

AWS SECURITY, COMPLIANCE & GOVERNANCE · SAMPLE DELIVERABLE

Identity: People in Identity Center, Workloads on Roles

Week 4 of 6: Identity · prepared for Tidepost, Inc.

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

January 30, 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
- People: everyone through Identity Center 4
 - The permission sets 4
 - When someone leaves 6
 - The policy that keeps it closed 6
- What had a password or a key, and what has now 7
 - The second factor, in the meantime 7
- CI: GitHub Actions on OIDC 8
- Workloads on roles 9
- Root and break-glass 11
- The one IAM user that stays 11
- What is not finished 11

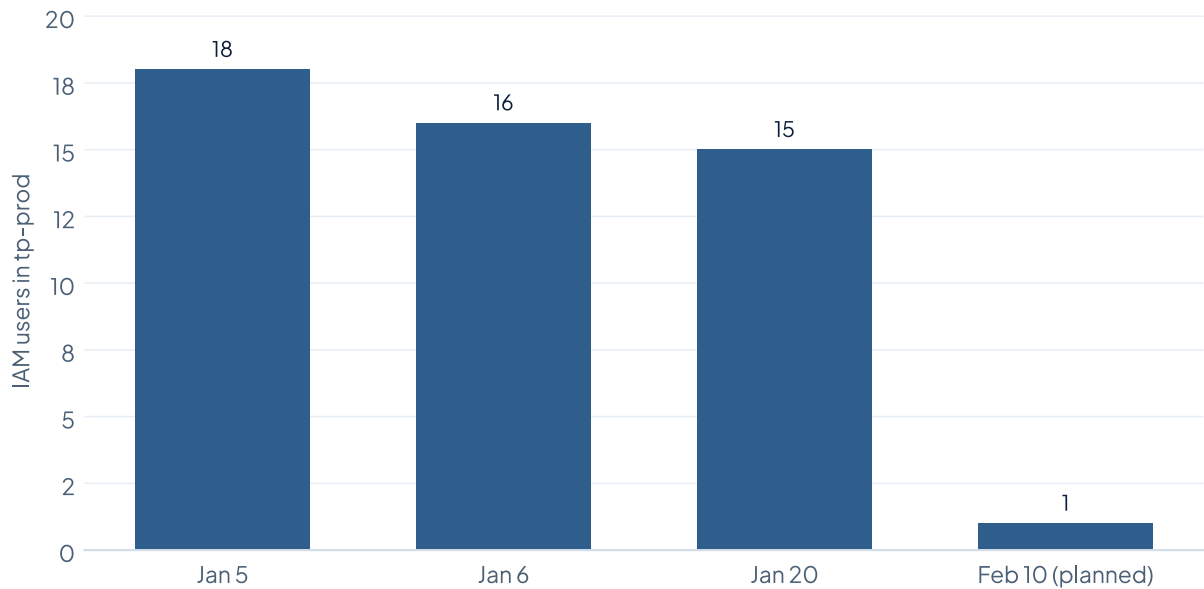
Summary

At the end of week 4 (January 30, 2026), the people who work in AWS have a home that is not an IAM user. The 5 administrators who remained and the 9 other people with console access, 14 in all, were provisioned into **IAM Identity Center** on January 26 and have signed in to tp-prod through it since January 28, the day the account joined the organization, each onto the smallest of 6 permission sets that does the job, with a second factor required at every sign-in, enforced by the identity provider. GitHub Actions has deployed through **OIDC federation**, with no key stored anywhere, since January 20. The root user holds no access keys, and since January 29 it holds no credentials at all. What is left is the IAM users of the 14 people themselves, kept only as a fallback: each is deleted on February 10, after that person’s first Identity Center sign-in appears in the trail, and one user stays, for a vendor that cannot assume a role.

IAM users in tp-prod fall from 18 on the first day to 15 today and 1 on February 10. The credential that matters most, the one for a person who has left, is gone: the 2 departed administrators’ users were deleted on January 6. Workloads are on roles: 8 of the 10 workload credentials in [Workloads on roles](#) are changed, and the remaining 2 are dated in the same table.

Identity in numbers on January 30, 2026.

Week 4 at a glance	Result
IAM users in tp-prod on January 5	18
IAM users in tp-prod on January 30	15
IAM users planned for February 10	1
People on IAM Identity Center	14
Permission sets	6
Root user access keys	0
Long-lived keys held by CI	0
Workload credentials changed, of those in the table	8 of 10



IAM users in tp-prod over time. The last bar is the plan, not a result.

What the sponsor needs to know

Dev Raman: two things are on your calendar. The IAM users of the 14 people are deleted on February 10, once the organization's policy set that denies new IAM users has reached Prod; that policy is the Foundation's step, approved in its change window. And the one IAM user that stays, for the video analytics vendor, needs a written exception that Sasha Ilyin signs on February 11. Neither needs a decision from you today.

People: everyone through Identity Center

On the first day, access was 7 IAM users with AdministratorAccess, 9 other users with console passwords, and no second factor on any of them. Now a person signs in at the company's identity provider, with the second factor it already enforces, and lands in the AWS access portal, which shows only the accounts and roles that person has been given. Nothing is typed into AWS: there is no AWS password to forget, share, or leave behind.

The identity provider connects to Identity Center over SAML for sign-in and SCIM for provisioning, so a person added to a group in the provider appears in AWS, and a person removed disappears from it, without anyone touching AWS. The connection went live in the Foundation's second change window, on January 23 (that window is foundation-4-landing-zone). SCIM provisioned the people by group on January 26, but an organization's Identity Center instance assigns permission sets only to member accounts, so the assignments to tp-prod waited for the account to join on January 28. The first sign-ins through Identity Center are that day.

The permission sets

A permission set is a named bundle of permissions that Identity Center turns into a role in each account it is assigned to. There are 6, and a person gets the smallest one that does the job. Sessions last 8 hours, 4 for PlatformAdministrator, and the identity provider requires a second factor at every sign-in, a security key or an authenticator app. Identity Center's own multi-factor settings do not apply to people who sign in through an external provider, which is why the requirement lives there.

The permission sets, each assigned to a group. Members named here are the leads; the groups hold the rest of each team.

Permission set	Group	Named members	Scope
PlatformAdministrator	aws-platform-admins	Kofi Brandvold	Every account; administrator, MFA required
Developer	aws-developers	Marcus Oyelaran, Bea Okafor	tp-dev and tp-staging with power-user access; read-only in tp-prod
MachineLearning	aws-ml	Yuki Sorensen	tp-sandbox-ml with power-user access; read-only in tp-prod
DataEngineer	aws-data	Lior Haddad	Read and query on the telemetry stores in tp-prod; power-user in tp-dev
SecurityAudit	aws-security-audit	Sasha Ilyin, Amara Delacroix	Read-only everywhere; tp-audit and tp-log-archive
BillingViewer	aws-billing	Gwen Tallis	Billing and Cost Explorer in tp-management

Administrator access is now one permission set, assigned to one group, held by the platform team, instead of seven standing users. Everyone else works in the accounts that match their job and can read, but not change, production. The commands that create a permission set and assign it have this shape:

BASH

```
aws sso-admin create-permission-set --instance-arn <instance-arn> --name PlatformAdministrator
--session-duration PT4H
aws sso-admin attach-managed-policy-to-permission-set --instance-arn <instance-arn> \
  --permission-set-arn <permission-set-arn> --managed-policy-arn arn:aws:iam::aws:policy/
AdministratorAccess
aws sso-admin create-account-assignment --instance-arn <instance-arn> --target-type AWS_ACCOUNT
--target-id <account-id> \
  --permission-set-arn <permission-set-arn> --principal-type GROUP --principal-id <group-id>
```

A developer's day starts with a browser sign-in and ends when the session does:

BASH

```
aws configure sso # once: the start URL, the account, and the permission set
aws sso login --profile tp-dev # opens the browser; the identity provider asks for the second
factor
aws sts get-caller-identity --profile tp-dev
```

When someone leaves

A session already open outlives the person's removal until it expires, which is why sessions are short and why SR-1 in security-8-runbooks-handover has a step for deleting active sessions.

How a leaver loses access, and what covers the gap.

Step	Who	When	What it does
The person is disabled in the identity provider	HR and Sasha Ilyin	The day they leave	Identity Center receives the change automatically (SCIM provisioning); the person cannot start a new AWS session.
Existing sessions run out	Nobody	Within the session duration	A session already open lasts until it expires: at most 8 hours, 4 for PlatformAdministrator. For a suspected misuse, delete the user's active sessions in IAM Identity Center and follow SR-1 for any role session.
The access review confirms it	Sasha Ilyin	Quarterly, and at each leaver	The assignment export is compared with the HR roster; a name on the export that is not on the roster is a finding.

The access review is the check on all of it. Each quarter, Sasha Ilyin exports the assignments and the group memberships and compares them with the HR roster. The first is on March 31, the day before the SOC 2 observation window opens, so the first export is already in the evidence folder when the auditor starts.

BASH

```
aws sso-admin list-account-assignments --instance-arn <instance-arn> --account-id <account-id>
--permission-set-arn <permission-set-arn>
aws identitystore list-group-memberships --identity-store-id <identity-store-id> --group-id
<group-id>
```

The policy that keeps it closed

Moving people is only worth doing if nobody can quietly make a new IAM user. tp-scp-no-iam-users does that for the whole organization. It blocks creating IAM users, console passwords, and access keys, and it has one exemption: the break-glass role; the vendor key rotation role in tp-prod, for the vendor user's access keys only; the Exceptions OU does not carry this policy.

Where tp-scp-no-iam-users is attached, and when. It reaches Prod after Sandbox and NonProd have run it for a day without a denial.

Reaches	Date	Status on January 30
Policy Staging	Jan 23	Attached
Sandbox, Workloads/NonProd	Feb 3	Scheduled
Workloads/Prod, Security, Infrastructure	Feb 5	Scheduled

What had a password or a key, and what has now

This is the inventory of long-lived credentials, with counts from the credential report on the first day and what has changed since.

Long-lived credentials in tp-prod. "At the end" is the plan; the dates are in [What is not finished](#).

Credential	On January 5	On January 30At the end	Finding
Root user access keys	2	0None; root credentials leave tp-prod on Jan 29	C-1
IAM users who had left the company	2	0None	C-7
IAM users with a console password	16	14None after Feb 10	H-3
Access keys unused for 90 days	11	0None	M-8
Access keys held by CI	2	0None	H-14
IAM users of every kind in tp-prod	18	151: the video analytics vendor's	MAF-05

The second factor, in the meantime

Before January 13, no console user had a second factor and the password or key alone was enough. From January 13 until January 28, when people still used their IAM users, a policy the Hardening applied to every console user denies everything except setting up a second factor until one is present:

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AllowManageOwnCredentialsAndMfa",
      "Effect": "Allow",
      "Action": ["iam:ChangePassword", "iam:GetUser", "iam:ListMFADevices",
        "iam:ListVirtualMFADevices", "iam:CreateVirtualMFADevice",
        "iam:EnableMFADevice", "iam:ResyncMFADevice"],
      "Resource": ["arn:aws:iam::*:user/${aws:username}", "arn:aws:iam::*:mfa/${aws:username}"]
    },
    {
      "Sid": "DenyEverythingElseWithoutMfa",
      "Effect": "Deny",
      "NotAction": ["iam:ChangePassword", "iam:GetUser", "iam:ListMFADevices",
        "iam:ListVirtualMFADevices", "iam:CreateVirtualMFADevice",
        "iam:EnableMFADevice", "iam:ResyncMFADevice", "sts:GetSessionToken"],
      "Resource": "*",
      "Condition": {"BoolIfExists": {"aws:MultiFactorAuthPresent": "false"}}
    }
  ]
}
```

BoolIfExists matters: it also denies a request that carries no MFA information at all, which is every request made with a long-lived access key, so a stolen key could no longer do anything but manage its own credentials. The policy went away with the users on February 10.

CI: GitHub Actions on OIDC

The pipeline used one IAM user with 2 access keys, 391 and 412 days old, kept as repository secrets: a credential that never expires, readable by anyone who can edit a workflow. Since January 20 the workflow asks GitHub for a short-lived signed token and exchanges it for a role session of about an hour. Nothing is stored. The 2 keys and the user are deleted.

The exchange needs an OIDC provider in the account, created once, and one role per repository whose trust policy accepts a token only from that repository and branch. Both conditions matter: without the sub condition, any repository on GitHub could ask for the role.

BASH

```
aws iam create-open-id-connect-provider --url https://token.actions.githubusercontent.com --
client-id-list sts.amazonaws.com
```

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [{
    "Effect": "Allow",
    "Principal": {"Federated": "arn:aws:iam::<tp-prod-account-id>:oidc-provider/
token.actions.githubusercontent.com"},
    "Action": "sts:AssumeRoleWithWebIdentity",
    "Condition": {
      "StringEquals": {
        "token.actions.githubusercontent.com:aud": "sts.amazonaws.com",
        "token.actions.githubusercontent.com:sub": "repo:tidepost/tidepost-cdk:ref:refs/heads/
main"
      }
    }
  }]
}
```

The workflow side is four lines, and holds no secret:

YAML

```
permissions:
  id-token: write
  contents: read
steps:
  - uses: aws-actions/configure-aws-credentials@v4
    with:
      role-to-assume: arn:aws:iam::<tp-prod-account-id>:role/tp-deploy-cdk
      aws-region: us-east-1
```

Each deploy role is limited to assuming the CDK bootstrap roles for tp-prod, so a compromised workflow can deploy the stacks it was meant to and nothing else. The role names the branch (main), so a pull request from a fork cannot assume it.

Workloads on roles

Every workload credential in the first scan appears here, with what it was, what it is now, and the date. A row marked "Scheduled" is a change dated after January 30.

Workload credentials by finding. Each owner is the person who answers for the change.

Workload	Before	Now	Finding	Owner	Status
CI and deployment (GitHub Actions)	One IAM user with 2 access keys, 391 and 412 days old, stored as repository secrets	OIDC federation: a short-lived token exchanged for a deploy role, one role per repository	H-14	Kofi Brandvold	Done Jan 20
Link-preview service	An instance role with AdministratorAccess	A role limited to the actions the service uses, with IMDSv2 required	C-6	Marcus Oyelaran	Done Jan 8
Backend application secrets	A database password and a third-party API key in the repository and in user data	Secrets Manager, read at runtime by the service's own role	C-8	Marcus Oyelaran	Done Jan 9
Redis cache	No password and no encryption in transit	An AUTH token from Secrets Manager and TLS	H-7	Marcus Oyelaran	Done Jan 27
Customer-managed encryption keys	Key policies that let any identity in the account use the key through IAM policy alone	Named key administrators and per-service grants	H-11	Sasha Ilyin	Done Jan 23
Vendor cross-account roles	Four roles a vendor could assume without proving which customer it acted for	An external ID on each trust policy, agreed with each vendor	M-14	Kofi Brandvold	Done Jan 22
Unused roles	Two roles that trust an account nobody at Tidepost can identify	Deleted	L-6	Kofi Brandvold	Done Jan 20
Kubernetes access	Every engineer mapped to cluster administrator	EKS access entries, with a role scoped to a namespace	H-9	Kofi Brandvold	Done Jan 30
Lambda configuration	Secrets in the environment variables of nine functions	Values in Secrets Manager, read at start-up	M-3	Marcus Oyelaran	Scheduled Feb 2
Moderation functions	One execution role with s3:* and sagemaker:* shared by every function	One role per function, limited to its own buckets and endpoints	H-13	Amara Delacroix	Scheduled Feb 9

Two of these deserve a sentence each. **Vendor roles** are the textbook case for an external ID: a vendor that serves many customers assumes a role in each, and without an ID agreed per customer, one customer could trick the vendor into acting in another's account. Each vendor role that had none now has one, set with the vendor. **Kubernetes access** was every engineer mapped to cluster administrator through a ConfigMap; it is now an EKS access entry with a role scoped to a namespace, so a developer can change what they own and read the rest.

Root and break-glass

The root user has no access keys. Two FIDO2 keys protected it from January 6, and on January 29 the Foundation removed its credentials from tp-prod under centralized root access, so a root task now needs a short session started from tp-management, and every session is in the trail. The break-glass path, for the day Identity Center itself is unavailable, was created on January 22 in the Foundation's work: a hardware key sealed in the custody of Sasha Ilyin and Noor Castellanos, and an alarm on every use that posts to the platform channel and opens a ticket. That path, its test, and its runbook are the Foundation's, in foundation-8-runbooks-handover.

The one IAM user that stays

One IAM user with an access key in tp-prod, for a video analytics vendor whose system cannot assume a role. Its permissions: read access to one prefix of the renditions bucket. When asked, the vendor has no plan to change that. So the user stays in tp-prod, outside the Exceptions OU, under a written waiver that Sasha Ilyin signs on February 11 (the Foundation's check MAF-05). Its key is rotated every 90 days through its own role, which the no-IAM-users policy exempts for that user's keys and nothing else, so a rotation never needs break glass; Kofi Brandvold owns the rotation, and the waiver is reviewed at each quarterly access review.

What is not finished

Everything dated after January 30, and the full timeline it sits in. The rows in the lower half are plans, each with an owner and a date; the next report that touches them is security-7-rescan.

Identity changes in the order they happened or will happen. 5 of 20 are still ahead, between February 2 and February 11.

Date	Who	What	Ref	Status
Jan 6	Tidepost on notice	Both root access keys deleted and two FIDO2 keys registered on the root user	C-1	Done
Jan 6	Tidepost on notice	The two departed administrators' passwords, keys, and users deleted	C-7	Done
Jan 13	Hardening	A policy condition makes every console user's access depend on MFA	H-3	Done
Jan 13	Hardening	All 11 access keys unused for 90 days deactivated, then deleted after a week	M-8	Done
Jan 20	Hardening	GitHub Actions moves to OIDC federation; the 2 CI keys are deleted	H-14	Done
Jan 20	Hardening	Two unused roles that trust an unknown account deleted	L-6	Done
Jan 22	Hardening	External IDs added to the four vendor roles	M-14	Done
Jan 22	Tidepost	The break-glass user created and its hardware key sealed	MAF-05	Done
Jan 23	Hardening	Both customer-managed key policies rewritten with named administrators	H-11	Done
Jan 23	Foundation	IAM Identity Center connected to the identity provider (SAML sign-in, SCIM provisioning) with the six permission sets	MAF-05	Done
Jan 26	Hardening	The current staff provisioned into Identity Center over SCIM, by group	C-7	Done
Jan 28	Hardening	The people sign in to tp-prod through Identity Center permission sets	C-7	Done
Jan 28	Foundation	tp-prod joins the organization and the permission sets are assigned to it	MAF-02	Done
Jan 29	Foundation	Root credentials removed from tp-prod under centralized root access	C-1	Done
Jan 30	Hardening	Developers move from cluster administrator to namespace-scoped EKS access entries	H-9	Done
Feb 2	Hardening	The secrets in the environment variables of 9 Lambda functions move to Secrets Manager	M-3	Scheduled

Date	Who	What	Ref	Status
Feb 5	Foundation	tp-scp-no-iam-users reaches Prod: no new IAM users, console passwords, or access keys	POL-S3	Scheduled
Feb 9	Hardening	Each moderation function gets its own execution role	H-13	Scheduled
Feb 10	Foundation	The IAM users of the current staff deleted, each after that person's Identity Center sign-in	C-7	Scheduled
Feb 11	Tidepost	The written exception for the video analytics vendor's user signed	MAF-05	Scheduled

© 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.