Inception Project - Complete Study Guide

A Comprehensive Reference for Docker Infrastructure Implementation

---

Table of Contents

---

1. Project Requirements Overview

Mandatory Requirements

- **Virtual Machine**: All work must be done in a VM
- **Docker Compose**: Use docker-compose.yml for orchestration
- **Custom Dockerfiles**: No pulling ready-made images (except Alpine/Debian base)
- **Three Core Services**:
  - NGINX with TLSv1.2/1.3 only
  - WordPress + PHP-FPM (without nginx)
  - MariaDB (without nginx)
- **Infrastructure Components**:
  - Two volumes (database + website files)
  - Custom Docker network
  - Automatic restart on crash
- **Security Requirements**:
  - No passwords in Dockerfiles
  - Use environment variables
  - Implement Docker secrets
  - NGINX as only entry point via port 443

Key Restrictions

- No `network: host` or `--link`
- No infinite loops (`tail -f`, `sleep infinity`, etc.)
- No `latest` tag usage
- Domain must be `login.42.fr`
- Database users: admin user cannot contain "admin" or "administrator"

---

2. Docker Fundamentals

What is Docker?

Docker is a containerization platform that packages applications and their dependencies into lightweight, portable containers. Unlike virtual machines, containers share the host OS kernel.

Key Concepts

Images vs Containers

- **Image**: Read-only template with application code and dependencies
- **Container**: Running instance of an image
- **Layer**: Each instruction in Dockerfile creates a new layer

Container Lifecycle

```
Docker Image → Docker Container → Running Process
```

Docker Architecture

- **Docker Daemon**: Background service managing containers
- **Docker Client**: Command-line interface
- **Docker Registry**: Storage for Docker images (DockerHub)

Essential Commands

bash

```bash
# Image management
docker build -t name:tag .
docker
```