

S-05, Anonymized sample from a real engagement

Daily progress report, sprint 1

An end of day report from a recovery sprint, with five work streams, open problems and next steps.

Anonymized sample from a real sprint for a US investment analytics platform, 2026

What this is. The end of day progress report from the first sprint, while the platform was being recovered. It lists what was done in each work stream, what is still open and why, and what happens next. Domains, mail provider names and file paths that named the product are replaced. The structure and the level of detail are as sent.

Progress report, [date]

Work streams	Files changed	Python files audited	Open problems
5	14	11	3

1. Summary

Today covered infrastructure fixes, code quality hardening, report updates and the next client message. Five work streams ran in parallel. They touched Terraform, the Python backend, the HTML reports and the outbound message.

The work breaks down into two infrastructure fixes (load balancer redirect rules and an email DNS correction), one code quality audit across 11 Python files, and two documentation updates (the implementation report and the sprint plan). All code changes are written and tested. Two items stay blocked on outside steps before deployment.

2. What was done today

Work stream 1, root domain redirect

Problem. Requests to the bare domain returned 503 and 504 errors. The root A record points to the load balancer, but the load balancer has no healthy backend targets. Every request hit the default action and failed.

Solution. I added four listener rules to the load balancer Terraform file. They catch the traffic before it reaches the target group.

- HTTP on the bare domain redirects 301 to the app address
- HTTPS on the bare domain redirects 301 to the app address
- HTTP on www redirects 301 to the app address
- HTTPS on www redirects 301 to the app address

Status. Code ready, waiting for the Terraform apply.

Work stream 2, email DNS fix (DSN 5.1.0 bounce)

Problem. Email to the support address bounced with DSN 5.1.0. The MX records pointed to the old mail host while the mailbox lives with the new mail provider. Inbound mail went to retired servers and was rejected.

Solution, in the DNS Terraform file.

- MX record moved from the old mail host to the new provider
- SPF record updated to the new provider, keeping the cloud mail service for outbound reports
- DKIM CNAMEs added for both of the new provider's selectors
- Autodiscover CNAME added so desktop mail clients configure themselves

Status. Code ready. Waiting for the client to confirm the mail tenant name and that the mailbox exists, then the Terraform apply.

Work stream 3, coding rules audit

Scope. I audited 11 Python files across the backend against the NASA Power of 10 rules, adapted to Python. Eight files had violations and three were already compliant.

Fixes applied.

- Rule 3, bounded memory. Added hard bounds in 6 files, among them the HTTP monitor, the holdings scraper, two price commands and one API view
- Rule 4, functions under 60 lines. Split one long per ticker function into 3 methods
- Rule 5, assertions on inputs and outputs. Added across all 8 files with violations
- Rule 6, smallest data scope. Made mutable module globals immutable in 2 files

- Rule 7, check every return value. Added error handling, return checks and timeouts in 6 files

Status. Code changes complete, not yet committed.

Work stream 4, implementation report update

- Added a summary that explains why one reported fault was five
- Expanded the redirect section to cover the root domain errors and the fix
- Rewrote the email DNS section with the bounce explanation and the provider steps
- Added an impact map for the index data scraper
- Updated the blocked items, the deploy order and the report date

Status. Complete.

Work stream 5, sprint plan update

- Retitled it as a resolution report
- Marked all five original issues resolved, each with a status tag
- Added the root domain redirect as in progress
- Updated the resolution summary to 12 deployed, 1 in progress and 1 client step

Status. Complete.

3. Open problems

Problem 1, root domain gateway timeout

The bare domain now returns 504, where it returned 503 earlier today. The load balancer has no healthy targets for its default action, so every request times out. The redirect rules are written but not applied. Once the apply runs, traffic to the bare domain goes to the app address before it reaches the unhealthy target group. The app itself works through the CDN.

Problem 2, API backend down

The API address also returns 503 and 504. No backend tasks are running, so the target group has no healthy targets. This was the state before this sprint and no change today caused it. The redirect fix keeps visitors away from it, because the static frontend doesn't depend on the backend.

Problem 3, email bounce, waiting on the client

The bounce stays until these steps finish, in this order.

1. The client confirms the mail tenant name

2. The client confirms the support mailbox exists
3. The Terraform apply deploys the new MX, SPF and DKIM records
4. DNS propagation finishes, up to 1 hour at a TTL of 3600

4. Next steps

Today and tonight

- Commit all Terraform and Python changes
- Client confirms the tenant name and the mailbox
- Run the Terraform apply for the redirect rules and the DNS changes
- Check that the bare domain redirects to the app address
- Check email delivery to the support address

This week

- Watch one full daily scheduler cycle end to end
- Check that every scheduled trigger fires
- Confirm that report generation and email dispatch complete
- Test delivery from outside senders

Later

- Find out why the backend tasks aren't running
- Start or scale the backend tasks if the API is needed
- Raise the DMARC policy from none to quarantine once mail flow is stable

5. File inventory

Area	Files	State
Terraform	2 (load balancer rules, DNS records)	Changed, not committed
Python backend	9 (monitor, price and holdings commands, one view, market data helpers)	Changed, not committed
Reports	2 (implementation report, sprint plan)	Updated
Client message	1 draft	Created

Axdale, Michael Lip. Warsaw, Poland (CET). michael@axdale.com. www.axdale.com. Code record github.com/theluckystrike. Names, tickers and amounts are replaced.