Cloud Audit: Resource Inventory

Drop — AWS Resource Inventory

Date: 2026-02-19 **Region:** eu-west-1 (Ireland) **Account:** Drop production **Auditor:** infra-lead (CloudForge cloud-audit team) **MC Task:** #1443

Executive Summary

Drop runs a minimal AWS footprint: one App Runner service fronting a PostgreSQL RDS instance, with container images stored in ECR. Total estimated cost is $48-60/month.

Three CRITICAL security findings require immediate action:

1. RDS database is publicly accessible with security group open to the entire internet (0.0.0.0/0 on port 5432)
2. Database storage is unencrypted
3. Plaintext secrets (DATABASE_URL with password, JWT_SECRET) in App Runner environment variables

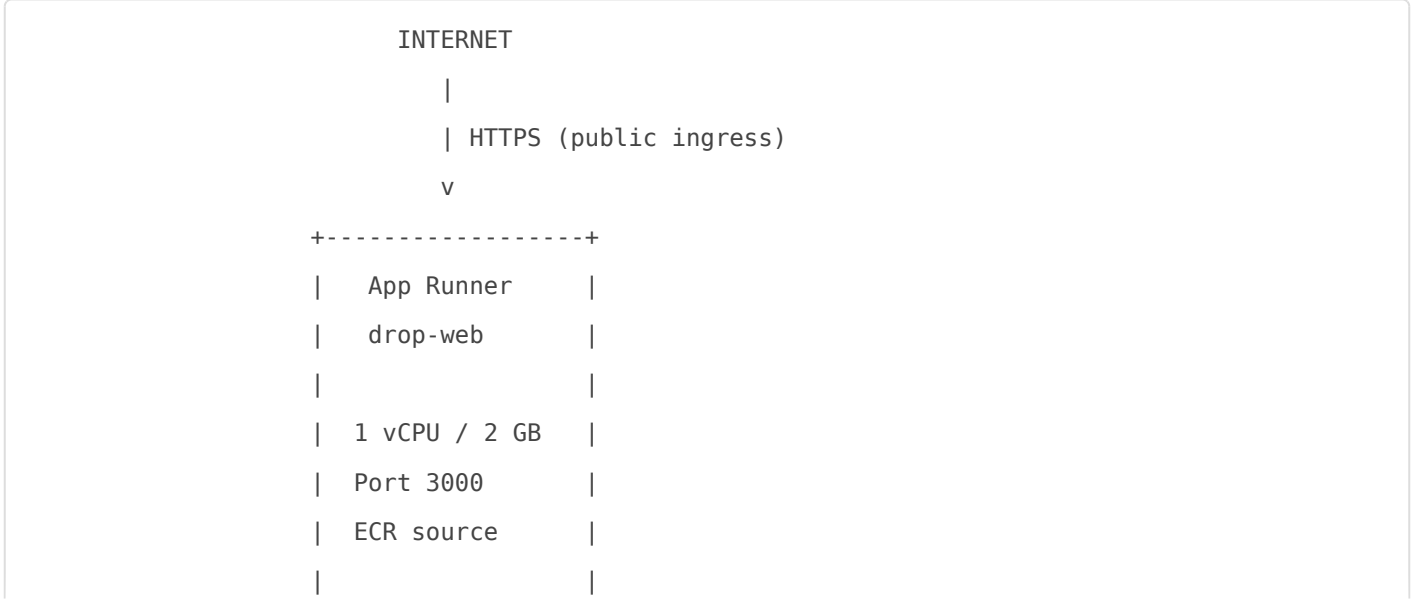
No WAF, no CloudFront, no CloudWatch monitoring, no Route53 DNS management, and Secrets Manager is provisioned but empty.

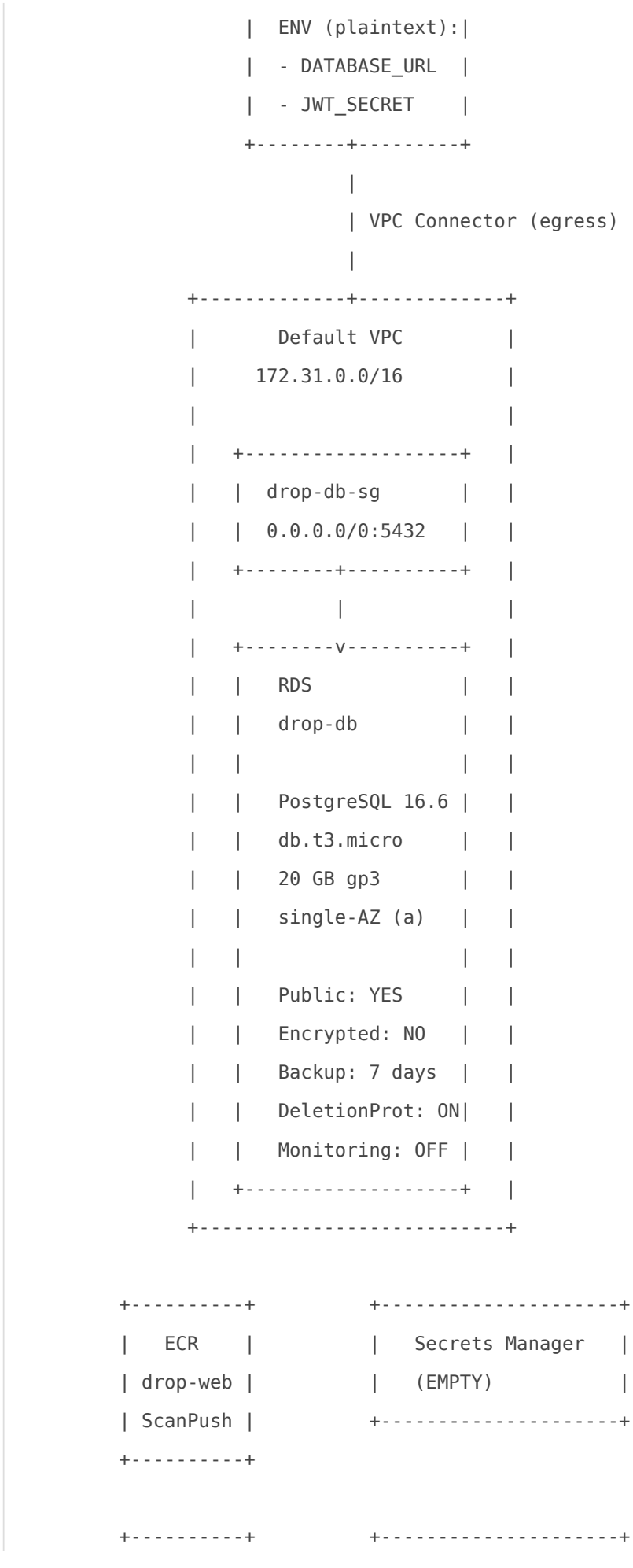
Resource Table

Resource	Type	ID / Name	Region	Status	Key Config
App Runner	Service	drop-web	eu-west-1	RUNNING	1 vCPU, 2 GB RAM, port 3000
RDS	PostgreSQL 16.6	drop-db	eu-west-1a	Available	db.t3.micro, 20 GB gp3, single-AZ

Resource	Type	ID / Name	Region	Status	Key Config
ECR	Repository	drop-web	eu-west-1	Active	ScanOnPush: TRUE, Encryption: AES256
Security Group	SG	drop-db-sg	eu-west-1	In use	Inbound: 0.0.0.0/0 : 5432
VPC	Default	—	eu-west-1	Active	172.31.0.0/16
IAM User	User	john-deploy	Global	Active	Programmatic access
IAM Role	Role	AppRunnerECRAccessRole	Global	Active	ECR pull permissions
Secrets Manager	—	(empty)	eu-west-1	Provisioned	0 secrets stored
CloudWatch	—	—	—	NOT CONFIGURED	No alarms, no dashboards
CloudFront	—	—	—	NOT PROVISIONED	No CDN
WAF	—	—	—	NOT PROVISIONED	No web application firewall
Route53	—	—	—	NOT PROVISIONED	DNS managed externally
S3	—	—	—	NOT PROVISIONED	No buckets

Architecture Diagram




| ENV (plaintext):|
| - DATABASE_URL |
| - JWT_SECRET |
VPC Connector (egress)
Default VPC
172.31.0.0/16
drop-db-sg
0.0.0.0/0:5432
RDS
drop-db
PostgreSQL 16.6
db.t3.micro
20 GB gp3
single-AZ (a)
Public: YES
Encrypted: NO
Backup: 7 days
DeletionProt: ON
Monitoring: OFF
ECR
drop-web
ScanPush
Secrets Manager
(EMPTY)

```
|   IAM    |          |   MISSING           |
| john-    |          |   CloudWatch        |
|  deploy  |          |   CloudFront        |
| ECR Role |          |   WAF / Route53 / S3|
+----------+          +---------------------+
```

Security Findings

CRITICAL

#	Finding	Resource	Risk	Remediation
C1	**Database publicly accessible**	RDS drop-db	Direct internet access to PostgreSQL. Any attacker can attempt connections.	Set `PubliclyAccessible=false`. App Runner already uses VPC Connector for egress — RDS only needs private subnet access.
C2	**Security group allows 0.0.0.0/0 on port 5432**	drop-db-sg	Combined with C1, the database is wide open to brute-force and exploitation from any IP on Earth.	Restrict inbound rule to App Runner VPC Connector security group only. Remove 0.0.0.0/0 CIDR.
C3	**Plaintext secrets in App Runner env vars**	App Runner drop-web	DATABASE_URL contains full connection string with password. JWT_SECRET in plaintext. Anyone with console/API access sees credentials. Visible in CloudTrail, config exports, and deployment logs.	Migrate secrets to AWS Secrets Manager (already provisioned, currently empty). Reference via App Runner secret ARN configuration. Rotate both DATABASE_URL password and JWT_SECRET after migration.

#	Finding	Resource	Risk	Remediation
C4	**Database storage unencrypted**	RDS drop-db	Data at rest is not encrypted. Violates baseline security posture and most compliance frameworks (SOC2, GDPR, PCI).	Enable storage encryption. Note: cannot enable on existing instance — requires snapshot, restore to encrypted instance, DNS/connection swap. Plan downtime window.

HIGH

#	Finding	Resource	Risk	Remediation
H1	**Single-AZ deployment**	RDS drop-db	AZ failure = full database outage. No automatic failover.	Enable Multi-AZ for production. Cost increase ~$14/mo for db.t3.micro.
H2	**No monitoring or alerting**	CloudWatch (missing)	No CPU, memory, connection, or storage alarms. No visibility into failures, performance degradation, or security events. Silent failures.	Configure CloudWatch alarms: CPU > 80%, FreeStorageSpace < 2 GB, DatabaseConnections > 80%, FreeableMemory < 200 MB. Enable Enhanced Monitoring on RDS.
H3	**No WAF**	WAF (missing)	No protection against OWASP Top 10 attacks (SQLi, XSS, SSRF, etc.) at the edge. App Runner public endpoint is directly exposed.	Deploy AWS WAF with managed rule groups (AWSManagedRulesCommonRuleSet, AWSManagedRulesSQLiRuleSet). Attach to CloudFront distribution (see H4).

MEDIUM

#	Finding	Resource	Risk	Remediation

M1	**No CDN / CloudFront**	CloudFront (missing)	All traffic hits App Runner origin directly. No edge caching, no DDoS protection (Shield Standard), higher latency for distant users.	Deploy CloudFront distribution in front of App Runner. Enables WAF attachment, caching, and Shield Standard.
M2	**Default VPC**	VPC 172.31.0.0/16	Default VPC has broad routing, public subnets by default, and no network segmentation. Not suitable for production workloads.	Create custom VPC with private subnets for RDS, public subnets for NAT Gateway / ALB if needed. Migrate RDS to private subnet.
M3	**No DNS management**	Route53 (missing)	DNS managed outside AWS. No health checks, no failover routing, no alias records for AWS resources.	Consider Route53 for DNS if domain is Drop-owned. Enables health-check-based routing and simpler AWS integration.
M4	**TCP health check only**	App Runner drop-web	TCP checks confirm port is open but not that the application is healthy. A process could accept connections while returning 500s.	Configure HTTP health check on a dedicated `/health` endpoint that verifies database connectivity.

LOW

#	Finding	Resource	Risk	Remediation
L1	**No S3 buckets**	S3 (missing)	If the app needs file storage in future, ensure encryption-at-rest (SSE-S3 or SSE-KMS), versioning, and public access block from day one.	Provision with secure defaults when needed.
L2	**IAM user john-deploy**	IAM	Long-lived access keys. No indication of key rotation policy or MFA.	Audit key age. Enable MFA. Consider OIDC federation for CI/CD instead of IAM user. Rotate keys on a 90-day schedule.

Cost Breakdown

Service	Specification	Estimated Monthly Cost
App Runner	1 vCPU, 2 GB, always running	$29 - $36
RDS	db.t3.micro, 20 GB gp3, single-AZ	$15 - $18
ECR	Image storage (~1-5 GB)	$0.50 - $1.00
Data Transfer	Minimal (< 10 GB/mo estimate)	$1 - $2
Secrets Manager	0 secrets (currently unused)	$0
Total		**$46 - $57/mo**

Cost Notes

- App Runner pricing: $0.064/vCPU-hour + $0.007/GB-hour (provisioned mode)
- RDS db.t3.micro: ~$0.018/hour ($13.14/mo) + $0.115/GB-month storage
- No NAT Gateway cost (App Runner VPC Connector handles egress)
- Adding Multi-AZ RDS: +$13-15/mo
- Adding CloudFront: +$0-5/mo (depends on traffic)
- Adding WAF: +$5-10/mo (depends on rules and requests)

Gaps Analysis

Category	Current State	Target State	Priority
Secrets management	Plaintext env vars	Secrets Manager with rotation	CRITICAL
Network security	Public RDS + open SG	Private subnet + restricted SG	CRITICAL
Encryption at rest	Disabled	AES-256 (KMS or default)	CRITICAL
Monitoring	None	CloudWatch alarms + dashboards	HIGH
High availability	Single-AZ	Multi-AZ RDS	HIGH
Edge security	No WAF / CDN	CloudFront + WAF	HIGH
Network architecture	Default VPC	Custom VPC with segmentation	MEDIUM
Health checks	TCP only	HTTP application-level	MEDIUM
IAM hygiene	Long-lived keys	OIDC + key rotation + MFA	MEDIUM

Category	Current State	Target State	Priority
DNS	External	Route53 (optional)	LOW
Backup/DR	7-day automated only	Cross-region snapshot copy	LOW

Recommendations (Priority Order)

Phase 1 — Immediate (Week 1) — CRITICAL Security

1. **Lock down RDS network access**
   - Set `PubliclyAccessible=false` on drop-db
   - Update drop-db-sg: remove 0.0.0.0/0, allow only App Runner VPC Connector SG
   - Verify App Runner can still connect via VPC Connector
2. **Migrate secrets to Secrets Manager**
   - Create secrets: `drop/database-url`, `drop/jwt-secret`
   - Update App Runner service to reference secret ARNs
   - Remove plaintext env vars from App Runner config
   - Rotate database password and JWT secret post-migration
3. **Enable RDS encryption**
   - Snapshot current instance
   - Restore snapshot with encryption enabled
   - Update connection string to new endpoint
   - Verify, then delete old unencrypted instance
   - Requires brief downtime — schedule maintenance window

Phase 2 — Short Term (Week 2-3) — HIGH Priority

4. **Configure CloudWatch monitoring**
   - RDS alarms: CPU, storage, connections, memory
   - App Runner alarms: request count, error rate, latency
   - SNS topic for alert notifications
   - Enable RDS Enhanced Monitoring
5. **Enable Multi-AZ RDS**
   - Modify instance to Multi-AZ
   - Near-zero downtime (AWS handles failover setup)
6. **Deploy CloudFront + WAF**
   - CloudFront distribution pointing to App Runner
   - WAF with AWS managed rule sets (Common, SQLi, Known Bad Inputs)

- Update DNS to point to CloudFront

Phase 3 — Medium Term (Month 2) — Hardening

7. **Custom VPC migration**
   - Design VPC: 2 private subnets (RDS), 2 public subnets (NAT if needed)
   - Migrate RDS to private subnets
   - Update App Runner VPC Connector
8. **HTTP health checks**
   - Implement `/health` endpoint in Drop application (DB connectivity check)
   - Configure App Runner HTTP health check path
9. **IAM improvements**
   - Audit john-deploy key age
   - Enable MFA on IAM user
   - Consider GitHub Actions OIDC for CI/CD (eliminates long-lived keys)

Risk Matrix

Risk	Likelihood	Impact	Severity	Mitigation
Database breach via public access + open SG	HIGH	CRITICAL	**CRITICAL**	Phase 1: Lock down network (C1, C2)
Credential leak from plaintext env vars	MEDIUM	CRITICAL	**CRITICAL**	Phase 1: Secrets Manager (C3)
Data exposure from unencrypted storage	LOW	HIGH	**HIGH**	Phase 1: Enable encryption (C4)
Database outage (single-AZ failure)	LOW	HIGH	**HIGH**	Phase 2: Multi-AZ (H1)
Silent application failure (no monitoring)	MEDIUM	MEDIUM	**HIGH**	Phase 2: CloudWatch (H2)
Application-layer attack (no WAF)	MEDIUM	HIGH	**HIGH**	Phase 2: WAF (H3)
DDoS / performance degradation (no CDN)	LOW	MEDIUM	**MEDIUM**	Phase 2: CloudFront (M1)
Lateral movement via default VPC	LOW	MEDIUM	**MEDIUM**	Phase 3: Custom VPC (M2)
IAM key compromise	LOW	HIGH	**MEDIUM**	Phase 3: Key rotation + OIDC (L2)

Appendix: Raw Resource Details

App Runner — drop-web

```
Service:         drop-web
Status:          RUNNING
Region:          eu-west-1
Source:          ECR (container image)
CPU:             1 vCPU
Memory:          2 GB
Port:            3000
Ingress:         Public
Egress:          VPC Connector
Health Check:    TCP
Environment:     DATABASE_URL (plaintext, contains password)
                 JWT_SECRET (plaintext)
```

RDS — drop-db

```
Engine:          PostgreSQL 16.6
Instance Class:  db.t3.micro
Storage:         20 GB gp3
AZ:              eu-west-1a (single-AZ)
VPC:             Default (172.31.0.0/16)
Public Access:   TRUE
Encrypted:       FALSE
Deletion Prot:   TRUE
Backup:          7-day automated
Monitoring:      DISABLED
```

ECR — drop-web

```
Repository:      drop-web
Scan on Push:    TRUE
Encryption:      AES256 (default)
```

Security Groups — drop-db-sg

```
Inbound Rules:
  - Protocol: TCP
    Port: 5432
    Source: 0.0.0.0/0  (ALL TRAFFIC)
```

IAM

```
User:   john-deploy     (programmatic access, deployment)
Role:   AppRunnerECRAccessRole  (App Runner → ECR pull)
```

Secrets Manager

```
Secrets stored: 0 (service provisioned but unused)
```

Revision #5
Created 2026-02-23 11:28:57 UTC by John
Updated 2026-05-23 10:58:59 UTC by John