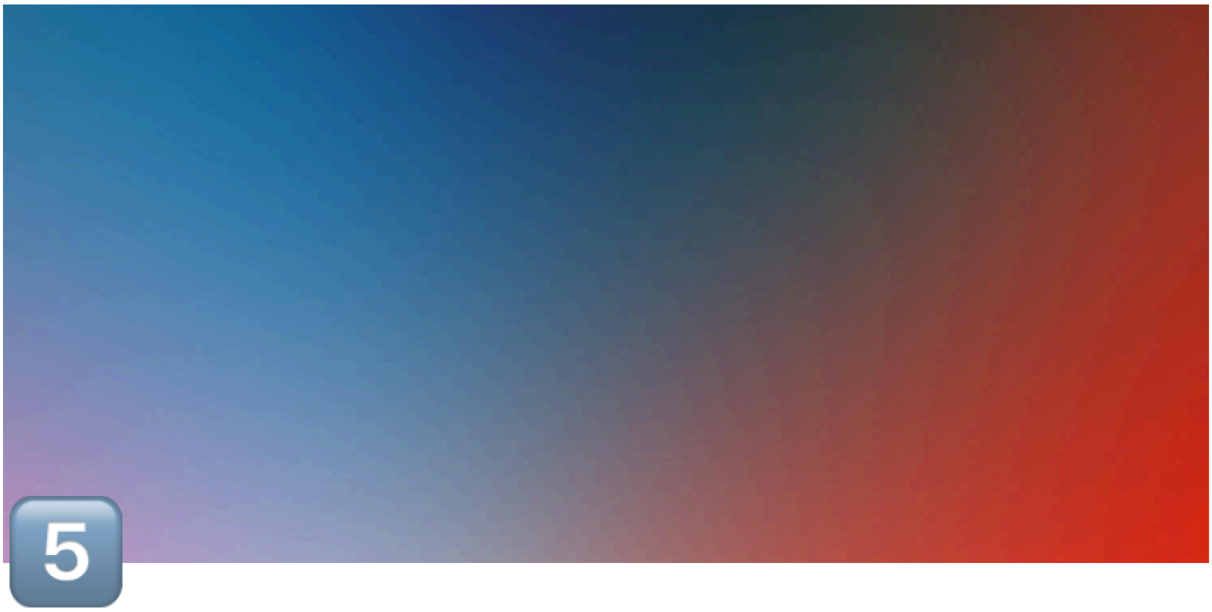

5️⃣

🐳 Phase 5 — DevOps & Deployment

Duration: 3 Months (±12 Weeks)

🎯 **Goal**

Learn how to **package, deploy, automate, monitor, and maintain backend systems** using Docker, CI/CD, cloud platforms, and basic observability.

By the end, your backend projects are **production-deployable and reliable**.

🗓️ Roadmap Phase 5 — Weekly Breakdown

📌 Month 1 — Containerization & Deployment Basics

🗓️ Week 1: Linux & Server Fundamentals 🧠

1. Review Linux essentials for servers:
    - File system structure
    - Users & permissions
    - Processes & ports
2. Understand how servers run backend applications.

3. Learn basic networking concepts:
    - Ports
    - Firewalls
4. Practice SSH into a remote server.
5. Learn secure server access.

🗓️ Week 2: Docker Fundamentals 🐳

1. Understand containerization concepts.
2. Learn Docker basics:
    - Images
    - Containers
    - Volumes
3. Write a `Dockerfile` for a backend application.
4. Understand multi-stage builds.
5. Run backend apps inside containers.

🗓️ Week 3: Docker Compose & Multi-Service Setup 🧩

1. Understand why Docker Compose is used.
2. Create `docker-compose.yml`:
    - Backend service
    - Database service
    - Redis service
3. Manage environment variables in containers.
4. Handle networking between containers.
5. Persist data using volumes.

🗓️ Week 4: Deployment Basics 🌍

1. Choose a cloud provider:
    - AWS / GCP / Azure

    - Or DigitalOcean

2. Deploy containerized backend to the cloud.
3. Configure:
    - Environment variables
    - Database connection
4. Expose API publicly.
5. Verify deployment stability.

📌 Month 2 — CI/CD & Production Practices

🗓️ Week 5: CI/CD Fundamentals 🔁

1. Understand CI/CD concepts:
    - Continuous Integration
    - Continuous Deployment
2. Learn GitHub Actions basics.
3. Create CI pipeline:
    - Lint
    - Test
    - Build
4. Fail builds on test errors.
5. Understand pipeline triggers.

🗓️ Week 6: Automated Deployment 🚀

1. Deploy backend automatically after successful CI.
2. Build Docker images in CI.
3. Push images to container registry.
4. Pull images on server.
5. Zero-downtime deployment concepts.

🗓️ Week 7: Environment Management & Secrets 🔐

1. Separate environments:
    - Development
    - Staging
    - Production
2. Manage secrets securely:
    - Environment variables
    - Secrets manager (basic)
3. Avoid committing secrets to Git.
4. Rotate credentials safely.
5. Validate environment configurations.

🗓️ Week 8: Logging & Monitoring 📊

1. Understand importance of observability.
2. Implement structured logging.
3. Centralize logs (basic):
    - ELK stack (concept)
4. Learn metrics:
    - CPU
    - Memory
    - Request latency
5. Intro to Prometheus & Grafana.

📌 Month 3 — Reliability, Scaling & Final Deployment

🗓️ Week 9: Reliability & Fault Tolerance 🛡️

1. Handle application crashes.
2. Restart policies.
3. Health checks:

- Liveness
- Readiness

4. Graceful shutdown handling.
5. Retry & timeout strategies.

🗓️ Week 10: Scaling & Performance ⚡

1. Vertical vs horizontal scaling.
2. Load balancing basics.
3. Stateless backend design.
4. Scale containers.
5. Handle traffic spikes.

🗓️ Week 11: Production Deployment Project 🧩

1. Choose a previous project:
    - E-commerce backend
    - Video platform backend
2. Deploy to production-grade environment.
3. Enable:
    - CI/CD
    - Logging
    - Monitoring
4. Secure endpoints.
5. Validate system behavior.

🗓️ Week 12: Final Review & Readiness 🏁

1. Full production testing.
2. Simulate failure scenarios.
3. Review deployment pipeline.
4. Optimize configuration.

5. Prepare system documentation.

🛠️ Phase 5 Project — Production Deployment

Deliverables

- Backend application running in cloud
- Dockerized services
- CI/CD pipeline (GitHub Actions)
- Secure environment variables
- Logs & basic monitoring
- Public API endpoint

🧠 Phase 5 Summary ✨

- **Month 1:** Docker & cloud deployment
- **Month 2:** CI/CD, secrets, monitoring
- **Month 3:** Reliability, scaling, production readiness

✅ Phase 5 Exit Criteria (Critical)

You are **Phase-5 complete** if you can:

- Deploy backend systems confidently
- Use Docker professionally
- Build CI/CD pipelines
- Monitor production systems
- Handle failures safely
- Explain deployment architecture

🔥 After Phase 5, You Are:

- **Backend Engineer + DevOps-capable**
- Ready for:

  - Capstone project (Phase 6)
  - Real production backend roles
  - Startup & enterprise environments