

**What causes the
'Docker: Cannot
Connect to the Docker
Daemon' error, and
how do you resolve it?**



- This error occurs when the Docker client cannot communicate with the Docker daemon.
- **Check Docker Daemon:** Ensure the Docker daemon is running by using `systemctl status docker` or `service docker status`.
- **Permissions Issue:** Ensure your user is in the 'docker' group by running `sudo usermod -aG docker $USER`.
- **Environment Variables:** Ensure the `DOCKER_HOST` environment variable is correctly set, especially in remote Docker setups.

Docker client cannot communicate with Docker daemon

Docker daemon is running

`systemctl status docker` or `service docker status`

User is in the “docker” group

`sudo usermod -aG docker $USER`

`DOCKER_HOST` environment variable is correctly set



**How do you fix
the 'Docker:
Image Pull
Backoff' error?**



- This error occurs when Docker fails to pull the specified image.
- **Check Image Name/Tag:**
Ensure the image name and tag are correct.
- **Network Issues:** Ensure there's no network issue preventing access to the registry.
- **Registry Credentials:** If pulling from a private registry, verify that the credentials are correct and configured with docker login.

Docker fails to pull the specified image

Image name and tag are correct

No network issues

Access to the image registry

Credentials are correct

Configured with docker login

**What is the cause of
the 'No space left on
device' error in
Docker, and how do
you resolve it?**



- This error occurs when the host system runs out of storage space.
- **Clean Up Unused Containers/Images:** Use **docker system prune -a** to remove unused data.
- **Check Disk Usage:** Use **df -h** to identify disk usage and free up space.
- **Increase Disk Size:** Expand the disk size on the host machine, if possible.

Host system runs out of storage space

Clean up unused containers or images

docker system prune -a

Check disk usage

df -h to check disk usage

Increase disk size on host machine



**What does the
'Container is
Unhealthy' status
indicate, and how do
you troubleshoot it?**



- The '**Unhealthy**' status means that the health check command specified in the Dockerfile is failing.
- **Inspect Logs:** Check the container logs using `docker logs <container_id>` for error messages.
- **Review Health Check Command:** Ensure the health check command is correctly implemented and accessible within the container.
- **Container Dependencies:** Verify that all dependencies required for the health check are available and functioning.

Health command in Dockerfile is failing

Check the container logs

`docker logs apache_container`

Health check command is correct

Accessible within the container

All dependencies are available and functioning



**How do you resolve a
'Permission Denied'
error when accessing
files inside a Docker
container?**



- This error typically occurs due to file ownership or permission issues.
- **Run Container as Root:** Try running the container with root privileges using `docker run --user root <image_name>`.
- **Change File Permissions:** Adjust file permissions within the container using `chmod` or `chown` to allow access.
- **Mount Options:** When mounting volumes, ensure that the correct permissions are set on the host directory.

File ownership or permission issues

Run container with root privileges

`docker run --user root apache`

Adjust file permissions within container

User `chmod` or `chown` to allow access

Correct permissions are set on host directory



**How do you fix the
'Docker: Bind for
0.0.0.0:80 failed: port
is already allocated'
error?**



- This error occurs when the specified port is already in use on the host system.
- **Stop Conflicting Containers:** Identify and stop the container using the port with `docker ps` and `docker stop <container_id>`.
- **Change Host Port:** Modify the port mapping to use a different host port, e.g., `docker run -p 8080:80 <image_name>`.
- **Release the Port:** Use `sudo lsof -i :80` to find and stop any process occupying the port.

Specified port is already use on host machine

Identify and stop the container

`docker ps` and `docker stop apache_container`

Modify port mappings

`docker run -p 8080:80 apache`

`sudo lsof -i :80`



How do you resolve the 'Docker: Cannot Start Service: Mounts Denied' error?



- This error usually occurs due to incorrect or restricted mount points.
- **Verify Mount Path:** Ensure the path you're trying to mount exists and has the correct permissions.
- **Check Docker Desktop Settings:** On Docker Desktop (Mac/Windows), ensure that the file paths are shared and allowed.
- **Use Valid Bind Mounts:** Ensure that the bind mount syntax is correct, e.g., `docker run -v /host/path:/container/path <image_name>`.

Incorrect or restricted mount points

Mount path exists

Mount path has correct permissions

File paths are shared and allowed

Bind mount syntax is correct

`docker run -v /host/path:/container/path apache`



**What causes the
'Docker: connection
reset by peer' error,
and how do you
troubleshoot it?**



- This error often occurs due to network connectivity issues between Docker containers or the Docker daemon and external services.
- **Check Network Configuration:** Ensure Docker network settings are correctly configured.
- **Firewall Rules:** Verify that firewalls or security groups are not blocking traffic.
- **Container Resource Limits:** Ensure containers have sufficient resources and are not being throttled or killed.

Network connectivity issues

Between docker containers or docker daemon and external services

Docker network settings are correctly configured

Firewalls are not blocking traffic

Security groups are not blocking traffic

Have sufficient resources and not being throttled or killed



**How do you resolve
the 'Docker: Layer
already exists' error
during a docker
build?**



- This error can occur when Docker tries to reuse an existing layer but encounters a conflict.
- **Clear Build Cache:** Use `docker builder prune` or `docker build --no-cache` to avoid reusing cached layers.
- **Inspect Dockerfile:** Ensure that the Dockerfile commands are correctly defined and that changes are necessary to create new layers.
- **Rebuild Image:** Try rebuilding the image from scratch to resolve any cache issues.

Tries to reuse an existing layer but encounters a conflict

`docker builder prune` or `docker build --no-cache`

Avoid reusing cached layers

Dockerfile commands are correctly defined

Changes are necessary to create new layers

Rebuild image from scratch to resolve cache issues



**What causes the
'Cannot Delete Docker
Network: Network has
active endpoints' error,
and how do you fix it?**



- This error occurs when you try to delete a network that still has active containers connected.
- **List Active Containers:** Use `docker network inspect <network_name>` to see which containers are still using the network.
- **Disconnect Containers:** Disconnect the containers from the network using `docker network disconnect <network_name> <container_id>`.
- **Stop/Remove Containers:** Stop or remove the containers if they are no longer needed, then retry deleting the network.

Delete a network that still has active containers connected

`docker network inspect overlay`

Which containers are still using the network

Disconnect containers from network

`docker network disconnect overlay apache_container`

Stop or remove containers if no longer needed



**How do you resolve
a 'Docker: Unknown
instruction in
Dockerfile' error?**



- This error occurs when the Dockerfile contains an invalid or misspelled instruction.
- **Check Instruction Syntax:** Ensure that all instructions in the Dockerfile (e.g., FROM, RUN, CMD) are spelled correctly and in uppercase.
- **Dockerfile Version Compatibility:** Ensure the Dockerfile syntax is compatible with the Docker version you are using.
- **Consult Docker Documentation:** Refer to the official Dockerfile reference guide for valid instructions.

Dockerfile container invalid misspelled instruction

Instructions are spelled corrected

Instructions are in uppercase

FROM, RUN, CMD

Dockerfile syntax is compatible

Refer official Dockerfile reference guide



**What is the 'Docker:
Cannot kill container'
error, and how do
you resolve it?**



- This error occurs when Docker cannot stop or remove a container.
- **Force Stop:** Use `docker kill <container_id>` to forcefully stop the container.
- **Remove with Force:** Use `docker rm -f <container_id>` to forcefully remove the container.
- **Check Container State:** Ensure the container is not in a paused state before trying to kill it.

Cannot stop or remove a container

`docker kill apache_container`

Forcefully stop the container

`docker rm -f apache_container`

Forcefully remove the container

Container is not in paused state



How do you resolve the 'Docker: invalid reference format' error?



- This error occurs when the image name, tag, or repository format is incorrect.
- **Correct Image Name Format:**
Ensure the image name and tag follow the repository:tag format, e.g., myimage:latest.
- **Remove Invalid Characters:**
Ensure there are no spaces or special characters in the image name or tag.
- **Use Valid Repository Names:**
Ensure the repository name adheres to Docker's naming conventions.

Image name, tag or repo format is incorrect

Image name and tag follow the right format

repository:tag format

No spaces in the image name

No special characters in the image name

Repo name adheres to Docker's naming conventions

**What causes the 'Docker:
Cannot connect to Docker
Daemon at
unix:///var/run/docker.so
ck' error, and how do you
fix it?**



- This error occurs when the Docker client cannot access the Docker daemon socket.
- **Start Docker Daemon:** Ensure the Docker daemon is running by checking with **systemctl status docker** or **service docker start**.
- **Check Socket Permissions:** Ensure the **/var/run/docker.sock** file has the correct permissions, e.g.,
sudo chmod 666 /var/run/docker.sock.
- **User Group Permissions:** Add your user to the Docker group with **sudo usermod -aG docker \$USER**.

Docker client cannot access Docker daemon socket

Docker daemon is running

systemctl status docker or service docker start

/var/run/docker.sock file has correct permissions

sudo chmod 666 /var/run/docker.sock

sudo usermod -aG docker \$USER



**How do you
troubleshoot the
'Docker: Container
Exited with Code 137'
error?**



- **Exit code 137** indicates that the container was terminated due to an out-of-memory (OOM) condition or was manually killed.
- **Increase Memory Limits:** Adjust memory limits for the container using **--memory** and **--memory-swap** flags.
- **Check Resource Usage:** Monitor the container's resource usage with **docker stats** and ensure it's not exceeding available memory.
- **Optimize Application:** Review and optimize the application running inside the container to reduce memory consumption.

Container terminated due to OOM or manually killed

Adjust memory limits for container

--memory and --memory-swap flags

Monitor container's resource usage

docker stats

Review and optimize app running

