Contents

What is Azure Front Door Service?

11/22/2019 • 6 minutes to read • Edit Online

Azure Front Door Service enables you to define, manage, and monitor the global routing for your web traffic by optimizing for best performance and instant global failover for high availability. With Front Door, you can transform your global (multi-region) consumer and enterprise applications into robust, high-performance personalized modern applications, APIs, and content that reach a global audience with Azure.

Front Door works at Layer 7 or HTTP/HTTPS layer and uses anycast protocol with split TCP and Microsoft's global network for improving global connectivity. So, per your routing method selection in the configuration, you can ensure that Front Door is routing your client requests to the fastest and most available application backend. An application backend is any Internet-facing service hosted inside or outside of Azure. Front Door provides a range of traffic-routing methods and backend health monitoring options to suit different application needs and automatic failover models. Similar to Traffic Manager, Front Door is resilient to failures, including the failure of an entire Azure region.

> **NOTE**
>
> Azure provides a suite of fully managed load-balancing solutions for your scenarios. If you are looking for a DNS based global routing and do **not** have requirements for Transport Layer Security (TLS) protocol termination ("SSL offload") or per-HTTP/HTTPS request, application-layer processing, review Traffic Manager. If you are looking for load balancing between your servers in a region, for application layer, review Application Gateway and for network layer load balancing, review Load Balancer. Your end-to-end scenarios might benefit from combining these solutions as needed.
>
> For an Azure load-balancing options comparison, see Overview of load-balancing options in Azure.

The following features are included with Front Door:

Accelerate application performance

Using split TCP-based anycast protocol, Front Door ensures that your end users promptly connect to the nearest Front Door POP (Point of Presence). Using Microsoft's global network for connecting to your application backends from Front Door POPs, ensure higher availability and reliability while maintaining performance. This connectivity to your backend is also based on least network latency. Learn more about Front Door routing techniques like Split TCP and Anycast protocol.

Increase application availability with smart health probes

Front Door delivers high availability for your critical applications using its smart health probes, monitoring your backends for both latency and availability and providing instant automatic failover when a backend goes down. So, you can run planned maintenance operations on your applications without downtime. Front Door directs traffic to alternative backends while the maintenance is in progress.

URL-based routing

URL Path Based Routing allows you to route traffic to backend pools based on URL paths of the request. One of the scenarios is to route requests for different content types to different backend pools.

For example, requests for `http://www.contoso.com/users/*` are routed to UserProfilePool, and `http://www.contoso.com/products/*` are routed to ProductInventoryPool. Front Door allows even more complex route matching scenarios using best match algorithm and so if none of the path patterns match then your default routing rule for `http://www.contoso.com/*` is selected and the traffic is directed to default catch-all routing rule. Learn more at Route Matching.

Multiple-site hosting

Multiple-site hosting enables you to configure more than one web site on the same Front Door configuration. This feature allows you to configure a more efficient topology for your deployments by adding different web sites to a single Front Door configuration. Based on your application's architecture, you can configure Azure Front Door Service to either direct each web site to its own backend pool or have various web sites directed to the same backend pool. For example, Front Door can serve traffic for `images.contoso.com` and `videos.contoso.com` from two backend pools called ImagePool and VideoPool. Alternatively you can configure both the front-end hosts to direct traffic to a single backend pool called MediaPool.

Similarly, you can have two different domains `www.contoso.com` and `www.fabrikam.com` configured on the same Front Door.

Session affinity

The cookie-based session affinity feature is useful when you want to keep a user session on the same application backend. By using Front Door managed cookies, subsequent traffic from a user session gets directed to the same application backend for processing. This feature is important in cases where session state is saved locally on the backend for a user session.

Secure Sockets Layer (SSL) termination

Front Door supports SSL termination at the edge that is, individual users can set up SSL connection with Front Door environments instead of establishing it over long haul connections with the application backend. Additionally, Front Door supports both HTTP as well as HTTPS connectivity between Front Door environments and your backends. So, you can also set up end-to-end SSL encryption. For example, if Front Door for your application workload receives over 5000 requests in a minute, due to warm connection reuse, for active services, it will only establish say about 500 connections with your application backend, thereby reducing significant load from your backends.

Custom domains and certificate management

When you use Front Door to deliver content, a custom domain is necessary if you would like your own domain name to be visible in your Front Door URL. Having a visible domain name can be convenient for your customers and useful for branding purposes. Front Door also supports HTTPS for custom domain names. Use this feature by either choosing Front Door managed certificates for your traffic or uploading your own custom SSL certificate.

Application layer security

Azure Front Door allows you to author custom Web Application Firewall (WAF) rules for access control to protect your HTTP/HTTPS workload from exploitation based on client IP addresses, country code, and http parameters. Additionally, Front Door also enables you to create rate limiting rules to battle malicious bot traffic. For more information about Web Application Firewall, see What is Azure Web Application Firewall?

Front Door platform itself is protected by Azure DDoS Protection Basic. For further protection, Azure DDoS Protection Standard may be enabled at your VNETs and safeguard resources from network layer (TCP/UDP) attacks via auto tuning and mitigation. Front Door is a layer 7 reverse proxy, it only allows web traffic to pass through to backends and block other types of traffic by default.

URL redirection

With the strong industry push on supporting only secure communication, web applications are expected to automatically redirect any HTTP traffic to HTTPS. This ensures that all communication between the users and the application occurs over an encrypted path.

Traditionally, application owners have dealt with this requirement by creating a dedicated service, whose sole purpose was to redirect requests it receives on HTTP to HTTPS. Azure Front Door Service supports the ability to redirect traffic from HTTP to HTTPS. This simplifies application configuration, optimizes the resource usage, and supports new redirection scenarios, including global and path-based redirection. URL redirection from Azure Front Door Service is not limited to HTTP to HTTPS redirection alone, but also to redirect to a different hostname, redirecting to a different path, or even redirecting to a new query string in the URL.

For more information, see redirecting traffic with Azure Front Door Service.

URL rewrite

Front Door supports URL rewrite by allowing you to configure an optional Custom Forwarding Path to use when constructing the request to forward to the backend. Front Door further allows you to configure Host header to be sent when forwarding the request to your backend.

Protocol support - IPv6 and HTTP/2 traffic

Azure Front Door natively supports end-to-end IPv6 connectivity and also HTTP/2 protocol.

The HTTP/2 protocol enables full-duplex communication between application backends and a client over a long-running TCP connection. HTTP/2 allows for a more interactive communication between the backend and the client, which can be bidirectional without the need for polling as required in HTTP-based implementations. HTTP/2 protocol has low overhead, unlike HTTP, and can reuse the same TCP connection for multiple request or responses resulting in a more efficient utilization of resources. Learn more about HTTP/2 support in Azure Front Door Service.

Pricing

For pricing information, see Front Door Pricing.

Next steps

- Learn how to create a Front Door.
- Learn how Front Door works.

Quickstart: Create a Front Door for a highly available global web application

11/19/2019 • 4 minutes to read • Edit Online

This quickstart describes how to create a Front Door profile that delivers high availability and high performance for your global web application.

The scenario described in this quickstart includes two instances of a web application running in different Azure regions. A Front Door configuration based on equal weighted and same priority backends is created that helps direct user traffic to the nearest set of site backends running the application. Front Door continuously monitors the web application and provides automatic failover to the next available backend when the nearest site is unavailable.

If you don't have an Azure subscription, create a free account before you begin.

Sign in to Azure

Sign in to the Azure portal at https://portal.azure.com.

Prerequisites

This quickstart requires that you have deployed two instances of a web application running in different Azure regions (*East US* and *West Europe*). Both the web application instances run in Active/Active mode, that is, either of them can take traffic at any time unlike a Active/Stand-By configuration where one acts as a failover.

1. On the top left-hand side of the screen, select **Create a resource** > **Web** > **Web App** > **Create**.
2. In **Web App**, enter or select the following information and enter default settings where none are specified:

SETTING	VALUE
Name	Enter a unique name for your web app
Resource group	Select **New**, and then type *myResourceGroupFD1*
App Service plan/Location	Select **New**. In the App Service plan, enter *myAppServicePlanEastUS*, and then select **OK**.
Location	East US

3. Select **Create**.
4. A default website is created when the Web App is successfully deployed.
5. Repeat steps 1-3 to create a second website in a different Azure region with the following settings:

SETTING	VALUE
Name	Enter a unique name for your Web App

SETTING	VALUE
Resource group	Select **New**, and then type *myResourceGroupFD2*
App Service plan/Location	Select **New**. In the App Service plan, enter *myAppServicePlanWestEurope*, and then select **OK**.
Location	West Europe

Create a Front Door for your application

A. Add a frontend host for Front Door

Create a Front Door configuration that directs user traffic based on lowest latency between the two backends.

1. On the top left-hand side of the screen, select **Create a resource** > **Networking** > **Front Door** > **Create**.
2. In the **Create a Front Door**, you start with adding the basic info and provide a subscription where you want the Front Door to be configured. Similarly, like any other Azure resource you also need to provide a ResourceGroup and a Resource Group region if you are creating a new one. Lastly, you need to provide a name for your Front Door.
3. Once the basic info is filled in, the first step you need to define is the **frontend host** for the configuration. The result should be a valid domain name like `myappfrontend.azurefd.net`. This hostname needs to be globally unique but Front Door will take care of that validation.

B. Add application backend and backend pools

Next, you need to configure your application backend(s) in a backend pool for Front Door to know where your application resides.

1. Click the '+' icon to add a backend pool and then specify a **name** for your backend pool, say `myBackendPool`.
2. Next, click on Add Backends to add your websites created earlier.
3. Select **Target host type** as 'App Service', select the subscription in which you created the web site and then choose the first web site from the **Target host name**, that is, *myAppServicePlanEastUS.azurewebsites.net*.
4. Leave the remaining fields as is for now and click **Add'**.
5. Repeat steps 2 to 4 to add the other website, that is, *myAppServicePlanWestEurope.azurewebsites.net*
6. You can optionally choose to update the Health Probes and Load Balancing settings for the backend pool, but the default values should also work. Click **Add**.

C. Add a routing rule

Lastly, click the '+' icon on Routing rules to configure a routing rule. This is needed to map your frontend host to the backend pool, which basically is configuring that if a request comes to `myappfrontend.azurefd.net`, then forward it to the backend pool `myBackendPool`. Click **Add** to add the routing rule for your Front Door. You should now be good to creating the Front Door and so click on **Review and Create**.

> **WARNING**
>
> You **must** ensure that each of the frontend hosts in your Front Door has a routing rule with a default path ('/*') associated with it. That is, across all of your routing rules there must be at least one routing rule for each of your frontend hosts defined at the default path ('/*'). Failing to do so, may result in your end-user traffic not getting routed correctly.

View Front Door in action

Once you create a Front Door, it will take a few minutes for the configuration to be deployed globally everywhere. Once complete, access the frontend host you created, that is, go to a web browser and hit the URL `myappfrontend.azurefd.net`. Your request will automatically get routed to the nearest backend to you from the specified backends in the backend pool.

View Front Door handle application failover

If you want to test Front Door's instant global failover in action, you can go to one of the web sites you created and stop it. Based on the Health Probe setting defined for the backend pool, we will instantly fail over the traffic to the other web site deployment. You can also test behavior, by disabling the backend in the backend pool configuration for your Front Door.

Clean up resources

When no longer needed, delete the resource groups, web applications, and all related resources.

Next steps

In this quickstart, you created a Front Door that allows you to direct user traffic for web applications that require high availability and maximum performance. To learn more about routing traffic, read the Routing Methods used by Front Door.

Tutorial: Add a custom domain to your Front Door

11/19/2019 • 7 minutes to read • Edit Online

This tutorial shows how to add a custom domain to your Front Door. When you use Azure Front Door Service for application delivery, a custom domain is necessary if you would like your own domain name to be visible in your end-user request. Having a visible domain name can be convenient for your customers and useful for branding purposes.

After you create a Front Door, the default frontend host, which is a subdomain of `azurefd.net`, is included in the URL for delivering Front Door content from your backend by default (for example, https://contoso.azurefd.net/activeusers.htm). For your convenience, Azure Front Door provides the option of associating a custom domain with the default host. With this option, you deliver your content with a custom domain in your URL instead of a Front Door owned domain name (for example, https://www.contoso.com/photo.png).

In this tutorial, you learn how to:

- Create a CNAME DNS record.
- Associate the custom domain with your Front Door.
- Verify the custom domain.

If you don't have an Azure subscription, create a free account before you begin.

Prerequisites

Before you can complete the steps in this tutorial, you must first create a Front Door. For more information, see Quickstart: Create a Front Door.

If you do not already have a custom domain, you must first purchase one with a domain provider. For example, see Buy a custom domain name.

If you are using Azure to host your DNS domains, you must delegate the domain provider's domain name system (DNS) to an Azure DNS. For more information, see Delegate a domain to Azure DNS. Otherwise, if you are using a domain provider to handle your DNS domain, proceed to Create a CNAME DNS record.

Create a CNAME DNS record

Before you can use a custom domain with your Front Door, you must first create a canonical name (CNAME) record with your domain provider to point to your Front Door's default frontend host (say contoso.azurefd.net). A CNAME record is a type of DNS record that maps a source domain name to a destination domain name. For Azure Front Door Service, the source domain name is your custom domain name and the destination domain name is your Front Door default hostname. After Front Door verifies the CNAME record that you create, traffic addressed to the source custom domain (such as www.contoso.com) is routed to the specified destination Front Door default frontend host (such as contoso.azurefd.net).

A custom domain and its sub-domain can be associated with only a single Front Door at a time. However, you can use different sub-domains from the same custom domain for different Front Doors by using multiple CNAME records. You can also map a custom domain with different sub-domains to the same Front Door.

Map the temporary afdverify sub-domain

When you map an existing domain that is in production, there are special considerations. While you are registering your custom domain in the Azure portal, a brief period of downtime for the domain can occur. To avoid interruption of web traffic, first map your custom domain to your Front Door default frontend host with the Azure afdverify sub-domain to create a temporary CNAME mapping. With this method, users can access your domain without interruption while the DNS mapping occurs.

Otherwise, if you are using your custom domain for the first time and no production traffic is running on it, you can directly map your custom domain to your Front Door. Proceed to Map the permanent custom domain.

To create a CNAME record with the afdverify subdomain:

1. Sign in to the web site of the domain provider for your custom domain.

2. Find the page for managing DNS records by consulting the provider's documentation or searching for areas of the web site labeled **Domain Name**, **DNS**, or **Name server management**.

3. Create a CNAME record entry for your custom domain and complete the fields as shown in the following table (field names may vary):

SOURCE	TYPE	DESTINATION
afdverify.www.contoso.com	CNAME	afdverify.contoso.azurefd.net

   - Source: Enter your custom domain name, including the afdverify subdomain, in the following format: afdverify.<*custom domain name*>. For example, afdverify.www.contoso.com.

   - Type: Enter *CNAME*.

   - Destination: Enter your default Front Door frontend host, including the afdverify subdomain, in the following format: afdverify.<*endpoint name*>.azurefd.net. For example, afdverify.contoso.azurefd.net.

4. Save your changes.

For example, the procedure for the GoDaddy domain registrar is as follows:

1. Sign in and select the custom domain you want to use.

2. In the Domains section, select **Manage All**, then select **DNS** | **Manage Zones**.

3. For **Domain Name**, enter your custom domain, then select **Search**.

4. From the **DNS Management** page, select **Add**, then select **CNAME** in the **Type** list.

5. Complete the following fields of the CNAME entry:

   - Type: Leave *CNAME* selected.

   - Host: Enter the subdomain of your custom domain to use, including the afdverify subdomain name. For example, afdverify.www.

   - Points to: Enter the host name of your default Front Door frontend host, including the afdverify subdomain name. For example, afdverify.contoso.azurefd.net.

   - TTL: Leave *1 Hour* selected.

6. Select **Save**.

   The CNAME entry is added to the DNS records table.

Associate the custom domain with your Front Door

After you've registered your custom domain, you can then add it to your Front Door.

1. Sign in to the Azure portal and browse to the Front Door containing the frontend host that you want to map to a custom domain.

2. On the **Front Door designer** page, click on '+' to add a custom domain.

3. Specify **Custom domain**.

4. For **Frontend host**, the frontend host to use as the destination domain of your CNAME record is pre-filled and is derived from your Front Door: <*default hostname*>.azurefd.net. It cannot be changed.

5. For **Custom hostname**, enter your custom domain, including the subdomain, to use as the source domain of your CNAME record. For example, www.contoso.com or cdn.contoso.com. Do not use the afdverify subdomain name.

6. Select **Add**.

   Azure verifies that the CNAME record exists for the custom domain name you entered. If the CNAME is correct, your custom domain will be validated.

> **WARNING**
>
> You **must** ensure that each of the frontend hosts (including custom domains) in your Front Door has a routing rule with a default path ('/*') associated with it. That is, across all of your routing rules there must be at least one routing rule for each of your frontend hosts defined at the default path ('/*'). Failing to do so, may result in your end-user traffic not getting routed correctly.

Verify the custom domain

After you have completed the registration of your custom domain, verify that the custom domain references your default Front Door frontend host.

In your browser, navigate to the address of the file by using the custom domain. For example, if your custom domain is robotics.contoso.com, the URL to the cached file should be similar to the following URL: http://robotics.contoso.com/my-public-container/my-file.jpg. Verify that the result is that same as when you access the Front Door directly at <*Front Door host*>.azurefd.net.

Map the permanent custom domain

If you have verified that the afdverify subdomain has been successfully mapped to your Front Door (or if you are using a new custom domain that is not in production), you can then map the custom domain directly to your default Front Door frontend host.

To create a CNAME record for your custom domain:

1. Sign in to the web site of the domain provider for your custom domain.

2. Find the page for managing DNS records by consulting the provider's documentation or searching for areas of the web site labeled **Domain Name**, **DNS**, or **Name Server Management**.

3. Create a CNAME record entry for your custom domain and complete the fields as shown in the following table (field names may vary):

SOURCE	TYPE	DESTINATION
<www.contoso.com>	CNAME	contoso.azurefd.net

   - Source: Enter your custom domain name (for example, www.contoso.com).

- Type: Enter *CNAME*.
- Destination: Enter your default Front Door frontend host. It must be in the following format:<*hostname*>.azurefd.net. For example, contoso.azurefd.net.

4. Save your changes.
5. If you're previously created a temporary afdverify subdomain CNAME record, delete it.
6. If you are using this custom domain in production for the first time, follow the steps for Associate the custom domain with your Front Door and Verify the custom domain.

For example, the procedure for the GoDaddy domain registrar is as follows:

1. Sign in and select the custom domain you want to use.
2. In the Domains section, select **Manage All**, then select **DNS** | **Manage Zones**.
3. For **Domain Name**, enter your custom domain, then select **Search**.
4. From the **DNS Management** page, select **Add**, then select **CNAME** in the **Type** list.
5. Complete the fields of the CNAME entry:
   - Type: Leave *CNAME* selected.
   - Host: Enter the subdomain of your custom domain to use. For example, www or profile.
   - Points to: Enter the default host name of your Front Door. For example, contoso.azurefd.net.
   - TTL: Leave *1 Hour* selected.
6. Select **Save**.

   The CNAME entry is added to the DNS records table.
7. If you have an afdverify CNAME record, select the pencil icon next to it, then select the trash can icon.
8. Select **Delete** to delete the CNAME record.

Clean up resources

In the preceding steps, you added a custom domain to a Front Door. If you no longer want to associate your Front Door with a custom domain, you can remove the custom domain by performing these steps:

1. In your Front Door designer, select the custom domain that you want to remove.
2. Click Delete from the context menu for the custom domain.

   The custom domain is disassociated from your endpoint.

Next steps

In this tutorial, you learned how to:

- Create a CNAME DNS record.
- Associate the custom domain with your Front Door.
- Verify the custom domain.

Tutorial: Configure HTTPS on a Front Door custom domain

1/22/2020 • 12 minutes to read • Edit Online

This tutorial shows how to enable the HTTPS protocol for a custom domain that's associated with your Front Door under the frontend hosts section. By using the HTTPS protocol on your custom domain (for example, https://www.contoso.com), you ensure that your sensitive data is delivered securely via TLS/SSL encryption when it is sent across the internet. When your web browser is connected to a web site via HTTPS, it validates the web site's security certificate and verifies it's issued by a legitimate certificate authority. This process provides security and protects your web applications from attacks.

Azure Front Door Service supports HTTPS on a Front Door default hostname, by default. For example, if you create a Front Door (such as https://contoso.azurefd.net), HTTPS is automatically enabled for requests made to https://contoso.azurefd.net. However, once you onboard the custom domain 'www.contoso.com' you will need to additionally enable HTTPS for this frontend host.

Some of the key attributes of the custom HTTPS feature are:

- No additional cost: There are no costs for certificate acquisition or renewal and no additional cost for HTTPS traffic.
- Simple enablement: One-click provisioning is available from the Azure portal. You can also use REST API or other developer tools to enable the feature.
- Complete certificate management is available: All certificate procurement and management is handled for you. Certificates are automatically provisioned and renewed prior to expiration, which removes the risks of service interruption due to a certificate expiring.

In this tutorial, you learn how to:

- Enable the HTTPS protocol on your custom domain.
- Use an AFD-managed certificate
- Use your own certificate, that is, a custom SSL certificate
- Validate the domain
- Disable the HTTPS protocol on your custom domain

NOTE

This article has been updated to use the new Azure PowerShell Az module. You can still use the AzureRM module, which will continue to receive bug fixes until at least December 2020. To learn more about the new Az module and AzureRM compatibility, see Introducing the new Azure PowerShell Az module. For Az module installation instructions, see Install Azure PowerShell.

Prerequisites

Before you can complete the steps in this tutorial, you must first create a Front Door and with at least one custom domain onboarded. For more information, see Tutorial: Add a custom domain to your Front Door.

SSL certificates

To enable the HTTPS protocol for securely delivering content on a Front Door custom domain, you must use an SSL certificate. You can choose to use a certificate that is managed by Azure Front Door Service or use your own certificate.

Option 1 (default): Use a certificate managed by Front Door

When you use a certificate managed by Azure Front Door Service, the HTTPS feature can be turned on with just a few clicks. Azure Front Door Service completely handles certificate management tasks such as procurement and renewal. After you enable the feature, the process starts immediately. If the custom domain is already mapped to the Front Door's default frontend host (`{hostname}.azurefd.net`), no further action is required. Front Door will process the steps and complete your request automatically. However, if your custom domain is mapped elsewhere, you must use email to validate your domain ownership.

To enable HTTPS on a custom domain, follow these steps:

1. In the Azure portal, browse to your **Front Door** profile.
2. In the list of frontend hosts, select the custom domain you want to enable HTTPS for containing your custom domain.
3. Under the section **Custom domain HTTPS**, click **Enabled**, and select **Front Door managed** as the certificate source.
4. Click Save.
5. Proceed to Validate the domain.

Option 2: Use your own certificate

You can use your own certificate to enable the HTTPS feature. This process is done through an integration with Azure Key Vault, which allows you to store your certificates securely. Azure Front Door Service uses this secure mechanism to get your certificate and it requires a few additional steps. When you create your SSL certificate, you must create it with an allowed certificate authority (CA). Otherwise, if you use a non-allowed CA, your request will be rejected. For a list of allowed CAs, see Allowed certificate authorities for enabling custom HTTPS on Azure Front Door Service.

Prepare your Azure Key vault account and certificate

1. Azure Key Vault: You must have a running Azure Key Vault account under the same subscription as your Front Door that you want to enable custom HTTPS. Create an Azure Key Vault account if you don't have one.

> **WARNING**
>
> Azure Front Door Service currently only supports Key Vault accounts in the same subscription as the Front Door configuration. Choosing a Key Vault under a different subscription than your Front Door will result in a failure.

2. Azure Key Vault certificates: If you already have a certificate, you can upload it directly to your Azure Key Vault account or you can create a new certificate directly through Azure Key Vault from one of the partner CAs that Azure Key Vault integrates with. Upload your certificate as a **certificate** object, rather than a **secret**.

> **IMPORTANT**
>
> You must upload the certificate in PFX format **without** password protection.

Register Azure Front Door Service

Register the service principal for Azure Front Door Service as an app in your Azure Active Directory via PowerShell.

> **NOTE**
>
> This action requires Global Administrator permissions, and needs to be performed only **once** per tenant.

1. If needed, install Azure PowerShell in PowerShell on your local machine.

2. In PowerShell, run the following command:

   ```
   New-AzADServicePrincipal -ApplicationId "ad0e1c7e-6d38-4ba4-9efd-0bc77ba9f037"
   ```

Grant Azure Front Door Service access to your key vault

Grant Azure Front Door Service permission to access the certificates in your Azure Key Vault account.

1. In your key vault account, under SETTINGS, select **Access policies**, then select **Add new** to create a new policy.

2. In **Select principal**, search for **ad0e1c7e-6d38-4ba4-9efd-0bc77ba9f037**, and choose **Microsoft.Azure.Frontdoor**. Click **Select**.

3. In **Secret permissions**, select **Get** to allow Front Door to retrieve the certificate.

4. In **Certificate permissions**, select **Get** to allow Front Door to retrieve the certificate.

5. Select **OK**.

   Azure Front Door Service can now access this Key Vault and the certificates that are stored in this Key Vault.

Select the certificate for Azure Front Door Service to deploy

1. Return to your Front Door in the portal.

2. In the list of custom domains, select the custom domain for which you want to enable HTTPS.

   The **Custom domain** page appears.

3. Under Certificate management type, select **Use my own certificate**.

4. Azure Front Door Service requires that the subscription of the Key Vault account is the same as for your Front Door. Select a key vault, certificate (secret), and certificate version.

   Azure Front Door Service lists the following information:

   - The key vault accounts for your subscription ID.
   - The certificates (secrets) under the selected key vault.
   - The available certificate versions.

5. When you use your own certificate, domain validation is not required. Proceed to Wait for propagation.

Validate the domain

If you already have a custom domain in use that is mapped to your custom endpoint with a CNAME record or you're using your own certificate, proceed to Custom domain is mapped to your Front Door. Otherwise, if the CNAME record entry for your domain no longer exists or it contains the afdverify subdomain, proceed to Custom domain is not mapped to your Front Door.

Custom domain is mapped to your Front Door by a CNAME record

When you added a custom domain to your Front Door's frontend hosts, you created a CNAME record in the DNS table of your domain registrar to map it to your Front Door's default .azurefd.net hostname. If that CNAME record still exists and does not contain the afdverify subdomain, the DigiCert Certificate Authority uses it to automatically validate ownership of your custom domain.

If you're using your own certificate, domain validation is not required.

Your CNAME record should be in the following format, where *Name* is your custom domain name and *Value* is your Front Door's default .azurefd.net hostname:

NAME	TYPE	VALUE
<www.contoso.com>	CNAME	contoso.azurefd.net

For more information about CNAME records, see Create the CNAME DNS record.

If your CNAME record is in the correct format, DigiCert automatically verifies your custom domain name and creates a dedicated certificate for your domain name. DigitCert won't send you a verification email and you won't need to approve your request. The certificate is valid for one year and will be autorenewed before it expires. Proceed to Wait for propagation.

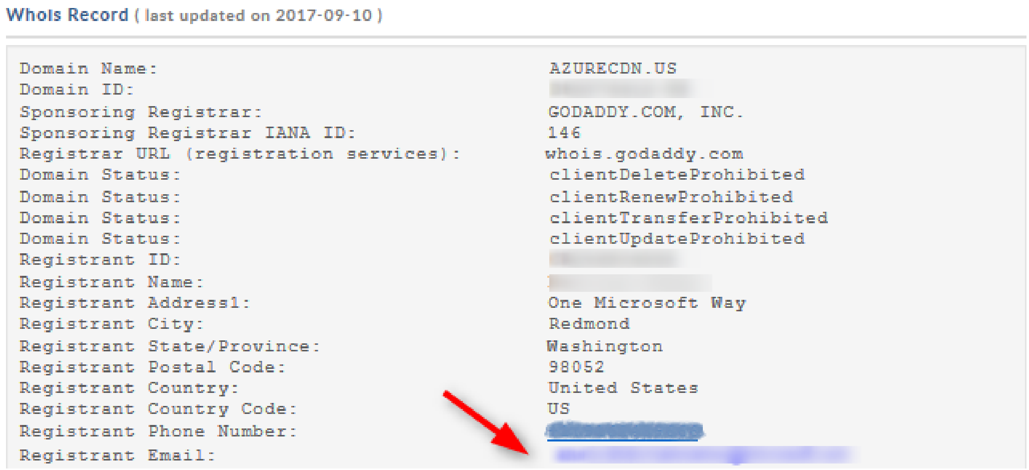
Automatic validation typically takes a few mins. If you don't see your domain validated within an hour, open a support ticket.

> **NOTE**
>
> If you have a Certificate Authority Authorization (CAA) record with your DNS provider, it must include DigiCert as a valid CA. A CAA record allows domain owners to specify with their DNS providers which CAs are authorized to issue certificates for their domain. If a CA receives an order for a certificate for a domain that has a CAA record and that CA is not listed as an authorized issuer, it is prohibited from issuing the certificate to that domain or subdomain. For information about managing CAA records, see Manage CAA records. For a CAA record tool, see CAA Record Helper.

Custom domain is not mapped to your Front Door

If the CNAME record entry for your endpoint no longer exists or it contains the afdverify subdomain, follow the rest of the instructions in this step.

After you enable HTTPS on your custom domain, the DigiCert CA validates ownership of your domain by contacting its registrant, according to the domain's WHOIS registrant information. Contact is made via the email address (by default) or the phone number listed in the WHOIS registration. You must complete domain validation before HTTPS will be active on your custom domain. You have six business days to approve the domain. Requests that are not approved within six business days are automatically canceled.



WhoIs Record (last updated on 2017-09-10)

```
Domain Name:                                  AZURECDN.US
Domain ID:                                    [redacted]
Sponsoring Registrar:                         GODADDY.COM, INC.
Sponsoring Registrar IANA ID:                 146
Registrar URL (registration services):       whois.godaddy.com
Domain Status:                                clientDeleteProhibited
Domain Status:                                clientRenewProhibited
Domain Status:                                clientTransferProhibited
Domain Status:                                clientUpdateProhibited
Registrant ID:                                [redacted]
Registrant Name:                              [redacted]
Registrant Address1:                          One Microsoft Way
Registrant City:                              Redmond
Registrant State/Province:                    Washington
Registrant Postal Code:                       98052
Registrant Country:                           United States
Registrant Country Code:                      US
Registrant Phone Number:                      [redacted]
Registrant Email:                             [redacted]
```

DigiCert also sends a verification email to additional email addresses. If the WHOIS registrant information is private, verify that you can approve directly from one of the following addresses:

admin@<your-domain-name.com>
administrator@<your-domain-name.com>

webmaster@<your-domain-name.com>
hostmaster@<your-domain-name.com>
postmaster@<your-domain-name.com>

You should receive an email in a few minutes, similar to the following example, asking you to approve the request. If you are using a spam filter, add admin@digicert.com to its allow list. If you don't receive an email within 24 hours, contact Microsoft support.

When you click on the approval link, you are directed to an online approval form. Follow the instructions on the form; you have two verification options:

- You can approve all future orders placed through the same account for the same root domain; for example, contoso.com. This approach is recommended if you plan to add additional custom domains for the same root domain.
- You can approve just the specific host name used in this request. Additional approval is required for subsequent requests.

After approval, DigiCert completes the certificate creation for your custom domain name. The certificate is valid for one year and will be autorenewed before it's expired.

Wait for propagation

After the domain name is validated, it can take up to 6-8 hours for the custom domain HTTPS feature to be activated. When the process is complete, the custom HTTPS status in the Azure portal is set to **Enabled** and the four operation steps in the custom domain dialog are marked as complete. Your custom domain is now ready to use HTTPS.

Operation progress

The following table shows the operation progress that occurs when you enable HTTPS. After you enable HTTPS, four operation steps appear in the custom domain dialog. As each step becomes active, additional substep details appear under the step as it progresses. Not all of these substeps will occur. After a step successfully completes, a green check mark appears next to it.

OPERATION STEP	OPERATION SUBSTEP DETAILS
1 Submitting request	Submitting request
	Your HTTPS request is being submitted.
	Your HTTPS request has been submitted successfully.
2 Domain validation	Domain is automatically validated if it is CNAME mapped to the default .azurefd.net frontend host of your Front Door. Otherwise, a verification request will be sent to the email listed in your domain's registration record (WHOIS registrant). Verify the domain as soon as possible.
	Your domain ownership has been successfully validated.
	Domain ownership validation request expired (customer likely didn't respond within 6 days). HTTPS will not be enabled on your domain. *
	Domain ownership validation request was rejected by the customer. HTTPS will not be enabled on your domain. *

OPERATION STEP	OPERATION SUBSTEP DETAILS
3 Certificate provisioning	The certificate authority is currently issuing the certificate needed to enable HTTPS on your domain.
	The certificate has been issued and is currently being deployed for your Front Door. This process could take up to 1 hour.
	The certificate has been successfully deployed for your Front Door.
4 Complete	HTTPS has been successfully enabled on your domain.

* This message doesn't appear unless an error has occurred.

If an error occurs before the request is submitted, the following error message is displayed:

```
We encountered an unexpected error while processing your HTTPS request. Please try again and contact support if the
issue persists.
```

Clean up resources - disable HTTPS

In the preceding steps, you enabled the HTTPS protocol on your custom domain. If you no longer want to use your custom domain with HTTPS, you can disable HTTPS by performing theses steps:

Disable the HTTPS feature

1. In the Azure portal, browse to your **Azure Front Door Service** configuration.

2. In the list of frontend hosts, click the custom domain for which you want to disable HTTPS.

3. Click **Disabled** to disable HTTPS, then click **Save**.

Wait for propagation

After the custom domain HTTPS feature is disabled, it can take up to 6-8 hours for it to take effect. When the process is complete, the custom HTTPS status in the Azure portal is set to **Disabled** and the three operation steps in the custom domain dialog are marked as complete. Your custom domain can no longer use HTTPS.

Operation progress

The following table shows the operation progress that occurs when you disable HTTPS. After you disable HTTPS, three operation steps appear in the Custom domain dialog. As each step becomes active, additional details appear under the step. After a step successfully completes, a green check mark appears next to it.

OPERATION PROGRESS	OPERATION DETAILS
1 Submitting request	Submitting your request
2 Certificate deprovisioning	Deleting certificate
3 Complete	Certificate deleted

Frequently asked questions

1. *Who is the certificate provider and what type of certificate is used?*

   A dedicated/single certificate, provided by Digicert, is used for your custom domain.

2. *Do you use IP-based or SNI TLS/SSL?*

Azure Front Door Service uses SNI TLS/SSL.

3. *What if I don't receive the domain verification email from DigiCert?*

   If you have a CNAME entry for your custom domain that points directly to your endpoint hostname (and you are not using the afdverify subdomain name), you won't receive a domain verification email. Validation occurs automatically. Otherwise, if you don't have a CNAME entry and you haven't received an email within 24 hours, contact Microsoft support.

4. *Is using a SAN certificate less secure than a dedicated certificate?*

   A SAN certificate follows the same encryption and security standards as a dedicated certificate. All issued SSL certificates use SHA-256 for enhanced server security.

5. *Do I need a Certificate Authority Authorization record with my DNS provider?*

   No, a Certificate Authority Authorization record is not currently required. However, if you do have one, it must include DigiCert as a valid CA.

Next steps

- Learn how to create a Front Door.
- Learn how Front Door works.

How to set up a geo-filtering WAF policy for your Front Door

11/19/2019 • 2 minutes to read • Edit Online

This tutorial shows how to use Azure PowerShell to create a sample geo-filtering policy and associate the policy with your existing Front Door frontend host. This sample geo-filtering policy will block requests from all other countries/regions except United States.

If you don't have an Azure subscription, create a free account now.

Prerequisites

Before you begin to set up a geo-filter policy, set up your PowerShell environment and create a Front Door profile.

Set up your PowerShell environment

Azure PowerShell provides a set of cmdlets that use the Azure Resource Manager model for managing your Azure resources.

You can install Azure PowerShell on your local machine and use it in any PowerShell session. Follow the instructions on the page, to sign in with your Azure credentials, and install the Az PowerShell module.

Connect to Azure with an interactive dialog for sign-in

```
Install-Module -Name Az
Connect-AzAccount
```

Make sure you have the current version of PowerShellGet installed. Run below command and reopen PowerShell.

```
Install-Module PowerShellGet -Force -AllowClobber
```

Install Az.FrontDoor module

```
Install-Module -Name Az.FrontDoor
```

Create a Front Door profile

Create a Front Door profile by following the instructions described in Quickstart: Create a Front Door profile.

Define geo-filtering match condition

Create a sample match condition that selects requests not coming from "US" using New-AzFrontDoorWafMatchConditionObject on parameters when creating a match condition. Two letter country codes to country mapping are provided here.

```
$nonUSGeoMatchCondition = New-AzFrontDoorWafMatchConditionObject `
-MatchVariable RemoteAddr `
-OperatorProperty GeoMatch `
-NegateCondition $true `
-MatchValue "US"
```

Add geo-filtering match condition to a rule with Action and Priority

Create a CustomRule object `nonUSBlockRule` based on the match condition, an Action, and a Priority using New-AzFrontDoorWafCustomRuleObject. A CustomRule can have multiple MatchCondition. In this example, Action is set to Block and Priority to 1, the highest priority.

```
$nonUSBlockRule = New-AzFrontDoorWafCustomRuleObject `
-Name "geoFilterRule" `
-RuleType MatchRule `
-MatchCondition $nonUSGeoMatchCondition `
-Action Block `
-Priority 1
```

Add rules to a policy

Find the name of the resource group that contains the Front Door profile using `Get-AzResourceGroup`. Next, create a `geoPolicy` policy object containing `nonUSBlockRule` using New-AzFrontDoorWafPolicy in the specified resource group that contains the Front Door profile. You must provide a unique name for the geo policy.

The below example uses the Resource Group name *myResourceGroupFD1* with the assumption that you have created the Front Door profile using instructions provided in the Quickstart: Create a Front Door article. In the below example, replace the policy name *geoPolicyAllowUSOnly* with a unique policy name.

```
$geoPolicy = New-AzFrontDoorWafPolicy `
-Name "geoPolicyAllowUSOnly" `
-resourceGroupName myResourceGroupFD1 `
-Customrule $nonUSBlockRule  `
-Mode Prevention `
-EnabledState Enabled
```

Link WAF policy to a Front Door frontend host

Link the WAF policy object to the existing Front Door frontend host and update Front Door properties.

To do so, first retrieve your Front Door object using Get-AzFrontDoor.

```
$geoFrontDoorObjectExample = Get-AzFrontDoor -ResourceGroupName myResourceGroupFD1
$geoFrontDoorObjectExample[0].FrontendEndpoints[0].WebApplicationFirewallPolicyLink = $geoPolicy.Id
```

Next, set the frontend WebApplicationFirewallPolicyLink property to the resourceId of the `geoPolicy` using Set-AzFrontDoor.

```
Set-AzFrontDoor -InputObject $geoFrontDoorObjectExample[0]
```

NOTE

You only need to set WebApplicationFirewallPolicyLink property once to link a WAF policy to a Front Door frontend host. Subsequent policy updates are automatically applied to the frontend host.

Next steps

- Learn about Azure web application firewall.

- Learn how to create a Front Door.

Azure Resource Manager deployment model templates for Front Door

11/19/2019 • 2 minutes to read • Edit Online

The following table includes links to Azure Resource Manager deployment model templates for Azure Front Door Service.

Create a basic Front Door	Creates a basic Front Door configuration with a single backend.
Create a Front Door with multiple backends and backend pools and URL based routing	Creates a Front Door with load balancing configured for multiple backends in ta backend pool and also across backend pools based on URL path.
Onboard a custom domain with HTTPS (Front Door managed cert) with Front Door	Add a custom domain to your Front Door and enable HTTPS traffic for it with a Front Door managed certificate generated via DigiCert.
Create Front Door with geo filtering	Create a Front Door that allows/blocks traffic from certain countries/regions.
Control Health Probes for your backends on Front Door	Update your Front Door to change the health probe settings by updating the probe path and also the intervals in which the probes will be sent.
Create Front Door with Active/Standby backend configuration	Creates a Front Door that demonstrates priority-based routing for Active/Standby application topology, that is, by default send all traffic to the primary (highest-priority) backend until it becomes unavailable.
Create Front Door with caching enabled for certain routes	Creates a Front Door with caching enabled for the defined routing configuration thus caching any static assets for your workload.
Configure Session Affinity for your Front Door host names	Updates a Front Door to enable session affinity for your frontend host, thereby, sending subsequent traffic from the same user session to the same backend.
Configure Front Door for client IP whitelisting or blacklisting	Configures a Front Door to restrict traffic certain client IPs using custom access control using client IPs.
Configure Front Door to take action with specific http parameters	Configures a Front Door to allow or block certain traffic based on the http parameters in the incoming request by using custom rules for access control using http parameters.
Configure Front Door rate limiting	Configures a Front Door to rate limit incoming traffic for a given frontend host.

Next steps

- Learn how to create a Front Door.
- Learn how Front Door works.

Backends and backend pools in Azure Front Door Service

11/20/2019 • 5 minutes to read • Edit Online

This article describes concepts about how to map your app deployment with Azure Front Door Service. It also explains the different terms in Front Door configuration around app backends.

Backends

A backend is equal to an app's deployment instance in a region. Front Door Service supports both Azure and non-Azure backends, so the region isn't only restricted to Azure regions. Also, it can be your on-premises datacenter or an app instance in another cloud.

Front Door Service backends refer to the host name or public IP of your app, which can serve client requests. Backends shouldn't be confused with your database tier, storage tier, and so on. Backends should be viewed as the public endpoint of your app backend. When you add a backend in a Front Door backend pool, you must also add the following:

- **Backend host type**. The type of resource you want to add. Front Door Service supports autodiscovery of your app backends from app service, cloud service, or storage. If you want a different resource in Azure or even a non-Azure backend, select **Custom host**.

  > IMPORTANT
  >
  > During configuration, APIs don't validate if the backend is inaccessible from Front Door environments. Make sure that Front Door can reach your backend.

- **Subscription and Backend host name**. If you haven't selected **Custom host** for backend host type, select your backend by choosing the appropriate subscription and the corresponding backend host name in the UI.
- **Backend host header**. The host header value sent to the backend for each request. For more information, see Backend host header.
- **Priority**. Assign priorities to your different backends when you want to use a primary service backend for all traffic. Also, provide backups if the primary or the backup backends are unavailable. For more information, see Priority.
- **Weight**. Assign weights to your different backends to distribute traffic across a set of backends, either evenly or according to weight coefficients. For more information, see Weights.

Backend host header

Requests forwarded by Front Door to a backend include a host header field that the backend uses to retrieve the targeted resource. The value for this field typically comes from the backend URI and has the host and port.

For example, a request made for www.contoso.com will have the host header www.contoso.com. If you use Azure portal to configure your backend, the default value for this field is the host name of the backend. If your backend is contoso-westus.azurewebsites.net, in the Azure portal, the autopopulated value for the backend host header will be contoso-westus.azurewebsites.net. However, if you use Azure Resource Manager templates or another method without explicitly setting this field, Front Door Service will send the incoming host name as the value for the host header. If the request was made for www.contoso.com, and your backend is contoso-westus.azurewebsites.net that has an empty header field, Front Door Service will set the host header as www.contoso.com.

Most app backends (Azure Web Apps, Blob storage, and Cloud Services) require the host header to match the domain of the backend. However, the frontend host that routes to your backend will use a different hostname such as www.contoso.azurefd.net.

If your backend requires the host header to match the backend host name, make sure that the backend host header includes the host name backend.

Configuring the backend host header for the backend

To configure the **backend host header** field for a backend in the backend pool section:

1. Open your Front Door resource and select the backend pool with the backend to configure.
2. Add a backend if you haven't done so, or edit an existing one.
3. Set the backend host header field to a custom value or leave it blank. The hostname for the incoming request will be used as the host header value.

Backend pools

A backend pool in Front Door Service refers to the set of backends that receive similar traffic for their app. In other words, it's a logical grouping of your app instances across the world that receive the same traffic and respond with expected behavior. These backends are deployed across different regions or within the same region. All backends can be in Active/Active deployment mode or what is defined as Active/Passive configuration.

A backend pool defines how the different backends should be evaluated via health probes. It also defines how load balancing occurs between them.

Health probes

Front Door Service sends periodic HTTP/HTTPS probe requests to each of your configured backends. Probe requests determine the proximity and health of each backend to load balance your end-user requests. Health probe settings for a backend pool define how we poll the health status of app backends. The following settings are available for load-balancing configuration:

- **Path**. The URL used for probe requests for all the backends in the backend pool. For example, if one of your backends is contoso-westus.azurewebsites.net and the path is set to /probe/test.aspx, then Front Door Service environments, assuming the protocol is set to HTTP, will send health probe requests to http://contoso-westus.azurewebsites.net/probe/test.aspx.
- **Protocol**. Defines whether to send the health probe requests from Front Door Service to your backends with HTTP or HTTPS protocol.
- **Interval (seconds)**. Defines the frequency of health probes to your backends, or the intervals in which each of the Front Door environments sends a probe.

  > **NOTE**
  >
  > For faster failovers, set the interval to a lower value. The lower the value, the higher the health probe volume your backends receive. For example, if the interval is set to 30 seconds with 90 Front Door environments or POPs globally, each backend will receive about 3-5 probe requests per second.

For more information, see Health probes.

Load-balancing settings

Load-balancing settings for the backend pool define how we evaluate health probes. These settings determine if the backend is healthy or unhealthy. They also check how to load-balance traffic between different backends in the backend pool. The following settings are available for load-balancing configuration:

- **Sample size**. Identifies how many samples of health probes we need to consider for backend health evaluation.
- **Successful sample size**. Defines the sample size as previously mentioned, the number of successful samples needed to call the backend healthy. For example, assume a Front Door health probe interval is 30 seconds, sample size is 5, and successful sample size is 3. Each time we evaluate the health probes for your backend, we look at the last five samples over 150 seconds (5 x 30). At least three successful probes are required to declare the backend as healthy.
- **Latency sensitivity (additional latency)**. Defines whether you want Front Door to send the request to backends within the latency measurement sensitivity range or forward the request to the closest backend.

For more information, see Least latency based routing method.

Next steps

- Create a Front Door profile
- How Front Door works

Caching with Azure Front Door Service

11/8/2019 • 5 minutes to read • Edit Online

The following document specifies behavior for Front Door with routing rules that have enabled caching.

Delivery of large files

Azure Front Door Service delivers large files without a cap on file size. Front Door uses a technique called object chunking. When a large file is requested, Front Door retrieves smaller pieces of the file from the backend. After receiving a full or byte-range file request, a Front Door environment requests the file from the backend in chunks of 8 MB.

After the chunk arrives at the Front Door environment, it is cached and immediately served to the user. Front Door then pre-fetches the next chunk in parallel. This pre-fetch ensures that the content stays one chunk ahead of the user, which reduces latency. This process continues until the entire file is downloaded (if requested), all byte ranges are available (if requested), or the client terminates the connection.

For more information on the byte-range request, read RFC 7233. Front Door caches any chunks as they're received and so the entire file doesn't need to be cached on the Front Door cache. Subsequent requests for the file or byte ranges are served from the cache. If not all the chunks are cached, pre-fetching is used to request chunks from the backend. This optimization relies on the ability of the backend to support byte-range requests; if the backend doesn't support byte-range requests, this optimization isn't effective.

File compression

Front Door can dynamically compress content on the edge, resulting in a smaller and faster response to your clients. All files are eligible for compression. However, a file must be of a MIME type that eligible for compression list. Currently, Front Door does not allow this list to be changed. The current list is:

- "application/eot"
- "application/font"
- "application/font-sfnt"
- "application/javascript"
- "application/json"
- "application/opentype"
- "application/otf"
- "application/pkcs7-mime"
- "application/truetype"
- "application/ttf",
- "application/vnd.ms-fontobject"
- "application/xhtml+xml"
- "application/xml"
- "application/xml+rss"
- "application/x-font-opentype"
- "application/x-font-truetype"
- "application/x-font-ttf"

- "application/x-httpd-cgi"
- "application/x-mpegurl"
- "application/x-opentype"
- "application/x-otf"
- "application/x-perl"
- "application/x-ttf"
- "application/x-javascript"
- "font/eot"
- "font/ttf"
- "font/otf"
- "font/opentype"
- "image/svg+xml"
- "text/css"
- "text/csv"
- "text/html"
- "text/javascript"
- "text/js", "text/plain"
- "text/richtext"
- "text/tab-separated-values"
- "text/xml"
- "text/x-script"
- "text/x-component"
- "text/x-java-source"

Additionally, the file must also be between 1 KB and 8 MB in size, inclusive.

These profiles support the following compression encodings:

- Gzip (GNU zip)
- Brotli

If a request supports gzip and Brotli compression, Brotli compression takes precedence.
When a request for an asset specifies compression and the request results in a cache miss, Front Door performs compression of the asset directly on the POP server. Afterward, the compressed file is served from the cache. The resulting item is returned with a transfer-encoding: chunked.

Query string behavior

With Front Door, you can control how files are cached for a web request that contains a query string. In a web request with a query string, the query string is that portion of the request that occurs after a question mark (?). A query string can contain one or more key-value pairs, in which the field name and its value are separated by an equals sign (=). Each key-value pair is separated by an ampersand (&). For example, `http://www.contoso.com/content.mov?field1=value1&field2=value2`. If there is more than one key-value pair in a query string of a request, their order does not matter.

- **Ignore query strings**: Default mode. In this mode, Front Door passes the query strings from the requestor to the backend on the first request and caches the asset. All subsequent requests for the asset that are served from the Front Door environment ignore the query strings until the cached asset expires.

- **Cache every unique URL**: In this mode, each request with a unique URL, including the query string, is treated as a unique asset with its own cache. For example, the response from the backend for a request for `www.example.ashx?q=test1` is cached at the Front Door environment and returned for subsequent caches with the same query string. A request for `www.example.ashx?q=test2` is cached as a separate asset with its own time-to-live setting.

Cache purge

Front Door will cache assets until the asset's time-to-live (TTL) expires. After the asset's TTL expires, when a client requests the asset, the Front Door environment will retrieve a new updated copy of the asset to serve the client request and store refresh the cache.
The best practice to make sure your users always obtain the latest copy of your assets is to version your assets for each update and publish them as new URLs. Front Door will immediately retrieve the new assets for the next client requests. Sometimes you may wish to purge cached content from all edge nodes and force them all to retrieve new updated assets. This might be due to updates to your web application, or to quickly update assets that contain incorrect information.

Select what assets you wish to purge from the edge nodes. If you wish to clear all assets, click the Purge all checkbox. Otherwise, type the path of each asset you wish to purge in the Path textbox. Below formats are supported in the path.

1. **Single URL purge**: Purge individual asset by specifying the full URL, with the file extension, for example, /pictures/strasbourg.png;
2. **Wildcard purge**: Asterisk (*) may be used as a wildcard. Purge all folders, subfolders and files under an endpoint with /* in the path or purge all subfolders and files under a specific folder by specifying the folder followed by /*, for example, /pictures/*.
3. **Root domain purge**: Purge the root of the endpoint with "/" in the path.

Cache purges on the Front Door are case-insensitive. Additionally, they are query string agnostic, meaning purging a URL will purge all query-string variations of it.

Cache expiration

The following order of headers is used in order to determine how long an item will be stored in our cache:

1. Cache-Control: s-maxage=<seconds>
2. Cache-Control: max-age=<seconds>
3. Expires: <http-date>

Cache-Control response headers that indicate that the response won't be cached such as Cache-Control: private, Cache-Control: no-cache, and Cache-Control: no-store are honored. However, if there are multiple requests in-flight at a POP for the same URL, they may share the response. If no Cache-Control is present the default behavior is that AFD will cache the resource for X amount of time where X is randomly picked between 1 to 3 days.

Request headers

The following request headers will not be forwarded to a backend when using caching.

- Authorization
- Content-Length
- Transfer-Encoding

Next steps

- Learn how to create a Front Door.
- Learn how Front Door works.

What is geo-filtering on a domain for Azure Front Door?

7/12/2019 • 3 minutes to read • Edit Online

By default, Azure Front Door Service responds to user requests regardless of the location of the user making the request. However, in some cases, you may want to restrict access to your web applications by country/region. Web application firewall (WAF) service at Front Door enables you to define a policy using custom access rules for specific path on your endpoint to allow or block access from specified countries/regions.

A WAF policy usually includes a set of custom rules. A rule consists of match conditions, an action, and a priority. In match condition, you define a match variable, operator, and match value. For geo filtering rule, match variable is REMOTE_ADDR, operator is GeoMatch, value is the two letter country code of interest. You may combine a GeoMatch condition and a REQUEST_URI string match condition to create a path-based geo-filtering rule.

You can configure a geo-filtering policy for your Front Door by either using Azure PowerShell or by using our quickstart template.

Country code reference

COUNTRY CODE	COUNTRY NAME
AD	Andorra
AE	United Arab Emirates
AF	Afghanistan
AG	Antigua and Barbuda
AL	Albania
AM	Armenia
AO	Angola
AR	Argentina
AS	American Samoa
AT	Austria
AU	Australia
AZ	Azerbaijan
BA	Bosnia and Herzegovina
BB	Barbados

COUNTRY CODE	COUNTRY NAME
BD	Bangladesh
BE	Belgium
BF	Burkina Faso
BG	Bulgaria
BH	Bahrain
BI	Burundi
BJ	Benin
BL	Saint Barthelemy
BN	Brunei Darussalam
BO	Bolivia
BR	Brazil
BS	Bahamas
BT	Bhutan
BW	Botswana
BY	Belarus
BZ	Belize
CA	Canada
CD	Democratic Republic of the Congo
CF	Central African Republic
CH	Switzerland
CI	Cote d'Ivoire
CL	Chile
CM	Cameroon
CN	China
CO	Colombia

COUNTRY CODE	COUNTRY NAME
CR	Costa Rica
CU	Cuba
CV	Cabo Verde
CY	Cyprus
CZ	Czech Republic
DE	Germany
DK	Denmark
DO	Dominican Republic
DZ	Algeria
EC	Ecuador
EE	Estonia
EG	Egypt
ES	Spain
ET	Ethiopia
FI	Finland
FJ	Fiji
FM	Micronesia, Federated States of
FR	France
GB	United Kingdom
GE	Georgia
GF	French Guiana
GH	Ghana
GN	Guinea
GP	Guadeloupe
GR	Greece

COUNTRY CODE	COUNTRY NAME
GT	Guatemala
GY	Guyana
HK	Hong Kong SAR
HN	Honduras
HR	Croatia
HT	Haiti
HU	Hungary
ID	Indonesia
IE	Ireland
IL	Israel
IN	India
IQ	Iraq
IR	Iran, Islamic Republic of
IS	Iceland
IT	Italy
JM	Jamaica
JO	Jordan
JP	Japan
KE	Kenya
KG	Kyrgyzstan
KH	Cambodia
KI	Kiribati
KN	Saint Kitts and Nevis
KP	Korea, Democratic People's Republic of
KR	Korea, Republic of

COUNTRY CODE	COUNTRY NAME
KW	Kuwait
KY	Cayman Islands
KZ	Kazakhstan
LA	Lao People's Democratic Republic
LB	Lebanon
LI	Liechtenstein
LK	Sri Lanka
LR	Liberia
LS	Lesotho
LT	Lithuania
LU	Luxembourg
LV	Latvia
LY	Libya
MA	Morocco
MD	Moldova, Republic of
MG	Madagascar
MK	North Macedonia
ML	Mali
MM	Myanmar
MN	Mongolia
MO	Macao SAR
MQ	Martinique
MR	Mauritania
MT	Malta
MV	Maldives

COUNTRY CODE	COUNTRY NAME
MW	Malawi
MX	Mexico
MY	Malaysia
MZ	Mozambique
NA	Namibia
NE	Niger
NG	Nigeria
NI	Nicaragua
NL	Netherlands
NO	Norway
NP	Nepal
NR	Nauru
NZ	New Zealand
OM	Oman
PA	Panama
PE	Peru
PH	Philippines
PK	Pakistan
PL	Poland
PR	Puerto Rico
PT	Portugal
PW	Palau
PY	Paraguay
QA	Qatar
RE	Reunion

COUNTRY CODE	COUNTRY NAME
RO	Romania
RS	Serbia
RU	Russian Federation
RW	Rwanda
SA	Saudi Arabia
SD	Sudan
SE	Sweden
SG	Singapore
SI	Slovenia
SK	Slovakia
SN	Senegal
SO	Somalia
SR	Suriname
SS	South Sedan
SV	El Salvador
SY	Syrian Arab Republic
SZ	Swaziland
TC	Turks and Caicos Islands
TG	Togo
TH	Thailand
TN	Tunisia
TR	Turkey
TT	Trinidad and Tobago
TW	Taiwan
TZ	Tanzania, United Republic of

COUNTRY CODE	COUNTRY NAME
UA	Ukraine
UG	Uganda
US	United States
UY	Uruguay
UZ	Uzbekistan
VC	Saint Vincent and the Grenadines
VE	Venezuela
VG	Virgin Islands, British
VI	Virgin Islands, U.S.
VN	Vietnam
ZA	South Africa
ZM	Zambia
ZW	Zimbabwe

Next steps

- Learn about application layer security with Front Door.
- Learn how to create a Front Door.

Health probes

8/2/2019 • 2 minutes to read • Edit Online

In order to determine the health of each backend, each Front Door environment periodically sends a synthetic HTTP/HTTPS request to each of your configured backends. Front Door then uses responses from these probes to determine the "best" backends to which it should route real client requests. Note that since Front Door have many edge environments globally, health probe requests volume to your backends can be as high as more than one request per second depends on the health probe frequency configured.

Supported protocols

Front Door supports sending probes over either HTTP or HTTPS protocols. These probes are sent over the same TCP ports configured for routing client requests, and cannot be overridden.

Health probe responses

RESPONSES	DESCRIPTION
Determining Health	A 200 OK status code indicates the backend is healthy. Everything else is considered a failure. If for any reason (including network failure) a valid HTTP response is not received for a probe, the probe is counted as a failure.
Measuring Latency	Latency is the wall-clock time measured from the moment immediately before we send the probe request to the moment when we receive the last byte of the response. We use a new TCP connection for each request, so this measurement is not biased towards backends with existing warm connections.

How Front Door determines backend health

Azure Front Door Service uses the same three-step process below across all algorithms to determine health.

1. Exclude disabled backends.
2. Exclude backends that have health probes errors:
   - This selection is done by looking at the last *n* health probe responses. If at least *x* are healthy, the backend is considered healthy.
   - *n* is configured by changing the SampleSize property in load balancing settings.
   - *x* is configured by changing the SuccessfulSamplesRequired property in load balancing settings.
3. Out of the set of healthy backends in the backend pool, Front Door additionally measures and maintains the latency (round-trip time) for each backend.

Complete health probe failure

If health probes fail for every backend in a backend pool, then Front Door considers all backends healthy and routes traffic in a round robin distribution across all of them.

Once any backend returns to a healthy state, then Front Door will resume the normal load balancing algorithm.

Next steps

- Learn how to create a Front Door.
- Learn how Front Door works.

How Front Door matches requests to a routing rule

3/15/2019 • 3 minutes to read • Edit Online

After establishing a connection and doing an SSL handshake, when a request lands on a Front Door environment one of the first things that Front Door does is determining from all the configurations, which particular routing rule to match the request to and then taking the defined action. The following document explains how Front Door determines which Route configuration to use when processing an HTTP request.

Structure of a Front Door route configuration

A Front Door routing rule configuration is composed of two major parts: a "left-hand side" and a "right-hand side". We match the incoming request to the left-hand side of the route while the right-hand side defines how we process the request.

Incoming match (left-hand side)

The following properties determine whether the incoming request matches the routing rule (or left-hand side):

- **HTTP Protocols** (HTTP/HTTPS)
- **Hosts** (for example, www.foo.com, *.bar.com)
- **Paths** (for example, /*, /users/*, /file.gif)

These properties are expanded out internally so that every combination of Protocol/Host/Path is a potential match set.

Route data (right-hand side)

The decision of how to process the request, depends on whether caching is enabled or not for the specific route. So, if we don't have a cached response for the request, we'll forward the request to the appropriate backend in the configured backend pool.

Route matching

This section will focus on how we match to a given Front Door routing rule. The basic concept is that we always match to the **most-specific match first** looking only at the "left-hand side". We first match based on HTTP protocol, then Frontend host, then the Path.

Frontend host matching

When matching Frontend hosts, we use the logic as below:

1. Look for any routing with an exact match on the host.
2. If no exact frontend hosts match, reject the request and send a 400 Bad Request error.

To explain this process further, let's look at an example configuration of Front Door routes (left-hand side only):

ROUTING RULE	FRONTEND HOSTS	PATH
A	foo.contoso.com	/*
B	foo.contoso.com	/users/*
C	www.fabrikam.com, foo.adventure-works.com	/*, /images/*

If the following incoming requests were sent to Front Door, they would match against the following routing rules from above:

INCOMING FRONTEND HOST	MATCHED ROUTING RULE(S)
foo.contoso.com	A, B
www.fabrikam.com	C
images.fabrikam.com	Error 400: Bad Request
foo.adventure-works.com	C
contoso.com	Error 400: Bad Request
www.adventure-works.com	Error 400: Bad Request
www.northwindtraders.com	Error 400: Bad Request

Path matching

After determining the specific frontend host and filtering possible routing rules to just the routes with that frontend host, Front Door then filters the routing rules based on the request path. We use a similar logic as frontend hosts:

1. Look for any routing rule with an exact match on the Path
2. If no exact match Paths, look for routing rules with a wildcard Path that matches
3. If no routing rules are found with a matching Path, then reject the request and return a 400: Bad Request error HTTP response.

> **NOTE**
>
> Any Paths without a wildcard are considered to be exact-match Paths. Even if the Path ends in a slash, it's still considered exact match.

To explain further, let's look at another set of examples:

ROUTING RULE	FRONTEND HOST	PATH
A	www.contoso.com	/
B	www.contoso.com	/*
C	www.contoso.com	/ab
D	www.contoso.com	/abc
E	www.contoso.com	/abc/
F	www.contoso.com	/abc/*
G	www.contoso.com	/abc/def

ROUTING RULE	FRONTEND HOST	PATH
H	www.contoso.com	/path/

Given that configuration, the following example matching table would result:

INCOMING REQUEST	MATCHED ROUTE
www.contoso.com/	A
www.contoso.com/a	B
www.contoso.com/ab	C
www.contoso.com/abc	D
www.contoso.com/abzzz	B
www.contoso.com/abc/	E
www.contoso.com/abc/d	F
www.contoso.com/abc/def	G
www.contoso.com/abc/defzzz	F
www.contoso.com/abc/def/ghi	F
www.contoso.com/path	B
www.contoso.com/path/	H
www.contoso.com/path/zzz	B

WARNING

If there are no routing rules for an exact-match frontend host with a catch-all route Path (`/*`), then there will not be a match to any routing rule.

Example configuration:

ROUTE	HOST	PATH
A	profile.contoso.com	/api/*

Matching table:

INCOMING REQUEST	MATCHED ROUTE
profile.domain.com/other	None. Error 400: Bad Request

Routing decision

Once we've matched to a single Front Door routing rule, we then need to choose how to process the request. If for the matched routing rule, Front Door has a cached response available then the same gets served back to the client. Otherwise, the next thing that gets evaluated is whether you have configured URL Rewrite (custom forwarding path) for the matched routing rule or not. If there isn't a custom forwarding path defined, then the request gets forwarded to the appropriate backend in the configured backend pool as is. Else, the request path is updated as per the custom forwarding path defined and then forward to the backend.

Next steps

- Learn how to create a Front Door.
- Learn how Front Door works.

HTTP/2 support in Azure Front Door Service

8/2/2019 • 2 minutes to read • Edit Online

Currently, HTTP/2 support is active for all Front Door configurations. No further action is required from customers.

HTTP/2 is a major revision to HTTP/1.1. It provides faster web performance, reduced response time, and improved user experience, while maintaining the familiar HTTP methods, status codes, and semantics. Though HTTP/2 is designed to work with HTTP and HTTPS, many client web browsers only support HTTP/2 over Transport Layer Security (TLS).

HTTP/2 benefits

The benefits of HTTP/2 include:

- **Multiplexing and concurrency**

  Using HTTP 1.1, making multiple resource requests requires multiple TCP connections, and each connection has performance overhead associated with it. HTTP/2 allows multiple resources to be requested on a single TCP connection.

- **Header compression**

  By compressing the HTTP headers for served resources, time on the wire is reduced significantly.

- **Stream dependencies**

  Stream dependencies allow the client to indicate to the server which resources have priority.

HTTP/2 browser support

All of the major browsers have implemented HTTP/2 support in their current versions. Non-supported browsers automatically fallback to HTTP/1.1.

BROWSER	MINIMUM VERSION
Microsoft Edge	12
Google Chrome	43
Mozilla Firefox	38
Opera	32
Safari	9

Next Steps

To learn more about HTTP/2, visit the following resources:

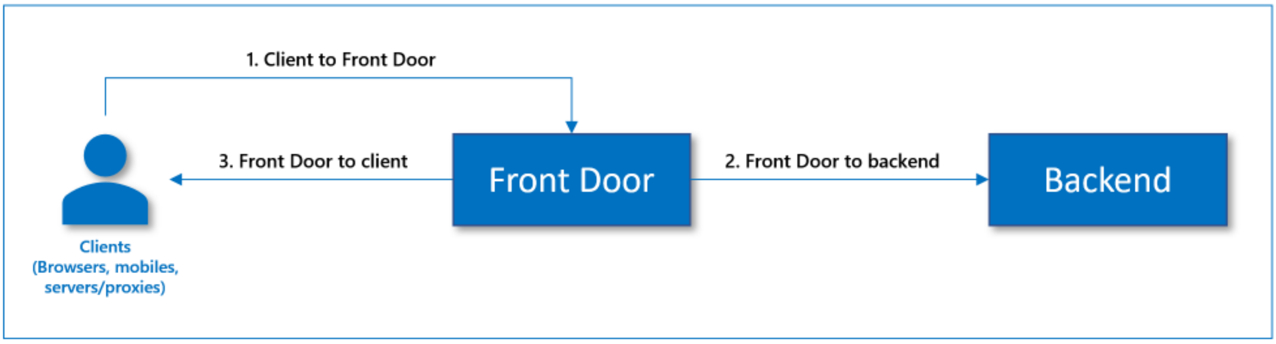
- HTTP/2 specification homepage
- Official HTTP/2 FAQ
- Learn how to create a Front Door.

- Learn how Front Door works.

Protocol support for HTTP headers in Azure Front Door Service

1/23/2020 • 2 minutes to read • Edit Online

This article outlines the protocol that Front Door Service supports with parts of the call path (see image). The following sections provide more information about HTTP headers supported by Front Door Service.



IMPORTANT

Front Door Service doesn't certify any HTTP headers that aren't documented here.

Client to Front Door Service

Front Door Service accepts most headers from the incoming request without modifying them. Some reserved headers are removed from the incoming request if sent, including headers with the X-FD-* prefix.

Front Door Service to backend

Front Door Service includes headers from an incoming request unless removed because of restrictions. Front Door also adds the following headers:

HEADER	EXAMPLE AND DESCRIPTION
Via	Via: 1.1 Azure Front Door adds the client's HTTP version followed by *Azure* as the value for the Via header. This indicates the client's HTTP version and that Front Door was an intermediate recipient for the request between the client and the backend.
X-Azure-ClientIP	X-Azure-ClientIP: 127.0.0.1 Represents the client IP address associated with the request being processed. For example, a request coming from a proxy might add the X-Forwarded-For header to indicate the IP address of the original caller.
X-Azure-SocketIP	X-Azure-SocketIP: 127.0.0.1 Represents the socket IP address associated with the TCP connection that the current request originated from. A request's client IP address might not be equal to its socket IP address because it can be arbitrarily overwritten by a user.

HEADER	EXAMPLE AND DESCRIPTION
X-Azure-Ref	X-Azure-Ref: 0zxV+XAAAAABKMMOjBv2NT4TY6SQVjC0zV1NURURHRTA2MTkANDM3YzgyY2QtMzYwYS00YTU0LTk0YzMtNWZmNzA3NjQ3Nzgz A unique reference string that identifies a request served by Front Door. It's used to search access logs and critical for troubleshooting.
X-Azure-RequestChain	X-Azure-RequestChain: hops=1 A header that Front Door uses to detect request loops, and users should not take a dependency on it.
X-Forwarded-For	X-Forwarded-For: 127.0.0.1 The X-Forwarded-For (XFF) HTTP header field often identifies the originating IP address of a client connecting to a web server through an HTTP proxy or load balancer. If there's an existing XFF header, then Front Door appends the client socket IP to it or adds the XFF header with the client socket IP.
X-Forwarded-Host	X-Forwarded-Host: contoso.azurefd.net The X-Forwarded-Host HTTP header field is a common method used to identify the original host requested by the client in the Host HTTP request header. This is because the host name from Front Door may differ for the backend server handling the request.
X-Forwarded-Proto	X-Forwarded-Proto: http The X-Forwarded-Proto HTTP header field is often used to identify the originating protocol of an HTTP request because Front Door, based on configuration, might communicate with the backend by using HTTPS. This is true even if the request to the reverse proxy is HTTP.
X-FD-HealthProbe	X-FD-HealthProbe HTTP header field is used to identify the health probe from Front Door. If this header set to 1, the request is health probe. You can use when want to strict access from particular Front Door with X-Forwarded-Host header field.

Front Door Service to client

Any headers sent to Front Door from the backend are also passed through to the client. The following are headers sent from Front Door to clients.

HEADER	EXAMPLE
X-Azure-Ref	*X-Azure-Ref: 0zxV+XAAAAABKMMOjBv2NT4TY6SQVjC0zV1NURURHRTA2MTkANDM3YzgyY2QtMzYwYS00YTU0LTk0YzMtNWZmNzA3NjQ3Nzgz* This is a unique reference string that identifies a request served by Front Door. This is critical for troubleshooting as it's used to search access logs.

Next steps

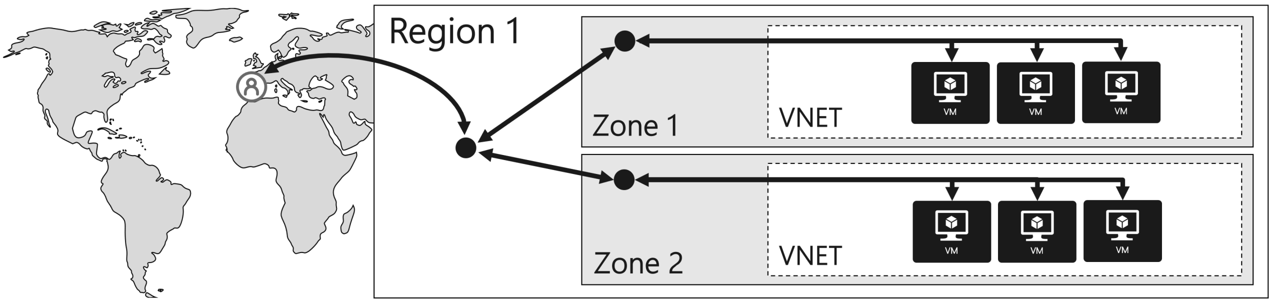
- Create a Front Door
- How Front Door works

Load-balancing with Azure's application delivery suite

4/2/2019 • 6 minutes to read • Edit Online

Introduction

Microsoft Azure provides multiple global and regional services for managing how your network traffic is distributed and load balanced: Traffic Manager, Front Door Service, Application Gateway, and Load Balancer. Along with Azure's many regions and zonal architecture, using these services together enable you to build robust, scalable high-performance applications.



These services are broken into two categories:

1. **Global load balancing services** such as Traffic Manager and Front Door distribute traffic from your end users across your regional backends, across clouds or even your hybrid on-premises services. Global load balancing routes your traffic to your closest service backend and reacts to changes in service reliability or performance to maintain always-on, maximal performance for your users.
2. **Regional load balancing services** such as Standard Load Balancer or Application Gateway provide the ability to distribute traffic within virtual networks (VNETs) across your virtual machines (VMs) or zonal service endpoints within a region.

Combining global and regional services in your application provides an end-to-end reliable, performant, and secure way to route traffic to and from your users to your IaaS, PaaS, or on-premises services. In the next section, we describe each of these services.

Global load balancing

Traffic Manager provides global DNS load balancing. It looks at incoming DNS requests and responds with a healthy backend, in accordance with the routing policy the customer has selected. Options for routing methods are:

- Performance routing to send the requestor to the closest backend in terms of latency.
- Priority routing to direct all traffic to a backend, with other backends as back up.
- Weighted round-robin routing, which distributes traffic based on the weighting that is assigned to each backend.
- Geographic routing to ensure that requestors located in specific geographic regions are directed to the backends mapped to those regions (for example, all requests from Spain should be directed to the France Central Azure region)
- Subnet routing that allows you to map IP address ranges to backends so that requests coming from those will be sent to the specified backend (for example, all users connecting from your corporate HQ's IP address range should get different web content than the general users)

The client connects directly to that backend. Azure Traffic Manager detects when a backend is unhealthy and then redirects the clients to another healthy instance. Refer to Azure Traffic Manager documentation to learn more about the service.

Azure Front Door Service provides dynamic website acceleration (DSA) including global HTTP load balancing. It looks at incoming HTTP requests routes to the closest service backend / region for the specified hostname, URL path, and configured rules.
Front Door terminates HTTP requests at the edge of Microsoft's network and actively probes to detect application or infrastructure health or latency changes. Front Door then always routes traffic to the fastest and available (healthy) backend. Refer to Front Door's routing architecture details and traffic routing methods to learn more about the service.

Regional load balancing

Application Gateway provides application delivery controller (ADC) as a service, offering various Layer 7 load-balancing capabilities for your application. It allows customers to optimize web farm productivity by offloading CPU-intensive SSL termination to the application gateway. Other Layer 7 routing capabilities include round-robin distribution of incoming traffic, cookie-based session affinity, URL path-based routing, and the ability to host multiple websites behind a single application gateway. Application Gateway can be configured as an Internet-facing gateway, an internal-only gateway, or a combination of both. Application Gateway is fully Azure managed, scalable, and highly available. It provides a rich set of diagnostics and logging capabilities for better manageability. Load Balancer is an integral part of the Azure SDN stack, providing high-performance, low-latency Layer 4 load-balancing services for all UDP and TCP protocols. It manages inbound and outbound connections. You can configure public and internal load-balanced endpoints and define rules to map inbound connections to back-end pool destinations by using TCP and HTTP health-probing options to manage service availability.

Choosing a global load balancer

When choosing a global load balancer between Traffic Manager and Azure Front Door for global routing, you should consider what's similar and what's different about the two services. Both services provide

- **Multi-geo redundancy:** If one region goes down, traffic seamlessly routes to the closest region without any intervention from the application owner.
- **Closest region routing:** Traffic is automatically routed to the closest region

The following table describes the differences between Traffic Manager and Azure Front Door Service:

TRAFFIC MANAGER	AZURE FRONT DOOR SERVICE
Any protocol: Because Traffic Manager works at the DNS layer, you can route any type of network traffic; HTTP, TCP, UDP, etc.	**HTTP acceleration:** With Front Door traffic is proxied at the Edge of Microsoft's network. Because of this, HTTP(S) requests see latency and throughput improvements reducing latency for SSL negotiation and using hot connections from AFD to your application.
On-premises routing: With routing at a DNS layer, traffic always goes from point to point. Routing from your branch office to your on premises datacenter can take a direct path; even on your own network using Traffic Manager.	**Independent scalability:** Because Front Door works with the HTTP request, requests to different URL paths can be routed to different backend / regional service pools (microservices) based on rules and the health of each application microservice.
Billing format: DNS-based billing scales with your users and for services with more users, plateaus to reduce cost at higher usage.	**Inline security:** Front Door enables rules such as rate limiting and IP ACL-ing to let you protect your backends before traffic reaches your application.

Because of the performance, operability and security benefits to HTTP workloads with Front Door, we recommend customers use Front Door for their HTTP workloads. Traffic Manager and Front Door can be used in parallel to serve all traffic for your application.

Building with Azure's application delivery suite

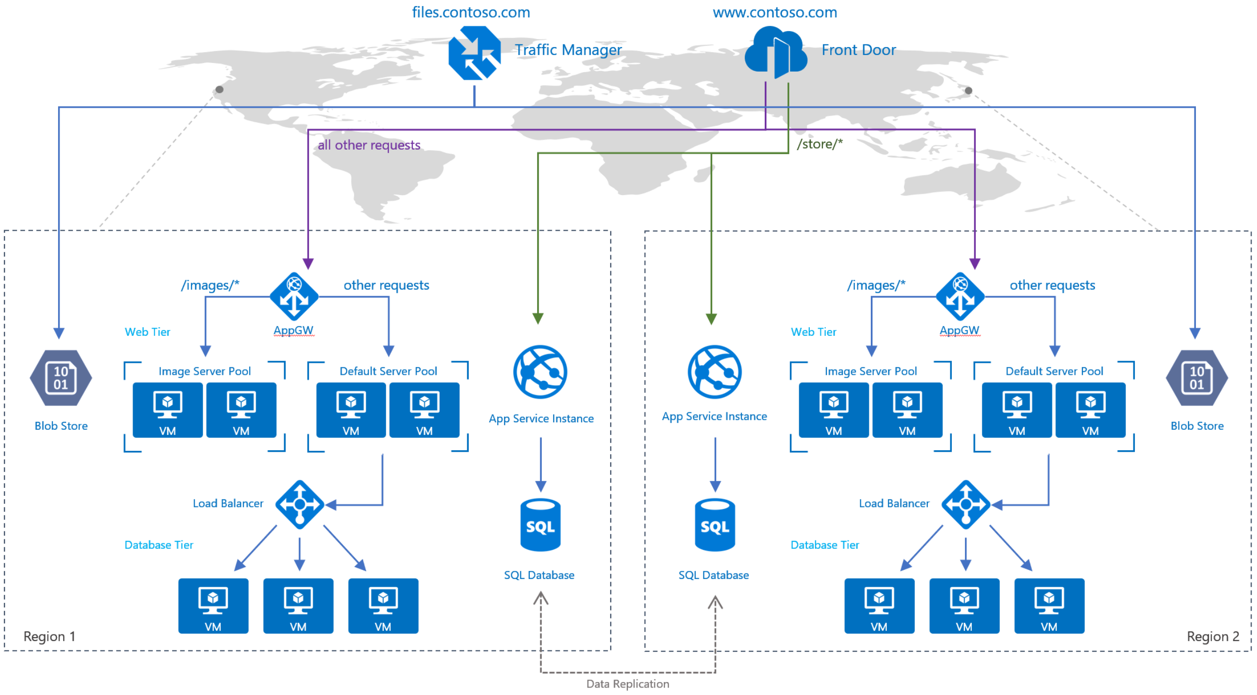
We recommend that all websites, APIs, services be geographically redundant and deliver traffic to its users from the closest (lowest latency) location to them whenever possible. Combining services from Traffic Manager, Front Door Service, Application Gateway, and Load Balancer enables you to build geographically and zonally redundant to maximize reliability, scale, and performance.

In the following diagram, we describe an example service that uses a combination of all these services to build a global web service. In this case, the architect has used Traffic Manager to route to global backends for file and object delivery, while using Front Door to globally route URL paths that match the pattern /store/* to the service they've migrated to App Service while routing all other requests to regional Application Gateways.

In the region, for their IaaS service, the application developer has decided that any URLs that match the pattern /images/* are served from a dedicated pool of VMs that are different from the rest of the web farm.

Additionally, the default VM pool serving the dynamic content needs to talk to a back-end database that is hosted on a high-availability cluster. The entire deployment is set up through Azure Resource Manager.

The following diagram shows the architecture of this scenario:



> **NOTE**
>
> This example is only one of many possible configurations of the load-balancing services that Azure offers. Traffic Manager, Front Door, Application Gateway, and Load Balancer can be mixed and matched to best suit your load-balancing needs. For example, if SSL offload or Layer 7 processing is not necessary, Load Balancer can be used in place of Application Gateway.

Next Steps

- Learn how to create a Front Door.
- Learn how Front Door works.

Monitoring metrics and logs in Azure Front Door Service

1/8/2020 • 3 minutes to read • Edit Online

By using Azure Front Door Service, you can monitor resources in the following ways:

- **Metrics**. Azure Front Door currently has seven metrics to view performance counters.
- **Logs**. Activity and diagnostic logs allow performance, access, and other data to be saved or consumed from a resource for monitoring purposes.

Metrics

Metrics are a feature for certain Azure resources that allow you to view performance counters in the portal. The following are available Front Door metrics:

METRIC	METRIC DISPLAY NAME	UNIT	DIMENSIONS	DESCRIPTION
RequestCount	Request Count	Count	HttpStatus HttpStatusGroup ClientRegion ClientCountry	The number of client requests served by Front Door.
RequestSize	Request Size	Bytes	HttpStatus HttpStatusGroup ClientRegion ClientCountry	The number of bytes sent as requests from clients to Front Door.
ResponseSize	Response Size	Bytes	HttpStatus HttpStatusGroup ClientRegion ClientCountry	The number of bytes sent as responses from Front Door to clients.
TotalLatency	Total Latency	Milliseconds	HttpStatus HttpStatusGroup ClientRegion ClientCountry	The time calculated from the client request received by Front Door until the client acknowledged the last response byte from Front Door.
BackendRequestCount	Backend Request Count	Count	HttpStatus HttpStatusGroup Backend	The number of requests sent from Front Door to backends.
BackendRequestLatency	Backend Request Latency	Milliseconds	Backend	The time calculated from when the request was sent by Front Door to the backend until Front Door received the last response byte from the backend.

METRIC	METRIC DISPLAY NAME	UNIT	DIMENSIONS	DESCRIPTION
BackendHealthPercentage	Backend Health Percentage	Percent	Backend BackendPool	The percentage of successful health probes from Front Door to backends.
WebApplicationFirewallRequestCount	Web Application Firewall Request Count	Count	PolicyName RuleName Action	The number of client requests processed by the application layer security of Front Door.

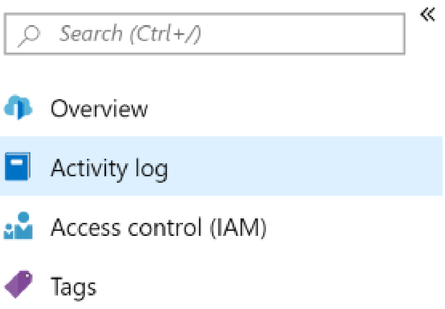
Activity logs

Activity logs provide information about the operations done on Front Door Service. They also determine the what, who, and when for any write operations (put, post, or delete) taken on Front Door Service.

> **NOTE**
>
> Activity logs don't include read (get) operations. They also don't include operations that you perform by using either the Azure portal or the original Management API.

Access activity logs in your Front Door Service or all the logs of your Azure resources in Azure Monitor. To view activity logs:

1. Select your Front Door instance.
2. Select **Activity log**.

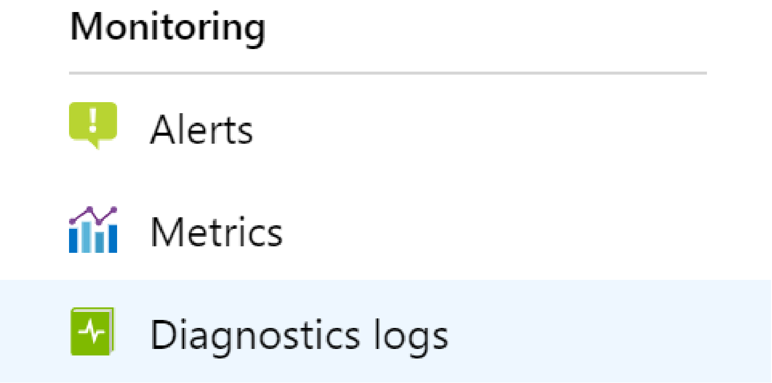


3. Choose a filtering scope, and then select **Apply**.

Diagnostic logs

Diagnostic logs provide rich information about operations and errors that are important for auditing and troubleshooting. Diagnostic logs differ from activity logs.

Activity logs provide insights into the operations done on Azure resources. Diagnostic logs provide insight into operations that your resource performed. For more information, see Azure Monitor diagnostic logs.



To configure diagnostic logs for your Front Door Service:

1. Select your Azure Front Door service.
2. Choose **Diagnostic settings**.
3. Select **Turn on diagnostics**. Archive diagnostic logs along with metrics to a storage account, stream them to an event hub, or send them to Azure Monitor logs.

Front Door Service currently provides diagnostic logs (batched hourly). Diagnostic logs provide individual API requests with each entry having the following schema:

PROPERTY	DESCRIPTION
ClientIp	The IP address of the client that made the request.
ClientPort	The IP port of the client that made the request.
HttpMethod	HTTP method used by the request.
HttpStatusCode	The HTTP status code returned from the proxy.
HttpStatusDetails	Resulting status on the request. Meaning of this string value can be found at a Status reference table.
HttpVersion	Type of the request or connection.
RequestBytes	The size of the HTTP request message in bytes, including the request headers and the request body.
RequestUri	URI of the received request.
ResponseBytes	Bytes sent by the backend server as the response.
RoutingRuleName	The name of the routing rule that the request matched.
SecurityProtocol	The TLS/SSL protocol version used by the request or null if no encryption.
TimeTaken	The length of time that the action took, in milliseconds.
UserAgent	The browser type that the client used

PROPERTY	DESCRIPTION
TrackingReference	The unique reference string that identifies a request served by Front Door, also sent as X-Azure-Ref header to the client. Required for searching details in the access logs for a specific request.

Next steps

- Create a Front Door profile
- How Front Door works

Routing architecture overview

9/24/2018 • 2 minutes to read • Edit Online

The Azure Front Door Service when it receives your client requests then it either answers them (if caching is enabled) or forwards them to the appropriate application backend (as a reverse proxy).

There are opportunities to optimize the traffic when routing to Azure Front Door as well as when routing to backends.

Selecting the Front Door environment for traffic routing (Anycast)

Routing to the Azure Front Door environments leverages Anycast for both DNS (Domain Name System) and HTTP (Hypertext Transfer Protocol) traffic, so user traffic will go to the closest environment in terms of network topology (fewest hops). This architecture typically offers better round-trip times for end users (maximizing the benefits of Split TCP). Front Door organizes its environments into primary and fallback "rings". The outer ring has environments that are closer to users, offering lower latencies. The inner ring has environments that can handle the failover for the outer ring environment in case an issue happens. The outer ring is the preferred target for all traffic, but the inner ring is necessary to handle traffic overflow from the outer ring. In terms of VIPs (Virtual Internet Protocol addresses), each frontend host, or domain served by Front Door is assigned a primary VIP, which is announced by environments in both the inner and outer ring, as well as a fallback VIP, which is only announced by environments in the inner ring.

This overall strategy ensures that requests from your end users always reach the closest Front Door environment and that even if the preferred Front Door environment is unhealthy then traffic automatically moves to the next closest environment.

Connecting to Front Door environment (Split TCP)

Split TCP is a technique to reduce latencies and TCP problems by breaking a connection that would incur a high round-trip time into smaller pieces. By placing the Front Door environments closer to end users and terminating TCP connections inside the Front Door environment, one TCP connection with a large round-trip time (RTT) to application backend is split into two TCP connections. The short connection between the end user and the Front Door environment means the connection gets established over three short round trips instead of three long round trips, saving latency. The long connection between the Front Door environment and the backend can be pre-established and reused across multiple end-user calls, again saving the TCP connection time. The effect is multiplied when establishing a SSL/TLS (Transport Layer Security) connection as there are more round trips to secure the connection.

Processing request to match a routing rule

After establishing a connection and doing an SSL handshake, when a request lands on a Front Door environment, matching a routing rule is the first step. This match basically is determining from all the configurations in Front Door, which particular routing rule to match the request to. Read about how Front Door does route matching to learn more.

Identifying available backends in the backend pool for the routing rule

Once Front Door has a match for a routing rule based on the incoming request and if there is no caching, then the next step is to pull the health probe status for the backend pool associated with the matched route. Read about how Front Door monitors backend health using Health Probes to learn more.

Forwarding the request to your application backend

Finally, assuming there is no caching configured, the user request is forwarded to the "best" backend based on your Front Door routing method configuration.

Next steps

- Learn how to create a Front Door.

Front Door routing methods

2/6/2019 • 7 minutes to read • Edit Online

Azure Front Door Service supports various traffic-routing methods to determine how to route your HTTP/HTTPS traffic to the various service endpoints. With each of your client requests reaching Front Door, the configured routing method gets applied to ensure the requests are forwarded to the best backend instance.

There are four main concepts to traffic routing available in Front Door:

- **Latency:** The latency-based routing ensures that requests are sent to the lowest latency backends acceptable within a sensitivity range. Basically, your user requests are sent to the "closest" set of backends with respect to network latency.
- **Priority:** You can assign priorities to your different backends when you want to use a primary service backend for all traffic, and provide backups in case the primary or the backup backends are unavailable.
- **Weighted:** You can assign weights to your different backends when you want to distribute traffic across a set of backends, either evenly or according to weight coefficients.
- **Session Affinity:** You can configure session affinity for your frontend hosts or domains when you want that subsequent requests from a user are sent to the same backend as long as the user session is still active and the backend instance still reports healthy based on health probes.

All Front Door configurations include monitoring of backend health and automated instant global failover. For more information, see Front Door Backend Monitoring. Your Front Door can be configured to either work based off a single routing method and depending on your application needs you can use multiple or all of these routing methods in combination to build an optimal routing topology.

Lowest latencies based traffic-routing

Deploying backends in two or more locations across the globe can improve the responsiveness of many applications by routing traffic to the location that is 'closest' to your end users. The default traffic-routing method for your Front Door configuration forwards requests from your end users to the closest backend from the Front Door environment that received the request. Combined with the Anycast architecture of Azure Front Door Service, this approach ensures that each of your end users get maximum performance personalized based on their location.

The 'closest' backend is not necessarily closest as measured by geographic distance. Instead, Front Door determines the closest backends by measuring network latency. Read more about Front Door's routing architecture.

Below is the overall decision flow:

AVAILABLE BACKENDS	PRIORITY	LATENCY SIGNAL (BASED ON HEALTH PROBE)	WEIGHTS

AVAILABLE BACKENDS	PRIORITY	LATENCY SIGNAL (BASED ON HEALTH PROBE)	WEIGHTS
Firstly, select all backends that are enabled and returned healthy (200 OK) for the health probe. Say, there are six backends A, B, C, D, E, and F, and among them C is unhealthy and E is disabled. So, list of available backends is A, B, D, and F.	Next, the top priority backends amongst the available ones are selected. Say, backend A, B, and D have priority 1 and backend F has a priority of 2. So, selected backends will be A, B, and D.	Select the backends with latency range (least latency & latency sensitivity in ms specified). Say, if A is 15 ms, B is 30 ms and D is 60 ms away from the Front Door environment where the request landed, and latency sensitivity is 30 ms, then lowest latency pool comprises of backend A and B, because D is beyond 30 ms away from the closest backend that is A.	Lastly, Front Door will round robin the traffic among the final selected pool of backends in the ratio of weights specified. Say, if backend A has a weight of 5 and backend B has a weight of 8, then the traffic will be distributed in the ratio of 5:8 among backends A and B.

> **NOTE**
>
> By default, the latency sensitivity property is set to 0 ms, that is, always forward the request to the fastest available backend.

Priority-based traffic-routing

Often an organization wants to provide reliability for its services by deploying one or more backup services in case their primary service goes down. Across the industry, this topology is also referred to as Active/Standby or Active/Passive deployment topology. The 'Priority' traffic-routing method allows Azure customers to easily implement this failover pattern.

Your default Front Door contains an equal priority list of backends. By default, Front Door sends traffic only to the top priority backends (lowest value for priority) that is, the primary set of backends. If the primary backends are not available, Front Door routes the traffic to the secondary set of backends (second lowest value for priority). If both the primary and secondary backends are not available, the traffic goes to the third, and so on. Availability of the backend is based on the configured status (enabled or disabled) and the ongoing backend health status as determined by the health probes.

Configuring priority for backends

Each of the backends in your backend pool within the Front Door configuration has a property called 'Priority', which can be a number between 1 and 5. With Azure Front Door Service, you configure the backend priority explicitly using this property for each backend. This property is a value between 1 and 5. Lower values represent a higher priority. Backends can share priority values.

Weighted traffic-routing method

The 'Weighted' traffic-routing method allows you to distribute traffic evenly or to use a pre-defined weighting.

In the Weighted traffic-routing method, you assign a weight to each backend in the Front Door configuration of your backend pool. The weight is an integer from 1 to 1000. This parameter uses a default weight of '50'.

Amongst the list of available backends within the accepted latency sensitivity (as specified), the traffic gets distributed in a round-robin mechanism in the ratio of weights specified. If the latency sensitivity is set to 0 milliseconds, then this property doesn't take effect unless there are two backends with the same network latency.

The weighted method enables some useful scenarios:

- **Gradual application upgrade**: Allocate a percentage of traffic to route to a new backend, and gradually increase the traffic over time to bring it at par with other backends.

- **Application migration to Azure**: Create a backend pool with both Azure and external backends. Adjust the weight of the backends to prefer the new backends. You can gradually set this up starting with having the new backends disabled, then assigning them the lowest weights, slowly increasing it to levels where they take most traffic. Then finally disabling the less preferred backends and removing them from the pool.
- **Cloud-bursting for additional capacity**: Quickly expand an on-premises deployment into the cloud by putting it behind Front Door. When you need extra capacity in the cloud, you can add or enable more backends and specify what portion of traffic goes to each backend.

Session Affinity

By default, without session affinity, Front Door forwards requests originating from the same client to different backends based on load balancing configuration particularly as the latencies to different backends change or if different requests from the same user lands on a different Front Door environment. However, some stateful applications or in certain other scenarios, prefer that subsequent requests from the same user land on the same backend that processed the initial request. The cookie-based session affinity feature is useful when you want to keep a user session on the same backend. By using Front Door managed cookies, Azure Front Door Service can direct subsequent traffic from a user session to the same backend for processing as long as the backend is healthy and the user session hasn't expired.

Session affinity can be enabled at a frontend host level that is for each of your configured domains (or subdomains). Once enabled, Front Door adds a cookie to the user's session. Cookie-based session affinity allows Front Door to identify different users even if behind the same IP address, which in turn allows a more even distribution of traffic between your different backends.

The lifetime of the cookie is the same as the user's session, as Front Door currently only supports session cookie.

> **NOTE**
>
> Public proxies may interfere with session affinity. This is because establishing a session requires Front Door to add a session affinity cookie to the response, which cannot be done if the response is cacheable as it would disrupt the cookies of other clients requesting the same resource. To protect against this, session affinity will **not** be established if the backend sends a cacheable response when this is attempted. If the session has already been established, it does not matter if the response from the backend is cacheable. Session affinity will be established in the following circumstances, **unless** the response has an HTTP 304 status code:
>
> - The response has specific values set for the `Cache-Control` header that prevent caching, such as "private" or no-store".
> - The response contains an `Authorization` header that has not expired.
> - The response has an HTTP 302 status code.

Next steps

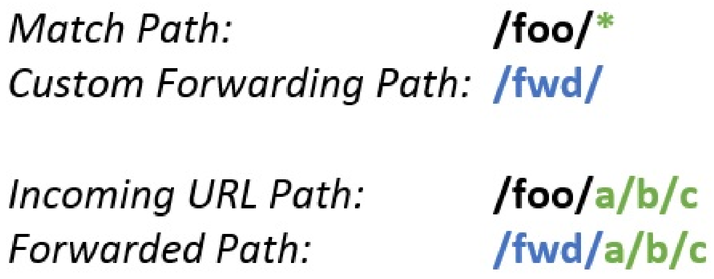
- Learn how to create a Front Door.
- Learn how Front Door works.

URL rewrite (custom forwarding path)

3/14/2019 • 2 minutes to read • Edit Online

Azure Front Door Service supports URL rewrite by allowing you to configure an optional **Custom Forwarding Path** to use when constructing the request to forward to the backend. By default, if no custom forwarding path is provided, then Front Door will copy the incoming URL path to the URL used in the forwarded request. The Host header used in the forwarded request is as configured for the selected backend. Read Backend Host Header to learn what it does and how you can configure it.

The powerful part of URL rewrite using custom forwarding path is that it will copy any part of the incoming path that matches to a wildcard path to the forwarded path (these path segments are the **green** segments in the example below):



Match Path: **/foo/***
Custom Forwarding Path: **/fwd/**

Incoming URL Path: **/foo/a/b/c**
Forwarded Path: **/fwd/a/b/c**

URL rewrite example

Consider a routing rule with the following frontend hosts and paths configured:

HOSTS	PATHS
www.contoso.com	/*
	/foo
	/foo/*
	/foo/bar/*

The first column of the table below shows examples of incoming requests and the second column shows what would be the "most-specific" matching route 'Path'. The third and subsequent columns of the first row of the table are examples of configured **Custom Forwarding Paths**, with the rest of the rows in those columns representing examples of what the forwarded request path would be if it matched to the request in that row.

For example, if we read across the second row, it's saying that for incoming request `www.contoso.com/sub`, if the custom forwarding path was `/`, then the forwarded path would be `/sub`. If the custom forwarding path was `/fwd/`, then the forwarded path would be `/fwd/sub`. And so forth, for the remaining columns. The **emphasized** parts of the paths below represent the portions that are part of the wildcard match.

INCOMING REQUEST	MOST-SPECIFIC MATCH PATH	/	/FWD/	/FOO/	/FOO/BAR/
www.contoso.com/	/*	/	/fwd/	/foo/	/foo/bar/

INCOMING REQUEST	MOST-SPECIFIC MATCH PATH	/	/FWD/	/FOO/	/FOO/BAR/
www.contoso.com/**sub**	/*	**/sub**	/fwd/**sub**	/foo/**sub**	/foo/bar/**sub**
www.contoso.com/**a/b/c**	/*	**/a/b/c**	/fwd/**a/b/c**	/foo/**a/b/c**	/foo/bar/**a/b/c**
www.contoso.com/foo	/foo	/	/fwd/	/foo/	/foo/bar/
www.contoso.com/foo/	/foo/*	/	/fwd/	/foo/	/foo/bar/
www.contoso.com/foo/**bar**	/foo/*	**/bar**	/fwd/**bar**	/foo/**bar**	/foo/bar/**bar**

Optional settings

There are additional optional settings you can also specify for any given routing rule settings:

- **Cache Configuration** - If disabled or not specified, then requests that match to this routing rule will not attempt to use cached content and instead will always fetch from the backend. Read more about Caching with Front Door.

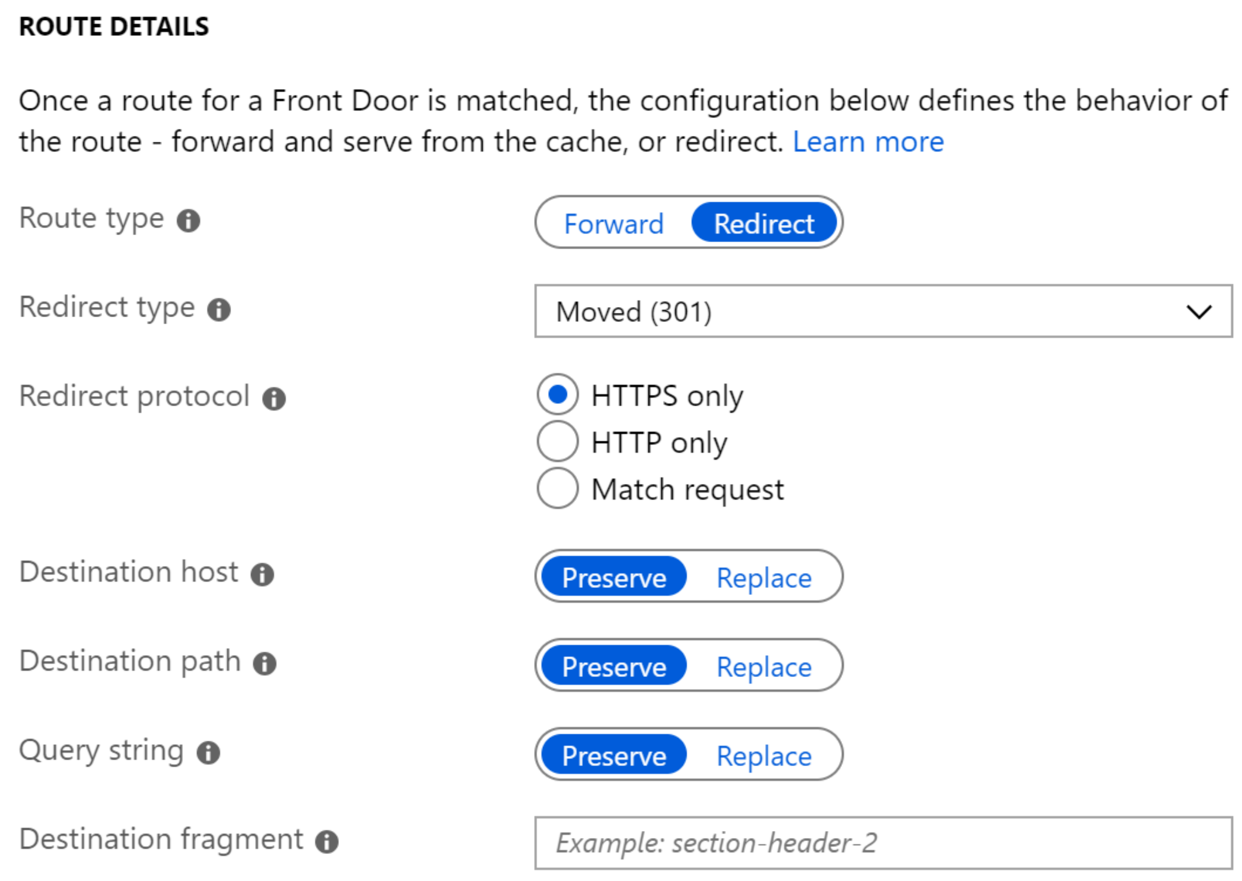
Next steps

- Learn how to create a Front Door.
- Learn how Front Door works.

URL redirect

6/21/2019 • 3 minutes to read • Edit Online

You can use Azure Front Door Service to redirect traffic. You can redirect traffic at multiple levels (protocol, hostname, path, query string) and all of the functionality can be configured for individual microservices as the redirection is path-based. This simplifies application configuration, optimizes the resource usage, and supports new redirection scenarios including global and path-based redirection.



ROUTE DETAILS

Once a route for a Front Door is matched, the configuration below defines the behavior of the route - forward and serve from the cache, or redirect. Learn more

Route type: Forward [✓ Redirect]

Redirect type: Moved (301)

Redirect protocol:
● HTTPS only
○ HTTP only
○ Match request

Destination host: [✓ Preserve] Replace

Destination path: [✓ Preserve] Replace

Query string: [✓ Preserve] Replace

Destination fragment: *Example: section-header-2*

Redirection types

A redirect type sets the response status code for the clients to understand the purpose of the redirect. The following types of redirection are supported:

- **301 (Moved permanently)**: Indicates that the target resource has been assigned a new permanent URI and any future references to this resource ought to use one of the enclosed URIs. Use 301 status code for HTTP to HTTPS redirection.
- **302 (Found)**: Indicates that the target resource resides temporarily under a different URI. Since the redirection might be altered on occasion, the client ought to continue to use the effective request URI for future requests.
- **307 (Temporary redirect)**: Indicates that the target resource resides temporarily under a different URI and the user agent MUST NOT change the request method if it performs an automatic redirection to that URI. Since the redirection can change over time, the client ought to continue using the original effective request URI for future requests.
- **308 (Permanent redirect)**: Indicates that the target resource has been assigned a new permanent URI and any future references to this resource ought to use one of the enclosed URIs. Clients with link editing capabilities, ought to automatically relink references to the effective request URI to one or more of the new

references sent by the server, where possible.

Redirection protocol

You can set the protocol that will be used for redirection. This allows for one of the most common use cases of redirect feature, that is to set HTTP to HTTPS redirection.

- **HTTPS only**: Set the protocol to HTTPS only, if you are looking to redirect the traffic from HTTP to HTTPS. Azure Front Door Service recommends that you should always set the redirection to HTTPS only.
- **HTTP only**: This redirects the incoming request to HTTP. Use this value only if you want to keep your traffic HTTP that is, non-encrypted.
- **Match request**: This option retains the protocol used by the incoming request. So, an HTTP request remains HTTP and an HTTPS request remains HTTPS post redirection.

Destination host

As part of configuring a redirect routing, you can also change the hostname or domain for the redirect request. You can set this field to change the hostname in the URL for the redirection or otherwise preserve the hostname from the incoming request. So, using this field you can redirect all requests sent on https://www.contoso.com/* to https://www.fabrikam.com/*.

Destination path

For cases where you want to replace the path segment of a URL as part of redirection, you can set this field with the new path value. Otherwise, you can choose to preserve the path value as part of redirect. So, using this field, you can redirect all requests sent to https://www.contoso.com/* to https://www.contoso.com/redirected-site.

Query string parameters

You can also replace the query string parameters in the redirected URL. In order to replace any existing query string from the incoming request URL, set this field to 'Replace' and then set the appropriate value. Otherwise, you can retain the original set of query strings by setting the field to 'Preserve'. As an example, using this field, you can redirect all traffic sent to https://www.contoso.com/foo/bar to https://www.contoso.com/foo/bar?&utm_referrer=https%3A%2F%2Fwww.bing.com%2F.

Destination fragment

The destination fragment is the portion of URL after '#', normally used by browsers to land on a specific section on a page. You can set this field to add a fragment to the redirect URL.

Next steps

- Learn how to create a Front Door.
- Learn how Front Door works.

Create a Front Door with HTTP to HTTPS redirection using the Azure portal

7/5/2019 • 4 minutes to read • Edit Online

You can use the Azure portal to create a Front Door with a certificate for SSL termination. A routing rule is used to redirect HTTP traffic to HTTPS.

In this article, you learn how to:

- Create a Front Door with an existing Web App resource
- Add a custom domain with SSL certificate
- Setup HTTPS redirect on the custom domain

If you don't have an Azure subscription, create a free account before you begin.

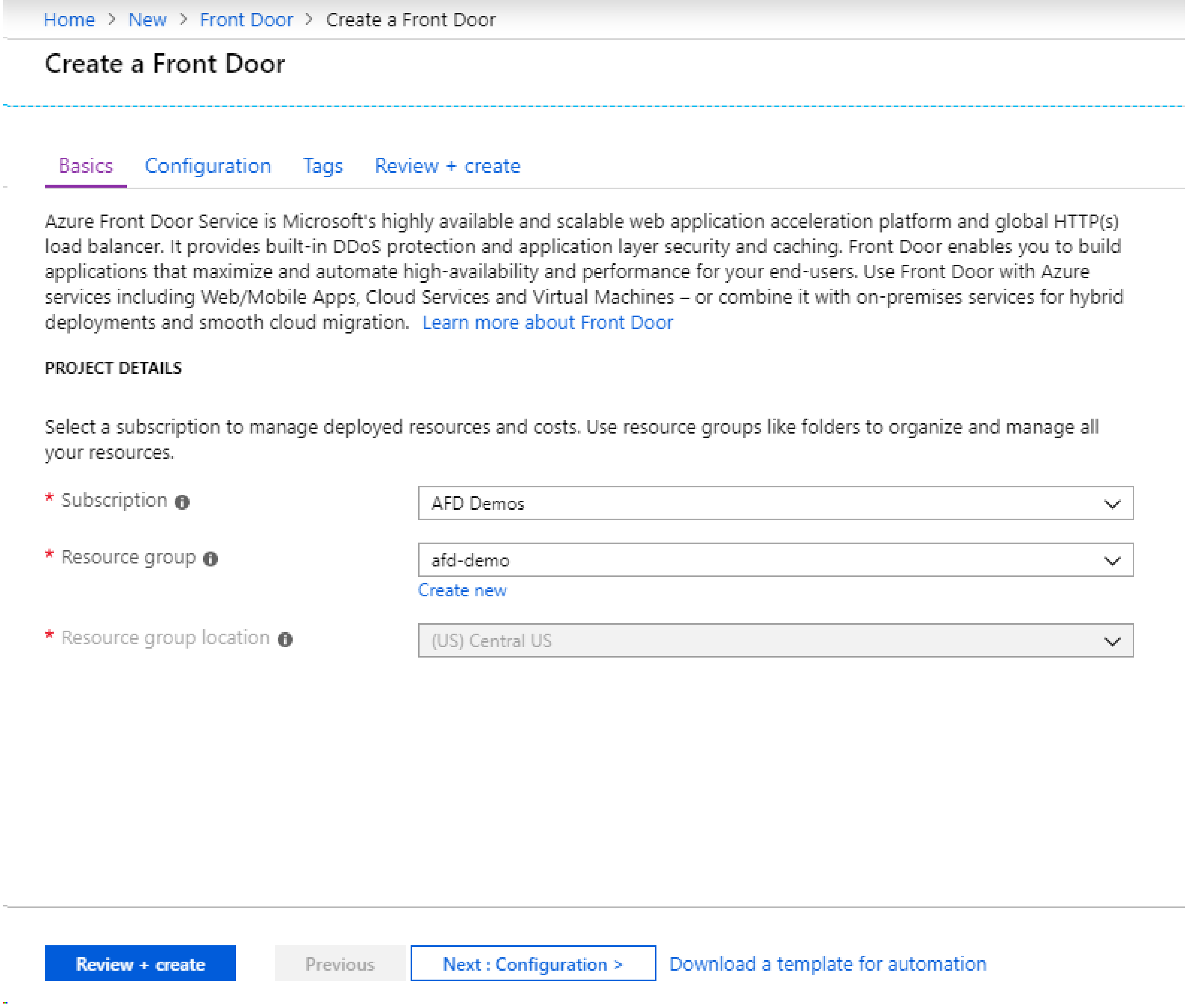
Create a Front Door with an existing Web App resource

1. Sign in to the Azure portal at https://portal.azure.com.

2. Click **Create a resource** found on the upper left-hand corner of the Azure portal.

3. Search for **Front Door** using the search bar and once you find the resource type, click **Create**.

4. Choose a subscription and then either use an existing resource group or create a new one. Note, the location asked in the UI is for the resource group only. Your Front Door configuration will get deployed across all of Azure Front Door's POP locations.



Home > New > Front Door > Create a Front Door

Create a Front Door

Basics | Configuration | Tags | Review + create

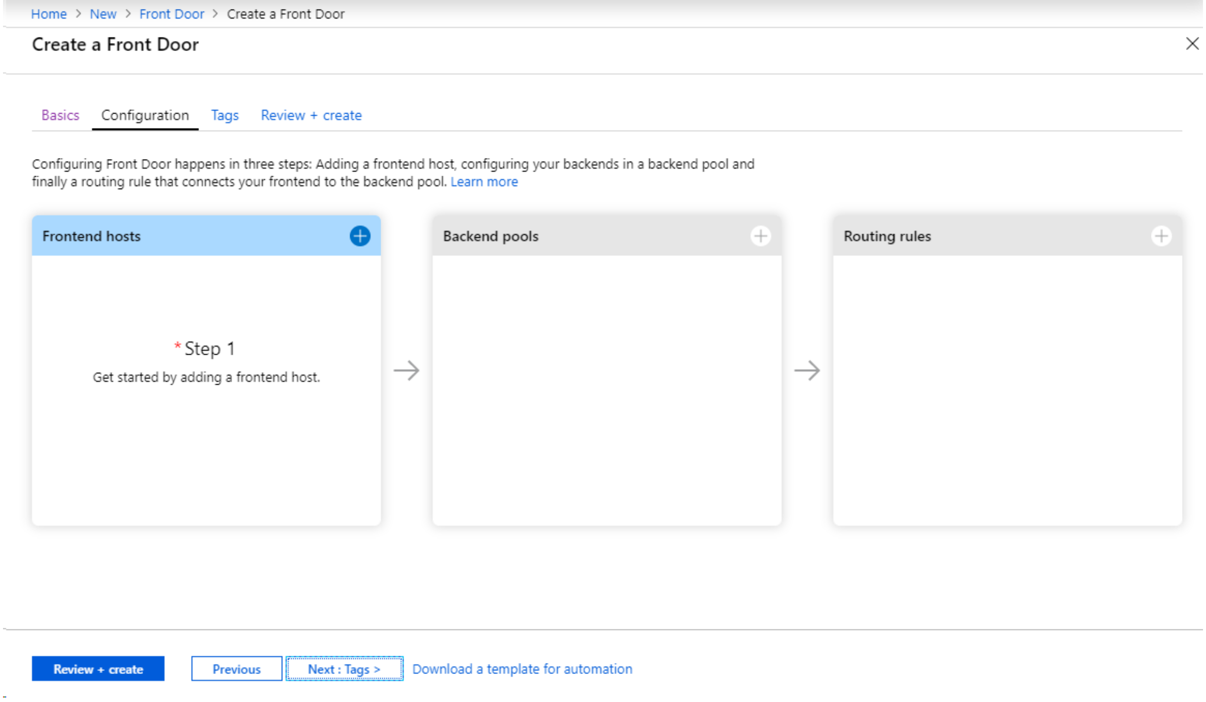
Azure Front Door Service is Microsoft's highly available and scalable web application acceleration platform and global HTTP(s) load balancer. It provides built-in DDoS protection and application layer security and caching. Front Door enables you to build applications that maximize and automate high-availability and performance for your end-users. Use Front Door with Azure services including Web/Mobile Apps, Cloud Services and Virtual Machines – or combine it with on-premises services for hybrid deployments and smooth cloud migration. Learn more about Front Door

PROJECT DETAILS

Select a subscription to manage deployed resources and costs. Use resource groups like folders to organize and manage all your resources.

* Subscription	AFD Demos
* Resource group	afd-demo Create new
* Resource group location	(US) Central US

5. Click **Next** to enter the configuration tab. The configuration for Front Door happens in three steps - adding a default frontend host, adding backends in a backend pool and then creating routing rules to map the routing behavior for frontend host.



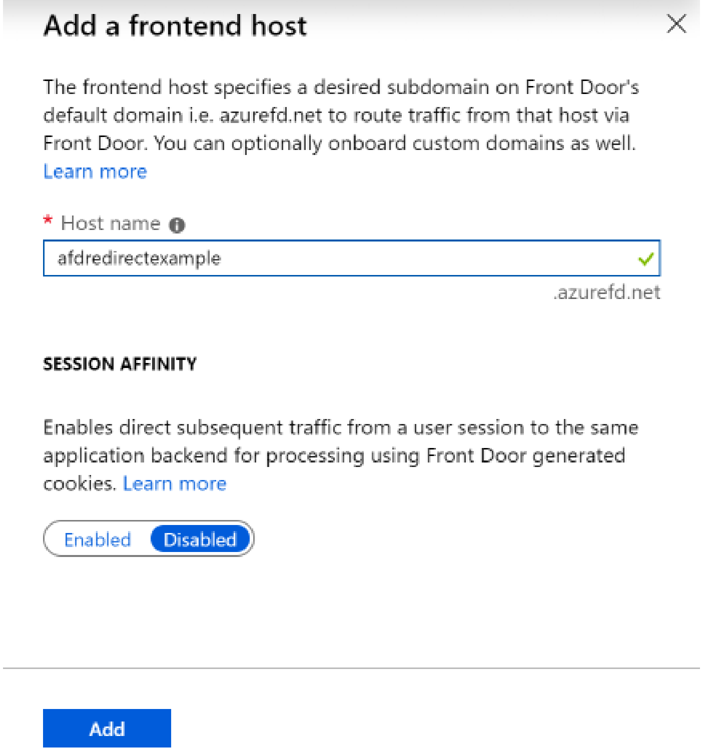
Home > New > Front Door > Create a Front Door

Create a Front Door

Basics | Configuration | Tags | Review + create

Configuring Front Door happens in three steps: Adding a frontend host, configuring your backends in a backend pool and finally a routing rule that connects your frontend to the backend pool. Learn more

6. Click the '**+**' icon on the *Frontend hosts* to create a frontend host, enter a globally unique name for your default frontend host for your Front Door (`\<**name**\>.azurefd.net`). Click **Add** to proceed to the next step.



Add a frontend host

The frontend host specifies a desired subdomain on Front Door's default domain i.e. azurefd.net to route traffic from that host via Front Door. You can optionally onboard custom domains as well. Learn more

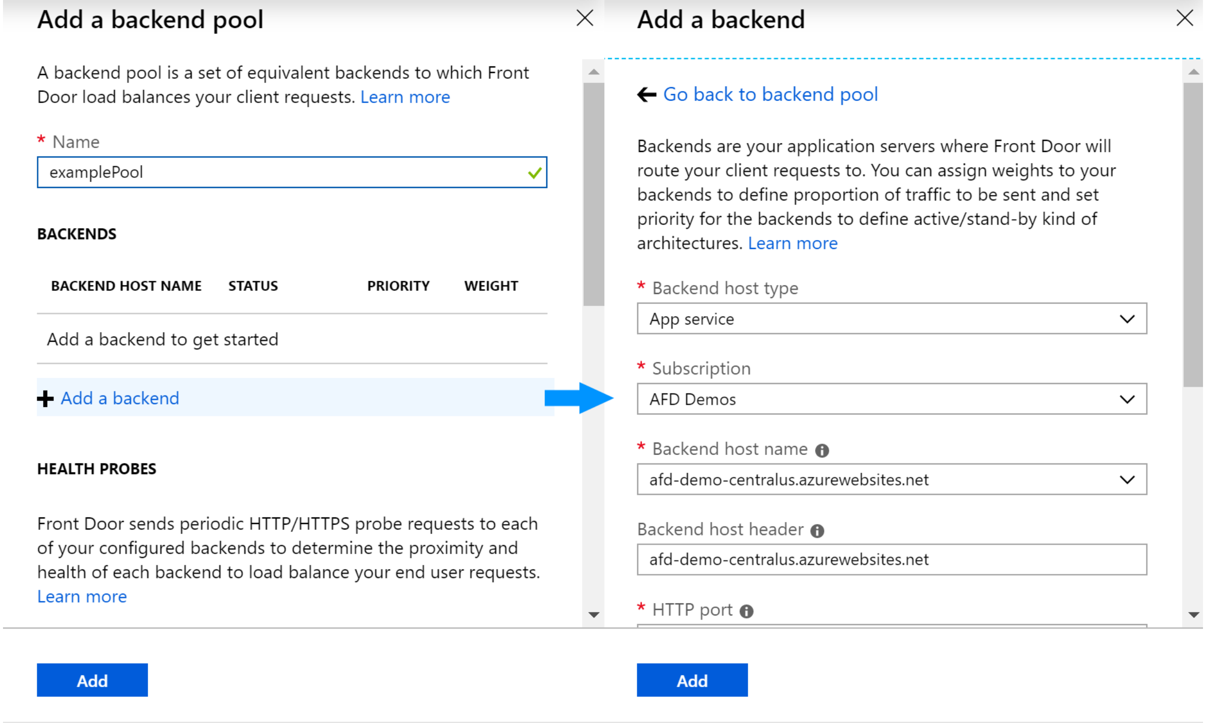
* Host name: afdredirectexample .azurefd.net

SESSION AFFINITY

Enables direct subsequent traffic from a user session to the same application backend for processing using Front Door generated cookies. Learn more

Enabled [✓ Disabled]

7. Click the '**+**' icon on the *Backend pools* to create a backend pool. Provide a name for the backend pool and then click '**Add a backend**'.
8. Select the Backend Host Type as *App service*. Select the subscription where your web app is hosted and then select the specific web app from the dropdown for **Backend host name**.
9. Click **Add** to save the backend and click **Add** again to save the backend pool config.



Add a backend pool

A backend pool is a set of equivalent backends to which Front Door load balances your client requests. Learn more

* Name: examplePool

BACKENDS

BACKEND HOST NAME	STATUS	PRIORITY	WEIGHT
Add a backend to get started			

HEALTH PROBES

Front Door sends periodic HTTP/HTTPS probe requests to each of your configured backends to determine the proximity and health of each backend to load balance your end user requests. Learn more

Add a backend

Backends are your application servers where Front Door will route your client requests to. You can assign weights to your backends to define proportion of traffic to be sent and set priority for the backends to define active/stand-by kind of architectures. Learn more

* Backend host type: App service

* Subscription: AFD Demos

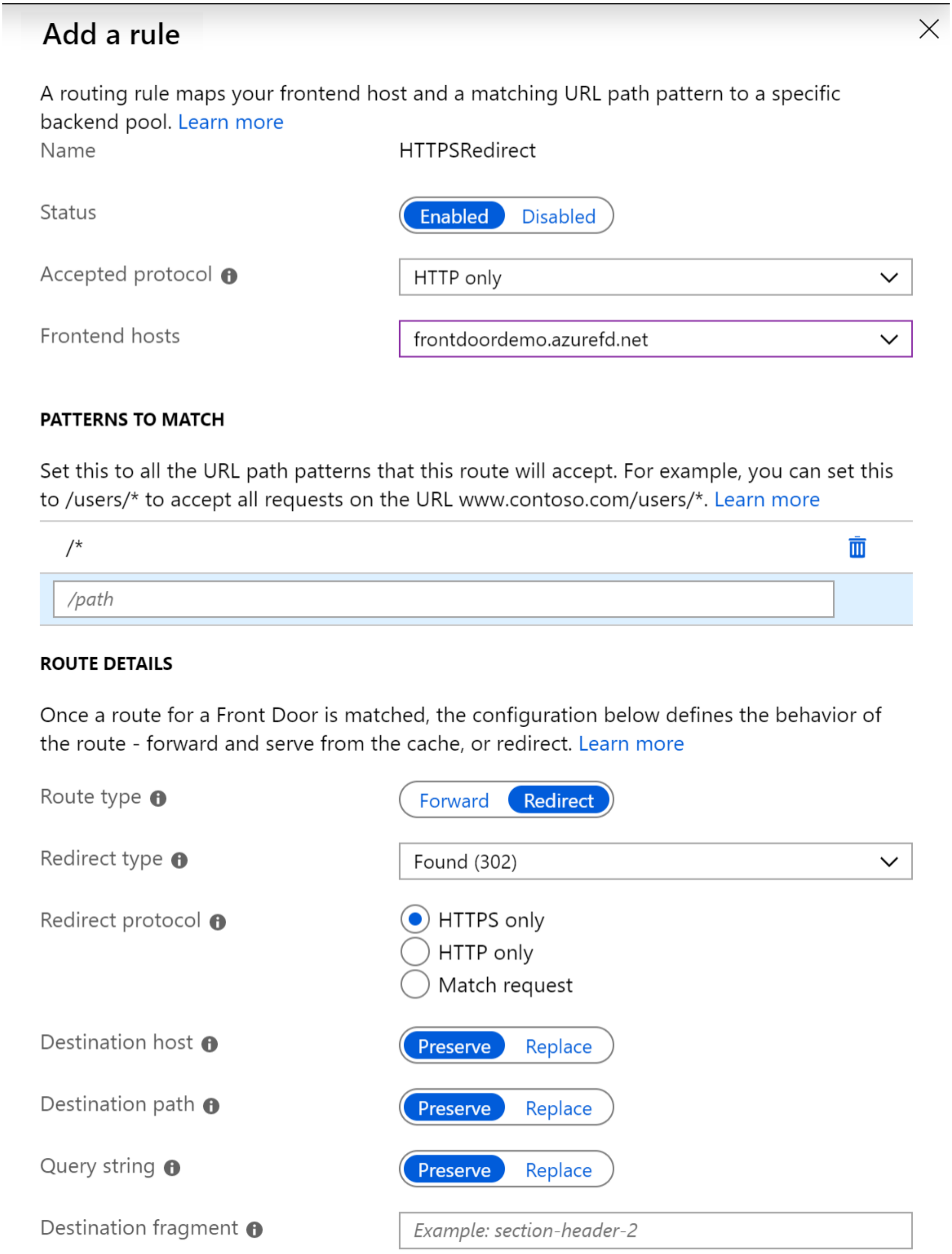
* Backend host name: afd-demo-centralus.azurewebsites.net

Backend host header: afd-demo-centralus.azurewebsites.net

* HTTP port

10. Click the '**+**' icon on the *Routing rules* to create a route. Provide a name for the route, say 'HttpToHttpsRedirect', and then set the *Accepted Protocols* field to **'HTTP only'**. Ensure that the appropriate *frontend host* is selected.
11. On the *Route Details* section, set the *Route Type* to **Redirect**, ensure that the *Redirect type* is set to **Found (302)** and *Redirect protocol* is set to **HTTPS only**.

12. Click Add to save the routing rule for HTTP to HTTPS redirect.



Add a rule

A routing rule maps your frontend host and a matching URL path pattern to a specific backend pool. Learn more

Name: HTTPSRedirect

Status: ● Enabled ○ Disabled

Accepted protocol: HTTP only

Frontend hosts: frontdoordemo.azurefd.net

PATTERNS TO MATCH

Set this to all the URL path patterns that this route will accept. For example, you can set this to /users/* to accept all requests on the URL www.contoso.com/users/*. Learn more

/*

/path

ROUTE DETAILS

Once a route for a Front Door is matched, the configuration below defines the behavior of the route - forward and serve from the cache, or redirect. Learn more

Route type: ○ Forward ● Redirect

Redirect type: Found (302)

Redirect protocol:
● HTTPS only
○ HTTP only
○ Match request

Destination host: ● Preserve ○ Replace

Destination path: ● Preserve ○ Replace

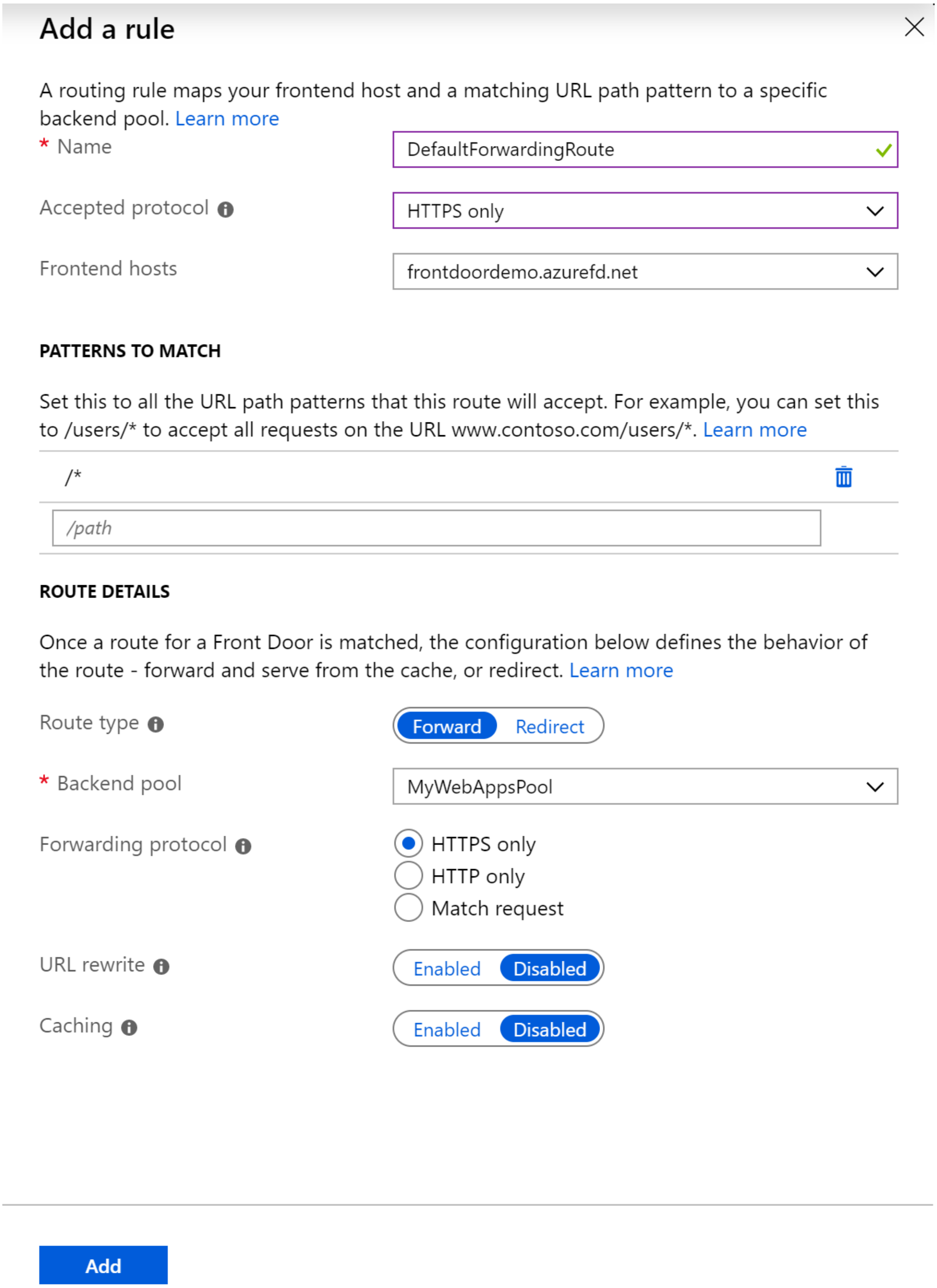
Query string: ● Preserve ○ Replace

Destination fragment: Example: section-header-2

13. Add another routing rule for handling the HTTPS traffic. Click the '**+**' sign on the *Routing rules* and provide a name for the route, say 'DefaultForwardingRoute', and then set the *Accepted Protocols* field to **'HTTPS only'**. Ensure that the appropriate *frontend host* is selected.

14. On the Route Details section, set the *Route Type* to **Forward**, ensure that the right backend pool is selected and the *Forwarding Protocol* is set to **HTTPS only**.

15. Click Add to save the routing rule for request forwarding.



Add a rule

A routing rule maps your frontend host and a matching URL path pattern to a specific backend pool. Learn more

* Name: DefaultForwardingRoute ✓

Accepted protocol: HTTPS only

Frontend hosts: frontdoordemo.azurefd.net

PATTERNS TO MATCH

Set this to all the URL path patterns that this route will accept. For example, you can set this to /users/* to accept all requests on the URL www.contoso.com/users/*. Learn more

/*

/path

ROUTE DETAILS

Once a route for a Front Door is matched, the configuration below defines the behavior of the route - forward and serve from the cache, or redirect. Learn more

Route type: [✓ Forward] Redirect

* Backend pool: MyWebAppsPool

Forwarding protocol:
● HTTPS only
○ HTTP only
○ Match request

URL rewrite: Enabled [✓ Disabled]

Caching: Enabled [✓ Disabled]

16. Click **Review + create** and then **Create**, to create your Front Door profile. Go to the resource once created.

Add a custom domain to your Front Door and enable HTTPS on it

The following steps showcase how you can add a custom domain on an existing Front Door resource and then enable HTTP to HTTPS redirection on it.

Add a custom domain

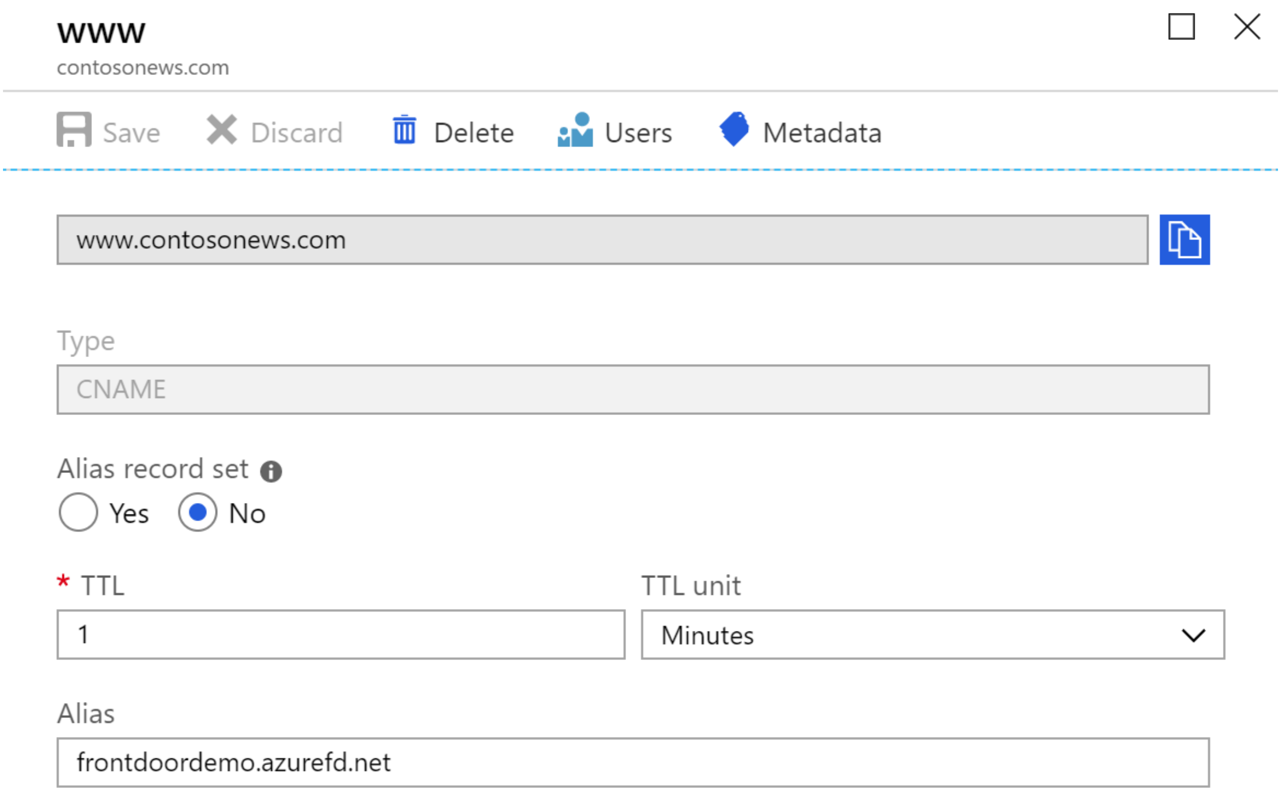
In this example, you add a CNAME record for the `www` subdomain (for example, `www.contosonews.com`).

Create the CNAME record

Add a CNAME record to map a subdomain to your Front Door's default frontend host (`<name>.azurefd.net` , where `<name>` is the name of your Front Door profile).

For the `www.contoso.com` domain, as an example, add a CNAME record that maps the name `www` to `<name>.azurefd.net` .

After you add the CNAME, the DNS records page looks like the following example:



www

contosonews.com

www.contosonews.com

Type

CNAME

Alias record set

○ Yes ● No

* TTL

1

TTL unit

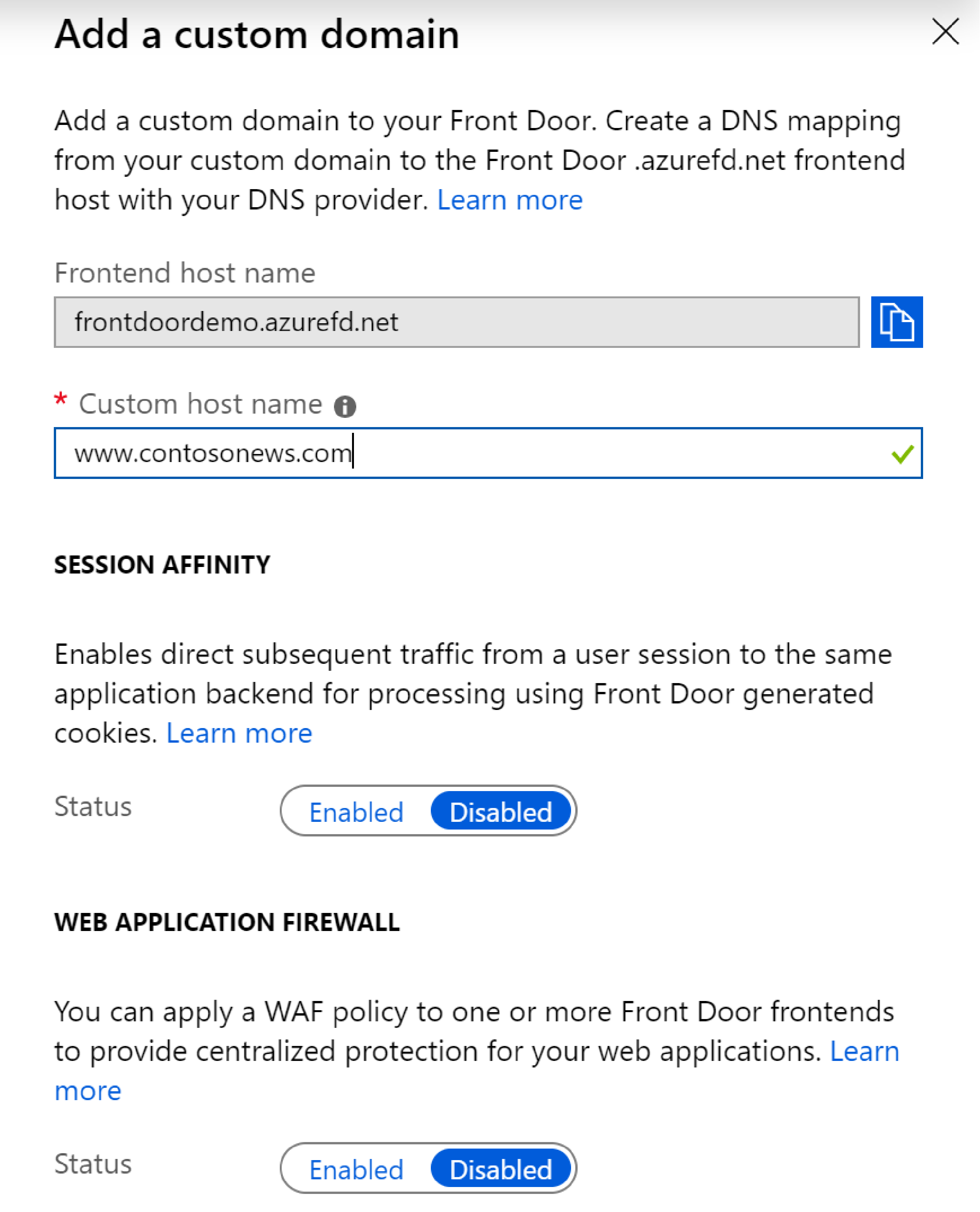
Minutes

Alias

frontdoordemo.azurefd.net

Onboard the custom domain on your Front Door

1. On the Front Door designer tab, click on '+' icon on the Frontend hosts section to add a new custom domain.
2. Enter the fully qualified custom DNS name in the custom host name field, example `www.contosonews.com` .
3. Once the CNAME mapping from the domain to your Front Door is validated, click on **Add** to add the custom domain.
4. Click **Save** to submit the changes.



Add a custom domain

Add a custom domain to your Front Door. Create a DNS mapping from your custom domain to the Front Door .azurefd.net frontend host with your DNS provider. Learn more

Frontend host name

frontdoordemo.azurefd.net

* Custom host name

www.contosonews.com

SESSION AFFINITY

Enables direct subsequent traffic from a user session to the same application backend for processing using Front Door generated cookies. Learn more

Status: Enabled [✓ Disabled]

WEB APPLICATION FIREWALL

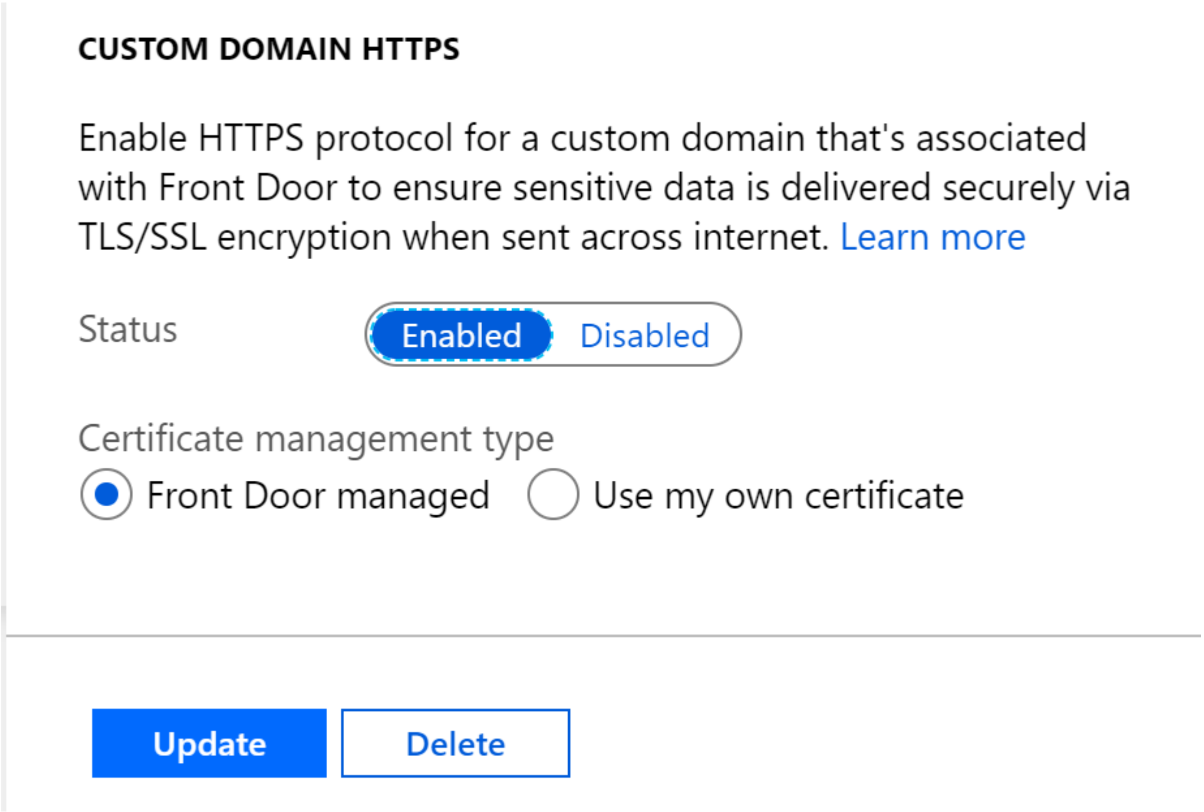
You can apply a WAF policy to one or more Front Door frontends to provide centralized protection for your web applications. Learn more

Status: Enabled [✓ Disabled]

Enable HTTPS on your custom domain

1. Click on the custom domain that was added and under the section **Custom domain HTTPS**, change the status to **Enabled**.

2. You can leave the **Certificate management type** set to *Front Door managed* for the free certificate maintained, managed, and autorotated by Front Door. You can also choose to use your own custom SSL certificate stored with Azure Key Vault. This tutorial assumes that the use of Front Door managed certificate.



CUSTOM DOMAIN HTTPS

Enable HTTPS protocol for a custom domain that's associated with Front Door to ensure sensitive data is delivered securely via TLS/SSL encryption when sent across internet. Learn more

Status [✓ Enabled] Disabled

Certificate management type

● Front Door managed ○ Use my own certificate

3. Click on **Update** to save the selection and then click **Save**.
4. Click **Refresh** after a couple of minutes and then click on the custom domain again to see the progress of certificate provisioning.

> **WARNING**
> Enabling HTTPS for a custom domain may take several minutes and also depends on domain ownership validation if the CNAME is not directly mapped to your Front Door host `<name>.azurefd.net`. Learn more about how to enable HTTPS for a custom domain.

Configure the routing rules for the custom domain

1. Click on the redirect routing rule created earlier.
2. Click on the dropdown for Frontend hosts and select your custom domain to apply this route for your domain as well.
3. Click **Update**.
4. Do the same operation for the other routing rule as well that is, for your forwarding route to add the custom domain.
5. Click **Save** to submit your changes.

Next steps

- Learn how to create a Front Door.
- Learn how Front Door works.
- Learn more about URL redirect on Front Door.

Onboard a root or apex domain on your Front Door

11/19/2019 • 3 minutes to read • Edit Online

Azure Front Door uses CNAME records to validate domain ownership for onboarding of custom domains. Also, Front Door doesn't expose the frontend IP address associated with your Front Door profile and so you can't map your apex domain to an IP address, if the intent is to onboard it to Azure Front Door.

The DNS protocol prevents the assignment of CNAME records at the zone apex. For example, if your domain is `contoso.com`; you can create CNAME records for `somelabel.contoso.com`; but you can't create CNAME for `contoso.com` itself. This restriction presents a problem for application owners who have load-balanced applications behind Azure Front Door. Since using a Front Door profile requires creation of a CNAME record, it isn't possible to point at the Front Door profile from the zone apex.

This problem is solved using alias records on Azure DNS. Unlike CNAME records, alias records are created at the zone apex and application owners can use it to point their zone apex record to a Front Door profile that has public endpoints. Application owners point to the same Front Door profile that's used for any other domain within their DNS zone. For example, `contoso.com` and `www.contoso.com` can point to the same Front Door profile.

Mapping your apex or root domain to your Front Door profile basically requires CNAME flattening or DNS chasing, which is a mechanism where in the DNS provider recursively resolves the CNAME entry until it hits an IP address. This functionality is supported by Azure DNS for Front Door endpoints.

> **NOTE**
>
> There are other DNS providers as well that support CNAME flattening or DNS chasing, however, Azure Front Door recommends using Azure DNS for its customers for hosting their domains.

You can use the Azure portal to onboard an apex domain on your Front Door and enable HTTPS on it by associating it with a certificate for SSL termination. Apex domains are also referred as root or naked domains.

In this article, you learn how to:

- Create an alias record that points to your Front Door profile
- Add the root domain to the Front Door
- Setup HTTPS on the root domain

> **NOTE**
>
> This tutorial requires that you already have a Front Door profile created. Refer other tutorials like Quickstart: Create a Front Door or Create a Front Door with HTTP to HTTPS redirection to get started.

Create an alias record for zone apex

1. Open **Azure DNS** configuration for the domain to be onboarded.

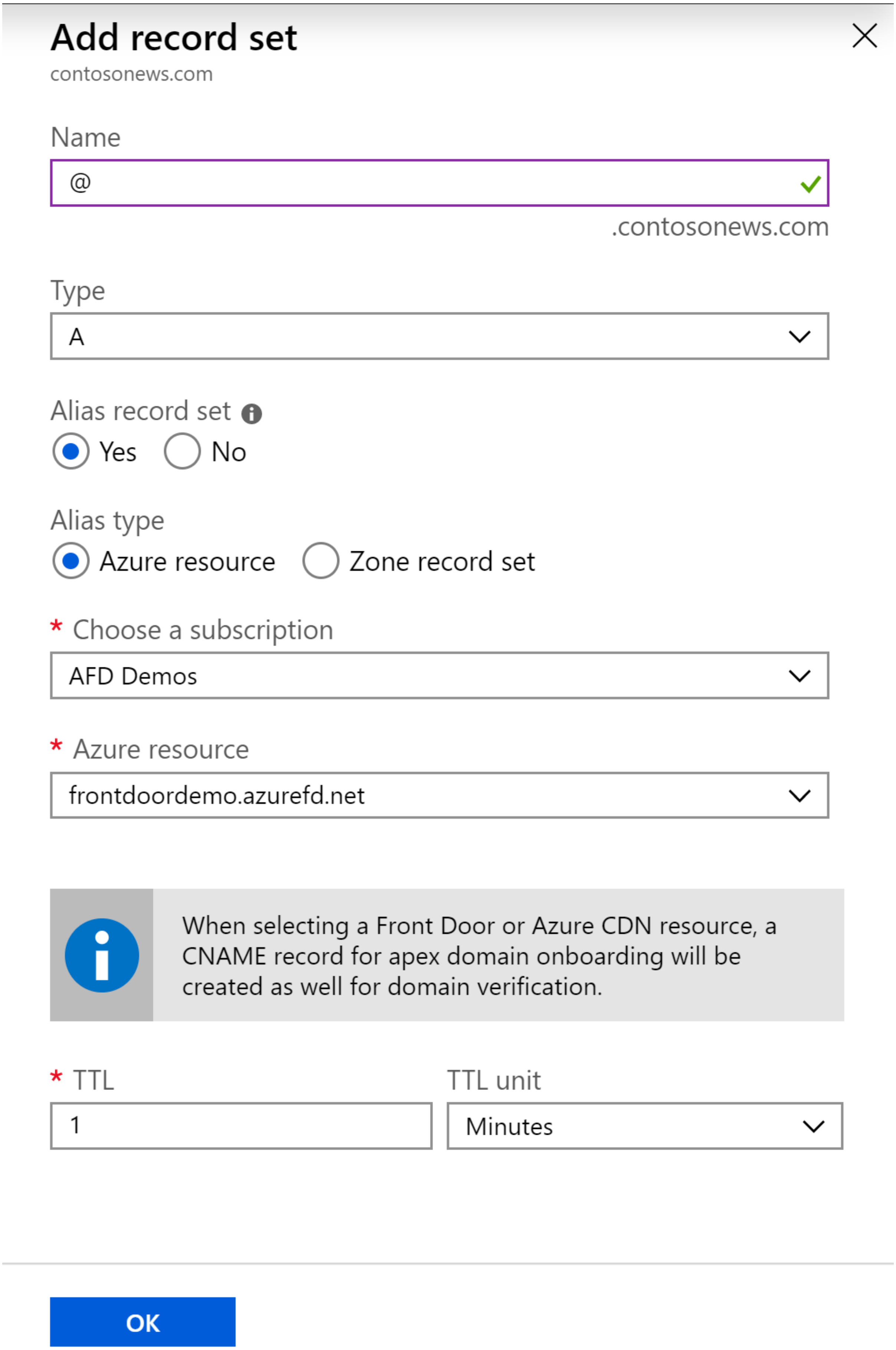
2. Create or edit the record for zone apex.

3. Select the record **type** as *A* record and then select *Yes* for **Alias record set**. **Alias type** should be set to *Azure resource*.

4. Choose the Azure subscription where your Front Door profile is hosted and then select the Front Door

resource from the **Azure resource** dropdown.

5. Click **OK** to submit your changes.



Add record set

contosonews.com

Name

@ ✓

.contosonews.com

Type

A

Alias record set

● Yes ○ No

Alias type

● Azure resource ○ Zone record set

* Choose a subscription

AFD Demos

* Azure resource

frontdoordemo.azurefd.net

When selecting a Front Door or Azure CDN resource, a CNAME record for apex domain onboarding will be created as well for domain verification.

* TTL

1

TTL unit

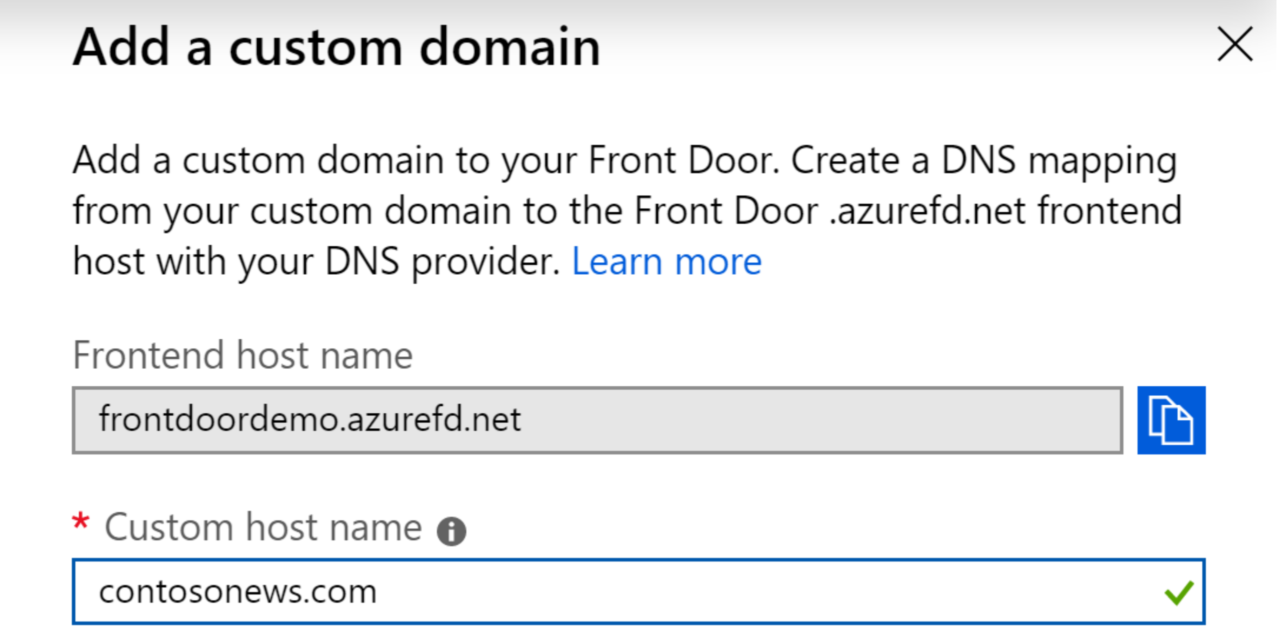
Minutes

6. The above step will create a zone apex record pointing to your Front Door resource and also a CNAME record mapping 'afdverify' (example - `afdverify.contosonews.com`) to `afdverify.<name>.azurefd.net` which

will be used for onboarding the domain on your Front Door profile.

Onboard the custom domain on your Front Door

1. On the Front Door designer tab, click on '+' icon on the Frontend hosts section to add a new custom domain.
2. Enter the root or apex domain name in the custom host name field, example `contosonews.com`.
3. Once the CNAME mapping from the domain to your Front Door is validated, click on **Add** to add the custom domain.
4. Click **Save** to submit the changes.



Add a custom domain

Add a custom domain to your Front Door. Create a DNS mapping from your custom domain to the Front Door .azurefd.net frontend host with your DNS provider. Learn more

Frontend host name

frontdoordemo.azurefd.net

* Custom host name

contosonews.com ✓

Enable HTTPS on your custom domain

1. Click on the custom domain that was added and under the section **Custom domain HTTPS**, change the status to **Enabled**.
2. Select the **Certificate management type** to '*Use my own certificate*'.

> **WARNING**
>
> Front Door managed certificate management type is not currently supported for apex or root domains. The only option available for enabling HTTPS on an apex or root domain for Front Door is using your own custom SSL certificate hosted on Azure Key Vault.

3. Ensure that you have setup the right permissions for Front Door to access your key Vault as noted in the UI, before proceeding to the next step.
4. Choose a **Key Vault account** from your current subscription and then select the appropriate **Secret** and **Secret version** to map to the right certificate.
5. Click on **Update** to save the selection and then click **Save**.
6. Click **Refresh** after a couple of minutes and then click on the custom domain again to see the progress of certificate provisioning.

> **WARNING**
>
> Ensure that you have created appropriate routing rules for your apex domain or added the domain to existing routing rules.

Next steps

- Learn how to create a Front Door.
- Learn how Front Door works.

Troubleshooting common routing issues

11/19/2019 • 3 minutes to read • Edit Online

This article describes how to troubleshoot some of the common routing issues you may face for your Azure Front Door Service configuration.

Hostname Not Routing to Backend and Returns 400 Status Code

Symptom

- You have created a Front Door but a request to the Frontend host is returning an HTTP 400 status code.
  - You have created a DNS mapping from a custom domain to the frontend host you have configured. However, sending a request to the custom domain hostname returns an HTTP 400 status code and does not appear to route to the backend(s) you have configured.

Cause

- This symptom can happen if you have not configured a routing rule for the custom domain that you added as a frontend host. A routing rule needs to be explicitly added for that frontend host, even if one has already been configured for the frontend host under the Front Door subdomain (*.azurefd.net) that your custom domain has a DNS mapping to.

Troubleshooting Steps

- Add a routing rule from the custom domain to the desired backend pool.

Request to Frontend hostname Returns 404 Status Code

Symptom

- You have created a Front Door and configured a frontend host, a backend pool with at least one backend in it, and a routing rule that connects the frontend host to the backend pool. Your content does not seem to be available when sending a request to the configured frontend host because an HTTP 404 status code is returned.

Cause

There are several possible causes for this symptom:

- The backend is not a public facing backend and is not visible to the Front Door service.
- The backend is misconfigured, which is causing the Front Door service to send the wrong request (that is, your backend only accepts HTTP but you have not unchecked allowing HTTPS so Front Door is attempting to forward HTTPS requests).
- The backend is rejecting the host header that was forwarded with the request to the backend.
- The configuration for the backend has not yet been fully deployed.

Troubleshooting Steps

1. Deployment Time
   - Ensure that you have waited ~10 minutes for the configuration to be deployed.
2. Check the Backend Settings
   - Navigate to the backend pool that the request should be routing to (depends on how you have the routing rule configured) and verify that the *backend host type* and backend host name are correct. If

  the backend is a custom host, ensure that you have spelled it correctly.

  - Check your HTTP and HTTPS ports. In most cases, 80 and 443 (respectively), are correct and no changes will be required. However, there is a chance that your backend is not configured this way and is listening on a different port.

    - Check the *Backend host header* configured for the backends that the Frontend host should be routing to. In most cases, this header should be the same as the *Backend host name*. However, an incorrect value can cause various HTTP 4xx status codes if the backend expects something different. If you input the IP address of your backend, you might need to set the *Backend host header* to the hostname of the backend.

3. Check the Routing Rule Settings

  - Navigate to the routing rule that should route from the Frontend hostname in question to a backend pool. Ensure that the accepted protocols are correctly configured, or if not, ensure that the protocol Front Door will use when forwarding the request is correctly configured. The *accepted protocols* determines which requests Front Door should accept and the *Forwarding protocol* determines what protocol Front Door should use to forward the request to the backend.

    - As an example, if the backend only accepts HTTP requests the following configurations would be valid:

      - *Accepted protocols* are HTTP and HTTPS. *Forwarding protocol* is HTTP. Match request will not work, since HTTPS is an allowed protocol and if a request came in as HTTPS, Front Door would try to forward it using HTTPS.

      - *Accepted protocols* are HTTP. *Forwarding protocol* is either match request or HTTPS.

  - *Url Rewrite* is disabled by default and you should only use this field if you want to narrow the scope of backend-hosted resources that you want to make available. When disabled, Front Door will forward the same request path it receives. It is possible that this field is misconfigured and Front Door is requesting a resource from the backend that is not available, thus returning an HTTP 404 status code.

Allowed certificate authorities for enabling custom HTTPS on Azure Front Door Service

11/19/2019 • 2 minutes to read • Edit Online

For an Azure Front Door Service custom domain, when you enable the HTTPS feature by using your own certificate, you must use an allowed certificate authority (CA) to create your SSL certificate. Otherwise, if you use a non-allowed CA or a self-signed certificate, your request will be rejected.

The following CAs are allowed when you create your own certificate:

- AddTrust External CA Root
- AlphaSSL Root CA
- AME Infra CA 01
- AME Infra CA 02
- Ameroot
- APCA-DM3P
- Autopilot Root CA
- Baltimore CyberTrust Root
- Class 3 Public Primary Certification Authority
- COMODO RSA Certification Authority
- COMODO RSA Domain Validation Secure Server CA
- D-TRUST Root Class 3 CA 2 2009
- DigiCert Cloud Services CA-1
- DigiCert Global Root CA
- DigiCert High Assurance CA-3
- DigiCert High Assurance EV Root CA
- DigiCert SHA2 Extended Validation Server CA
- DigiCert SHA2 High Assurance Server CA
- DigiCert SHA2 Secure Server CA
- DST Root CA X3
- D-trust Root Class 3 CA 2 2009
- Encryption Everywhere DV TLS CA
- Entrust Root Certification Authority
- Entrust Root Certification Authority - G2
- Entrust.net Certification Authority (2048)
- GeoTrust Global CA
- GeoTrust Primary Certification Authority
- GeoTrust Primary Certification Authority - G2
- Geotrust RSA CA 2018
- GlobalSign
- GlobalSign Extended Validation CA - SHA256 - G2
- GlobalSign Organization Validation CA - G2
- GlobalSign Root CA
- Go Daddy Root Certificate Authority - G2

- Go Daddy Secure Certificate Authority - G2
- QuoVadis Root CA2 G3
- RapidSSL RSA CA 2018
- Symantec Class 3 EV SSL CA - G3
- Symantec Class 3 Secure Server CA - G4
- Symantec Enterprise Mobile Root for Microsoft
- Thawte Primary Root CA
- Thawte Primary Root CA - G2
- Thawte Primary Root CA - G3
- Thawte RSA CA 2018
- Thawte Timestamping CA
- TrustAsia TLS RSA CA
- VeriSign Class 3 Extended Validation SSL CA
- VeriSign Class 3 Extended Validation SSL SGC CA
- VeriSign Class 3 Public Primary Certification Authority - G5
- VeriSign International Server CA - Class 3
- VeriSign Time Stamping Service Root
- VeriSign Universal Root Certification Authority

Frequently asked questions for Azure Front Door Service

1/12/2020 • 10 minutes to read • Edit Online

This article answers common questions about Azure Front Door Service features and functionality. If you don't see the answer to your question, you can contact us through the following channels (in escalating order):

1. The comments section of this article.
2. Azure Front Door Service UserVoice.
3. **Microsoft Support:** To create a new support request, in the Azure portal, on the **Help** tab, select the **Help + support** button, and then select **New support request**.

General

What is Azure Front Door Service?

Azure Front Door Service is an Application Delivery Network (ADN) as a service, offering various layer 7 load-balancing capabilities for your applications. It provides dynamic site acceleration (DSA) along with global load balancing with near real-time failover. It is a highly available and scalable service, which is fully managed by Azure.

What features does Azure Front Door Service support?

Azure Front Door Service supports dynamic site acceleration (DSA), SSL offloading and end to end SSL, Web Application Firewall, cookie-based session affinity, url path-based routing, free certificates and multiple domain management, and others. For a full list of supported features, see Overview of Azure Front Door Service.

What is the difference between Azure Front Door Service and Azure Application Gateway?

While both Front Door and Application Gateway are layer 7 (HTTP/HTTPS) load balancers, the primary difference is that Front Door is a global service whereas Application Gateway is a regional service. While Front Door can load balance between your different scale units/clusters/stamp units across regions, Application Gateway allows you to load balance between your VMs/containers etc. that is within the scale unit.

When should we deploy an Application Gateway behind Front Door?

The key scenarios why one should use Application Gateway behind Front Door are:

- Front Door can perform path-based load balancing only at the global level but if one wants to load balance traffic even further within their virtual network (VNET) then they should use Application Gateway.
- Since Front Door doesn't work at a VM/container level, so it cannot do Connection Draining. However, Application Gateway allows you to do Connection Draining.
- With an Application Gateway behind AFD, one can achieve 100% SSL offload and route only HTTP requests within their virtual network (VNET).
- Front Door and Application Gateway both support session affinity. While Front Door can direct subsequent traffic from a user session to the same cluster or backend in a given region, Application Gateway can direct affinitize the traffic to the same server within the cluster.

Can we deploy Azure Load Balancer behind Front Door?

Azure Front Door Service needs a public VIP or a publicly available DNS name to route the traffic to. Deploying an Azure Load Balancer behind Front Door is a common use case.

What protocols does Azure Front Door Service support?

Azure Front Door Service supports HTTP, HTTPS and HTTP/2.

How does Azure Front Door Service support HTTP/2?

HTTP/2 protocol support is available to clients connecting to Azure Front Door Service only. The communication to backends in the backend pool is over HTTP/1.1. HTTP/2 support is enabled by default.

What resources are supported today as part of backend pool?

Backend pools can be composed of Storage, Web App, Kubernetes instances, or any other custom hostname that has public connectivity. Azure Front Door Service requires that the backends are defined either via a public IP or a publicly resolvable DNS hostname. Members of backend pools can be across zones, regions, or even outside of Azure as long as they have public connectivity.

What regions is the service available in?

Azure Front Door Service is a global service and is not tied to any specific Azure region. The only location you need to specify while creating a Front Door is the resource group location, which is basically specifying where the metadata for the resource group will be stored. Front Door resource itself is created as a global resource and the configuration is deployed globally to all the POPs (Point of Presence).

What are the POP locations for Azure Front Door Service?

Azure Front Door Service has the same list of POP (Point of Presence) locations as Azure CDN from Microsoft. For the complete list of our POPs, kindly refer Azure CDN POP locations from Microsoft.

Is Azure Front Door Service a dedicated deployment for my application or is it shared across customers?

Azure Front Door Service is a globally distributed multi-tenant service. So, the infrastructure for Front Door is shared across all its customers. However, by creating a Front Door profile, you define the specific configuration required for your application and no changes made to your Front Door impact other Front Door configurations.

Is HTTP->HTTPS redirection supported?

Yes. In fact, Azure Front Door Service supports host, path, and query string redirection as well as part of URL redirection. Learn more about URL redirection.

In what order are routing rules processed?

Routes for your Front Door are not ordered and a specific route is selected based on the best match. Learn more about How Front Door matches requests to a routing rule.

How do I lock down the access to my backend to only Azure Front Door?

To lock down your application to accept traffic only from your specific Front Door, you will need to set up IP ACLs for your backend and then restrict the set of accepted values for the header 'X-Forwarded-Host' sent by Azure Front Door. These steps are detailed out as below:

- Configure IP ACLing for your backends to accept traffic from Azure Front Door's backend IP address space and Azure's infrastructure services only. We are working towards integrating with Azure IP Ranges and Service Tags but for now you can refer the IP ranges as below:
  - Front Door's **IPv4** backend IP space: `147.243.0.0/16`
  - Front Door's **IPv6** backend IP space: `2a01:111:2050::/44`
  - Azure's basic infrastructure services through virtualized host IP addresses: `168.63.129.16` and `169.254.169.254`

  > **WARNING**
  >
  > Front Door's backend IP space may change later, however, we will ensure that before that happens, that we would have integrated with Azure IP Ranges and Service Tags. We recommend that you subscribe to Azure IP Ranges and Service Tags for any changes or updates.

- Filter on the values for the incoming header '**X-Forwarded-Host**' sent by Front Door. The only allowed values for the header should be all of the frontend hosts as defined in your Front Door config. In fact even more specifically, only the host names for which you want to accept traffic from, on this particular backend of yours.

  - Example – let's say your Front Door config has the following frontend hosts `contoso.azurefd.net` (A), `www.contoso.com` (B), `api.contoso.com` (C), and `notifications.contoso.com` (D). Let's assume that you have two backends X and Y.
  - Backend X should only take traffic from host names A and B. Backend Y can take traffic from A, C, and D.
  - So, on Backend X you should only accept traffic that has the header '**X-Forwarded-Host**' set to either `contoso.azurefd.net` or `www.contoso.com`. For everything else, backend X should reject the traffic.
  - Similarly, on Backend Y you should only accept traffic that has the header "**X-Forwarded-Host**" set to either `contoso.azurefd.net`, `api.contoso.com` or `notifications.contoso.com`. For everything else, backend Y should reject the traffic.

Can the anycast IP change over the lifetime of my Front Door?

The frontend anycast IP for your Front Door should typically not change and may remain static for the lifetime of the Front Door. However, there are **no guarantees** for the same. Kindly do not take any direct dependencies on the IP.

Does Azure Front Door Service support static or dedicated IPs?

No, Azure Front Door Service currently doesn't support static or dedicated frontend anycast IPs.

Does Azure Front Door Service support x-forwarded-for headers?

Yes, Azure Front Door Service supports the X-Forwarded-For, X-Forwarded-Host, and X-Forwarded-Proto headers. For X-Forwarded-For if the header was already present then Front Door appends the client socket IP to it. Else, it adds the header with the client socket IP as the value. For X-Forwarded-Host and X-Forwarded-Proto, the value is overridden.

Learn more about the Front Door supported HTTP headers.

How long does it take to deploy an Azure Front Door Service? Does my Front Door still work when being updated?

A new Front Door creation or any updates to an existing Front Door takes about 3 to 5 minutes for global deployment. That means in about 3 to 5 minutes, your Front Door configuration will be deployed across all of our POPs globally.

Note - Custom SSL certificate updates take about 30 minutes to be deployed globally.

Configuration

Can Azure Front Door load balance or route traffic within a virtual network?

Azure Front Door (AFD) requires a public IP or publicly resolvable DNS name to route traffic. So, the answer is no AFD directly cannot route within a virtual network, but using an Application Gateway or Azure Load Balancer in between will solve this scenario.

What are the various timeouts and limits for Azure Front Door Service?

Learn about all the documented timeouts and limits for Azure Front Door Service.

Performance

How does Azure Front Door Service support high availability and scalability?

Azure Front Door Service is a globally distributed multi-tenant platform with huge volumes of capacity to cater to your application's scalability needs. Delivered from the edge of Microsoft's global network, Front Door provides global load-balancing capability that allows you to fail over your entire application or even individual microservices across regions or different clouds.

SSL configuration

What TLS versions are supported by Azure Front Door Service?

All Front Door profiles created after September 2019 use TLS 1.2 as the default minimum.

Front Door supports TLS versions 1.0, 1.1 and 1.2. TLS 1.3 is not yet supported.

What certificates are supported on Azure Front Door Service?

To enable the HTTPS protocol for securely delivering content on a Front Door custom domain, you can choose to use a certificate that is managed by Azure Front Door Service or use your own certificate. The Front Door managed option provisions a standard SSL certificate via Digicert and stored in Front Door's Key Vault. If you choose to use your own certificate, then you can onboard a certificate from a supported CA and can be a standard SSL, extended validation certificate, or even a wildcard certificate. Self-signed certificates are not supported. Learn how to enable HTTPS for a custom domain.

Does Front Door support autorotation of certificates?

For the Front Door managed certificate option, the certificates are autorotated by Front Door. If you are using a Front Door managed certificate and see that the certificate expiry date is less than 60 days away, file a support ticket.
For your own custom SSL certificate, autorotation isn't supported. Similar to how it was set up the first time for a given custom domain, you will need to point Front Door to the right certificate version in your Key Vault and ensure that the service principal for Front Door still has access to the Key Vault. This updated certificate rollout operation by Front Door is atomic and doesn't cause any production impact provided the subject name or SAN for the certificate doesn't change.

What are the current cipher suites supported by Azure Front Door Service?

The following are the current cipher suites supported by Azure Front Door Service:

- TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256
- TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384
- TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256
- TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384
- TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256
- TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384
- TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256
- TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384
- TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA
- TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA
- TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA
- TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA
- TLS_RSA_WITH_AES_256_GCM_SHA384
- TLS_RSA_WITH_AES_128_GCM_SHA256
- TLS_RSA_WITH_AES_256_CBC_SHA256
- TLS_RSA_WITH_AES_128_CBC_SHA256
- TLS_RSA_WITH_AES_256_CBC_SHA
- TLS_RSA_WITH_AES_128_CBC_SHA
- TLS_DHE_RSA_WITH_AES_128_GCM_SHA256
- TLS_DHE_RSA_WITH_AES_256_GCM_SHA384

Does Azure Front Door Service also support re-encryption of traffic to the backend?

Yes, Azure Front Door Service supports SSL offload, and end to end SSL, which re-encrypts the traffic to the backend. In fact, since the connections to the backend happen over it's public IP, it is recommended that you configure your Front Door to use HTTPS as the forwarding protocol.

Can I configure SSL policy to control SSL Protocol versions?

You can configure a minimum TLS version in Azure Front Door via the Azure REST API. Currently, you can choose between 1.0 and 1.2.

Can I configure Front Door to only support specific cipher suites?

No, configuring Front Door for specific cipher suites is not supported.

Diagnostics and logging

What types of metrics and logs are available with Azure Front Door Service?

For information on logs and other diagnostic capabilities, see Monitoring metrics and logs for Front Door.

What is the retention policy on the diagnostics logs?

Diagnostic logs flow to the customers storage account and customers can set the retention policy based on their preference. Diagnostic logs can also be sent to an Event Hub or Azure Monitor logs. For more information, see Azure Front Door Service Diagnostics.

How do I get audit logs for Azure Front Door Service?

Audit logs are available for Azure Front Door Service. In the portal, click **Activity Log** in the menu blade of your Front Door to access the audit log.

Can I set alerts with Azure Front Door Service?

Yes, Azure Front Door Service does support alerts. Alerts are configured on metrics.

Next steps

- Learn how to create a Front Door.
- Learn how Front Door works.