



SHARED ACCESS SIGNATURE BLOB LEVEL

In Microsoft Azure, shared access keys are a set of security credentials that are used to authenticate and authorize access to Azure resources. These keys are associated with an Azure storage account, and they provide a way to securely access and manage the resources within that account. Shared access keys consist of two components: the storage account name and the key itself.

Here's a brief overview of how shared access keys work in Azure:

1. **Storage Account:** A storage account in Azure is a logical container for storing and managing data objects, such as blobs, queues, tables, and files.
2. **Shared Access Keys:** Each Azure storage account has two primary keys, known as the primary key and the secondary key. Both keys provide the same level of access and can be used interchangeably. The use of primary and secondary keys allows for seamless key rotation without disrupting access to resources.
3. **Access Control:** To access resources within an Azure storage account, you must include the shared access key as part of the request. This key is used for authentication and authorization purposes. By sharing the key with trusted entities, you grant them the necessary permissions to perform specific operations on the storage account.
4. **Key Management:** It's crucial to manage shared access keys securely. Azure provides the option to regenerate keys when needed. Regenerating keys helps in scenarios such as a compromised key or routine key rotation for security best practices.
5. **Limited Access Periods:** Shared access signatures (SAS) provide a more secure way to grant limited access to resources without exposing the actual keys. SAS tokens can be generated with specific permissions and validity periods, reducing the risk associated with using shared access keys directly.

In this tutorial, we're demonstrating how to generate a Shared Access Signature (SAS) for an Azure Storage blob and access it using the generated SAS URL. The end goal is to provide controlled and temporary access to the blob, allowing users to share it securely without compromising the primary access keys of the Azure Storage account. By generating a SAS with specific permissions, validity periods, IP restrictions, and protocol options, users can ensure that only authorized individuals or applications can access the blob for a limited duration and under defined conditions. This enhances security and control over access to Azure Storage resources.



TO BEGIN WITH LAB:

1. Log in to azure portal. Navigate to your storage account.
2. Then go to your container.

+ Container Change access level Restore containers Refresh Delete Give feedback			
Search containers by prefix <input type="text"/>			
<input type="checkbox"/> Show deleted containers			
Name	Last modified	Anonymous access level	Lease state
<input type="checkbox"/> \$logs	1/6/2024, 11:16:35 PM	Private	Available ...
<input type="checkbox"/> data	1/10/2024, 1:25:05 PM	Private	Available ...

Upload

Change access level

Refresh

Delete

Change tier

Acquire lease

Break lease

View snapshots

Create snapshot

Give feedback



Authentication method: Access key (Switch to Microsoft Entra user account)

Location: data

Search blobs by prefix (case-sensitive)

Show deleted blobs

Add filter

	Name	Modified	Access tier	Archive status	Blob type	Size	Lease state	
<input type="checkbox"/>	 Dockerfile	1/10/2024, 1:23:21 PM	Hot (Inferred)		Block blob	113 B	Available	...
<input type="checkbox"/>	 script.yml	1/10/2024, 1:23:08 PM	Hot (Inferred)		Block blob	58 B	Available	...

3. Now open one of your files. And you will see a bunch of options to choose from.

«

Dockerfile

Blob

↑ Upload

🔒 Change access level

⋮

Authentication method: Access key ([Switch to Microsoft Entra user account](#))

Location: data

Search blobs by prefix (case-...)

Show deleted blobs

+ Add filter

Name

☒ Dockerfile

⋮

☐ script.yml

⋮

Save

✕ Discard

↓ Download

🔄 Refresh

🗑️ Delete

↔️ Change tier

🔗 Acquire lease

🔗 Break lease

🗨️ Give feedback

Overview

Versions

Snapshots

Edit

Generate SAS

Properties

URL

https://appstorage2711...

LAST MODIFIED

1/10/2024, 1:23:21 PM

CREATION TIME

1/10/2024, 1:23:21 PM

VERSION ID

-

TYPE

Block blob

SIZE

113 B

ACCESS TIER

Hot (Inferred)

ACCESS TIER LAST MODIFIED

N/A

ARCHIVE STATUS

-

REHYDRATE PRIORITY

-

SERVER ENCRYPTED

true

ETAG

0x8DC11B137108150

VERSION-LEVEL IMMUTABILITY POLICY

Disabled

CACHE-CONTROL

CONTENT-TYPE

application/octet-stream

CONTENT-MD5

YFPHfQafEI7SzUlgRqHig==

CONTENT-ENCODING

CONTENT-LANGUAGE

CONTENT-DISPOSITION

LEASE STATUS

Unlocked

LEASE STATE

Available

LEASE DURATION

-

COPY STATUS

-

COPY COMPLETION TIME

-

4. If you will click on generate SAS (Shared Access Signature)

Overview Versions Snapshots Edit Generate SAS

5. Here you can see that you have so many options to generate SAS.
6. Like you can set permission to read, add, create, write etc.
7. You can set a start and expiry date for it.
8. Then you can allow only a certain set of IP addresses.
9. You can also allow certain level of protocols.
10. So, if you will just click on Generate SAS token and URL.

Overview Versions Snapshots Edit **Generate SAS**

A shared access signature (SAS) is a URI that grants restricted access to an Azure Storage blob. Use it when you want to grant access to storage account resources for a specific time range without sharing your storage account key. [Learn more about creating an account SAS](#)

Signing method
☒ Account key ☐ User delegation key

Signing key
 Key 1

Stored access policy
 None

Permissions *
 Read

Start and expiry date/time
 Start
 10/01/2024 1:41:35 PM
 (UTC+05:30) Chennai, Kolkata, Mumbai, New Delhi

Expiry
 10/01/2024 9:41:35 PM
 (UTC+05:30) Chennai, Kolkata, Mumbai, New Delhi

Allowed IP addresses
 for example, 168.1.5.65 or 168.1.5.65-168.1.5.65

Allowed protocols
☒ HTTPS only ☐ HTTPS and HTTP

Generate SAS token and URL

11. You can see that you have two different things, one is token and other is URL.

Blob SAS token
 sp=r&st=2024-01-10T08:11:35Z&se=2024-01-10T16:11:35Z&spr=https&sv=2022-11-02&sr=b&sig=6dEJhz%2F1Q%2FE4rtRc8lHYNbFus%2B0drWU0YXdWUqW%2F4%3D

Blob SAS URL
 https://appstorage2711.blob.core.windows.net/data/Dockerfile?sp=r&st=2024-01-10T08:11:35Z&se=2024-01-10T16:11:35Z&spr=https&sv=2022-11-02&sr=b&sig=6dEJhz%2F1Q%2FE4rtRc8lHYNbFus%2B0drWU...

12. Now you have to copy Blob SAS URL, and paste it in a new tab.



13. You can see that it is asking you to save to some place which the SAS URL is working properly.

