> **Note**
>
> Virtual network integration isn't supported for Docker Compose scenarios in App Service. Access restriction policies are ignored if a private endpoint is present.

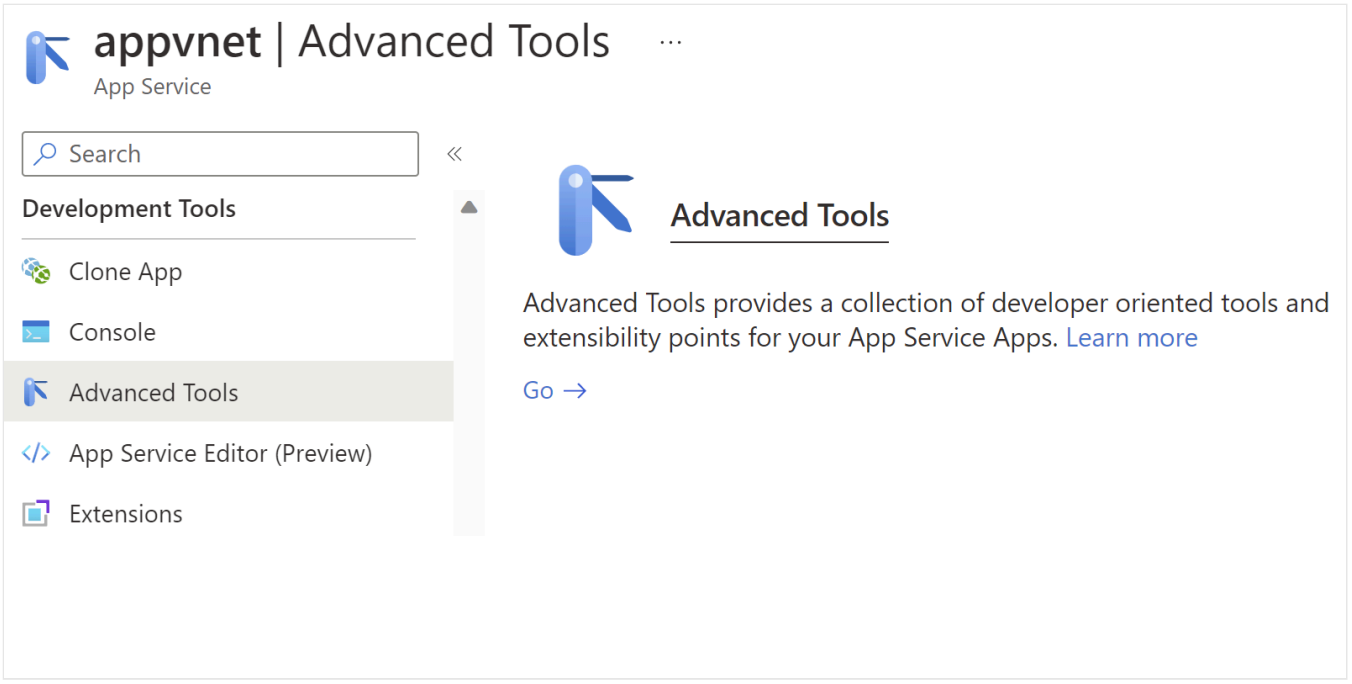
Verify virtual network integration

To troubleshoot the connection issues, you must first verify whether the virtual network integration is configured correctly and whether the private IP is assigned to all instances of the App Service Plan.

To do this, use one of the following methods:

Check the private IP in the Kudu Debug console

To access the Kudu console, select the app service in the Azure portal, go to **Development Tools**, select **Advanced Tools**, and then select **Go**. In the Kudu service page, select **Tools** > **Debug Console** > **CMD**.



You can also go to the Kudu Debug console directly by the URL `[sitename].scm.azurewebsites.net/DebugConsole`.

In the Debug console, run one of the following commands:

Windows OS-based apps

```
Console
SET WEBSITE_PRIVATE_IP
```

If the private IP is assigned successfully, you'll get the following output:

```
Output
WEBSITE_PRIVATE_IP=<IP address>
```

Linux OS-based apps

```
Console
set| egrep --color 'WEBSITE_PRIVATE_IP'
```

Check the private IP in the Kudu environment

Go to the Kudu environment at `[sitename].scm.azurewebsites.net/Env` and search for `WEBSITE_PRIVATE_IP`.

Once we've established that the virtual network integration is configured successfully, we can proceed with the connectivity test.

Troubleshoot outbound connectivity on Windows Apps

In native Windows Apps, the tools **ping**, **nslookup**, and **tracert** won't work through the console because of security constraints (they work in custom Windows Containers).

Go to the Kudu console directly at `[sitename].scm.azurewebsites.net/DebugConsole`.

To test DNS functionality, you can use **nameresolver.exe**. The syntax is:

```
Console

nameresolver.exe hostname [optional:DNS Server]
```

You can use **nameresolver** to check the hostnames that your app depends on. This way, you can test if you have anything misconfigured with your DNS or perhaps don't have access to your DNS server. You can see the DNS server that your app uses in the console by looking at the environmental variables WEBSITE_DNS_SERVER and WEBSITE_DNS_ALT_SERVER.

> **Note**
>
> The nameresolver.exe tool currently doesn't work in custom Windows containers.

To test TCP connectivity to a host and port combination, you can use **tcpping**. The syntax is.

```
Console

tcpping.exe hostname [optional: port]
```

The **tcpping** utility tells you if you can reach a specific host and port. It can show success only if there's an application listening at the host and port combination and there's network access from your app to the specified host and port.

Troubleshoot outbound connectivity on Linux Apps

Go to Kudu directly at `[sitename].scm.azurewebsites.net`. In the Kudu service page, select **Tools** > **Debug Console** > **CMD**.

To test DNS functionality, you can use the command **nslookup**. The syntax is:

```
Console

nslookup hostname [optional:DNS Server]
```

Depending on the above results, you can check if there's something misconfigured on your DNS server.

> **Note**
>
> The nameresolver.exe tool currently doesn't work in Linux apps.

To test connectivity, you can use the **Curl** command. The syntax is:

```
Console

curl -v https://hostname
curl hostname:[port]
```

Debug access to virtual network-hosted resources

A number of factors can prevent your app from reaching a specific host and port. Most of the time, it's one of the following:

- **A firewall is in the way.** If you have a firewall in the way, you hit the TCP timeout. The TCP timeout is 21 seconds in this case. Use the **tcpping** tool to test connectivity. TCP timeouts can be caused by many things beyond firewalls, but start there.
- **DNS isn't accessible.** The DNS timeout is three seconds per DNS server. If you have two DNS servers, the timeout is six seconds. Use nameresolver to see if the DNS is working. You can't use nslookup because that doesn't use the DNS your virtual network is configured with. If inaccessible, you could have a firewall or NSG blocking access to DNS, or it could be down. Some DNS architectures that use custom DNS servers can be complex and may occasionally experience timeouts. To determine if this is the case, the environment variable `WEBSITE_DNS_ATTEMPTS` can be set. For more information about DNS in App Services, see Name resolution (DNS) in App Service.

If those items don't answer your problems, look first for things like:

Regional virtual network integration

- Is your destination a non-RFC1918 address and you don't have **Route All** enabled?

- Is there an NSG blocking egress from your integration subnet?
- If you're going across Azure ExpressRoute or a VPN, is your on-premises gateway configured to route traffic back up to Azure? If you can reach endpoints in your virtual network but not on-premises, check your routes.
- Do you have enough permissions to set delegation on the integration subnet? During regional virtual network integration configuration, your integration subnet is delegated to Microsoft.Web/serverFarms. The VNet integration UI delegates the subnet to Microsoft.Web/serverFarms automatically. If your account doesn't have sufficient networking permissions to set delegation, you'll need someone who can set attributes on your integration subnet to delegate the subnet. To manually delegate the integration subnet, go to the Azure Virtual Network subnet UI and set the delegation for Microsoft.Web/serverFarms.

Debugging networking issues is a challenge because you can't see what's blocking access to a specific host:port combination. Some causes include:

- You have a firewall up on your host that prevents access to the application port from your point-to-site IP range. Crossing subnets often requires public access.
- Your target host is down.
- Your application is down.
- You had the wrong IP or hostname.
- Your application is listening on a different port than what you expected. You can match your process ID with the listening port by using "netstat -aon" on the endpoint host.
- Your network security groups are configured in such a manner that they prevent access to your application host and port from your point-to-site IP range.

You don't know what address your app actually uses. It could be any address in the integration subnet or point-to-site address range, so you need to allow access from the entire address range.

More debug steps include:

- Connect to a VM in your virtual network and attempt to reach your resource host:port from there. To test for TCP access, use the PowerShell command **Test-NetConnection**. The syntax is:

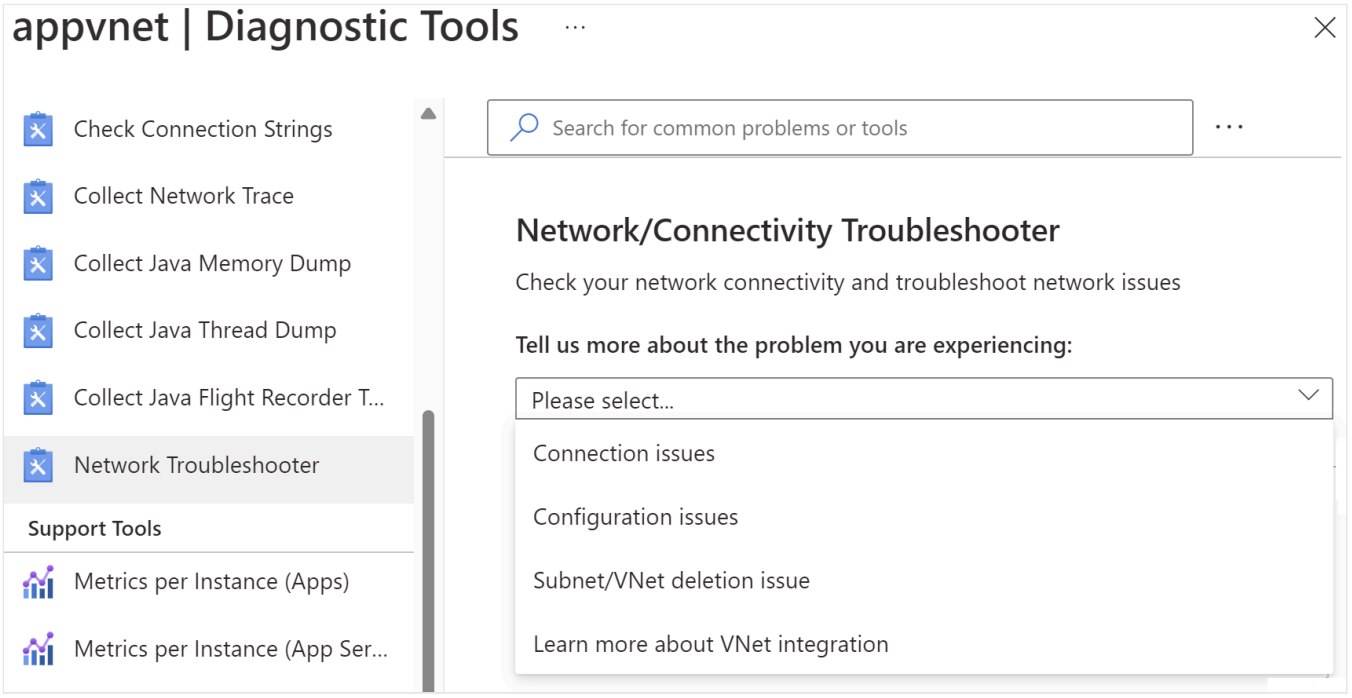
PowerShell

```
Test-NetConnection hostname [optional: -Port]
```

- Bring up an application on a VM and test access to that host and port from the console from your app by using **tcpping**.

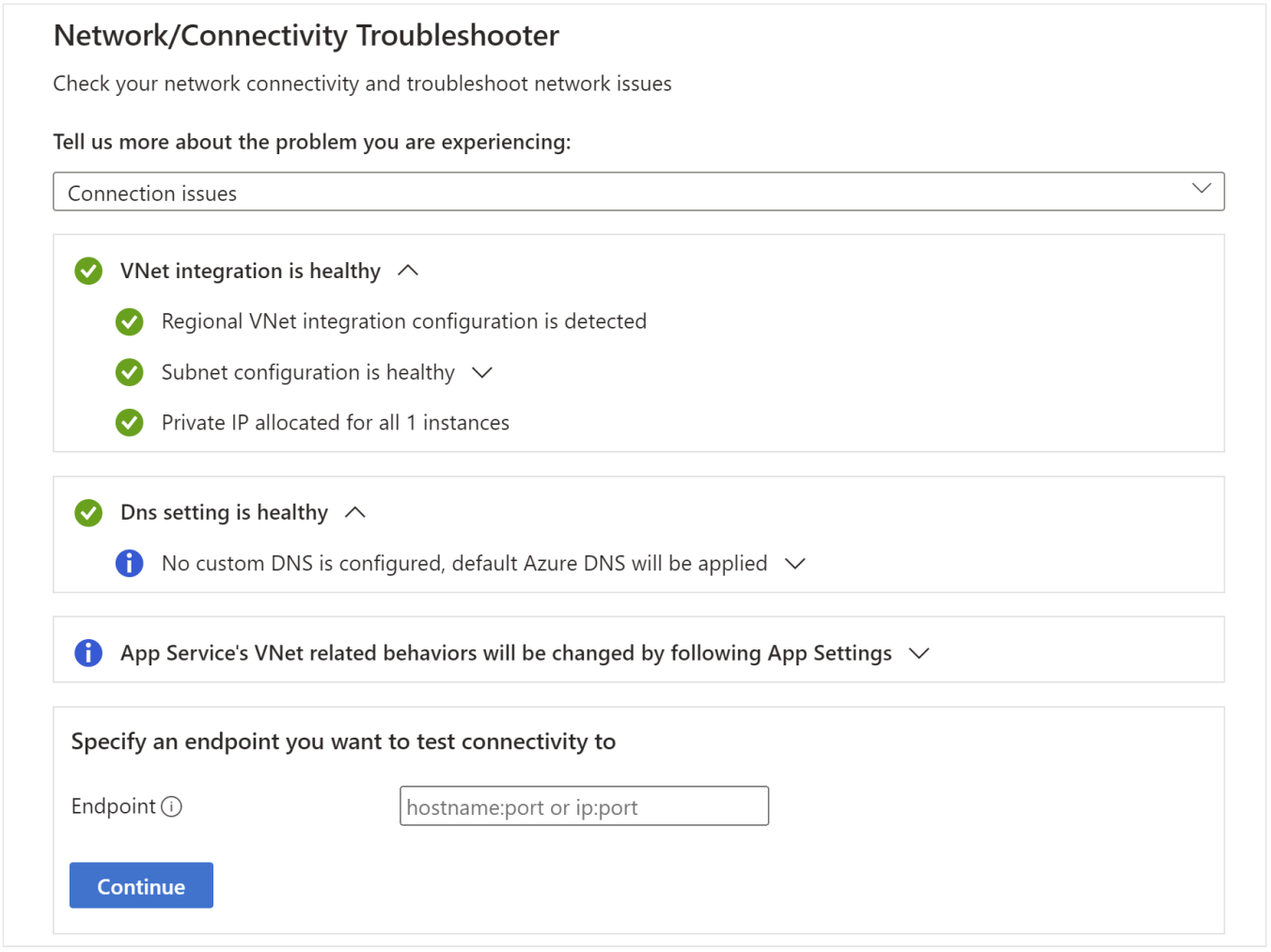
Network troubleshooter

You can also use the Network troubleshooter to troubleshoot the connection issues for the apps in the App Service. To open the network troubleshooter, go to the app service in the Azure portal. Select **Diagnostic and solve problem**, and then search for **Network troubleshooter**.

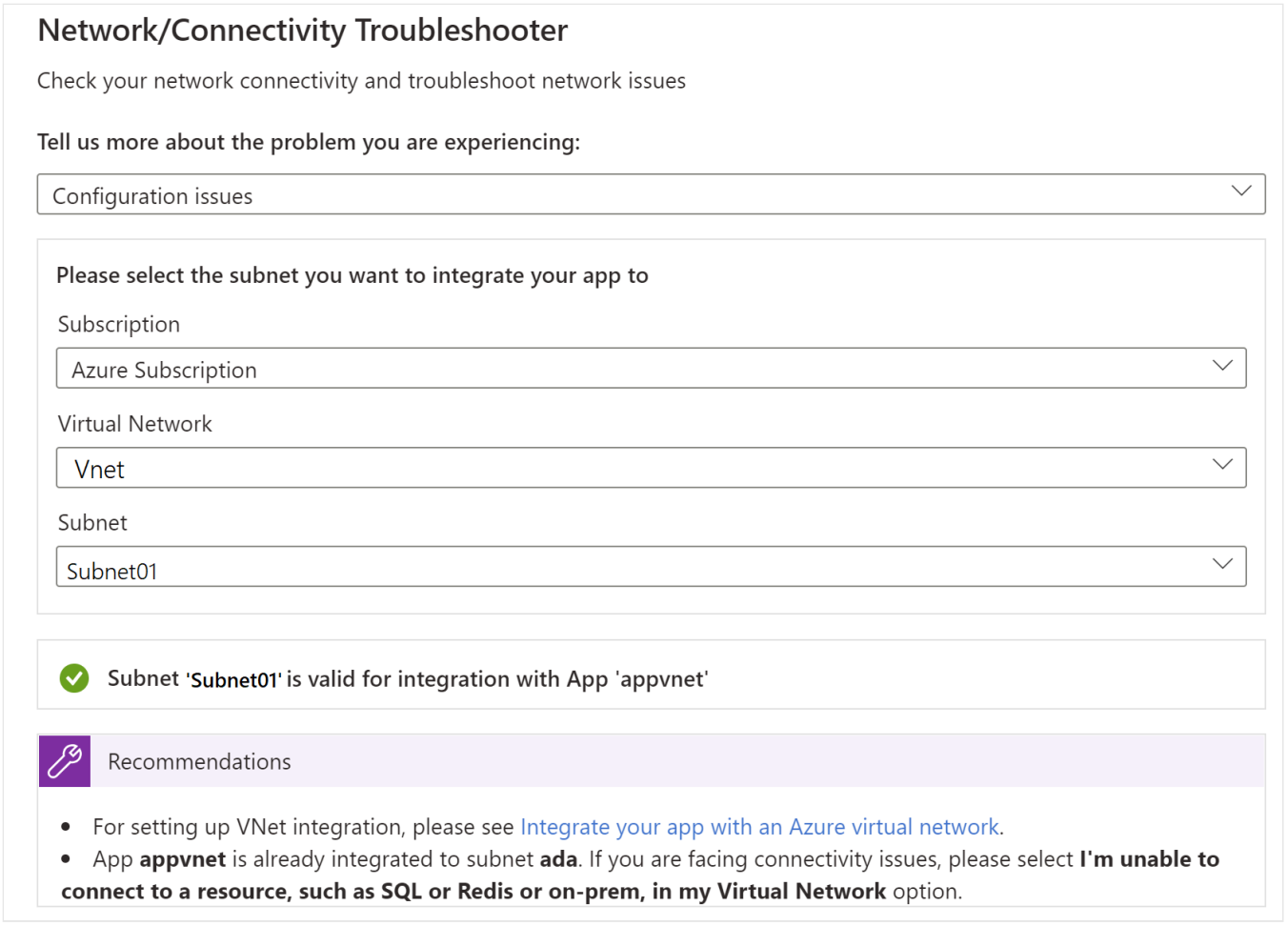


> **Note**
>
> The **connection issues** scenario doesn't support Linux or Container-based apps yet.

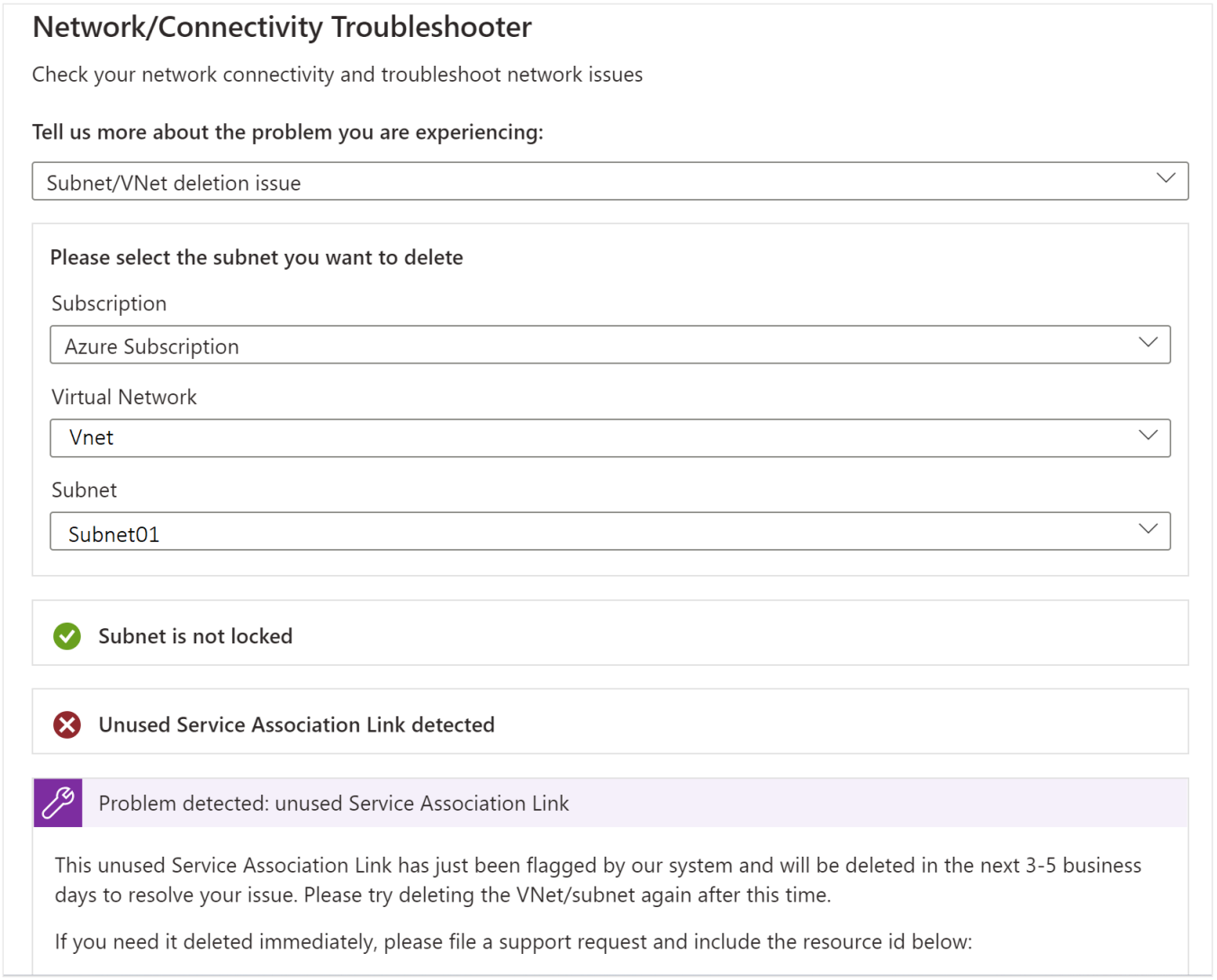
Connection issues - It will check the status of the virtual network integration, including checking if the Private IP has been assigned to all instances of the App Service Plan and the DNS settings. If a custom DNS isn't configured, default Azure DNS will be applied. You can also run tests against a specific endpoint that you want to test connectivity to.



Configuration issues - This troubleshooter will check if your subnet is valid for virtual network Integration.



Subnet/VNet deletion issue - This troubleshooter will check if your subnet has any locks and if it has any unused Service Association Links that might be blocking the deletion of the VNet/subnet.



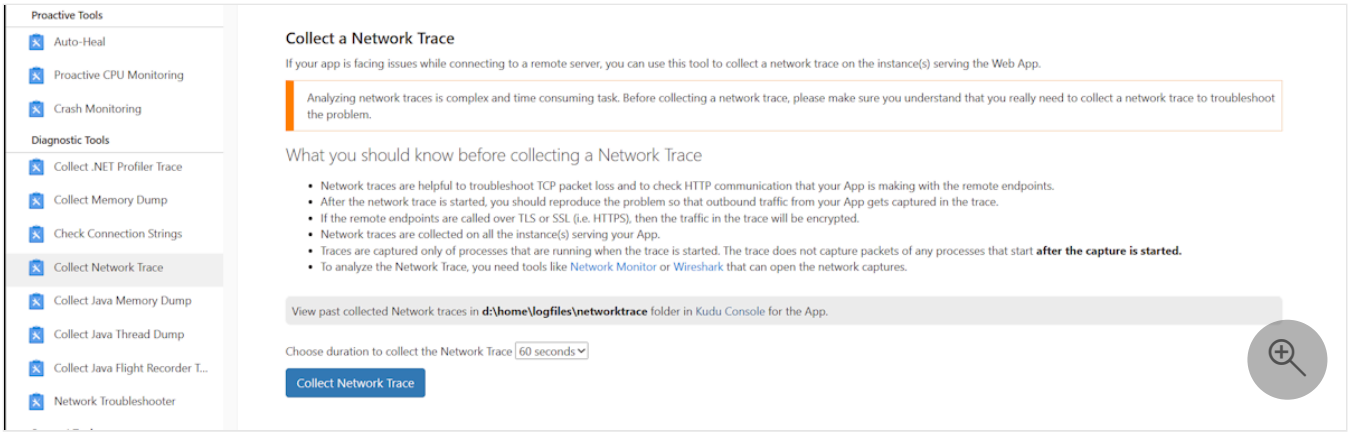
Collect network traces

Collecting network traces can be helpful in analyzing issues. In Azure App Services, network traces are taken from the application process. To obtain accurate information, reproduce the issue while starting the network trace collection.

Windows App Services

To collect network traces for Windows App Services, follow these steps:

1. In the Azure portal, navigate to your Web App.
2. In the left navigation, select **Diagnose and Solve Problems**.
3. In the search box, type *Collect Network Trace* and select **Collect Network Trace** to start the network trace collection.



To get the trace file for each instance serving a Web App, on your browser, go to the Kudu console for the Web App (`https://<sitename>.scm.azurewebsites.net`). Download the trace file from the *C:\home\LogFiles\networktrace* or *D:\home\LogFiles\networktrace* folder.

Linux App Services

To collect network traces for Linux App Services that don't use a custom container, follow these steps:

1. Install the `tcpdump` command line utility by running the following commands:

   Bash

   ```
   apt-get update
   apt install tcpdump
   ```

2. Connect to the container via the Secure Shell Protocol (SSH).

3. Identify the interface that's up and running by running the following command (for example, `eth0`):

   Bash

   ```
   root@<hostname>:/home# tcpdump -D

   1.eth0 [Up, Running, Connected]
   2.any (Pseudo-device that captures on all interfaces) [Up, Running]
   3.lo [Up, Running, Loopback]
   4.bluetooth-monitor (Bluetooth Linux Monitor) [Wireless]
   5.nflog (Linux netfilter log (NFLOG) interface) [none]
   6.nfqueue (Linux netfilter queue (NFQUEUE) interface) [none]
   7.dbus-system (D-Bus system bus) [none]
   8.dbus-session (D-Bus session bus) [none]
   ```

4. Start the network trace collection by running the following command:

Bash

```
root@<hostname>:/home# tcpdump -i eth0 -w networktrace.pcap
```

Replace `eth0` with the name of the actual interface.

To download the trace file, connect to the Web App via methods such as Kudu, FTP, or a Kudu API request. Here's a request example for triggering the file download:

`https://<sitename>.scm.azurewebsites.net/api/vfs/<path to the trace file in the /home directory>/filename`

Third-party information disclaimer

The third-party products that this article discusses are manufactured by companies that are independent of Microsoft. Microsoft makes no warranty, implied or otherwise, about the performance or reliability of these products.

Contact us for help

If you have questions or need help, create a support request, or ask Azure community support. You can also submit product feedback to Azure feedback community.