SEAS-8414

Analytical Tools for Cyber Analytics

Survey of analytical tools for analyzing cyber security data with particular attention to the use of data analytics procedures in supporting appropriate cyber security policy decisions.

Dr. M

Welcome to SEAS Online at George Washington University

SEAS-8414 class will begin shortly

- **Audio:** To eliminate background noise, please be sure your audio is muted. To speak, please click the hand icon at the bottom of your screen (**Raise Hand**). When instructor calls on you, click microphone icon to unmute. When you've finished speaking, ***be sure to mute yourself again***.
- **Chat:** Please type your questions in Chat.
- **Recordings:** As part of the educational support for students, we provide downloadable recordings of each class session to be used exclusively by registered students in that particular class for their own private use. Releasing these recordings is strictly prohibited.

Agenda

Week-5: Code-centric security analytics tools

Since the data, servers, and infrastructure is secured, we will learn how to ensure the gwuscc.com code and supply-chain components are secure. We will cover the following tools:

- Static Application Security Testing (SAST)
- Dynamic Application Security Testing (DAST)
- Software Composition Analysis (SCA)

You will be introduced to expectations and variance to understand how to interpret the results from these tools.

Class-5

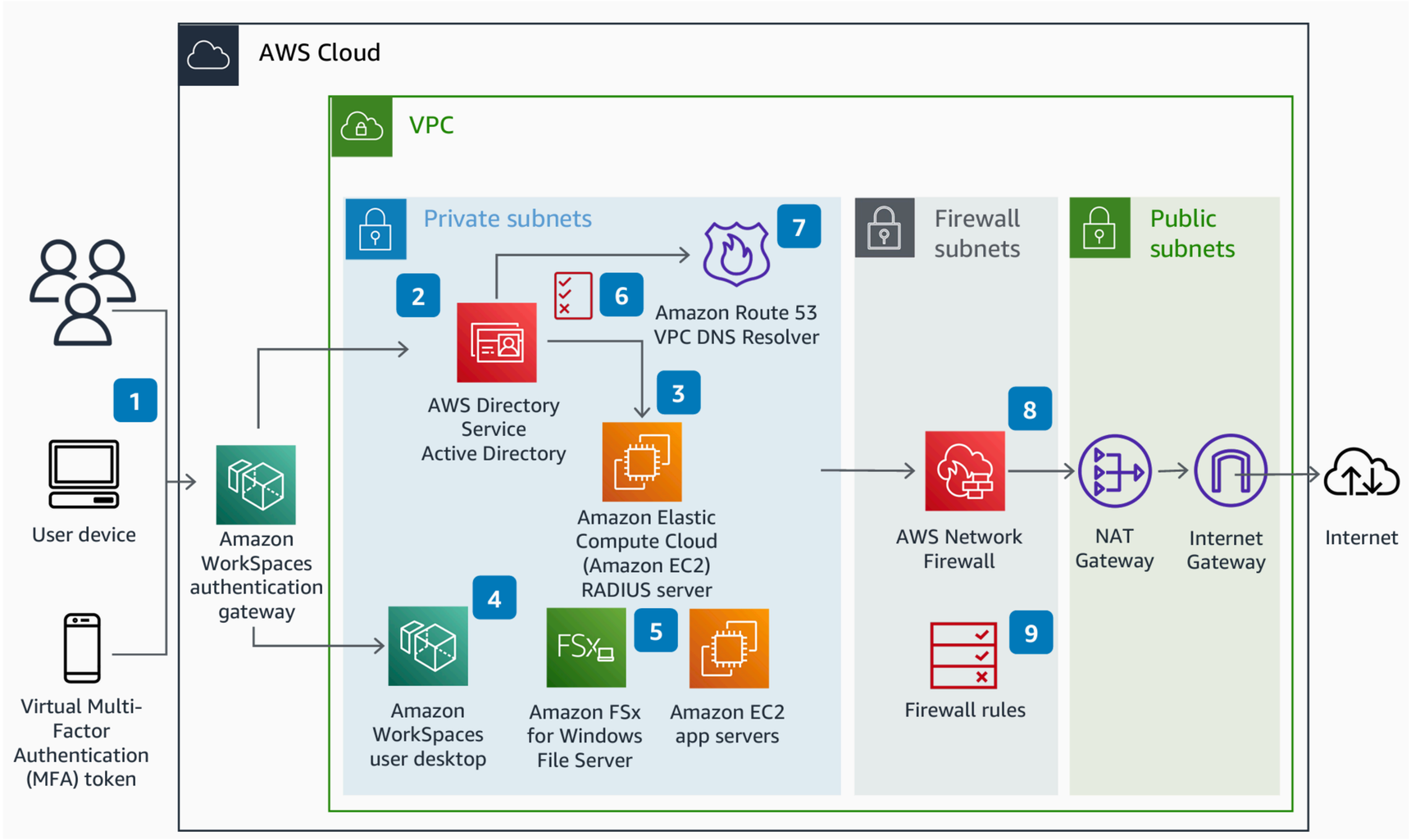
Structure

- Reading Cloud Architecture Diagrams
- Case #1: US Department of Defense Data Breach
- Case #2: RSA Secure ID Data Breach
- Case #3: Pharmacy Denial of Service Attack
- Case #4: Capital One Data Breach
- Live Practice

Reading the Cloud Architecture Diagrams

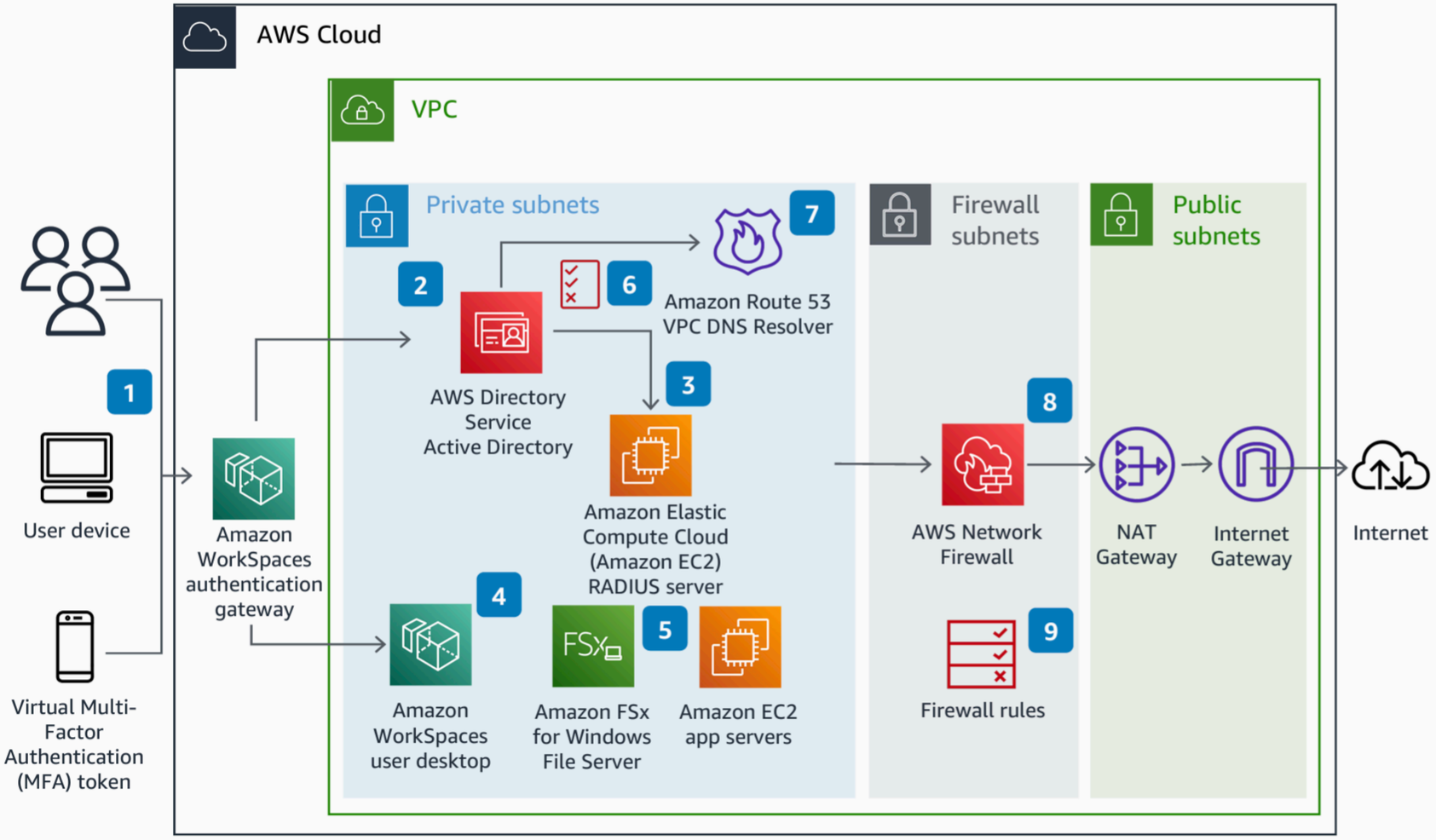
https://aws.amazon.com/architecture/reference-architecture-diagrams

What is happening here?



Secure Remote Worker Environment

Build a secure desktop environment for remote workers to access key line of business applications and data.

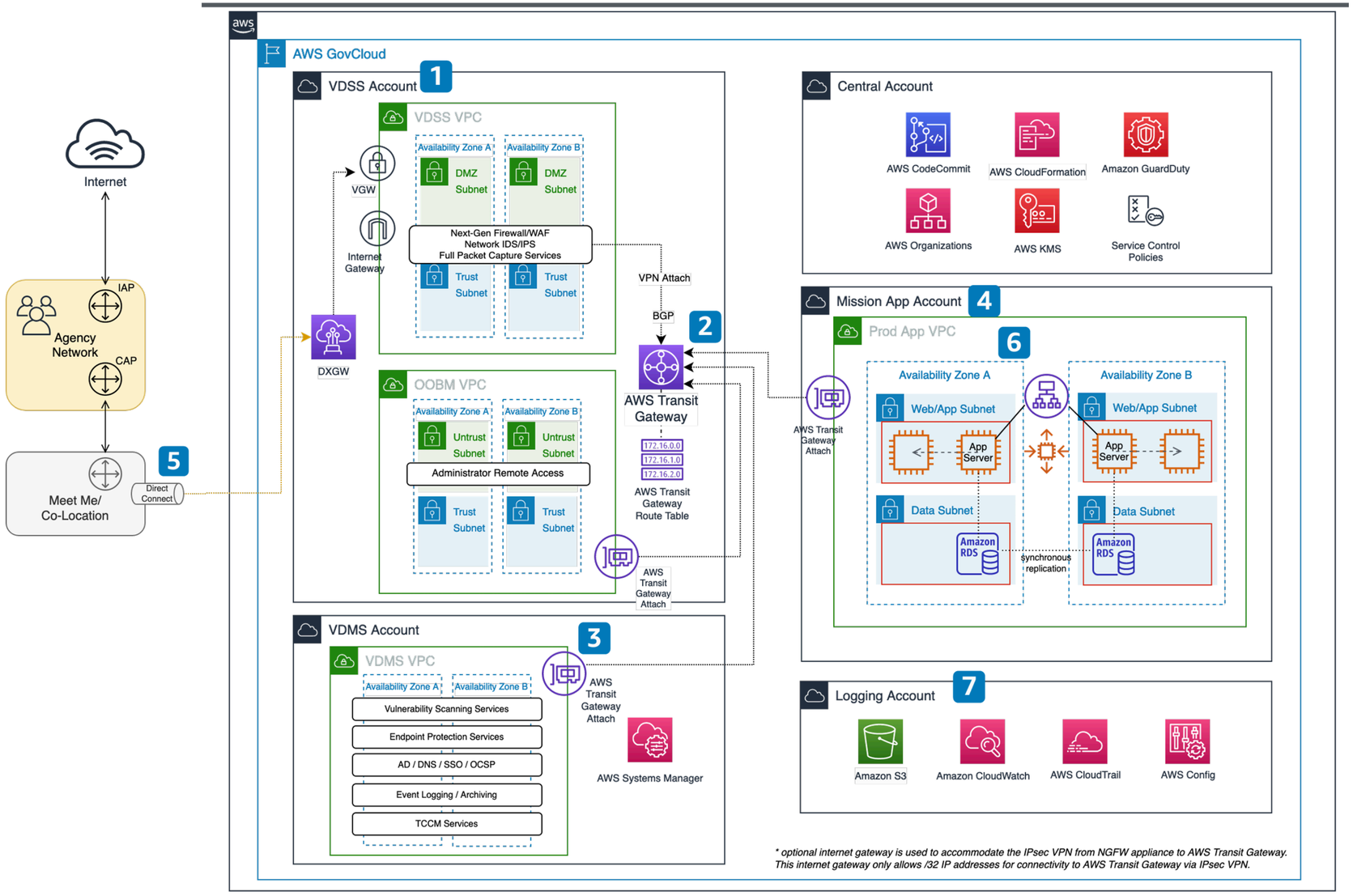


1. Users connect to their desktop using the **Amazon WorkSpaces** application by supplying a username, password, and MFA code.
2. **Amazon WorkSpaces** authentication gateway authenticates against Directory Service.
3. MFA code is authenticated against MFA service's RADIUS server. For example, OneLogin.
4. Users are connected to their desktop through **Amazon WorkSpaces.**
5. Users access core systems and files hosted on **Amazon EC2** and **Amazon FSx**.
6. Group policy is implemented in Active Directory to prevent unwanted activities, such as printing to local printers from **Amazon WorkSpaces**.
7. Domain Controller DNS forwards to **Amazon Route 53** VPC DNS resolver with applied Route 53 Resolver DNS Firewall rules.
8. Outbound internet traffic is filtered first by **AWS Network Firewall**, then sent through a NAT gateway and internet gateway to the public internet.
9. Firewall rules are set up to block outbound traffic to unwanted sites (such as file-sharing platforms) to prevent data leaks.

Reviewed for technical accuracy January 31, 2022

AWS Reference Architecture

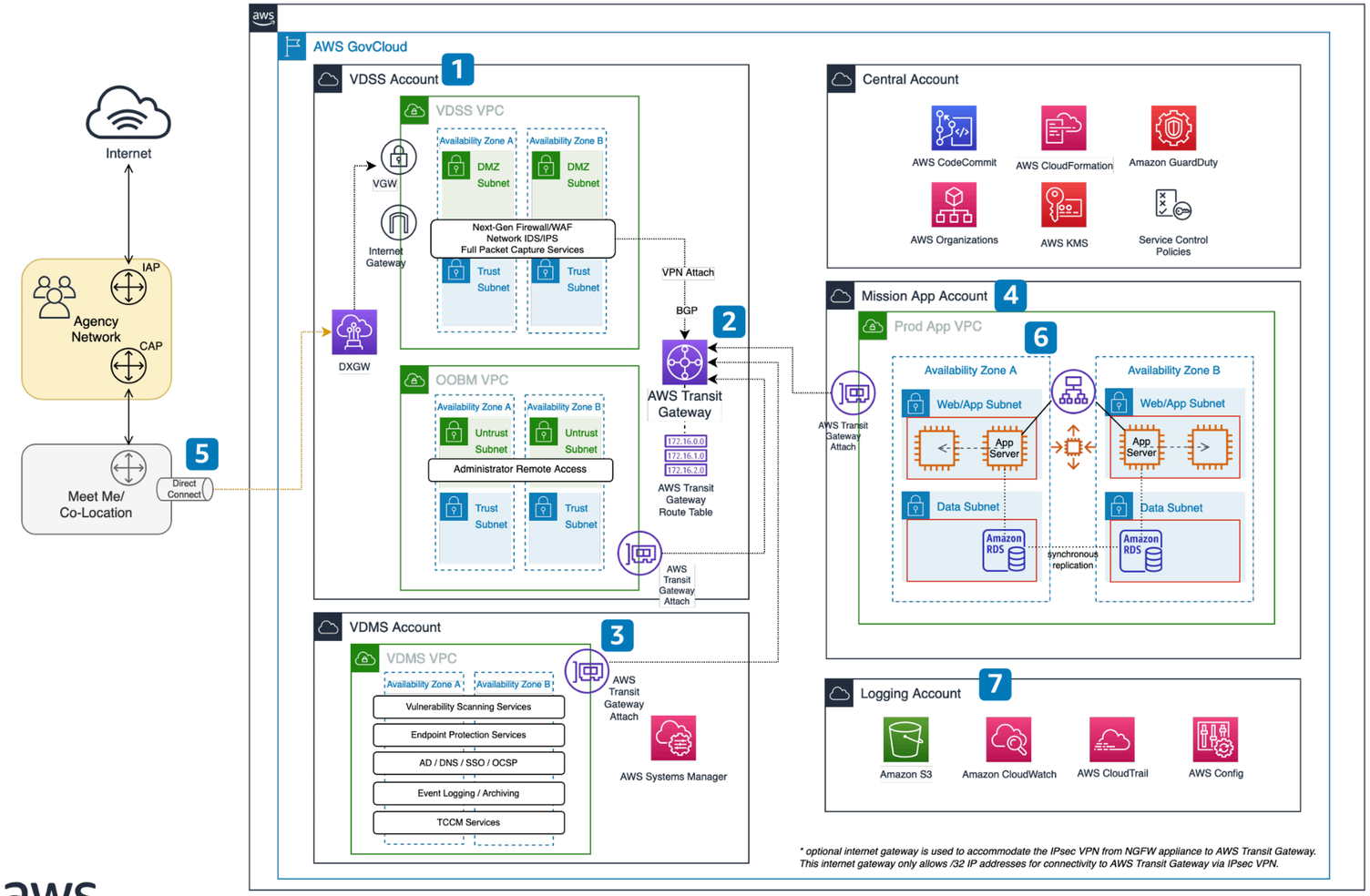
What is happening here?



Secure Cloud Computing Architecture (SCCA) on AWS GovCloud (US)

Build a Defense Information Systems Agency (DISA) Compliant Landing Zone on AWS

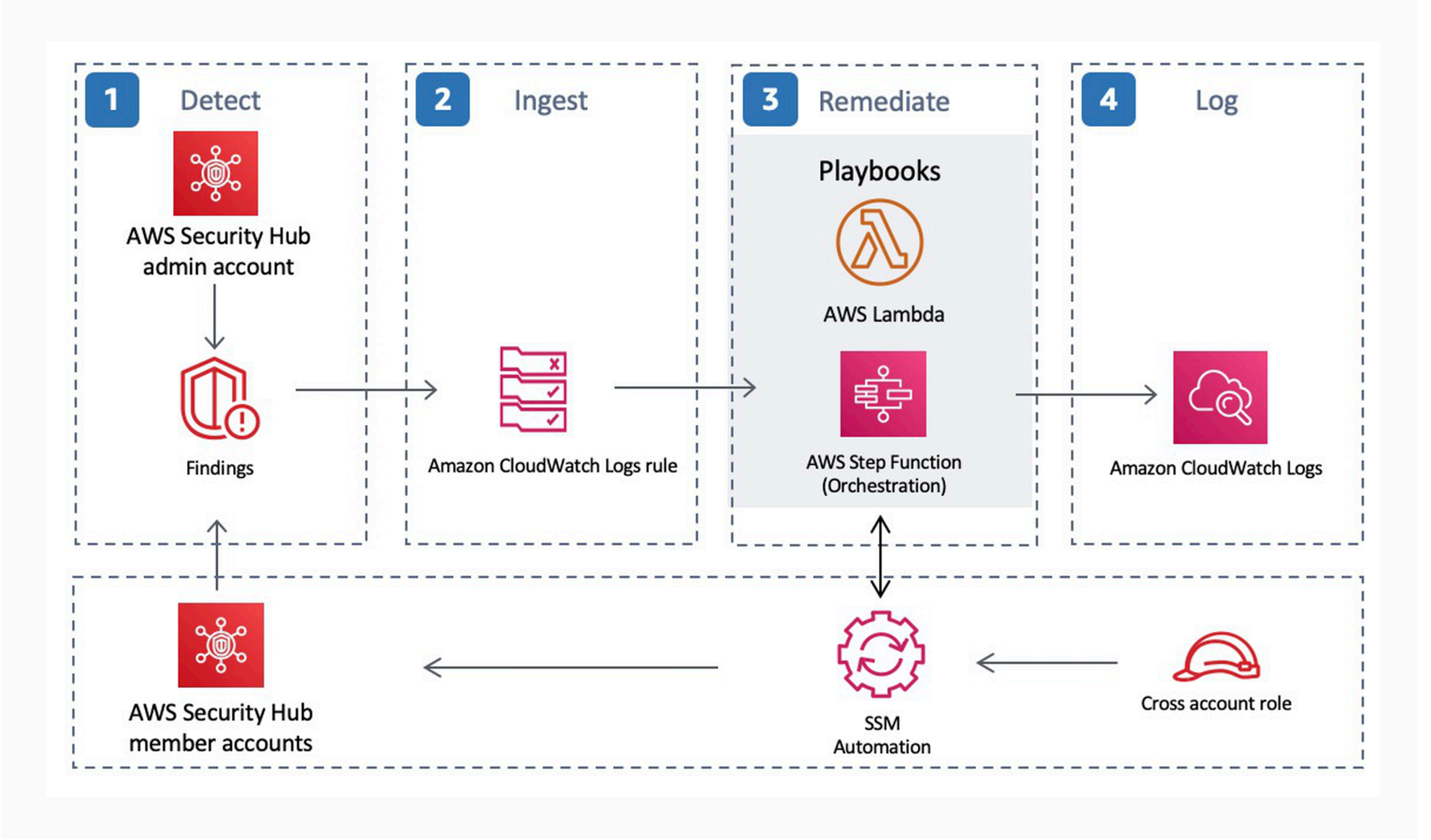
This architecture provides the prerequisite framework for securely running U.S. Department of Defense workloads and storing Impact Level 4 and 5 data on AWS GovCloud (US) Regions, according to DISA standards.



1. Virtual Data Center Security Stack (VDSS) Account acts as boundary used for protection of mission owner applications.
2. **AWS Transit Gateway** acts as a hub that controls how traffic is routed among all the connected networks which act like spokes.
3. Virtual Data Center Management Stack (VDMS) Account includes capabilities such as Host Based Security System (HBSS), Assured Compliance Assessment Solution (ACAS), authentication systems, and other common services.
4. Mission App Account is where core workloads are deployed. All communications to and from the Mission App VPC transit the VDSS and consume shared services from the VDMS.
5. Connectivity to the Department of Defense Information Network (or other agency networks) is achieved through a Virtual Private Gateway (VGW).
6. Typical multi-tier mission workloads use **Elastic Load Balancing**, **AWS Auto Scaling Groups** and multiple **Availability Zones** for high availability and scalability.
7. The Logging Account represents the immutable location where logs are aggregated and stored.

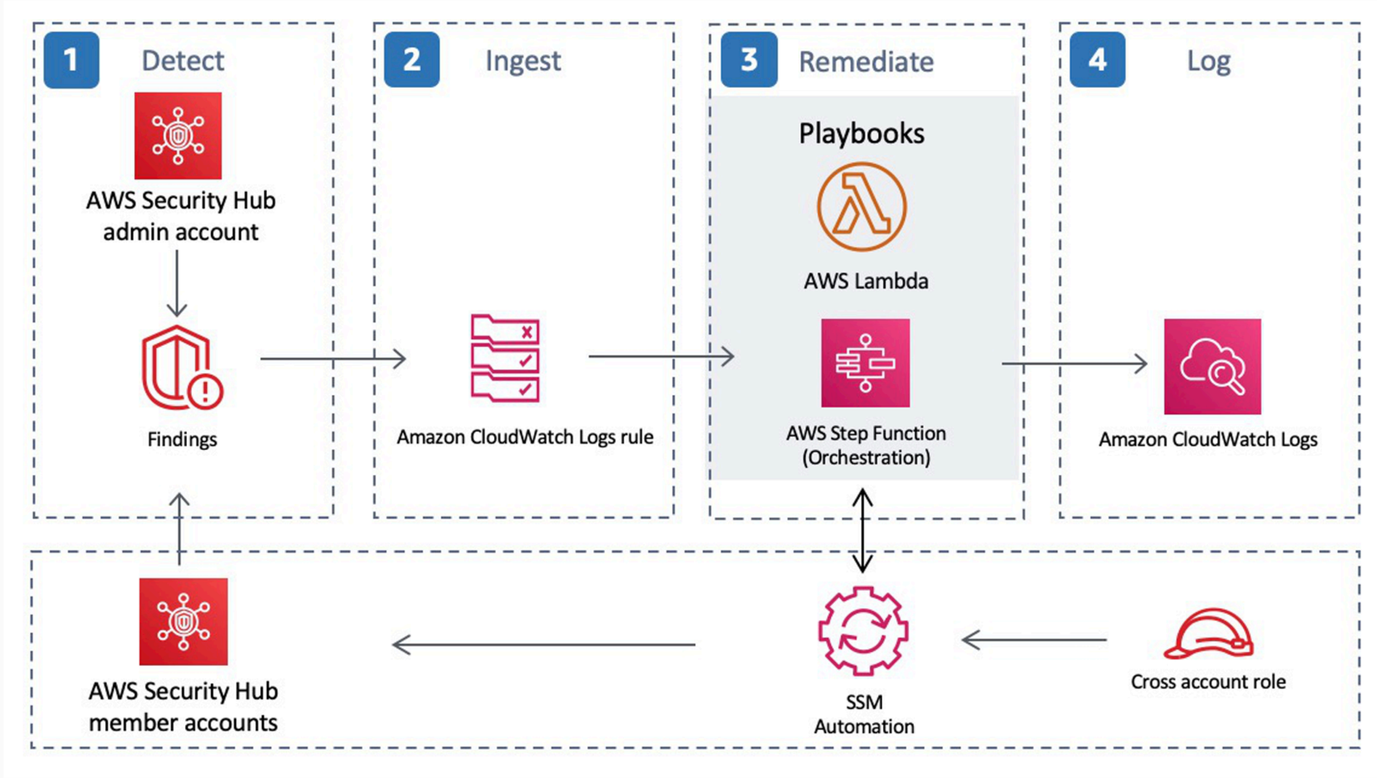
AWS Reference Architecture

What is happening here?



AWS Security Hub Automated Response and Remediation

This solution helps AWS Security Hub customers to resolve common security findings and to improve their security posture in AWS. To deploy this solution using the available AWS CloudFormation template, select **Deploy with AWS**.



1. **Detect**: AWS Security Hub provides customers with a comprehensive view of their AWS security state. It helps them to measure their environment against security industry standards and best practices. It works by collecting events and data from other AWS services, such as AWS Config, Amazon Guard Duty, and AWS Firewall Manager. These events and data are analyzed against security standards, such as CIS AWS Foundations Benchmark. Exceptions are asserted as findings in the AWS Security Hub console. New findings are sent as Amazon CloudWatch Events.

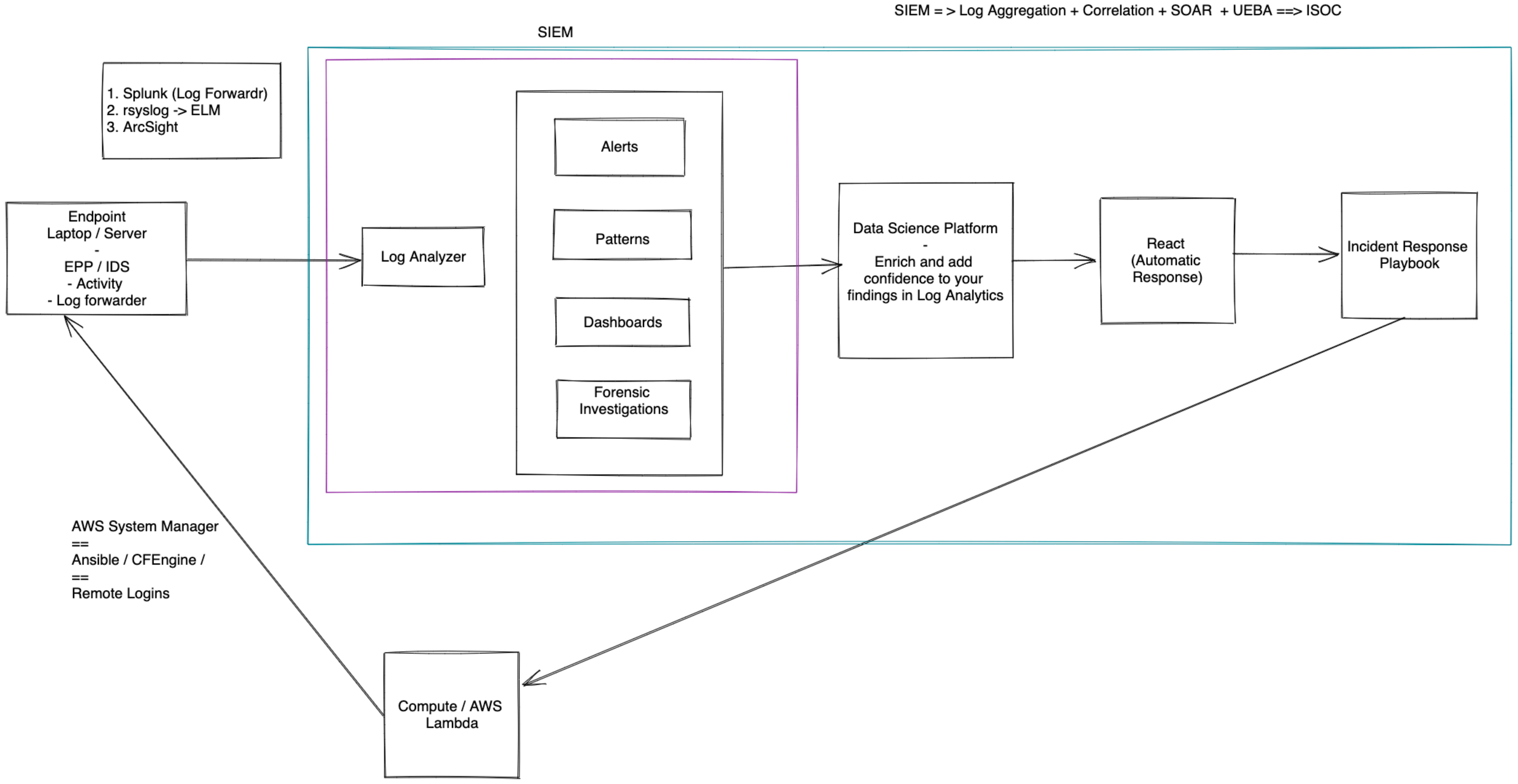
2. **Ingest**: AWS Security Hub Custom Actions and Amazon CloudWatch Events rules initiate Security Hub Automated Response and Remediation playbooks to address findings. Two CloudWatch Event Rules are deployed for each supported control by the solution: one rule to match the custom action event (user-initiated remediation), and one rule (disabled by default) to match the real-time finding event. Customers can use the Security Hub Custom Action menu to initiate automated remediation, or after careful testing in a non-production environment, they can enable automatic triggering for automated remediation. This decision can be made per remediation—it is not necessary to enable automatic triggers on all remediations.

3. **Remediate**: Using cross-account AWS Identity and Access Management (IAM) roles, the automated remediation uses the AWS API to perform the tasks needed to remediate findings. All playbooks in this solution call AWS Lambda functions. Some Lambda functions perform remediation directly. Others use AWS Systems Manager automation documents.

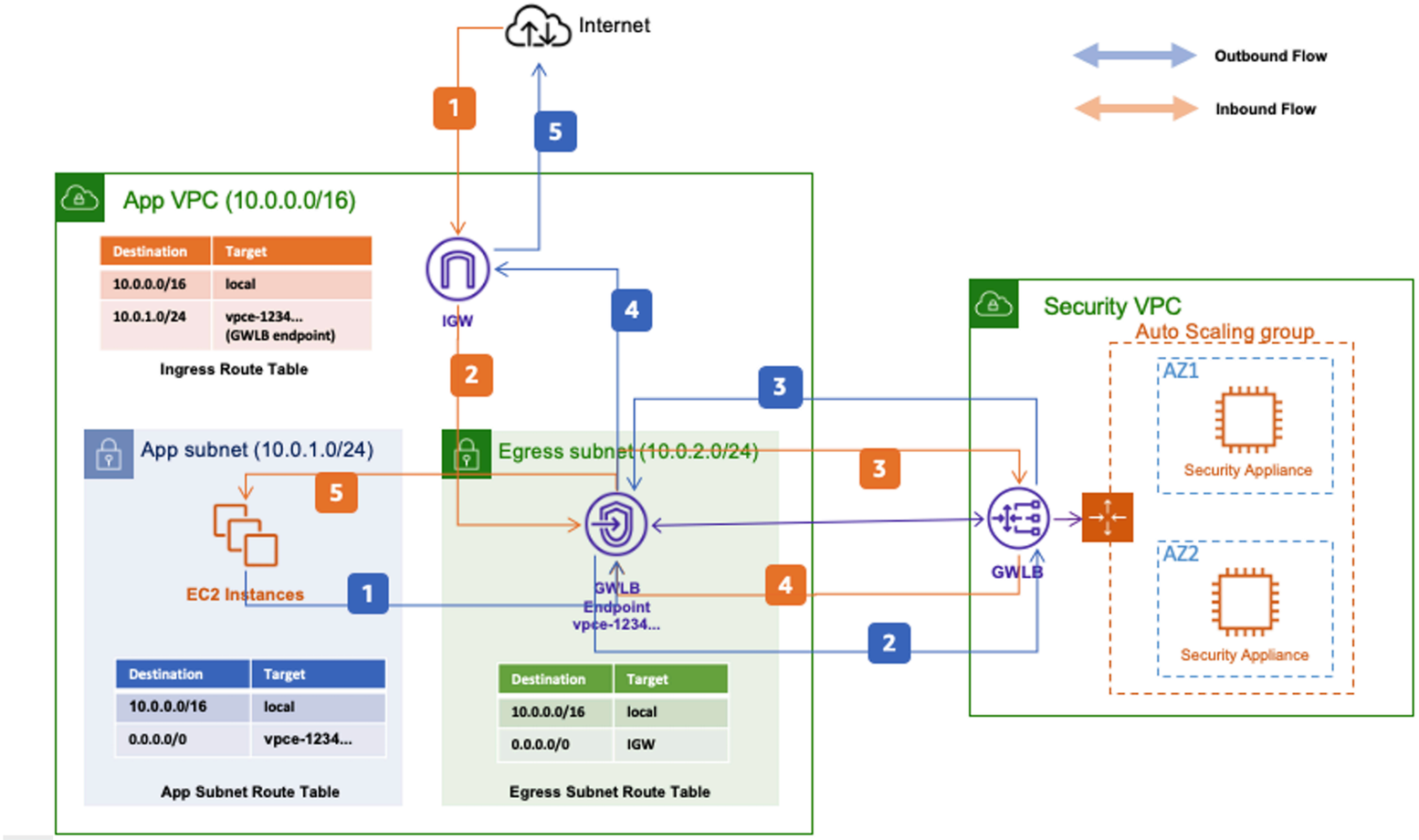
4. **Log**: The playbook logs the results to the Amazon CloudWatch Logs group for the solution, sends a notification to an Amazon Simple Notification Service (Amazon SNS) topic, and updates the Security Hub finding. An audit trail of actions taken is maintained in the finding notes. On the Security Hub dashboard, the finding workflow status is changed from NEW to either NOTIFIED or RESOLVED on the Security Hub dashboard. The security finding notes are updated to reflect the remediation performed.

Deploy with AWS

aws **Deployable AWS Reference Implementation**

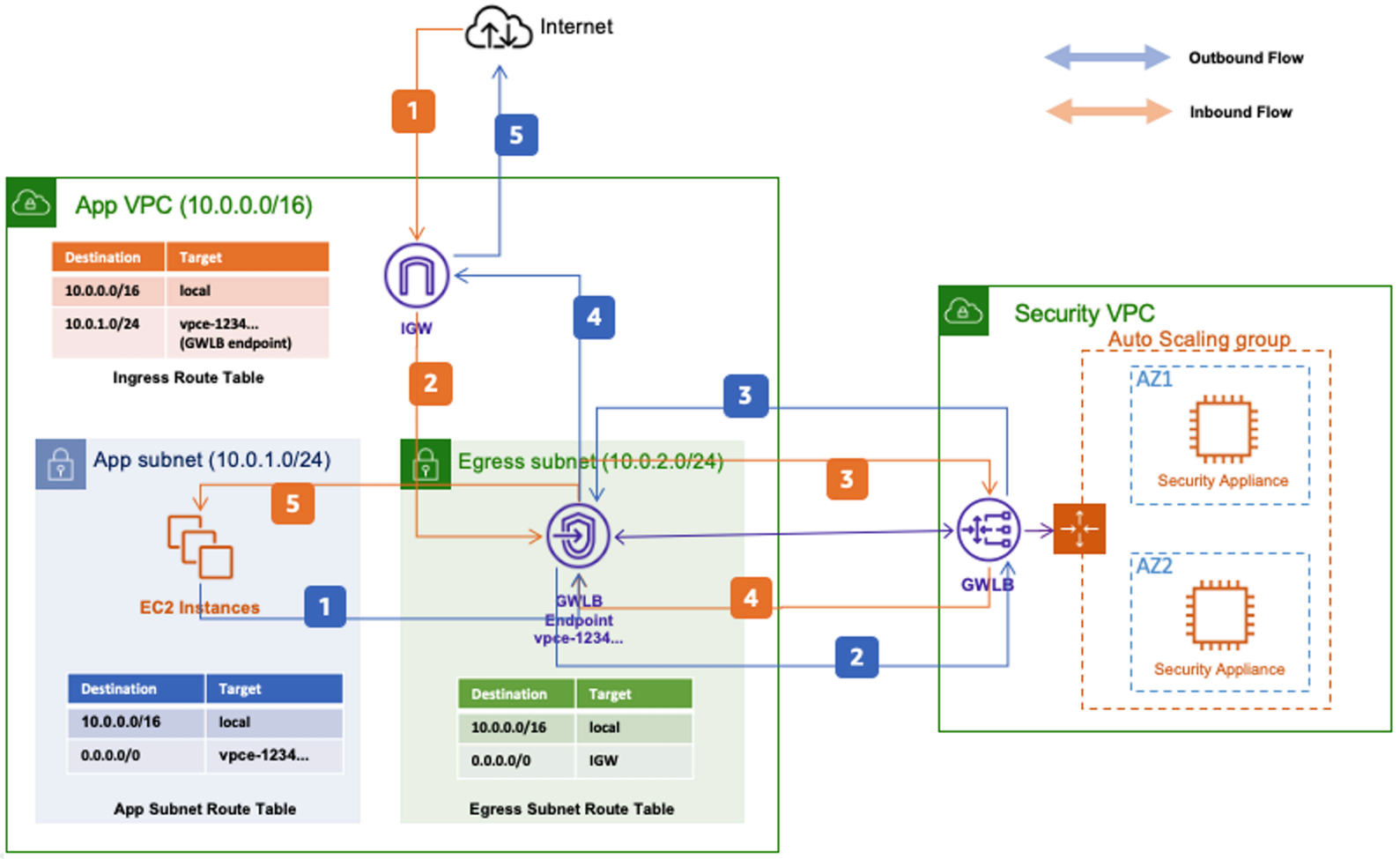

SIEM = > Log Aggregation + Correlation + SOAR + UEBA ==> ISOC
SIEM
1. Splunk (Log Forwardr)
2. rsyslog -> ELM
3. ArcSight
Endpoint
Laptop / Server
-
EPP / IDS
- Activity
- Log forwarder
Log Analyzer
Alerts
Patterns
Dashboards
Forensic Investigations
Data Science Platform
-
Enrich and add confidence to your findings in Log Analytics
React
(Automatic Response)
Incident Response Playbook
AWS System Manager
==
Ansible / CFEngine /
==
Remote Logins
Compute / AWS Lambda

What is happening here?



Architecture for Gateway Load Balancer – North/South Inspection

Use Gateway Load Balancer to create a highly available and scalable bump-in-the-wire solution for North/South inspection.



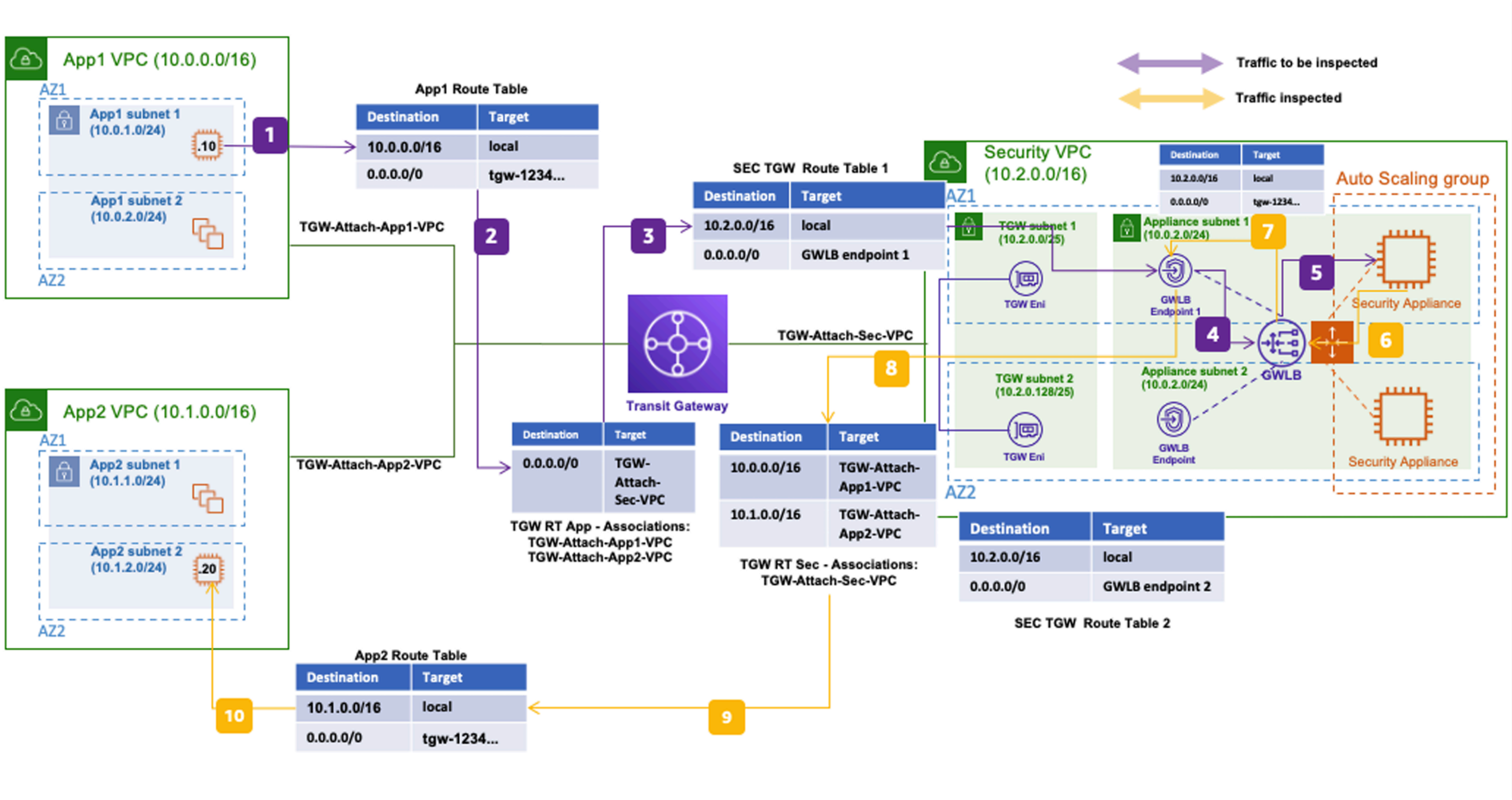
1. Traffic from resources in the APP subnet to the Internet is routed to the **Gateway Load Balancer** (GWLB) endpoint in the same virtual private cloud (VPC).
2. The **GWLB** endpoint is attached to the endpoint service for the **GWLB** in the Security VPC. Once **GWLB** receives the traffic, it forwards it encapsulated to the backend appliances.
3. Once the traffic is inspected by the appliances, the traffic returns to **GWLB** and then to the **GWLB** endpoint.
4. Once the traffic is back to the origin VPC, it follows the Egress Subnet Route Table and is sent to the internet gateway (IGW).
5. IGW sends the traffic to the internet.

1. Traffic coming from the Internet arrives at the IGW
2. Following the Ingress Route Table, traffic is routed to the **GWLB** endpoint.
3. **GWLB** endpoint is attached to the endpoint service for the **GWLB** in the Security VPC. Once **GWLB** receives the traffic, it forwards it encapsulated to the backend appliances.
4. Once the traffic is inspected by the appliances, it returns to **GWLB** and then to the **GWLB** endpoint.
5. Traffic arrives in the App VPC from the **GWLB** endpoint is locally routed to the resources in the App subnet.

AWS Reference Architecture

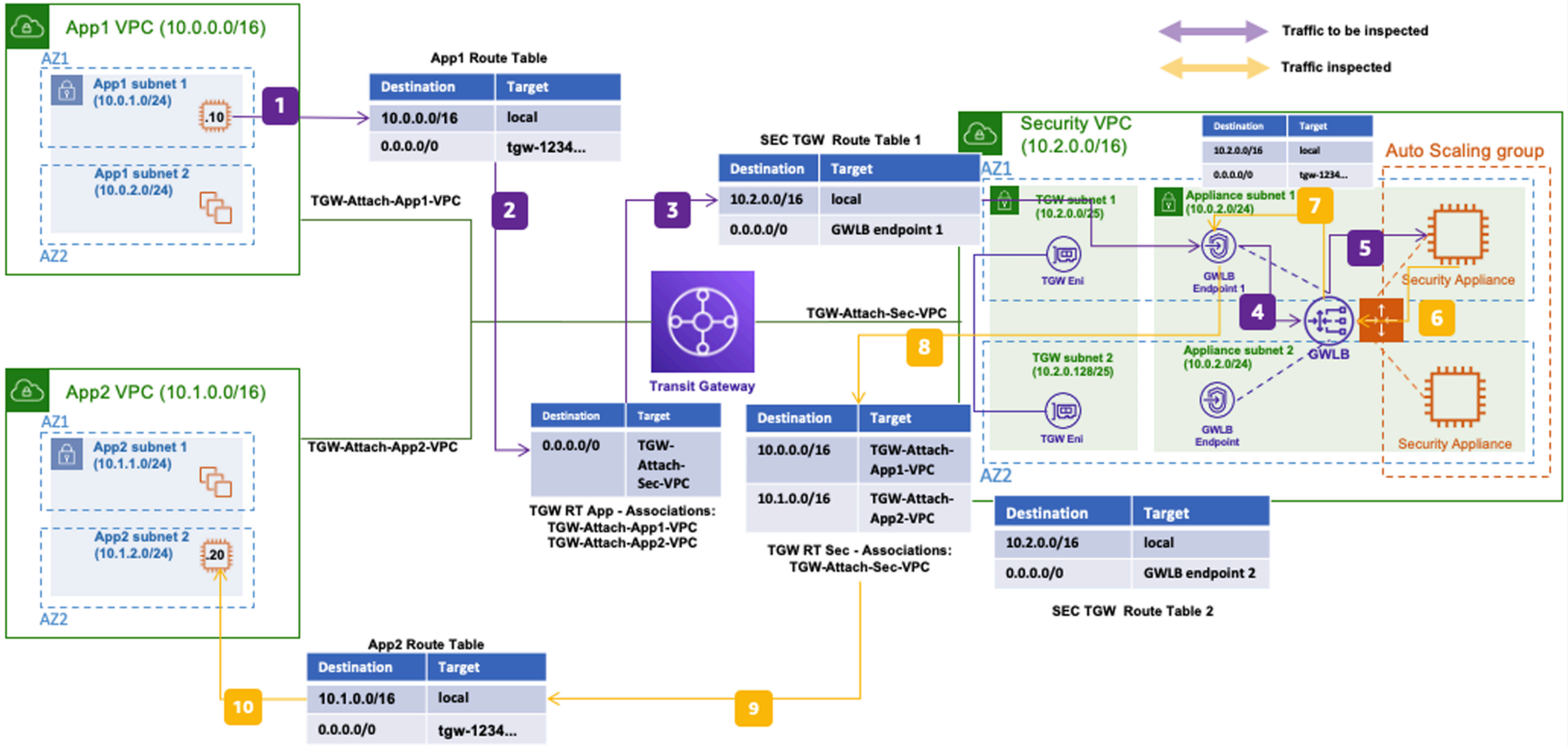
See also: Architecture for Gateway Load Balancer – East/West Inspection

What is happening here?



Architecture for Gateway Load Balancer – East/West Inspection

Use Gateway Load Balancer and Transit Gateway to create a highly available and scalable bump-in-the-wire solution for East/West inspection.

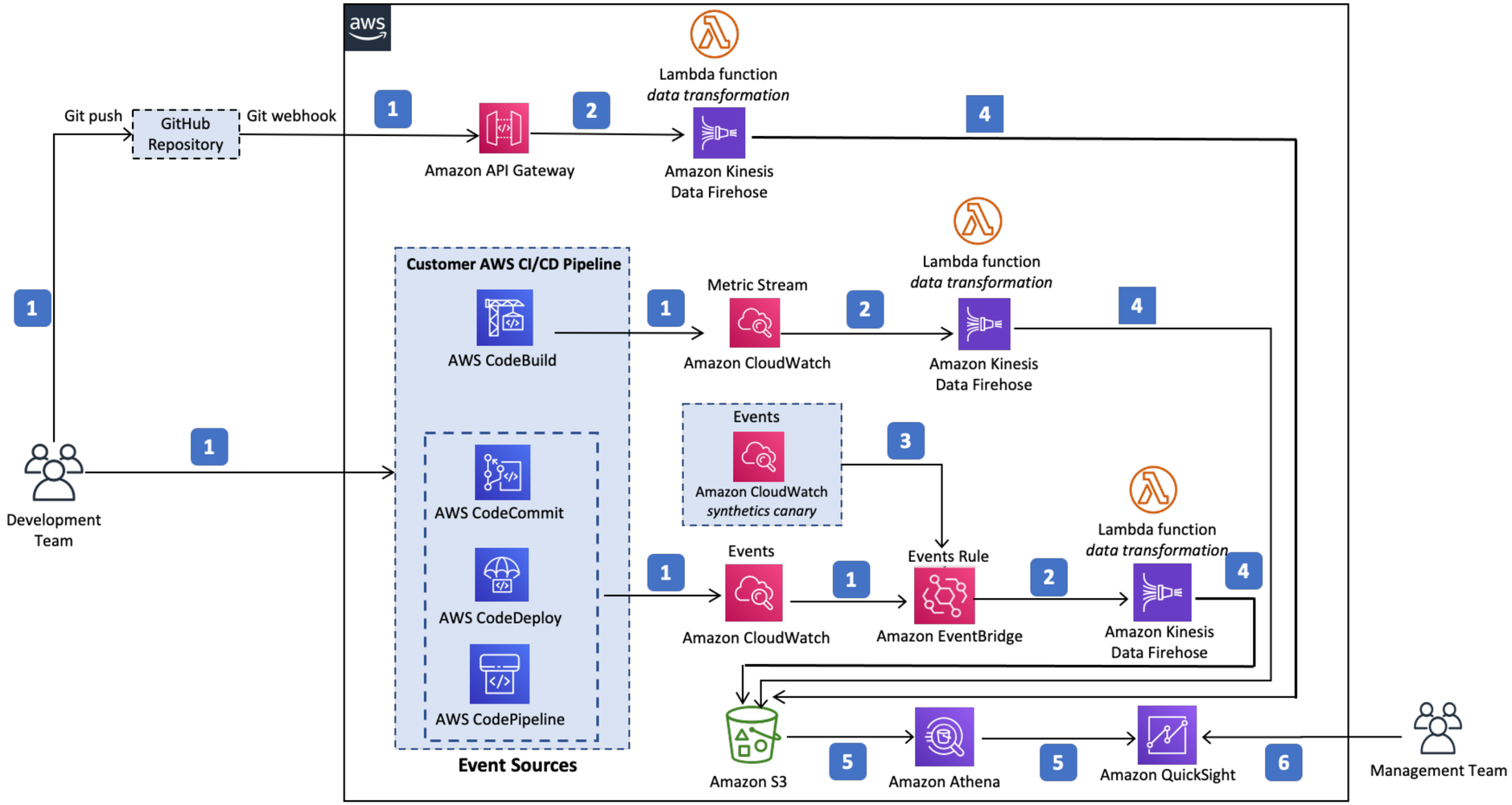


1. Traffic from IP 10.0.1.10 wants to reach IP 10.1.2.20 in the App2 virtual private cloud (VPC). The subnet's route table routes it to the TGW via the default route (0.0.0.0/0).
2. App1 VPC is associated with the "TGW RT APP" route table in the AWS Transit Gateway (TGW), which forwards all traffic (0.0.0.0/0) via the Security VPC (Virtual Private Cloud) attachment.
3. The TGW ENI in the Security VPC uses its subnet's route table to forward all traffic to the **Gateway Load Balancer** (**GWLB**) endpoint 1.
4. **GWLB** endpoint forwards the traffic to **GWLB**.
5. Traffic is send for inspection to one of the security appliance instances behind the **GWLB**.
6. Once the traffic is inspected, it is sent back to **GWLB**.
7. **GWLB** forwards the traffic to the **GWLB** endpoint.
8. The **GLWB** endpoint uses its subnet route table to forward all non-local traffic to the TGW via the TGW ENI using the TGW attachment.
9. Once the traffic reaches TGW, it uses the "TGW RT Sec" route table that it is associated with Security VPC to find the destination via the App2 VPC attachment.
10. Once it reaches the App2 Route table, the destination of the packet (10.1.2.20) is a local address, and it is forwarded to the destination instance.

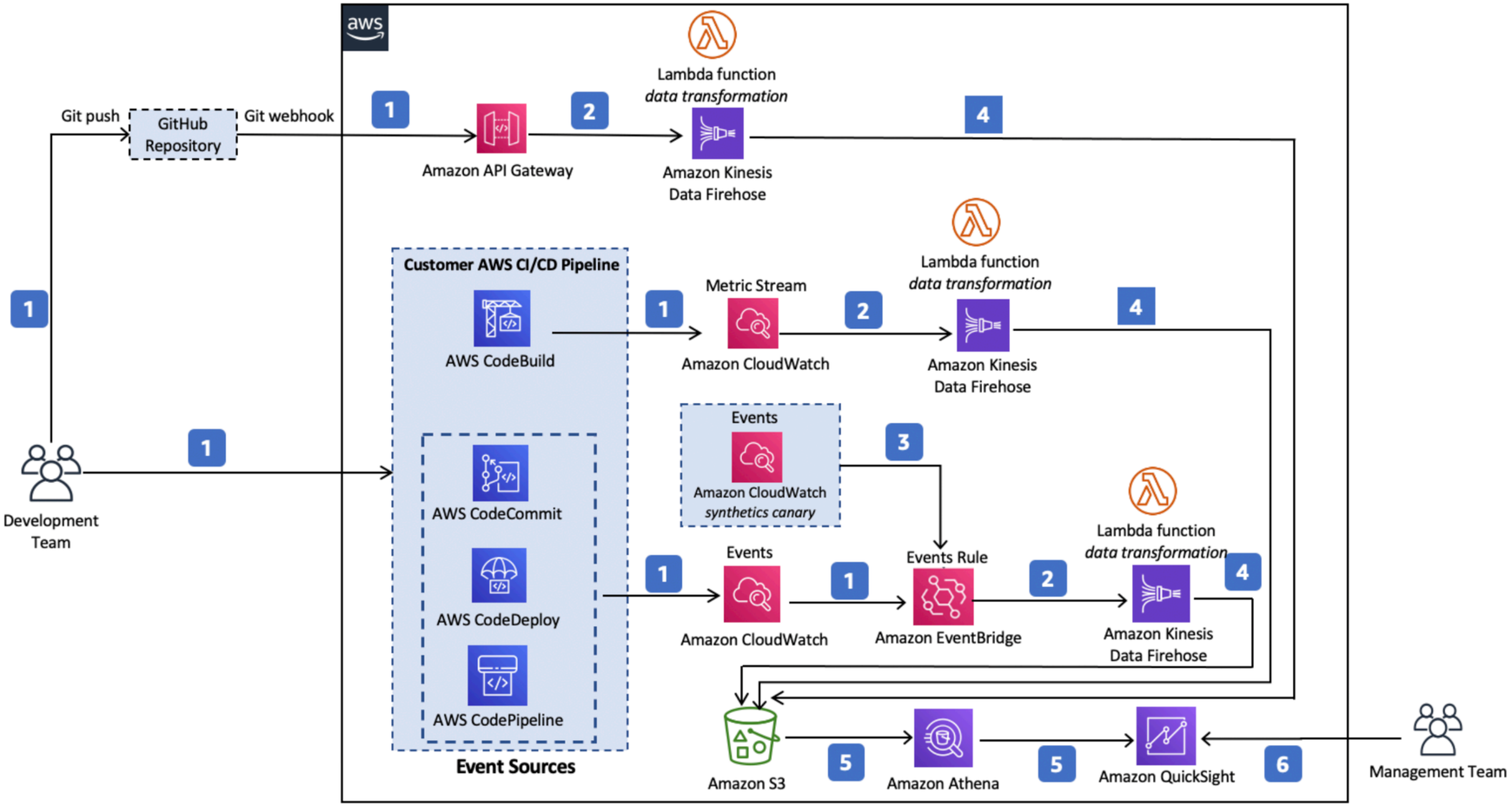
See also: [Architecture for Gateway Load Balancer – North/South Inspection]

AWS Reference Architecture

What is happening here?

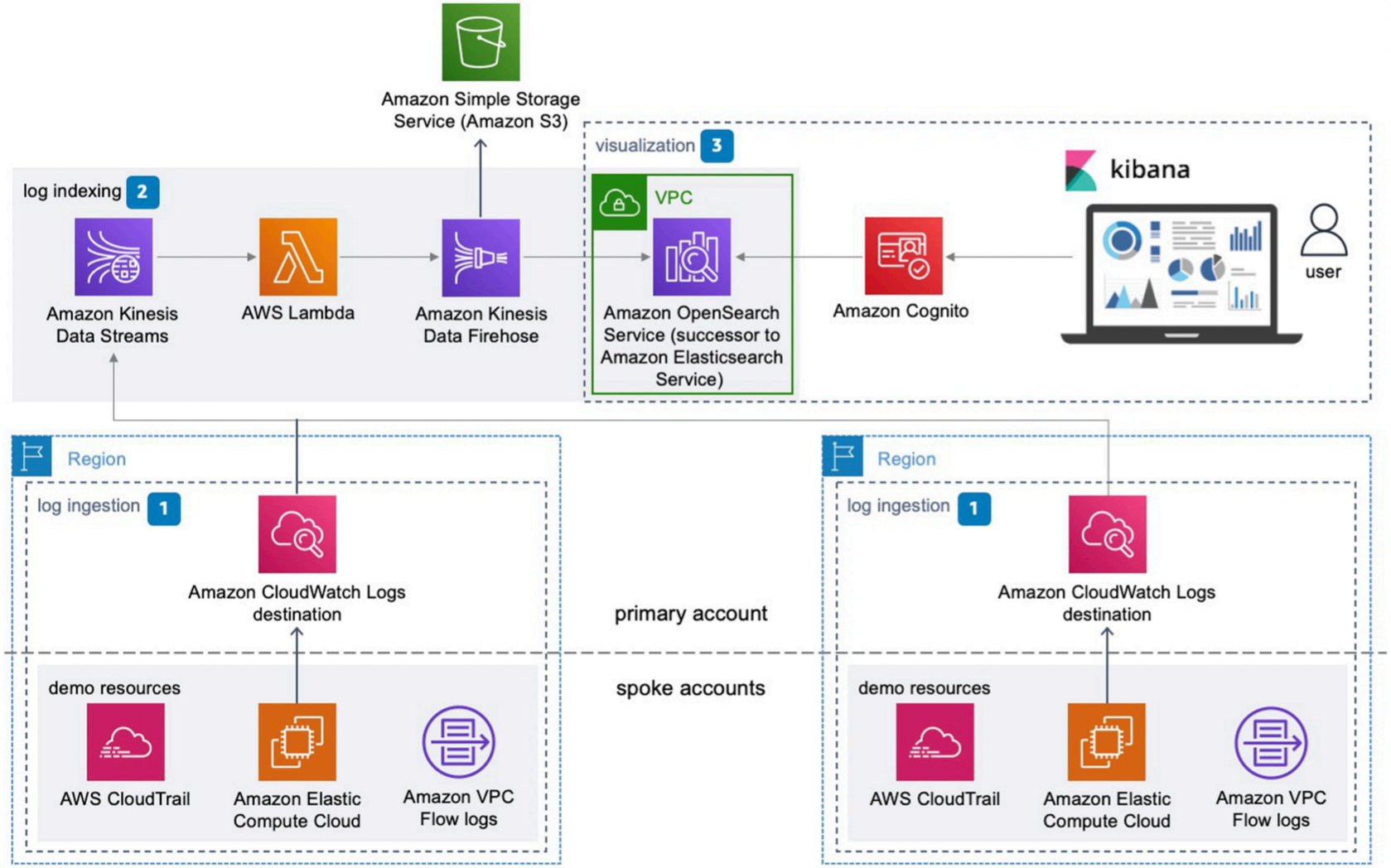


AWS DevOps Monitoring Dashboard



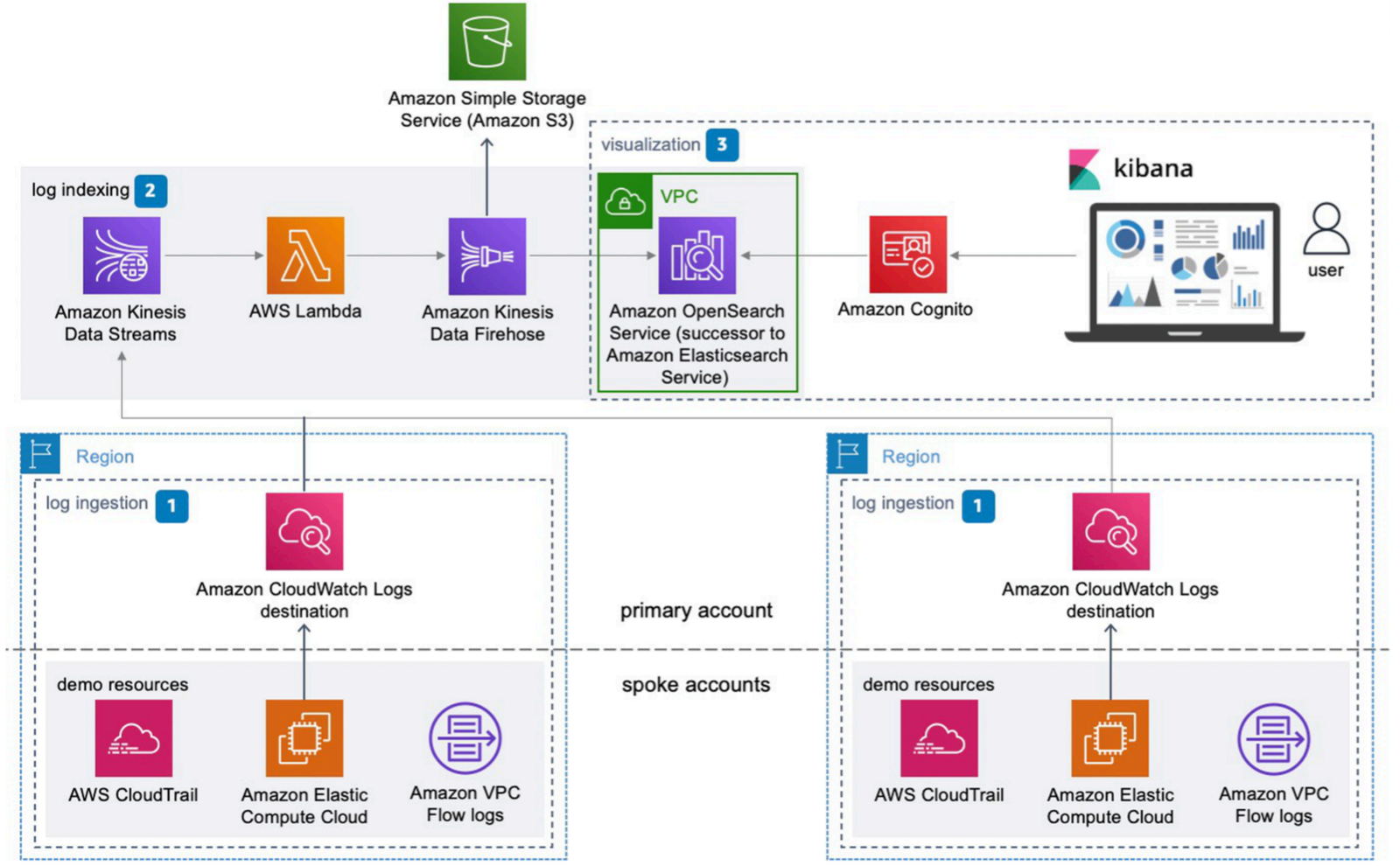
1. An **Amazon EventBridge** events rule detects the events based on predefined event patterns and then sends the event data to an **Amazon Kinesis Data Firehose** delivery stream. One event rule is created per event source. For activities in **AWS CodeBuild**, a **CloudWatch** metric stream is set up to capture **CloudWatch** metrics and deliver them to a **Kinesis Data Firehose** delivery stream. For GitHub push events, an Amazon API endpoint is created to post these events and deliver them to a **Kinesis Data Firehose** delivery stream.
2. An **Amazon EventBridge** events rule is also created to capture events from an **Amazon CloudWatch** alarm that monitors the status of an **CloudWatch** synthetics canary, if you have set up the canary and alarm in your account. This alarm is needed to gather data for calculating Mean Time to Recovery (MTTR) metrics.
3. **Kinesis Data Firehose** uses an **Lambda** function for data transformation. The **Lambda** function extracts relevant data to each metric and sends it to an **Amazon S3** bucket for downstream processing.
4. The data in **Amazon S3** is linked to an **Amazon Athena** database, which runs queries against this data and returns query results to **Amazon QuickSight**.
5. **Amazon QuickSight** obtains the query results and builds dashboard visualizations for your management team.

What is happening here?



AWS Centralized Logging

This solution helps organizations collect, analyze, and display log files from various sources in a single dashboard.
To deploy this solution using the available AWS CloudFormation template, select **Deploy with AWS**.



1. **Amazon CloudWatch Logs** destinations in your primary account for log streaming.
2. (Optional) Sample **CloudWatch Logs** for **AWS CloudTrail**, **Amazon Virtual Private Cloud** (Amazon VPC) flow logs, and an **Amazon Elastic Compute Cloud** (Amazon EC2) web server.
3. **Amazon Kinesis Data Streams** to index log events.
4. **AWS Lambda** to transform each log event.
5. **Kinesis Data Firehose** to index the documents.
6. **Amazon S3** for low-cost log records storage.
7. **Amazon OpenSearch Service** (successor to Amazon Elasticsearch Service) domain for log indexing and data visualization.
8. **Amazon VPC** for the Amazon ES domain to prevent public access to the Kibana dashboard.
9. **Amazon Cognito** for authentication and authorization to access the Kibana dashboard.

Deploy with AWS

Reviewed for technical accuracy September 9, 2021

Deployable AWS Reference Implementation

Case #1: DoD Data Breach

https://hackerone.com/reports/998981

Case #1: DoD Data Breach

https://hackerone.com/reports/998981

- **Context**: A few years back US Department of Defense started using AWS services. They used AWS S3 as a static web server to serve many production and administration infrastructure documents. A government penetration testing team confirmed that they could not access anything but what was being served.

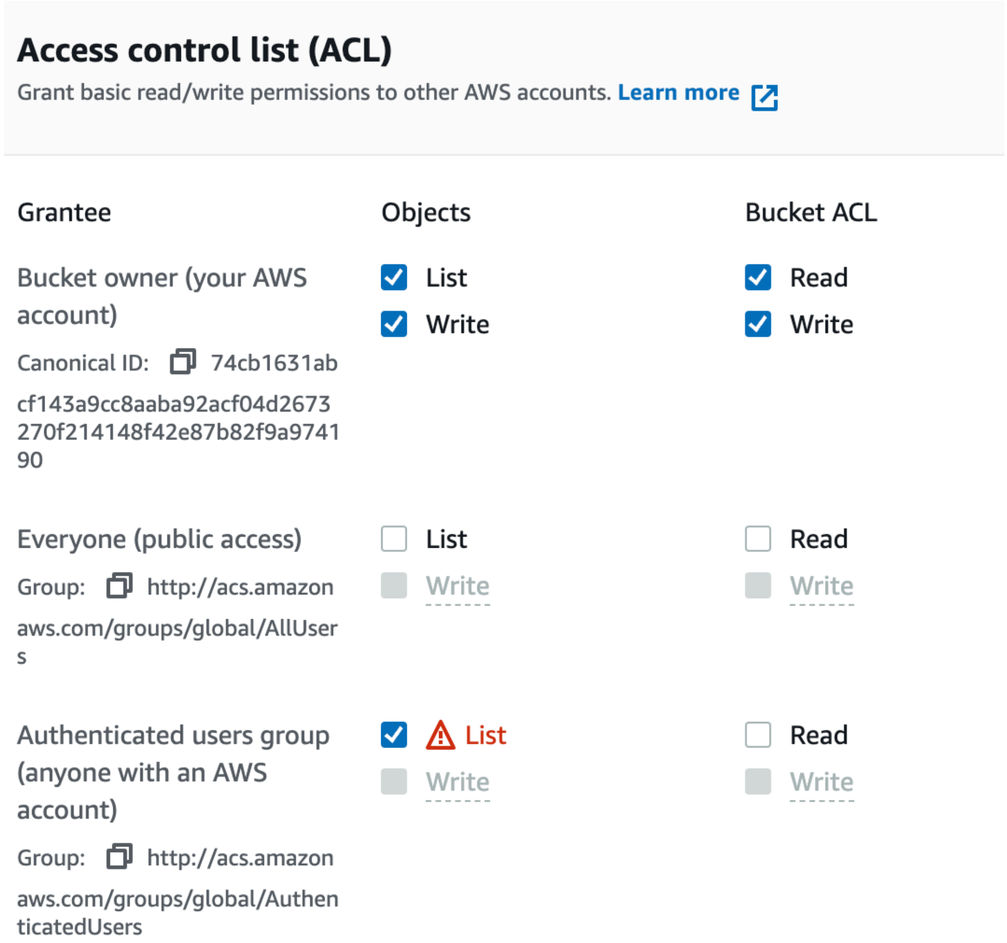
Case #1: DoD Data Breach

https://hackerone.com/reports/998981

How would you **fix** the issue?

Case #1: DoD Data Breach

https://hackerone.com/reports/998981



Access control list (ACL)

Grant basic read/write permissions to other AWS accounts. Learn more

Grantee	Objects	Bucket ACL
Bucket owner (your AWS account) Canonical ID: 74cb1631abcf143a9cc8aaba92acf04d2673270f214148f42e87b82f9a974190	☑ List ☑ Write	☑ Read ☑ Write
Everyone (public access) Group: http://acs.amazonaws.com/groups/global/AllUsers	☐ List ☐ Write	☐ Read ☐ Write
Authenticated users group (anyone with an AWS account) Group: http://acs.amazonaws.com/groups/global/AuthenticatedUsers	☑ ⚠ List ☐ Write	☐ Read ☐ Write

Case #2: RSA Data Breach

https://bits.blogs.nytimes.com/2011/04/02/the-rsa-hack-how-they-did-it/

Case #2: RSA Data Breach

https://bits.blogs.nytimes.com/2011/04/02/the-rsa-hack-how-they-did-it/

- **Context**: In 2011, security vendor RSA was compromised using an advanced persistent threat compromising sensitive data related to their flagship SecureID product and implementation details of high-profile clients around the world such as the U. S. Government and Defense contractors.

<story time about APT and the actual incident>

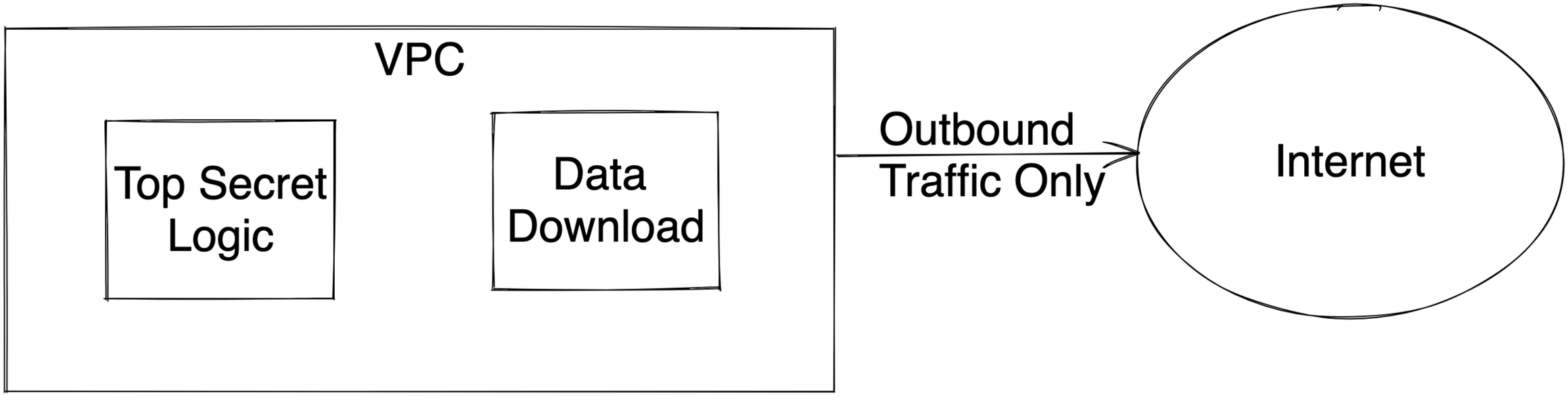
Case #2: RSA Data Breach

https://bits.blogs.nytimes.com/2011/04/02/the-rsa-hack-how-they-did-it/

- **Scenario**: You have two EC2 hosts in your VPC. One server hosts a top-secret program and the second host is for you to download data from the Internet. You ensured the ScoutSuite report is clean for zero inbound access from the Internet.

Case #2: RSA Data Breach

https://bits.blogs.nytimes.com/2011/04/02/the-rsa-hack-how-they-did-it/



Case #2: RSA Data Breach

https://bits.blogs.nytimes.com/2011/04/02/the-rsa-hack-how-they-did-it/

- **Issue**: You were browsing a website for critical information, and your "data download" system got infected by "drive-by malware" (a malware that gets installed on your system without your consent). After reconnaissance, the malware moved laterally and started exfiltrating data from the top-secret system to the Internet. A C2 (command and control) system orchestrates all the malicious activity.

<Story time about types of DNS domains>

Case #2: RSA Data Breach

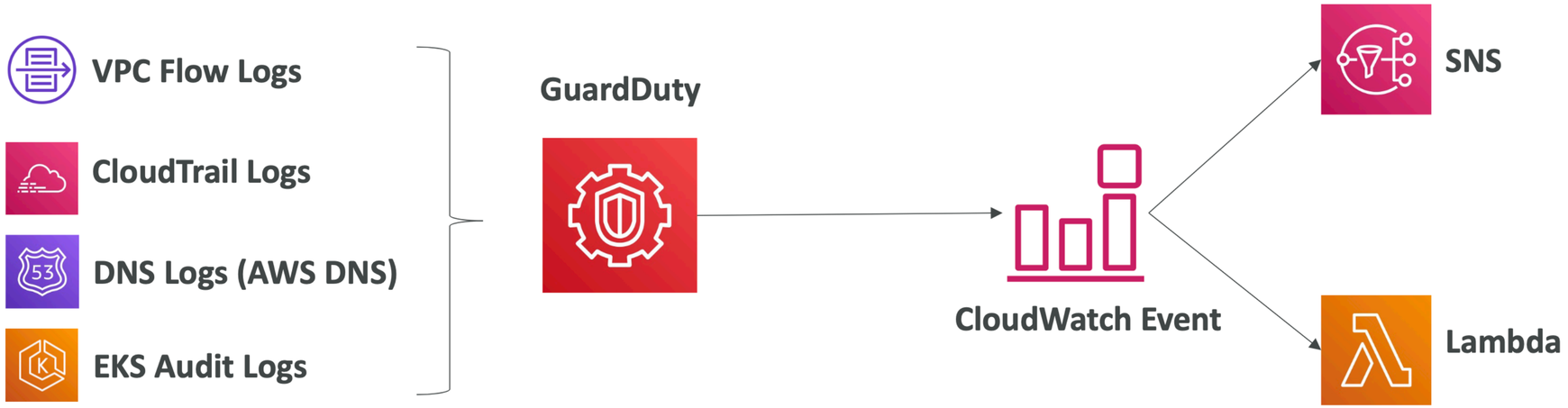
https://bits.blogs.nytimes.com/2011/04/02/the-rsa-hack-how-they-did-it/

How would you **fix** the issue?

AWS GuardDuty

Threat Intelligence & Machine Learning based Security

- Intelligent Threat discovery to Protect AWS Account
- Uses Machine Learning algorithms, anomaly detection, 3rd party data
- Integrates with CloudWatch Event rules for alerting



Case #3: Denial of Service Attack

COVID-19 Vaccine Registration Website

Case #3: Denial of Service Attack

COVID-19 Vaccine Registration Website

- **Context**: In 2021, a local pharmacy in New York announced a web service to register for the Covid-19 vaccine. They hosted their web server in AWS EC2 and DNS on AWS Route 53. They were leveraging Let's Encrypt SSL certificates to keep the costs low. There was widespread botnet malware floating around at that time. Consequentially, the new website went down with a denial of service attack. The IT team at the local pharmacy thought it was due to much traffic and migrated to an auto-scaling group with a load balancer. Despite the expensive architecture, the number of registrations was hardly significant.

Case #3: Denial of Service Attack

COVID-19 Vaccine Registration Website

- **Issue**: By observing the number of connections and tracing the source IP addresses, the IT team figured out it was a DoS attack.

Case #3: Denial of Service Attack

COVID-19 Vaccine Registration Website

How would you fix it?

AWS Shield

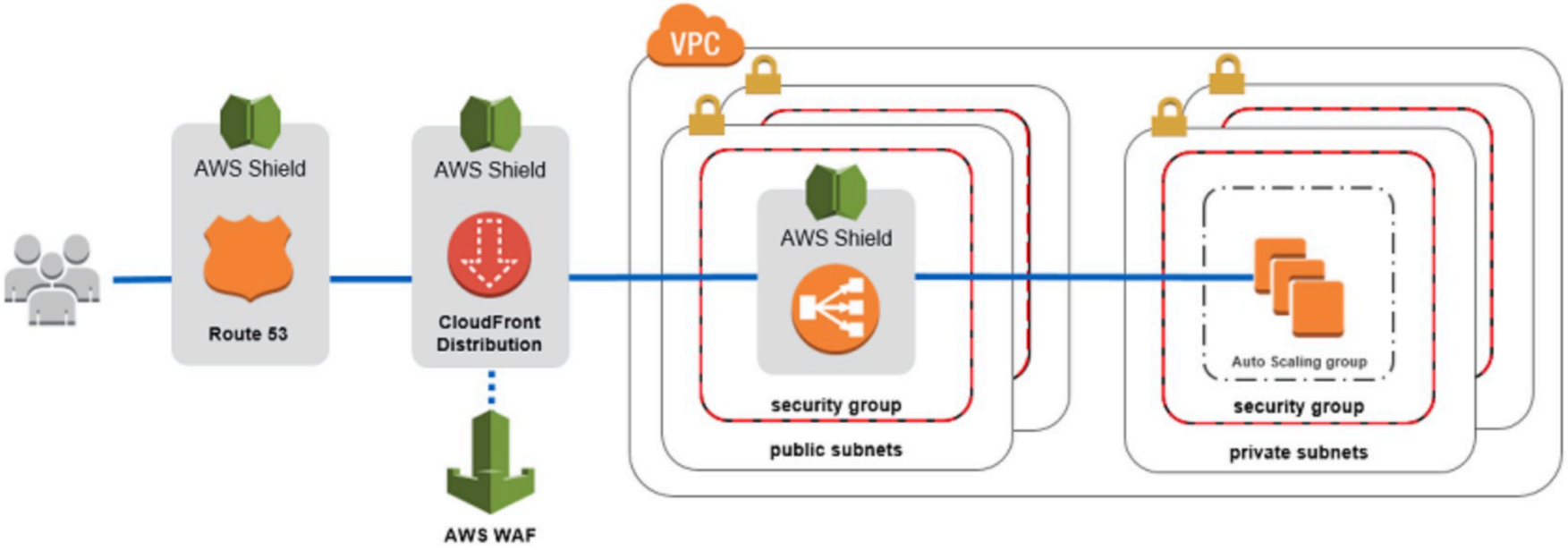
Denial of Service (DoS) Protection

- **AWS Shield Standard:**
  - Free service that is activated for every AWS customer
  - Provides protection from attacks such as SYN/UDP Floods, Reflection attacks, and other layers 3/layer 4 attacks

- **AWS Shield Advanced:**
  - Optional Distributed DoS mitigation service
  - Protect against more sophisticated attacks on Amazon EC2, Elastic Load Balancing (ELB), Amazon CloudFront, AWS Global Accelerator, and Route 53
  - 24/7 access to AWS DDoS response team (DRP)
  - Protect against higher fees during usage spikes due to DDoS

DDoS Protection

https://aws.amazon.com/answers/networking/aws-ddos-attack-mitigation/



Case #4: Capital One Data Breach

https://web.mit.edu/smadnick/www/wp/2020-16.pdf

Case #4: Capital One Incident

https://web.mit.edu/smadnick/www/wp/2020-16.pdf

Capital One

Responsible Disclosure (Shared) <responsibledisclosure@capitalone.com>

[External Sender] Leaked s3 data

[redacted] Wed, Jul 17, 2019 at 1:25 AM

To: "responsibledisclosure@capitalone.com" <responsibledisclosure@capitalone.com>

Hello there,

There appears to be some leaked s3 data of yours in someone's github / gist:

https://gist.github.com[redacted]

Let me know if you want help tracking them down.

Thanks,

[redacted]

Case #4: Capital One Data Breach

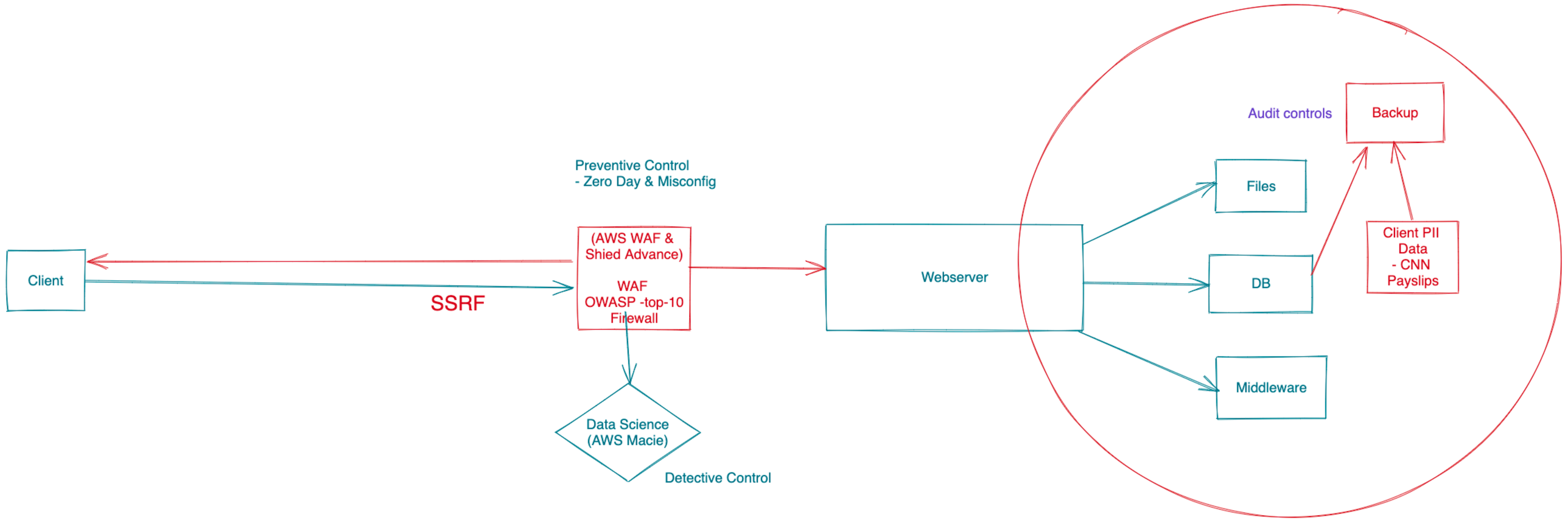
https://web.mit.edu/smadnick/www/wp/2020-16.pdf

- **Context**: “On July 19, 2019, we determined that an outside individual gained unauthorized access and obtained certain types of personal information from Capital One credit card customers and individuals (...).” The company claimed that compromised data corresponded to “personal information Capital One routinely collects at the time it receives credit card applications, including names, addresses, zip codes/postal codes, phone numbers, e-mail addresses, dates of birth, and self-reported income.” The unauthorized access “affected approximately 100 million individuals in the United States and approximately 6 million in Canada”.

Case #4: Capital One Data Breach

https://web.mit.edu/smadnick/www/wp/2020-16.pdf

- **Context**: After analyzing the records of the Seattle Court, cloud security company CloudSploit published an analysis of the incident in its corporate blog (CloudSploit, 2019), describing that the access to the vulnerable server was possible thanks to a Server-Side Request Forgery (SSRF) attack that was able to bypass the misconfigured Web Application Firewall (WAF) solution deployed by Capital One: *"An SSRF attack tricks a server into executing commands on behalf of a remote user, enabling the user to treat the server as a proxy for his or her requests and get access to non-public endpoints."*



1. Get in - Recon -- SSRF - Misconfig
2. Elevated access -> restrictive
3. Lateral Movement
4. Exfiltration

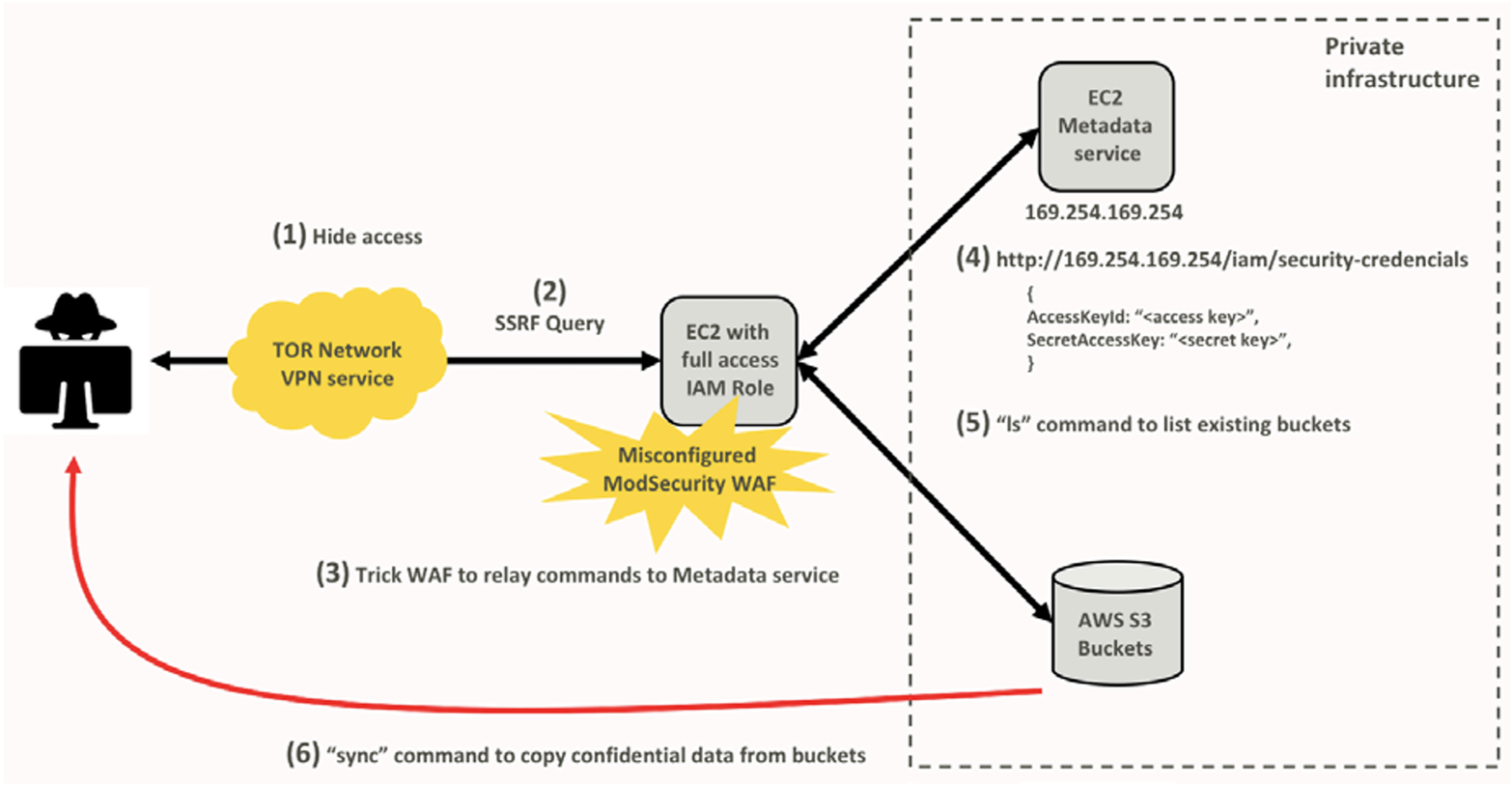
Case #4: Capital One Data Breach

https://web.mit.edu/smadnick/www/wp/2020-16.pdf

- **Server-Side Request Forgery**, (SSRF) is a software vulnerability class where servers can be tricked into connecting to another server it did not intend to, them making a request that's under the attacker's control (Abma, 2017), enabling an attacker to send crafted requests from the back-end server of a vulnerable web application (O'Donnell, 2019).
- **Modsecurity** is a popular open-source, host-based Web Application Firewall (WAF) solution.

Case #4: Capital One Data Breach

Attack Sequence



Case #4: Capital One Data Breach

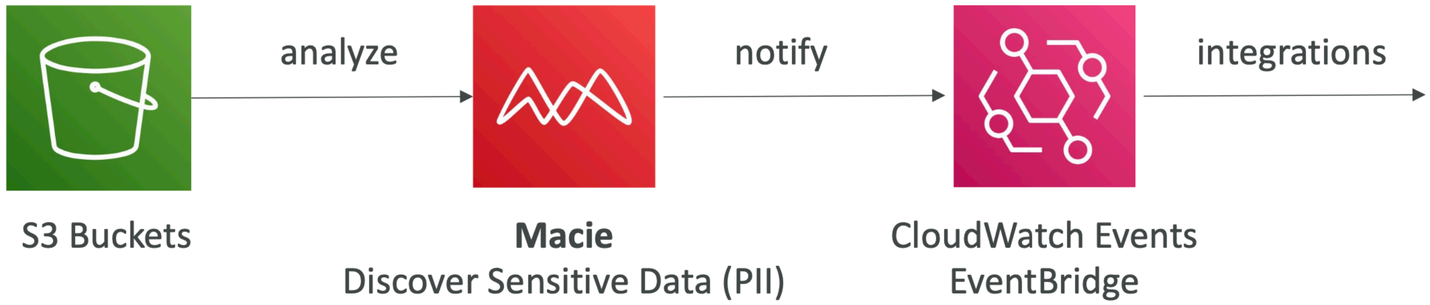
https://web.mit.edu/smadnick/www/wp/2020-16.pdf

How would you fix it?

AWS Macie

Data Security

- Amazon Macie is a fully managed data security and data privacy service that uses machine learning and pattern matching to discover and protect your sensitive data in AWS.
- Macie helps identify and alert you to sensitive data, such as personally identifiable information (PII)



Incident Report Walkthrough

Synthesize an Incident Report

Incident Report Walkthrough

Pick one report & Synthesize

- https://www.comparitech.com/blog/vpn-privacy/choice-hotels-data-leak/
- https://www.comparitech.com/blog/vpn-privacy/188-million-data-breach/
- https://www.comparitech.com/blog/vpn-privacy/quickbit-database-leak/
- https://www.comparitech.com/blog/information-security/utah-covid-test-center-leak/
- https://www.comparitech.com/blog/information-security/microsoft-customer-service-data-leak/
- https://www.comparitech.com/blog/information-security/267-million-phone-numbers-exposed-online/
- https://www.comparitech.com/blog/information-security/centurylink-data-leak/

Practice

Practice

- AWS VPC Account. - DONE
- AWS EC2 server - DONE
- Start Web Service - DONE
- Create DNS - DONE
- Expose it as a website (optional) - DONE
- Create Splunk server - DONE
- Create a Splunk dashboard - DONE
- Run ScoutSuite for auditing (optional) - Previous done
- Add Extremely Vulnerable Application — DONE
- Run ZAP Proxy Service - DONE
- Add files to S3 - DONE
- ===
- Add Splunk forwarder - https://www.splunk.com/en_us/download/universal-forwarder.html

Homework-3

- Design AWS-based application development environment with code and web security controls learned during the class. HW requires hands-on AWS practice, which we will go through in class.
    - Create an architecture diagram
    - Create the environment in the cloud
    - Run DVWA using docker - https://github.com/digininja/DVWA
    - Run Zap proxy
    - Stand-up Splunk server
    - Read logs directly from the file (or add Splunk forwarder for extra credit)
    - Create a dashboard in Splunk with newly read logs
    - Add a screenshots for the following
        1. Create a VPC
        2. Create a EC2 server
        3. DVWA Web site
        4. ZAP Proxy results
        5. Splunk Dashboard