Docker Fundamentals with Linux – Intermediate Training Guide

1. Introduction to Docker

Docker is a platform for building, running, and shipping applications inside containers.
It helps isolate applications using namespaces and cgroups on Linux. This document provides an intermediate-level overview focused on commands, concepts, and hands-on examples.

2. Core Concepts

• Container: A lightweight isolated runtime environment.
• Image: A read-only template used to create containers.
• Registry: Storage for images (Docker Hub, ECR, etc.).
• Volume: Persistent storage for containers.
• Network: Communication layer between containers.
• Docker Engine: Runtime and daemon responsible for container operations.

3. Docker Architecture on Linux

Docker uses Linux kernel features:
• Namespaces: Process isolation (PID, NET, MNT, IPC, UTS).
• Control Groups (cgroups): Resource limits.
• Union Filesystems: Layered images (OverlayFS).
The Docker daemon (dockerd) listens for API requests and manages images, containers, networks, and volumes.

4. Installation Commands (Linux)

Update system:
$ sudo apt update && sudo apt install -y ca-certificates curl

Install Docker:
$ curl -fsSL https://get.docker.com | sudo sh

Enable service:
$ sudo systemctl enable --now docker

Verify installation:
$ docker --version

5. Working with Docker Images

List images:
$ docker images

Download image:
$ docker pull nginx:latest

Build an image:
$ docker build -t myapp:1.0 .

Remove image:
$ docker rmi image_id

6. Working with Containers

Run a container:
$ docker run -d --name webserver -p 8080:80 nginx

View logs:
$ docker logs webserver

Exec into container:
$ docker exec -it webserver bash

Stop container:
$ docker stop webserver

Remove container:
$ docker rm webserver

7. Docker Networking

List networks:
$ docker network ls

Create network:
$ docker network create mynet

Run container in a network:
$ docker run -d --network=mynet nginx

Inspect network:
$ docker network inspect mynet

8. Docker Volumes

Create volume:
$ docker volume create data_vol

Use volume in container:

$ docker run -d -v data_vol:/var/lib/mysql mysql

List volumes:
$ docker volume ls

9. Docker Compose Basics

docker-compose.yaml example:

version: '3'
services:
app:
image: nginx
ports:
- "8080:80"
volumes:
- ./html:/usr/share/nginx/html

Run:
$ docker compose up -d

Stop:
$ docker compose down

10. Helpful Short Notes

• Use 'docker ps -a' to view all containers.
• Use tags to version images: myapp:1.0.
• Clean unused objects: docker system prune.
• Use volumes for persistent data rather than bind mounts.
• Use docker inspect for detailed container metadata.