### **Lab Assignment: Designing Resilient and Reliable IT Infrastructure with Docker Containers**

#### **Objective:**

The objective of this lab assignment is to provide students with hands-on experience in designing and implementing a resilient and reliable IT infrastructure using Docker containers. By the end of this lab, students will understand how to set up a scalable and fault-tolerant environment using Docker's tools and features.

#### **Prerequisites:**

* Basic knowledge of Docker and containerization.
* Familiarity with Linux command line.
* Understanding of networking and web server concepts.

#### **Lab Overview:**

1. Part 1: Setting Up Docker and Docker Compose
2. Part 2: Designing a Scalable Web Application with Docker
3. Part 3: Implementing Load Balancing and Failover
4. Part 4: Monitoring and Logging
5. Part 5: Disaster Recovery and Backup

### **Part 1: Setting Up Docker and Docker Compose**

#### **Instructions:**

1. Install Docker:
   * Follow the official Docker installation guide for your operating system.
   * Verify the installation by running docker --version.
2. Install Docker Compose:
   * Follow the official Docker Compose installation guide.
   * Verify the installation by running docker-compose --version.
3. Create a Docker Network:
   * Set up a custom Docker network:

### **Part 2: Designing a Scalable Web Application with Docker**

#### **Instructions:**

1. Create a basic web application (e.g., a simple Node.js app) and Dockerize it.
2. Create a docker-compose.yml file to define the services.
3. Build the Docker images and start the services

### **Part 3: Implementing Load Balancing and Failover**

1. Set Up a Load Balancer: Use NGINX as a load balancer for the web application. Create a Dockerfile for NGINX
2. Update Docker Compose for Load Balancing: Add the NGINX service to docker-compose.yml
3. Run the Load-Balanced Application:Build and run the updated Docker Compose setup

### **Part 4: Monitoring and Logging**

#### 

1. Set Up Monitoring with Prometheus: Add Prometheus to docker-compose.yml:
2. Create a prometheus.yml configuration file for Prometheus:

### **Part 5: Disaster Recovery and Backup**

1. Set Up Database Backup:Create a backup script for the PostgreSQL database
2. Automate Backups: Use cron jobs to automate the execution of the backup script (can be set up on the host or within a Docker container).

#### **Submission Requirements:**

* A working Docker Compose setup with the services described above.
* Screenshots or logs showing successful load balancing and failover.
* Documentation explaining the setup, configurations, and any challenges faced.
* A brief report on monitoring results and how the system handles different loads.
* Backup scripts and cron job setup.

#### **Evaluation Criteria:**

* Correctness and completeness of the Docker setup.
* Functionality of load balancing and failover mechanisms.
* Effective implementation of monitoring and logging.
* Proper setup and automation of disaster recovery and backup procedures.
* Clarity and thoroughness of documentation and report.