SANS

CloudSecNext

2025

Summit: Oct 2 - 3 | Training: Oct 4 - 9

Denver, CO & Live Online

sans.org/CloudSecNextSummit

Get The Flock Out of My Cloud:

Using DuckDB to Detect Spousal Sabotage

Jared Gore + Liz Gore

CloudSecNext SUMMIT 2025

sans.org/CloudSecNextSummit

SANS

Jared Gore

Cloud Security Engineer

Primary Household Contribution:

🍞 Sourdough Sweetie

Liz Gore

Director of IT & Operations

Primary Household Contribution:

🔧 1/2 finished DIY projects

CloudSecNext SUMMIT 2025

sans.org/CloudSecNextSummit

SANS

Accidental IT Guy

Liz graduated from SANS Cyber Academy!
She got promoted!
She played Baldur's Gate 3!
Then…she got bored.

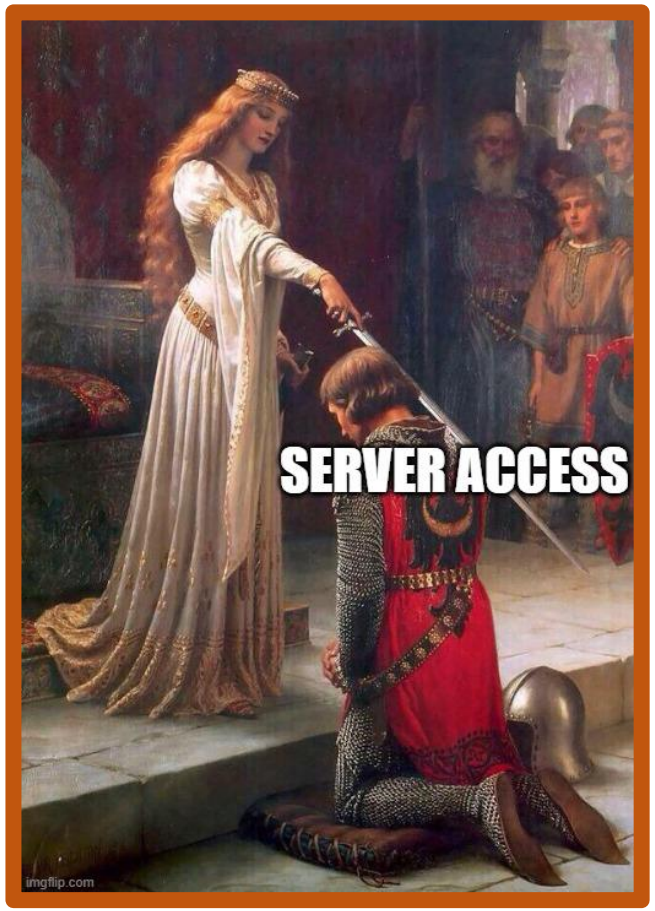
How can we stay sharp and have fun at the same time?

Make it a Capture The Flag!

Play > Perfection
Collaboration Strengthens Both Sides
Real Scenarios = Real Learning
Push Each Other to Level Up

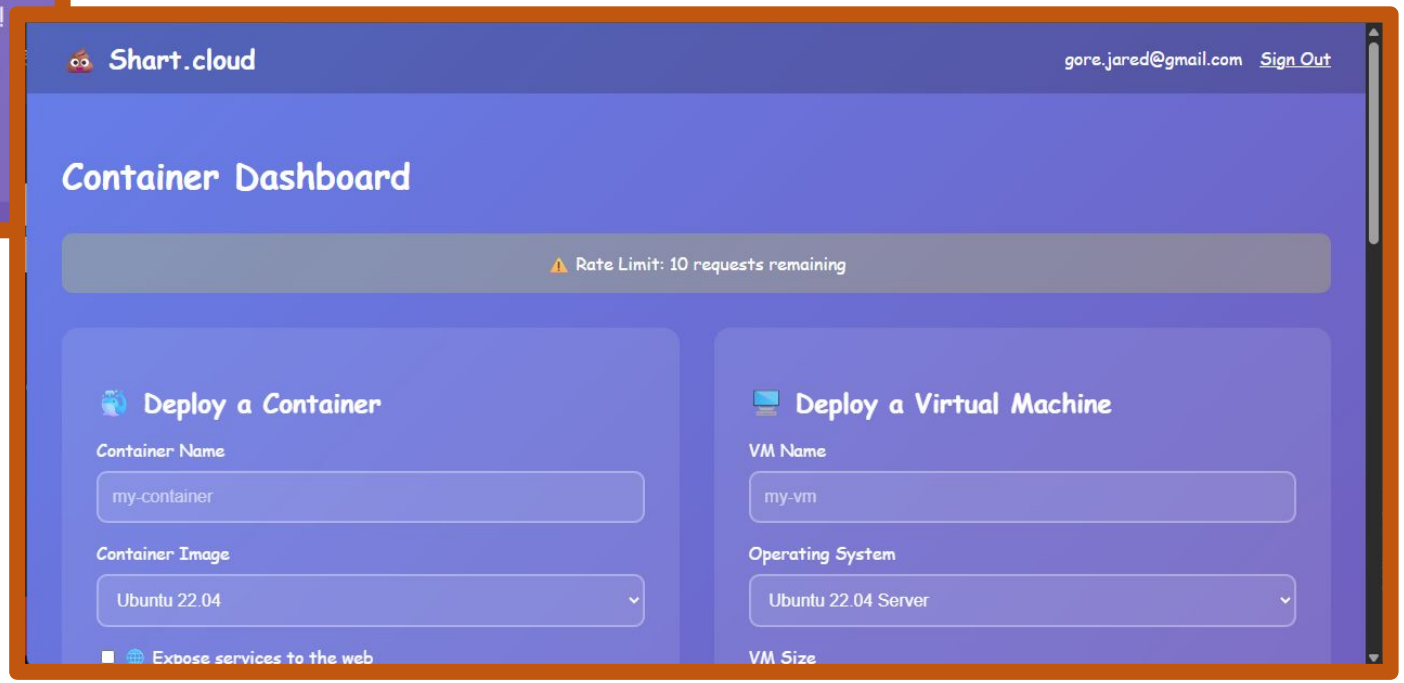
Winner Takes All

Welcome to the Homelab

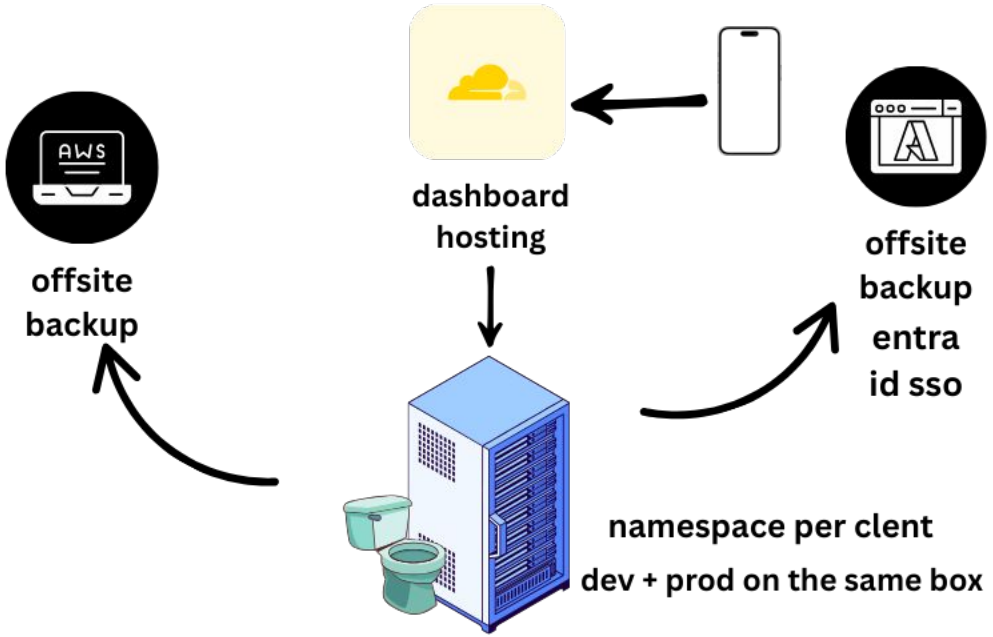


~vibe coded~

Web Console / API



shart.cloud infrastructure



short.cloud's "Customers"

Fortune 100^100 enterprises*
Popular Minecraft server (7 concurrent viewers!)
Multiple fraud customers mining crypto

*1e+200†

† I don't know what that means either

CTF Begins:

What access does our new employee have?

Liz's Starting Point

- K8s "read-only" access
- kubelogin SSO to prod cluster

DPRK TTP Playbook

Insider Threat Motivation:

Money - Access to financials

Power - Control over infrastructure

Information - Secrets that can be extorted

Plan of Attack:

Begin data reconnaissance for valuable targets

Establish persistent access via service accounts and roles

Escalate permissions across K8s → AWS → Azure

Detect with DuckDB

DuckDB - online analytics processing database

DuckDB has cloud native integrations to services like S3 or Azure Blob
Supports CSV, JSON, and Parquet files

Fast, Fun, and Free!

Detect with DuckDB (pt. 2)

```
tailpipe collect
aws_cloudtrail_logs.shart_trail_001
```

Tailpipe from Turbot!

Open Source SIEM for Terminal
Modular Plugins for AWS, Azure, & more

```
tailpipe query "select * from
aws_cloudtrail_log"
```

https://tailpipe.io

Detect with DuckDB (pt. 3)

Corkscrew from me!

Open Source Cloud Configuration Scanner
Modular Plugins for AWS, Azure, GCP, K8s
Query Configuration w/ SQL

```
corkscrew scan --services ec2,s3
```

```
corkscrew query --query "SELECT
type, COUNT(*) as count FROM
aws_resources"
```

https://github.com/jlgore/corkscrew

Detection Game

🟢 = Defender Point: Detection query catches attack

❌ = Attacker Point: Attack goes undetected

SCOREBOARD

Defender: 0 | Attacker: 0

ROUND 1: "The New Employee Special"

Enumerating shart.cloud

What Liz Actually Does:

- Authenticates as `oidc:lg@shart.cloud` via Entra ID OIDC
- Tests permissions: `kubectl auth can-i list secrets -n backup`
- Misconfigured cluster-reader role can read all secrets
- North Korean worker behavior: legitimate access for unauthorized recon

Detection Challenge:

Can Jared catch secret enumeration from a "read-only" user?

Can We Catch It? (Round 1 Results)

🟢 Found specific secret targeting behavior
🟡 Targeted high value namespaces only
❌ No mass enumeration alerts = flew under traditional detection

SCOREBOARD

Defender: 1 | Attacker: 1

Can We Catch It? (Round 1 Results)

Tailpipe (SIEM) ❌

```sql
-- Detect unusual secret access patterns
SELECT
    user_name,
    COUNT(DISTINCT object_namespace) as namespaces_accessed,
    COUNT(*) as secret_reads,
    string_agg(DISTINCT object_name, ', ') as secrets_accessed
FROM kubernetes_audit_logs
WHERE user_name = 'oidc:lg@shart.cloud'
    AND verb = 'get'
    AND resource = 'secrets'
    AND stage_timestamp > now() - interval '1 hour'
GROUP BY user_name
HAVING COUNT(DISTINCT object_namespace) >= 1;
```

Corkscrew (CSPM) ✅

```sql
-- Detect ClusterRoles with dangerous secrets access
SELECT
    name,
    type,
    CASE
        WHEN raw_data LIKE '%"resources":["*"]%' AND (raw_data LIKE
'%"get"%' OR raw_data LIKE '%"list"%')
        THEN '🔴 CRITICAL: Wildcard read access (includes secrets)'
        WHEN raw_data LIKE '%"resources":["secrets"]%' AND (raw_data LIKE
'%"get"%' OR raw_data LIKE '%"list"%')
        THEN '🟡 MEDIUM: Direct secrets read access'
        ELSE '✅ SAFE'
    END as risk_level
FROM kubernetes_resources
WHERE type = 'ClusterRole'
    AND ((raw_data LIKE '%"resources":["*"]%' AND (raw_data LIKE
'%"get"%' OR raw_data LIKE '%"list"%'))
         OR (raw_data LIKE '%"resources":["secrets"]%' AND (raw_data LIKE
'%"get"%' OR raw_data LIKE '%"list"%')))
ORDER BY risk_level DESC;
```

https://github.com/jlgore/shart-sql/blob/main/ROUND4.md

Round 2: "From Reader to Root”

backup-operator Token = Game Over

✅ Can exec into pods in default namespace (!!)
✅ Can create pods in kube-system namespace (!!!)
✅ Can read ALL secrets (why??)

Detection Challenge:

Can Jared detect a "backup" service account executing into application pods?

Can We Catch It? (Round 2 Results)

Tailpipe (SIEM) ✅

```
-- Detect service account pod exec
SELECT stage_timestamp, user_name,
       REGEXP_EXTRACT(request_uri, '/pods/([^/]+)/', 1) as pod_name,
       REGEXP_EXTRACT(request_uri, '/namespaces/([^/]+)/', 1) as
namespace
FROM kubernetes_audit_logs
WHERE subresource = 'exec'
  AND user_name LIKE '%backup-operator%'
ORDER BY stage_timestamp DESC;)))
ORDER BY risk_level DESC;
```

Corkscrew (CSPM) ✅

```
SELECT name,
       CASE WHEN raw_data LIKE '%pods/exec%' AND raw_data LIKE '%create%'
            THEN '=4 CRITICAL: Pod exec + secrets access'
            ELSE '• SAFE' END as risk
FROM kubernetes_resources
WHERE type = 'ClusterRole'
  AND name = 'backup-operator-role';
```

https://github.com/jlgore/shart-sql/blob/main/ROUND2.md

Round 3: "The SSE-C Heist"

Can We Catch It? (Round 3 Results)

🟢 DETECTED: shart-cloud-velero-backups bucket
🔴 MISSING: MFA Delete status not enabled

SCOREBOARD

Defender: 2 | Attacker: 3

Can We Catch It? (Round 3 Results)

Tailpipe (SIEM) ✅

```
-- Detect S3 SSE-C operations from CloudWatch/CloudTrail logs
SELECT event_time, event_name,
       user_identity,
       request_parameters
FROM aws_cloudtrail_log
WHERE event_source = 's3.amazonaws.com'
  AND event_name IN ('CopyObject', 'PutObject')
  AND request_parameters LIKE '%sse-customer%'
ORDER BY event_time DESC;
```

Corkscrew (CSPM) ❌

```
SELECT name,
       CASE WHEN raw_data LIKE '%MfaDelete%Enabled%'
            THEN '🟢 MFA DELETE: Enabled'
            WHEN raw_data LIKE '%Versioning%Enabled%'
            THEN '🟡 VERSIONING: Enabled but no MFA delete'
            ELSE '🔴 VULNERABLE: No versioning or MFA delete'
END as protection_level
FROM aws_resources
WHERE type = 'Bucket'
  AND (name LIKE '%velero%' OR name LIKE '%backup%');
```

https://github.com/jlgore/shart-sql/blob/main/ROUND3.md

"The Encryption Finale"

Ransomware Deployment

Attack Phase:

Deploy encryption tools across all cloud storage platforms
Target customer backup data for maximum impact
Encrypt data in-place using cloud-native capabilities
Leave ransom demand with specific…requirements

Can We Catch It? (FINAL SCORE)

```sql
SELECT
    event_time,
    event_name,
    user_identity.user_name,
    user_identity.type as user_type,
    source_ip_address,
    request_parameters->'key' as object_key,
    request_parameters->'bucketName' as bucket,
    json_extract(additional_event_data, '$.bytesTransferredIn') as file_size,
    json_extract(response_elements, '$.x-amz-version-id') as version_id
FROM aws_cloudtrail_log
WHERE event_name = 'PutObject'
    AND (
        lower(cast(request_parameters->'key' as varchar)) LIKE '%ransom%'
        OR lower(cast(request_parameters->'key' as varchar)) LIKE '%readme%'
        OR lower(cast(request_parameters->'key' as varchar)) LIKE '%decrypt%'
        OR lower(cast(request_parameters->'key' as varchar)) LIKE '%recover%'
        OR lower(cast(request_parameters->'key' as varchar)) LIKE '%puppy%'
    )
    AND event_time >= '2025-09-25T15:00:00Z'
ORDER BY event_time;
```

🟢 Mass modifications detected
🟢 Encryption tools detected

FINAL SCORE

Defender: 3 | Attacker: 5

https://github.com/jlgore/shart-sql/blob/main/ROUND4.md

Liz's Demand:

WE WANT A DOG!

Did shart.cloud pay the ransom?

Meet

Rue!

Adopted:

9/2/2025

Fun Exercise → Real Threat

DPRK TTP Playbook demonstrated:

Legitimate access as attack vector
Multi-cloud targeting
Evolution to active extortion

Lessons Learned

Jared: Detection is iterative, not perfect.

Liz: The magic isn't in the complexity - it's in the curiosity.

Practical Takeaways

BUDGET-FRIENDLY STACK

DuckDB is great in a pinch, but not perfect

Many features, many drawbacks

Cloud-native support is nice

LEARNING TOGETHER

The joy of play

Deeper learning for both sides

If you want to go fast go alone, if you want to go far go together

Your turn?

We're planning to release a CTF!

YOU CAN PLAY TOO:

Check out our website → **shart.cloud**
Early access starting November 28th
Planned CTF release on 12/21

Connect with us on LinkedIn or email: jg@shart.cloud / lg@shart.cloud

Thank you for joining our cyber shenanigans!

CloudSecNext SUMMIT 2025

sans.org/CloudSecNextSummit

SANS