General Docker Commands

Troubleshooting Docker involves identifying and fixing issues related to containers, images, networks, and more. Below is a comprehensive guide to basic Docker troubleshooting commands, organized by category.

@BUHRAHN503

Verify Docker Installation

Verify Docker Installation

```
docker --version
docker info
```

Check Docker Service Status

```
sudo systemctl status docker
```

Container Troubleshooting

List All Containers

```
docker ps # List running containers docker ps -a # List all containers (including stopped ones)
```

Inspect a Container

```
docker inspect <container_id_or_name>
```

Provides detailed information about the container.

Check Logs of a Container

Check Logs of a Container

```
docker logs <container_id_or_name>
docker logs -f <container_id_or_name>  # Follow live logs
```

Execute a Command Inside a Container

```
docker exec -it <container_id_or_name> /bin/bash
# For bash shell
docker exec -it <container_id_or_name> /bin/bash
# For minimal shell
```

Check Resource Usage of Containers

```
docker stats
```

Image Troubleshooting

List All Images

```
docker images
```

Inspect an Image

```
docker inspect <image_id_or_name>
```

Remove Unused or Dangling Images

```
docker image prune
docker image prune -a  # Remove all unused images
```

Pull the Latest Image

```
docker pull <image_name>:<tag>
```

Check Image Layers

```
docker history <image_id_or_name>
```

Network Troubleshooting

List Networks

```
docker network ls
```

Inspect a Network

```
docker network inspect <network_name>
```

Connect a Container to a Network

```
docker network connect <network_name> <container_id_or_name>
```

Disconnect a Container from a Network

```
docker network disconnect <network_name> <container_id_or_name>
```

Remove Unused Networks

```
docker network prune
```

Volume Troubleshooting

List Volumes

```
docker volume ls
```

Inspect a Volume

```
docker volume inspect <volume_name>
```

Remove Unused Volumes

```
docker volume prune
```

Common Issues and Fixes

Container Won't Start Check logs

```
docker logs <container_id_or_name>
```

Verify ports and bindings

```
docker inspect <container_id_or_name> | grep "HostPort"
```

Container Exited Unexpectedly Check exit code

```
docker ps -a
```

Analyze logs

```
docker logs <container_id_or_name>
```

Image Pull Fails

Check network

```
ping <registry_url>
```

Use a specific registry

```
docker pull registry.example.com/<image_name>:<tag>
```

High Resource Usage

Monitor real-time stats

```
docker stats
```

Orphaned Containers Remove exited containers

```
docker container prune
```

Debugging Docker Daemon

Restart Docker Daemon

```
sudo systemctl restart docker
```

Check Docker Daemon Logs

```
sudo journalctl -u docker
```

Advanced Debugging

Debugging Networking with nsenter

Enter a container’s network namespace

```
nsenter --target $(docker inspect --format '{{
.State.Pid }}' <container_id_or_name>) --net
```

View IPTables Rules

```
sudo iptables -L -n -v
```

Clean Up Docker System

Remove Stopped Containers

```
docker container prune
```

Remove Unused Images

```
docker image prune -a
```

Remove Unused Networks

```
docker network prune
```

Clean Everything (Use Cautiously)

```
docker system prune -a
```

Then restart the Docker service

```
sudo systemctl restart docker
```