CI/CD PIPELINE & MONITORING STRATEGY

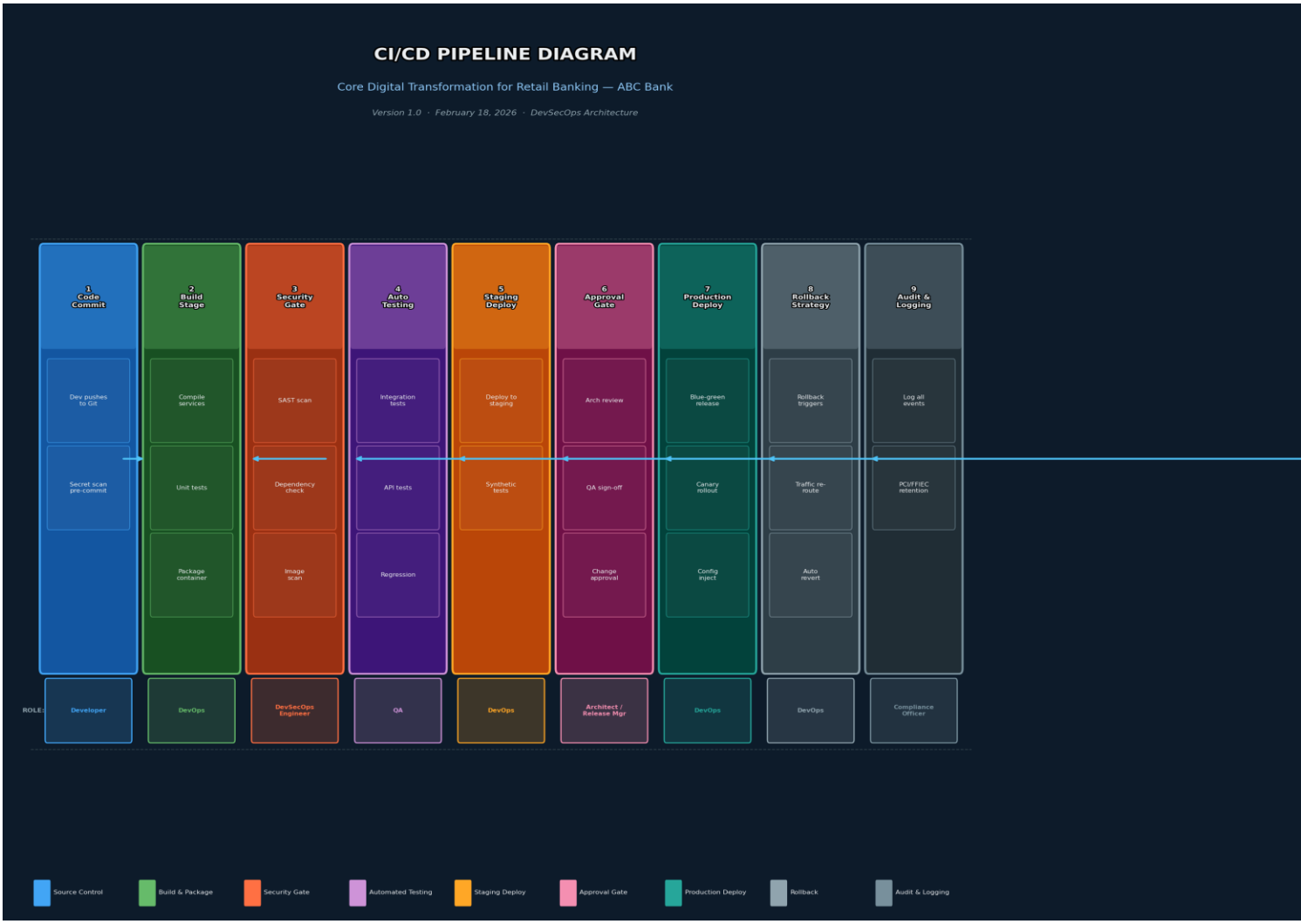
Core Digital Transformation for Retail Banking — ABC Bank

DevSecOps Architecture · Version 1.0 · February 18, 2026

Date: Feb 18, 2026	Prepared By: DevOps Architect	Status: Approved	Environment: Cloud-Native

Task 1: CI/CD Pipeline Diagram

The pipeline below implements nine stages for safe, automated, and compliant delivery of ABC Bank's core banking microservices. Each stage has an assigned role and defined control points enforcing DevSecOps, regulatory compliance (PCI-DSS, FFIEC), and zero-downtime deployment strategies.



CI/CD Stage	Responsible Role	Key Control Point
1. Code Commit — Source & secret scanning	Developer	*Pre-commit secret scan blocks credentials in code*

2. Build Stage — Compile, unit test, package	DevOps Engineer	*Build fails if unit test coverage < 80%*
3. Security Gate — SAST, dependency, image scan	DevSecOps Engineer	*Critical CVEs block pipeline progression*
4. Automated Testing — Integration, API, regression	QA Engineer	*All 3 test suites must pass with zero failures*
5. Staging Deployment — Deploy + synthetic tests	DevOps Engineer	*Synthetic test suite validates staging health*
6. Approval Gate — Arch, QA, change-management	Architect / Release Mgr	*Manual 3-way sign-off required before production*
7. Production Deployment — Blue-green / canary	DevOps Engineer	*Traffic shifted 10% → 50% → 100% over 30 minutes*
8. Rollback Strategy — Triggers, re-route, auto-revert	DevOps Engineer	*Automated revert if error rate exceeds 1%*
9. Audit & Compliance Logging	Compliance Officer	*All events retained per PCI-DSS & FFIEC requirements*

Task 2: Deployment Risk Strategy

The following table documents five deployment risks for the high-availability banking environment, with corresponding triggers, mitigation strategies, and rollback mechanisms.

Risk	Trigger	Mitigation Strategy	Rollback Plan
Deployment Failure	Failed health checks on ≥2 pods within 5 minutes of deploy	Canary deployment: route 10% traffic first; monitor error rate for 10 minutes before promoting	Auto-revert to previous container image version; Kubernetes rollout undo command
API Incompatibility	4xx / 5xx spike >2% on API Gateway immediately after release	Contract testing (Pact) between all microservices; versioned API endpoints with backward-compatibility checks	Blue-green swap: redirect all traffic to green (stable) environment within 60 seconds
Config / Secret Error	Service fails to start due to missing environment variable or misconfigured secret	Secret rotation via IBM Secrets Manager prior to deploy; config diff check in CI pipeline against known-good baseline	Redeploy with last known working config snapshot from version-controlled config store
Database Migration Failure	Schema migration script exits non-zero or data validation fails post-migration	Run migration in dry-run mode first; take automated DB snapshot before every migration; use backward-compatible schema changes	Restore from pre-migration snapshot; revert application to previous version; alert DBA on-call
Security Gate Regression	SAST scan detects new Critical CVE in	Pin base image versions; add CVE exceptions process with CCO	Block release; create P1 security ticket; deploy hotfix

	code that passed previous pipeline	approval; mandatory re-scan on every commit regardless of code delta	through emergency change process only

Task 3: System Monitoring Plan

Step 1: Key Performance Indicators (KPIs)

KPI	Definition	Target / Baseline
API Response Latency	P95 response time for all public API endpoints measured at API Gateway	< 200ms P95; < 500ms P99; alert if > 300ms sustained for 2 minutes
Service Availability (%)	Uptime of each microservice container measured per 30-day rolling window	≥ 99.9% SLA; alert if availability drops below 99.5%
Transaction Throughput (TPS)	Transactions processed per second across the payment and loan microservices	> 1,000 TPS baseline; alert if < 800 TPS during business hours
Error Rate (%)	Ratio of 5xx HTTP responses to total requests over 5-minute sliding window	< 0.1%; alert at > 0.5%; critical at > 1%
Fraud Rule Engine Time	Time taken by fraud detection service to evaluate and return a risk score	< 150ms per transaction; alert if > 200ms; critical if > 500ms
Database Replication Lag	Seconds behind primary for PostgreSQL read replicas used by reporting services	< 1 second; alert at > 5 seconds; critical at > 30 seconds

Step 2: Alert Triggers and Thresholds

KPI	Threshold	Alert Trigger	Severity	Notification Method
API Latency	> 300ms P95	Sustained > 300ms P95 for 2+ minutes	**High**	PagerDuty alert → On-call DevOps; Slack #alerts-prod
Availability	< 99.5%	Any 5-minute window with availability < 99.5%	**Critical**	PagerDuty P1 → DevOps + Architect + CCO; SMS to Release Manager
Error Rate	> 0.5%	5xx rate exceeds 0.5% over 5-min window	**Critical**	PagerDuty P1; auto-trigger rollback evaluation script
Transaction Throughput	< 800 TPS	TPS drops below 800 during 09:00–18:00 IST	**High**	Slack #ops-banking; Grafana alert; email to Operations Lead

Fraud Engine Latency	> 200ms	Fraud score response exceeds 200ms average	**High**	PagerDuty alert → Fraud Engineering team; Slack #fraud-alerts
DB Replication Lag	> 5 seconds	Replica lag exceeds 5s for more than 1 minute	**High**	PagerDuty → DBA on-call; Grafana dashboard annotation
DB Replication Lag	> 30 seconds	Replica lag exceeds 30s — potential data loss risk	**Critical**	PagerDuty P1; trigger read-replica failover; alert CCO

Step 3: Escalation Path

Level	Escalated To	Time to Respond
L1 — Alert	Automated monitoring system (Grafana / Datadog) triggers alert; On-call DevOps Engineer receives PagerDuty notification; Reviews dashboards and attempts immediate remediation	**< 5 minutes**
L2 — Incident	If unresolved in 5 minutes: DevOps Lead and Systems Architect paged; Incident bridge opened in Zoom; Kafka event published to incident-management topic; Rollback evaluated	**< 15 minutes**
L3 — Critical	If unresolved in 15 minutes: Release Manager + CTO notified; Compliance Officer paged for regulatory impact assessment; Customer Communication drafted; War-room initiated	**< 30 minutes**
L4 — Executive	If unresolved in 30 minutes or if regulatory impact confirmed: CEO / COO briefed; Regulatory authority notification prepared (per RBI/FFIEC mandate); Disaster Recovery Plan activated if data affected	**< 60 minutes**

Step 4: Post-Mortem Workflow

Phase	Activities
1. Incident Summary	Document: incident title, affected services, start/end timestamps, total customer impact (# of users affected, financial transactions delayed), severity level, and incident commander name.
2. Timeline Reconstruction	Reconstruct minute-by-minute timeline using Grafana logs, Kafka audit trail, Kubernetes event logs, and PagerDuty alert history. Identify the exact moment the issue manifested vs. when it was detected.
3. Root Cause Analysis (RCA)	Apply 5-Why methodology to identify the root cause. Distinguish between immediate cause, contributing factors, and systemic gaps. Use fishbone diagram if multiple contributing factors are present.

4. Fix Validation	Confirm that the fix was applied and verified in staging before production. Validate that all affected KPIs have returned to baseline. Confirm no secondary incidents were triggered by the fix.
5. Prevention Actions	Define 3–5 concrete action items with owners, due dates, and success criteria. Categorize: immediate (< 1 week), short-term (< 1 month), strategic (< 1 quarter). Add to sprint backlog.
6. Compliance Reporting	File incident report with Compliance Officer within 24 hours for any incident affecting transactions, KYC data, or AML systems. Retain all post-mortem artefacts for minimum 7 years per PCI-DSS Requirement 10.7.

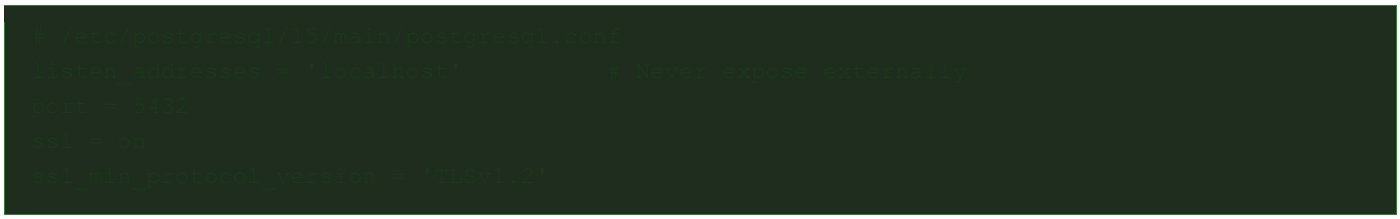
Configuration Guide: Database, API Gateway & CI/CD Components

This section provides configuration reference guides for the three primary infrastructure components: PostgreSQL database, the API Gateway (Kong/NGINX), and the Jenkins/GitHub Actions CI/CD components. All configurations are aligned with PCI-DSS, GDPR, and FFIEC compliance requirements.

A1: PostgreSQL Database Configuration Guide

Config Area	Configuration Detail
Connection Pooling	Use PgBouncer in transaction pooling mode. Max pool size: 100 connections per service. Timeout: 30 seconds. Configure separate pools per microservice (auth-db-pool, loan-db-pool, onboarding-db-pool).
SSL/TLS Encryption	Enforce ssl = on in postgresql.conf. Set ssl_min_protocol_version = TLSv1.2. All client connections must use sslmode=require. Certificate rotation every 90 days via IBM Secrets Manager.
Role-Based Access	Each microservice connects with a dedicated read/write role (auth_rw, loan_rw, onboard_rw). Compliance reporting uses a read-only role (compliance_ro). No service uses superuser credentials.
Write-Ahead Logging	wal_level = replica. max_wal_senders = 5. archive_mode = on. wal_keep_size = 1GB. Enables point-in-time recovery (PITR) and streaming replication to standby.
Performance Tuning	shared_buffers = 25% of RAM. work_mem = 64MB per query. effective_cache_size = 75% of RAM. autovacuum = on with autovacuum_vacuum_scale_factor = 0.01 for high-churn transaction tables.
Audit Logging	Enable pgaudit extension. Log all DDL, WRITE, and ROLE operations. Log destination: syslog forwarded to centralized SIEM. Retain audit logs for 7 years per PCI-DSS Requirement 10.7.

Key postgresql.conf settings:



```
shared_buffers = 4GB                    # 25% of RAM
wal_level = replica
max_wal_senders = 3
archive_mode = on
archive_command = 'cp %p /var/lib/postgresql/archive/%f'
log_connections = on
log_disconnections = on
shared_preload_libraries = 'pgaudit'
pgaudit.log = 'ddl, write, role'
```

A2: API Gateway (Kong / NGINX) Configuration Guide

Config Area	Configuration Detail
TLS Termination	Terminate TLS at the API Gateway. Enforce TLS 1.2 minimum. Use HSTS header: Strict-Transport-Security: max-age=31536000; includeSubDomains. Redirect all HTTP traffic to HTTPS.
JWT Validation	Validate JWT tokens on every request using Kong JWT plugin. Secret stored in Kubernetes Secret mounted as environment variable. Token expiry: 15 minutes. Refresh token: 8 hours.
Rate Limiting	Global rate limit: 1,000 requests/minute per IP. Per-consumer limit: 200 requests/minute for authenticated users. Loan submission endpoint: 10 requests/minute per account. Return HTTP 429 on breach.
Service Routing	Route /api/auth/* → auth-service:3000. Route /api/loans/* → loan-service:3001. Route /api/onboard/* → onboarding-service:3002. Health check route: /health → all services in parallel.
Request Logging	Log all requests: method, path, status code, latency, consumer ID, IP. Forward to ELK stack / IBM Log Analysis. Exclude sensitive headers (Authorization, X-API-Key) from logs.
Circuit Breaker	Enable circuit breaker: trip after 5 consecutive failures within 30 seconds. Half-open state after 60 seconds. Full open after 2 successful health checks. Return 503 with Retry-After header.

Kong declarative config snippet (kong.yml):

```
_format_version: "3.0"
services:
  - name: auth-service
    url: http://auth-service:3000
    routes:
      - name: auth-route
        paths: ["/api/auth"]
    plugins:
      - name: jwt
      - name: rate-limiting
```

A3: CI/CD Component Configuration Guide (Jenkins / GitHub Actions)

Component	Configuration Detail
Source Control (Git)	Branch protection on main: require 2 PR approvals + passing CI checks. Signed commits enforced via GPG keys. Pre-commit hooks: detect-secrets, gitleaks for credential scanning.
Build Agent Config	Use ephemeral Docker-in-Docker (DinD) agents. Agent image: jenkins/inbound-agent:jdk17. CPU: 2 cores, RAM: 4GB per agent. Auto-scale agents using Kubernetes Pod Template; destroy after each job.
Secret Management	No secrets in Jenkinsfile or GitHub Actions YAML. Inject at runtime via HashiCorp Vault plugin (Jenkins) or GitHub Actions Secrets. Rotate all CI/CD secrets every 30 days.
SAST Configuration	Run SonarQube with Quality Gate: 0 Critical bugs, 0 Critical vulnerabilities, Coverage ≥ 80%. Use Trivy for container image scanning with CRITICAL severity threshold blocking promotion.
Artifact Registry	Push signed images to IBM Container Registry with image digest pinning. Tag format: {service}:{git-commit-sha}. Retain last 10 versions. Auto-delete images older than 90 days.
Deployment Config	Use Helm charts for Kubernetes deployments. Values per environment: values-staging.yaml, values-prod.yaml. Deployment strategy: RollingUpdate with maxSurge=1, maxUnavailable=0 for zero-downtime.

GitHub Actions pipeline snippet (.github/workflows/cicd.yml):

Backup and Disaster Recovery Plan

This section defines the backup strategy, recovery objectives, and disaster recovery procedures for ABC Bank's cloud-native digital core platform. All procedures are designed to meet RTO < 4 hours and RPO < 1 hour for Tier-1 banking systems, in compliance with RBI Business Continuity Guidelines and PCI-DSS Requirement 12.3.

B1: Recovery Objectives

Service / Component	Tier	RTO Target	RPO Target
Customer Onboarding Service	Tier 1 — Critical	< 1 hour	< 15 minutes
Loan Eligibility & Servicing	Tier 1 — Critical	< 1 hour	< 15 minutes
Transaction Processing	Tier 1 — Critical	< 30 minutes	< 5 minutes
Fraud Detection Engine	Tier 1 — Critical	< 30 minutes	< 5 minutes
Authentication Service	Tier 1 — Critical	< 15 minutes	< 5 minutes
PostgreSQL Primary DB	Tier 1 — Critical	< 1 hour	< 1 minute (WAL streaming)
Kafka Event Bus	Tier 1 — Critical	< 2 hours	< 15 minutes
AML Monitoring Service	Tier 2 — Important	< 4 hours	< 1 hour
Compliance Reporting DB	Tier 2 — Important	< 4 hours	< 1 hour

Analytics / Tableau Dashboards	**Tier 3 — Standard**	**< 8 hours**	**< 4 hours**

B2: Backup Schedule and Retention Policy

Backup Type	Frequency	Storage Location	Retention	Encryption
PostgreSQL Full Backup (pg_basebackup)	Daily — 02:00 UTC	IBM Cloud Object Storage (cross-region)	**35 days**	AES-256
PostgreSQL WAL Archiving (PITR)	Continuous — every 5 minutes	IBM Cloud Object Storage (primary region)	**7 days**	AES-256
Kafka Topic Snapshots	Every 6 hours	IBM Cloud Object Storage (cross-region)	**30 days**	AES-256
Container Image Snapshots	On every production deploy	IBM Container Registry (2 regions)	**10 versions**	SHA-256 signed
Kubernetes etcd Backup	Every 4 hours	Encrypted S3-compatible store	**30 days**	AES-256
Application Config / Secrets Vault	On every change	HashiCorp Vault (replicated standby)	**Indefinite**	AES-256 + TLS

B3: Disaster Recovery Procedures

Step	Phase	Procedure Detail
1	**Declare Disaster**	Incident Commander declares DR event when RTO is at risk or system-wide outage exceeds 30 minutes. Notify: CTO, Release Manager, Compliance Officer, DBA on-call. Open Zoom war-room bridge.
2	**Activate DR Environment**	Spin up pre-provisioned DR Kubernetes cluster in secondary IBM Cloud region (e.g., us-south → eu-gb). DNS failover via IBM Cloud Internet Services. Expected switchover: < 20 minutes.
3	**Database Recovery**	For PostgreSQL: promote hot standby replica in DR region to primary. Verify data consistency by comparing row counts on key tables (transactions, accounts, kyc_records). Run validation script: db-dr-validate.sh.
4	**Kafka Recovery**	Restore Kafka topics from latest 6-hour snapshot. Replay missing events from WAL-backed transaction log. Validate consumer group offsets are aligned. Expected data loss: < 6 hours (RPO target met if < 15 min).
5	**Service Health Check**	Run synthetic test suite against DR environment endpoints. Validate: auth token issuance, KYC submission, loan eligibility call, fraud score return. All must return HTTP 200 before customer traffic is routed.

6	**Traffic Cutover**	Update DNS A records with 60-second TTL. Route 10% traffic to DR environment for 5 minutes. Monitor error rate and latency. If within SLA, route 100% traffic. Document cutover timestamp.
7	**Compliance Notification**	File incident report with Compliance Officer within 1 hour of DR activation. Prepare regulatory notification for RBI / FFIEC if customer data or transactions were impacted. Retain all DR activation logs.
8	**Recovery & Failback**	Once primary region is restored: run smoke tests on primary. Sync any data written to DR back to primary via bidirectional replication. Failback during off-peak hours (01:00–04:00 UTC). Post-mortem within 48 hours.

Task 4: Validation Checklist

Validation Item	Yes / No	Evidence
Pipeline includes all CI/CD stages (commit → build → security → test → staging → approval → deploy → rollback → audit)	**Yes**	*9 stages documented with roles, tools, and control points*
Security gates included (SAST scan, container image scan, dependency check)	**Yes**	*Stage 3: SonarQube SAST, Trivy image scan, OWASP dependency check*
Roles assigned for each stage	**Yes**	*All 9 stages have named responsible roles in the roles table*
Rollback strategy documented (triggers, traffic re-routing, automated revert)	**Yes**	*Stage 8 + Risk Strategy table: 5 risks with explicit rollback plans*
KPIs have thresholds and alerts (latency, availability, throughput, error rate)	**Yes**	*6 KPIs defined with thresholds, triggers, severity, and notification channels*
Escalation path defined (L1 → L4 with response times)	**Yes**	*4-level escalation: Alert → Incident → Critical → Executive*
Monitoring covers compliance and uptime (PCI-DSS audit logging, AML monitoring)	**Yes**	*KPI #6: DB replication; Audit & Compliance Logging stage; post-mortem compliance reporting*
Database configuration guide included (PostgreSQL)	**Yes**	*Section A1: 6 config areas + postgresql.conf snippet*
API Gateway configuration guide included (Kong / NGINX)	**Yes**	*Section A2: 6 config areas + kong.yml declarative config snippet*
CI/CD component configuration guide included (Jenkins / GitHub Actions)	**Yes**	*Section A3: 6 config areas + GitHub Actions workflow YAML snippet*
Backup and Disaster Recovery Plan included	**Yes**	*Section B: RTO/RPO targets, backup schedule, 8-step DR procedure*
CI/CD Pipeline diagram created and saved as PNG	**Yes**	*CICD_pipeline.png — 9-stage colour-coded diagram with roles and legend*