🧭 System Evolution (Levels 1 → 3)

```
flowchart LR
    subgraph L1[Level 1 • Local Docker MVP]
        A1[Ingest] --> B1[Features]
        B1 --> C1[Train]
        C1 --> D1[Infer]
        D1 --> E1[/predictions_daily (Parquet)/]
        C1 -.-> F1[/artifacts/model/]
        D1 --> G1[FastAPI /predict]
        D1 --> H1[Dashboard]
    end

    subgraph L2[Level 2 • AWS Serverless]
        A2[EventBridge Schedules] --> B2[ECS Fargate Tasks:
ingest→features→infer]
        B2 --> C2[S3 (Parquet)]
        C2 --> D2[API Gateway → Lambda/ECS]
        D2 --> E2[/predict]
        B2 -. weekly .-> F2[ECS Task: train]
        F2 -.-> G2[S3 artifacts + version tags]
    end

    subgraph L3[Level 3 • SageMaker + Regimes + Monitoring]
        A3[Macro Ingest] --> B3[curated.macro_release_aligned]
        B3 --> C3[Features + Regime GMM/HMM]
        C3 --> D3[SageMaker Pipeline Train/Eval]
        D3 --> E3[Model Registry]
        E3 --> F3[Conditional Deploy (Batch/RT)]
        C3 --> G3[Model Monitor + Drift (PSI/KS)]
        G3 --> H3[Retrain Trigger]
        I3[(Optional RAG)] --> J3[/qa • /daily-brief (cited)/]
    end

    L1 --> L2 --> L3
```

🧱 Level 1 — Local Docker MVP

```
flowchart LR
```

```
    subgraph Local
        I[services/ingest] --> F[services/features]
        F --> T[services/train]
        T --> N[services/infer]
        T -. SHAP .-> X[Top Drivers]
        N --> P[/S3 (or MinIO): predictions_daily/]
        I --> RM[/raw.market/]
        I --> CM[/curated.market_daily/]
        F --> FD[/features_daily/]
        T --> A[/artifacts/model/]

        N --> API[services/api (FastAPI)]
        API --> EP[/predict/]

        N --> DASH[services/dashboard (Streamlit)]
        DASH --> VIEW[Charts & SHAP]
    end
```

☁️ Level 2 — AWS Serverless Production

```
flowchart TB
    EB[EventBridge Cron] -->|22:05 UTC| ING[ECS Fargate Task: ingest]
    ING --> CM[curated.market_daily/ (S3)]
    EB -->|22:10 UTC| FEAT[ECS Fargate Task: features]
    FEAT --> FD[features_daily/ (S3)]
    EB -->|Fri 22:15 UTC| TRAIN[ECS Fargate Task: train]
    TRAIN --> ART[artifacts/model/<version> (S3)]
    EB -->|22:30 UTC| INFER[ECS Fargate Task: infer]
    INFER --> PRED[predictions_daily/ (S3)]
    PRED -. optional .-> DDB[(DynamoDB mirror)]

    APIGW[API Gateway] --> LAMBDA[Lambda (container) or ECS Service]
    LAMBDA -->|GET latest model| ART
    LAMBDA --> RESP[/predict JSON/]

    subgraph Observability
        CW[CloudWatch Logs/Metrics] --> ALARM[SNS Alarms: task fail, p95, 5xx]
    end
```

```
    subgraph CI/CD
        GH[GitHub Actions] --> ECR[ECR Images]
        GH --> TF[Terraform Apply]
    end
```

🧠 Level 3 — SageMaker + Regimes + Monitoring (+ RAG)

```
flowchart LR
    FRED[Macro Ingest (FRED/TE)] --> RAWM[raw.macro/]
    RAWM --> CURM[curated.macro_release_aligned/]

    CM[curated.market_daily/] --> FEAT[Feature Builder]
    CURM --> FEAT
    FEAT --> REG[Unsupervised Regimes (GMM)]
    REG --> FD[features_daily/ (with regime + macro cols)]

    subgraph SageMaker Pipeline
        FD --> TRN[Train + Eval]
        TRN --> MET[Metrics (RMSE/QLIKE, Brier/AUC)]
        TRN --> ART[Model Artifacts (S3)]
        MET --> REGISTRY[Model Registry]
        REGISTRY -->|Pass gate| DEPLOY[Conditional Deploy
(Batch/RT)]
    end

    subgraph Monitoring
        FD --> MON[Model Monitor + Great Expectations]
        MON --> DRIFT[PSI/KS alerts]
        DRIFT --> RETRAIN[Trigger Pipeline]
    end

    subgraph Optional RAG
        DOCS[Docs ingest] --> CHUNK[Chunk + Embed]
        CHUNK --> VECDB[(OpenSearch/pgvector)]
        VECDB --> RAGAPI[/qa, /daily-brief (cited)/]
    end

    DEPLOY --> PRED[predictions_daily/]
```

🧪 (Bonus) CI/CD Pipeline — Source to Deploy

```
sequenceDiagram
    participant Dev as Developer
    participant GH as GitHub Actions
    participant ECR as AWS ECR
    participant TF as Terraform
    participant AWS as AWS (ECS/Lambda/APIGW)

    Dev->>GH: Push/PR to main
    GH->>GH: Run unit/integration tests
    GH->>ECR: Build & push images (tag: git SHA)
    GH->>TF: terraform plan & apply (with approval)
    TF->>AWS: Create/Update ECS Tasks, EventBridge rules, API
Gateway, Lambda
    GH->>AWS: Smoke test /predict
    AWS-->>GH: 200 OK + latency metrics
```