Devops Shack

100 DOCKER ERRORS AND TROUBLESHOOTING GUIDE

www.devopsshack.com

2024
2025
EBOOK

DevOps Shack

100 Docker Errors and Troubleshooting Guide

Table of Contents

Docker Daemon and Service Issues

Container Management Errors

18. Cannot remove container: device or resource busy
19. Error response from daemon: endpoint with name already exists

---

Image Management Errors

20. Unable to find image 'IMAGE:TAG' locally
21. Image pull rate limit exceeded
22. Error: invalid reference format
23. Manifest for IMAGE:TAG not found
24. Error response from daemon: failed to create endpoint
25. Cannot delete image: conflict
26. Docker push fails with denied: requested access to the resource is denied
27. Error response from daemon: unsupported: manifest unknown
28. Error: no matching manifest for architecture in the manifest list entries
29. Error response from daemon: name unknown

Volume and File Issues

30. Mounts denied: The path does not exist
31. Error while creating mount source path
32. Mounts denied: Path outside sandbox (macOS)
33. Error: failed to create endpoint on network bridge
34. Cannot execute command in container: no such file or directory
35. Docker-compose: invalid mode for volume
36. Error response from daemon: read-only file system
37. Cannot remove volume: volume is in use
38. Docker-compose: cannot locate volume
39. Docker-compose: error parsing volume specification

Networking Issues

40. Ports are already in use
41. Error: network not found
42. Error response from daemon: failed to attach network
43. Docker-compose: error creating overlay network: invalid subnet
44. Error response from daemon: client version is newer than server version
45. Cannot connect to container's exposed port
46. Error: Docker container cannot resolve DNS
47. Error response from daemon: network name must be a string
48. Docker-compose: service depends on condition not met
49. Error: Docker-compose: cannot start service due to network conflicts

---

Build and Dockerfile Errors

50. OCI runtime create failed
51. Error response from daemon: failed to solve with frontend dockerfile.v0
52. Error: unable to locate package in Docker build
53. Error: ADD failed: file not found in build context
54. Dockerfile parse error: unknown instruction
55. Docker build fails with 'stat /var/lib/docker/tmp: no such file or directory'
56. Docker build fails: failed to solve
57. Docker build fails with 'file exists'
58. Docker-compose: service build context must be a directory
59. Docker-compose: build context must be a mapping

Configuration and YAML Syntax Issues

60. Docker-compose up fails with 'service not found'

61. Docker-compose: unsupported configuration option for services
62. Docker-compose: invalid key in service definition
63. Docker-compose: cannot start service build
64. Docker-compose up fails with 'service name not valid'
65. Docker-compose: YAML parsing errors
66. Docker-compose: missing or invalid configuration key
67. Docker-compose: cannot merge networks: network name must be a string
68. Docker-compose: invalid restart policy
69. Docker-compose: error while merging services

Log, Resource, and Debugging Errors

70. Docker logs not showing
71. Error response from daemon: no space left on device
72. Error: too many open files
73. Docker-compose: service fails to start with 'exec format error'
74. Docker logs show 'Out of memory'
75. Docker-compose: logs fail with 'error accessing files'
76. Debugging Docker container failing to start
77. Error: health check failing
78. Error starting userland proxy: Bind for 0.0.0.0:PORT failed
79. Cannot resolve Docker DNS while using custom resolvers
80. Docker-compose: cannot attach to logs

Registry and Authentication Issues

81. Docker push fails with unauthorized
82. Docker pull unauthorized: authentication required

Introduction

Docker is a powerful containerization platform that simplifies the deployment and management of applications. However, like any technology, it comes with its own set of challenges. From setup issues to runtime errors, users may encounter various problems while working with Docker. These errors can stem from configuration mistakes, network conflicts, resource limitations, or compatibility issues.

This comprehensive guide categorizes and explains 100 common Docker errors, along with practical troubleshooting steps for each. It is designed to help developers, DevOps engineers, and system administrators quickly diagnose and resolve issues, whether they arise during development, building, deployment, or runtime.

Purpose of the Guide

1. Ease of Troubleshooting: Provides clear steps to fix common errors.
2. Categorized for Clarity: Errors are grouped into logical sections such as Daemon Issues, Container Management, Networking, Build Errors, and Registry Authentication.
3. Reference for Learning: Acts as a resource for understanding common pitfalls and how to avoid them.

Whether you're a beginner or an experienced professional, this guide will help you navigate through Docker challenges efficiently, minimizing downtime and ensuring smooth operations.

1. Error: "Cannot connect to the Docker daemon"

Cause: Docker daemon is not running or the user does not have the necessary permissions.

Troubleshooting:

- Check if the Docker daemon is running: sudo systemctl status docker.
- Start Docker if it’s not running: sudo systemctl start docker.
- Ensure the user is added to the Docker group: sudo usermod -aG docker $USER (then log out and back in).

2. Error: "Error response from daemon: Conflict. The container name is already in use"

Cause: A container with the same name already exists.

Troubleshooting:

- List existing containers: docker ps -a.
- Remove or rename the existing container: docker rm <container_name> or use --name with a unique name when running a new container.

3. Error: "No space left on device"

Cause: Disk space is full due to unused images, containers, or volumes.

Troubleshooting:

- Remove unused containers: docker container prune.
- Remove unused images: docker image prune.
- Remove unused volumes: docker volume prune.
- Free up space: docker system prune -a.

---

4. Error: "permission denied while trying to connect to the Docker daemon socket"

Cause: User does not have permission to access the Docker socket.

Troubleshooting:

- Run the command with sudo (temporary fix).
- Add the user to the Docker group: sudo usermod -aG docker $USER (requires re-login).

5. Error: "OCI runtime create failed"

Cause: Invalid or unsupported runtime configuration.

Troubleshooting:

- Check the container's runtime configuration for errors in the docker run command.
- If using a specific runtime, ensure it's installed and configured: docker info | grep Runtimes.
- Inspect logs: journalctl -u docker.service or docker logs <container_id>.

6. Error: "Unable to find image 'IMAGE:TAG' locally"

Cause: The specified image is not available locally or in the Docker registry.

Troubleshooting:

- Verify the image name and tag are correct.
- Pull the image manually: docker pull IMAGE:TAG.
- Check the Docker Hub or private registry for image availability.

7. Error: "Ports are already in use"

Cause: The port specified in docker run -p is being used by another container or application.

Troubleshooting:

- Check which process is using the port: sudo netstat -tuln | grep <port_number>.
- Stop the conflicting process or container.
- Use a different port for the container.

8. Error: "Mounts denied: The path does not exist"

Cause: Trying to bind a local path that doesn't exist to a container.

Troubleshooting:

- Ensure the specified path exists on the host system.
- Use absolute paths for bind mounts: docker run -v /absolute/path:/container/path.

9. Error: "Image pull rate limit exceeded"

Cause: Docker Hub has rate limits for anonymous and free-tier users.

Troubleshooting:

- Log in to Docker Hub: docker login.
- Use a Docker Pro or Team account to increase the rate limit.
- Mirror the required images to a private registry.

10. Error: "Container is restarting repeatedly"

Cause: The container encounters an error and exits immediately.

Troubleshooting:

- Check logs: docker logs <container_id>.
- Debug entrypoint or command: Use an interactive shell: docker run -it --entrypoint /bin/bash IMAGE.
- Verify dependencies, environment variables, and configurations.

Here are the next 10 common Docker errors and their troubleshooting steps:

11. Error: "Exec user process caused: no such file or directory"

Cause: The script or binary being executed is missing or has incorrect line endings (e.g., Windows CRLF).

Troubleshooting:

- Check if the script or file exists in the specified path inside the container.
- Ensure the file has executable permissions: chmod +x <filename>.
- Convert line endings to Unix format: dos2unix <filename

12. Error: "Docker network not found"

Cause: The specified network in docker run or docker-compose does not exist.

Troubleshooting:

- List available networks: docker network ls.
- Create the network if it doesn’t exist: docker network create <network_name>.
- Ensure the correct network name is used in docker-compose.yml.

13. Error: "Manifest for IMAGE:TAG not found"

Cause: The specified image tag does not exist in the registry.

Troubleshooting:

- Verify the image and tag exist in the Docker registry (e.g., Docker Hub).
- Use docker pull <image> to list available tags.
- Specify an existing tag or the default latest tag.

---

14. Error: "Mount denied: Path is outside the build context"

Cause: Attempting to copy or mount a file outside the Docker build context.

Troubleshooting:

- Ensure all files being referenced are within the build context directory.
- Move files into the build context directory or specify correct paths.
- Use absolute paths in volume bindings for docker run.

15. Error: "Cannot kill container: No such container"

Cause: The container ID or name is incorrect, or the container no longer exists.

Troubleshooting:

- Verify the container exists: docker ps -a.
- Use the correct container ID or name in the command: docker kill <container_id>.
- If removed, no action is needed.

16. Error: "Image build fails with 'COPY failed: no source files were specified'"

Cause: The specified file or directory in the COPY instruction does not exist in the build context.

Troubleshooting:

- Verify the file path relative to the Dockerfile.
- Ensure files are within the build context directory.
- Double-check the syntax of the COPY instruction in the Dockerfile.

17. Error: "Container is unhealthy"

Cause: The container's health check command is failing.

Troubleshooting:

- Check the health status: docker inspect --format='{{json .State.Health}}' <container_id>.
- Verify the health check command in the Dockerfile.
- Adjust health check parameters (interval, timeout, retries, etc.).

18. Error: "Failed to start container: Address already in use"

Cause: The container is trying to bind a port that is already in use by another application or container.

Troubleshooting:

- Identify which process is using the port: sudo netstat -tuln | grep <port_number>.
- Stop the conflicting container or process.
- Use a different port in the docker run command.

19. Error: "Docker pull unauthorized: authentication required"

Cause: Trying to pull a private image without being authenticated.

Troubleshooting:

- Log in to the registry: docker login.
- Ensure you have permissions for the private repository.
- Verify the image and repository name are correct.

20. Error: "Error response from daemon: client is newer than server"

Cause: The Docker client version is incompatible with the Docker daemon version.

Troubleshooting:

- Update the Docker daemon to match the client version.
- Alternatively, downgrade the Docker client to match the daemon version.
- Use the DOCKER_API_VERSION environment variable to specify a compatible API version.

21. Error: "Dockerfile parse error: unknown instruction"

Cause: A syntax or formatting error in the Dockerfile, or an unrecognized instruction.

Troubleshooting:

- Verify the instruction syntax is correct and supported by your Docker version.
- Ensure there are no typos (e.g., COPY instead of CPY).
- Check for unexpected whitespace or characters in the Dockerfile.

22. Error: "Mounts denied: Path outside sandbox" on macOS

Cause: The host path is not shared with Docker Desktop.

Troubleshooting:

- Go to Docker Desktop > Preferences > Resources > File Sharing.
- Add the host path to the shared directories.
- Restart Docker Desktop after making changes.

23. Error: "EACCES: permission denied, mkdir"

Cause: The container process does not have permission to create directories or files.

Troubleshooting:

- Ensure the directory has appropriate permissions: chmod -R 777 <directory>.
- Run the container with a specific user: docker run -u $(id -u):$(id -g).
- Use USER instruction in the Dockerfile to set the correct user.

24. Error: "Docker: invalid reference format"

Cause: The image name or tag provided is not in a valid format.

Troubleshooting:

- Ensure the image name follows the correct format: [registry/]repository[:tag].
- Avoid using spaces, uppercase letters, or invalid characters in the image name or tag.
- Example of a valid name: myrepo/myimage:1.0.

25. Error: "Error response from daemon: pull access denied for IMAGE, repository does not exist"

Cause: The image or repository name is incorrect, or you lack access permissions.

Troubleshooting:

- Verify the image name and tag exist in the registry.

- Log in to the private registry: docker login.
- Ensure you have the correct permissions to access the repository.

26. Error: "Error starting userland proxy: Bind for 0.0.0.0:PORT failed"

Cause: The specified port is already in use on the host.

Troubleshooting:

- Identify the process using the port: sudo netstat -tuln | grep <port_number>.
- Stop the conflicting process or application.
- Use a different port for the container.

27. Error: "Container not found"

Cause: The specified container ID or name does not exist.

Troubleshooting:

- List all containers: docker ps -a.
- Double-check the container ID or name in the command.
- If the container has been removed, no further action is needed.

28. Error: "Error response from daemon: failed to create endpoint"

Cause: The container could not connect to the network.

Troubleshooting:

- Check the Docker network: docker network ls.
- Recreate the network if necessary: docker network create <network_name>.
- Ensure the container is being started with the correct network settings.

29. Error: "Docker: Cannot locate specified Dockerfile"

Cause: The Dockerfile is not in the build context or is misnamed.

Troubleshooting:

- Ensure the Dockerfile is in the directory from which you are running docker build.
- Use the -f flag to specify the Dockerfile location: docker build -f /path/to/Dockerfile ..

30. Error: "Connection reset by peer" when pulling images

Cause: Network issues or Docker Hub rate-limiting.

Troubleshooting:

- Check your internet connection and DNS settings.
- Use a VPN if you are in a restricted network environment.
- Log in to Docker Hub to avoid anonymous pull rate limits: docker login.

31. Error: "Cannot start service: Mounts denied"

Cause: The specified volume or bind mount path does not exist or is inaccessible.

Troubleshooting:

- Verify the host path exists and has appropriate permissions.
- Ensure the path is shared in Docker Desktop (on macOS/Windows).
- Check the syntax of the -v or --mount option in the docker run command.

32. Error: "Error response from daemon: driver failed programming external connectivity"

Cause: Port mapping or network configuration failed.

Troubleshooting:

- Check if the port is already in use: sudo netstat -tuln | grep <port_number>.
- Restart Docker to reset network configurations: sudo systemctl restart docker.
- Use a different port for mapping in the docker run command.

33. Error: "Docker build fails: failed to solve with frontend dockerfile.v0"

Cause: Dockerfile syntax errors or issues with build arguments.

Troubleshooting:

- Validate the Dockerfile syntax and ensure all required files exist in the build context.
- Pass missing build arguments explicitly: docker build --build-arg ARG_NAME=value ..
- Upgrade Docker to the latest version to ensure compatibility.

34. Error: "Error: unknown filesystem type on layer"

Cause: A corrupted or unsupported image layer.

Troubleshooting:

- Remove the problematic image: docker rmi IMAGE_ID.
- Pull the image again: docker pull IMAGE_NAME.
- If the issue persists, try an alternative image tag or build from the source.

---

35. Error: "Error while creating mount source path"

Cause: Invalid or inaccessible path provided for a volume mount.

Troubleshooting:

- Ensure the mount source directory exists on the host.
- Use an absolute path in the -v or --mount option.
- Verify permissions of the directory: chmod -R 755 /path/to/source.

36. Error: "Error initializing graphdriver"

Cause: Issues with the Docker storage driver.

Troubleshooting:

- Verify the configured storage driver in daemon.json matches the system: docker info | grep Storage.
- Check disk space and file system compatibility.
- Restart Docker and, if needed, reset the configuration.

37. Error: "Docker daemon out of memory"

Cause: The Docker daemon has insufficient memory to run containers.

Troubleshooting:

- Increase memory limits for Docker Desktop (if using macOS/Windows).
- Limit container memory usage: docker run --memory=512m IMAGE_NAME.
- Check and stop unused containers to free resources.

38. Error: "Error loading shared libraries: No such file or directory"

Cause: Missing dependencies inside the container.

Troubleshooting:

- Verify the required libraries are installed in the Docker image.
- Add installation steps to the Dockerfile using package managers (e.g., apt, yum).
- Use a base image with the required libraries pre-installed.

39. Error: "Docker-compose up fails with 'invalid container port'"

Cause: Incorrect port mapping in docker-compose.yml.

Troubleshooting:

- Verify the port mapping format: <host_port>:<container_port>.
- Ensure both ports are integers and valid.
- Update the docker-compose.yml file with the correct port values.

40. Error: "Docker daemon is not running"

Cause: The Docker daemon service is stopped or crashed.

Troubleshooting:

- Start the Docker daemon: sudo systemctl start docker.
- Check its status: sudo systemctl status docker.
- Inspect logs for issues: journalctl -u docker.service.

41. Error: "Error response from daemon: unsupported: manifest unknown"

Cause: The specified image tag or architecture is not supported in the registry.

Troubleshooting:

- Verify the image tag and architecture compatibility in the Docker registry.
- Use a supported image tag or pull the default latest tag.
- If using a multi-architecture image, ensure your Docker CLI is set up for the correct architecture.

---

42. Error: "Error: Unable to locate package in Docker build"

Cause: Missing or outdated package repositories in the base image.

Troubleshooting:

- Add RUN apt-get update or equivalent command before installing packages in the Dockerfile.
- Ensure the base image supports the desired package manager (e.g., apt, yum).

43. Error: "No such container: <container_name>"

Cause: The specified container does not exist.

Troubleshooting:

- List all containers: docker ps -a.
- Use the correct container name or ID.
- Ensure the container hasn't been removed before running the command.

44. Error: "Docker-compose: services.<service_name>.build must be a mapping"

Cause: Syntax error in the docker-compose.yml file.

Troubleshooting:

- Verify the syntax for the build section:

```
build:
 context: ./path_to_build
 dockerfile: Dockerfile
```

- Use a YAML linter to validate the file structure.

45. Error: "Cannot delete image: conflict"

Cause: The image is being used by a running or stopped container.

Troubleshooting:

- Stop and remove containers using the image:

```
docker ps -a | grep <image_name>
docker rm -f <container_id>
```

- Force remove the image: docker rmi -f <image_id>.

46. Error: "Docker-compose: network not found"

Cause: The specified network does not exist or is not created.

Troubleshooting:

- Ensure the network is defined in the docker-compose.yml file.
- Create the network manually: docker network create <network_name>.
- Restart the docker-compose stack: docker-compose up -d.

47. Error: "Docker push fails with denied: requested access to the resource is denied"

Cause: You do not have permission to push the image to the registry.

Troubleshooting:

- Log in to the registry: docker login.
- Ensure you have write permissions to the repository.
- Tag the image correctly before pushing:

```
docker tag local_image registry/repository:tag
docker push registry/repository:tag
```

48. Error: "Error response from daemon: invalid restart policy"

Cause: Invalid or unsupported restart policy in the docker run or docker-compose command.

Troubleshooting:

- Use a valid restart policy:
  - no
  - always
  - on-failure[:max-retries]
  - unless-stopped
- Example:

```
docker run --restart always <image_name>
```

49. Error: "Cannot stop container: container already stopped"

Cause: The container is not running but the stop command was issued.

Troubleshooting:

- Use docker ps -a to confirm the container status.
- Remove the container if it's no longer needed: docker rm <container_id>.

50. Error: "Error response from daemon: manifest blob unknown"

Cause: The specified layer in the image is unavailable in the registry.

Troubleshooting:

- Pull the image again: docker pull <image_name>.
- Verify the image tag and architecture compatibility in the Docker registry.
- If using a private registry, ensure the image has been correctly pushed.

51. Error: "Docker-compose: unsupported configuration option for services"

Cause: Typo or invalid configuration in the docker-compose.yml file.

Troubleshooting:

- Verify the docker-compose.yml file syntax against the Docker Compose version.
- Use a YAML linter to check for formatting issues.
- Confirm that the specified option is supported for the corresponding service or Docker Compose version.

52. Error: "Error response from daemon: container has non-zero exit code"

Cause: The container terminated with an error during execution.

Troubleshooting:

- Check the container logs: docker logs <container_id>.
- Debug the error message in the logs to identify the root cause.
- Run the container interactively for testing: docker run -it <image_name> /bin/bash.

53. Error: "Error: failed to initialize logging driver"

Cause: The logging driver specified is unavailable or misconfigured.

Troubleshooting:

- Verify the logging driver in use: docker info | grep Logging.
- Use a valid logging driver: json-file, syslog, journald, etc.
- Specify the logging driver in daemon.json or the docker run command:

```
{
  "log-driver": "json-file"
}
```

54. Error: "Invalid reference format: repository name must be lowercase"

Cause: The image name contains uppercase letters or invalid characters.

Troubleshooting:

- Rename the image using only lowercase letters and valid characters:

```
docker tag MyImage:1.0 myimage:1.0
```

- Follow the correct format for naming: [registry/]repository[:tag].

55. Error: "Error: failed to create endpoint on network bridge"

Cause: A network conflict or misconfiguration in the default bridge network.

Troubleshooting:

- Restart Docker to reset the network stack: sudo systemctl restart docker.
- Inspect the bridge network: docker network inspect bridge.

- Recreate the bridge network if needed: docker network rm bridge && docker network create bridge.

---

56. Error: "Docker daemon: cgroups: cannot find cgroup mount destination"

Cause: The cgroups file system is not mounted or configured properly.

Troubleshooting:

- Check if cgroups are mounted: cat /proc/mounts | grep cgroup.
- Ensure your Linux distribution supports cgroups and they are enabled in the kernel.
- Restart Docker after fixing the cgroup configuration: sudo systemctl restart docker.

57. Error: "No matching manifest for architecture in the manifest list entries"

Cause: The image does not support the current system architecture.

Troubleshooting:

- Use an image compatible with your architecture (e.g., arm64, amd64).
- Check available architectures for the image: docker manifest inspect <image_name>.
- Use a multi-architecture image if possible.

58. Error: "Cannot remove volume: volume is in use"

Cause: The volume is being used by a running or stopped container.

Troubleshooting:

- Identify the container using the volume: docker ps -a --filter volume=<volume_name>.
- Stop and remove the container using the volume:

```
docker stop <container_id>
```

```
docker rm <container_id>
```

- Remove the volume: docker volume rm <volume_name>.

59. Error: "Docker image export/import fails with 'permission denied'"

Cause: Insufficient permissions to access the export or import destination.

Troubleshooting:

- Use sudo when exporting or importing images:

```
sudo docker save -o image.tar <image_name>
sudo docker load -i image.tar
```

- Verify write permissions for the specified path.

60. Error: "Error response from daemon: device or resource busy"

Cause: The container or volume is actively being used by another process.

Troubleshooting:

- Stop the container using the resource: docker stop <container_id>.
- Use lsof to identify the process using the resource:

```
lsof | grep <resource_name>
```

- Force the removal if necessary: docker rm -f <container_id>.

61. Error: "Error response from daemon: client version is newer than server version"

Cause: The Docker client version is incompatible with the Docker server version.

Troubleshooting:

- Downgrade the Docker client to match the server version.
- Alternatively, set an environment variable to specify an older API version:

```
export DOCKER_API_VERSION=1.xx
```

- Upgrade the Docker server to match the client version.

62. Error: "Cannot remove image: image is being used by stopped containers"

Cause: The image is still referenced by one or more stopped containers.

Troubleshooting:

- List containers using the image: docker ps -a --filter ancestor=<image_name>.
- Remove the stopped containers: docker rm <container_id>.
- Force-remove the image if necessary: docker rmi -f <image_name>.

63. Error: "Error: network <network_name> is already used"

Cause: A network with the same name already exists and conflicts with a new setup.

Troubleshooting:

- List existing networks: docker network ls.
- Remove the conflicting network: docker network rm <network_name>.
- Recreate the network with a different name if necessary.

64. Error: "Docker-compose up fails with 'service not found'"

Cause: The specified service in docker-compose.yml does not exist or has a typo.

Troubleshooting:

- Verify the service name in docker-compose.yml.
- Ensure proper indentation and correct YAML syntax.
- Use docker-compose config to validate the configuration file.

65. Error: "Container is running but cannot connect to the service"

Cause: Misconfigured ports, firewalls, or network settings.

Troubleshooting:

- Ensure the container is exposing the correct ports: docker inspect <container_id>.
- Confirm the host port is accessible: netstat -tuln | grep <port_number>.
- Check firewall rules and allow the necessary ports.

66. Error: "Error creating overlay network: invalid subnet"

Cause: The specified subnet is invalid or conflicts with an existing network.

Troubleshooting:

- Use a valid and unique subnet for the overlay network.
- List existing networks to identify conflicts: docker network ls.
- Create the overlay network with a unique subnet:

```
docker network create --driver overlay --subnet=<subnet> <network_name>
```

67. Error: "Image build fails with 'ADD failed: file not found in build context'"

Cause: The file specified in the ADD or COPY instruction is missing or outside the build context.

Troubleshooting:

- Ensure the file exists relative to the Dockerfile’s build context.
- Use absolute paths for the build context when running docker build:

```
docker build -f Dockerfile .
```

- Verify the file name and case sensitivity.

68. Error: "Docker daemon: too many open files"

Cause: The system has hit the limit for open files, often due to too many active containers or processes.

Troubleshooting:

- Increase the open files limit for Docker:

```
ulimit -n 65535
```

- Update the Docker daemon configuration:

```
{
 "default-ulimits": {
  "nofile": {
   "Name": "nofile",
   "Hard": 65535,
   "Soft": 65535
  }
 }
}
```

- Restart Docker after making changes: sudo systemctl restart docker.

69. Error: "Error creating container: volume name is too short or missing"

Cause: The volume name is not specified or too short.

Troubleshooting:

- Use a valid volume name when creating or referencing a volume:

```
docker volume create <volume_name>
```

- Ensure the volume name in the docker-compose.yml file is properly defined.

70. Error: "Docker-compose: invalid mode for volume"

Cause: Incorrect or unsupported volume mount mode specified in docker-compose.yml.

Troubleshooting:

- Use valid mount modes like ro (read-only) or rw (read-write).
- Verify the syntax in the volumes section of docker-compose.yml:

```
volumes:
 - ./host_path:/container_path:rw
```

- Validate the file using docker-compose config.

71. Error: "Docker-compose: error while merging networks: network name must be a string"

Cause: Incorrect syntax or data type for network names in docker-compose.yml.

Troubleshooting:

- Ensure network names are defined as strings:

```
networks:
  my_network:
    driver: bridge
```

- Validate the docker-compose.yml file with docker-compose config.

72. Error: "Error response from daemon: cannot link to a non-running container"

Cause: A linked container is stopped or not running.

Troubleshooting:

- Start the linked container: docker start <container_name>.
- Check if the link is necessary; consider using Docker networks instead of --link, as it is deprecated.

73. Error: "Cannot kill container: permission denied"

Cause: Insufficient permissions to manage the container.

Troubleshooting:

- Run the command with sudo:

```
sudo docker kill <container_id>
```

- Add the current user to the Docker group: sudo usermod -aG docker $USER.

74. Error: "Error response from daemon: read-only file system"

Cause: The container is trying to write to a read-only file system.

Troubleshooting:

- Verify if the container is running in read-only mode.
- Use a writable volume for the required directory:

```
docker run -v /writable/path:/container/path:rw <image_name>
```

- Modify the Dockerfile to set proper write permissions for required directories.

75. Error: "Service fails to start with 'exec format error'"

Cause: The image architecture does not match the host system's architecture.

Troubleshooting:

- Verify the image architecture using docker inspect or docker manifest inspect.
- Use an image compatible with the host system's architecture (e.g., amd64, arm64).
- If using a multi-arch environment, enable experimental Docker CLI features to pull architecture-specific images.

76. Error: "Docker-compose: error parsing volume specification"

Cause: Invalid volume syntax in the docker-compose.yml file.

Troubleshooting:

- Ensure the volume syntax follows the correct format:

```
volumes:
 - /host_path:/container_path:mode
```

- Validate the docker-compose.yml file: docker-compose config.

77. Error: "Error response from daemon: failed to mount overlay: no space left on device"

Cause: The disk or inode usage on the host has reached its limit.

Troubleshooting:

- Free up disk space:

```
docker system prune -a
```

- Check inode usage:

```
df -i
```

- Remove unused volumes, images, and containers.

78. Error: "Docker logs not showing"

Cause: Log rotation or a misconfigured logging driver.

Troubleshooting:

- Check if the logs are being rotated:

```
docker inspect <container_id> | grep LogPath
```

- Verify the logging driver: docker info | grep Logging.
- Reconfigure the logging driver in daemon.json if needed.

79. Error: "Cannot connect to container's exposed port"

Cause: The port is not properly mapped or a firewall is blocking access.

Troubleshooting:

- Verify port mapping with docker ps.

- Use -p to map the container port to the host port when running the container:

```
docker run -p 8080:80 <image_name>
```

- Check firewall rules and ensure the required port is open.

80. Error: "Docker push fails with 'error: name unknown'"

Cause: The repository name or tag is incorrect or the repository does not exist in the registry.

Troubleshooting:

- Verify the repository name and tag format: [registry/]repository[:tag].
- Ensure the repository exists in the target registry.
- Authenticate to the registry using docker login.

81. Error: "Error response from daemon: conflict: unable to delete repository reference"

Cause: The image is still referenced by a tag or another container.

Troubleshooting:

- List all references to the image: docker images --digests.
- Remove all containers or tags referencing the image:

```
docker rm <container_id>
docker rmi <image_name>:<tag>
```

- Use docker rmi -f <image_id> to force deletion if necessary.

82. Error: "Error: unable to prepare context: path not found"

Cause: The build context directory specified does not exist.

Troubleshooting:

- Verify the path to the build context directory:

```
docker build -f Dockerfile /correct/context/path
```

- Ensure the Dockerfile and all required files are in the specified context.

83. Error: "Docker container exits immediately"

Cause: The container's main process terminates, causing the container to exit.

Troubleshooting:

- Check the container logs: docker logs <container_id>.
- Run the container interactively to debug: docker run -it <image_name> /bin/bash.
- Verify the entrypoint or command in the Dockerfile:

```
CMD ["executable", "arg1", "arg2"]
```

84. Error: "Docker-compose: cannot locate volume"

Cause: The specified volume does not exist or is not properly defined.

Troubleshooting:

- Check the volume definition in docker-compose.yml:

```
volumes:
  my_volume:
    driver: local
```

- Create the volume manually if it's missing:

```
docker volume create <volume_name>
```

- Validate the docker-compose.yml file: docker-compose config.

85. Error: "Error response from daemon: failed to attach network"

Cause: The container cannot connect to the specified network.

Troubleshooting:

- Check if the network exists: docker network ls.

- Recreate the network if needed:

```
docker network rm <network_name>
```

```
docker network create <network_name>
```

- Reconnect the container to the network:

```
docker network connect <network_name> <container_id>
```

86. Error: "Cannot execute command in container: no such file or directory"

Cause: The specified command or file is missing inside the container.

Troubleshooting:

- Use docker exec to explore the container and verify file paths:

bash

Copy code

```
docker exec -it <container_id> /bin/bash
```

- Confirm the command or file exists in the Docker image.
- Adjust the CMD or ENTRYPOINT in the Dockerfile to point to the correct location.

87. Error: "Error response from daemon: layer is in use"

Cause: An image layer is being used by another container.

Troubleshooting:

- Identify containers using the layer:

```
docker ps -a | grep <image_name>
```

- Stop and remove the dependent containers:

```
docker rm <container_id>
```

- Retry removing the image or layer.

88. Error: "Docker-compose: service build context must be a directory"

Cause: The build context in docker-compose.yml is not pointing to a valid directory.

Troubleshooting:

- Ensure the build context is a valid directory containing the Dockerfile:

```
services:
  app:
    build:
      context: ./app
```

- Validate the docker-compose.yml file: docker-compose config.

89. Error: "Docker container cannot resolve DNS"

Cause: The container cannot access the DNS server due to network issues.

Troubleshooting:

- Specify a DNS server manually when starting the container:

```
docker run --dns 8.8.8.8 <image_name>
```

- Verify Docker's default DNS configuration:

```
cat /etc/docker/daemon.json
```

Add a DNS entry if missing:

```
{
  "dns": ["8.8.8.8", "8.8.4.4"]
}
```

- Restart the Docker daemon.

90. Error: "Docker: file exists"

Cause: A file or directory in the host path conflicts with a mount point in the container.

Troubleshooting:

- Verify if the host path exists and is empty.
- Change the mount point to avoid conflicts:

```
docker run -v /new_host_path:/container_path <image_name>
```

- Remove or rename the conflicting host file or directory.

91. Error: "Error response from daemon: too many symbolic links encountered"

Cause: Circular symbolic links in the host file system are causing the error.

Troubleshooting:

- Inspect the file path for circular links using ls -l.
- Remove or resolve the circular symlink.
- Avoid bind-mounting directories with circular links.

92. Error: "Docker-compose: cannot start service: driver failed programming external connectivity"

Cause: Port conflict or network misconfiguration.

Troubleshooting:

- Verify if the required port is in use: sudo netstat -tuln | grep <port_number>.
- Change the port mapping in docker-compose.yml:

```
ports:
 - "8081:80"
```

- Restart Docker and try again: sudo systemctl restart docker.

93. Error: "Docker build fails with 'stat /var/lib/docker/tmp: no such file or directory'"

Cause: Temporary Docker build files are missing or inaccessible.

Troubleshooting:

- Verify the existence of the /var/lib/docker/tmp directory.
- Recreate the directory if missing:

```
sudo mkdir -p /var/lib/docker/tmp
sudo chmod 1777 /var/lib/docker/tmp
```

- Restart Docker: sudo systemctl restart docker.

94. Error: "Container running but cannot ping external network"

Cause: The container cannot access the internet due to incorrect network settings.

Troubleshooting:

- Check if the container is connected to the default bridge network:

```
docker network inspect bridge
```

- Restart the Docker service to reset network settings: sudo systemctl restart docker.
- Add a custom network with proper DNS settings:

```
docker network create --driver bridge --subnet=192.168.1.0/24 my_bridge
docker run --network my_bridge <image_name>
```

95. Error: "Docker-compose: invalid key in service definition"

Cause: Unsupported or misspelled key in the docker-compose.yml file.

Troubleshooting:

- Check the docker-compose.yml file for typos or unsupported keys.
- Use a YAML linter or run docker-compose config to validate the file.
- Reference the official Docker Compose documentation for valid configuration options.

96. Error: "Error response from daemon: write /var/lib/docker/tmp: no space left on device"

Cause: Disk space or inode limit has been reached.

Troubleshooting:

- Free up space by removing unused containers, images, and volumes:

```
docker system prune -a
```

- Check disk usage and inode availability:

```
df -h
```

```
df -i
```

- Extend the storage capacity of the Docker host if necessary.

97. Error: "Docker-compose: service depends on condition not met"

Cause: The dependent service has not started successfully.

Troubleshooting:

- Check the status of the dependent service with docker-compose ps.
- Review the logs for the dependent service: docker-compose logs <service_name>.
- Adjust the depends_on configuration to ensure proper startup sequencing:

```
depends_on:
  service_a:
    condition: service_healthy
```

98. Error: "Docker logs show 'Out of memory'"

Cause: The container is exceeding the memory limit.

Troubleshooting:

- Increase the memory allocated to the container:

```
docker run --memory=1g <image_name>
```

- Limit memory usage to prevent container crashes:

```
docker run --memory=512m --memory-swap=1g <image_name>
```

- Check the host system's memory usage and free up resources if necessary.

99. Error: "Cannot connect to Docker registry"

Cause: Network issues or authentication problems with the Docker registry.

Troubleshooting:

- Verify internet connectivity and DNS settings.
- Authenticate to the registry with docker login.
- If using a private registry, ensure it is accessible and properly configured.

100. Error: "Error response from daemon: endpoint with name already exists"

Cause: A conflicting container or network endpoint is already using the specified name.

Troubleshooting:

- Check for existing containers or endpoints with the conflicting name:

```
docker ps -a | grep <name>
```

- Remove the conflicting container or endpoint:

```
docker rm -f <container_id>
```

```
docker network disconnect <network_name> <endpoint_name>
```

- Restart Docker if necessary.

Conclusion

Docker has revolutionized application deployment by simplifying the process of creating, managing, and running containers. However, like any complex system, it is not without its challenges. The errors outlined in this guide reflect some of the most common obstacles encountered by Docker users. By understanding these issues and their resolutions, developers and system administrators can save valuable time and maintain efficient workflows.

Key Takeaways

1. Proactive Troubleshooting: Many Docker errors can be avoided or resolved by following best practices, such as proper resource allocation, YAML syntax validation, and staying updated with Docker versions.
2. Leverage Docker Documentation: The official Docker documentation and community forums are invaluable resources for in-depth problem-solving.
3. Continuous Learning: Errors are an opportunity to learn. Understanding their root causes strengthens overall expertise in containerization and DevOps practices.

As Docker continues to evolve, staying up to date with its features and resolving errors methodically will help ensure smoother deployments and operations. Use this guide as a quick reference to overcome Docker-related challenges and keep your containerized applications running reliably.