🧪 Lab: Design and Present a Scalable AI Infrastructure

🎯 Objective

Learners will **design, document, and present** a scalable AI infrastructure that covers **data pipelines, GPU training clusters, deployment strategies, and monitoring/observability**. The goal is to demonstrate architectural thinking, tool selection, and scalability planning in a **real-world enterprise scenario**.

🔷 Step 1: Define the Use Case and Business Requirements

1. Choose a **domain**: Healthcare, finance, retail, autonomous vehicles, or smart cities.
2. Define **requirements**:
    - Latency expectations (real-time inference vs. batch processing).
    - Scale of data (TBs/day, streaming vs. static).
    - Compliance needs (GDPR, HIPAA, FedRAMP).
3. State **KPIs**: accuracy, throughput, cost per inference, uptime SLAs.

💡 *Explanation:* This ensures the infrastructure design is not abstract, but mapped to a business challenge with measurable goals.

🔷 Step 2: Design the Data Pipeline

1. **Ingest:** Identify sources (IoT sensors, EMR databases, transaction logs).
2. **Preprocess:** Specify tools (RAPIDS, Spark, Dask).
3. **Storage layer:** Decide between object storage (S3), shared file systems (NFS), or hybrid.
4. **Governance:** Include metadata tracking, version control, and lineage.

💡 *Explanation:* The pipeline must be reproducible, auditable, and performant, ensuring clean data flows into training systems.

🔷 Step 3: Architect the Training Environment

1. Select **GPU type**: A100/H100 for large-scale training, Jetson for edge cases.
2. Define **scaling strategy**: multi-GPU vs. multi-node clusters.
3. **Scheduler choice:** Kubernetes vs. Slurm.
4. Include **checkpointing and experiment tracking** (MLflow, Kubeflow).

💡 *Explanation:* This stage ensures the design accounts for both compute scale and fault tolerance.

🔷 Step 4: Plan the Deployment Strategy

1. Package the model using **ONNX/TensorRT**.
2. Choose serving solution: **Triton Inference Server**.
3. Decide deployment mode: **Cloud, on-prem, edge, or hybrid**.
4. Include autoscaling policies (Kubernetes HPA, GPU Operator).

💡 *Explanation:* The design must balance performance and cost-efficiency, with flexibility for scaling workloads.

🔷 Step 5: Define Monitoring & Governance Layer

1. Metrics: latency, throughput, GPU utilization, inference cost.
2. Drift detection: integrate model/data drift tools.
3. Telemetry: use Prometheus, Grafana, DCGM.
4. Incident response: define automated alerts + escalation policies.

💡 *Explanation:* Monitoring ensures resilience, compliance, and cost control—critical in enterprise workflows.

🔷 Step 6: Security & Compliance Considerations

1. RBAC: control access across users and teams.
2. Encryption: secure data in transit and at rest.
3. Regulatory alignment: map design choices to **GDPR, HIPAA, FedRAMP**.
4. Auditability: logs and lineage for accountability.

💡 *Explanation:* Infrastructure must meet legal, ethical, and business trust requirements.

🔷 Step 7: Create a Scalable Architecture Diagram

1. Draw a **system diagram** with:
    - Data sources → preprocessing → storage.
    - GPU training clusters.
    - Deployment layer (Triton, APIs).
    - Monitoring and governance stack.
2. Use **Lucidchart, Draw.io, or NVIDIA's diagrams**.

💡 *Explanation:* Visualization makes the design tangible and easier to communicate to stakeholders.

🔷 Step 8: Document the Design Choices

- Summarize why each tool/approach was chosen.
- Show trade-offs (e.g., Kubernetes vs. Slurm, cloud vs. hybrid).
- Highlight **scalability, cost, and compliance features**.

💡 *Explanation:* Writing forces clarity and prepares learners for stakeholder conversations.

🔷 Step 9: Present to a Panel (Peer Review or Instructor)

1. Deliver a **10–15 minute presentation**.
2. Cover:
    - Business problem.
    - End-to-end architecture.
    - Tool selection rationale.
    - Scalability & compliance considerations.
3. Receive **feedback and critique** from peers/instructors.

💡 *Explanation:* This mimics real-world proposal reviews in enterprise settings.

🔷 Step 10: Reflection & Iteration

- Identify **weaknesses** in your design based on feedback.
- Suggest **improvements or alternatives**.
- Document a **Version 2.0 architecture**.

💡 *Explanation:* Reflection and iteration build the mindset of continuous improvement, critical for enterprise AI success.

✅ Deliverables

- **Architecture diagram** of AI infrastructure.
- **Written design document** (3–5 pages).
- **Presentation deck** (5–7 slides).
- **Peer feedback summary** and revised design notes.
