AZ-204 - 3. Implement Azure security -> 3.1 Implement user authentication and authorization -> 3.1.3 Create and implement shared access signatures

1. What is a Shared Access Signature (SAS)?
2. What are the types of SAS and when should each be used?
3. How do you create a SAS using Azure Storage SDK or CLI?
4. What permissions can be specified in a SAS token?
5. How do you specify expiration, allowed IPs, and protocols in a SAS?
6. What is the difference between service SAS and account SAS?
7. How do you restrict SAS access by resource type?
8. How do you implement stored access policies?
9. What are security best practices for using SAS?
10. How do you revoke a SAS token?

---

1. What is a Shared Access Signature (SAS)?
A SAS is a signed URI that grants limited access to Azure Storage resources without exposing account keys. It defines permissions, scope, and expiry.

---

2. What are the types of SAS and when should each be used?

- **User delegation SAS**: Uses Azure AD credentials. Most secure.
- **Service SAS**: Grants access to specific resource (blob, file, etc.).
- **Account SAS**: Grants access to any service in the account (blob, queue, file, table). Use for broader access needs.

---

3. How do you create a SAS using Azure Storage SDK or CLI?

- CLI:

```
az storage blob generate-sas --account-name <name> --container-name <c> --name <blob> --permissions r --expiry <time>
```

- SDK: Use BlobSasBuilder in .NET or equivalent in other languages.

---

4. What permissions can be specified in a SAS token?
Depends on resource type. Examples:

- **Blob**: r (read), w (write), d (delete), l (list), a (add), c (create)
- **Queue**: r, a, u (update), p (process)

---

5. How do you specify expiration, allowed IPs, and protocols in a SAS?
In the SAS definition:

- --expiry (e.g., 2025-05-01T00:00Z)
- --ip (e.g., 168.1.5.60-168.1.5.70)
- --https-only true to restrict to HTTPS.

6. What is the difference between service SAS and account SAS?

- **Service SAS**: Grants access to a specific resource (e.g., a blob).
- **Account SAS**: Grants access across services (Blob, File, Queue, Table) in a storage account. Account SAS is broader and riskier if leaked.

---

7. How do you restrict SAS access by resource type?

Use the --resource-types parameter (for account SAS):

- s (service), c (container), o (object)
  Example:
  --resource-types sco limits access to services, containers, and objects.

---

8. How do you implement stored access policies?

Stored access policies are defined on containers and linked to SAS tokens to centrally manage expiry and permissions.

Create with:

```
az storage container policy create
```

Then reference the --policy-name in SAS generation.

---

9. What are security best practices for using SAS?

- Set short expiry times.
- Use HTTPS only.
- Restrict IP range if possible.
- Prefer user delegation SAS over account SAS.
- Avoid hardcoding SAS; store securely.

---

10. How do you revoke a SAS token?

- For **account/service SAS**: Rotate the storage account key.
- For **stored access policy SAS**: Modify or delete the policy; tokens linked to it become invalid.