🧠 3-Hour Study Plan (Topic: Load Balancing & Scalable File Server)

⏱️ Total Duration: 3 hours

Goal: Understand load balancing deeply, explore scaling strategies, and design a scalable file/image server from scratch.

🕐 Hour 1 — Core Concepts (Theory & Understanding)

Objective: Build a solid conceptual foundation.

1️⃣Concepts to Study (45 mins)

- **Horizontal vs Vertical Scaling**
  - Horizontal → Add more servers (scale out).
  - Vertical → Increase resources on a single server (scale up).
  - Pros/Cons, when to use each.
- **Reverse Proxy**
  - How it routes requests from client → backend servers.
  - Examples: Nginx, HAProxy, Envoy.
- **Load Balancing Algorithms**
  - Round Robin
  - Least Connections
  - IP Hash
  - Consistent Hashing
  - Weighted Round Robin
- **Health Checks & Failover**
  - Importance of removing unhealthy servers.
  - Active vs passive health checks.

2️⃣Quick Reading/Video (15 mins)

- Nginx Load Balancing docs: https://docs.nginx.com/nginx/admin-guide/load-balancer/http-load-balancer/
- YouTube: “System Design – Load Balancer” (Gaurav Sen or ByteByteGo).

📓 *Take notes on when to use each LB algorithm.*

🕑 Hour 2 — Design the System (Scalable File/Image Server)

Objective: Apply theory to a practical design.

1️⃣ System Requirements (10 mins)

- Users upload/download images/files.
- Handle millions of requests concurrently.
- Ensure reliability, low latency, and fault tolerance.

2️⃣ High-Level Design (30 mins)

Draw architecture (paper or Miro):

```
Client → CDN → Load Balancer → App Servers → Object Storage (S3 / GFS)
```

Include:

- **Reverse Proxy (Nginx/HAProxy)**
- **Static Content Caching (CDN)**
- **Application Layer** (handles metadata, file validation)
- **Storage Layer** (Object store or distributed FS)
- **Database** for metadata (PostgreSQL or MongoDB).

3️⃣ Scaling & Trade-offs (20 mins)

- **Horizontal scaling:** add more app servers behind LB.
- **Stateless app layer:** store sessions externally (Redis).
- **Caching:** CDN and edge caching to reduce load.
- **Fault Tolerance:** health checks + multiple LB instances.

- **Security:** HTTPS termination at LB.

🕒 Hour 3 — Hands-on Practice + Reflection

Objective: Reinforce learning by implementing and summarizing.

1️⃣Implementation (45 mins)

- Install **Nginx** locally or use Docker.

Configure **Round Robin load balancing** between two dummy backend servers.
Example:

```
upstream backend {
    server localhost:8081;
    server localhost:8082;
}

server {
    listen 80;
    location / {
        proxy_pass http://backend;
    }
}
```

-

Test with:

```
curl localhost
```

- Observe load distribution.

2️⃣Reflection & Notes (15 mins)

Write down:

- When to use consistent hashing vs round robin.
- Why CDNs are crucial for image servers.
- How scaling affects cost and complexity.

✅ Outcome After 3 Hours

You will:

- Understand all major load balancing strategies.
- Know how to architect a scalable file/image delivery system.
- Have a working local Nginx load balancer setup.
- Be prepared to answer interview questions like:
  - *How would you scale an image upload service?*
  - *What’s the difference between reverse proxy and load balancer?*