Step 10: Deployment Architecture

- What are the major components of the system?
    - Client / User Interface
    - Flask API Service
    - ML Model & Preprocessing Pipeline
    - Container Layer (Docker)
    - Cloud Infrastructure (AWS EC2)
    - Logging and Monitoring Layer
- What are the inputs and outputs?
    - Inputs: annual_income, debt_to_income_ratio, credit_score, loan_amount interest_rate gender, marital_status, education_level, employment_status, loan_purposegrade_subgrade
    - Outputs: Probability of default and a binary decision
- Where and how will the data be stored?
    - Training data: Stored as CSV files or Parquet in AWS S3
    - Stored as serialized files (.pkl or .joblib) in S3
- How will data get from one component of the system to another?
    - Client sends HTTP POST request to /predict endpoint
    - Flask receives JSON payload
    - Input validation + preprocessing applied
    - Model predicts risk
    - Response returned as JSON
    - Logs written asynchronously
- What is the lifecycle of the ML/DL model?
    - The machine learning model follows a lifecycle that includes data collection, data cleaning and preprocessing, model training, model evaluation, serialization, deployment, monitoring, retraining, and redeployment.
- How frequently does the model need retraining? Is it at fixed intervals or when certain conditions are met?

    - The model will be retrained on a fixed monthly schedule.
- What kind of data is needed for retraining? How will the data be stored and managed?
    - Retraining uses newly collected loan application data and observed repayment outcomes, which are appended to historical datasets and stored in versioned snapshots for traceability.
- How will the retrained model be evaluated?
    - The retrained model will be evaluated on most recent production data using ROC-AUC.
- How will the retrained model be deployed?
    - Train new model > Validate metrics exceed thresholds > Serialize model with version tag > Build new Docker image > Push to registry > Redeploy container on AWS
- How will the retrained model be stored as an artifact?
    - It will be stored as .joblib or .pkl files
- How will the system be monitored and debugged?
    - The system is monitored through application logs, error tracking, and basic statistical monitoring of input feature distributions and prediction outputs.
- What are the specific tools/technologies that will be used to build this system?
    - The system is built using Python, pandas, and scikit-learn for data processing and modeling, Flask for the API layer, Docker for containerization, AWS EC2 and S3 for infrastructure and storage, Git for version control, and AWS CloudWatch for logging and monitoring.
- What is the estimated implementation cost in terms of resources, time, and money as applicable?
    - Cost: $0 - $20
    - Time: ~2 weeks
    - Compute: Free-tier EC2