



AWS SECURITY, COMPLIANCE & GOVERNANCE · SAMPLE DELIVERABLE

Criticals Closed and Workload Guardrails

Week 3 of 6: Criticals and guardrails · prepared for Tidepost, Inc.

Prepared by Charles Sieg · charles.sieg@vantalect.com · vantalect.com

January 23, 2026 · AWS Security, Compliance & Governance Audit: The Hardening (six weeks)

SAMPLE DELIVERABLE

This is a sample deliverable. Tidepost, Inc. is a fictional company: its people, products, users, accounts, investors, and figures are invented for this sample, and any resemblance to a real company, person, product, or publication is coincidental. AWS prices are list prices from the AWS Price List API retrieved on October 13, 2026, and from the public AWS Price List bulk API on October 18, 2026.

Contents

Summary	2
Where each critical stands	5
The ten criticals, step by step	6
C-1: Root user with two active access keys and no MFA	6
C-2: tp-media-originals public-read, and location metadata on 61% of images	7
C-3: tp-exports public with listing on: 104,318 user records	9
C-4: RDS PostgreSQL publicly accessible from 0.0.0.0/0	10
C-5: OpenSearch domain with a public endpoint and an open access policy	11
C-6: Link-preview SSRF reaches IMDSv1 and an instance role with AdministratorAccess	13
C-7: Seven IAM users with administrator access and long-lived keys, two of them departed	14
C-8: Database password and a third-party API key in the repository and in launch template user data	15
C-9: No CloudTrail trail	16
C-10: A forgotten g5.12xlarge with SSH open and a Jupyter server without a token	17
The other findings closed since week one	18
Workload guardrails	19
How each one works, and how it was shown to refuse	21
What the guardrails do not do	22
What the organization adds from the join	22
Exceptions, and how they are granted	23
What comes next	24

Summary

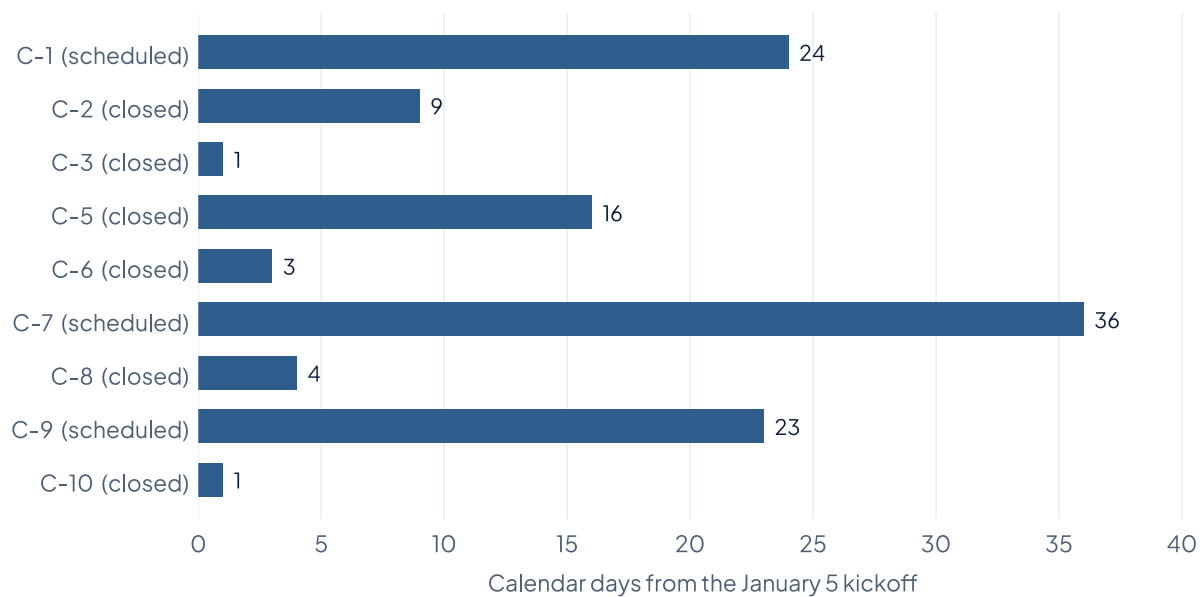
At the end of week 3 (January 23, 2026), **6 of the 10 critical findings are closed**: C-2, C-3, C-5, C-6, C-8, and C-10. One, C-4, is accepted in writing by Dev Raman until the March 22 cutover, with the reason and the controls around it set out in [C-4: RDS PostgreSQL publicly accessible from 0.0.0.0/0](#). The other 3 (C-1, C-7, and C-9) are closed in part: the first step of each is done, and the last step needs an AWS Organization, so it belongs to the Foundation and lands between January 28 and February 10. Across all severities, 23 findings are closed since the January 5 kickoff.

Tidepost’s engineers made every change in this report. The Hardening specified each one, reviewed it, and verified it afterward; where the table says "Tidepost on notice," the fix was made within about a day of our telling the team, before the Hardening began. Each critical below gives what was found, the dated steps, the control as applied, how it was proved, and whether it can be undone. The gate agreed at the January 16 readout is decided by the February 12 re-scan: a critical counts when it is closed and verified there, or accepted in writing by a named Tidepost signer with a closing date.

The second half of the report is the 9 **workload guardrails** that keep these findings closed ([Workload guardrails](#)). They live inside the account, so they work today. The organization’s policy set, which the Foundation attaches to Prod on February 5, backs the ones it can, and [What the organization adds from the join](#) says which.

Week 3 in numbers. The re-scan on February 12 decides which findings count as closed for the gate.

Week 3 at a glance	Result
Critical findings agreed at the readout	10
Closed by January 23	6
Accepted in writing	1
Closed in part, finished by the Foundation	3
All findings closed since kickoff, any severity	23
High findings closed	6
Workload guardrails in place	9
Organization policies the Foundation attaches to Prod on Feb 5	11



Days from kickoff to the closing step of each critical. Bars marked "scheduled" are the Foundation's steps still ahead.
C-4 is not on the chart: its date is the cutover, not a fix.

Where each critical stands

The ten criticals agreed at the readout. For a finding closed in part, the date is when its last step is scheduled.

ID	Critical finding	Owner	Status on Jan 23	Date	Closed by
C-1	Root user with two active access keys and no MFA	Kofi Brandvold	Closed in part	closes Jan 29	Foundation
C-2	tp-media-originals public-read, and location metadata on 61% of images	Amara Delacroix	Closed	Jan 14	Hardening
C-3	tp-exports public with listing on: 104,318 user records	Lior Haddad	Closed	Jan 6	Tidepost on notice
C-4	RDS PostgreSQL publicly accessible from 0.0.0.0/0	Kofi Brandvold	Accepted in writing	to Mar 22	the Well-Architected work
C-5	OpenSearch domain with a public endpoint and an open access policy	Marcus Oyelaran	Closed	Jan 21	Hardening
C-6	Link-preview SSRF reaches IMDSv1 and an instance role with AdministratorAccess	Marcus Oyelaran	Closed	Jan 8	Hardening
C-7	Seven IAM users with administrator access and long-lived keys, two of them departed	Sasha Ilyin	Closed in part	closes Feb 10	Foundation
C-8	Database password and a third-party API key in the repository and in launch template user data	Marcus Oyelaran	Closed	Jan 9	Hardening
C-9	No CloudTrail trail	Sasha Ilyin	Closed in part	closes Jan 28	Foundation
C-10	A forgotten g5.12xlarge with SSH open and a Jupyter server without a token	Yuki Sorensen	Closed	Jan 6	Tidepost on notice

The ten criticals, step by step

Every critical has the same parts. **How it was found** is the evidence behind the rating, gathered read-only. **The steps** are dated and name who took each one. **The control** is what was applied, shown as the command or policy that went in. **How it was proved** is what we ran afterward to confirm the finding no longer reproduces. **Undoing it** says whether the change can be reversed and what to do instead. For the three criticals still in progress, the last part is written as how the remaining step will be proved.

C-1: Root user with two active access keys and no MFA

Owner: Kofi Brandvold, Head of Platform. Resource: Root user of tp-prod. In progress: the last step is scheduled for January 29, with the Foundation.

Why it is critical. Root can do anything the account can, and in a standalone account nothing can limit it. Two long-lived root keys, one in a CI secret and one in a deploy script on a laptop, meant one leaked secret or one lost laptop was a full takeover with no second factor in the way.

How it was found. iam get-account-summary (AccountAccessKeysPresent 2, AccountMFAEnabled 0) and the credential report, read-only through the SecurityAudit and ViewOnlyAccess managed policies; one key sits in a GitHub Actions secret and one in a deploy script on the CTO's laptop.

The steps.

C-1 in the order the steps were or will be taken.

Date	By	Step	Status
Jan 6	Tidepost on notice	Both root access keys deleted after the pipeline and the script moved to roles; two FIDO2 keys registered	Done
Jan 29	Foundation	Root credentials removed from the member account under centralized root access after the January 28 join; root tasks run through sts:AssumeRoot	Scheduled

Why the last step is the Foundation's. Centralized root access removes root credentials from member accounts, which only an organization can do.

The control. Root keys deleted, then root credentials removed under centralized root access.

BASH

```
# January 6: signed in to the console as root with a new FIDO2 key, Security credentials,
delete both access keys.
# After the January 28 join, from tp-management (the Foundation's step; the organization must
have root credentials management on):
aws sts assume-root --target-principal <tp-prod-account-id> \
  --task-policy-arn arn:aws:iam::aws:policy/root-task/IAMDeleteRootUserCredentials --
duration-seconds 900
# then, with the credentials it returns: delete the root password, signing certificates, and
MFA devices
```

How it is proved, and how the last step will be.

- `aws iam get-account-summary` shows `AccountAccessKeysPresent 0` and `AccountMFAEnabled 1` from January 6.
- The pipeline and the deploy script each ran once with their new role before the keys were deleted, so nothing broke.
- After January 29 the credential report's `<root_account>` row should show no password, no access keys, and no MFA device: there is nothing left to sign in with. A root task then needs a short session from tp-management, and the session appears in the organization trail.

Undoing it. Deleted keys cannot be restored and nothing needs them. If a task really needs root, an administrator in tp-management starts a 15-minute root session with `sts:AssumeRoot` and the narrowest task policy; the start of that session is itself an alarm.

C-2: tp-media-originals public-read, and location metadata on 61% of images

Owner: Amara Delacroix, Head of Trust and Safety. Resource: S3 bucket tp-media-originals. Closed January 14, by Hardening.

Why it is critical. The bucket held every uploaded image at full resolution and anyone on the internet could read it. 61% of the images carried the GPS coordinates of where the photo was taken, so a public bucket was also a map to where contributors live and work.

How it was found. `s3 get-bucket-policy-status` (IsPublic true) and an unauthenticated HEAD request on 20 sampled keys; EXIF GPS tags read on a random sample of 1,000 image objects, 610 of them carrying coordinates; nothing was copied out of the bucket.

The steps.

C-2 in the order the steps were or will be taken.

Date	By	Step	Status
Jan 6	Tidepost on notice	Block Public Access turned on and CloudFront origin access control put in front of the bucket; originals answer only a short-lived signed URL from the API, and every other reader gets renditions	Done
Jan 14	Hardening	Location metadata stripped from 9,800,000 image originals by an S3 Batch Operations job that ran January 7 to 14	Done

The control. Block Public Access on the bucket, then reads only through CloudFront.

BASH

```
aws s3api put-public-access-block --bucket tp-media-originals \
  --public-access-block-configuration
BlockPublicAcls=true,IgnorePublicAcls=true,BlockPublicPolicy=true,RestrictPublicBuckets=true
# bucket policy: only the CloudFront service principal, only for this distribution (origin
access control)
aws s3api put-bucket-policy --bucket tp-media-originals --policy file://oac-policy.json
```

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [{
    "Sid": "AllowCloudFrontOriginAccessControl",
    "Effect": "Allow",
    "Principal": {"Service": "cloudfront.amazonaws.com"},
    "Action": "s3:GetObject",
    "Resource": "arn:aws:s3:::tp-media-originals/*",
    "Condition": {"StringEquals": {"AWS:SourceArn": "arn:aws:cloudfront::<tp-prod-account-id>:distribution/<distribution-id>"}}
  }]
}
```

How it was proved.

- get-public-access-block shows all four settings true and get-bucket-policy-status shows IsPublic false.
- The sampled keys that answered an unauthenticated HEAD request now return 403 directly, 403 through the distribution without a signature, and 200 only on a signed URL the API issued. Public readers get renditions; an original goes only to its poster and to the moderation tools, on a URL that expires in minutes.
- The S3 Batch Operations job rewrote 9,800,000 image originals without their GPS tags; its completion report lists every object as succeeded, and a fresh random sample of the same size as the one that found them holds no coordinates.

- The bucket is not versioned, so the rewrite replaced each object in place and no earlier copy keeps the coordinates.

What it changed for users. App versions older than 4.2, about 7% of daily sessions, read the originals bucket on its S3 address, about 30 TB a month of internet egress. From January 6 they saw images and video that would not load (S3 answered 403). Bea Okafor turned on the required-update prompt for those versions the same afternoon, so they showed the prompt instead of broken media, and the current version reads through CloudFront. By January 13 under 1% of sessions were on an old version. Nobody reopened the bucket to keep them working.

Undoing it. The strip cannot be undone and nobody wants it to be. Reopening the bucket would republish every original. If the CloudFront path ever fails, the fix is the distribution's AWS:SourceArn condition, never Block Public Access.

C-3: tp-exports public with listing on: 104,318 user records

Owner: Lior Haddad, Head of Data. Resource: S3 bucket tp-exports. Closed January 6, by Tidepost on notice.

Why it is critical. The exports bucket held a CSV of 104,318 user records and anyone could list the bucket and download the file. The bucket kept no access logs, so nobody can say whether anyone did.

How it was found. Unauthenticated list-objects-v2 returned the key list; the file's line count (104,318 data rows) came from a streamed read and no record was kept or printed; the bucket has no server access logging.

The steps.

C-3 in the order the steps were or will be taken.

Date	By	Step	Status
Jan 6	Tidepost on notice	Closed within the hour; the CSV deleted after counsel copied it. With no access logs, exposure could not be ruled out; counsel was told, and the decision on notification is Tidepost's (Vantalect records only that counsel was told)	Done

The control. Block Public Access on the exports bucket.

BASH

```
aws s3api put-public-access-block --bucket tp-exports \
  --public-access-block-configuration
BlockPublicAcls=true,IgnorePublicAcls=true,BlockPublicPolicy=true,RestrictPublicBuckets=true
```

How it was proved.

- An unauthenticated list-objects-v2 now returns AccessDenied, and get-bucket-policy-status shows IsPublic false.
- head-object on the CSV returns 404: counsel copied the file first, then it was deleted.
- Exposure cannot be ruled out, because the bucket had no server access logging (M-7 records why logging stays off on this class of bucket and what covers the two that hold user data). Counsel was told on the day; whether to notify anyone is Tidepost’s decision and these reports record only that counsel knows.

Undoing it. Do not undo this. An export goes to a named recipient by a presigned URL that expires, never by opening the bucket.

C-4: RDS PostgreSQL publicly accessible from 0.0.0.0/0

Owner: Kofi Brandvold, Head of Platform. Resource: RDS for PostgreSQL 13 production writer. Accepted in writing on January 16, until March 22.

Why it is critical. A production database reachable from any address on the internet is one weak or reused password away from the user table. It cannot simply be made private yet: the BI vendor’s dashboards reach it over the internet today, and the private path the vendor will use (PrivateLink to the new instance) does not exist until the cutover.

How it was found. rds describe-db-instances (PubliclyAccessible true) and the security group (5432 from 0.0.0.0/0), read-only through the SecurityAudit and ViewOnlyAccess managed policies; a TCP handshake from outside the account reached the listener and no login was attempted.

The steps.

C-4 in the order the steps were or will be taken.

Date	By	Step	Status
Jan 7	Hardening	Security group restricted to the BI vendor's four published ranges and TLS required; the staging copy's security group restricted the same day	Done

The control. The security group admits the vendor’s four ranges; TLS is required.

BASH

```
aws ec2 revoke-security-group-ingress --group-id <db-sg-id> --protocol tcp --port 5432 --cidr 0.0.0.0/0
aws ec2 authorize-security-group-ingress --group-id <db-sg-id> \
  --ip-permissions 'IpProtocol=tcp,FromPort=5432,ToPort=5432,IpRanges=[{CidrIp=<vendor-range-1>},{CidrIp=<vendor-range-2>},{CidrIp=<vendor-range-3>},{CidrIp=<vendor-range-4>}]'
aws rds modify-db-parameter-group --db-parameter-group-name <pg13-prod-params> \
  --parameters 'ParameterName=rds.force_ssl,ParameterValue=1,ApplyMethod=immediate'
```

How it was proved.

- describe-security-groups lists exactly four source ranges on port 5432 and nothing else; the staging copy's group the same.
- A connection with sslmode=disable from a vendor range is refused (rds.force_ssl is 1); a connection from outside the four ranges never completes its TCP handshake.
- PubliclyAccessible is still true, which is why RDS.2 keeps failing on one instance and why this is an acceptance, not a closure.

Undoing it. Nothing to undo. Widening the group is what guardrail G-6 exists to catch, and the acceptance names the day it ends.

Written acceptance of C-4

Accepted by: Dev Raman, Co-founder and Chief Technology Officer, on January 16.

Ends: March 22, when the new encrypted instance launches private and the BI vendor moves to PrivateLink. It closes through the Well-Architected work; until then the finding is open, not closed.

Reason, in the signer's words as recorded: PubliclyAccessible stays true until the new encrypted private instance replaces this one at the March 22 cutover; until then the security group admits only the BI vendor's four published ranges and TLS is required.

What keeps the exposure from growing: the security group admits four ranges and nothing else, TLS is required, and guardrail G-6 posts to the platform channel on any change to the group. Security Hub's RDS.2 will keep failing on this one instance, by decision, until the cutover.

C-5: OpenSearch domain with a public endpoint and an open access policy

Owner: Marcus Oyelaran, Backend Lead. Resource: OpenSearch domain tp-search (2.13). Closed January 21, by Hardening.

Why it is critical. The search domain answered anyone on the internet, and its access policy allowed every action to every principal. The index of posts and accounts could be read, changed, or deleted without credentials.

How it was found. `opensearch describe-domain` (no `VPCOptions`; access policy `Principal *` with `es:*`), read-only through the `SecurityAudit` and `ViewOnlyAccess` managed policies; an unauthenticated `GET` / returned the cluster banner and no index was read.

The steps.

C-5 in the order the steps were or will be taken.

Date	By	Step	Status
Jan 21	Hardening	Domain moved into the VPC with fine-grained access control and a restrictive access policy	Done

The control. A new domain in the VPC with fine-grained access control, restored from a snapshot.

BASH

```
# a domain cannot change from public to VPC access in place, so this is a new domain (named tp-search-vpc while both exist) and a snapshot restore
aws opensearch create-domain --domain-name tp-search-vpc --engine-version OpenSearch_2.13 \
  --vpc-options SubnetIds=<subnet-a>,<subnet-b>,SecurityGroupIds=<search-sg-id> \
  --advanced-security-options
Enabled=true,InternalUserDatabaseEnabled=false,MasterUserOptions={MasterUserARN=<search-admin-role-arn>} \
  --encryption-at-rest-options Enabled=true --node-to-node-encryption-options Enabled=true \
  --domain-endpoint-options EnforceHTTPS=true,TLSSecurityPolicy=Policy-Min-TLS-1-2-2019-07
```

How it was proved.

- `describe-domain` shows a `VPCOptions` block, no public endpoint, and `AdvancedSecurityOptions.Enabled true`.
- From outside the VPC the endpoint does not resolve to a public address; from an application subnet an unauthenticated request returns 401.
- The application ran against the new domain for a full business day before the public domain was deleted.

Undoing it. There is no way back to a public endpoint and none is wanted. A domain cannot change between public and VPC access in place, which is why this was a new domain and a restore. If the application cannot reach it, check the domain's security group and the application role's mapping in fine-grained access control.

C-6: Link-preview SSRF reaches IMDSv1 and an instance role with AdministratorAccess

Owner: Marcus Oyelaran, Backend Lead. Resource: Link-preview service on the web tier. Closed January 8, by Hardening.

Why it is critical. The link-preview service fetched any URL a user typed and followed redirects. A URL aimed at the instance metadata address returned the instance's temporary credentials, and because the instance role carried AdministratorAccess, one request could reach the whole account.

How it was found. Static review of the link-preview handler (no URL allowlist, follows redirects) and of the instance profile (AdministratorAccess attached); IMDS HttpTokens optional on the tier. No exploit was run.

The steps.

C-6 in the order the steps were or will be taken.

Date	By	Step	Status
Jan 8	Hardening	Allowlisted egress proxy and URL validation added; IMDSv2 required on the tier; the role cut to 14 actions	Done

The control. IMDSv2 required on the tier.

BASH

```
aws ec2 modify-instance-metadata-options --instance-id <instance-id> \
  --http-tokens required --http-put-response-hop-limit 1 --http-endpoint enabled
# and the same in a new launch template version, rolled with an instance refresh
```

How it was proved.

- From an instance on the tier, a plain curl `http://169.254.169.254/latest/meta-data/` returns 401: IMDSv2 needs a token from a PUT request, which a request forwarded by the application cannot make.
- The handler refuses the metadata address, private ranges, and redirects to them (a unit test in the repository for each); egress from the service's security group reaches the allowlisted proxy only.
- `get-role-policy` lists the 14 actions the service uses and nothing else; Access Analyzer's policy generation against the trail found no other action in use.

Undoing it. Nothing to undo. If a legitimate destination is blocked, it goes on the proxy's allowlist by pull request.

C-7: Seven IAM users with administrator access and long-lived keys, two of them departed

Owner: Sasha Ilyin, Head of Security and Compliance. Resource: Seven IAM users in tp-prod. In progress: the last step is scheduled for February 10, with the Foundation.

Why it is critical. Seven people held administrator access through IAM users whose keys never expire, and two of the seven had left the company: two valid keys the company could not account for.

How it was found. The credential report and list-attached-user-policies, read-only through the SecurityAudit and ViewOnlyAccess managed policies; the two departed users' keys were last used between 60 and 140 days earlier (HR roster compared by Sasha Ilyin).

The steps.

C-7 in the order the steps were or will be taken.

Date	By	Step	Status
Jan 6	Tidepost on notice	The two departed users' passwords and keys deleted, and the users with them	Done
Jan 26	Hardening	The 14 current staff provisioned into IAM Identity Center over SCIM, by group (Identity Center connected January 23)	Scheduled
Jan 28	Hardening	Permission sets assigned to tp-prod the day it joined the organization: the 5 remaining administrators and the 9 other people with console access sign in through Identity Center	Scheduled
Feb 10	Foundation	The five IAM users deleted after the policy set that denies new IAM users and keys reached Prod on February 5	Scheduled

Why the last step is the Foundation's. The permission sets live in IAM Identity Center in tp-management, and the policy set that denies new IAM users and keys is an organization policy.

The control. The departed users removed first (January 6), the rest after Identity Center.

BASH

```
# for each departed user, in this order (delete-user fails while any policy, group, MFA device,
# or certificate remains attached)
aws iam delete-access-key --user-name <user> --access-key-id <key-id>
aws iam delete-login-profile --user-name <user>
aws iam delete-user --user-name <user>
```

How it is proved, and how the last step will be.

- Proved on January 6 for the two departed users: get-user returns NoSuchEntity for both.
- Still ahead: each person’s first Identity Center sign-in appears in the trail before that person’s IAM user is deleted, and after February 10 aws iam list-users in tp-prod returns one user, the video analytics vendor’s.

Undoing it. Deleted users are not restored. A person who lost access gets a permission set assignment, never a new IAM user; the policy set the Foundation attaches to Prod on February 5 denies creating one.

C-8: Database password and a third-party API key in the repository and in launch template user data

Owner: Marcus Oyelaran, Backend Lead. Resource: The backend repository and launch template versions. Closed January 9, by Hardening.

Why it is critical. A database password and a third-party API key sat in the repository and in launch template user data. Anyone with read access to either, a contractor, a laptop, a leaked clone, held production credentials.

How it was found. A secret-pattern scan of the default branch and 90 days of history, and describe-launch-template-versions user data; matches were counted and no value was printed.

The steps.

C-8 in the order the steps were or will be taken.

Date	By	Step	Status
Jan 9	Hardening	Both secrets rotated into Secrets Manager and read at runtime, and user data cleaned; the history rewrite was declined because rotation closes the exposure	Done

The control. Rotated into Secrets Manager and read at runtime; user data cleaned.

```
BASH
aws secretsmanager create-secret --name tidepost/prod/<secret-name> --secret-string file://<new-value>.json # a newly generated value
aws ec2 create-launch-template-version --launch-template-id <lt-id> --source-version <n> \
  --launch-template-data file://user-data-without-secrets.json
aws ec2 modify-launch-template --launch-template-id <lt-id> --default-version <new-version>
```

How it was proved.

- The old database password and the old API key are both rejected: the database refuses the password and the vendor reports the key revoked. That is the proof that matters, because the old values remain in history.
- The secret-pattern scan of the default branch finds no live value, and describe-launch-template-versions shows no secret in the default version's user data; earlier versions with secrets were deleted.
- The history rewrite was declined on January 9 (D-2): rotation closes the exposure, and a rewrite would break every clone and open pull request. Guardrail G-3 stops the next one at the push.

Undoing it. Rotation is not reversible. If the application cannot read the secret, check its role's secretsmanager:GetSecretValue permission and the Secrets Manager endpoint, not the old value.

C-9: No CloudTrail trail

Owner: Sasha Ilyin, Head of Security and Compliance. Resource: CloudTrail in tp-prod. In progress: the last step is scheduled for January 28, with the Foundation.

Why it is critical. With no trail, nobody could say who did what in the account beyond the last 90 days of management events, and nothing about data events. After the public export and the forgotten GPU instance, the first question was whether anyone else had been inside, and there was no record to answer it.

How it was found. cloudtrail describe-trails returned none and lookup-events reaches only 90 days of management events, read-only through the SecurityAudit and ViewOnlyAccess managed policies.

The steps.

C-9 in the order the steps were or will be taken.

Date	By	Step	Status
Jan 6	Hardening	An account trail created, multi-Region, writing to a locked bucket	Done
Jan 28	Foundation	Replaced by the organization trail into tp-log-archive when the account joined	Scheduled

Why the last step is the Foundation's. The organization trail writes to tp-log-archive, which only an organization provides.

The control. An account trail first, the organization trail at the join.

BASH

```
aws cloudtrail create-trail --name tp-prod-trail --s3-bucket-name <trail-bucket> \
  --is-multi-region-trail --include-global-service-events --enable-log-file-validation
aws cloudtrail start-logging --name tp-prod-trail
```

How it is proved, and how the last step will be.

- get-trail-status shows IsLogging true and a LatestDeliveryTime within minutes; validate-logs reports every digest file valid for the window.
- Still ahead on January 28: the organization trail, already logging into tp-log-archive since January 21, starts receiving tp-prod’s events at the join; the account trail is stopped only after the first organization trail files for tp-prod arrive.

Undoing it. Do not stop a trail. Once the organization’s policy set reaches Prod on February 5, stopping it is denied to everyone but the break-glass role.

C-10: A forgotten g5.12xlarge with SSH open and a Jupyter server without a token

Owner: Yuki Sorensen, Head of Machine Learning. Resource: EC2 instance (g5.12xlarge) in tp-prod. Closed January 6, by Tidepost on notice.

Why it is critical. A GPU instance had run since September 2025 with SSH open to the world and a Jupyter server that needed no token: a free GPU and a shell for whoever found it. It cost \$4,141 a month.

How it was found. ec2 describe-instances (running since September 2025; security group open to 0.0.0.0/0 on 22 and 8888), read-only through the SecurityAudit and ViewOnlyAccess managed policies; a status-code request to the Jupyter API returned 200 with no token.

The steps.

C-10 in the order the steps were or will be taken.

Date	By	Step	Status
Jan 6	Tidepost on notice	Snapshotted for the record and terminated. No indicators in its shell history, processes, or packages; with no flow logs, compromise could not be ruled out, and counsel was told. The cost engagement does not count this instance	Done

The control. Snapshotted for the record, then terminated.

BASH

```
aws ec2 create-snapshots --instance-specification InstanceId=<instance-id> --description 'C-10
evidence, counsel hold' \
  --tag-specifications 'ResourceType=snapshot,Tags=[{Key=hold,Value=counsel}]'
aws ec2 terminate-instances --instance-ids <instance-id>
```

How it was proved.

- describe-instances shows the instance terminated; the security group that opened 22 and 8888 to the internet is deleted.
- The snapshot exists with the counsel-hold tag. Shell history, processes, and packages on the volume showed no indicator of compromise; with no flow logs for the period, that is absence of evidence, which is why counsel was told.
- The \$4,141 a month leaves the bill as a security change. The cost engagement does not count it.

Undoing it. Terminated. The snapshot is the record, and counsel decides how long it is kept.

The other findings closed since week one

The 10 criticals are the gate, but the other severities are moving too: 6 high, 6 medium, and 5 low findings are closed as of January 23. The high findings are listed here because several of them are the reason a critical stays closed.

High findings closed by January 23. The full register, with every step and its owner, is in `security-2-audit` and in the findings register file.

ID	High finding	Owner	Closed
H-2	IMDSv1 on every instance	Kofi Brandvold	Jan 22
H-3	No MFA on any of the 16 IAM users with a console password	Sasha Ilyin	Jan 13
H-4	SSH open to the internet on 18 instances	Kofi Brandvold	Jan 14
H-5	Account-level S3 Block Public Access off	Kofi Brandvold	Jan 6
H-11	Two customer-managed KMS keys whose policies allow any identity in the account	Sasha Ilyin	Jan 23
H-14	CI on 400-day-old access keys instead of OIDC federation	Kofi Brandvold	Jan 20

Workload guardrails

A fix stays fixed only if something refuses the next person who undoes it by accident. Tidepost is a single AWS account until the join on January 28, so no organization policy can do that yet. These 9 guardrails do it inside the account: some refuse the act, one undoes it within seconds, and two tell a person. Each one names the findings it protects, the person who runs it, and where its evidence lives.

The workload guardrails. Preventive ones refuse the act; the corrective one reverses it within seconds; detective ones post to the platform channel.

ID	Guardrail	What it stops	Kind	Keeps closed	In place	Runs it
G-1	Account-level S3 Block Public Access	A bucket becoming public, whoever creates it and however it is configured	Preventive	C-2, C-3, H-5	Jan 6	Kofi Brandvold
G-2	Media originals reachable only through CloudFront	Direct reads of the originals bucket from the internet	Preventive	C-2	Jan 6	Kofi Brandvold
G-3	Secret scanning with push protection	A password or API key reaching the repository again	Preventive	C-8	Jan 9	Marcus Oyelaran
G-4	Open SSH and RDP closed automatically	A security group rule that opens port 22 or 3389 to 0.0.0.0/0 staying open	Corrective	H-4, C-10	Jan 14	Kofi Brandvold
G-5	Alarms on root, unauthorized calls, IAM changes, and the trail	A change to identity or logging that nobody sees	Detective	C-1, C-7, C-9	Jan 20	Kofi Brandvold
G-6	Alarm on any change to the database's security group	The accepted database exposure (C-4) widening without a decision	Detective	C-4	Jan 20	Kofi Brandvold
G-7	EBS encryption by default	A new volume or snapshot copy created unencrypted	Preventive	H-1	Jan 21	Kofi Brandvold
G-8	IMDSv2 required by default	A new instance launching with IMDSv1 enabled	Preventive	C-6, H-2	Jan 22	Kofi Brandvold
G-9	Link-preview egress through an allowlisted proxy	The service fetching an address inside the VPC or the metadata endpoint	Preventive	C-6	Jan 8	Marcus Oyelaran

How each one works, and how it was shown to refuse

A guardrail we have not seen refuse is a guess, so each was tested by trying the thing it forbids.

The mechanism, the evidence location, and the test for each workload guardrail.

ID	The control	Evidence is in	How it was shown
G-1	S3 Block Public Access on the account, all four settings on	The account's Block Public Access settings, and Config history for the S3 public-access rules from January 29	On a scratch bucket in tp-prod, put-bucket-acl with public-read and a bucket policy naming Principal: * were both refused with AccessDenied.
G-2	A bucket policy that admits only the CloudFront service principal for one distribution (origin access control)	The bucket policy version history and the distribution's origin configuration	A direct GET of an original from outside the account returned 403; the same key through the distribution returned 200.
G-3	GitHub secret scanning with push protection on every repository, and a pre-commit hook that runs the same patterns	The secret scanning alerts and the push-protection bypass log in each repository's settings	A branch holding a test token that matches a provider pattern was refused at the push, naming the file and line; the pre-commit hook refused the same string locally.
G-4	An EventBridge rule on AuthorizeSecurityGroupIngress that invokes a function to revoke the rule and post to the platform channel	The rule's invocation metrics and the function's log group	Port 22 opened to 0.0.0.0/0 on a scratch group; the rule was revoked within seconds and the platform channel carried the post.
G-5	CloudWatch metric filters and alarms on the trail's log group (the CIS monitoring set), notifying the platform channel	CloudWatch alarm history	Each alarm was set off once on purpose: a scratch IAM policy created and deleted, a denied call from a read-only role, an UpdateTrail on a scratch trail, and one planned root console sign-in approved by Dev Raman. All four reached the platform channel.
G-6	An EventBridge rule on changes to the one security group, notifying the platform channel	The rule's event history and the security group's change history	A scratch rule added to and removed from the database's group during a test window: the post arrived both times.

ID	The control	Evidence is in	How it was shown
G-7	EBS encryption by default on in both Regions	The account attribute in each Region and Config history from January 29	A volume created with no encryption setting came back with Encrypted true in both Regions.
G-8	The account's instance metadata defaults set to HttpTokens required in each Region, and every launch template pinned to it	The account's instance metadata defaults and launch template versions	An instance launched from a template that says nothing about metadata options came up with HttpTokens required.
G-9	The service's security group admits egress to the proxy only, the proxy validates the destination, and the role is limited to 14 actions	The proxy allowlist in the repository and the role's policy versions	A request through the service for the metadata address, and for a public address that redirects to it, was refused; a request to an ordinary site passed through the proxy.

What the guardrails do not do

Three limits to know before relying on them. First, the detective and corrective guardrails (G-4, G-5, and G-6) act after the event, not before it: an open SSH rule exists for the seconds before G-4 revokes it, and G-5 and G-6 only tell people. Second, G-8 sets a default; a launch that asks for IMDSv1 explicitly can still get it until the organization policy that denies that call reaches Prod. Third, none of these can stop a person with administrator access from changing them. That is the Foundation's job, and it is why the next section matters.

What the organization adds from the join

From January 28 the account is a member of the organization, and the Foundation's policy set applies to it OU by OU: first in Policy Staging, where its policies were attached to the empty OU on January 23 and tested against tp-policy-test from its creation on January 29, then to Prod on February 5 after Sandbox and NonProd have run it without a denial. Two of these guardrails gain a backstop that even an administrator in the account cannot remove.

The organization policies that back the workload guardrails. The full policy set, with its tests, is the Foundation's report `foundation-6-guardrails-vending`.

Guardrail	Backed by	What it adds	Reaches Prod
G-5 Alarms on root, unauthorized calls, IAM changes, and the trail	tp-scp-baseline (SCP)	Denies stopping or deleting CloudTrail, Config, GuardDuty, and Security Hub, so the alarm on the trail has less to catch.	Feb 5
G-5 Alarms on root, unauthorized calls, IAM changes, and the trail	tp-scp-no-member-root (SCP)	Denies any action by a member account's root user, so a root sign-in in tp-prod can do nothing even if it happens.	Feb 5
G-8 IMDSv2 required by default	tp-decl-ec2-imds (declarative (EC2))	Makes IMDSv2 the default for every new launch, and no one in a member account can change the default.	Feb 5
G-8 IMDSv2 required by default	tp-scp-baseline (SCP)	Denies launching or changing an instance with IMDSv1 allowed, which closes the one gap the default leaves: an explicit request for IMDSv1.	Feb 5

Exceptions, and how they are granted

A guardrail with no way around it gets worked around, and one with an easy way around it is decoration. Tidepost has two kinds of exception today, each with its own approver and its own clock.

The two kinds of exception and who may grant each.

Kind	Example	Who approves	Written where	Ends
Excluding something from a guardrail	One named security group left out of G-4 for a day	Kofi Brandvold, with Sasha Ilyin told	A pull request or a ticket, before the change	The same day
Accepting a finding's risk	C-4, H-1, H-12, M-7, and L-4: signed by Dev Raman for C-4 and H-1, and by Sasha Ilyin for H-12, M-7, and L-4	The owner of the risk	The acceptance log in the findings register, signed and dated	The date or review date written in it; never open-ended

Four rules apply to both: it is **written before it takes effect** and the reviewer is not the requester; it is **time-boxed**, so no date means no exception; it is **narrow**, a named resource and never a wildcard; and it is **counted**, because the monthly review reads the list and removes any exception nobody can explain.

What comes next

Week 4 moves people off long-lived keys and onto IAM Identity Center, and CI onto OIDC federation (security-5-identity). Week 5 turns on and routes detection (security-6-detection-response). The three criticals still in progress close on these dates:

The remaining critical steps, in the order the calendar reaches them.

Date	Critical	Step	By
Jan 26	C-7	The 14 current staff provisioned into IAM Identity Center over SCIM, by group (Identity Center connected January 23)	Hardening
Jan 28	C-7	Permission sets assigned to tp-prod the day it joined the organization: the 5 remaining administrators and the 9 other people with console access sign in through Identity Center	Hardening
Jan 28	C-9	Replaced by the organization trail into tp-log-archive when the account joined	Foundation
Jan 29	C-1	Root credentials removed from the member account under centralized root access after the January 28 join; root tasks run through sts:AssumeRoot	Foundation
Feb 10	C-7	The five IAM users deleted after the policy set that denies new IAM users and keys reached Prod on February 5	Foundation

© 2026 Vantalect, LLC. Sample deliverable; Tidepost, Inc. is fictional.

AWS, Amazon Web Services, AWS Graviton, AWS Inferentia, and the AWS service names are trademarks of Amazon.com, Inc. or its affiliates. PostgreSQL is a registered trademark of the PostgreSQL Community Association of Canada. Kubernetes is a registered trademark of The Linux Foundation. Redis is a registered trademark of Redis Ltd. Valkey is a trademark of the Linux Foundation. OpenSearch is a registered trademark of Amazon.com, Inc. SOC 2 is a registered trademark of the American Institute of Certified Public Accountants. CIS Benchmarks is a trademark of the Center for Internet Security, Inc. GitHub is a trademark of GitHub, Inc. Jupyter is a trademark of LF Charities. Vantalect is not affiliated with or endorsed by any of them. Tidepost, Inc. is fictional.