4. Security and compliance
└ **4.3 Secret and sensitive data management**
└ **4.3.3 Secure files during deployment**

---

1. What are secure files in Azure Pipelines and what are common use cases?
2. How do you upload secure files to Azure Pipelines?
3. How are secure files used in pipeline steps?
4. What permissions are required to access secure files?
5. How does Azure Pipelines handle encryption and storage of secure files?
6. What are best practices for managing certificates or signing keys securely in deployment pipelines?
7. Can secure files be used in GitHub Actions?
8. How can you audit access to secure files in Azure DevOps?
9. What is the difference between pipeline secrets and secure files?
10. How do you prevent accidental exposure of sensitive files during deployment?

---

1. What are secure files in Azure Pipelines and what are common use cases?

Secure files are encrypted files (e.g., certificates, provisioning profiles, signing keys) stored securely in Azure Pipelines.
Use cases: iOS provisioning profiles, Android keystore files, SSL certs, signing keys.

---

2. How do you upload secure files to Azure Pipelines?

Navigate to:
Pipelines → Library → Secure files → "+ Secure file" → Upload and save.
Files are encrypted at rest.

---

3. How are secure files used in pipeline steps?

Use the `DownloadSecureFile@1` task to access them during runtime.
Example:

```
- task: DownloadSecureFile@1
inputs: secureFile: 'cert.pfx'
```

---

4. What permissions are required to access secure files?

Users must have "Download secure files" permission for the pipeline or release.
Access is restricted via Library security roles.

---

5. How does Azure Pipelines handle encryption and storage of secure files?

Files are encrypted with AES-256 at rest and decrypted only in the pipeline execution context. Files are temporary and auto-deleted post-job.

---

6. What are best practices for managing certificates or signing keys securely in deployment pipelines?

- Store certs as secure files
- Restrict access via RBAC
- Use per-environment scopes
- Rotate keys regularly
- Never log or output cert data

7. Can secure files be used in GitHub Actions?
No native secure file equivalent in GitHub. Use base64-encoded secrets or store encrypted files in repo and decrypt during workflow using stored secrets.

---

8. How can you audit access to secure files in Azure DevOps?
Review audit logs in *Azure DevOps* under *Organization Settings → Auditing*. Logs include who uploaded, downloaded, or deleted secure files.

---

9. What is the difference between pipeline secrets and secure files?

- **Secrets:** key-value strings (e.g., API keys, passwords)
- **Secure files:** binary files (e.g., certificates)
  Both are encrypted but serve different deployment needs.

---

10. How do you prevent accidental exposure of sensitive files during deployment?

- Use secure file tasks (never inline file content)
- Disable verbose logging
- Never echo secrets or paths
- Restrict debug mode
- Run deployments in isolated environments