drovops

SAMPLE REPORT

aws audit findings

startup name redacted · prepared by walid amine · 36 checks, read only access

health score

typical for an account that has shipped fast for two years with no dedicated owner

21
findings

3
critical, fix this week

$1,899
per month recoverable

36
checks performed

that is **39%** of the current $4,860 monthly bill, recovered without touching what customers see

score by area

observability	46
security	52
cost efficiency	64
reliability	70

observability scores lowest. the account is not telling anyone when something breaks.

where the money leaks

oversized compute	$610
no savings plans	$430
dev and staging 24/7	$340
nat gateway traffic	$185
log retention	$120
orphaned resources	$119
old volumes and snapshots	$95

total recoverable **$1,899 per month**

monthly spend

39% less

today	$4,860
after the fixes	$2,961

this week, in order

1. **close the public database.** the production database accepts connections from any ip address. one leaked password away from a breach.
2. **kill the root access keys.** active root keys mean anyone holding a copy owns the account, mfa or not.
3. **point alarms at a human.** alarms have been firing into the void for months. right now a customer is your monitoring.

all findings

21 findings, grouped by area. a star marks the ones most audits never look for.

security

6 findings

critical	the production database port is open to the whole internet	network	**breach risk**
critical	root account access keys are active and in use	identity	**account takeover**
high	★ iam access keys belonging to departed employees are still active	identity	**breach risk**
high	database credentials sit in plaintext environment variables	secrets	**credential leak**
medium	★ imdsv1 is still enabled, one ssrf bug reaches instance credentials	compute	**credential theft**
medium	★ cloudtrail records one region only, activity elsewhere is invisible	audit	**blind to intrusion**

cost

7 findings

high	compute is provisioned for roughly four times observed peak	compute	**$610 / mo**
high	★ zero savings plans coverage on a baseline that never drops	billing	**$430 / mo**
high	dev and staging run 24/7, including nights and weekends	environments	**$340 / mo**
medium	★ all traffic to s3 and ecr routes through nat, no vpc endpoints	network	**$185 / mo**
medium	★ log groups have no retention set, they grow forever	logs	**$120 / mo**
medium	★ orphaned elastic ips, an idle load balancer, incomplete multipart uploads	waste	**$119 / mo**
low	★ volumes and snapshots left behind by instances deleted long ago	storage	**$95 / mo**

reliability

4 findings

high	the database runs in a single zone with no standby	data	**extended outage**
high	★ infrastructure exists only in the console, nothing is in code	iac	**cannot rebuild**
high	backups have never been restored, nobody knows if they work	backups	**data loss**
medium	★ burstable instances silently throttle at peak, cpu credits reach zero	compute	**slow under load**

observability

4 findings

critical	★ alarms fire into an sns topic with zero subscribers, nothing reaches a human	alerting	**silent downtime**
high	★ no external uptime check, downtime is reported by customers	monitoring	**silent downtime**
medium	no billing alarm, a cost spike would run for a full month unnoticed	cost	**surprise bill**
medium	no dashboard, there is no one place that answers is it healthy	visibility	**slow diagnosis**

21 findings is normal. this is what two years of shipping without a devops owner looks like.

the full checklist

every audit runs the same 36 checks, in the same order, whether the account is healthy or not. the findings differ, the checklist does not.

security

- ✓ root account mfa and usage
- ✓ iam users, keys and key age
- ✓ over permissive iam policies
- ✓ public s3 buckets and objects
- ✓ security group exposure to the internet
- ✓ secrets handling and storage
- ✓ imdsv1 versus imdsv2
- ✓ cloudtrail coverage and regions
- ✓ guardduty and threat detection

cost

- ✓ instance sizing against real load
- ✓ savings plans and reserved coverage
- ✓ non production running hours
- ✓ nat gateway and data transfer
- ✓ log retention and volume
- ✓ orphaned ips, volumes, snapshots
- ✓ idle load balancers and endpoints
- ✓ s3 storage class and lifecycle
- ✓ incomplete multipart uploads

reliability

- ✓ database multi zone and failover
- ✓ backup configuration
- ✓ restore actually tested
- ✓ load balancer health checks
- ✓ auto scaling and capacity headroom
- ✓ cpu credit exhaustion
- ✓ single points of failure
- ✓ tls certificate expiry
- ✓ infrastructure as code coverage

observability

- ✓ alarms on errors and latency
- ✓ alarm delivery to a human
- ✓ external uptime check
- ✓ billing and anomaly alarms
- ✓ centralised logs and searchability
- ✓ log retention policy
- ✓ dashboard for the whole system
- ✓ deploy visibility and history
- ✓ on call path and escalation

36 checks performed on this account. **21 came back with something worth fixing.** the other 15 are listed here too, because knowing what is already fine is half of what an audit is for.

the 30 day plan

the order matters. risk first, visibility second, money third, and only then the rebuild.

WEEK 1 stop the bleeding

the three critical findings are closed. by friday nothing is one mistake away from a breach or a full outage.

- close the database port to the internet
- remove the root access keys
- subscribe a real phone to the alarm topic
- rotate keys of departed employees
- move credentials into secrets manager
- promote the database to a standby in a second zone

WEEK 2 turn the lights on

you stop finding out from customers. the system tells you first, and there is one screen that answers is it healthy.

- alarms on errors, latency, and spend
- external uptime check from outside aws
- log retention set, logs searchable
- one dashboard a founder can read
- on call path written down

WEEK 3 stop the leaks

the waste in the bill gets cut, without touching what customers see. the savings start the month they are made.

- right size compute to real peak
- savings plans on the steady baseline
- dev and staging off at night and weekends
- vpc endpoints for s3 and ecr, nat traffic drops
- delete orphaned ips, volumes, snapshots

WEEK 4 make it rebuildable

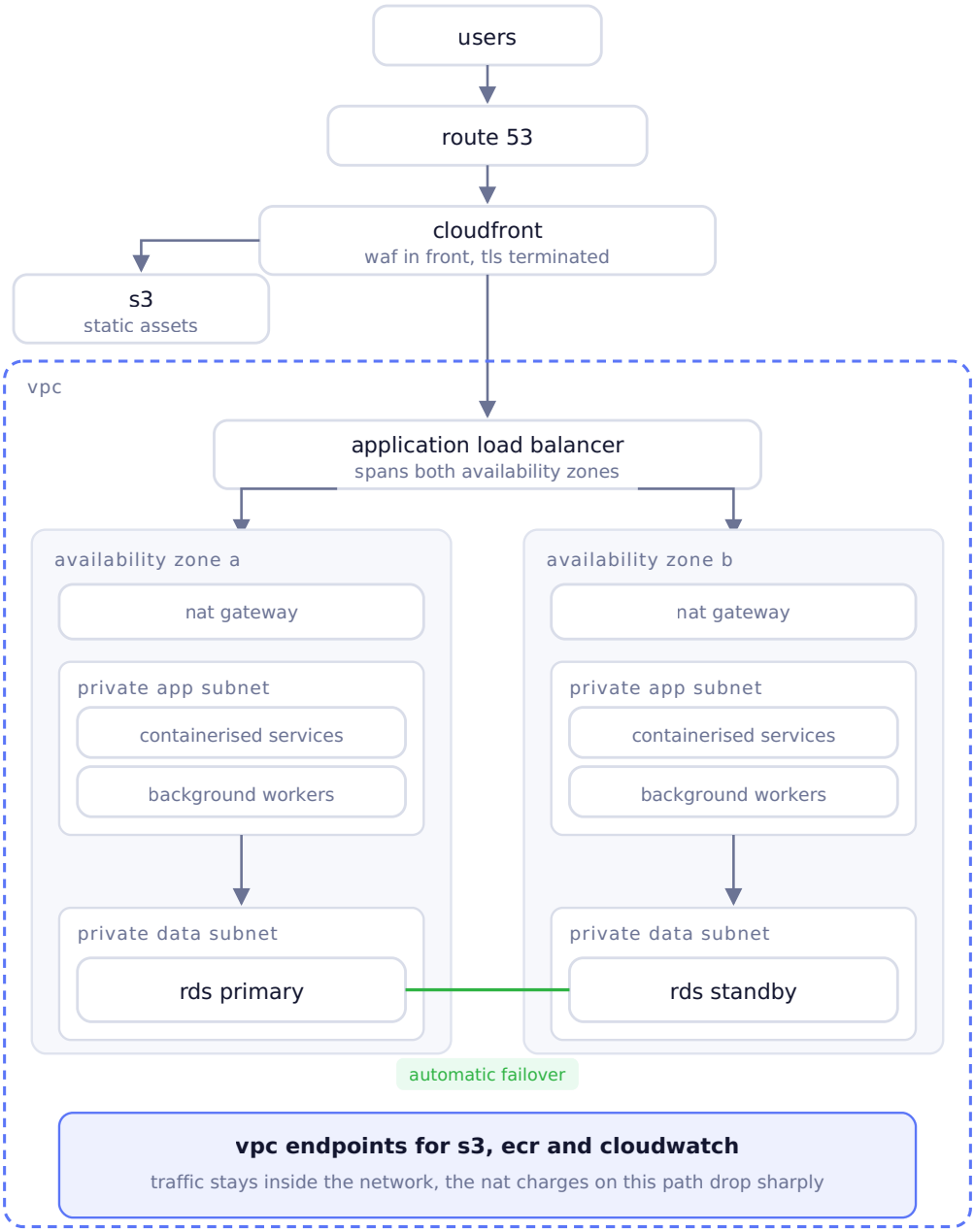
the setup stops living on one person's laptop. it is in code, reviewable, and can be brought back from nothing.

- infrastructure captured in terraform
- deploys with zero downtime and one click rollback
- backups restored in a real test
- runbook for the top failures
- handover, you own everything

the audit is free and read only. this plan is what the monthly subscription executes.

target architecture

the end state the 30 day plan builds toward. nothing exotic, just the boring setup that stays up.



every subnet above is private. nothing holds a public address except the load balancer.

the deploy pipeline

- github as the source of truth
- ci runs tests and builds the image
- image pushed to a private registry
- rolling deploy with zero downtime
- one click rollback to the last good build
- every deploy recorded and visible

monitoring that reaches a human

- alarms on errors, latency and spend
- routed to a phone, not an inbox nobody reads
- external uptime check from outside aws
- one dashboard a founder can read in ten seconds
- monthly report: what we did, what we saved, what is next

the security baseline

- guardduty watching for threats
- cloudtrail in every region
- secrets in secrets manager, never in env files
- least privilege iam, no shared keys
- mfa on every human account
- imdsv2 enforced
- encryption at rest and in transit

everything above is terraform. versioned, reviewable, and rebuildable from code.

how we work

what happens next, what it costs you to find out, and where it stops.

01

the audit

thirty minutes on a call, then read only access to the account. we run the 36 checks and send this report. it is free, and it is yours either way, whether you work with us or hand it to someone else.

02

the first 30 days

the plan on page four gets executed in that order. the critical findings close in week one. you approve anything that changes what customers see.

03

then we keep it

infrastructure stops being your problem. one fixed monthly price, and one report a month: what we did, what we saved, what is next.

what the subscription covers

- your aws account, owned and watched every day
- when something breaks, we get the alarm and we handle it
- cost kept under control as you grow
- security baseline maintained, not set once
- deploys, pipelines and rollbacks kept working
- backups tested on a schedule, not assumed
- one fixed monthly price, no hourly billing, no surprise invoices

what it does not cover

- writing your product code
- feature work inside your application
- office it, laptops and email

our promises

- the account stays in your name and on your billing, we work inside it
- leaving needs no migration. revoke our access and everything keeps running
- no rebuild you did not ask for

this report is an illustrative sample. the account, the findings and the numbers are composites drawn from patterns seen across real audits, not a real client.