*3 Môi trường triển khai
- Development
- Pre-production (Staging)
- Production

***3 Thành phần Pipeline**
- Build
- Test
- Deploy

*3 Cách triển khai dự án
- Service (Standalone)
- Container Standalone
- Container Orchestration (K8s)

***2 Phương pháp deploy**
- Build & Deploy cùng server
- Build & Deploy khác server

*4 Data Architecture
- Centralized (Monolithic DB)
- Database per Service
- Event Sourcing (CQRS)
- Data Mesh (Domain-oriented)

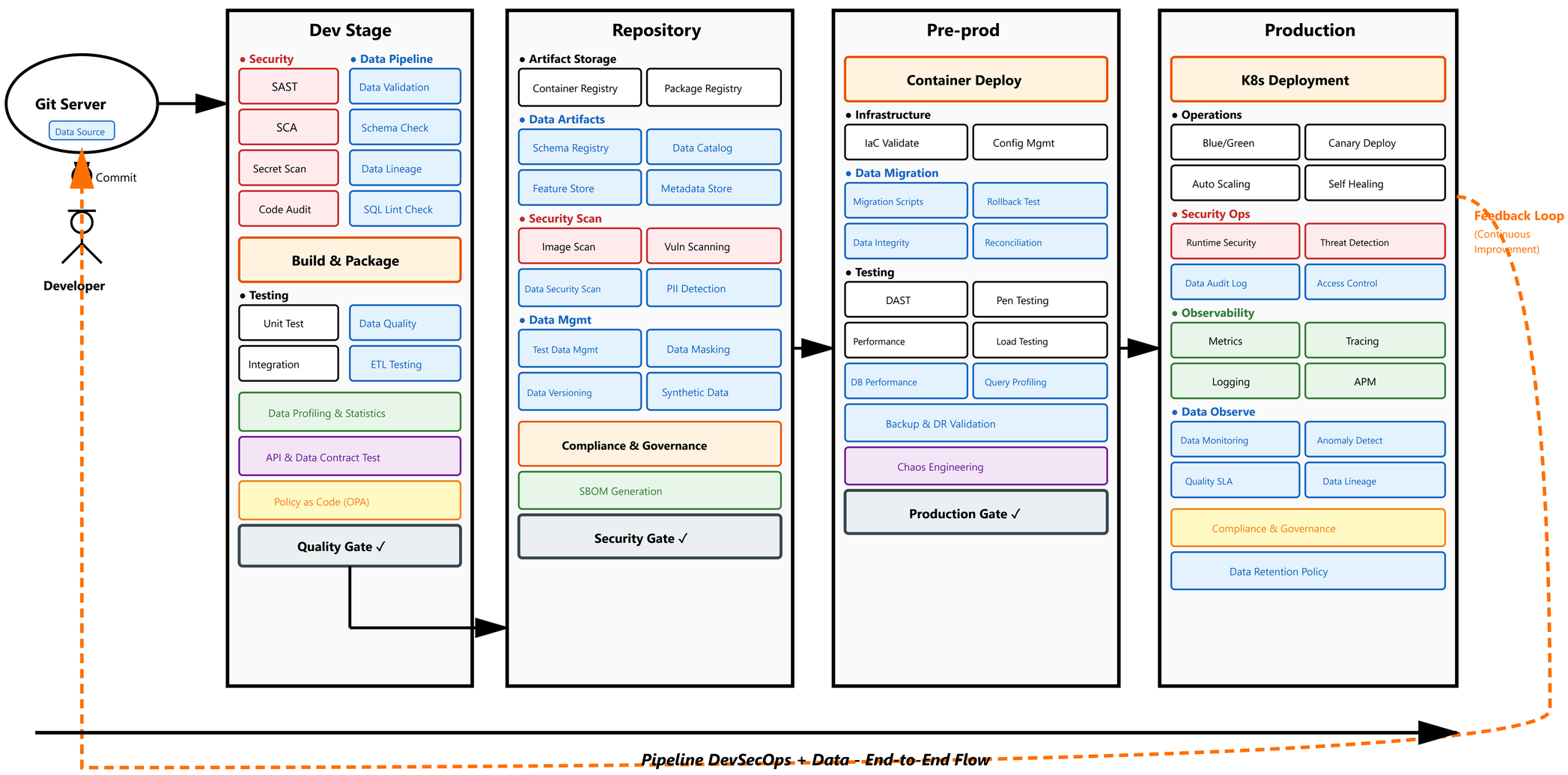
+ Data Pipeline Layers:
Ingestion → Processing → Storage → Serving → Governance

*Pipeline DevSecOps + Data Workflow
App: Dev → Commit → SAST → SCA → Build → Artifact → Image Scan → Deploy → DAST → Release → Monitor

Data: Validate → Schema → Quality → Catalog → Security → Migration → Integrity → Performance → Monitor → Anomaly Detect → Lineage → Governance

Security: Secret Scan → Compliance → Audit → Threat Detection

Observability: Metrics → Logs → Traces → Alerts



*Pipeline GIT + Data Workflow

Git Flow:

Master → Hotfix → Release → Dev → Feature → Config

Build → Scan Src → Scan Image → Deploy → Scan Website → Performance Test (K6) → Report (Telegram)

Data Git Flow:
- Schema Version Control (Liquibase/Flyway) → Migration Scripts → Code Review → Quality Gate
- Data Quality Rules as Code → dbt tests → Great Expectations → Pre-commit hooks
- Feature Branch Data Validation → Automated Data Testing → Merge to Main

*Pipeline K8s + Data Workflow

K8s Architecture:

Traffic → Ingress (routing, SSL) → Service (ClusterIP/LoadBalancer) → Deployment → ReplicaSet → Pods

• Deployment: Rolling update, Rollback, Auto-scaling (HPA) | • ConfigMap + Secret: Configuration management

Data on K8s:
- StatefulSets: PostgreSQL, MongoDB, Cassandra, Kafka | • PV/PVC: Persistent storage (dynamic provisioning)
- Operators: postgres-operator, mongodb-operator, kafka-operator | • Init Containers: Schema migration
- Backup: Velero for K8s backup | • Service Mesh: Istio for data traffic management

*Pipeline Microservices + Data Architecture Patterns

Service Deployment Order:

1. Infrastructure: Discovery Server (Eureka/Consul) → Config Server (Spring Cloud Config)
2. Data Layer: PostgreSQL → MySQL → MongoDB → Redis → Kafka
3. Security: Keycloak/Auth Service → API Gateway (Kong/Spring Cloud Gateway)
4. Business Services: User Service → Order Service → Payment Service → Notification Service

Data Architecture Patterns:

1. Database per Service
OrderService → order_db | UserService → user_db
✓ Independence ✓ Technology flexibility
✗ Distributed transactions ✗ Join queries
Solution: Saga Pattern, API Composition

2. Event-Driven Architecture (EDA)
Service A → Kafka Topic → [Service B, C, D]
✓ Decoupling ✓ Scalability ✓ Audit trail
✗ Eventual consistency ✗ Debugging complexity
Tools: Kafka, RabbitMQ, Event Sourcing + CQRS

3. Saga Pattern (Distributed TX)
Choreography: Event-based | Orchestration: Central coordinator
✓ Handles distributed transactions ✓ Rollback/Compensation
✗ Complex implementation ✗ Testing difficulty
Example: Order → Payment → Inventory (compensate if fail)

4. Data Mesh
Domain-oriented data ownership → Data as Product
✓ Domain expertise ✓ Scalable governance ✓ Self-serve
✗ Initial overhead ✗ Cultural shift required
4 Pillars: Domain, Product, Platform, Governance

5. API Gateway Data Aggregation:
Client → API Gateway → [Service A, B, C] → Aggregate → Response (GraphQL Federation, BFF pattern)

6. Change Data Capture (CDC):
Database → Debezium → Kafka → Stream Processing → Target Systems (Real-time sync)

Tools & Technologies:
Service Discovery: Consul, Eureka | Config: Spring Cloud Config | Messaging: Kafka, RabbitMQ, NATS

*Pipeline Machine Learning + MLOps Data Workflow

ML Pipeline Stages:

Data Collection → EDA → Preprocessing → Feature Engineering → Model Building → Evaluation → Deployment → Monitoring

MLOps Data Components:

1. Feature Store
Feast, Tecton, Hopsworks
✓ Consistent features ✓ Low latency ✓ Time-travel

2. Data Validation
TFDV, Great Expectations, Pandera
✓ Schema validation ✓ Drift detection

3. Model Registry
MLflow, DVC, Weights & Biases
✓ Versioning ✓ Lineage ✓ Reproducibility

4. Model Monitoring
Evidently AI, WhyLabs, Arize
✓ Data/Model/Concept drift ✓ Performance tracking

Orchestration & Deployment:

Airflow (Batch) | Kubeflow (K8s-native) | Metaflow (Production) | Shadow → Canary → A/B → Full Rollout

CI/CD for ML: Data versioning (DVC) → Training pipeline → Validation → Registry → Deploy → Monitor → Retrain

Chú thích & Color Coding:

Security | Data Pipeline | Observability | Build/Deploy | Testing | Governance

Essential Tools:

Quality: Great Expectations, dbt, Soda | Schema: Liquibase, Flyway
Security: Vault, Snyk, Trivy, OPA | Observability: Prometheus, Grafana, ELK
Data: Kafka, Debezium, Airbyte, Airflow | ML: MLflow, Kubeflow, Feast
Container: Docker, K8s, Helm, ArgoCD | CI/CD: GitLab CI, GitHub Actions

Best Practices:

1. Shift-Left Security & Data Quality
✓ Early validation ✓ Pre-commit hooks ✓ Fail fast

2. Infrastructure as Code (IaC)
✓ Terraform/Pulumi ✓ GitOps (ArgoCD) ✓ Version everything

3. Observability-Driven Development
✓ Metrics/logs/traces from day 1 ✓ SLO/SLI definition

4. Data Contract & Schema Evolution
✓ Define contracts ✓ Backward compatibility ✓ Version schemas

5. Progressive Delivery
✓ Feature flags ✓ Canary ✓ Blue/Green ✓ Auto rollback

Implementation Roadmap - 12 Week Plan:

Phase 1: Foundation (Weeks 1-3)
✓ CI/CD setup | Container registry | K8s cluster | Monitoring baseline (Prometheus/Grafana)
✓ Schema Registry | Data Catalog | Version control (DVC) | Secret management (Vault)

Phase 2: Security & Quality (Weeks 4-6)
✓ SAST/SCA (SonarQube, Snyk) | Secret scanning | DAST (OWASP ZAP) | Image scanning (Trivy)
✓ Data validation (Great Expectations) | PII detection | Data masking | Quality tests (dbt)

Phase 3: Testing & Migration (Weeks 7-9)
✓ Automated testing | Performance (K6) | Contract testing | Chaos engineering
✓ Migration scripts (Liquibase) | Integrity checks | Reconciliation | Backup/DR (Velero)

Phase 4: Production & Observability (Weeks 10-11)
✓ Blue/Green, Canary | Auto-scaling | Self-healing | GitOps (ArgoCD)
✓ Tracing (Jaeger) | APM (Datadog) | Alerting (PagerDuty)
✓ Data monitoring dashboard | Anomaly detection | Quality SLA | Lineage viz

Phase 5: Advanced (Week 12)
✓ Service mesh (Istio) | Feature flags | Cost optimization | Documentation
✓ Event-driven (Kafka) | CQRS | Data Mesh | ML pipeline integration

Success Metrics (DORA + Data):
- Deployment: Multiple/day | Lead time: <1h | MTTR: <30min | Change failure: <5%
- Data quality: ≥95% | Availability: 99.9% | Compliance: 100% | Detection: <5min