Testing Across Cloud Resources: A GOAT's Cheatsheet

Map, Hunt, and Headbutt Your Way to Keeping Your Cloud Resources Safer

In the cloud, VMs, containers, serverless, storage buckets, load balancers, and all the other bits and bobs operate in the one barn. Some assets vanish before you even spot them, others hide secrets in plain sight. The challenge is not about determining if they're exposed, but mapping how they can be chained together into real attack paths.

1. Framework to Determine the Risk

For every asset, ask:

- Reachability: Is it reachable from my current attack footholds?
- Data Exposure: What data does it unlock?
- Privilege: What permissions does it hold (and could escalate to)?
- Pivot: Can it be chained to something juicier?

2. Tactical Cheats

Fill in the Blanks of your CMDB:

- [ ] Use automated resource mapping tools to spot unmanaged & ephemeral assets.
- [ ] Look for shadow IT, short-lived resources, and untagged deployments.

Credential Hunting in Code & Configs:

- [ ] Hunt for hardcoded credentials in Lambdas, App Services, and containers.
- [ ] Scan IaC templates & git repos for plaintext tokens.

Slippery Suspects (Dynamic Assets):

- [ ] Don't let ASGs, containers, and serverless functions fall off your radar.
- [ ] Test for: shared node privilege leakage, embedded secrets, access to prod storage without logging

3. Common Attack Paths to Validate

Ephemeral ↔ Persistent
Short-lived workloads opening doors to long-lived assets.

Secrets ↔ Lateral Movement
Short-lived workloads opening doors to long-lived assets.

Containers ↔ Storage Buckets
Chaining across services.
Map container attack paths against MITRE ATT&CK for Containers.

Top Tip **Cloud diversity = cloud complexity.**
Attackers thrive on chaos. Automate discovery, and don't ask "Is it exposed?" but "How does it chain?"