

2. Source control strategy

└ 2.3 Package Management

└ 2.3.4 Version Pipeline Artifacts

1. What are pipeline artifacts in Azure DevOps and GitHub Actions?
 2. Why is it important to version pipeline artifacts?
 3. How do you set a version for a pipeline artifact in Azure DevOps?
 4. How do you version artifacts in GitHub Actions?
 5. What strategies can you use for artifact versioning (e.g., SemVer, build number, commit SHA)?
 6. How do you ensure traceability between artifacts and source code versions?
 7. How can you automate artifact versioning in a CI/CD pipeline?
 8. How do you store and retrieve versioned artifacts in Azure Pipelines?
 9. How do you promote or consume specific versions of pipeline artifacts?
 10. What are best practices for artifact retention and cleanup?
-

1. What are pipeline artifacts in Azure DevOps and GitHub Actions?

Pipeline artifacts are files or packages produced by CI/CD builds and stored for use in later stages (e.g., deployment, testing) or for sharing across jobs or pipelines.

2. Why is it important to version pipeline artifacts?

Versioning ensures traceability, reproducibility, and prevents conflicts or accidental overwrites. It allows teams to identify which build or commit produced a given artifact.

3. How do you set a version for a pipeline artifact in Azure DevOps?

Define a variable (e.g., \$(Build.BuildNumber) or custom SemVer/CalVer) and use it as the artifact's name or version in PublishPipelineArtifact Or PublishBuildArtifacts tasks.

4. How do you version artifacts in GitHub Actions?

Set a version tag using

- environment variables,
- Git commit SHA,
- or tags,

and include it in the artifact name when using the actions/upload-artifact action.

5. What strategies can you use for artifact versioning (e.g., SemVer, build number, commit SHA)?

Common strategies:

- SemVer: Aligns with application/package version
 - Build number: Unique CI/CD pipeline run
 - Commit SHA: Direct link to source version
 - CalVer: Date-based versioning
-

6. How do you ensure traceability between artifacts and source code versions?

Include commit SHA, build number, or version info in the artifact metadata or name. Store artifact metadata with references to build and source details.

7. How can you automate artifact versioning in a CI/CD pipeline?

Use

- pipeline variables,
- scripts,
- or tasks

to generate and assign a version to artifacts automatically during build or release.

8. How do you store and retrieve versioned artifacts in Azure Pipelines?

Artifacts are published to *Azure DevOps* (*Artifacts > Pipeline Artifacts/Build Artifacts*). Use *DownloadPipelineArtifact* OR *DownloadBuildArtifacts* tasks with the specified version or build number.

9. How do you promote or consume specific versions of pipeline artifacts?

Reference the desired artifact version in pipeline tasks or release stages using

- tags,
- build numbers,
- or metadata.

Promotion may involve moving artifacts to a separate feed or environment.

10. What are best practices for artifact retention and cleanup?

Configure retention policies

- to automatically delete old or unused artifacts,
- keep only recent or released versions,
- and prevent storage bloat.
- Document retention settings for compliance.