Docker Basic Commands with Examples and Explanations:

1. **docker --version**

- Displays the installed Docker version.
- docker --version

2. docker pull

- Downloads a Docker image from a registry like Docker Hub.
- docker pull ubuntu

3. docker images

- Lists all available images on your system.
- docker images

4. docker rmi <image_id>

- Removes a Docker image by its ID.
- docker rmi abc123

5. docker ps

- Lists all running containers.
- docker ps

6. docker ps -a

- Lists all containers, including stopped ones.
- docker ps -a

7. docker run <image_name>

- Creates and starts a new container from a specified image.
- docker run ubuntu

8. docker run -it <image_name>

- Starts an interactive container session.
- docker run -it ubuntu /bin/bash

9. docker start <container_id>

- Starts an existing stopped container.
- docker start abc123

10. docker stop <container_id>

- Stops a running container.
- docker stop abc123

11. docker restart <container_id>

- Restarts a running or stopped container.
- docker restart abc123

12. docker rm <container_id>

- Removes a stopped container.
- docker rm abc123

13. docker exec -it <container_id> <command>

- Executes a command in a running container.
- docker exec -it abc123 /bin/bash

14. docker logs <container_id>

- Displays the logs of a container.
- docker logs abc123

15. **docker inspect <container_id>**
    - Provides detailed information about a container.
    - docker inspect abc123
16. **docker build -t <image_name> <Dockerfile_path>**
    - Builds an image from a Dockerfile.
    - docker build -t my_image .
17. **docker tag <image_id>** repository:tag
    - Tags an image for a repository.
    - docker tag abc123 myrepo/myimage:v1
18. **docker push** repository:tag
    - Pushes an image to a Docker registry.
    - docker push myrepo/myimage:v1
19. **docker pull repository:tag**
    - Pulls an image from a Docker registry.
    - docker pull myrepo/myimage:v1
20. **docker network ls**
    - Lists all Docker networks.
    - docker network ls
21. **docker network create <network_name>**
    - Creates a new Docker network.

- docker network create my_network

22. **docker network rm <network_name>**
    - Removes an existing Docker network.
    - docker network rm my_network
23. **docker volume ls**
    - Lists all Docker volumes.
    - docker volume ls
24. **docker volume create <volume_name>**
    - Creates a new Docker volume.
    - docker volume create my_volume
25. **docker volume rm <volume_name>**
    - Removes a Docker volume.
    - docker volume rm my_volume
26. **docker-compose up**
    - Starts services defined in a docker-compose.yml file.
    - docker-compose up
27. **docker-compose down**
    - Stops and removes services created by Docker Compose.
    - docker-compose down
28. **docker-compose ps**

- Lists the status of Docker Compose services.
- docker-compose ps

29. **docker commit <container_id> <new_image_name>**
    - Creates a new image from a container's changes.
    - docker commit abc123 my_image:v2
30. **docker system prune**
    - Removes unused data (stopped containers, networks, etc.).
    - docker system prune
31. **docker stats**
    - Displays a live stream of container resource usage statistics.
    - docker stats
32. **docker save -o <output_file> <image_name>**
    - Saves an image to a tar archive.
    - docker save -o my_image.tar my_image:v1
33. **docker load -i <input_file>**
    - Loads an image from a tar archive.
    - docker load -i my_image.tar
34. **docker export <container_id> -o <output_file>**
    - Exports a container's filesystem to a tar archive.
    - docker export abc123 -o container_fs.tar

35. **docker import <input_file>**

    - Imports a container's filesystem from a tar archive as an image.
    - docker import container_fs.tar

36. **docker pause <container_id>**

- Pauses all processes within a container.
- docker pause abc123

37. **docker unpause <container_id>**

- Unpauses a paused container, resuming its processes.
- docker unpause abc123

38. **docker rename <old_name> <new_name>**

- Renames an existing container.
- docker rename old_container_name new_container_name

39. **docker update <container_id> --cpus <value>**

- Updates the resource limits for a container (e.g., CPU).
- docker update abc123 --cpus 2

40. **docker attach <container_id>**

- Attaches to a running container's standard input, output, and error streams.
- docker attach abc123

41. **docker cp <container_id>:/path/to/file <host_destination>**

- Copies files from a container to the host machine.
- docker cp abc123:/var/log/nginx.log /home/user/nginx.log

42. **docker diff <container_id>**

- Shows the changes made to a container's filesystem.
- docker diff abc123

43. **docker login**

- Logs into a Docker registry (Docker Hub or private).
- docker login

44. **docker logout**

- Logs out from a Docker registry.
- docker logout

45. **docker history <image_name>**

- Displays the history of an image’s layers.
- docker history ubuntu

46. **docker export <container_id> -o <filename.tar>**

- Exports a container’s filesystem as a tar archive.
- docker export abc123 -o container_backup.tar

47. **docker import <filename.tar> <image_name>**

- Imports a tarball as a Docker image.
- docker import container_backup.tar new_image_name

48. **docker commit <container_id> <image_name>**

- Creates a new image from a container’s changes.
- docker commit abc123 my_custom_image

49. **docker exec <container_id> <command>**

- Executes a command inside a running container.
- docker exec abc123 ls /app

50. **docker stats**

- Displays a live stream of resource usage statistics for running containers.
- docker stats

51. **docker events**

- Displays real-time events from the Docker daemon.
- docker events

52. **docker version**

- Displays the Docker version information.
- docker version

53. **docker info**

- Displays system-wide information about Docker.
- docker info

54. **docker wait <container_id>**

- Blocks until a container stops, then prints the exit code.
- docker wait abc123

55. **docker prune**

- Cleans up unused containers, networks, images (dangling), and volumes.
- docker system prune

56. **docker image prune**

- Removes unused or dangling images.
- docker image prune

57. **docker volume prune**

- Removes unused volumes.
- docker volume prune

58. **docker network prune**

- Removes unused networks.
- docker network prune

59. **docker network create <network_name>**

- Creates a new network for containers.
- docker network create my_network

60. **docker network ls**

- Lists all available Docker networks.
- docker network ls

61. **docker network inspect <network_name>**

- Shows detailed information about a specific network.
- docker network inspect my_network

62. **docker volume create <volume_name>**

- Creates a new Docker volume.
- docker volume create my_volume

63. **docker volume ls**

- Lists all Docker volumes.
- docker volume ls

64. **docker volume inspect <volume_name>**

- Shows detailed information about a specific volume.
- docker volume inspect my_volume

65. **docker-compose up**

- Starts containers defined in a docker-compose.yml file.
- docker-compose up

66. **docker-compose down**

- Stops and removes containers, networks, images, and volumes created by docker-compose up.
- docker-compose down

67. **docker-compose build**

- Builds or rebuilds services defined in the docker-compose.yml file.
- docker-compose build

68. **docker-compose stop**

- Stops running containers without removing them.

- docker-compose stop

69. **docker-compose restart**

- Restarts all services in a docker-compose setup.
- docker-compose restart

70. **docker-compose ps**

- Lists all containers in a docker-compose environment.
- docker-compose ps

71. **docker-compose logs**

- Displays output from the running containers managed by docker-compose.
- docker-compose logs

72. **docker-compose rm**

- Removes stopped service containers.
- docker-compose rm

73. **docker-compose pull**

- Pulls images for services defined in the docker-compose.yml file.
- docker-compose pull

74. **docker-compose exec <service_name> <command>**

- Executes a command inside a running service container.
- docker-compose exec web ls /app

75. **docker-compose scale <service_name>=<num_instances>**

- Scales the number of containers for a service.
- docker-compose scale web=3

76. **docker tag <image> <new_image_tag>**

- Tags an image with a new tag.
- docker tag my_image my_image:v2

77. **docker save -o <filename.tar> <image_name>**

- Saves an image to a tar archive.
- docker save -o my_image.tar my_image

78. **docker load -i <filename.tar>**

- Loads an image from a tar archive.
- docker load -i my_image.tar

79. **docker stop $(docker ps -q)**

- Stops all running containers.
- docker stop $(docker ps -q)

80. **docker rm $(docker ps -a -q)**

- Removes all stopped containers.
- docker rm $(docker ps -a -q)

81. **docker rmi $(docker images -q)**

- Removes all images.
- docker rmi $(docker images -q)

82. **docker rename <old_container_name> <new_container_name>**

- Renames a running or stopped container.
- docker rename my_old_container my_new_container

83. **docker update --cpu-shares <value> <container_id>**

- Updates CPU shares for a container.
- docker update --cpu-shares 512 abc123

84. **docker update --memory <value> <container_id>**

- Updates memory limit for a container.
- docker update --memory 512m abc123

85. **docker daemon**

- Runs the Docker daemon manually (typically done automatically).
- docker daemon

86. **docker logs -f <container_id>**

- Follows the logs of a container in real-time.
- docker logs -f abc123

87. **docker history <image_name>**

- Shows the history of an image's layers.
- docker history ubuntu

88. **docker inspect <container_id>**

- Shows detailed information about a container or image.
- docker inspect abc123

89. **docker diff <container_id>**

- Shows changes made to a container’s filesystem.
- docker diff abc123

90. **docker cp <container_id>:<source_path> <destination_path>**

- Copies files from a container to the host system.
- docker cp abc123:/app/file.txt /tmp

91. **docker wait <container_id>**

- Blocks until a container stops, then prints its exit code.
- docker wait abc123

92. **docker kill <container_id>**

- Forcefully stops a running container.
- docker kill abc123

93. **docker attach <container_id>**

- Attaches to a running container to view logs or interact with it.
- docker attach abc123

94. **docker pause <container_id>**

- Pauses all processes within a container.
- docker pause abc123

95. **docker unpause <container_id>**

- Resumes all processes within a paused container.
- docker unpause abc123

96. **docker top <container_id>**

- Shows running processes within a container.
- docker top abc123

97. **docker port <container_id>**

- Displays a list of port mappings for a container.
- docker port abc123

98. **docker exec -it <container_id> /bin/bash**

- Opens an interactive shell inside a running container.
- docker exec -it abc123 /bin/bash

99. **docker network create <network_name>**

- Creates a new network for Docker containers to communicate.
- docker network create my_network

100. **docker network ls**

- Lists all available networks.
- docker network ls