Annexure 10. Interim Hosting Environment Strategy

Architecture, Governance & Migration Proposal

Until Onboarding to the GovCloud

Document Control

Document Information

Document Title	Interim Hosting Solution – Architecture, Governance & Migration Proposal
Document Reference	TBD — to be assigned by Document Control
Version	0.4 (Draft)
Classification	Internal / Restricted — to be confirmed with Document Control
Prepared By	TBD (Name / Role)
CI/CD Section Prepared By	DSD Team — DevOps Expert
Cluster Creation	DSD Team — DevOps Expert
Migration Strategy	DSD Team — DevOps Expert
Reviewed By	TBD
Approved By	TBD
Date Issued	20 July 2026
Status	Draft — Interim Solution, until onboarding to GovCloud

Version History

Version	Date	Author	Description of Change
0.1	20 Jul 2026	TBD	Initial draft submitted as interim solution proposal
0.2	20 Jul 2026	DSD Team — DevOps SME	Expanded CI/CD Pipeline section (5.3) with detailed stage-by-stage pipeline design, tooling, and promotion gates
0.3	20 Jul 2026	Advisory Team	Added Environmental Strategy section (Section 2) covering objectives for GovStack Core Team, Cloud Division, BNSW, GovStack vendor, GA systems read replica, interim architecture, expected GovCloud, and

Version	Date	Author	Description of Change
			migration strategy; added always-on environment requirement (Section 4.5)
0.4	20 Jul 2026	DSD Team — DevOps Expert	Consolidated DSD Team edits: attributed Cluster Creation and Migration Strategy sections to DSD Team — DevOps Expert; clarified GovCloud as hosted on premises; added CI/CD scope reference; updated Distribution List purposes for Infrastructure/Cloud Team and Application Development Team / DSD

Distribution List

Recipient / Team	Purpose
Documents Control	Official record and version control
GovStack Core Team	Guidance on where, how, and when building block environments are required
Cloud Division	Input for provisioning and sizing decisions for GovStack/BNSW environments
BNSW Vendor	Non-functional requirement reference for BNSW hosting environment and GovStack integration
GovStack Vendor	Guidance on how GovStack should be hosted in GDC
Infrastructure / Cloud Team	Network design and Infra provisioning
Application Development Team / DSD	Compliance with statelessness and CI/CD requirements, and implementation of cluster
Security / Compliance	Review of network segmentation and access control
GovTech GDC Onboarding Team	Reference for future migration to GovCloud

Table of Contents

1. Introduction....6
    1.1 Purpose....6
    1.2 Background....6
    1.3 Scope....6
    1.4 Definitions & Abbreviations....6
2. Environmental Strategy — Objectives & Stakeholder Alignment....7
    2.1 Objective 1 — For the GovStack Core Team....7
    2.2 Objective 2 — For the Cloud Division....7
    2.3 Objective 3 — For BNSW (National Single Window)....8
    2.4 Objective 4 — For the GovStack Vendor....9
    2.5 Objective 5 — Data Access to Government Agency (GA) Systems....9
    2.6 Objective 6 — Interim Hosting Environment Architecture for GovStack and BNSW....10
    2.7 Objective 7 — Expected Government Cloud....10
    2.8 Objective 8 — Migration Strategy: Interim Hosting Environment to Full-Fledged Cloud-Native Environment....10
3. Solution Overview....10
4. Cluster Architecture....10
    4.1 Cluster Topology....11
    4.2 Environment Isolation Controls (Cluster A)....11
    4.3 Production Cluster Controls (Cluster B)....11
    4.4 High-Level Cluster & Network Diagram....12
    4.5 Always-On Environments — Code Availability Across All Environments....12
5. Network Architecture — VLAN Design....13
6. Source Code Management....13
    6.1 Single Source of Truth — Official GovTech GitLab....13
    6.2 Repository & Branching Strategy....14
    6.3 CI/CD Pipeline....14
        6.3.1 Pipeline Design Principles....14
        6.3.2 Pipeline Stages....15
        6.3.3 Environment Promotion Gates....16
        6.3.4 Tooling Summary....16
        6.3.5 Rollback within CI/CD....17
    6.4 Access Control....17
7. Application Design Principles — Statelessness....17
    7.1 Requirement....18
    7.2 Implementation Guidelines....18

7.3 Benefit to Migration....18
8. Migration Strategy — Interim Environment to GovCloud....18
8.1 Guiding Principles.... 19
8.2 Pre-Migration Readiness Checklist....19
8.3 Migration Phases....20
8.4 Cutover Approach....20
8.5 Rollback Plan....21
8.6 Indicative Timeline.... 21
9. Roles & Responsibilities....22
10. Risks & Mitigations....22
11. Assumptions & Dependencies.... 23
12. Appendix....23
A. Glossary.... 23
B. References....23
C. Open Items for Documents Control.... 23

1. Introduction

1.1 Purpose

This document defines the architecture, governance controls, and migration approach for an interim hosting solution to be used until onboarding to the GovCloud is completed. It is submitted to Document Control as the reference design for review, approval, and future audit.

1.2 Background

Pending full availability of GovCloud infrastructure and completion of the associated onboarding process, the application(s) in scope require an interim hosting environment that (a) meets baseline security and segregation requirements, (b) keeps source code under official GovTech governance, and (c) is architected from day one so that cutover to GovCloud can be executed with minimal disruption.

1.3 Scope

This document covers:

- Cluster architecture for non-production and production environments
- Network segmentation (VLAN design)
- Source code management and governance via the official GovTech GitLab
- Application design principles (statelessness) required for portability
- The migration strategy and cutover plan from the interim environment to GovCloud
- Use of the CI/CD pipeline created by the DSD team

Out of scope: detailed GovCloud architecture (to be issued by GovTech upon onboarding), hardware/data-center facility specifications, and budget/costing.

1.4 Definitions & Abbreviations

Term	Definition
GDC	Government Data Center — the target sovereign hosting facility/platform
GovCloud	Government-operated cloud platform hosted on premises
UAT	User Acceptance Testing environment
STG	Staging environment, used for pre-production validation
VLAN	Virtual Local Area Network, used for logical network segmentation
GitLab (GovTech)	The official, centrally governed source code management platform operated by GovTech

Term	Definition
Stateless Application	An application that stores no session or persistent state on the local compute instance, enabling any instance to serve any request
CI/CD	Continuous Integration / Continuous Deployment pipeline

2. Environmental Strategy — Objectives & Stakeholder Alignment

This Environmental Strategy is prepared to serve as a single reference point that different stakeholders can rely on to understand where, how, and when environments are required for GovStack building blocks and BNSW (National Single Window), how these systems are hosted, and how they migrate toward the full-fledged Government Cloud/GovCloud. The objectives below define who this document is for and what each stakeholder should be able to obtain from it.

2.1 Objective 1 — For the GovStack Core Team

This document informs the GovStack Core Team where, how, and when environments for each GovStack building block are required across the delivery lifecycle:

- Where: environments are hosted in the interim clusters described in Section 4 (Cluster A for Dev/UAT/Staging, Cluster B for Production) until GovCloud onboarding is complete.
- How: each building block is deployed as a stateless, containerized workload via the standard GitLab CI/CD pipeline described in Section 6, ensuring consistent build artifacts across every environment.
- When**:** Environment provisioning will begin in preparation for the developer onboarding scheduled for **November 2026**. The clusters must be fully ready by **mid-October 2026**, with stable, thoroughly tested environments available for Development, QA, and Staging (STG).

2.2 Objective 2 — For the Cloud Division

This document acts as an input for the Cloud Division to plan and decide on the required environments for GovStack:

- Provides the cluster topology (Section 4), VLAN/network segmentation (Section 5), and capacity/sizing placeholders that the Cloud Division uses to size and provision infrastructure.

- Signals the timing of environment requirements in accordance with **Objective 1**, enabling the Cloud Division to proactively plan and provision the required infrastructure ahead of GovStack Core Team milestones. The Cloud Division will allocate and provision the necessary resources based on the infrastructure requirements and solution architecture defined by the GovStack Team
- Provides the non-functional baseline (statelessness, always-on environments, CI/CD-driven deployment) that the Cloud Division should account for in capacity and cost planning for both the interim environment and eventual GovCloud.

2.3 Objective 3 — For BNSW (National Single Window)

This document forms part of the **Non-Functional Requirements (NFR)** provided to the BNSW vendor. It enables the vendor to understand:

- The hosting environment for **BNSW**, which will consist of **QA, Staging, and Production** environments, including the cluster topology, network segmentation, infrastructure architecture, and high-availability requirements.
- How GovStack — the building blocks BNSW integrates with — is hosted, so the vendor can design BNSW's integration and connectivity approach accordingly.
- The data-consumption model for GA (Government Agency) systems described in Objective 5, which directly affects BNSW's integration architecture.
- The hosting environment for **GovStack**, which consists of **Development, QA, Staging, and Production** environments deployed across **two Kubernetes clusters and a database cluster**:

  i. **Non-Production Cluster:** Development, QA, and Staging

  ii. **Production Cluster:** Production Application and Production database
- The cluster topology, network segmentation, infrastructure architecture, high-availability design, and environment separation for both BNSW and GovStack to ensure secure and reliable deployment.
- All GovStack environments (Development, QA, Staging, and Production) are expected to remain continuously available to support ongoing development, integration, testing, validation, and production operations. Since BNSW integrates with GovStack building blocks, the vendor shall design BNSW's deployment, connectivity, and integration approach based on the GovStack environment architecture and the separation of the two Kubernetes clusters for application and database cluster.

2.4 Objective 4 — For the GovStack Vendor

This document informs the GovStack vendor how GovStack should be hosted in the GDC:

- Target hosting model: containerized, stateless workloads deployed to segregated non-production and production clusters, consistent with Sections 4, 5, and 7.

- Source code and pipeline expectations: all GovStack source code maintained exclusively in the official GovTech GitLab, deployed via the standard CI/CD pipeline (Section 6).
- Expectation that the vendor's build is portable from the interim environment to GovCloud with no architectural rework or with minimum configuration changes only, as per the Migration Strategy in Section 8.

2.5 Objective 5 — Data Access to Government Agency (GA) Systems

BNSW shall consume data from Government Agency (GA) systems through standardized APIs exposed via the **National Data Exchange (NDE)**. Direct database access to GA systems, including read replicas, shall not be permitted, in line with modern service-oriented and API-first architecture principles.

- BNSW shall access GA data only through secure, authenticated, and authorized APIs published by the NDE.
- Government Agency systems shall expose the required business data through NDE-compliant APIs rather than allowing direct database connectivity.
- API communication shall be secured using appropriate authentication, authorization, encryption, and auditing mechanisms to ensure data confidentiality, integrity, and traceability.
- Network access shall be restricted so that only the NDE platform and authorized consumers can invoke the published APIs, following the least-privilege and network segmentation principles defined in Section 5.
- This API-based integration pattern shall remain unchanged when migrating to GovCloud, as it represents the target enterprise integration architecture independent of the underlying hosting environment.

Note: Direct database read access (including read-replica access) is intentionally not supported. While database read replicas can reduce load on production systems, the preferred enterprise architecture is to expose data through standardized NDE APIs. This approach improves security, governance, loose coupling, versioning, monitoring, and interoperability across Government Agency systems and BNSW.

2.6 Objective 6 — Interim Hosting Environment Architecture for GovStack and BNSW

The interim hosting environment architecture — Cluster A (Dev/UAT/Staging),Cluster B (Production) and Database Cluster, segmented across three or more VLANs, with all source code in the official GovTech GitLab and all workloads stateless — applies equally to GovStack building blocks and to BNSW. Full detail is provided in Sections 4 through 7 of this document, and both systems follow the same always-on environment model described in Section 4.5.

2.7 Objective 7 — Expected Government Cloud

The expected target state is the GovCloud, a government-operated cloud platform providing equivalent or stronger cluster segregation, network segmentation, and security controls than the interim environment. GovStack and BNSW are expected to onboard GovCloud once it is available, adopting the same architecture patterns established in the interim environment (stateless workloads, GitLab-governed source code, CI/CD-driven promotion, always-on environments) with infrastructure endpoints updated accordingly.

2.8 Objective 8 — Migration Strategy: Interim Hosting Environment to Full-Fledged Cloud-Native Environment

The detailed migration approach — covering readiness checklist, phased migration, cutover, and rollback — is provided in Section 8 (Migration Strategy — Interim Environment to GovCloud) and applies to both GovStack and BNSW as cloud-native, stateless workloads.

3. Solution Overview

The interim solution consists of two logically and physically separated Kubernetes/container clusters — one shared cluster for lower environments (Development, UAT, and Staging) and one dedicated cluster for Production — connected through segregated VLANs. All source code is version-controlled exclusively in the official GovTech GitLab instance, and all workloads are built to be stateless so that they can be re-platformed onto GovCloud with no architectural rework, only a change of underlying infrastructure and connection endpoints.

This design intentionally mirrors expected GovCloud conventions (environment isolation, network segmentation, centralized source control, immutable/stateless workloads) so the interim platform functions as a low-risk rehearsal for the eventual production platform.

4. Cluster Architecture

This section — Cluster Creation — is prepared by the DSD Team DevOps Expert and defines the cluster topology, isolation controls, and network diagram for the interim hosting environment.

4.1 Cluster Topology

Two clusters are provisioned:

Cluster	Environments Hosted	Purpose / Isolation Approach
Cluster A — Non-Production	Dev, UAT, Staging	Shared cluster; environments isolated using separate namespaces, RBAC boundaries, resource quotas, and network policies per environment. Lower SLA, used for build, test, and pre-release validation.
Cluster B — Production	Production	Dedicated, isolated cluster. No shared tenancy with lower environments. Hardened configuration, restricted access, higher SLA and monitoring/alerting thresholds.

Note: *Sizing (node count, CPU/RAM per node, autoscaling limits) is to be confirmed with the infrastructure team based on expected workload profiles and finalized prior to build.*

4.2 Environment Isolation Controls (Cluster A)

- Dedicated Kubernetes namespace per environment (dev / UAT/ staging)
- Namespace-scoped RBAC — engineers granted access only to the namespace(s) required for their role
- NetworkPolicy objects to prevent cross-namespace east-west traffic unless explicitly required
- Separate secrets/config stores per environment (no shared credentials across dev/UAT/staging)
- Resource quotas and limit ranges per namespace to prevent noisy-neighbor impact

4.3 Production Cluster Controls (Cluster B)

- Physically/logically separate cluster from non-production — no shared control plane or worker nodes
- Restricted administrative access (break-glass procedure + approval workflow for production changes)
- Mandatory change management and deployment approval gate prior to production release
- Enhanced logging, monitoring, and alerting with retention aligned to compliance requirements
- Backup and disaster-recovery procedures documented separately

4.4 High-Level Cluster & Network Diagram

Conceptual layout:

VLAN 1 — Non-Production	VLAN 2 — Production
Cluster A (shared) Dev namespace, UAT namespace, Staging namespace	Cluster B (dedicated) Production namespace, Restricted access only

4.5 Always-On Environments — Code Availability Across All Environments

All provisioned environments — Dev, UAT, Staging, and Production — are kept running continuously; none are spun up on demand or torn down between cycles. This means the application code (via its deployed container image) must be available and deployed in every environment at all times, not only in the environment currently under active test or release.

- Every environment (Dev, UAT, Staging, Production) always has the latest promoted build for that environment deployed and running — there is no environment left empty or stopped between releases.
- This ensures GovStack Core Team members, BNSW vendor teams, and testers can access any environment at any time without waiting for a fresh deployment or environment spin-up.
- Continuous availability is supported by the CI/CD promotion model in Section 6: each successful pipeline stage deploys directly to its target environment, keeping that environment's running workload current.
- Because workloads are stateless (Section 7), keeping all environments running does not require environment-specific data migration — only the container image and externalized configuration differ per environment.
- The Cloud Division should size cluster capacity (Section 4.1) assuming all environments run concurrently and continuously, rather than assuming only one active environment at a time.
- Access: There will be no direct communication between different environments. When GA systems and other systems, such as BNSW and Zhabtog Lekjur, need to integrate with GovStack, the access must follow the same environment segregation principle.For example, the Dev environment of external systems can only communicate with the Dev environment of GovStack. Similarly, QA, Staging, and Production environments of other systems should only access their corresponding environments in GovStack. Cross-environment communication will not be permitted.

Note: *This always-on requirement applies to the interim hosting environment and is expected to carry forward unchanged into GDC/GovCloud.*

5. Network Architecture — VLAN Design

Two VLANs provide the primary network segmentation boundary between non-production and production traffic.

VLAN	Maps To	Access & Controls
VLAN 1	Cluster A — Dev / UAT / Staging	Accessible to development, QA, and release teams via VPN/jump host. Outbound access restricted per environment policy. Firewall rules permit only required ports between tiers.
VLAN 2	Cluster B — Production	Access restricted to authorized operations/SRE personnel and CI/CD deployment service accounts only. Inbound traffic limited to the load balancer/ingress; all other ports denied by default (deny-all baseline, explicit allow list).

- Inter-VLAN routing is denied by default; any required exception (e.g., a controlled promotion pipeline) is explicitly firewalled and logged.
- VLAN IDs, subnet ranges, and firewall rule sets are to be finalized with the network team and recorded in an addendum to this document.
- Ingress/egress traffic is routed through a WAF / load balancer layer in front of each VLAN.

Note: *Specific subnet CIDR ranges, gateway addresses, and firewall rule numbers are placeholders pending network team sign-off and should be appended once confirmed.*

6. Source Code Management

6.1 Single Source of Truth — Official GovTech GitLab

All application source code, infrastructure-as-code, pipeline definitions, and configuration templates are maintained exclusively in the official GitLab instance operated by GovTech. No code is to be stored in personal, third-party,unmanaged repositories or vendor repositories. Environments related configuration and run book related documentation should be kept in Git-Lab along with read me manuals for each environment (Dev, QA, Staging and Production) step-by-step. No production keys or secrets should be kept in this.

6.2 Repository & Branching Strategy

- One repository per application/service (or a documented monorepo structure where applicable)
- Protected branches: main (or release) and develop, with merge-request-only changes
- Mandatory peer code review / merge-request approval before merging to protected branches
- Environment-specific branches or tags (dev, UAT, staging, prod-release) mapped to CI/CD deployment targets

- Documentation for each environment configuration and run-book should allow any one run this project without any proper knowledge.

Git Repository Naming Convention

The Git repository and branching strategy shall follow a service-based naming convention using camelCase to ensure consistency and clear identification across applications and deployment environments. Each application or microservice shall maintain a dedicated repository (e.g., ssoService, notificationService, documentService) unless a documented monorepo approach is approved. Source code management shall use protected branches such as main and develop, where all changes must be submitted through merge requests with mandatory peer review and approval before merging. Feature and bug-fix branches shall follow the naming convention feature/<featureName> and bugfix/<issueName> (e.g., feature/userLogin, bugfix/tokenExpiry). Environment-specific deployment branches or configurations shall follow the service/environment structure (e.g., ssoService/dev, ssoService/uat, ssoService/stg, and ssoService/prod) and shall be mapped to the respective CI/CD deployment pipelines. Production releases shall be controlled through approved release branches or version tags to ensure traceability, auditability, and controlled promotion of services across DEV, UAT, STAGING, and PRODUCTION environments.

6.3 CI/CD Pipeline

This section is prepared by the DSD Team DevOps Expert and defines the pipeline design used to build, test, secure, and promote application workloads through the interim environment, and later GovCloud, using GitLab CI/CD.

6.3.1 Pipeline Design Principles

- Pipeline-as-code — the full pipeline is defined in .gitlab-ci.yml and versioned alongside application code, subject to the same merge-request review process
- Every commit to a protected branch triggers an automated pipeline run; no manual/undocumented builds are promoted to any environment
- Immutable artifacts — a single container image is built once, scanned, and promoted unchanged across Dev → UAT → Staging → Production (no rebuilding per environment)
- Fail-fast — pipelines stop at the first failed stage (lint, test, scan) so defects are caught before promotion
- Least-privilege runners — GitLab Runners scoped per environment/VLAN, with production deployment credentials isolated from non-production runners

6.3.2 Pipeline Stages

Stage	Trigger	Activities
1. Source & Lint	On push / merge request	Checkout code from GovTech GitLab; run static code analysis and linting (e.g., SonarQube/ESLint/Checkstyle as applicable); enforce coding standards and merge-request approval rules
2. Build	On merge to develop/main	Compile application; build a versioned, immutable container image tagged with the commit SHA / semantic version
3. Unit & Integration Test	Automatic, post-build	Execute unit and integration test suites inside the pipeline; fail the pipeline on any test failure or coverage threshold breach
4. Security & Vulnerability Scan	Automatic, post-build	Scan the container image (e.g., Trivy/Grype) and dependencies (SCA) for known CVEs; scan source code for secrets (e.g., Gitleaks) prior to allowing promotion
5. Push to Registry	On scan pass	Push the signed, scanned image to the centrally governed container registry with immutable tagging
6. Deploy — Dev	Automatic	Deploy to the Dev namespace in Cluster A via Helm/Kustomize manifests stored in GitLab; smoke test executed automatically
7. Deploy — UAT	Automatic on Dev pass, or manual trigger	Promote the same image to the UAT namespace; notify QA/business testers via pipeline webhook
8. Deploy — Staging	Manual approval (QA sign-off)	Promote to Staging namespace for final pre-production validation, including performance/load testing where required
9. Deploy — Production	Manual approval gate (Change Management)	Deployment to Cluster B executed only after documented change approval; deployment performed by a restricted production service account with audit logging

Stage	Trigger	Activities
10. Post-Deploy Validation	Automatic, post-deploy	Automated health checks / synthetic transactions confirm the release is healthy; automatic rollback triggered on failed health checks where configured

6.3.3 Environment Promotion Gates

- Dev → UAT: automated, gated only by passing tests and security scans
- UAT → Staging: manual approval by QA/Business Analyst following UAT sign-off
- Staging → Production: manual approval gate tied to a documented Change Request; requires a second approver distinct from the deployer (four-eyes principle)

6.3.4 Tooling Summary

Function	Tooling (indicative — to be confirmed with DSD standards)
Source Control / CI Orchestration	Official GovTech GitLab (GitLab CI/CD)
Container Build	Docker / Buildah, produced as versioned immutable images
Container Registry	Centrally governed registry (e.g., GitLab Container Registry / Harbor)
Vulnerability & Secret Scanning	Trivy or Grype (image/SCA scanning), Gitleaks (secret detection)
Code Quality	SonarQube or equivalent static analysis
Deployment / Orchestration	Helm charts or Kustomize manifests, applied via GitLab CI to Kubernetes (Clusters A & B)
Secrets Management	GitLab CI/CD protected variables or an external secrets manager (e.g., HashiCorp Vault), never committed to source
Monitoring / Pipeline Observability	Pipeline success/failure metrics and deployment audit trail retained per GovTech policy

6.3.5 Rollback within CI/CD

- Because images are immutable and versioned, rollback is a re-deploy of the last known-good tagged image via the same pipeline — no rebuild required

- Production rollback follows the same change-approval process as forward deployment, executed on an expedited basis where an incident is declared

Note: *Section 5.3 (CI/CD Pipeline) is contributed by the DSD Team DevOps Subject Matter Expert and should be validated against current DSD pipeline standards, approved base images, and registry naming conventions before implementation.*

6.4 Access Control

- Access to the GitLab group/project governed by GovTech identity and access management policy
- Role-based permissions (Developer, Maintainer, Owner) aligned to least-privilege principles
- Audit logging of all repository and pipeline activity retained per GovTech policy

7. Application Design Principles — Statelessness

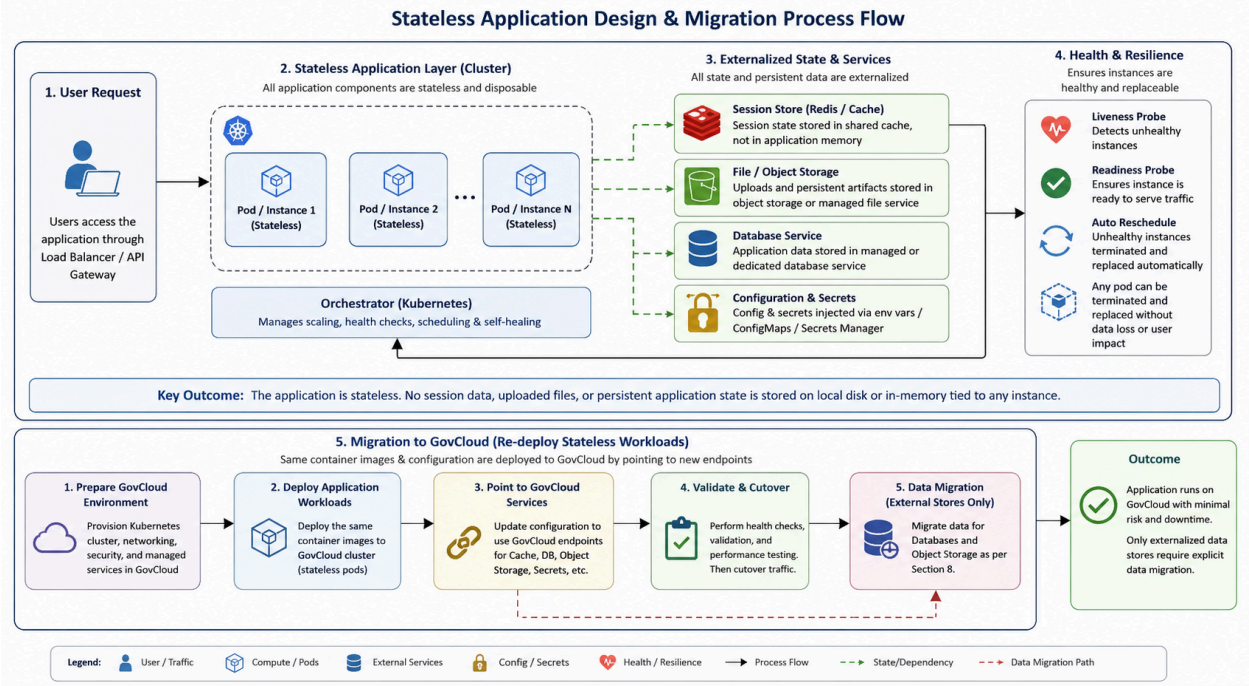


Fig1. Process Flow Diagram

7.1 Requirement

All application components deployed to either cluster must be stateless: no session data, uploaded files, or persistent application state may be stored on local container/pod filesystem or in-memory in a way that ties a user session to a specific instance. This is a hard prerequisite for horizontal scaling in the interim environment and for a clean, low-risk migration to GovCloud.

7.2 Implementation Guidelines

- Session state externalized to a shared cache (e.g., Redis) rather than in-process memory
- File uploads and persistent artifacts stored in object storage or a managed file service, not local disk
- Application configuration and secrets injected via environment variables/config maps/secrets manager, not baked into the image
- Database connections externalized to a managed or dedicated database service, not co-located with the application pod
- Pods designed to be disposable — any instance can be terminated and replaced without data loss or user impact
- Health checks (liveness/readiness probes) implemented so the orchestrator can safely reschedule instances

7.3 Benefit to Migration

Because no state is bound to a specific compute instance, workloads can be redeployed onto GovCloud infrastructure by pointing the same container images and configuration at new endpoints, without a complex state-transfer exercise for the application tier itself. Only externalized data stores (databases, object storage) require an explicit data migration step, described in Section 8.

8. Migration Strategy — Interim Environment to GovCloud

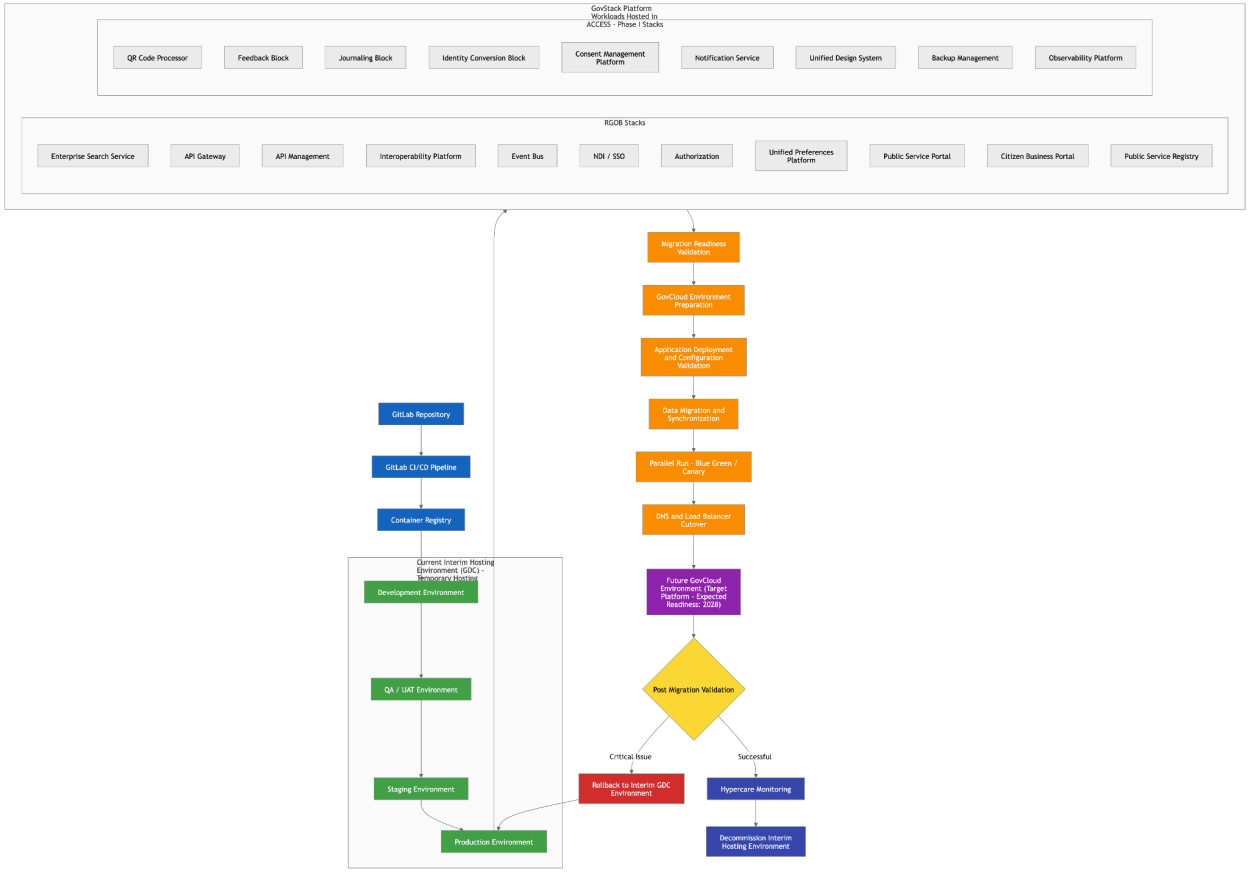


Fig2. Process Flow Diagram

This section — Migration Strategy — is prepared by the DSD Team DevOps Expert and defines the readiness checklist, phased migration approach, cutover method, and rollback plan for moving from the interim hosting environment to GovCloud.

8.1 Guiding Principles

- Zero or near-zero downtime for end users during cutover
- Phased, reversible migration — each phase has a defined rollback path
- Parity first — GovCloud environment is validated to match the interim environment before any traffic is shifted
- Data integrity is verified at every stage before the interim environment is decommissioned

8.2 Pre-Migration Readiness Checklist

1. GovCloud target clusters (non-prod and prod) provisioned and network-approved

2. VLAN/network segmentation in GovCloud confirmed equivalent to or stronger than the interim design
3. GitLab CI/CD pipelines updated with GovCloud deployment targets and credentials
4. Container registry access and image pull permissions established from GovCloud clusters
5. Externalized data stores (databases, object storage, cache) provisioned in GovCloud
6. DNS and certificate management strategy confirmed for cutover
7. Monitoring, logging, and alerting configured and validated on GovCloud prior to cutover

8.3 Migration Phases

Phase	Name	Activities
1	Assessment & Parity Setup	Stand up equivalent Dev/UAT/Staging and Production clusters, VLANs, and namespaces in GovCloud mirroring the interim design. Validate configuration parity.
2	Non-Production Cutover	Redeploy stateless application workloads to GovCloud Dev/UAT/Staging using existing GitLab pipelines (target endpoint change only). Run full regression/UAT sign-off in GovCloud.
3	Data Migration	Migrate externalized data (databases, object storage, cached reference data) from interim data stores to GovCloud equivalents using replication or a scheduled export/import, with checksum/record-count validation.
4	Production Parallel Run	Deploy production workload to GovCloud in parallel with the interim production environment. Synchronize data continuously (e.g., database replication) until cutover.
5	Traffic Cutover	Switch traffic via DNS/load-balancer weighting — either an immediate cutover during an agreed maintenance window or a gradual (canary/blue-green) shift — from interim to GovCloud.
6	Validation & Hypercare	Monitor application health, performance, and error rates intensively for an agreed hypercare period (e.g., 2 weeks) post-cutover.
7	Decommission Interim	Once GovCloud is confirmed stable, decommission the interim clusters and revoke access, after a final data backup/export is confirmed and retained per policy.

8.4 Cutover Approach

Because workloads are stateless, cutover is primarily a routing exercise rather than an application redeployment exercise:

- Blue-Green: GovCloud (green) runs fully in parallel with the interim environment (blue); traffic is switched at the load balancer/DNS layer once validated, with the interim environment kept warm as an immediate rollback target.
- Canary (alternative): a small percentage of traffic is shifted to GovCloud first, increased incrementally as confidence builds, monitored continuously for errors/latency regressions.
- Database cutover is sequenced immediately before or during the traffic switch, using replication to keep GovCloud data current up to the cutover point, minimizing the write-freeze window.

8.5 Rollback Plan

- The interim environment is kept fully operational and untouched until hypercare is complete
- DNS/load balancer routing can be reverted to the interim environment within minutes if critical issues are detected
- Any data written only to GovCloud during the parallel-run window is reconciled back to the interim database as part of rollback procedures, if rollback is invoked

8.6 Indicative Timeline

Milestone	Indicative Timing
GovCloud environment provisioned & validated	TBD — dependent on GDC onboarding schedule
Non-production cutover & UAT sign-off	TBD
Data migration rehearsal	TBD
Production parallel run begins	TBD
Production traffic cutover	TBD
Hypercare period	2 weeks post-cutover (recommended)
Interim environment decommissioned	After hypercare sign-off

Note: *Exact dates depend on the GDC onboarding schedule communicated by GovTech and should be inserted once confirmed.*

9. Roles & Responsibilities

Role	Responsibility	Owner
Infrastructure / Cloud Team	Cluster provisioning, VLAN/network configuration, GovCloud environment setup	TBD
Application Development Team	Ensuring application statelessness, GitLab pipeline maintenance, migration-readiness of code	TBD
Security / Compliance	Network segmentation review, access control audit, data protection sign-off	TBD
Documents Control	Version control, approval workflow, official record-keeping of this document	TBD
GovTech GDC Onboarding Team	GovCloud target-state guidance, onboarding schedule, final acceptance	TBD

10. Risks & Mitigations

Risk	Mitigation	Owner
GDC onboarding schedule slips, extending reliance on interim environment	Design interim environment to production-grade standards so it can be safely operated for an extended period if required	TBD
Data drift between interim and GovCloud during parallel run	Use continuous replication and pre-cutover reconciliation checks	TBD
Hidden state in application breaks portability	Mandatory statelessness review/code audit before migration cutover	TBD
Network policy mismatch between interim VLANs and GovCloud network zones	Joint review with network/security teams before non-production cutover	TBD
Rollback window insufficient if issues found late	Keep interim environment live through the full hypercare period before decommission	TBD

11. Assumptions & Dependencies

- GovCloud target infrastructure will support equivalent (or superior) cluster and VLAN segmentation described in this document.
- The official GovTech GitLab instance remains the source-code platform post-migration, with CI/CD runners able to reach GovCloud deployment targets.
- Network connectivity/firewall approvals between GitLab, the container registry, and GovCloud clusters will be granted in time for pipeline cutover.
- Externalized data stores selected for the interim solution have GovCloud-compatible equivalents (or a supported migration path) available at cutover.

12. Appendix

A. Glossary

See Section 1.4 for core definitions. Additional terms to be appended as required.

B. References

- GovTech GitLab platform documentation (internal)
- GDC/GovCloud onboarding guidelines (to be provided by GovTech)
- Organizational network and security policy documents (internal)

C. Open Items for Document Control

- Assign official document reference number and classification
- Confirm named approvers and route for sign-off
- Attach finalized VLAN/subnet details once issued by the network team
- Attach finalized GDC onboarding schedule once received from GovTech