2026

aws

Rossco's Coffee Cloud Security Plan

PORTFOLIO CASE STUDY

A Latte Data, No Downtime

Markus Walker
AUTHOR

ROSSCO'S COFFEE SHOP | Business address removed

TABLE OF CONTENTS
CONTENTS

Portfolio note: This document has been adapted from a previous capstone project for portfolio presentation. The technical content has been kept substantially word-for-word, while training-specific identifiers, author details, and similar presentation artifacts have been removed or neutralised where practical.

INTRODUCTION

Rossco's Coffee has grown quickly and the shop now takes many online orders every day. The current website runs on a single AWS EC2 instance with a local MariaDB database. As traffic grows the server is struggling to keep up, and the business worries about security and reliability. This Cloud Security Plan sets out how we will design, build and operate a more resilient cloud enviroment. It describes our assumptions and constraints, outlines the project scope and summarises how the new architecture works.

PURPOSE

The purpose of this plan is to give Rossco's Coffee a clear path to migrate its web application to a secure and scalable cloud platform. It aims to provide a consistent, reliable ordering experience for five cafes and more than 800 regular customers. The plan guides the project team on how to design and implement the upgraded architecture, how to test and cut over to the new enviroment and how to operate it safely over time with thought to cloud incidents and our responses. It also documents the business assumptions, budget limits and responsibilities between Rossco's Coffee and AWS. By following this plan, the team will reduce downtime, protect customer data and avoid over-engineering so they stay within the AU $200 000 budget.

ASSUMPTIONS AND CONSTRAINTS

SCHEDULE

The migration is expected to finish within six months. New features will be frozen in the last month except critical fixes. Cut-over to the new system should happen out of hours and take less than two hours, with a tested rollback procedure.

BUDGET

Total spend for the first year, including AWS usage, backups, monitoring, training and security services, must stay under AU $200 000. Wherever possible we will use managed services to reduce operational overheads. Multi-region active-active is out of scope.

RESOURCE AVAILABILITY AND SKILL SETS

One cloud architect will lead the change. Cafe managers will have limited console access for content updates and the webapp has front-end admin features. There is no dedicated ops team. We will rely on AWS managed services and external testers for penetration testing. Support will be business hours with on-call only during cut-over or major incidents.

PLATFORM, SOFTWARE AND TECHNOLOGY

We will stay on AWS and keep the PHP application running on Apache on Amazon Linux. The database will migrate to Amazon RDS MariaDB in a Multi-AZ configuration *(Amazon Web Services, n.d.-b*). CloudFormation will manage infrastructure with templates stored in S3. Session Manager *(Amazon Web Services, n.d.-e)* will replace SSH for administration. The focus remains on an EC2-based stack, but we may refactor the app slightly to improve authentication, input validation and secret handling.

INTEROPERABILITY

The current PHP code and MariaDB schema will be ported to the new platform with minimal changes. The application will pull secrets from AWS Secrets Manager, connect to RDS over TLS and use Cognito JWTs for sign-in. User and order data will migrate to the new database during cut-over.

ADDITIONAL INFORMATION

Data and logs stay in Australian Regions. Least privilege IAM with MFA is enforced everywhere. All data is encrypted in transit and at rest. Separate dev, staging and production environments will exist so testing is safe. The design aims to be simple enough for a small team to manage without sacrificing security.

PROJECT OVERVIEW

This plan delivers a secure, scalable two-tier web application on AWS. It uses managed services where they add value so the team can focus on running the cafe rather than infrastructure. A Virtual Private Cloud (VPC) spans two Availability Zones. Public subnets host an Application Load Balancer (ALB) and NAT Gateways. Private subnets host stateless EC2 instances running the PHP application and a managed RDS MariaDB database. The ALB terminates TLS using certificates from AWS Certificate Manager and forwards HTTPS traffic to the EC2 instances. Auto Scaling Groups *(Amazon Web Services, n.d.-a)*. adjust capacity based on demand. Each instance is built from a Golden Amazon Machine Image (AMI) and managed with AWS Systems Manager. There is no inbound SSH. The database uses RDS Multi-AZ with automated backups, encryption and point-in-time recovery. NAT Gateways allow outbound internet access from private subnets. An S3 Gateway Endpoint enables private access to object storage, reducing NAT costs. IAM Identity Center provides centralised identity with multi-factor authentication, and customers sign up and sign in through Amazon Cognito. Security services like WAF, Shield Standard, CloudTrail, GuardDuty, Security Hub, Inspector and Macie provide layered protection. Encryption at rest uses KMS keys, and TLS 1.2+ is enforced for data in transit. The environment is built and maintained with CloudFormation templates stored in versioned S3 buckets.

EVALUATION

CURRENT CLOUD ARCHITECTURE AND MODEL

Rossco's Coffee currently runs its web application on a single AWS EC2 instance within one Availability Zone. The instance resides in a public subnet and hosts both the web server and a local MariaDB database. An Application Load Balancer exists but is not actively routing traffic, so all customer requests go directly to the EC2. The deployment model is public IaaS because the company manages the operating system and software on the EC2 instance while AWS manages the underlying infrastructure. There is no evidence of other service models in use.

CURRENT CLOUD SERVICES

The current environment uses a Virtual Private Cloud (one VPC) with a public and a private subnet. However only the public subnet is used. The public subnet contains the EC2 instance, the ALB and an Internet Gateway. A security group allows inbound HTTP on port 80 and SSH on port 22 from any source. There is no NAT Gateway. CloudFormation provisions the stack but there is no version control or drift detection. No managed database or storage services are used, application files and database data are stored on the instance.

CURRENT ACCESS CONTROL

There is one IAM user with administrative privileges that manages the AWS account. Access to the EC2 instance uses a static SSH key pair. There is no multi-factor authentication. The application stores database credentials in user data and the AMI. Customer authentication is built into the PHP code and uses plaintext credentials stored in the database. There are no IAM roles or identity federation.

CURRENT SECURITY CONTROLS

Security relies on a security group that allows inbound HTTP and SSH from anywhere. The declared Network Firewall has no rules or routing. There is no Web Application Firewall, Data Loss Prevention or DDoS mitigation. Encryption is not enforced, the website uses HTTP only and the local database is unencrypted. Logging and monitoring consist only of default system logs stored on the instance. There is no CloudTrail, GuardDuty or Config. Backups are not configured so there is a single copy of data. Disaster recovery is not planned.

CURRENT PROTOCOLS

The website serves customers over HTTP on port 80. SSH on port 22 is open to the world. MariaDB listens on port 3306 but only locally. There is no TLS for web traffic or database connections. There is no VPN or private endpoint for administrative access. Logging uses local syslog only.

VULNERABILITY AND RISK ASSESSMENT

The current setup presents multiple risks. A single EC2 instance is a single point of failure. It hosts both application and database, so compromise of the instance exposes all data. Inbound SSH allows passwordless access if the key is compromised. HTTP means credentials and session tokens travel in plaintext. The ALB and Network Firewall exist but are not used. There are no backups or disaster recovery, so hardware failure or human error could cause permanent data loss. Credentials are

hardcoded in user data and AMIs. Without logging and monitoring, security incidents may go unnoticed. Overall the system meets basic function but fails on availability and security.

EVALUATION SUMMARY

The existing environment is a simple single instance running both the application and the database. It qualifies as a public cloud IaaS deployment but lacks any high availability, scalability or security features. The business is exposed to outages, data loss and breaches. These findings drive the proposed migration to a two-tier, multi-AZ design with managed database, proper identity management, encryption, logging and automated backups.

CLOUD ENVIRONMENT UPGRADE PLAN

CLOUD MODELS, FEATURES AND SECURITY RESPONSIBILITIES

CLOUD MODELS

Aspect	**Description**	**Advantages**	**Disadvantages**
Deployment	Public cloud on AWS, single Region with two AZs	Simple and cost effective, resilient to one AZ outage	No automatic multi Region failover

SERVICE MODELS

Model	**Examples**	**Features**	**Advantages**	**Disadvantages**
IaaS	EC2, VPC, security groups	Control of OS, network and instance sizing	Flexible and familiar operations	We must patch and harden instances, more day to day care
PaaS	Amazon RDS MariaDB Multi-AZ	Managed patching, automated backups, PITR, synchronous failover	Lower operational load and reliable recovery	Less low level tuning and fixed maintenance windows
SaaS-style	Amazon Cognito, Security Hub	Customer identity and centralised security findings	Fast to adopt with minimal maintenance	Feature changes depend on the vendor roadmap, limited customisation

SECURITY AND SOVEREIGNTY

Category	**Responsibilities and settings**
Shared responsibility	AWS secures facilities, hardware, core networking, hypervisors and managed layers. Rossco's Coffee configures the account, identities, encryption, logging, backups, monitoring, incident response and compliance
Sovereignty	All data remains in ap-southeast-2. KMS CMKs with tight key policies. Cross-Region replication deferred to control cost. HA is covered by Multi-AZ. DR is backup and restore

CLOUD SERVICES AND VIRTUAL APPLICATIONS

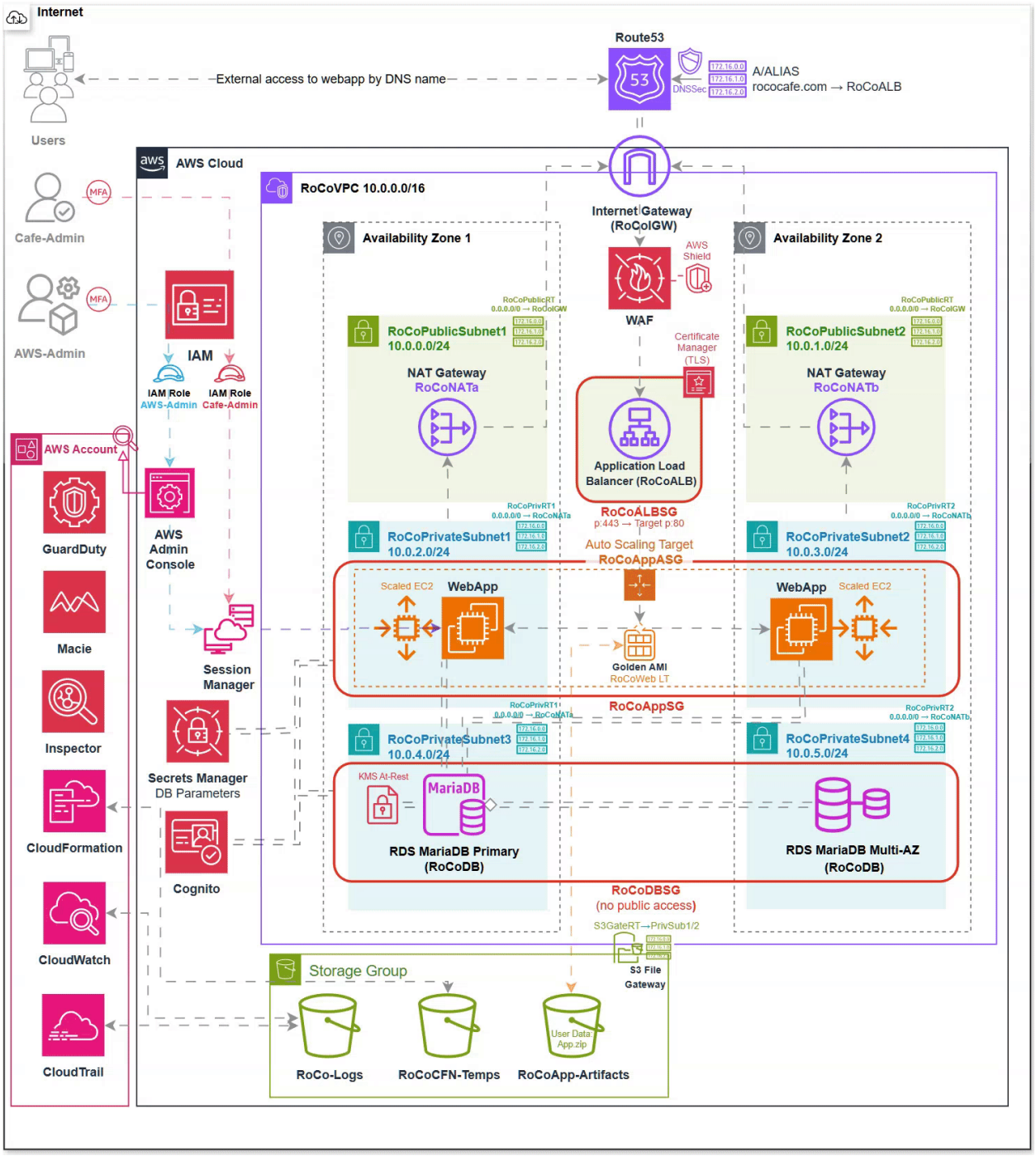


Figure 1. Proposed AWS architecture for Rossco's Coffee Shop. A two tier highly available design with WAF, ALB to stateless EC2 and RDS Multi-AZ in private subnets.

We deploy a two tiered architecture, a web tier and a data tier. The web tier consists of stateless EC2 instances in private subnets across two Availability Zones. An Application Load Balancer in public subnets terminates TLS and distributes traffic. The ALB integrates with AWS WAF (Amazon Web

Services, n.d.-d) to block common attacks and with Shield Standard *Shield* for basic DDoS protection. Auto Scaling maintains at least two instances and scales out based on CPU or request count to handle peaks. Instances are built from Golden AMIs using EC2 Image Builder and run Amazon Linux with Apache and PHP. On boot they pull code artefacts from S3, fetch secrets from Secrets Manager and start services. Session Manager enables admin access without opening SSH. The data tier is an Amazon RDS MariaDB Multi-AZ instance. It provides automated backups, replication and encryption. A read replica can be added later if read load grows. Database credentials are stored in Secrets Manager and rotated.

HIGH DEMAND SOLUTION

Auto Scaling covers demand spikes by adding EC2 instances automatically. We do not need cloud bursting to other providers because scaling within AWS can meet load. Target tracking policies adjust capacity based on CPU or requests per second. Minimum capacity is two instances. The ALB health checks ensure only healthy instances receive traffic. We do not use container orchestration or serverless because the current PHP application is not containerised and high level redesign of the base code is out of scope for our purposes.

INFRASTRUCTURE FAILS SOLUTION

High availability is achieved by deploying resources across two Availability Zones. If one zone fails, the other continues to serve traffic. The ALB and Auto Scaling group span both zones. RDS Multi-AZ replicates data synchronously and fails over automatically. EBS snapshots and automated backups allow restoring the database if there is corruption. For disaster recovery beyond the Region, snapshots can be copied to another Region but this is optional due to cost. We will test restoration quarterly. Route 53 health checks and failover policies can be configured in future if multi-Region becomes necessary.

ACCESS CONTROL

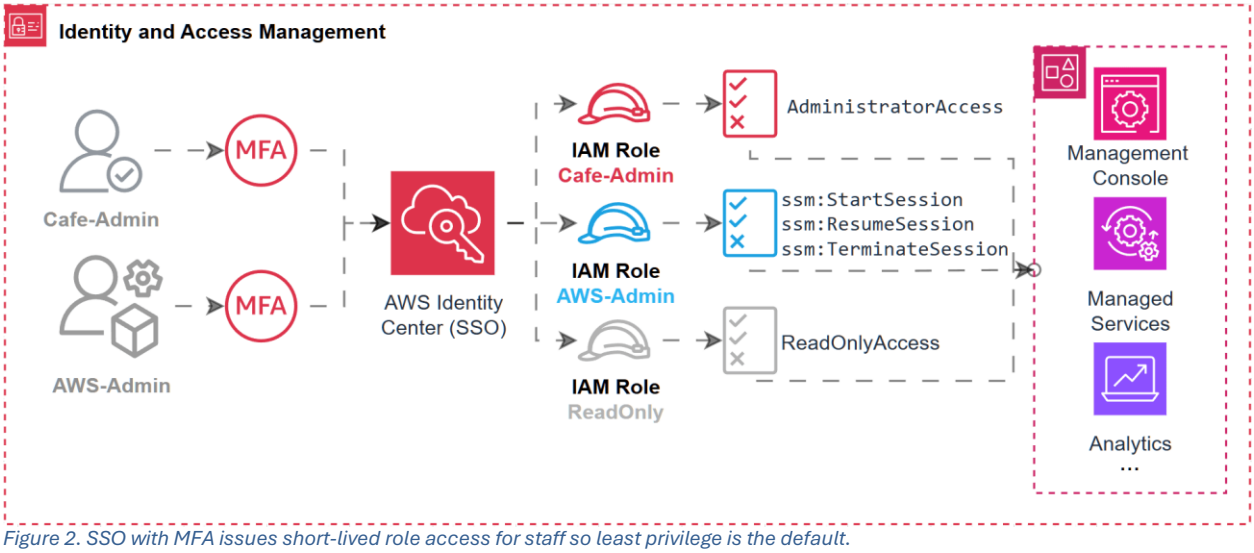


Figure 2. SSO with MFA issues short-lived role access for staff so least privilege is the default.

Component	***Purpose***	***Reason***	***Limitation***	***Alternative***
Identity Center SSO	Central sign in for staff with MFA	Strong auth and easy access control	Needs internet sign in	IAM users with MFA, or connect to an external IdP later
AWS-Admin role	Full admin for rare tasks	Keep day to day admins separate from break glass	High power if misused	Use ‘just in time’ elevation and session duration limits
Cafe-Admin role	Operate the workload	Scoped rights for daily ops	Can grow if not reviewed	Quarterly access reviews and tighter permission sets
ReadOnly role	View only	Safe access for audits and training	No change actions	Temporary elevate to Operator when needed
Session Manager	Shell and console without SSH	No inbound ports and full audit trail	Requires SSM agent and IAM	Temporary bastion only during outages
Break glass user	Emergency access with hardware MFA	Works if SSO is down	Risk if storage is sloppy	Second device sealed and sign-out log after use

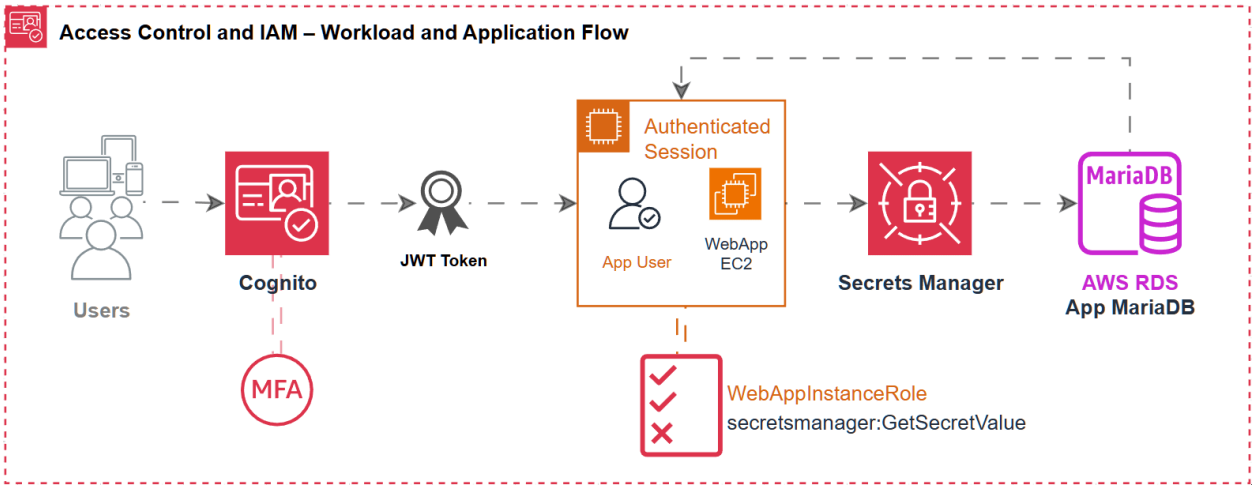


Figure 3. Customers authenticate with Cognito, the app validates a JWT, and EC2 uses an instance role to fetch secrets before connecting to RDS.

Component	**Purpose**	**Reason**	**Limitation**	**Alternative**
Break glass IAM user	*Last resort entry with hardware MFA*	*Works when SSO or SSM fail*	*Token can be lost if not stored right*	*Keep a second device sealed and logged*
CloudTrail	*Record every action*	*Complete audit trail to S3 with KMS*	*Storage grows overtime*	*Lifecycle to Glacier and quarterly review*
CloudWatch alarm	*Detect risky events*	*Fast signal to on-call*	*Noise if rules are loose*	*Tune rules and add metric filters*
EventBridge rule	*Route the event*	*Simple fanout to people and automation*	*Needs rule upkeep*	*Use a small incident function/pattern*
SNS	*Notify responders*	*Email or SMS to the team*	*Lists can go stale*	*Review recipients monthly*

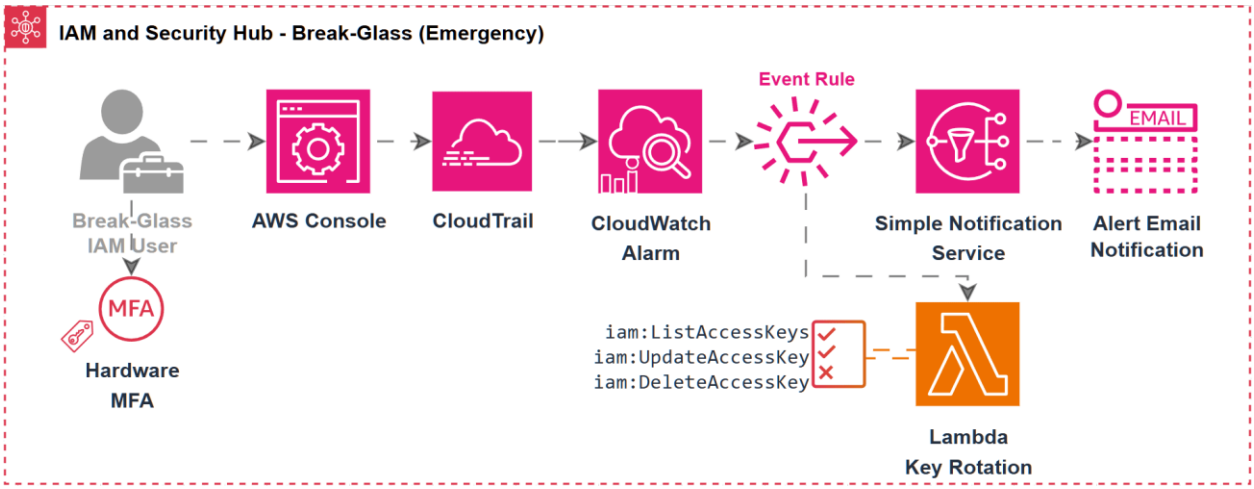


Figure 4. One hardware-MFA break-glass user exists and any use is logged, alerted and can trigger automated key rotation.

Component	**Purpose**	**Reason**	**Limitation**	**Alternative**
Cognito User Pools	Customer sign up and sign in	Managed identity without extra servers	Complex custom flows are limited	Third-party IdP or custom auth if required
JWT validation	Prove user session to the app	Stateless session and easy scale	Revocation is not instant	Short token life with refresh flow
WebAppInstanceRole	Give EC2 temprorary creds	No long lived keys on hosts	First secret fetch adds tiny latency	Cache secret in memory with short TTL
Secrets Manager	Store and rotate DB creds	Central control and rotation without code change	Rotation needs planning windows	Parameter Store for low impact values
RDS security group	Allow only web tier to DB port	Reduces blast radius	Blocks direct laptop access	Use SSM port forward for DB admin work

In brief, access for staff is managed through IAM Identity Center. Users authenticate via SSO with MFA. Roles define privileges: Admin, Operator and ReadOnly. A single break-glass user has hardware MFA for emergencies. No long-lived access keys are used. Session Manager provides shell access and logs sessions centrally. Customer authentication uses Cognito User Pools, the application validates JWTs and never stores passwords. Least privilege is enforced by attaching fine-grained IAM policies to roles and resources. Permissions are reviewed quarterly and removed when not needed.

CLOUD SECURITY CONTROL

We layer controls to protect data and systems. Web Application Firewall rules block SQL injection and cross-site scripting. Shield Standard defends against DDoS. WAF and Shield logs are sent to CloudWatch for alerting. Security groups restrict inbound traffic to the ALB (HTTPS only) and restrict east-west traffic between subnets. Network ACLs provide stateless filtering. RDS security groups allow only the web tier to connect. GuardDuty *(Amazon Web Services, n.d.-g)* analyses logs to detect threats. Security Hub *(Amazon Web Services, n.d.-h)* aggregates findings and checks resources against AWS best practices. Inspector *(Amazon Web Services, n.d.-i)* scans EC2 instances for vulnerabilities and Macie *(Amazon Web Services, n.d.-j)* discovers sensitive data in S3. CloudTrail *(Amazon Web Services, n.d.-f)* records API activity for auditing. AWS Config tracks configuration changes and compliance. KMS encrypts EBS volumes, RDS data and S3 objects. TLS 1.2+ is enforced on the ALB and to the database. Route 53 DNSSEC is enabled to prevent DNS spoofing.

CLOUD ACCESS SECURITY BROKER (CASB)

Solution. Use Security Hub as the central view with CloudTrail Lake for investigation queries. Ingest findings from GuardDuty, Inspector, Macie, WAF and Config.

Reason. One centre the small team can actually use.

Limitation. Needs some tuning to keep noise low.

Alternative. Forward findings to a managed SIEM if scale grows.

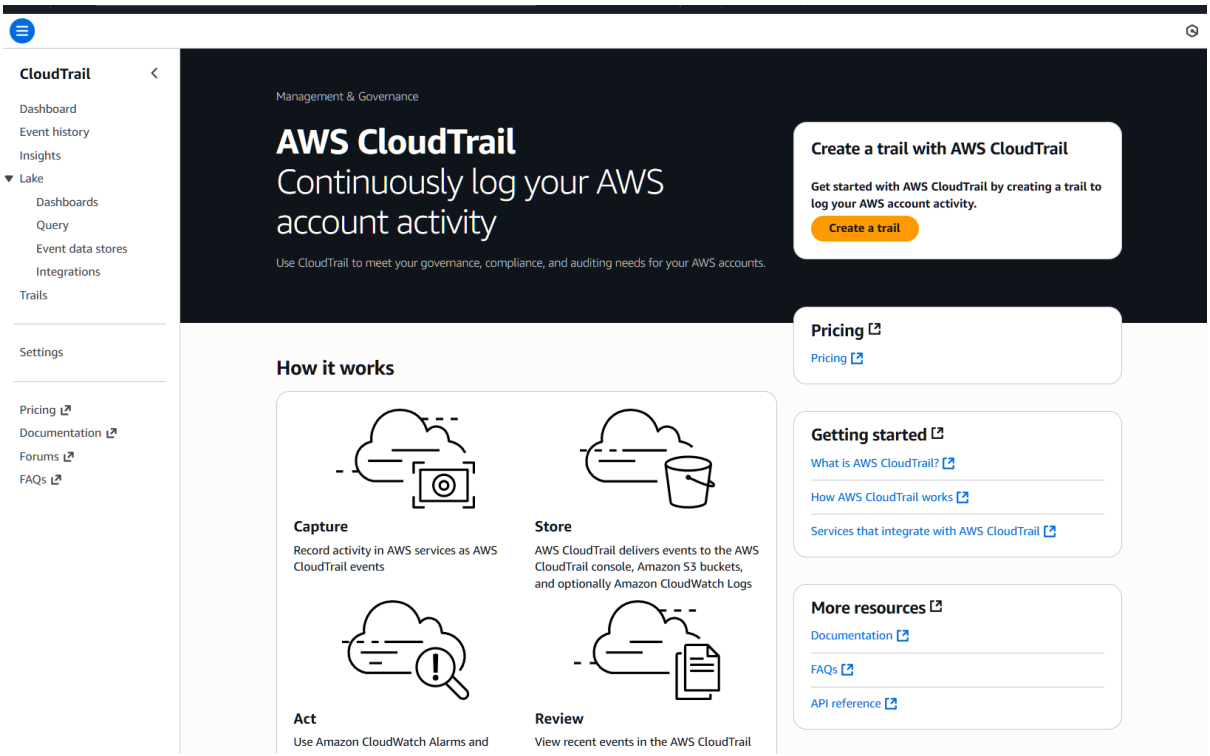


Figure 5. AWS CloudTrail service configured to capture and store all account activity, ensuring governance, compliance, and full auditability of administrative and security events across Rossco's Coffee cloud infrastructure.

FIREWALL WEB APPLICATION FIREWALL (WAF)

Solution. Attach AWS WAF to the ALB with AWS managed rule groups and IP reputation list. Enable logs to S3 and beyond.

Reason. Blocks common exploits at the edge with minimal upkeep.

Limitation. False positives can occur if rules are not reviewed.

Alternative. Add rate based rules or Bot Control if abuse rises.

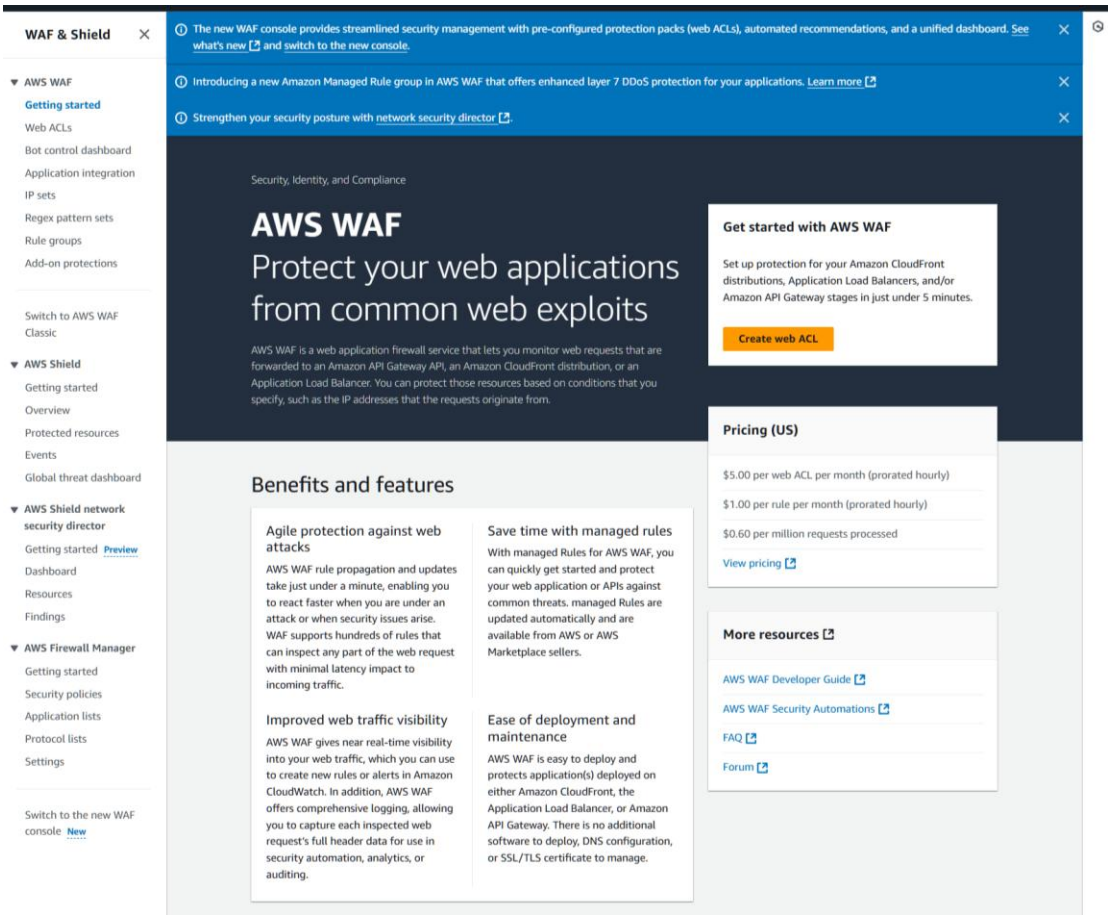


Figure 6. AWS WAF service dashboard showing protection options for web applications against exploits such as SQL injection and cross-site scripting, demonstrating best practice by integrating managed rulesets in front of the load balancer for Rossco's Coffee Shop

APPLICATION DELIVERY CONTROLLER (ADC)

Solution. Application Load Balancer for TLS termination, health checks and path based routing.

Reason. Layer 7 routing that fits a PHP web app.

Limitation. Not ideal for raw TCP only traffic.

Alternative. Use NLB for TCP heavy needs or API Gateway for REST APIs later.

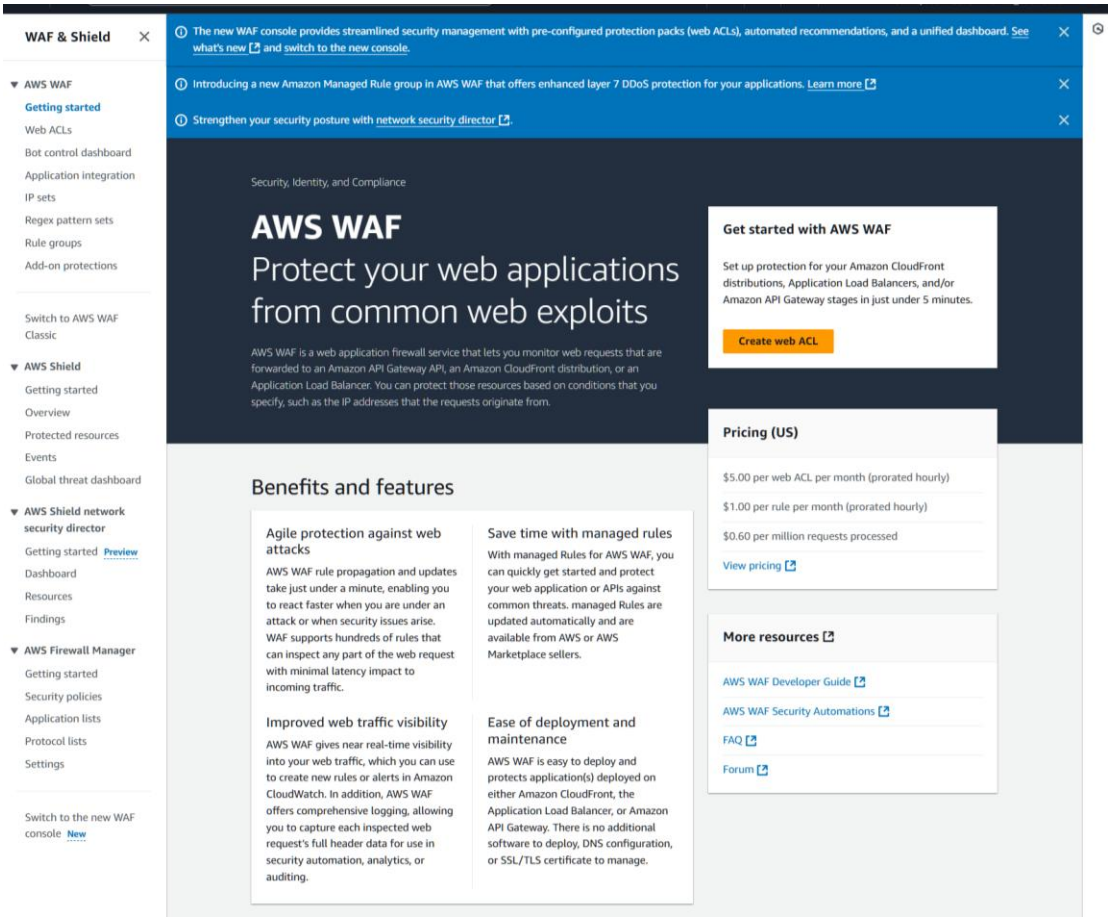


Figure 7. Application Load Balancer (RoCoALB) configured to forward web traffic on port 80 to the target group RoCoTG, demonstrating secure traffic management and fault-tolerant delivery for the public website http://rosscoscoffee.com.au/

DATA LOSS PREVENTION (DLP)

Solution. Amazon Macie scans S3 buckets that hold logs and artefacts. Findings flow into Security Hub.

Reason. Catch accidental PII or credentials before they spread.

Limitation. Can be noisy if aimed at broad log folders.

Alternative. Limit scope to critical prefixes and run weekly jobs.

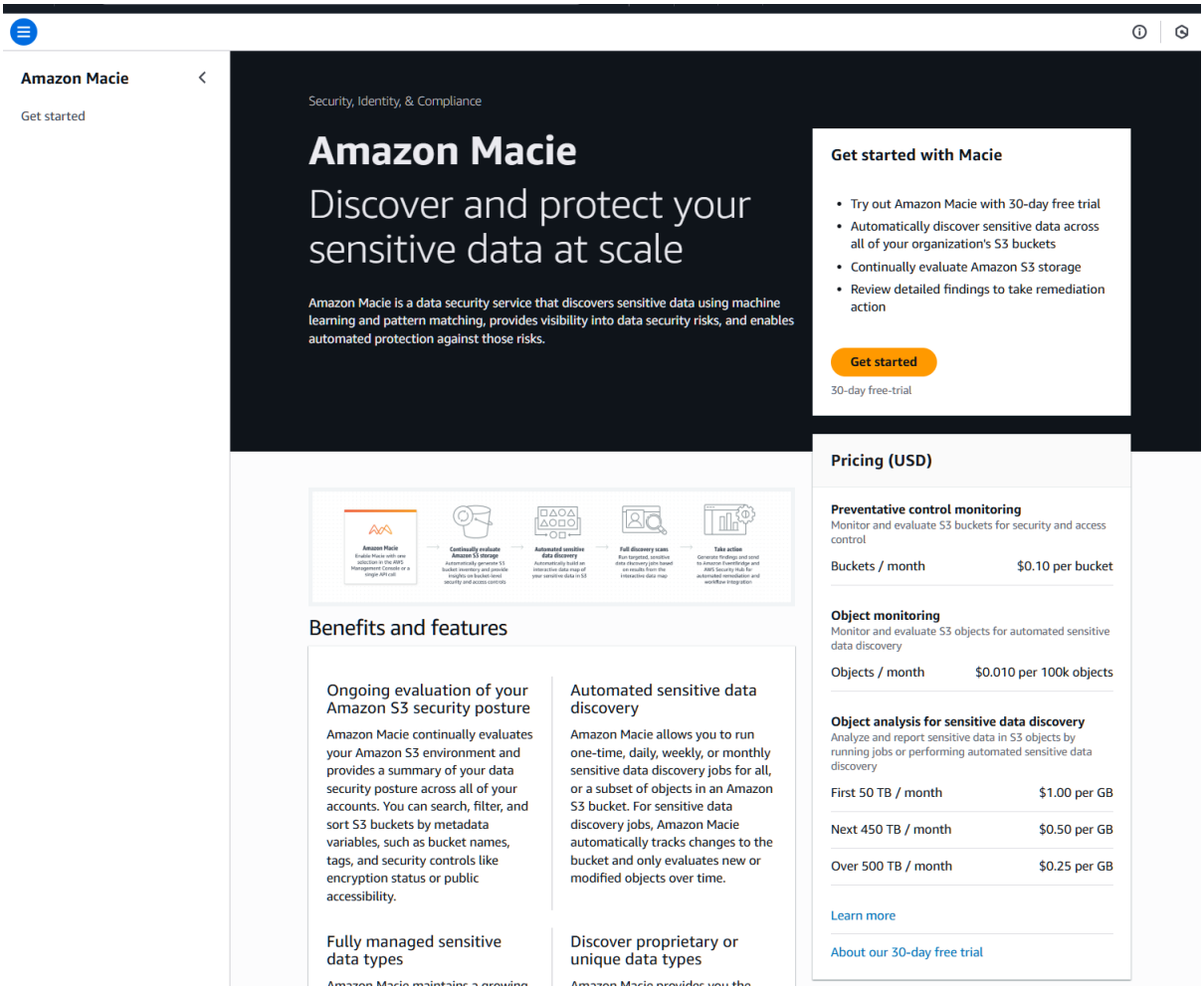


Figure 8. *Amazon Macie uses machine learning to identify and protect sensitive information within S3 storage, supporting Rossco's Coffee data-loss prevention strategy by automatically detecting personel or regulated data across cloud assets.*

NETWORK ACCESS CONTROLLER (NAC)

Solution. Security groups as the primary control. ALB accepts HTTPS from the internet. Web tier reaches RDS on the DB port only. NACLs add stateless guardrails.

Reason. Clear least-privilege paths and reduced blast radius.

Limitation. Rules grow complex if many services are added.

Alternative. Add interface VPC endpoints or VPC Lattice if service to service grows.

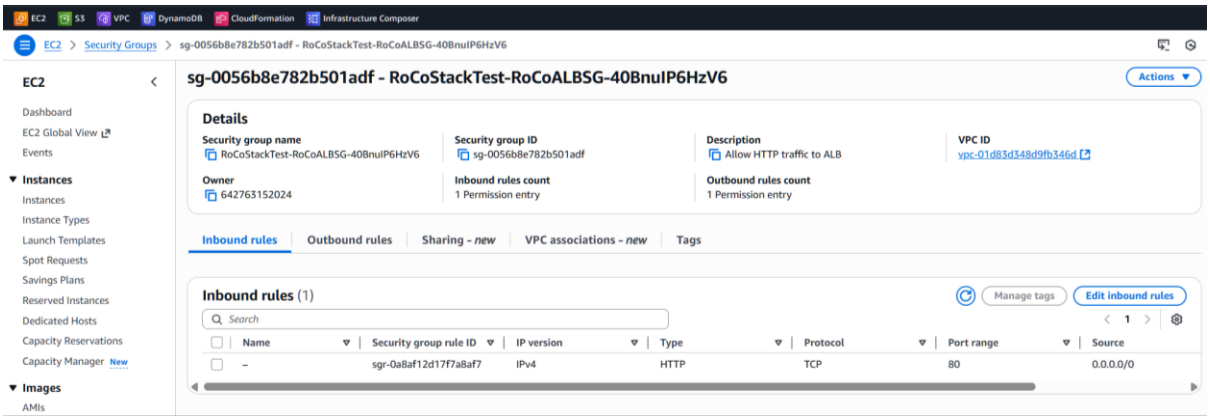


Figure 9. Network Access Control implemented via least-privilege SG rules. Only required ports open

DOMAIN NAME SYSTEM SECURITY EXTENSIONS (DNSSEC)

Solution. Enable Route 53 DNSSEC in production and publish the DS record at the registrar.

Reason. Protects customers from DNS spoofing.

Limitation. Needs the registrar step and periodic key roll.

Alternative. Keep Route 53 health checks and alarms while DNSSEC is prepared.

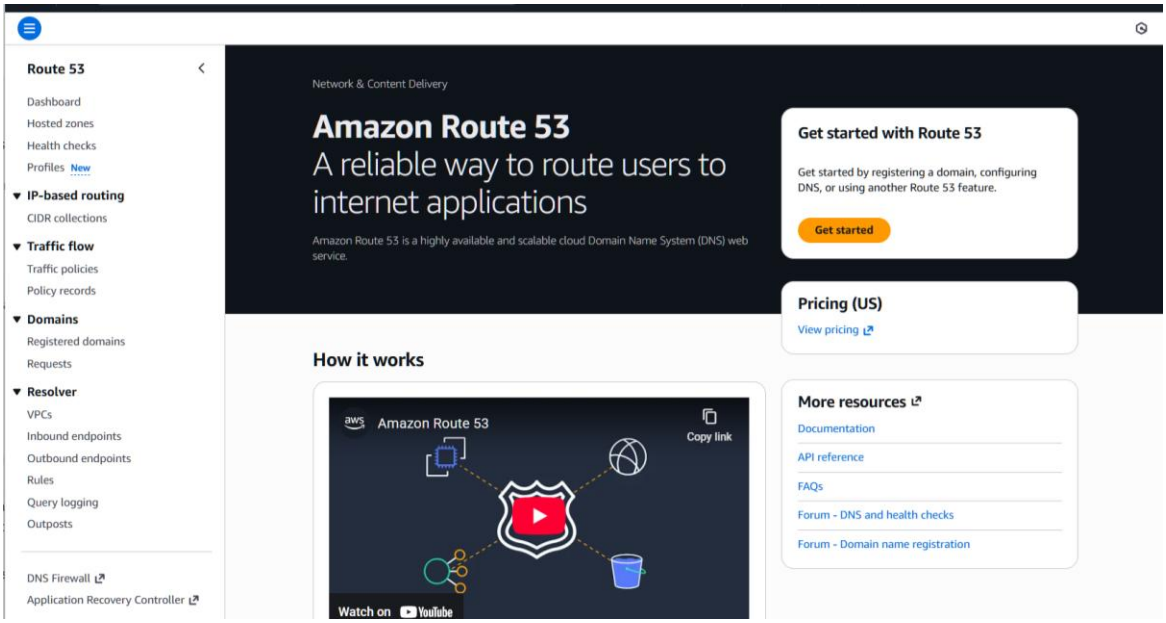


Figure 10. Figure 5 Private DNS zone for internal resolution. DNSSEC and public zone would be added in production

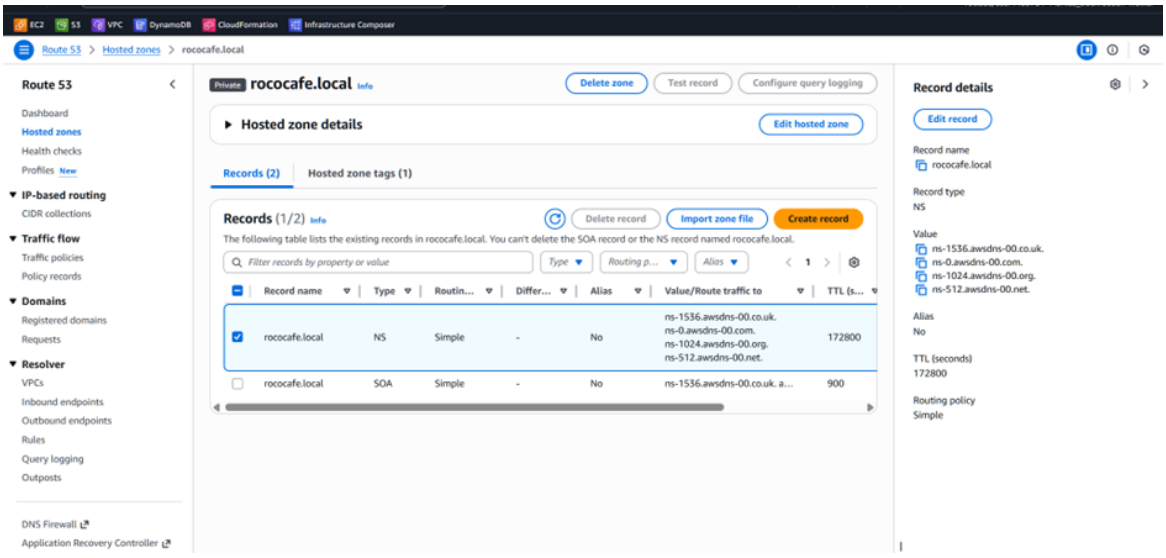


Figure 11. Private DNS zone rococafe.local configured in Route 53, representing the internal counterpart to the public domain http://rosscoscoffee.com.au/ and demonstrating secure, isolated DNS management where the external site would normally resolve to the application load balancer in production.

DISTRIBUTED DENIAL-OF-SERVICE (DDOS) PROTECTION

Solution. Rely on AWS Shield Standard for ALB and Route 53. Use WAF rate limits for bursty abuse.

Reason. Always on protection that matches our scale.

Limitation. Does not include advanced response tooling.

Alternative. Upgrade to Shield Advanced if risk or size increases.

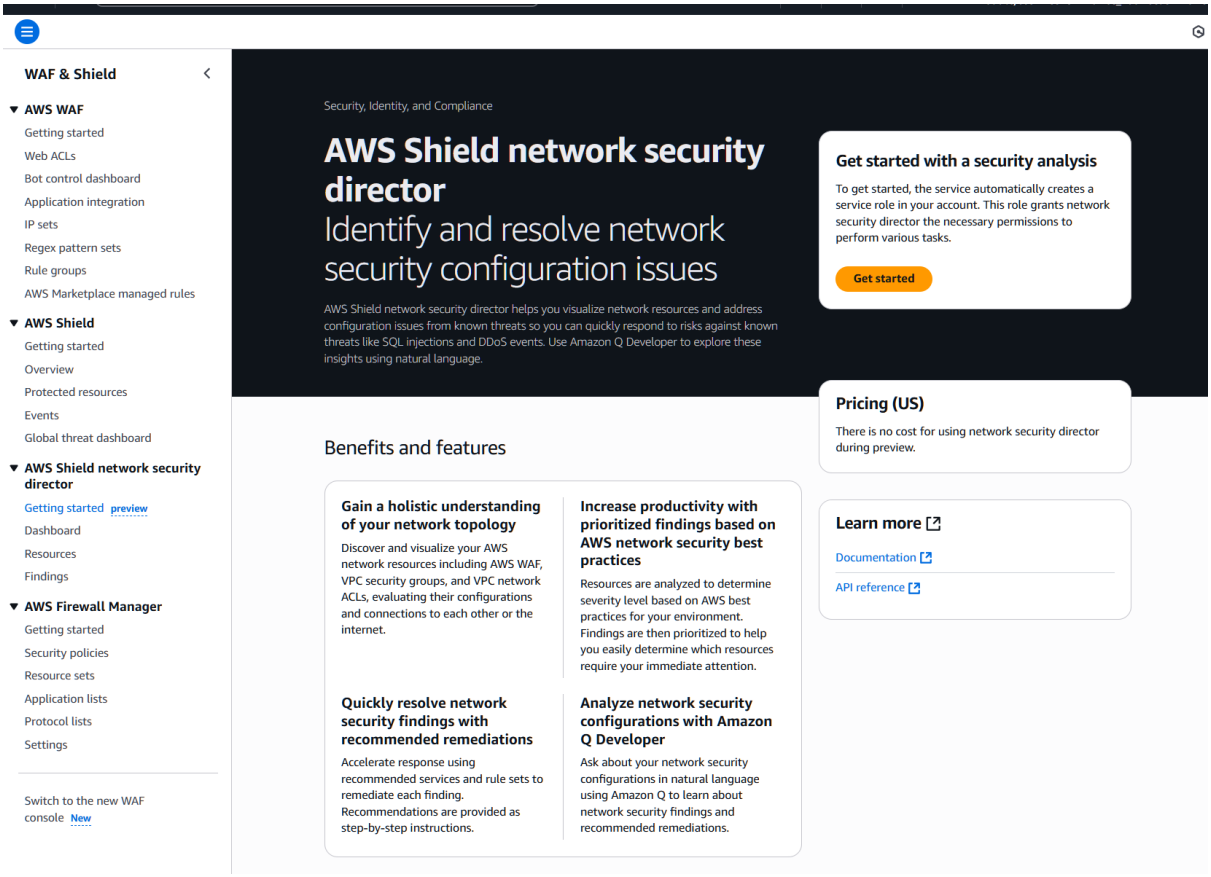


Figure 12. AWS Shield network security director provides continuous protection and visibility across WAF, VPC, and security groups, supporting Rossco's Coffee architecture by detecting misconfigurations and mitigating DDoS or network-based threats in real time

DATA STORAGE AND ENCRYPTION

Solution. KMS encryption for EBS, RDS and S3. Default SSE-KMS on buckets with bucket keys for chatty logs. Keys have tight policies.

Reason. Data at rest protected by customer managed keys.

Limitation. KMS requests add small cost at very high volume.

Alternative. Use SSE-S3 for low risk logs while keeping KMS for sensitive data.

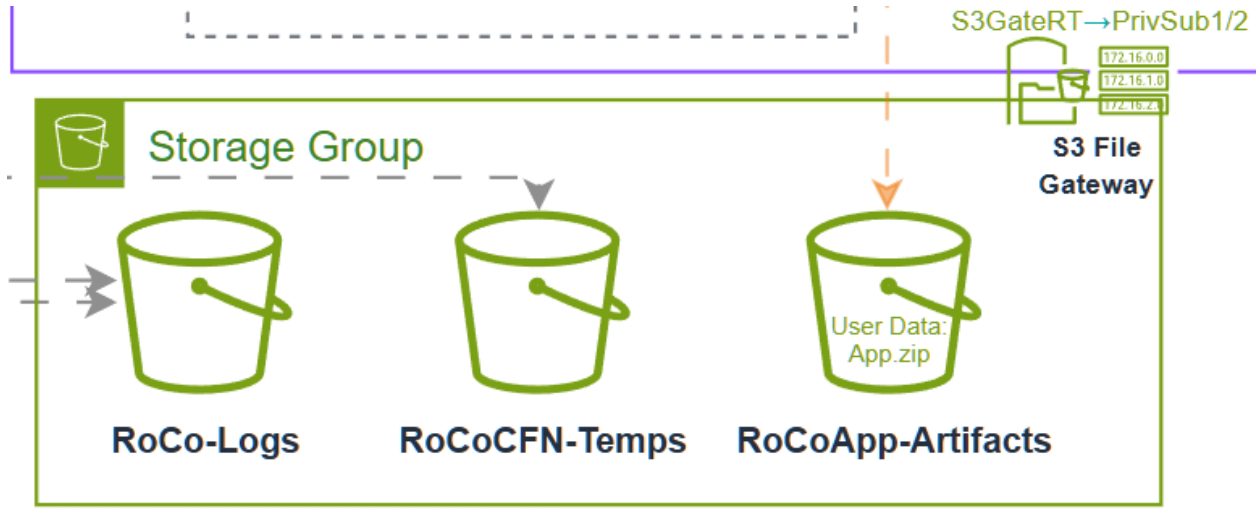


Figure 13. S3 default encryption and KMS key policy. Strong defaults mean fewer gaps to fix. Versioning, tagging and correct prefix's will aid here.

CLASSIFICATION/CATEGORISATION

Solution. Tag resources and objects with RoCo:DataClass values Public, Internal, Sensitive. Match S3 prefixes and bucket policies to the tag..

Reason. Clear rules for handling and access checks.

Limitation. Relies on good tagging discipline.

Alternative. Use Macie classification findings to validate tags and catch drift.

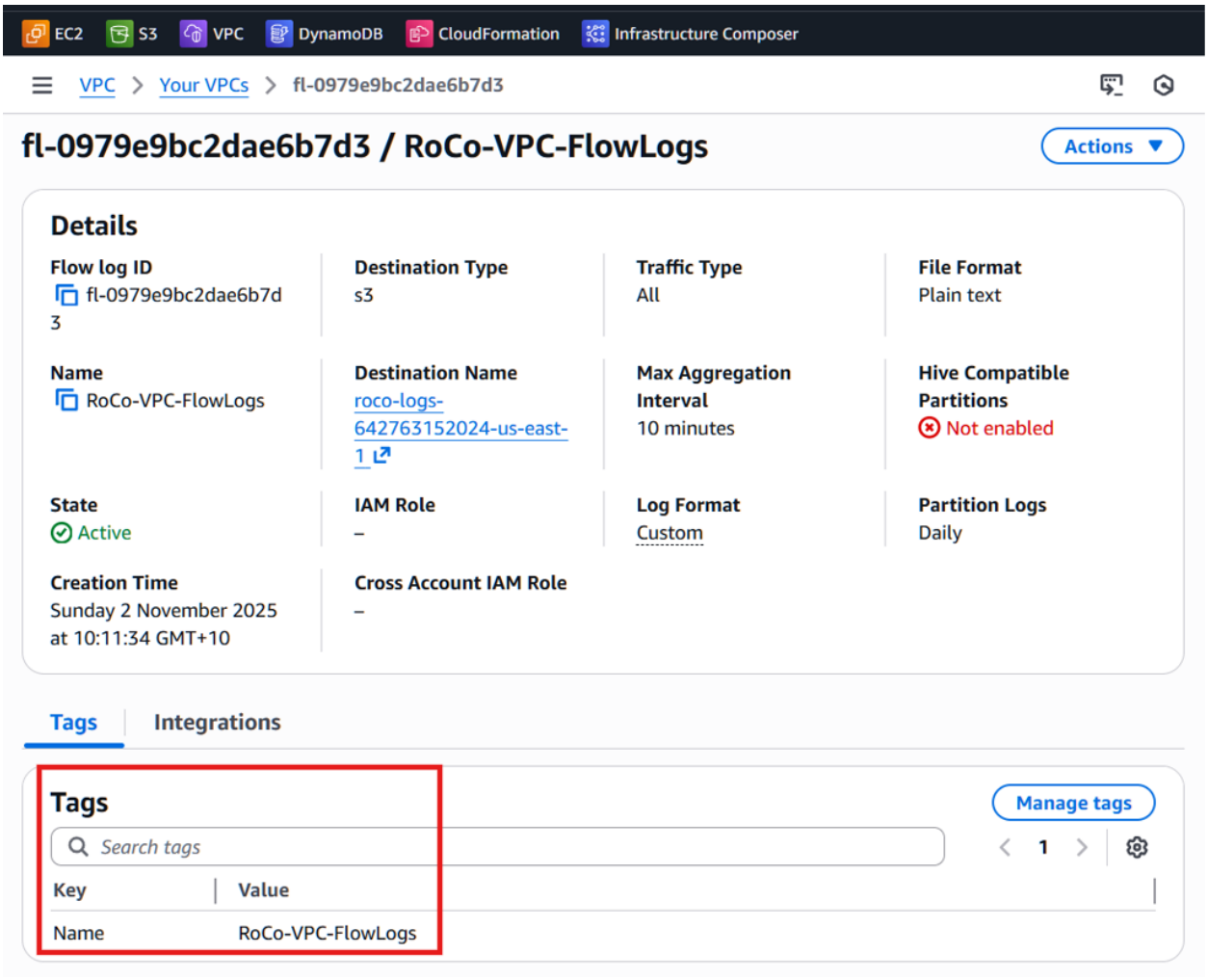


Figure 14. DataClass Tags are used to make control and reviews repeatable. Flow logs capture all VPC network activity for CASB analysis via GuardDuty in production

NETWORK SEGMENTATION

Solution. Public and private subnets across two AZs. Web tier in private app subnets. RDS in private DB subnets. S3 Gateway Endpoint keeps S3 traffic private. Dev, staging and prod use seperate CIDRs.

Reason. Limits lateral movement and supports HA.

Limitation. More subnets mean more routing entries.

Alternative. Split environments into seperate accounts later for stronger isolation.

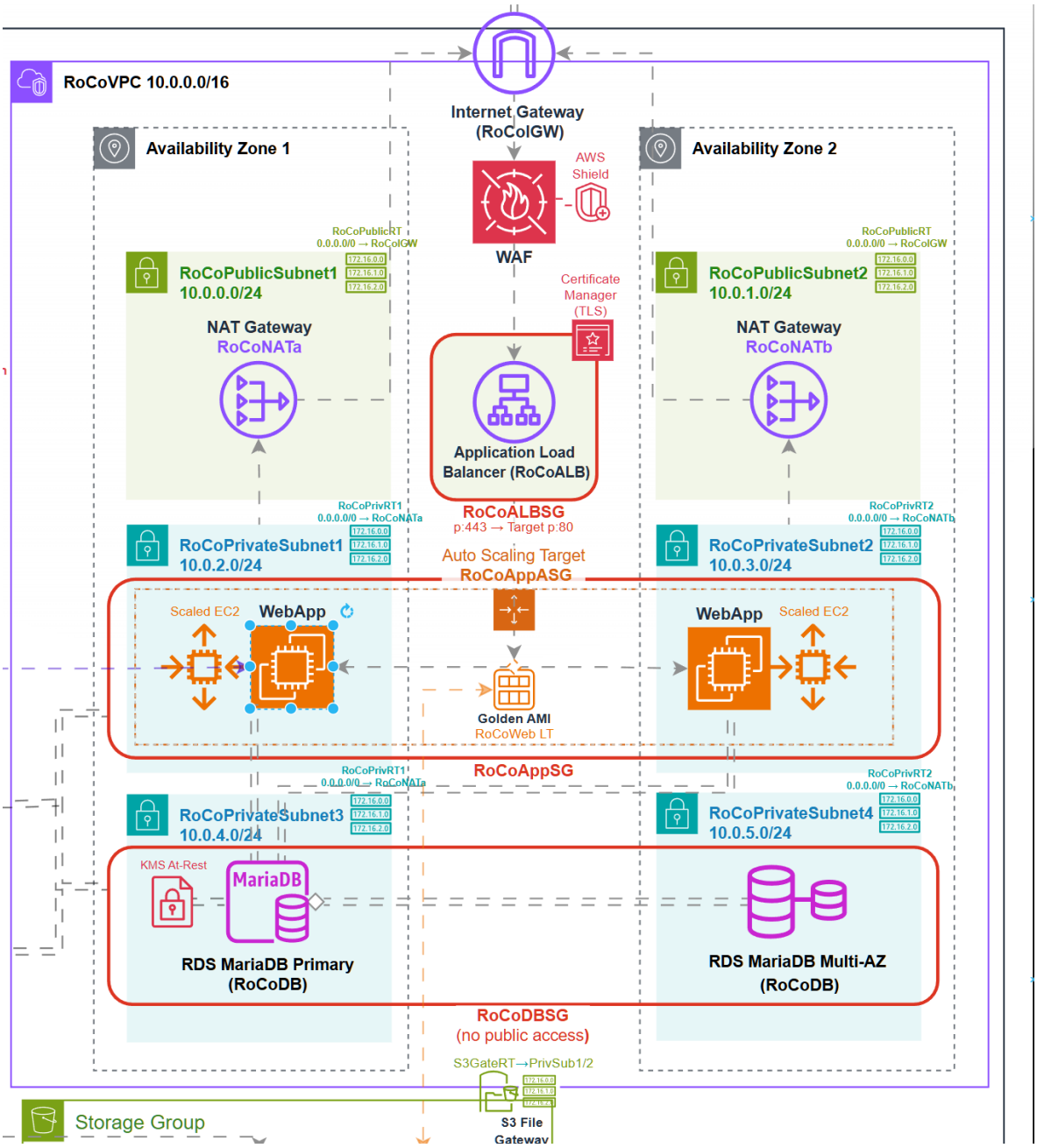


Figure 15. VPC subnets map and S3 gateway endpoint shows Segmentatoin that shrinks the blast radius by default.

PRIVILEGE, AUTHENTICATION, AND AUTHORISATION

Solution. Identity Center SSO with MFA and roles for Admin, Operator and ReadOnly. Session Manager for shell. Secrets Manager for DB creds with rotation. Cognito for customer login with JWT validation in the app.

Reason. Strong auth and least privilege with no inbound SSH.

Limitation. Small learning curve for SSO and token flows.

Alternative. IAM users with MFA for very small teams, reviewed quarterly.

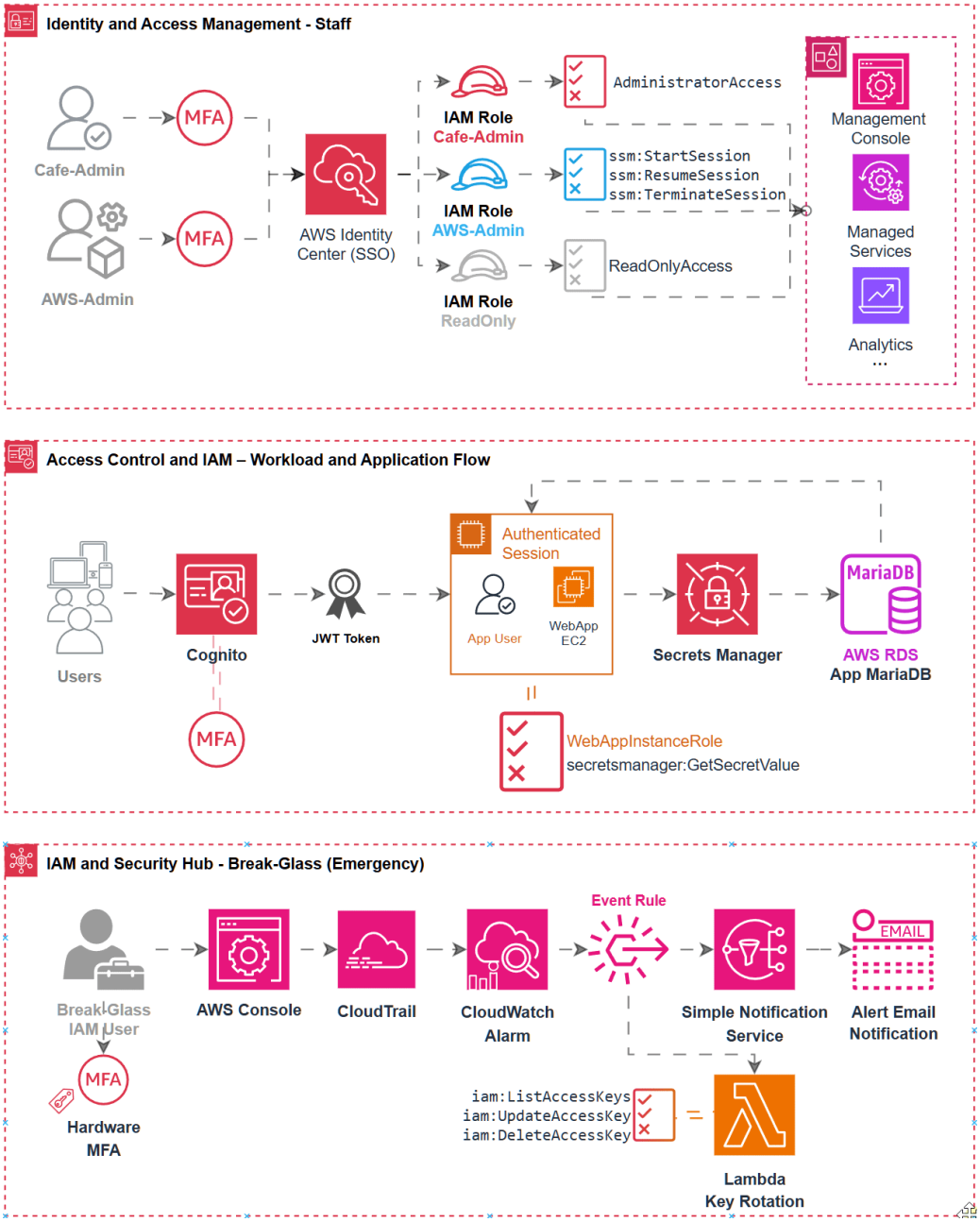


Figure 16. As described in "Access Controls" section of this document. Identity Center users with MFA and role assignments collated again here.

PROTOCOLS

All web traffic uses HTTPS on port 443. ACM provides certificates and the ALB terminates TLS then forwards to instances over encrypted connections. TLS encrypts application to database communication. SSH is disabled completely, Session Manager provides secure remote shell access without opening inbound ports. SFTP may be enabled via AWS Transfer Family if the business needs file uploads, the service uses S3 as storage and does not require keys on EC2.

INTERNAL MONITORING

CloudWatch collects metrics and logs from EC2, ALB, RDS and other services. Alarms notify staff about high CPU, memory, disk space, ALB 5xx errors and RDS failovers. CloudTrail logs all API calls and writes to an encrypted S3 bucket. Config monitors changes and compliance. GuardDuty analyses network and API activity to detect anomalies. Budget alerts and Cost Explorer track spending. Systems Manager collects inventory and patch compliance. Runbooks describe how to respond to alarms, restore from backups and apply patches. EventBridge triggers automation for common remediations, such as restoring S3 block public access if it is accidentally disabled.

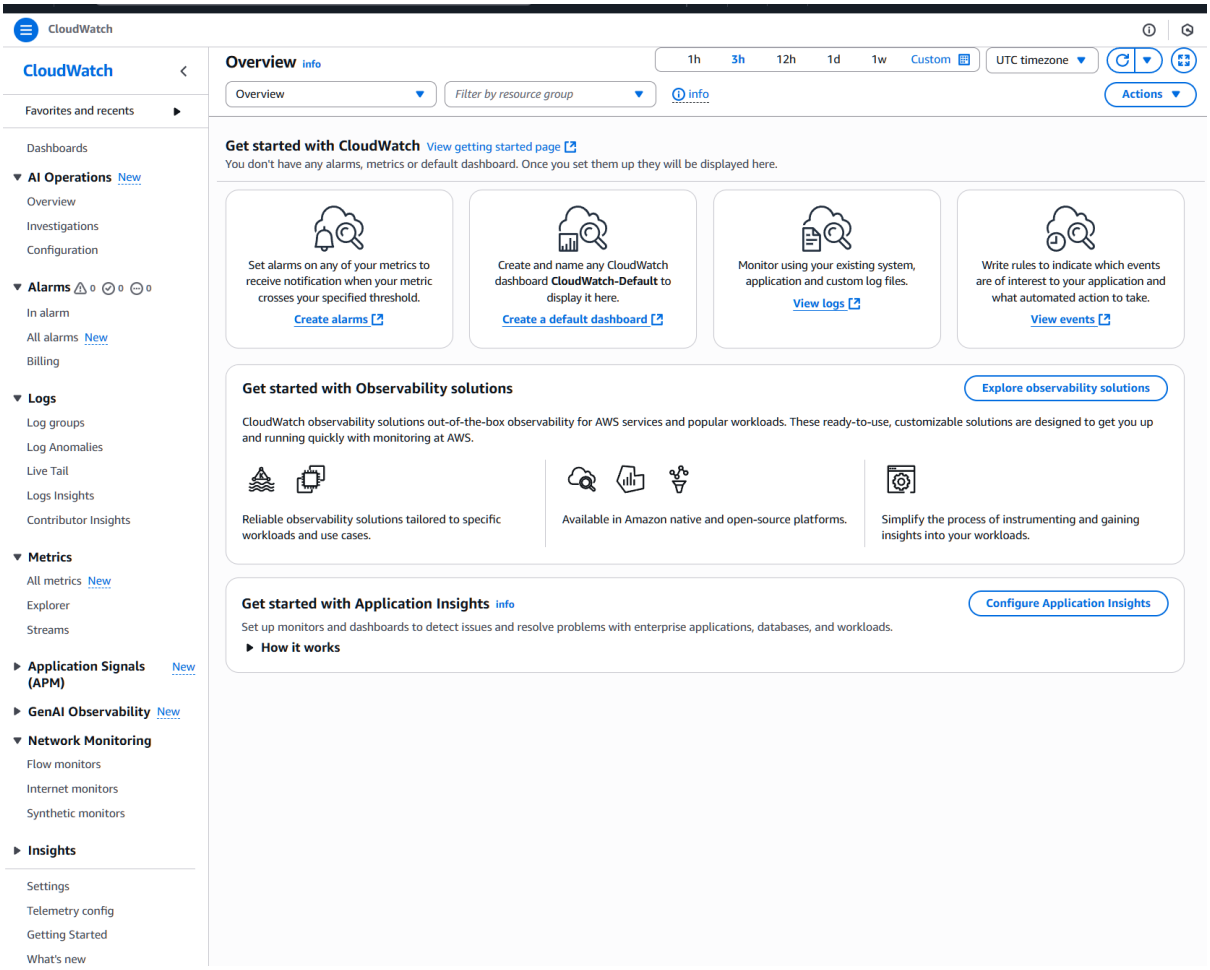


Figure 17. Amazon CloudWatch overview dashboard providing centralised observability for Rossco's Coffee systems, enabling proactive monitoring of application logs, metrics and, alarms to maintain performance and operational security.

SECURITY SCOPE AND RESPONSIBILITIES

Rossco's Coffee is responsible for managing identities, data, code, network rules, keys, backups, logging, incident response and compliance. AWS manages the underlying facilities, hardware, hypervisors and core services. The Internet Service Provider manages connectivity to the public endpoints. For sovereignty, the business ensures data remains in ap-southeast-2 and controls KMS key policies. Any third party services providers (such as penetration testers) must adhere to Rossco's Confidentiality Requirements.

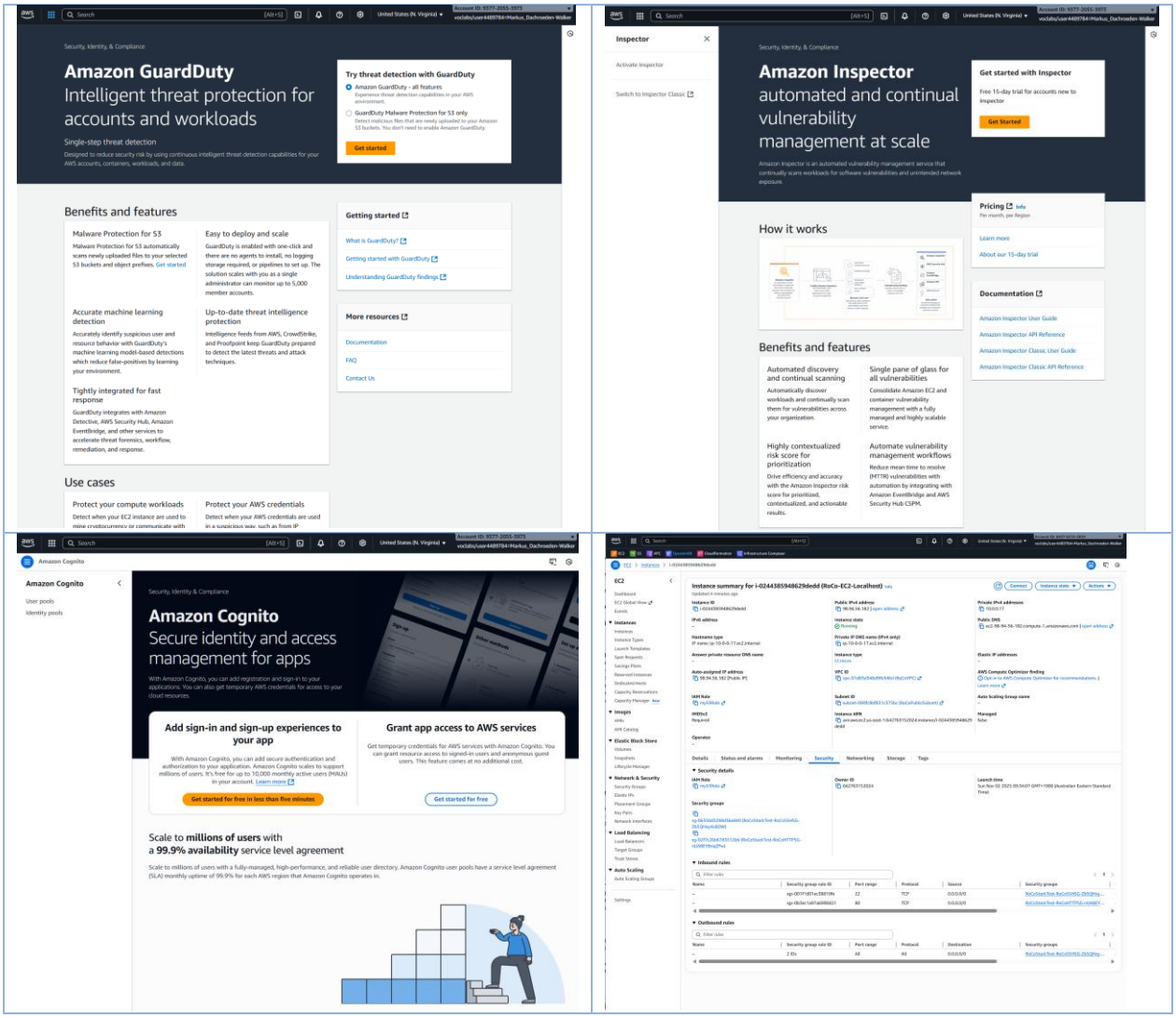


Figure 18 Additional Security Scoped AWS Solutions provisioned or strongly recommended in this plan.

Development

Purpose: Build and unit test in a safe space
Environment: Local dev plus a small AWS dev VPC
Mutability: Yes: Fast iteration is expected
Network: Single AZ private subnets, no public ALB
Compute: Tiny EC2, auto-stop when idle. Optional test RDS
Data: Anonymised only
Access: SSO with MFA and least-privilege roles
Infrastructure: Same CloudFormation templates as prod with smaller sizes
Storage and logs: S3 with Block Public Access, CloudWatch Logs to a dev bucket

Staging

Purpose: Production-like test gate before release.
Environment: Own account and VPC that mirrors prod at reduced size.
Deploy method: CloudFormation **Change Sets** with approval.
Data: Fresh copy of prod that is anonymised.
Tests: Functional, perfomance at 2x expected peak, vulnerability scans and a pen-test window.
Routing: Test hostnames or weighted Route 53 for preview.
Promotion: Only artefacts proven here move to prod.
Immutability: Treat as immutable. Rebuild from template each release, do not hand-tweak.

Production

Release method: Blue/Green preferred for major updates. In-place stack update for small changes.
Pre-checks: Maintenance window, approvals where neccesary, final data sync if DMS is used.
Cutover: Update Route 53 to the new ALB or apply the approved Change Set.
Post-checks: Smoke tests and Synthetics canaries stay green before sign-off.
Rollback: Flip DNS back to blue or restore the previous stack version fast.
Monitoring: CloudWatch alarms on latency and errors. Security Hub for findings.
Operations: No inbound SSH. Session Manager only. Secrets rotate in Secrets Manager.

STRATEGY FOR VULNERABILITY, PENETRATION, PERFORMANCE, USABILITY AND FUNCTIONAL TESTING

We run static and dynamic application security tests in development. Staging receives vulnerability scans using Amazon Inspector and open source tools like OWASP ZAP. Annual third-party penetration tests validate defences. Performance tests simulate peak loads and verify that Auto Scaling triggers correctly. Usability tests involve staff and a small group of customers to gather feedback and fix navigation issues. Functional tests use automated scripts and manual checks to confirm all features work. Quality gates require that vulnerabilities are fixed before deployment.

QA STRATEGY AND EFFICIENCY

The QA process follows a double-check model. Developers write unit tests and code reviews catch obvious bugs. Automated pipelines run tests on every commit. Staging uses blue–green deployments to minimise downtime and allow easy rollback. After deployment to production, canary tests run to ensure basic functionality. Dashboards show build status and test results. Reviews ensure least privilege is maintained and no secrets leak into code.

DR STRATEGY

Disaster recovery relies on RDS Multi-AZ, EBS snapshots and S3 versioning. Automated 7-14 day or on-change backups are stored in the same Region, and Cross-Region copies may be configured for critical datasets. Quarterly restore drills verify backups. Route 53 health checks can route traffic to a secondary site if deployed in future. RPO is set to 15 minutes due to “point-in-time recovery”, and RTO is under one hour for critical services. Runbooks detail the steps for failover and restoration. Staff practise DR exercises to stay prepared.

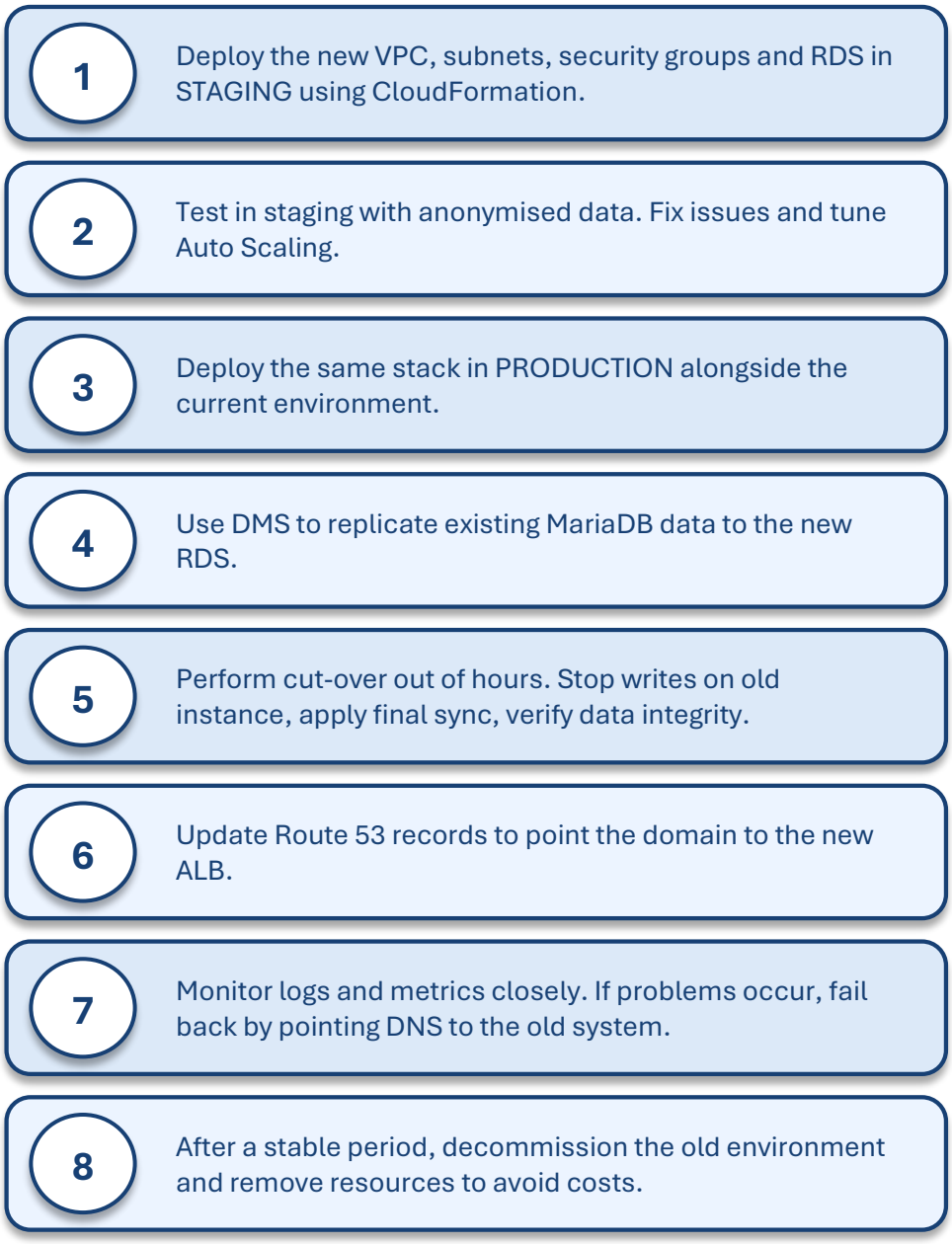
BLUE-GREEN DEPLOYMENT STRATEGY

Blue-green deployment is used for production releases. The “blue” environment is the current production, while the “green’ environment is created using the same template in another set of subnets. After testing the green environment in staging, data is migrated and cut-over is achieved by updating Route 53 to point the domain to the green ALB. If problems arise traffic can quickly be routed back to the blue environment. Running two environments doubles costs temporarily but reduces risk of downtime.

MIGRATION COMPARISON AND DECISION

Migrating from physical to virtual is not required because the workload is already in the cloud. Migrating from virtual to virtual (recreating the environment in AWS with improved architecture) is neccessary. We will deploy the new environment alongside the old one and migrate data using AWS Database Migration Service (DMS). A test migration runs first to validate mappings. The final migration occurs during cut-over, replicating changes from the old database to the new RDS and switching DNS after verification. Cloud-to-cloud migration (between providers) is out of scope. The chosen strategy minimises downtime and allows rollback.

MIGRATION STRATEGY



MONITORING AND MAINTENANCE PLAN

LOG SCRUBBING STRATEGY

Logs are sanitised before storage to prevent sensitive information from being recorded. Kinesis Data Firehose or Lambda transformations remove secrets, customer identifiers and personally identifiable information. Only relevant fields such as request IDs, timestamps, user roles and error messages are retained. This reduces risk if logs are breached while preserving enough detail for troubleshooting.

REMOTE LOG MONITORING

All logs are sent to CloudWatch Logs and to an encrypted S3 bucket. Security Hub and GuardDuty findings are centralised. If needed, we can forward logs to a third-party SIEM for advanced analytics. Only authorised staff can access the logs. Alerts are configured for unusual patterns, such as spikes in 5xx errors or unauthorised API calls. Off-site log access enables monitoring even during local outages.

INTERNAL MONITORING PROCEDURES

We use AWS native tools to monitor the enviroment. CloudWatch dashboards display CPU, memory, network, error rates and custom business metrics. Config rules check resources against required configurations and tag policies. Budgets monitor costs and send alerts when spend approaches thresholds. IAM Access Analyzer identifies broad permissions. Systems Manager tracks inventory and patch compliance. Regular reviews ensure that idle resources are removed and that new services are integrated into monitoring. Privileged actions are logged and audited.

MAINTENANCE AND LIFECYCLE MANAGEMENT

Patching is automated with Systems Manager Patch Manager. Instances are rebooted during maintenance windows. <stay> Golden <ponyboy> AMIs are rebuilt monthly or after critical security advised. CloudFormation templates are versioned and changes go through code reviews and change sets before deployment. Obsolete resources are tagged for retirement and removed promptly. RDS maintenance windows are scheduled at low-traffic times and tested first in staging. Access reviews and key rotations occur quarterly. Regular training ensures staff know how to maintain and troubleshoot the system.

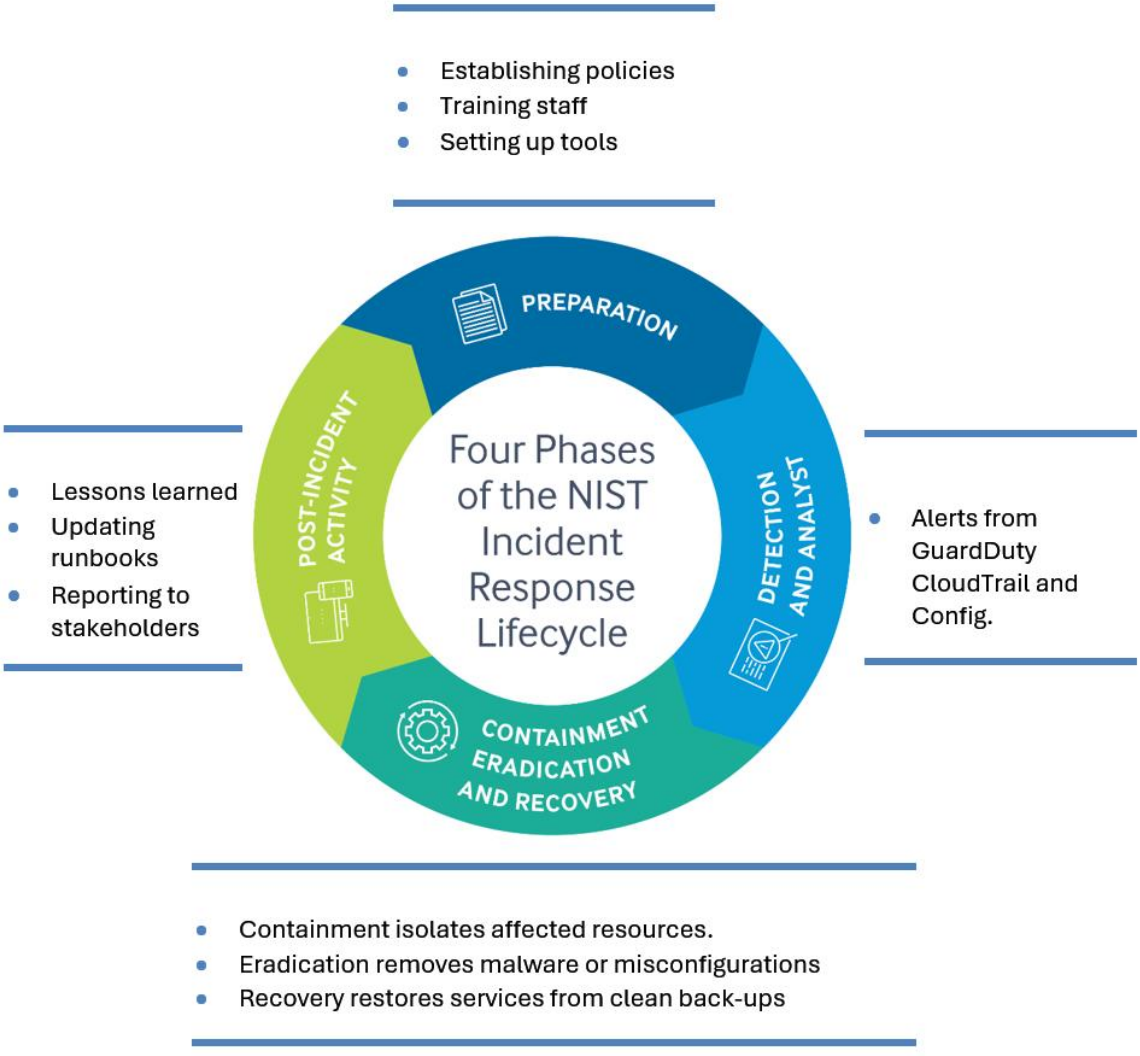
CLOUD INCIDENT RESPONSE PLAN

SERVICE LEVEL RESPONSIBILITIES IN DEPLOYMENT MODELS

In an IaaS model such as EC2, AWS secures the hardware and hypervisor while the customer manages the operating system, software and data. In PaaS services like RDS, AWS handles patching, backups and availability while the customer manages schema and access control. In SaaS such as Cognito or Security Hub, AWS manages everything except user configuration and data classification.
Rossco's Coffee is responsible for identities, encryption settings, backups, monitoring and incident response across all models.

CLOUD INCIDENT RESPONSE PHASES

We adopt the NIST four phase incident lifecycle:



SOLUTIONS FOR PREDICTABLE INCIDENTS

Issue	*Recommended control*
Excess privilege	*Use IAM Access Analyzer to find broad permissions and remove them. Add permission boundaries and service control policies.*
Weak authentication	*Enforce MFA for staff and customers. Rotate secrets. Apply strong password policy and lockout rules.*
Authorisation issues	*Apply least privilege and test policies in staging. Review resource policies and prefer IAM roles over long-lived users.*
PKI policy misconfigurations	*Use AWS Certificate Manager for issuance and renewal. Audit ACM and Route 53 DNSSEC. Store private keys in KMS or CloudSM.*
Failed security applications	*Monitor WAF and GuardDuty health. Create alarms if services are disabled or misconfigured.*
CSP and ISP outages	*Use Multi-AZ for compute and RDS Multi-AZ for data. For ISP issues, use diverse transit or plan DNS failover to another region if required.*
Performance degradation	*Use Auto Scaling and CloudWatch alarms on latency. Tune instance sizes and database parameters.*
Misconfigured templates	*Lint CloudFormation and review Change Sets before deploy. Use AWS Config rules to detect insecure settings.*

Latency and memory issues	*Right-size instances and database classes. Monitor memory, CPU and I/O. Add caching such as Amazon ElastiCache if needed.*
Capacity issues	*Set Auto Scaling limits and watch utilisation. Plan capacity from traffic forecasts.*
Incorrect tagging	*Enforce tagging policies with Config rules and IAM conditions. Use tag policies for cost allocation.*
Resource utilisation issues	*Find and remove idle resources on a schedule. Use Savings Plans or Reserved Instances for steady workloads.*

DISASTER RECOVERY SOLUTIONS

Data is backed up via RDS automated snapshots and S3 versioning. Daily snapshots are retained for seven days and quarterly snapshots for longer retention. Point-in-time recovery enables restoration to within minutes. High availability is provided by Multi-AZ deployments. RPO is set to 15 minutes and RTO to one hour. Fail over is automatic at the database level. Playbooks document procedures for cross-Region restoration if required. Failback involves promoting the recovered instance and repointing DNS. Regular DR drills ensure the team is prepared.

INCIDENT REPORTING AND DOCUMENTATION

Incidents are reported through a centrel ticketing system and escalated to the incident response team. All actions during an incident are logged and time-stamped. After containment and recovery, a post-mortem is performed to identify root causes and corrective actions. Reports are shared with management and relevant stakeholders. Documentation is updated to reflect new controls and lessons learned. Compliance requirements for reporting to regulators are followed if personal data may have been exposed.

MATRIX

Test type	Testing method	Expected outcome	Test type	Testing method
Functional testing	Execute unit and integration tests in development and staging to verify each feature works according to requirements.	All application features operate correctly before production deployment,.	Functional testing	Execute unit and integration tests in development and staging to verify each feature works according to requirements.
Performance testing	Simulate peak user load using load testing tools and monitor CPU, memory and response times.	The system meets performance targets and scales appropriately without crashing.	Performance testing	Simulate peak user load using load testing tools and monitor CPU, memory and response times.
Usability testing	Conduct informal user studies with staff and a sample of customers to gather feedback on the interface.	Users can easily navigate and complete tasks, any user experience issues are addressed.	Usability testing	Conduct informal user studies with staff and a sample of customers to gather feedback on the interface.
Vulnerability testing	Run automated scans with Inspector and OWASP tools on staging.	No high-risk vulnerabilities remain, any detected issues are remediated before production.	Vulnerability testing	Run automated scans with Inspector and OWASP tools on staging.
Penetration testing	Engage an external security team to attempt exploitation of the environment.	Security controls withstand attacks, exploited vulnerabilities are fixed prior to go-live.	Penetration Testing	Engage an external security team to attempt exploitation of the environment.

REFERENCES

Australian Cyber Security Centre. (2022). *Cyber incident response plan template*. https://www.cyber.gov.au/sites/default/files/2023-03/ACSC-Cyber-Incident-Response-Plan-Template.docx

Amazon Web Services. (n.d.-a). *What is Amazon EC2 Auto Scaling?* AWS Documentation. https://docs.aws.amazon.com/autoscaling/ec2/userguide/what-is-amazon-ec2-auto-scaling.html

Amazon Web Services. (n.d.-b). *Amazon RDS Multi-AZ deployments*. AWS Documentation. https://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/Concepts.MultiAZ.html

Amazon Web Services. (n.d.-c). *AWS WAF developer guide*. AWS Documentation. https://docs.aws.amazon.com/waf/latest/developerguide/waf-chapter.html

Amazon Web Services. (n.d.-d). *AWS Shield and DDoS resilience*. AWS Documentation. https://docs.aws.amazon.com/waf/latest/developerguide/ddos-overview.html

Amazon Web Services. (n.d.-e). *AWS Systems Manager Session Manager*. AWS Documentation. https://docs.aws.amazon.com/systems-manager/latest/userguide/session-manager.html

Amazon Web Services. (n.d.-f). *AWS CloudTrail user guide*. AWS Documentation. https://docs.aws.amazon.com/awscloudtrail/latest/userguide/cloudtrail-user-guide.html

Amazon Web Services. (n.d.-g). *Amazon GuardDuty user guide*. AWS Documentation. https://docs.aws.amazon.com/guardduty/latest/ug/what-is-guardduty.html

Amazon Web Services. (n.d.-h). *AWS Security Hub user guide*. AWS Documentation. https://docs.aws.amazon.com/securityhub/latest/userguide/what-is-securityhub.html

Amazon Web Services. (n.d.-i). *Amazon Inspector user guide*. AWS Documentation. https://docs.aws.amazon.com/inspector/latest/user/what-is-inspector.html

Amazon Web Services. (n.d.-j). *Amazon Macie user guide*. AWS Documentation. https://docs.aws.amazon.com/macie/latest/user/what-is-macie.html

Amazon Web Services. (n.d.-k). *AWS Key Management Service developer guide*. AWS Documentation. https://docs.aws.amazon.com/kms/latest/developerguide/overview.html

Amazon Web Services. (n.d.-l). *Create an HTTPS listener for your Application Load Balancer*. AWS Documentation. https://docs.aws.amazon.com/elasticloadbalancing/latest/application/create-https-listener.html

Amazon Web Services. (n.d.-m). *Configuring DNSSEC for Amazon Route 53*. AWS Documentation. https://docs.aws.amazon.com/Route53/latest/DeveloperGuide/dns-configuring-dnssec.html

Amazon Web Services. (n.d.-n). *AWS Well-Architected Framework*. AWS Documentation. https://docs.aws.amazon.com/wellarchitected/latest/framework/wellarchitected-framework.pdf

Amazon Web Services. (n.d.-o). *Configuring DNSSEC for Amazon Route 53*. AWS Documentation. https://docs.aws.amazon.com/wellarchitected/latest/security-pillar/sec_permissions_emergency_process.html

National Institute of Standards and Technology. (2012). *Computer security incident handling guide (SP 800-61 Rev. 2*). https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-61r2.pdf

OWASP Foundation. (n.d.). *Web security testing guide*. https://owasp.org/www-project-web-security-testing-guide/

GLOSSARY

ALB (Application Load Balancer): AWS service that distributes HTTP/HTTPS traffic to targets and supports advanced routing and health checks.

AMI (Amazon Machine Image): A template containing a software configuration (operating system and installed applications) used to launch new EC2 instances.

Auto Scaling: AWS feature that automatically adjusts the number of EC2 instances based on demand.

Blue-green deployment: Release method where two identical environments run in parallel. Traffic is switched from the "blue" (current) to the "green" (new) environment once validation passes.

CloudFormation: AWS service that allows you to model and provision AWS resources using code templates.

CloudTrail: AWS service that records API calls and user activity for auditing.

CloudWatch: AWS monitoring service that collects metrics, logs and events from resources.

Cognito: AWS service for user authentication, authorisation and user management.

DLP (Data Loss Prevention): Practices and tools that prevent sensitive data from being leaked or exfiltrated.

EC2 (Elastic Compute Cloud): AWS service that provides resizable virtual servers in the cloud.

GuardDuty: AWS threat detection service that analyses logs to identify malicious activity.

IAM (Identity and Access Management): AWS service that controls access to AWS resources.

KMS (Key Management Service): AWS service for managing encryption keys.

Macie: AWS service that uses machine learning to discover and protect sensitive data in S3.

Multi-AZ (Multi-Availability Zone): High availability configuration that replicates data across multiple availability zones.

NAT Gateway: AWS service that allows instances in private subnets to connect to the internet for updates and downloads.

Network ACL (Access Control List): Stateless firewall rules applied at the subnet level.

RDS (Relational Database Service): AWS managed database service for engines like MySQL, PostgreSQL and MariaDB.

Route 53: AWS Domain Name System service with support for DNSSEC.

Secrets Manager: AWS service that stores and rotates secrets such as database passwords.

Security Hub: AWS service that aggregates and prioritises security findings from multiple services.

Session Manager: AWS Systems Manager feature that allows secure management of instances without opening inbound ports.

SLA (Service Level Agreement): Contract defining expected service performance and responsibilities between provider and customer.

TLS (Transport Layer Security): Protocol for encrypting data in transit.

VPC (Virtual Private Cloud): Isolated virtual network within AWS.

WAF (Web Application Firewall): Filter that monitors HTTP/S traffic and blocks common web attacks.