

1. Design and implement processes and communications

└ 1.3 Configure collaboration and communication

└ 1.3.2 Configure Release Documentation

1. What is release documentation, and why is it important in DevOps?
 2. How do you generate release notes automatically in Azure DevOps?
 3. What information should be included in release notes?
 4. How can you document API changes as part of a release process?
 5. How do you configure and publish release documentation using CI/CD pipelines?
 6. What are best practices for maintaining accurate release documentation?
 7. How do you manage versioning in release documentation?
 8. How can you automate API documentation generation and integration into release notes?
 9. How do you provide access to release documentation for different stakeholders?
 10. What tools are commonly used for release and API documentation in Azure DevOps and GitHub?
-

1. What is release documentation, and why is it important in DevOps?

Release documentation summarizes

- the changes,
- features,
- fixes, and known issues

included in a deployment. It provides transparency, helps stakeholders understand what's delivered, and supports

- troubleshooting,
 - compliance,
 - and audit requirements.
-

2. How do you generate release notes automatically in Azure DevOps?

Use the *Azure DevOps* Release Notes extension or pipeline tasks to extract information from work items, commits, and pull requests. Configure templates to format notes and publish them as part of the release pipeline.

3. What information should be included in release notes?

Include

- new features,
 - bug fixes,
 - improvements,
 - breaking changes,
 - known issues,
 - deployment steps,
 - and references to related work items or tickets.
-

4. How can you document API changes as part of a release process?

Integrate API documentation tools like *Swagger* or *OpenAPI* in the CI/CD pipeline to generate and update API reference docs. Link or embed updated API docs in the release notes for visibility.

5. How do you configure and publish release documentation using CI/CD pipelines?

Add tasks in the pipeline to generate documentation artifacts, such as markdown or HTML files, then publish them to a wiki, repository, or shared location. Use pipeline variables to version and label documentation with each release.

6. What are best practices for maintaining accurate release documentation?

- Update documentation as part of each deployment
 - Review release notes during pull requests
 - Automate as much as possible
 - Assign ownership for validation
 - Ensure docs reflect actual release content
-

7. How do you manage versioning in release documentation?

Organize documentation by release version (e.g., folders or tags), include version numbers in doc titles, and maintain a changelog or index for easy navigation across versions.

8. How can you automate API documentation generation and integration into release notes?

Use tools like Swagger/OpenAPI with build tasks to generate API docs on each build. Automatically attach or link these docs in release notes through pipeline scripts or templates.

9. How do you provide access to release documentation for different stakeholders?

- Publish docs to accessible platforms such as project wikis, shared repos, or internal portals.
 - Set permissions to control access based on roles (e.g., read-only or writing).
-

10. What tools are commonly used for release and API documentation in Azure DevOps and GitHub?

For release documentation:

- Azure DevOps Wiki
- Release Notes Extensions

For API documentation:

- Swagger
- OpenAPI
- DocFX

And Markdown rendered in wikis or static site generators.