Hunting with AWS.
The Brief.

Hunting in a public cloud, like Amazon Web Services (AWS), has various advantages and challenges. Traditional techniques apply in many cases, but there are also novel areas involving cloud native services and APIs. We will explore several case studies and the techniques and tools one could apply to resolve them.
Conventional hunting in the cloud.
With Elastic Cloud Compute (EC2) instances you have control of everything from the kernel up. In our case this looks just like an Ubuntu server. As such you can do standard Linux things to these instances to perform hunting operations. Additionally AWS offers native orchestration tools. Combining traditional live response scripts with AWS Simple Systems Manager (SSM) yields a powerful hunting platform.
SIMPLE SYSTEM MANAGER
Cloud hunts for conventional threats.
Leveraging native AWS services provides solid logging capability and various control points: the building blocks of a hunting platform. Using these you can detect traditional attacks from the cloud layer.
ELB ACCESS LOGS FOR HUNTING

ELB Access Log Format

```
timestamp  elb  client:port  backend:port  request_processing_time
backend_processing_time  response_processing_time  elb_status_code
backend_status_code  received_bytes  sent_bytes  "request"  "user_agent"  ssl_cipher
ssl_protocol
```

Some good hunting to be had based on the highlighted fields

- String matches on request or user_agent
- Unusually long or large requests, or error codes from the backend

Use ELB name and backend IP address to find the instance. Look up ELB.

```
aws elb describe-load-balancers --load-balancer-name elb
```

Which yields list of instances, filter them on backend ip

```
aws ec2 describe-instances --filters Name=network-interface.addresses.private-ip-address,Values=backendIP --instance-Ids "instanceIds" "from" "elb"
```
Cloud-native threats and hunts.
Cloud infrastructure provides the defender a number of tools, but it also creates an attack surface. Hunting for cloud native threats that seek to exploit that attack surface for persistence, lateral movement and exfiltration is an important capability for those moving into the cloud.
CloudTrail Entries of eventName:"ModifySnapshotAttribute"

Time requestParameters.createVolumePermission.add.items userIdentity.arn
xxxx "{userId": "5ATTACKERXX5"} arn:aws:sts::xxxxxxxxxxx5:assumed-role/InstanceProfile/i-
0c46f696b6e6a6a6c
xxxx "{userId": "5ATTACKERXX5"} arn:aws:sts::xxxxxxxxxxx5:assumed-
role/aws_user/user@company.com

Look for accounts other than your own
I.e. a WHERE add.items NOT IN (list of accounts) or by joining with a table containing your account list if it is large.

These are the instances or users that might be compromised.
Also check these account Ids on any interesting API calls.

See also ModifyImageAttribute for sharing AMI’s, and look at cross account IAM Roles. Hunting through your policy settings is another way to catch cross account sharing. It helps to have your policies in source control.
Thank you.