3. Build and release pipelines
└─ **3.7 Maintain and optimize pipelines**
└─ **3.7.3 Artifact retention strategy**

1. What is an artifact retention policy in Azure DevOps?
2. Where do you configure artifact retention in Azure Pipelines?
3. What is the default retention policy for pipeline artifacts?
4. How can you prevent a pipeline run's artifacts from being deleted?
5. How do retention policies differ between classic and YAML pipelines?
6. How does pipeline retention impact storage costs and performance?
7. What permissions are needed to modify retention settings?
8. How do retention settings affect deployment history and auditability?
9. How can you override global retention policies per pipeline?
10. What are best practices for implementing artifact retention strategies?

1. What is an artifact retention policy in Azure DevOps?
A retention policy defines how long pipeline artifacts, logs, and test results are kept before being automatically deleted to manage storage usage.

2. Where do you configure artifact retention in Azure Pipelines?
Retention is configured under *Project Settings → Pipelines → Retention*, or directly in the pipeline YAML using retention under the pipeline block.

3. What is the default retention policy for pipeline artifacts?
The default policy retains artifacts for 30 days or 3 successful runs per branch, whichever is met last.

4. How can you prevent a pipeline run's artifacts from being deleted?
Mark the pipeline run as “retained” either manually via the UI or using `az pipelines runs update --id <runId> --retain true`.

5. How do retention policies differ between classic and YAML pipelines?
Classic pipelines use UI-based retention rules; YAML pipelines can define retention directly in code, allowing more granular, version-controlled policies.

6. How does pipeline retention impact storage costs and performance?
Longer retention increases storage usage and cost; excessive artifact history may degrade performance in large-scale environments.

7. What permissions are needed to modify retention settings?
You need at least *Project Administrator* or *Build Administrator* role to modify global retention settings or mark runs for retention.

8. How do retention settings affect deployment history and auditability?
Deleting artifacts or logs too early can break traceability and hinder auditing of what was deployed, when, and from which source.

9. How can you override global retention policies per pipeline?

In YAML, set retention under the pipeline root, e.g.:

```
retention: { daysToKeep: 10, minimumToKeep: 1, deleteBuildRecord: true }
```

11. **What are best practices for implementing artifact retention strategies?**
    - Use shorter retention for high-frequency pipelines.
    - Retain longer for production deployments.
    - Mark important releases as retained.
    - Periodically audit unused runs.