1. **Symptoms**

The taxonomy of Docker build fault (DBF) symptoms is organized into 5 inner categories and

15 leaf categories:

1. **Initialization failure**：At the start of executing build, Docker daemon needs to collect required configuration and context files and initialize the necessary environments. Failure in this stage are called Initialization failure. This inner category includes 3 leaf categories:
   1. Configuration resolution failure
   2. Context loading error
   3. Environment compatibility failure
2. **Build failure:** It means the execution functionality of Docker build is terminated and the Docker build exits irregularly mid-execution by throwing an error. This inner category includes 4 leaf categories:
   1. Command execution failure
   2. Dependency resolution error
   3. Access control issue
   4. Others
3. **Unexpected process:** Unexpected process refers to the cases where the process of a certain build task is not as expected although no error is triggered during that build task. This inner category includes 3 leaf categories:
   1. Wrong artifact used
   2. Step/artifact ignored
   3. Step stuck
4. **Incorrect output:** Some build faults do not trigger build failures explicitly during the build process, but generate problematic outputs (e.g., file/log missing). This inner category includes 3 leaf categories:
   1. File missing
   2. Unexpected result
   3. No log/output
5. **Poor performance:** This is another typical symptom for DBFs, which can slow down the execution processes and consume excessive resources. This inner category includes 2 leaf categories:
   1. Time issue
   2. Memory issue
6. **Root causes**

The taxonomy of Docker build fault (DBF) root causes is organized into 7 inner categories and 23 leaf categories:

1. **Configuration error:** Thosefaults are caused by the incorrect configuration. This inner category includes 4 leaf categories:
2. Invalid build argument
3. Invalid Dockerfile
4. Invalid compose file
5. Tool/plugin setting error
6. **Dependency error:** Those faults are caused by dependency problems. This inner category includes 4 leaf categories:
7. Findability error
8. Download error
9. Unsatisfiable constraints
10. Other dependency error
11. **Environment error:** Those faults are caused by imperfect support of Docker build execution for some hardware/software environments. This inner category includes 4 leaf categories:
12. Cache issue
13. Invalid Docker daemon
14. Platform/runtime incompatibility
15. Insufficient memory
16. **Execution error:** This category covers faults caused by specific execution errors. This inner category includes 4 leaf categories:
17. Compilation error
18. Parse error
19. Interaction interruption
20. Connection exception
21. **Resource error:** Those faults are caused by resource. This inner category includes 4 leaf categories:
22. File inaccessibility error
23. Directory/path error
24. Invalid context
25. Redundant files
26. **Security error:** Those faults are caused by security. This inner category includes 3 leaf categories:
27. Permission error
28. Authentication error
29. Other security error
30. **Others:** Those faults that are not belong to the above root cause categories, and are therefore classified as Others.
31. **Fix patterns**

We identify 35 common fix patterns, which involve 6 major objects:

1. **Environment:** Those faults are fixed via resolving the environment errors, including 9 fix patterns:

* Change base image
* Change container architecture
* Modify network setting
* Upgrade Docker version
* Change OS environment
* Increase memory
* Restart Docker daemon
* Restart build operation
* Reinstall runtime

1. **Dependency:** Those faults are fixed by modifying the dependency, including 5 fix patterns:

* Upgrade dependency version
* Add missing dependency
* Downgrade dependency version
* Resolve upstream error
* Use compatible dependency

1. **Configuration**: Those faults are fixed by modifying the configuration, including 7 fix patterns:

* Modify build argument
* Use suitable command
* Change command parameter
* Add build argument
* Adjust command format
* Modify code/script logic
* Remove improper command/step

1. Resource: Those faults are resolved by modifying the resource setting, including 7 fix patterns:

* Change path/location
* Set correct working directory
* Add missing resource
* Delete redundant files
* Use .dockerignore
* Change file name
* Mount correct resource

1. Tool/plugin: Those faults are resolved by fixing tool/plugin related errors, including 2 fix patterns:

* Modify tool/plugin setting
* Use suitable tool/plugin

1. Security: Those faults are resolved by fixing security issues, including 5 fix patterns:

* Modify authentication setting
* Add ssh setting
* Change user permission
* Add missing permission
* Change file/directory permission