4. Security and compliance

└ **4.1 Authentication and authorization**

└ **4.1.3 Azure DevOps service connections, tokens**

---

1. What is a service connection in Azure DevOps?
2. What types of service connections can you configure?
3. How do you securely store and manage service connection credentials?
4. What is the difference between service principals, managed identities, and PATs in Azure DevOps?
5. How do you configure a new Azure Resource Manager (ARM) service connection?
6. What permissions are required to create and use service connections?
7. What are personal access tokens (PATs) and when should they be used?
8. What are best practices for managing and rotating tokens in Azure DevOps?
9. How can you audit usage and security of service connections and tokens?
10. How do you restrict access to service connections in a project or pipeline?

---

1. What is a service connection in Azure DevOps?

A service connection is a secure configuration in *Azure DevOps* that enables pipelines to connect and authenticate with external systems or Azure resources, using credentials or identity.

---

2. What types of service connections can you configure?

Common types include

- Azure Resource Manager,
- GitHub,
- Docker Registry,
- Generic,
- AWS,
- GCP,
- and Service Fabric.

Each targets a specific external service or cloud.

---

3. How do you securely store and manage service connection credentials?

Credentials are encrypted and stored securely in *Azure DevOps*. Sensitive details (like secrets or certificates) are not exposed in pipelines and are managed by *Azure DevOps* security controls.

---

4. What is the difference between service principals, managed identities, and PATs in Azure DevOps?

- **Service principals:** Used for non-interactive authentication to Azure resources; recommended for ARM connections.
- **Managed identities:** Azure AD identities for Azure services; used for enhanced security and automation within Azure.
- **PATs (Personal Access Tokens):** User-scoped tokens for accessing *Azure DevOps* REST APIs; not recommended for automation or service connections due to security risks.

---

5. How do you configure a new Azure Resource Manager (ARM) service connection?

Go to *Project Settings → Service Connections → New service connection → Select Azure Resource Manager → Authenticate (typically with a service principal) → Grant permissions → Save*.

6. What permissions are required to create and use service connections?

- Creating: *Project Administrator* or *Service Connection Admin*.
- Using: Grant access to specific users, groups, or pipelines; restrict to those who need it.

---

7. What are personal access tokens (PATs) and when should they be used?

PATs are user-generated tokens for authenticating to Azure DevOps REST APIs. Use only for short-term, user-scoped automation; avoid for production pipelines or shared services.

---

8. What are best practices for managing and rotating tokens in Azure DevOps?

- Use service principals or managed identities over PATs.
- Set expiration dates on tokens.
- Regularly review and rotate credentials.
- Revoke unused or suspicious tokens promptly.

---

9. How can you audit usage and security of service connections and tokens?

Use *Azure DevOps* Audit Logs to track creation, modification, and usage of service connections and PATs. Review access and permission changes regularly.

---

10. How do you restrict access to service connections in a project or pipeline?

Set security on each service connection by configuring user/group permissions or restricting pipeline usage. Use the “Limit job authorization” and “Grant access permission to all pipelines” settings appropriately.