

Automating Disk and Memory Evidence Collection in AWS

Ryan Tick & Vaishnav Murthy

BSides SATX

July 11, 2020



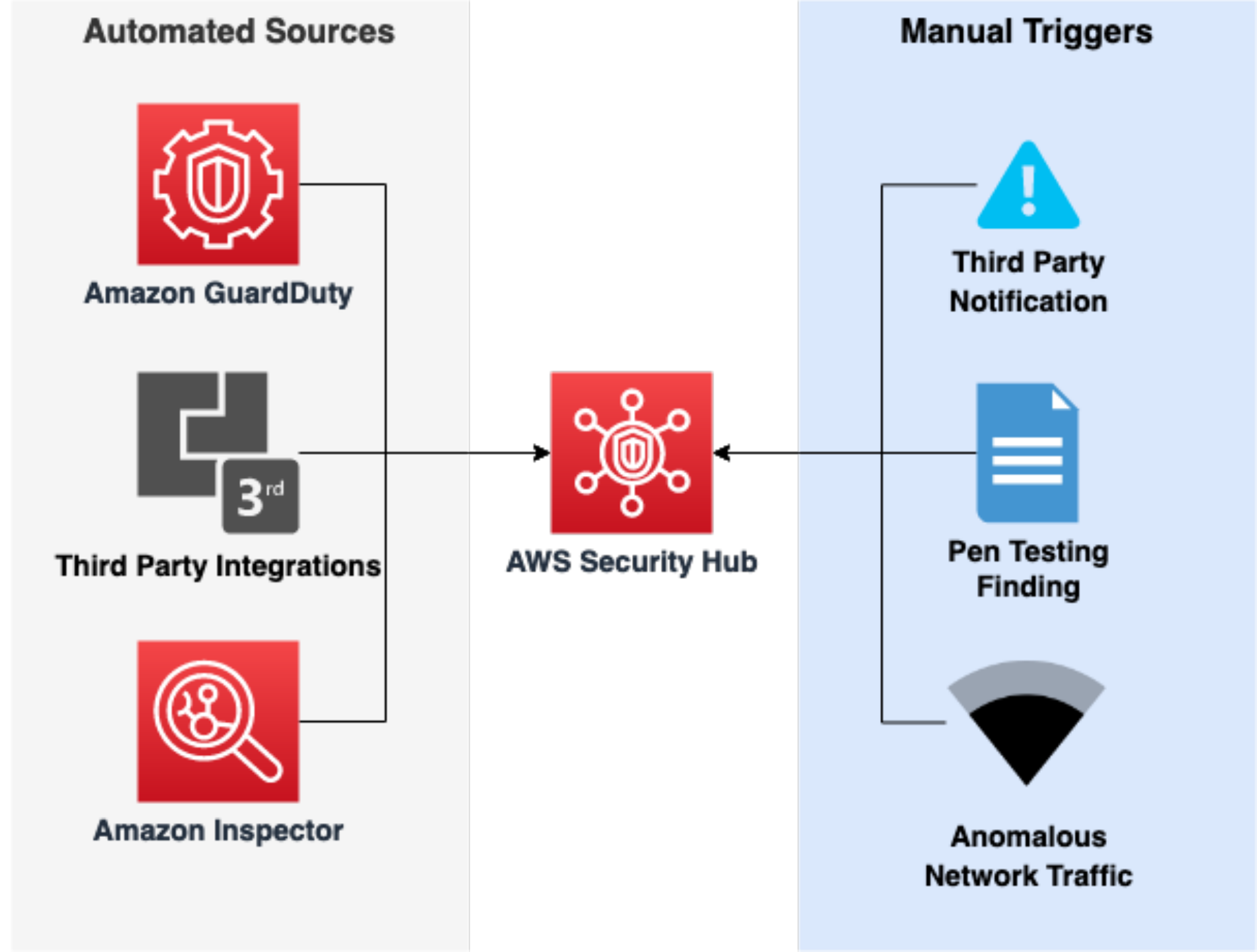
Ryan Tick and Vaishnav Murthy

- Cloud Security Architects at Goldman Sachs
 - SANS (GCFE, GCFA, GNFA) & AWS (SAA, SOA, DVA, SCS) certified
 - Previously DFIR consultants for KPMG
 - University of Notre Dame Graduates
-
- Fun facts

Overview

Problem Statement

- Companies have successfully automated on-prem evidence collection using both commercial and open-source tools. However, few have tackled this task in the cloud.
- Primarily concerned about:
 - Scalable (3 orgs and over 3000 accounts)
 - Auditable (AWS CloudTrail)
 - Automated/repeatable (cut down on human error)



Collection Outputs

– Disk Evidence

- Raw dd image per volume
- dc3dd log files
- Custom collection log files

– Memory Evidence

- Full memory capture
- If Linux, custom memory profile
- Collection log files

Main Takeaways

- Much more going on behind the scenes to do it right
- MTTR, cost, and error-rate reduced
- Scalable and auditable while following AWS best practices
- Collection done over the AWS backbone network



Well-Architected



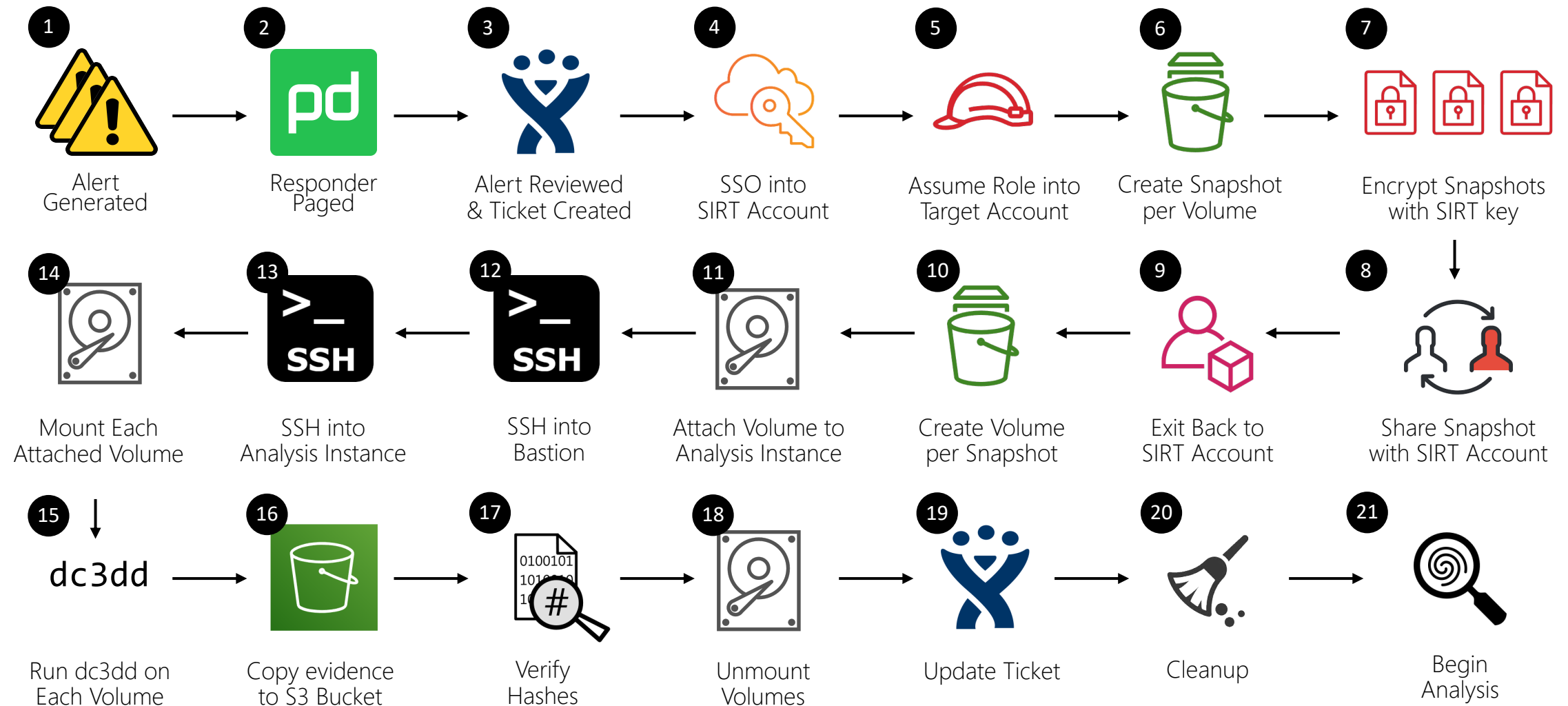
Disk Collection

Deep Dive

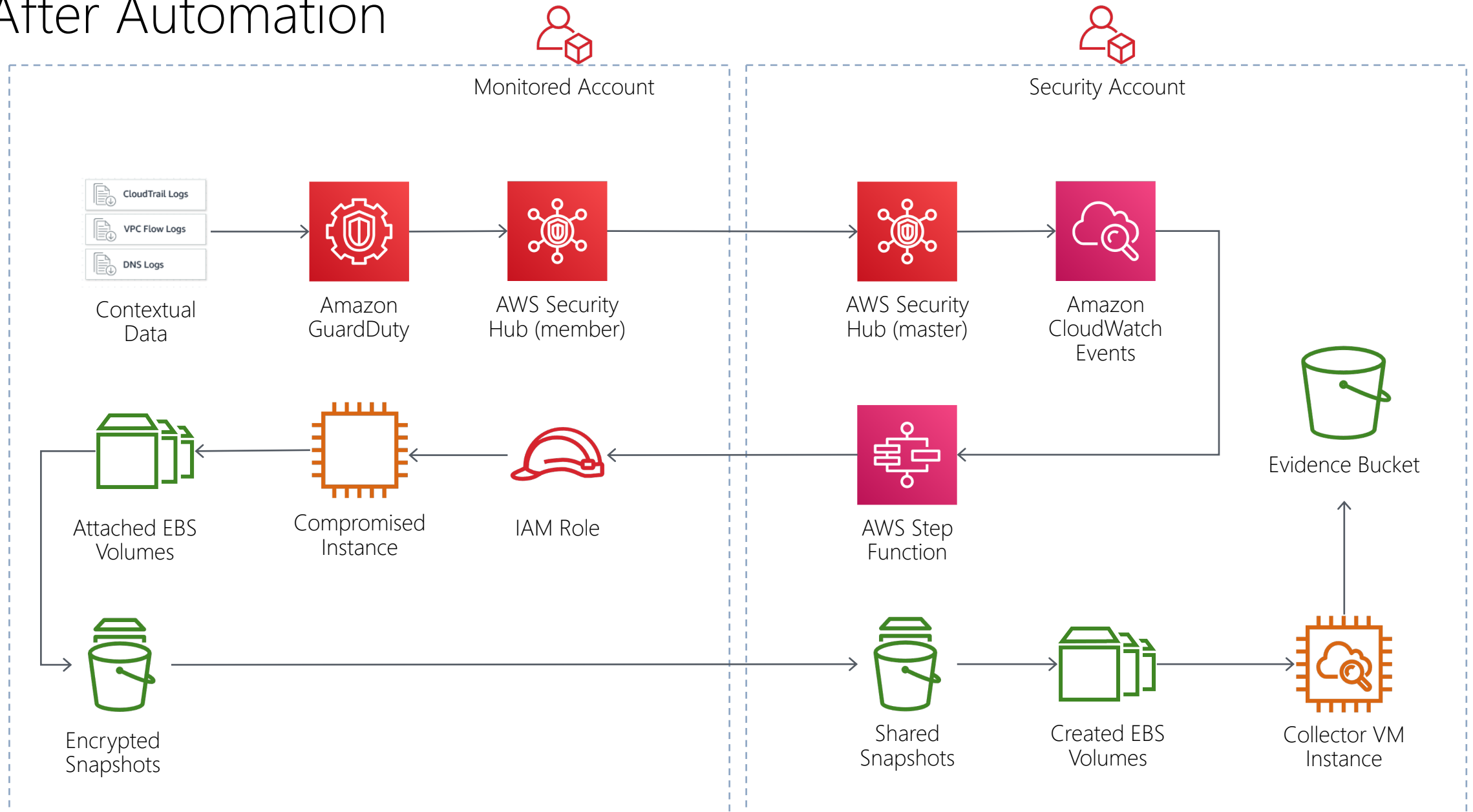
What Exists Today

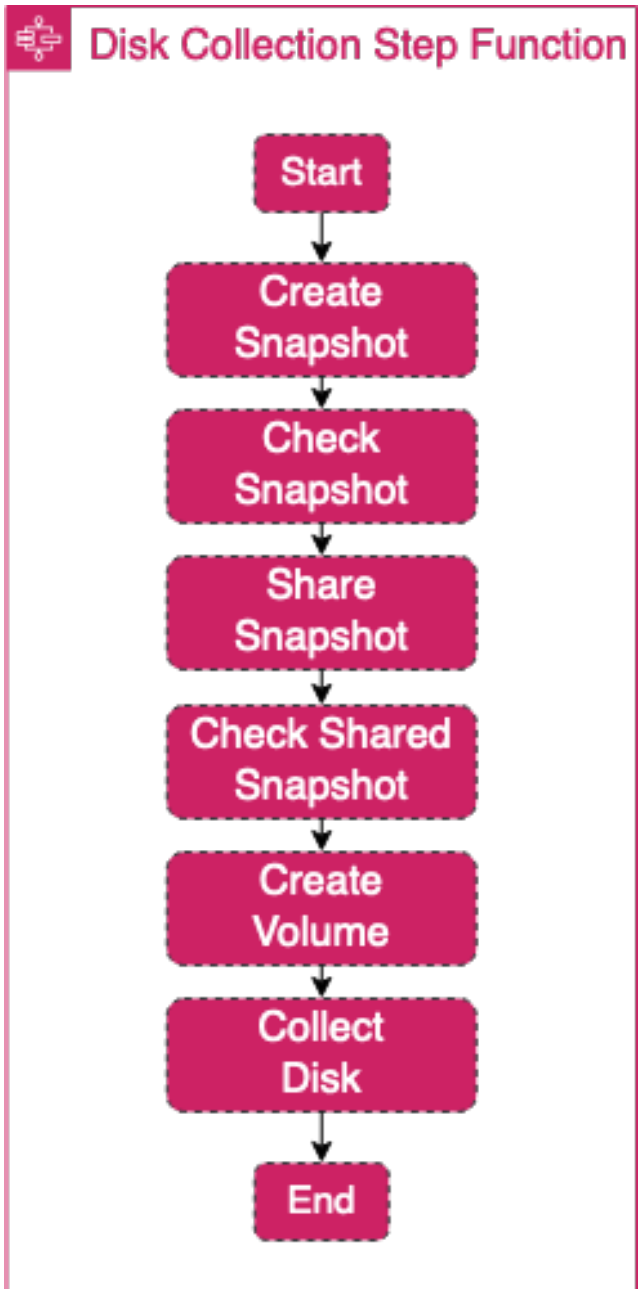
- Much of the documentation on best practices in AWS digital forensics suggests the use of a “forensic workstation” AMI, “forensic” VPC, and an “investigator” user or role in each account
- Does not fit our use case:
 - Does not necessarily follow principle of least privilege
 - Potentially takes up resources in a production account (AWS service limits)

Before Automation

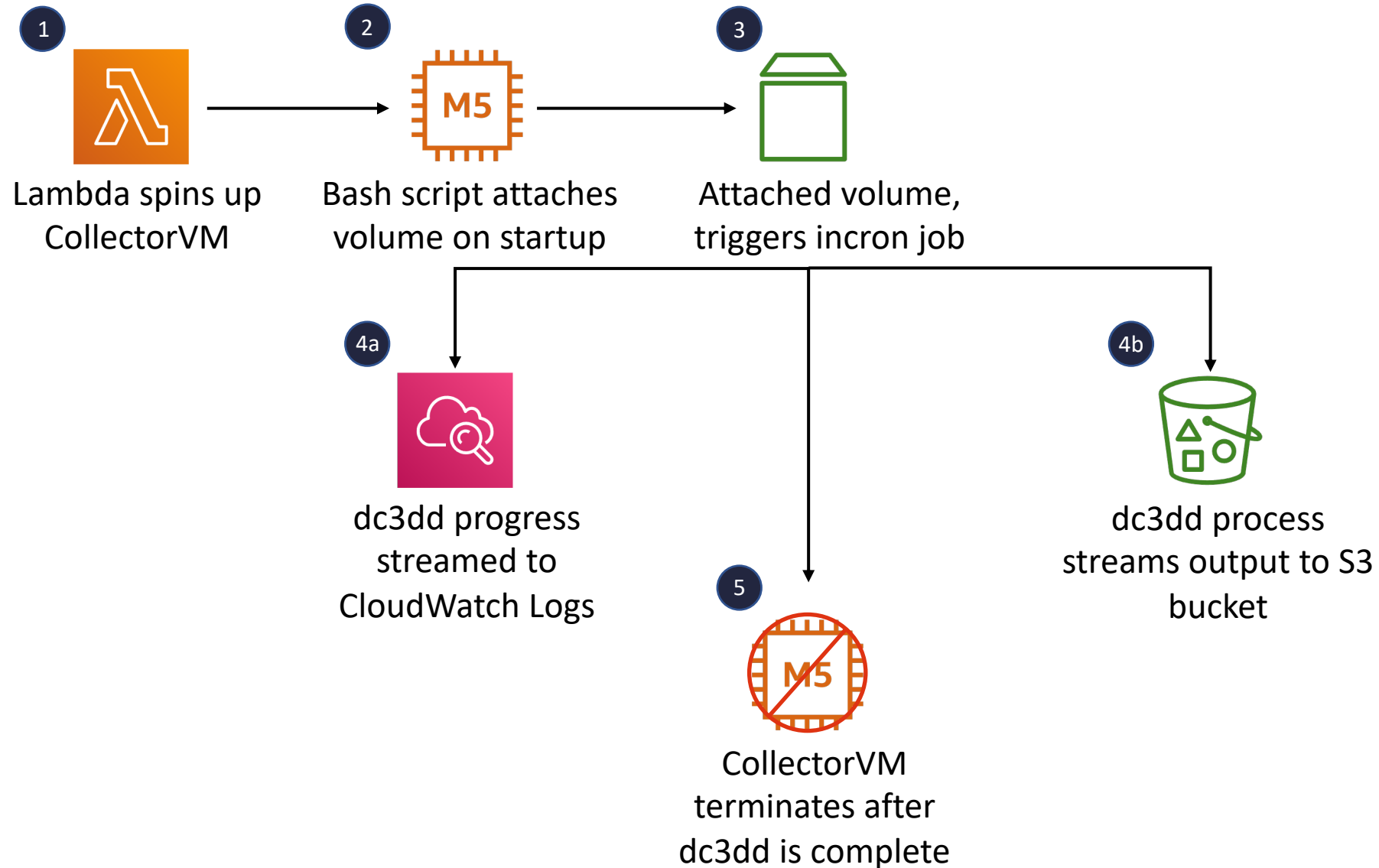


After Automation



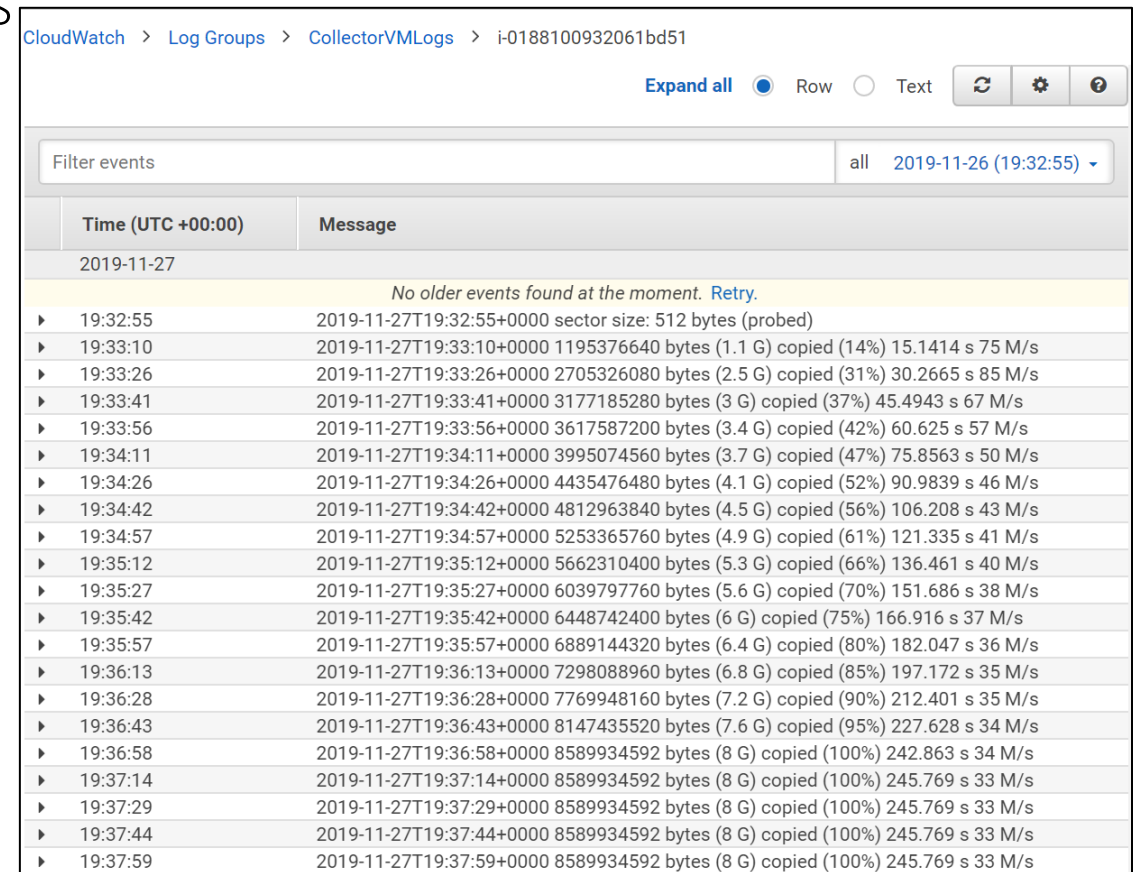


CollectorVM



CloudWatch Logs Agent

- Creates the log group “CollectorVMLogs” in the Security account
- Creates a log stream per instance ID of the CollectorVM
- Necessary since the CollectorVM does not have SSH access
- Allows real-time monitoring of speed and collection progress
 - Bytes copies
 - Collection speed
 - Time elapsed
- Additionally we’re streaming all standard logs off the instance that may help us debug



The screenshot shows the AWS CloudWatch Logs console interface. The breadcrumb navigation at the top reads: CloudWatch > Log Groups > CollectorVMLogs > i-0188100932061bd51. Below the navigation, there are controls for 'Expand all' (a radio button is selected), 'Row', and 'Text' view options, along with icons for refresh, settings, and help. A 'Filter events' input field is present, followed by a dropdown menu set to 'all' and a date/time filter '2019-11-26 (19:32:55)'. The main content area displays a table of log events. The table has two columns: 'Time (UTC +00:00)' and 'Message'. The events are listed chronologically, starting from 2019-11-27T19:32:55. The messages describe the progress of a disk collection process, including sector size probing and the copying of data in 1 G increments, with progress percentages and speeds (e.g., 15.1414 s 75 M/s) provided for each step.

Time (UTC +00:00)	Message
2019-11-27	No older events found at the moment. Retry .
2019-11-27T19:32:55	2019-11-27T19:32:55+0000 sector size: 512 bytes (probed)
2019-11-27T19:33:10	2019-11-27T19:33:10+0000 1195376640 bytes (1.1 G) copied (14%) 15.1414 s 75 M/s
2019-11-27T19:33:26	2019-11-27T19:33:26+0000 2705326080 bytes (2.5 G) copied (31%) 30.2665 s 85 M/s
2019-11-27T19:33:41	2019-11-27T19:33:41+0000 3177185280 bytes (3 G) copied (37%) 45.4943 s 67 M/s
2019-11-27T19:33:56	2019-11-27T19:33:56+0000 3617587200 bytes (3.4 G) copied (42%) 60.625 s 57 M/s
2019-11-27T19:34:11	2019-11-27T19:34:11+0000 3995074560 bytes (3.7 G) copied (47%) 75.8563 s 50 M/s
2019-11-27T19:34:26	2019-11-27T19:34:26+0000 4435476480 bytes (4.1 G) copied (52%) 90.9839 s 46 M/s
2019-11-27T19:34:42	2019-11-27T19:34:42+0000 4812963840 bytes (4.5 G) copied (56%) 106.208 s 43 M/s
2019-11-27T19:34:57	2019-11-27T19:34:57+0000 5253365760 bytes (4.9 G) copied (61%) 121.335 s 41 M/s
2019-11-27T19:35:12	2019-11-27T19:35:12+0000 5662310400 bytes (5.3 G) copied (66%) 136.461 s 40 M/s
2019-11-27T19:35:27	2019-11-27T19:35:27+0000 6039797760 bytes (5.6 G) copied (70%) 151.686 s 38 M/s
2019-11-27T19:35:42	2019-11-27T19:35:42+0000 6448742400 bytes (6 G) copied (75%) 166.916 s 37 M/s
2019-11-27T19:35:57	2019-11-27T19:35:57+0000 6889144320 bytes (6.4 G) copied (80%) 182.047 s 36 M/s
2019-11-27T19:36:13	2019-11-27T19:36:13+0000 7298088960 bytes (6.8 G) copied (85%) 197.172 s 35 M/s
2019-11-27T19:36:28	2019-11-27T19:36:28+0000 7769948160 bytes (7.2 G) copied (90%) 212.401 s 35 M/s
2019-11-27T19:36:43	2019-11-27T19:36:43+0000 8147435520 bytes (7.6 G) copied (95%) 227.628 s 34 M/s
2019-11-27T19:36:58	2019-11-27T19:36:58+0000 8589934592 bytes (8 G) copied (100%) 242.863 s 34 M/s
2019-11-27T19:37:14	2019-11-27T19:37:14+0000 8589934592 bytes (8 G) copied (100%) 245.769 s 33 M/s
2019-11-27T19:37:29	2019-11-27T19:37:29+0000 8589934592 bytes (8 G) copied (100%) 245.769 s 33 M/s
2019-11-27T19:37:44	2019-11-27T19:37:44+0000 8589934592 bytes (8 G) copied (100%) 245.769 s 33 M/s
2019-11-27T19:37:59	2019-11-27T19:37:59+0000 8589934592 bytes (8 G) copied (100%) 245.769 s 33 M/s

Pro Tips

- Lambdas multi-threading vs. concurrent execution
- Full disk collection or triage collection?
 - Full disk collection in tandem with high priority logs streamed to CloudWatch Logs
- Streaming dc3dd output directly to S3
- Auditable and done entirely over AWS backbone network
- Cost vs. collection speed
- Instance store data
- Logging
 - Hashing SSD vs. HDD

Memory Collection

Deep Dive

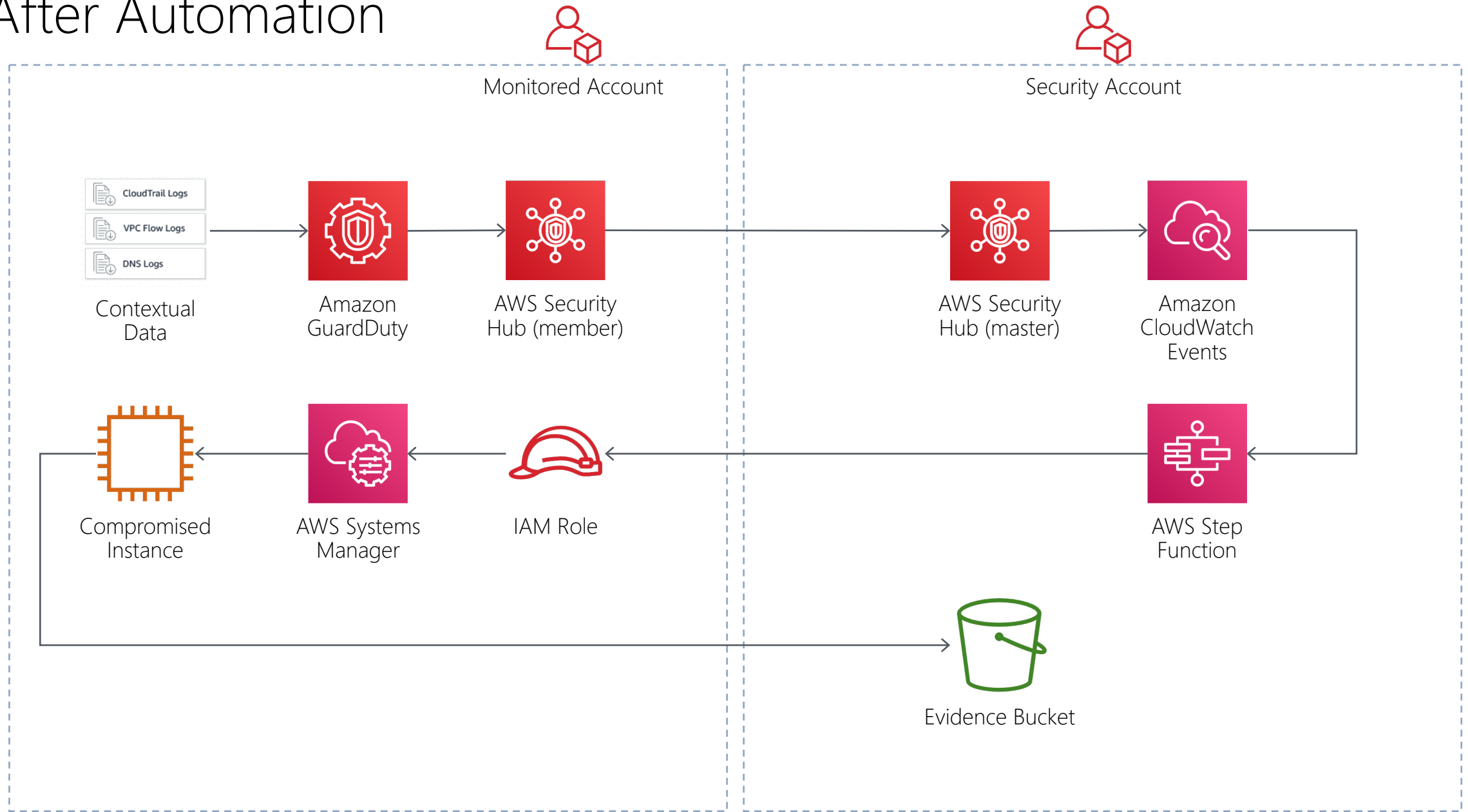
What Exists Today: Margarita Shotgun

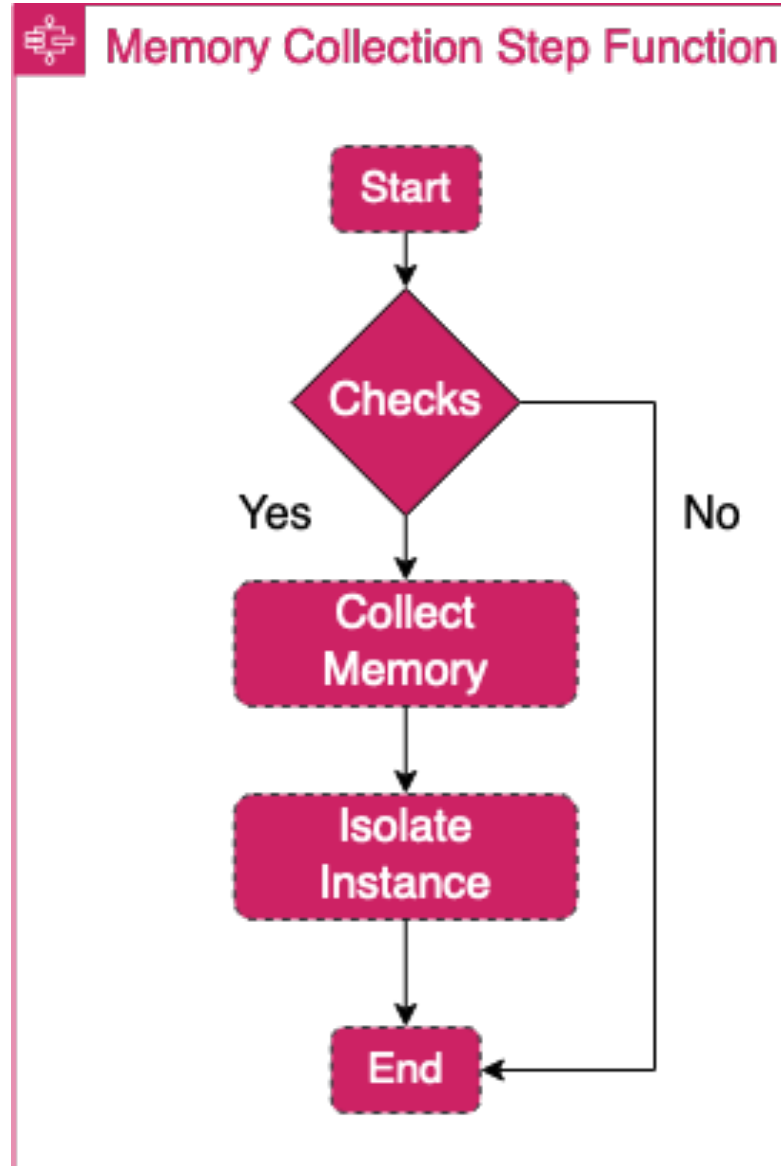
- Developed by ThreatResponse
- Remote memory acquisition tool:
<https://github.com/ThreatResponse/margaritashotgun>
- Allows streaming memory directly to an S3 bucket
- Great tool, but requires SSH access into the target instance

Before Automation



After Automation





Streaming Directly to S3

- **Initial challenge:** Instance needs to have enough storage to collect full memory dump
- **Solution:** Leverage S3 to upload a local file stream from standard input to an S3 evidence bucket
 - Need to compute memory size prior to stream and include **--expected-size** flag if size is larger than 50 GB
 - `aws s3 cp --acl bucket-owner-full-control - s3://evidence-bucket/{{FindingID}}/$(curl http://169.254.169.254/latest/meta-data/instance-id).mem`

Custom Memory Profiles (Linux)

- Create the LiME kernel module used to collect memory upon reboot if kernel changes
- Create the dwarf file containing the Linux kernel data structures
- Zip up dwarf file and System map file for the active kernel version

Ensuring Network Connectivity

- Are you comfortable streaming evidence to S3 over the public internet?
 - If yes, target VPC requires IGW or NAT Gateway
- What if the instance doesn't have internet access?
 - Dynamically create VPC endpoints (S3 and SSM)

Pro Tips

- Stream directly to S3
 - Expected size flag
- Consider business impact
 - VPC endpoints vs. NAT gateway vs. IGW
 - When to collect memory?
- File integrity monitoring
- Evidence verification
- Tracked vs untracked connections

Closing Thoughts

Future Work

- Collect instance store data
- Further leverage FIM to restrict access to necessary tools
- Automate analysis (T1/T2) of evidence using ECS/Fargate
 - Currently Fargate does not support Windows platform versions
- Assign confidence score based off analysis
- Generate example reports

Questions and Feedback

- LinkedIn
- Twitter
 - @tracer_tick
 - @VMurthyDFIR

Thank You!

Ryan Tick & Vaishnav Murthy

