# Docker Compose Stage 1 - Nginx + MySQL Setup

This document explains the step-by-step process of setting up a simple multi-container Docker Compose application with Nginx (website) and MySQL (database). It includes the commands used and the outcomes observed.

## Step 1: Create docker-compose.yml

Create a file named docker-compose.yml with the following content:

services:  
 website:  
 image: nginx:latest  
 ports:  
 - "8080:80"  
 database:  
 image: mysql:latest  
 ports:  
 - "3307:3306"  
 environment:  
 MYSQL\_ROOT\_PASSWORD: "1234"  
 MYSQL\_USER: "Rahi"

## Step 2: Start the Containers

Run the following command to start the containers in detached mode:

docker-compose up -d

## Step 3: Check Running Containers

To see if the containers are running:

docker ps

## Step 4: Login to MySQL as root

Run this command to connect to MySQL as root:

docker exec -it docker-compose-practice-database-1 mysql -uroot -p

When prompted, enter the root password (1234).

## Step 5: Create a MySQL User

Inside MySQL shell, create the Rahi user and give privileges:

CREATE USER 'Rahi'@'%' IDENTIFIED BY 'mypassword';  
GRANT ALL PRIVILEGES ON \*.\* TO 'Rahi'@'%' WITH GRANT OPTION;  
FLUSH PRIVILEGES;  
EXIT;

## Step 6: Test Connection from Website Container

Enter the website container:

docker exec -it docker-compose-practice-website-1 bash

Inside the container, install MySQL client:

apt update && apt install -y default-mysql-client

Test connection to the database service:

mysql -h database -uRahi -p

Enter 'mypassword' when prompted. You should see the MySQL shell.

## Summary

You have successfully set up two containers (Nginx + MySQL) using Docker Compose. Both containers are in the same internal Docker network and can communicate using their service names. Nginx is accessible from your host at http://localhost:8080 and MySQL at port 3307 (from your host).