Microsoft Azure Blob Storage

Last updated April 09, 2019

Microsoft Azure Blob Storage (https://azure.microsoft.com/en-us/services/storage/blobs/) public and private containers can be used as origins with Fastly.

> ★ **TIP:** With properly configured services in place, shared Fastly and Microsoft customers will benefit from Fastly's integration with Azure's ExpressRoute Direct Local, which results in Fastly including your outbound data transfer costs from Azure in your standard Fastly pricing. See our guide to outbound data transfers from Azure (/guides/integrations/outbound-data-transfer-from-azure) for more details.

Using Azure Blob Storage as an origin

To make your Azure Blob Storage stores available through Fastly, follow the steps below.

Creating a new service

Follow the instructions for creating a new service (/guides/basic-setup/working-with-services#creating-a-new-service). You'll add specific details about your origin when you fill out the **Create a new service** fields:

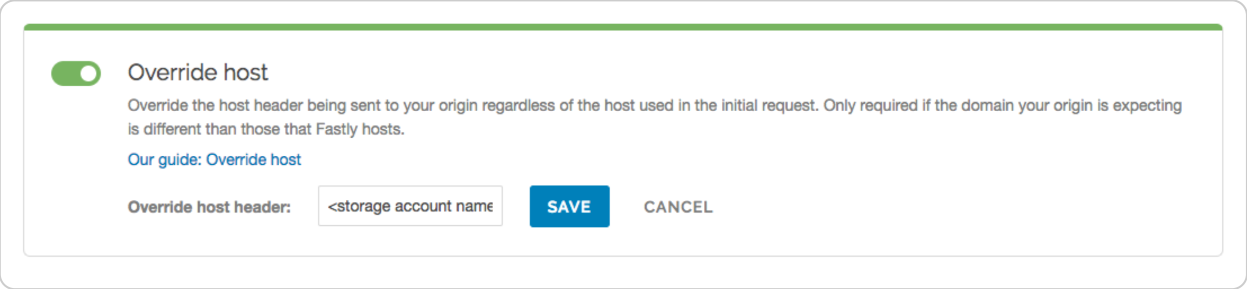
- In the **Name** field, type any descriptive name for your service.
- In the **Domain** field, type the hostname you want to use as the URL (e.g., `cdn.example.com`).
- In the **Address** field, type `<storage account name>.blob.core.windows.net`.
- In the **Transport Layer Security (TLS)** area, leave the **Enable TLS?** default set to **Yes** to secure the connection between Fastly and your origin.
- In the **Transport Layer Security (TLS)** area, type `<storage account name>.blob.core.windows.net` in the **Certificate hostname** field.

Setting the default host and correct path

Once the new service is created, set the default host to `azure` and then add your container path to the URL by following the steps below:

1. From the service menu, select the appropriate service.

2. Click the **Configuration** button and then select **Clone active**. The Domains page appears.
3. Click the **Settings** link. The Settings page appears.
4. Click the **Override host** switch. The Override host header field appears.



Override host

Override the host header being sent to your origin regardless of the host used in the initial request. Only required if the domain your origin is expecting is different than those that Fastly hosts.

Our guide: Override host

Override host header: <storage account name

5. Type the hostname of your Azure Blob Storage account. For example, `<storage account name>.blob.core.windows.net`.
6. Click the **Save** button. The new override host header appears in the Override host section.
7. Click the **Content** link. The Content page appears.
8. Click the **Create header** button. The Create a header page appears.
9. Fill out the **Create a header** fields as follows:
   - In the **Name** field, type `Modify URL`.
   - From the **Type** menu, select **Request**, and from the **Action** menu, select **Set**.
   - In the **Destination** field, type `url`.
   - In the **Source** field, type `"/<your container name>" req.url`.
   - From the **Ignore if set** menu, select **No**.
   - In the **Priority** field, type `10`.
10. Click the **Create** button. The new Modify URL header appears on the Content page.
11. Click the **Activate** button to deploy your configuration changes.

Testing your results

By default, we create DNS mapping called **yourdomain.global.prod.fastly.net**. In the example above, it would be `cdn.example.com.global.prod.fastly.net`. Create a DNS alias for the domain name you specified (e.g., CNAME `cdn.example.com` to `global-nossl.fastly.net`).

Fastly will cache any content without an explicit `Cache-Control` header for 1 hour. You can verify whether you are sending any cache headers using cURL. For example:

```
$ curl -I opscode-full-stack.blob.core.windows.net

HTTP/1.1 200 OK
Date: Fri, 04 May 2018 21:23:07 GMT
Content-Type: application/xml
Transfer-Encoding: chunked
Server: Blob Service Version 1.0 Microsoft-HTTPAPI/2.0
```

In this example, no cache control headers are set so the default TTL will be applied.

Using an Azure Blob Storage private container

To use an Azure Blob Storage private container with Fastly, follow the instructions below.

Before you begin

Be sure you've already made your Azure Blob Storage containers available to Fastly by pointing to the right container and setting your origin to port 443. This needs to be done before authenticating.

To complete the setup, you'll also need your Azure Storage Account shared key and storage account name to construct the Azure Blob Storage Authorization header, which takes the following form:

```
Authorization: SharedKey `_Account name_`:`_Signature_`
```

Finally, you'll also need to know your Blob Storage container name.

Setting up Fastly to use an Azure Blob Storage private container with a Shared Key

> ⚠ **WARNING:** Your account's Shared Key does not have detailed access control. Anyone with access to your Shared Key can read and write to your container. Consider using a Shared Access Signature (SAS) instead.

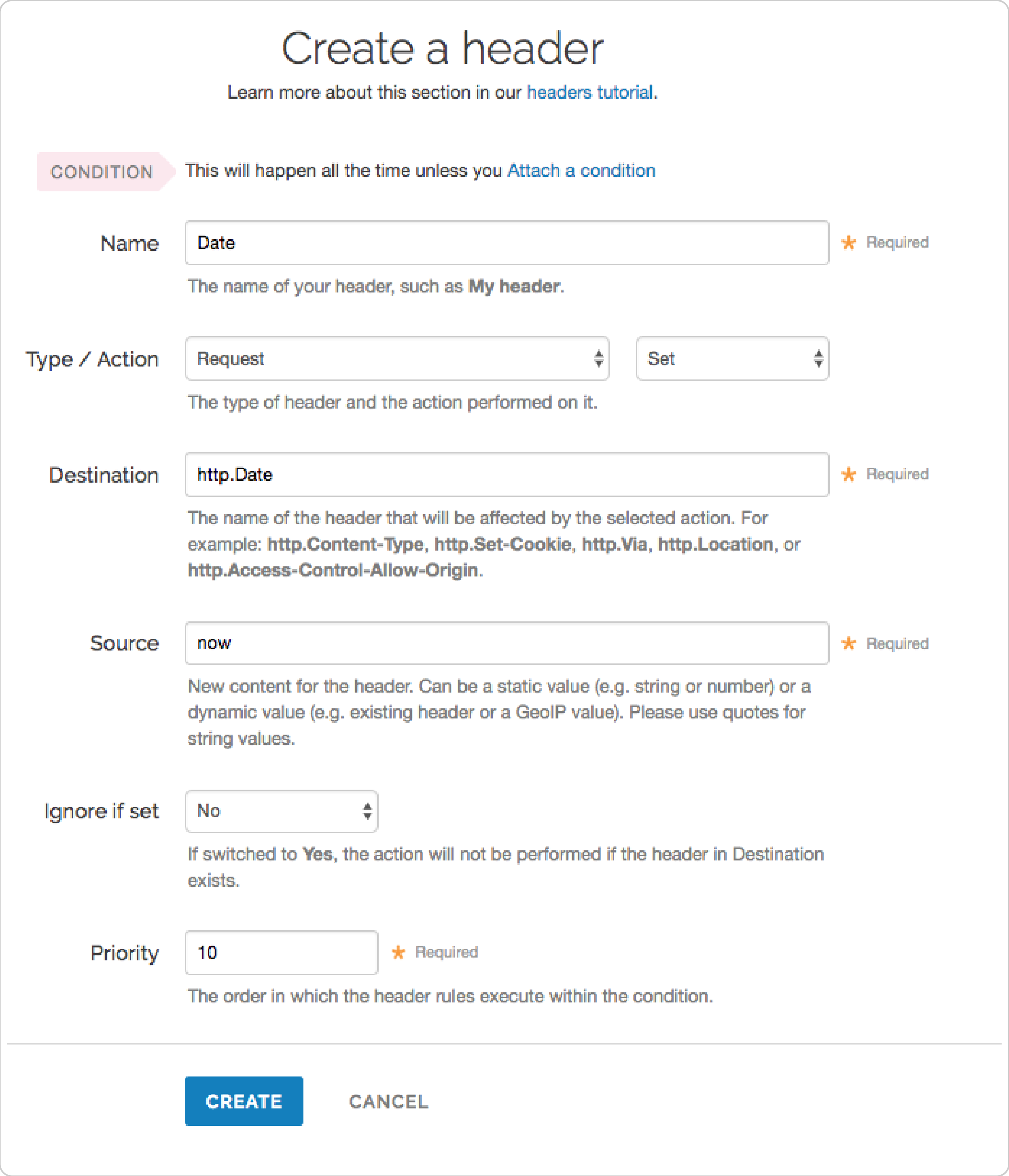
To access an Azure Blob Storage private container with Fastly using a Shared Key, read Microsoft's "Authorize with Shared Key (https://docs.microsoft.com/en-us/rest/api/storageservices/authorize-with-shared-key)" page. Then, create two headers (/guides/basic-configuration/adding-or-modifying-headers-on-http-requests-and-responses): a Date header (for use with the authorization Signature) and an Authorization header.

Create a Date header

Create the Date header using the steps below.

1. Log in to the Fastly web interface and click the **Configure** link.

2. From the service menu, select the appropriate service.
3. Click the **Configuration** button and then select **Clone active**. The Domains page appears.
4. Click the **Content** link. The Content page appears.
5. Click the **Create header** button. The Create a header page appears.



Create a header

Learn more about this section in our headers tutorial.

CONDITION This will happen all the time unless you Attach a condition

Name: Date ★ Required

The name of your header, such as **My header**.

Type / Action: Request / Set

The type of header and the action performed on it.

Destination: http.Date ★ Required

The name of the header that will be affected by the selected action. For example: **http.Content-Type**, **http.Set-Cookie**, **http.Via**, **http.Location**, or **http.Access-Control-Allow-Origin**.

Source: now ★ Required

New content for the header. Can be a static value (e.g. string or number) or a dynamic value (e.g. existing header or a GeoIP value). Please use quotes for string values.

Ignore if set: No

If switched to **Yes**, the action will not be performed if the header in Destination exists.

Priority: 10 ★ Required

The order in which the header rules execute within the condition.

6. Fill out the **Create a header** fields as follows:
   - In the **Name** field, type `Date`.

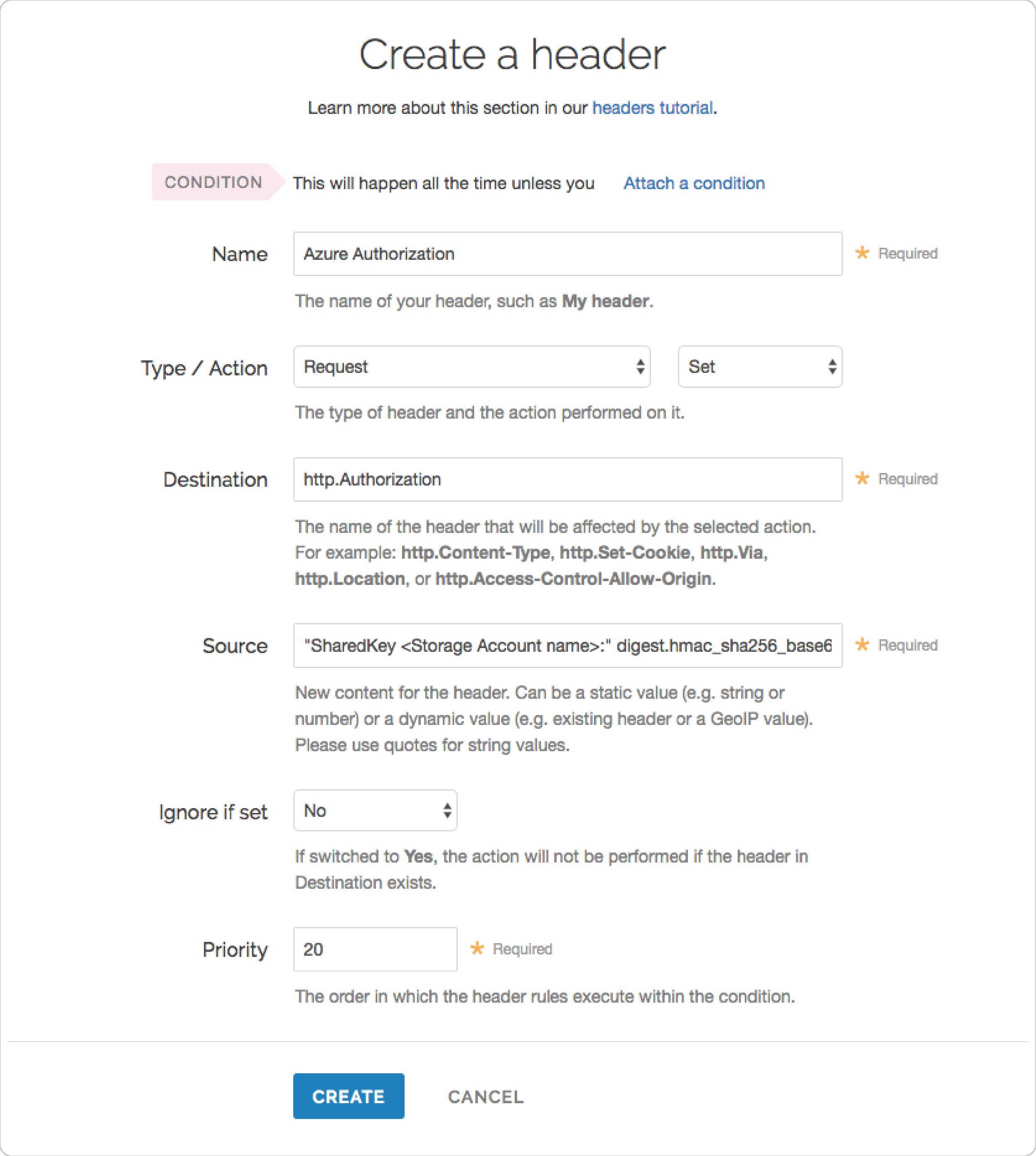
- From the **Type** menu, select **Request**, and from the **Action** menu, select **Set**.
- In the **Destination** field, type `http.Date`.
- In the **Source** field, type `now`.
- From the **Ignore if set** menu, select **No**.
- In the **Priority** field, type `10`.

7. Click the **Create** button. A new Date header appears on the Content page. You will use this later within the Signature of the Authorization header.

Create an Authorization header

Next, create the Authorization header with the specifications listed below.

1. Click the **Create header** button again to create another new header. The Create a header page appears.



Create a header

Learn more about this section in our headers tutorial.

This will happen all the time unless you Attach a condition

Name: Azure Authorization ★ Required

The name of your header, such as **My header**.

Type / Action: Request / Set

The type of header and the action performed on it.

Destination: http.Authorization ★ Required

The name of the header that will be affected by the selected action. For example: **http.Content-Type**, **http.Set-Cookie**, **http.Via**, **http.Location**, or **http.Access-Control-Allow-Origin**.

Source: "SharedKey <Storage Account name>:" digest.hmac_sha256_base6 ★ Required

New content for the header. Can be a static value (e.g. string or number) or a dynamic value (e.g. existing header or a GeoIP value). Please use quotes for string values.

Ignore if set: No

If switched to **Yes**, the action will not be performed if the header in Destination exists.

Priority: 20 ★ Required

The order in which the header rules execute within the condition.

2. Fill out the **Create a header** fields as follows:
   - In the **Name** field, type `Azure Authorization`.
   - From the **Type** menu, select **Request**, and from the **Action** menu, select **Set**.
   - In the **Destination** field, type `http.Authorization`.
   - From the **Ignore if set** menu, select **No**.
   - In the **Priority** field, type `20`.
3. In the **Source** field, type the header authorization information using the following format:

```
"SharedKey <Storage Account name>:" digest.hmac_sha256_base64(digest.base64_decode("<Azure Storage Account shared key>"), if(req.method == "HEAD", "GET", req.method) LF LF LF req.http.Date LF "/<Storage Account name>" req.url.path)
```

replacing `<Storage Account name>` and `<Azure Storage Account shared key>` with the information you gathered before you began. For example:

```
"SharedKey test123:" digest.hmac_sha256_base64(digest.base64_decode("UDJXUFN1NjhCZmw4OWo3MnZUK2JYWVpCN1NqbE93aFQ0d2hxdDI3"), if(req.method == "HEAD", "GET", req.method) LF LF LF req.http.Date LF "/test123" req.url.path)
```

We provide a detailed look at the Source field parameters below.

4. Click the **Create** button. The new Authorization header appears on the Content page.
5. Click the **Activate** button to deploy your configuration changes.

A detailed look at the Source field

So what's going on in the Source field of the Authorization header? Here's the basic format:

`SharedKey<storage account name><Signature Function><key><message>`

It tells us the following:

Element	Description
`SharedKey`	A constant placed before the storage account name. It's always SharedKey.
`storage account name`	The name of your Azure Storage Account. We used `test123` in this example.
`signature function`	The algorithm used to validate the key and message of the signature. We used `digest.hmac_sha256_base64(<key>, <message>)` in this example.
`key`	The Azure Storage Account shared key from your Azure Storage developer's account. We used `UDJXUFN1NjhCZmw4OWo3MnZUK2JYWVpCN1NqbE93aFQ0d2hxdDI3` in this example. It must be Base64 decoded.
`message`	The UTF-8 encoding of the StringToSign. See the table below for a break down of each portion of the message.

The message that's part of the Source field in the Authorization header takes on this basic format:

`<HTTP-verb></n><Content-MD5>/n<Content-Type></n><Date></n>`

`<CanonicalizedAmzHeader></n><CanonicalizedResource>`

It tells us the following:

Element	Description
`HTTP-verb`	The REST verb. We use `req.method` in this example. We rewrite HEAD to GET because Varnish does this internally before sending requests to origin.
`\n`	A newline indicator constant. It's always \n.
`Content-MD5`	The content-md5 header value, used as a message integrity check. It's often left blank. We use `LF` (line feed) in this example.
`Content-Type`	The content-type header value, used to specify the MIME-type. It's often left blank. We use `LF`in this example.
`Date`	The date and time stamp. We use `req.http.Date` (which we created first as a separate header in the steps above).
`CanonicalizedHeaders`	The x-ms headers, which customize your Azure Blob Storage implementation. It's often left blank. We use `LF` in this example.
`CanonicalizedResource`	Your Storage Account Name. We use `"/test123"` in this example.

Setting up Fastly to use an Azure Blob Storage private container with a Shared Access Signature (SAS)

To access an Azure Blob Storage private container with Fastly using a Service Shared Access Signature (SAS), read Microsoft's "Delegating Access with a Shared Access Signature (https://docs.microsoft.com/en-us/rest/api/storageservices/delegating-access-with-a-shared-access-signature)" page. Then, obtain the SAS and sign the access URL.

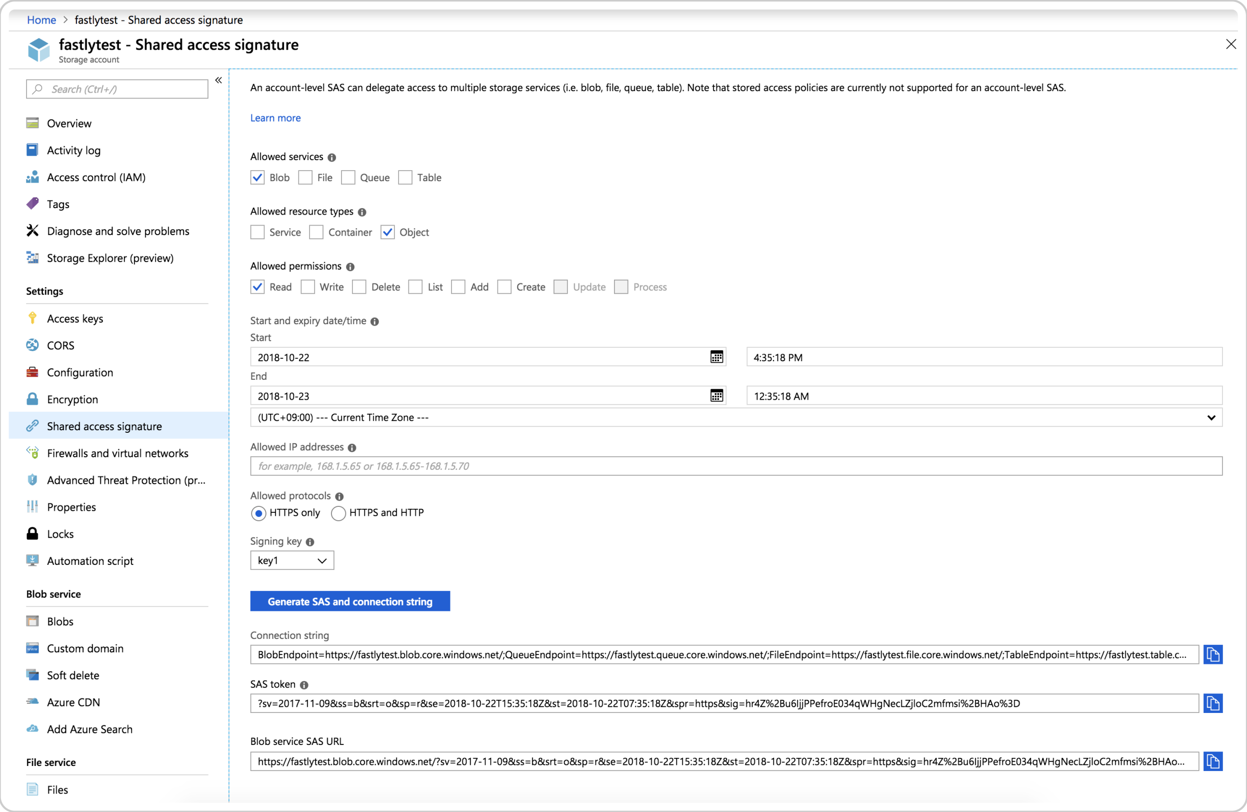
★ **TIP:** Using a Service Shared Access Signature gives you more detailed control over:

- The interval during which the SAS is valid, including the start time and the expiry time.
- The permissions granted by the SAS. For example, a SAS for a blob might grant read and write permissions to that blob, but not delete permissions.
- An optional IP address or range of IP addresses from which Azure Storage will accept the SAS. For example, you might specify a range of IP addresses belonging to your organization.
- The protocol over which Azure Storage will accept the SAS. You can use this optional parameter to restrict access to clients using HTTPS.

Obtaining the Shared Access Signature

Obtain the SAS using the steps below.

1. In the Azure portal, navigate to your storage account
2. Under settings navigate to **Shared access signature**. The Shared access signature controls appear.



An account-level SAS can delegate access to multiple storage services (i.e. blob, file, queue, table). Note that stored access policies are currently not supported for an account-level SAS.

Allowed services
☑ Blob ☐ File ☐ Queue ☐ Table

Allowed resource types
☐ Service ☐ Container ☑ Object

Allowed permissions
☑ Read ☐ Write ☐ Delete ☐ List ☐ Add ☐ Create ☐ Update ☐ Process

Start and expiry date/time
Start
2018-10-22 4:35:18 PM
End
2018-10-23 12:35:18 AM
(UTC+09:00) --- Current Time Zone ---

Allowed IP addresses
for example, 168.1.5.65 or 168.1.5.65-168.1.5.70

Allowed protocols
● HTTPS only ○ HTTPS and HTTP

Signing key
key1

Connection string
BlobEndpoint=https://fastlytest.blob.core.windows.net/;QueueEndpoint=https://fastlytest.queue.core.windows.net/;FileEndpoint=https://fastlytest.file.core.windows.net/;TableEndpoint=https://fastlytest.table.c...

SAS token
?sv=2017-11-09&ss=b&srt=o&sp=r&se=2018-10-22T15:35:18Z&st=2018-10-22T07:35:18Z&spr=https&sig=hr4Z%2Bu6ljjPPefroE034qWHgNecLZjloC2mfmsi%2BHAo%3D

Blob service SAS URL
https://fastlytest.blob.core.windows.net/?sv=2017-11-09&ss=b&srt=o&sp=r&se=2018-10-22T15:35:18Z&st=2018-10-22T07:35:18Z&spr=https&sig=hr4Z%2Bu6ljjPPefroE034qWHgNecLZjloC2mfmsi%2BHAo...

3. From the **Allowed services** controls, select **Blob**.
4. From the **Allowed resource types** controls, select **Object**.

5. From the **Allowed permissions** controls, select **Read**.
6. Leave the **Start time** set to the current date and time.
7. Set the **End time** as far in the future as you are comfortable (see note below).
8. Ensure the **Allowed protocols** remain set to **HTTPS only**.
9. Click the **Generate SAS and connection string** button. The generated information appears.
10. Copy and save the contents of the **SAS token** field. It will look something like:

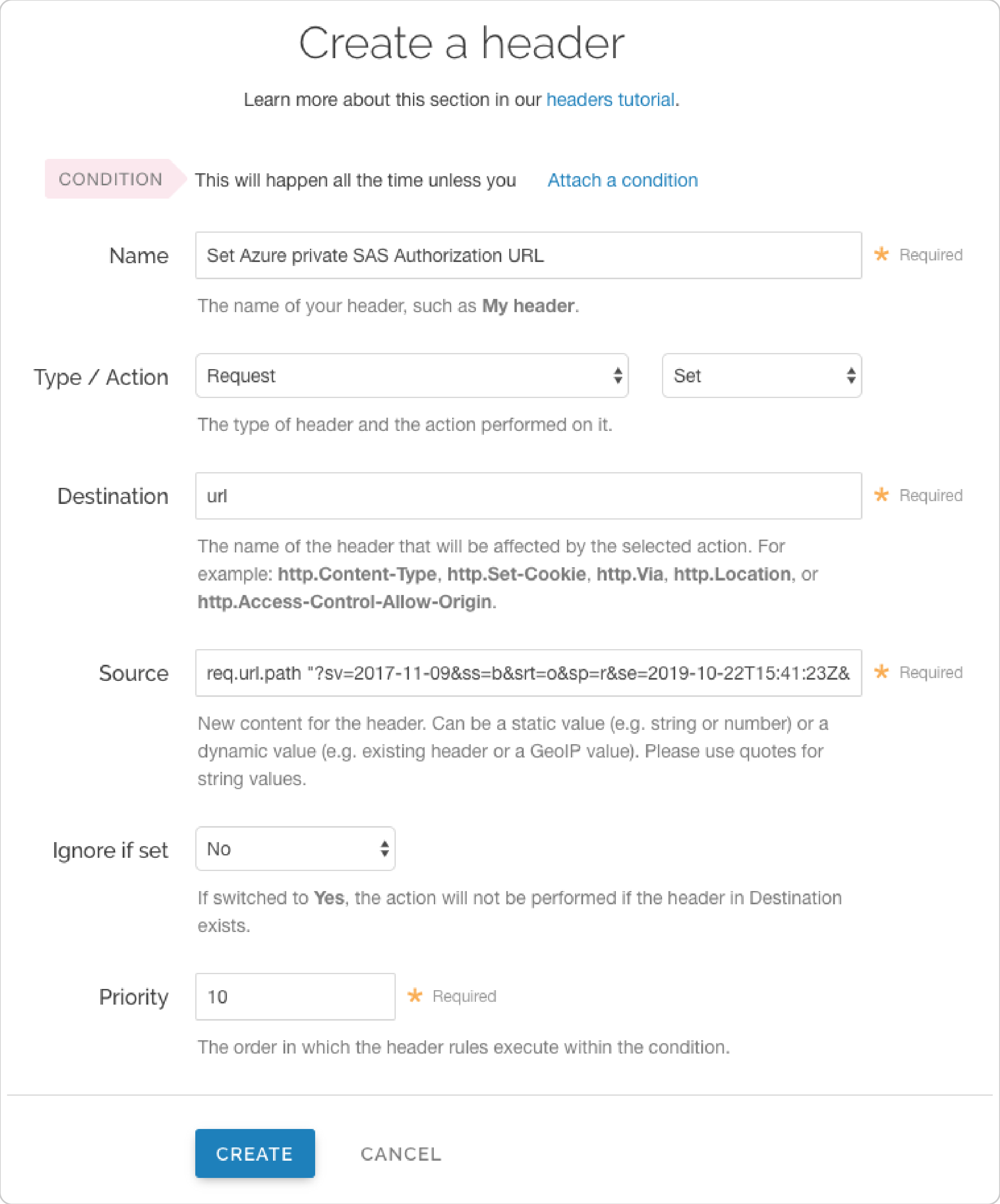
    ```
    ?sv=2017-11-09&ss=b&srt=o&sp=r&se=2019-10-22T15:41:23Z&st=2018-10-22T07:41:23Z&spr=https&sig=decafbaddeadbeef
    ```

    We provide a detailed look at the Shared Access Signature parameters below.

Signing the URL

Next, sign the access URL by creating an authorization header following the steps below.

1. Log in to the Fastly web interface and click the **Configure** link.
2. From the service menu, select the appropriate service.
3. Click the **Configuration** button and then select **Clone active**. The Domains page appears.
4. Click the **Content** link. The Content page appears.
5. Click the **Create header** button. The Create a header page appears.



Create a header

Learn more about this section in our headers tutorial.

CONDITION This will happen all the time unless you Attach a condition

Name: Set Azure private SAS Authorization URL ★ Required

The name of your header, such as **My header**.

Type / Action: Request | Set

The type of header and the action performed on it.

Destination: url ★ Required

The name of the header that will be affected by the selected action. For example: **http.Content-Type**, **http.Set-Cookie**, **http.Via**, **http.Location**, or **http.Access-Control-Allow-Origin**.

Source: req.url.path "?sv=2017-11-09&ss=b&srt=o&sp=r&se=2019-10-22T15:41:23Z& ★ Required

New content for the header. Can be a static value (e.g. string or number) or a dynamic value (e.g. existing header or a GeoIP value). Please use quotes for string values.

Ignore if set: No

If switched to **Yes**, the action will not be performed if the header in Destination exists.

Priority: 10 ★ Required

The order in which the header rules execute within the condition.

6. Fill out the **Create a header** fields as follows:
   - In the **Name** field, type a meaningful name such as `Set Azure private SAS Authorization URL`.
   - From the **Type** menu, select **Request**, and from the **Action** menu, select **Set**.
   - In the **Destination** field, type `url`.
   - In the **Source** field, type `req.url.path "<SAS TOKEN>"` replacing `"<SAS TOKEN>"` with the token you obtained from the Azure Portal.

  - From the **Ignore if set** menu, select **No**.
  - In the **Priority** field, type `10`.

7. Click the **Create** button. A new header appears on the Content page.
8. Click the **Activate** button to deploy your configuration changes.

A detailed look at the Shared Access Signature parameters

Microsoft's "Constructing a Service SAS (https://docs.microsoft.com/en-us/rest/api/storageservices/constructing-a-service-sas)" page provides more details on shared access signatures and how they are constructed.

Element	Description
`sv`	The `signedversion` field. This is required and should be whatever the Azure portal provided.
`ss`	The `signedservice` field. This is required and should be `b` for "blob storage."
`srt`	The `signedresourcetype` field. This is required and should be `o` for "object."
`sp`	The `signedpermissions` field. This is required and should be `r` for "read only."
`st`	The `signedstart` field. This is optional and specifies, in a UTC format compatible with ISO 8601, the time at which the shared access signature becomes valid. If omitted, the start time for this call is assumed to be the time when the storage service receives the request.
`se`	The `signedexpiry` field. This is required and specifies, in a UTC format compatible with ISO 8601, the time at which the shared access signature becomes invalid.
`spr`	The `signedprotocol` field. This is optional and specifies which HTTP protocol (`http` or `https`) the container should use for access. We recommend `https`.
`sig`	The `signature` field. This is required and should be whatever the Azure portal provided.

⚠ **WARNING:** Always keep track of the `se` expiry date. After it has passed, Fastly will not be able to access your private container.

This article describes an integration with a service provided by a third party. Please see our note on integrations (/guides/integrations/).