

ILLUSTRATIVE Fictional product and assumed figures, to show the format of a first assessment. Not a client report.

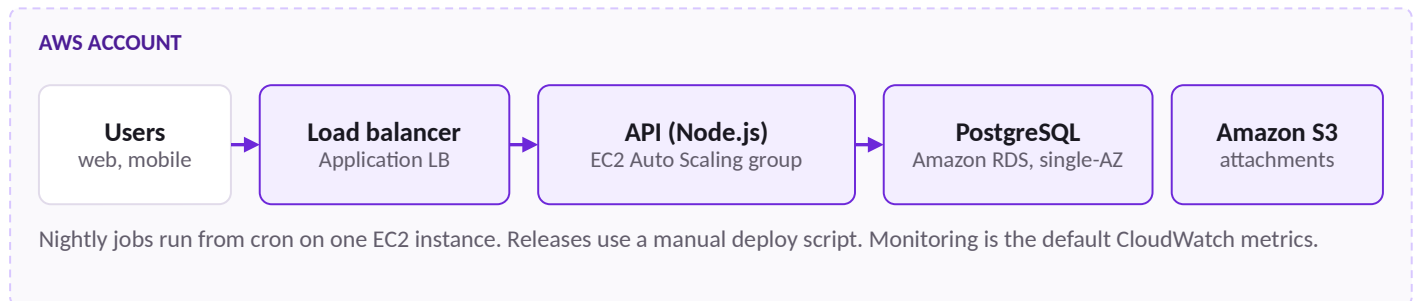
What a first assessment delivers: a completed sample

A finished example of the findings, options and first milestone a buyer receives. The product, the team and every number here are invented for illustration.

01 THE SCENARIO (FICTIONAL)

ITEM	ASSUMED SITUATION
Product	Example Scheduling, a B2B field-service scheduling application used by US businesses
Users	About 180 business customers and 12,000 weekly active users
Busiest hour	Monday 7 to 9 am Eastern, when crews are dispatched
Team	Six engineers; no dedicated DevOps or on-call owner
Why now	A large customer's renewal asks for evidence that recovery has been tested

02 THE CURRENT SYSTEM (ASSUMED)



03 BASELINE EVIDENCE (ASSUMED VALUES)

MEASURE	ASSUMED VALUE	WHERE IT WOULD COME FROM
AWS spend	\$14,200 per month	Cost and usage report
Incidents	9 in 90 days, 5 during the Monday peak	Support ticket history
Calendar API latency (p95)	2.8 s at peak; 0.6 s off-peak	Load balancer and API logs
Releases	About 3 a week; 2 rollbacks in 90 days	Deploy log
EC2 CPU, weekly average	18%	CloudWatch
Database restore tests	No documented restore test in the past 12 months	Team's change and test records; owner confirmation

Scope of this sample. Examined: architecture walkthrough, tickets, deploy logs, metrics, cost report and the database slow-query log. Not examined: security controls, penetration testing or compliance. A real assessment states what it did and did not examine in the same way.

ILLUSTRATIVE Fictional product and assumed figures, to show the format of a first assessment. Not a client report.

Findings, ranked by severity

Each finding links what was seen to its business effect and the action we propose.

F1	APPLICATION	HIGH	Confidence: High
FINDING	BUSINESS EFFECT		
The calendar view loads each crew's schedule with a separate database query, so a dispatcher's screen issues hundreds of queries at once.	Dispatchers wait or time out at the busiest hour. This pattern accounts for 5 of the 9 incidents.		
EVIDENCE	PROPOSED ACTION		
Slow-query log, a sampled request trace, and the 5 peak-hour incident tickets.	Batch the schedule query, add a composite index, and load-test at twice the assumed Monday peak.		
F2	DATA AND RECOVERY	HIGH	Confidence: High
FINDING	BUSINESS EFFECT		
The PostgreSQL database runs in one Availability Zone, and there is no documented test of restoring its backups.	Recovery time is unknown, and the renewal's request for recovery evidence cannot be met.		
EVIDENCE	PROPOSED ACTION		
RDS configuration; no restore test in the team's change and test records, confirmed with the database owner.	Run and time a restore test against an agreed restore objective and write the runbook. Assess Multi-AZ separately, for availability.		
F3	OPERATIONS	MEDIUM	Confidence: High
FINDING	BUSINESS EFFECT		
Nightly jobs run from cron on a single instance, with no alert when a job fails.	Failures are found by customers the next morning.		
EVIDENCE	PROPOSED ACTION		
Cron logs and two tickets where customers reported missing overnight updates.	Add failure alarms now; consider scheduled containers later.		
F4	DELIVERY	MEDIUM	Confidence: Medium
FINDING	BUSINESS EFFECT		
Releases use a manual script with no automated test gate.	Regressions reach production and rollbacks are improvised.		
EVIDENCE	PROPOSED ACTION		
Deploy log, including the two rollbacks.	A pipeline for the API with a test gate and a rehearsed rollback.		
F5	COST	LOW	Confidence: Medium
FINDING	BUSINESS EFFECT		
EC2 capacity is sized for the Monday peak all week; average CPU is 18%.	Spend on idle capacity. The size of the saving depends on F1, because the query fix changes the capacity the peak needs.		
EVIDENCE	PROPOSED ACTION		
Cost and usage report and CloudWatch.	Revisit scaling after F1 is fixed and measured. Not in the first milestone.		

ILLUSTRATIVE Fictional product and assumed figures, to show the format of a first assessment. Not a client report.

Options and the recommendation

Three ways to respond to the findings, including improving what exists. Each is compared on the same terms.

	A. IMPROVE IN PLACE	B. MOVE THE API TO CONTAINERS NOW	C. ADD CAPACITY AND WAIT
What it involves	Fix F1, test and document recovery (F2), alarm the nightly jobs (F3), add a release pipeline (F4).	Everything in A, plus moving the API from EC2 to Amazon ECS on Fargate in the same milestone.	Larger EC2 instances and a bigger database before the busy season. No code changes.
Dependencies	Staging environment, a maintenance window for the database, customer test data.	All of A, plus container build, new deployment patterns and team training.	Budget approval only.
Trade-offs	Leaves the hosting model as it is for now.	More change at once; containers do not fix the slow query by themselves.	Higher monthly cost; incidents and recovery risk remain.
Reversibility	High. Each change ships and rolls back on its own.	Medium. Two kinds of change are tangled together.	High, but nothing is learned.
Assessment	Recommended	Later, if evidence supports it	Not recommended

Why A. It removes the cause of most incidents, produces the recovery evidence the renewal asks for, and every change can be reversed on its own. Containers may still make sense, but only after the query fix shows how much capacity the peak really needs.

WHAT WOULD CHANGE THIS RECOMMENDATION

- A load test showing the API still fails at peak after the query fix: that would bring B forward.
- A product plan that needs independent scaling or more services soon: B becomes part of the roadmap.
- A restore test that misses the agreed restore objective: the recovery design is revised within the first milestone. Multi-AZ is judged separately, against the availability need, and the restore test and data checks stay either way.

ILLUSTRATIVE Fictional product and assumed figures, to show the format of a first assessment. Not a client report.

The proposed first milestone

Peak-hour reliability and recovery evidence. When it is done, the Monday dispatch screen holds up under load, and there is tested proof that the database can be restored.

INCLUDED

- F1 query fix, index and load test
- F2 restore test, timing and runbook
- F3 failure alarms for nightly jobs
- F4 pipeline with test gate for the API

NOT INCLUDED

- Moving to containers (option B)
- Cost changes (F5)
- Security review or penetration test
- New features or other services

ACCEPTANCE TESTS (ASSUMED THRESHOLDS)

TEST	PASSES WHEN	EVIDENCE
Calendar API under load	p95 at or under 0.8 s at twice the Monday peak	Load-test report
Database restore	Restore meets the agreed restore objective; time recorded; data checks pass	Restore log and runbook
Nightly job failure	A forced failure raises an alert within 10 minutes	Alarm test record
Release safety	A failing test blocks release; rollback rehearsed	Pipeline run history

OWNERS, INPUTS AND GATES

	AMPITY	EXAMPLE SCHEDULING
Owners	Technical lead for delivery and evidence	CTO approves; one engineer for data and domain questions
Inputs	Test plan, scripts and change records	Read access to the AWS account, staging, test data, a maintenance window
Handover	Code in the customer's repositories, runbook, decision records, load-test scripts and alarm configuration as code	Owns all of it from day one

GATE 1

Plan and test approach agreed

GATE 2

Staging results reviewed

GATE 3

Production change approved by the customer

GATE 4

Handover accepted

Commercial decision. Price and timing are set in the proposal for this milestone, once scope is confirmed. Any AWS program support is assessed separately and is not assumed here.