MODULE 7 · STUDY NOTES

Responding to and Managing an Incident

Telling incidents from ordinary events, the two-phase response lifecycle, the AWS services that detect and remediate them, and the practices that keep a response fast

Securing resources in the cloud is a job that is never fully complete — which is why the bank in this course, having decided to migrate, must now review and update its incident response plan. This module is where all the monitoring and logging tools you have met converge into a plan for what happens *when something goes wrong*. Under the shared responsibility model the customer owns security **in** the cloud, so responding to incidents is squarely the customer's job. Incident response is a set of information security policies and procedures used to identify, contain, and eliminate cyberattacks; its aim is to detect and halt attacks quickly to minimize damage and prevent repeats. The work splits into two phases — a **discovery and recognition** phase that finds and categorizes the problem, and a **resolution and recovery** phase that isolates, fixes, and closes it. AWS supplies a toolbox for each phase: detective services (Trusted Advisor, CloudWatch, Inspector, GuardDuty, Shield, Config) and responsive services (Systems Manager, CloudFormation, SNS, Step Functions, Lambda). Tie the two together with event-driven automation so detection triggers response, and back the whole thing with industry best practices — prepared plans, pre-provisioned access, and rehearsed game days.

LEARNING OBJECTIVES

- Identify an incident
- Describe AWS services that are used for incident recognition and remediation
- Identify best practices for incident response

7.1 Identifying an incident

Incident response is a set of information security policies and procedures that you use to identify, contain, and eliminate cyberattacks. Its goal is to enable an organization to quickly detect and halt attacks, which helps to minimize damage and prevent future attacks of the same type. The first challenge is simply recognition: how would an enterprise know abuse is taking place, and how would it tell an abnormal event that merely needs attention from an incident that needs immediate analysis and remediation?

The key insight is that **not all events are incidents in need of immediate remedy.** An event is any observed occurrence in the environment; only some events rise to the level of an incident. Reading events correctly avoids both complacency and panic — recognizing a drive is failing, for example, lets you schedule an orderly replacement instead of a last-minute *hot swap* (removing or inserting components while the power is still on) after it fails.

EVENTS THAT ARE NOT NECESSARILY INCIDENTS

- **Logging in from a remote location** — the employee might be traveling or using an approved virtual private network (VPN).
- **A failing hard drive that is still fully operational** — knowing this lets an enterprise schedule a timely cycling of the drives without a panicked hot swap when it is too late.
- **An employee trying to access resources they shouldn't** — if the access was *denied* this might not be a breach, but it is still a behavioral insight worth monitoring.

Once you decide an occurrence is an incident, response proceeds in **two phases.** The first is the **discovery and recognition phase**, where incident *identification, logging, and categorization* take place. After an incident is identified — through user reports, solution analyses, or manual identification — it is logged so investigation and categorization can begin. A *notification*, set up by the user and initiated by specified alerts, is sent by email, SMS text, or mobile push. *Escalation* happens when an employee can't resolve an incident themselves and must hand it off to a more experienced or specialized colleague. Finally, *investigation and diagnosis* gathers answers and develops strategies to resolve any threats.

When an enterprise has identified a failed component, a reduction in service quality, or an exploit needing remedy, it moves into the **resolution and recovery phase.** This phase begins with *forensic isolation* — isolating the incident and performing a deep dive to discover the issue, often by capturing the disk image or as-is configuration of an operating system. If the problem is a bug in the code base, you must reproduce the error, because you cannot fix what you cannot reproduce. From there you *stage a fix* (reproduce the issue, apply a fix, and test), *deploy the fix* (push new infrastructure-as-code or application code to production), and finally reach *incident closure*. Automate these processes wherever possible, and run **game days** in staging (safe) environments to fine-tune your corporate-wide response process.

PHASE 1 – DISCOVERY & RECOGNITION	PHASE 2 – RESOLUTION & RECOVERY
find, log, and understand the problem ▪ Incident identification, logging, and categorization ▪ Incident notification and escalation ▪ Investigation and diagnosis	*isolate, fix, and close it out* ▪ Forensic isolation (if software, reproduce the bug) ▪ Stage a fix — reproduce, apply, and test ▪ Deploy the fix to production ▪ Incident closure; automate and run game days

KEY TERMS

Incident response
A set of information security policies and procedures used to identify, contain, and eliminate cyberattacks; the goal is to quickly detect and halt attacks.

Escalation
Handing an incident to a more experienced or specialized employee when the initial responder cannot resolve it.

Forensic isolation
Isolating an affected component and deep-diving to find the issue — often by capturing a disk image or the as-is OS configuration.

Game day
A simulated incident-response event, run in a safe (staging) environment, used to rehearse and fine-tune the response process.

COMMON PITFALL

Escalation is an *internal* handoff, not a customer notification. It happens specifically when an employee cannot resolve an incident on their own and needs to pass it to someone more experienced or specialized — do not confuse it with the notification/alert step, which is what tells responders an incident occurred in the first place.

7.2 AWS services that support the discovery and recognition phase

AWS offers several services that help an enterprise identify events that could lead to — or have already become — an incident. The module highlights six of them; note that these are not all of the AWS services available. Each supports finding and recognizing a problem so that the response can begin.

TABLE 7.1 Detective services for the discovery and recognition phase

SERVICE	WHAT IT DOES	KEY DETAIL
Trusted Advisor	Inspects your AWS environment against best practices and recommends improvements	Draws on best practices from serving hundreds of thousands of customers; closes security gaps
CloudWatch	A reliable, scalable, flexible monitoring solution you can start using in minutes	Metrics for every service; alarms that watch metrics and send notifications
Amazon Inspector	A vulnerability management service that continuously scans your workloads	Scans EC2 and ECR container images; creates a finding for each issue
Amazon GuardDuty	A continuous security monitoring service for your AWS environment	Threat-intelligence feeds + ML flag unexpected or malicious activity
AWS Shield	Protects an enterprise network from distributed denial-of-service (DDoS) attacks	Automatic for common attacks; Shield Advanced is the managed add-on
AWS Config	A continuous monitoring and assessment service	Inventories resources, records config changes, notifies, and integrates to remediate

Trusted Advisor inspects your environment and makes recommendations when there are opportunities to improve performance or close security gaps. What you can access depends on your Support plan: with Basic or Developer Support you use the console for core security checks and all service-quota checks; with Business, Enterprise On-Ramp, or Enterprise Support you also get the Support API and AWS CLI and **all** checks — cost optimization, security, fault tolerance, performance, and service quotas. You can also use Amazon EventBridge to monitor the status of Trusted Advisor checks. **CloudWatch** automatically displays metrics about every AWS service you use, supports custom dashboards, and lets you create alarms that watch metrics and either send notifications or automatically change the resources you monitor when a threshold is breached — giving system-wide visibility into utilization, performance, and operational health.

Amazon Inspector continuously scans your workloads for vulnerabilities, automatically discovering and scanning EC2 instances and container images in Amazon ECR for software vulnerabilities and unintended network exposure. When it finds one, it creates a **finding** that describes the vulnerability, identifies the affected resource, rates the severity, and provides remediation guidance. **Amazon GuardDuty** is a continuous security monitoring service that uses threat-intelligence feeds (such as lists of malicious IP addresses and domains) and machine learning to identify unexpected and potentially unauthorized activity — privilege escalations, use of exposed credentials, or communication with malicious IPs or domains. It can detect a compromised EC2 instance serving malware or mining bitcoin, and it watches

account behavior for signs of compromise, such as an instance deployed in a Region that has never been used or an unusual API call that weakens the password policy.

AWS Shield helps protect against a DDoS attack — a malicious attempt to disrupt normal traffic by overwhelming a target or its surrounding infrastructure with a flood of internet traffic (picture an unexpected traffic jam clogging the highway). Building your application on AWS gives you automatic protection against common DDoS attacks; **Shield Advanced** is the managed threat-protection service that adds extra DDoS detection, mitigation, and response. **AWS Config** is a continuous monitoring and assessment service that gives you an inventory of your resources and records changes to their configuration. You can view current and historic configurations to troubleshoot outages and conduct security attack analyses, roll a resource back to a steady state during an outage, receive notifications when changes occur, and integrate with other services to remediate issues.

> **WORKED EXAMPLE – AWS CONFIG EVALUATING RULES**
>
> Suppose an Amazon S3 bucket is made public. As this configuration change occurs, AWS Config **records and normalizes** the change into a consistent format and **automatically evaluates** it against the rules you have set. You can then access the change history and compliance results through the console or API. You can configure **AWS Systems Manager** or **Amazon SNS** to be invoked so the change is remediated or you are alerted, and you can deliver the change-history and snapshot files of the monitored resources to an **S3 bucket** for analysis. This detect-evaluate-remediate loop is exactly what the module lab automates with a Config rule that calls a Lambda function.

> **COMMON PITFALL**
>
> Two pairs are easy to swap on the exam. **Inspector vs. GuardDuty:** Inspector is *vulnerability management* that scans EC2 and ECR images for known weaknesses and files findings; GuardDuty is *threat detection* that watches behavior and network activity with threat intel and ML. **Config vs. CloudTrail:** Config records configuration state and gives an inventory of resources; CloudTrail records API calls and provides no inventory.

7.3 AWS services that support the resolution and recovery phase

Once an incident is understood, AWS offers services that help remediate it and recover. The module highlights five — again, not an exhaustive list. Three of them (**Lambda, Amazon SNS, and Step Functions**) are described as **event-driven response** services: programs written to respond to actions generated by the user or the system. Combining these lets a detective mechanism automatically trigger a responsive one.

TABLE 7.2 Responsive services for the resolution and recovery phase

SERVICE	WHAT IT DOES	KEY DETAIL
Systems Manager	Gives you visibility and control of your AWS infrastructure	Unified view of operational data; keeps instances in a defined state; patching
CloudFormation	Models and provisions your AWS resources from a template	Can recreate a staging environment in an isolated (forensic) VPC
Amazon SNS	Event-driven web service to instantly send and receive notifications	Integrated with Lambda — a message on a topic can invoke a function
Step Functions	Low-code visual workflow service for distributed applications	Event-driven workflows manage failures, retries, parallelization, observability
AWS Lambda	Serverless, event-driven compute that runs code on demand	Stateless; pay only for compute used; scales by launching copies

Systems Manager provides a unified interface to view operational data from multiple AWS services and automate operational tasks; you can group resources by application, monitor and troubleshoot them, keep instances in their defined state, and perform on-demand changes such as updating applications or running shell scripts. **CloudFormation** lets you describe all the resources you want in a template and takes care of provisioning and dependencies — during an incident, a team can use it to recreate a staging environment inside an isolated, or forensic, VPC, then deep-dive, reproduce the issue, apply a fix, test, and push to production. **Amazon SNS** is an event-driven web service that lets applications, users, and devices instantly send and receive notifications from the cloud; because SNS and Lambda are integrated, publishing a message to a topic with a subscribed function invokes that function with the message payload — handy for receiving notifications of potential exploits.

Step Functions is a low-code, visual workflow service used to build distributed applications and automate IT and business processes; its event-driven workflows manage failures, retries, parallelization, service integrations, and observability so developers can focus on higher-value business logic. **AWS Lambda** is a serverless, event-driven compute service that runs code on demand without provisioning or managing servers — you pay only for the compute time you consume and nothing while your code is idle. Lambda functions are **stateless**, with no affinity to the underlying infrastructure, so Lambda can rapidly launch as many copies as needed to keep up with incoming events. You associate a function with specific resources (such as an S3 bucket or an SNS topic), and you can use AWS KMS to store and retrieve any secrets the function needs. How a function is invoked depends on the event source: the **push model** (the source invokes the function directly), the **pull model** (Lambda polls the source and invokes on events), or **request-response** (Lambda runs synchronously and returns the response immediately).

> **WORKED EXAMPLE – LAMBDA FOR INCIDENT RESPONSE (CLOUDTRAIL STOPLOGGING)**
>
> Assume an account has CloudTrail enabled. If CloudTrail is ever disabled, the response is to re-enable it and investigate who disabled it. You use **EventBridge** (formerly CloudWatch Events) to monitor for the specific `cloudtrail:StopLogging` event and invoke a **Lambda** function when it occurs. The function collects the details of the event — the identity of the principal that disabled CloudTrail, when it was disabled, and the affected resource — and that processed information is sent as an **Amazon SNS** notification carrying only the values a response analyst needs. A second Lambda function can be invoked automatically to restart the logging. This is event-driven response in miniature: a detective mechanism invokes a responsive mechanism to remediate automatically and reduce time-to-value.

WORKING TOGETHER – AUTOMATED RESPONSE TO A COMPROMISED INSTANCE

- A script or third-party tool pushes an instance ID to an **Amazon SNS** topic; a **Lambda** function verifies the ID and, if compromised, starts a **Step Functions** workflow.
- The instance is removed from its Auto Scaling group, a **snapshot** is taken of any attached EBS volumes, metadata (IP, AMI ID, subnets) is recorded, a **quarantine** tag is applied, and the team is notified.
- The instance is isolated by removing all its security groups and assigning a new **forensics security group** with no inbound or outbound permissions.
- A **CloudFormation** template builds a brand-new environment — a new VPC with a forensics instance and prebuilt tools attached to a copy of the snapshot volumes.
- A basic forensics investigation runs on the attached volumes, and a report is generated and sent to the team through an **SNS** topic.

COMMON PITFALL

Remember that in event-driven response the detection and the remediation are *separate* mechanisms wired together — the detective service (EventBridge, Config, GuardDuty) does not itself fix anything; it invokes a responsive service (Lambda, SNS, Step Functions) that does. And because Lambda is stateless, do not design a function that relies on data persisting from a previous invocation.

7.4 Best practices for handling an incident

Strong tooling only pays off inside a prepared process. The module closes with industry best practices for handling incidents, grouped around preparing your people and plans, pre-positioning access and tools, and rehearsing the whole thing before you need it. Preparation is what minimizes disruption when a real incident lands.

TABLE 7.3 Industry best practices for handling incidents

PRACTICE	WHAT IT MEANS
Identify key personnel, external resources, and tooling	Maintain a contact list of people to involve; engage external partners as needed; research and test tools that help you respond and recover
Automate containment capabilities	Automated containment of an incident reduces response times and organizational impact
Develop incident response plans	Create easy-to-follow runbooks; keep escalation and communications plans current; document root cause to build mitigation strategies and prevent recurrence
Pre-provision access and tools	Ensure security personnel have the right tools pre-deployed and the correct access pre-provisioned into AWS so they can respond appropriately
Run incident response game days	Rehearse frequently with simulated events for different threats, involving key staff and management; feed lessons learned back into your process

THE MODULE LAB, IN BRIEF – REMEDIATING AN INCIDENT BY USING AWS CONFIG AND LAMBDA

The lab (90 minutes) turns these ideas into a working automated remediation. You examine and update **IAM roles**, set up **AWS Config** to monitor resources, and modify a **security group** that Config monitors. You then create and run an **AWS Config rule that calls a Lambda function** to remediate the change, revisit the security-group configuration to confirm the fix, and use **CloudWatch logs** for verification. It is the discovery phase (Config detects a non-compliant security group) wired directly to the recovery phase (Lambda remediates it).

COMMON PITFALL

Pre-provisioning is deliberately done *ahead* of any incident. Scrambling to grant responders access or deploy forensic tools during an active incident wastes the very minutes that matter — the best practice is to have the correct access pre-provisioned and the right tools pre-deployed into AWS before the incident occurs.

Module summary

- Incident response is a set of information security policies and procedures used to identify, contain, and eliminate cyberattacks; its goal is to quickly detect and halt attacks to minimize damage and prevent repeats.
- Not all events are incidents — a remote VPN login, a failing-but-operational drive, or a denied access attempt are events to monitor, not necessarily incidents that need immediate remediation.
- Response runs in two phases: discovery and recognition (identification/logging/categorization, notification and escalation, investigation and diagnosis) and resolution and recovery (forensic isolation, stage a fix, deploy the fix, incident closure).
- Six detective services support discovery and recognition: Trusted Advisor, CloudWatch, Amazon Inspector, GuardDuty, Shield, and AWS Config — and AWS Config's sample-exam signature is a continuous monitoring/assessment service that inventories resources and records configuration changes.
- Five responsive services support resolution and recovery: Systems Manager, CloudFormation, Amazon SNS, Step Functions, and Lambda; SNS, Step Functions, and Lambda are the event-driven response services.
- Event-driven automation lets a detective mechanism invoke a responsive one — EventBridge catching cloudtrail:StopLogging to invoke a remediating Lambda, or an SNS/Lambda/Step Functions/CloudFormation workflow that snapshots, isolates, and forensically investigates a compromised instance.
- Watch the look-alikes: Inspector (vulnerability scanning of EC2/ECR, findings) vs. GuardDuty (threat detection via intel + ML); Config (configuration inventory) vs. CloudTrail (API-call history); Lambda is stateless.
- Best practices: identify key personnel/external resources/tooling, automate containment, develop IR plans and runbooks, pre-provision access and tools, and run game days in safe environments with lessons fed back into the process.

Review questions

1. **What is the difference between an event and an incident, and why does it matter?**

   Answer: An event is any observed occurrence; an incident is an event that needs immediate analysis and remediation. Not all events are incidents — a remote login on an approved VPN, a failing-but-operational drive, or a denied access attempt are events worth monitoring but not necessarily incidents to remedy immediately. Distinguishing them avoids both complacency and needless panic.

2. **Name the two phases of incident response and what happens in each.**

   Answer: Discovery and recognition — incident identification, logging, and categorization; notification and escalation; and investigation and diagnosis. Resolution and recovery — forensic isolation, staging a fix, deploying the fix, and incident closure.

3. **How do Amazon Inspector and Amazon GuardDuty differ?**

   Answer: Inspector is a vulnerability management service that continuously scans EC2 instances and ECR container images for software vulnerabilities and network exposure, creating a finding for each. GuardDuty is a continuous threat-detection service that uses threat-intelligence feeds and machine learning to spot unexpected or malicious activity and account-behavior anomalies.

4. **The sample exam asks for a continuous monitoring and assessment service that provides an inventory of AWS resources. Which service, and why not Lambda, CloudTrail, or Fargate?**

   Answer: AWS Config — it continuously captures configuration changes, inventories resources, sends notifications, and can invoke a Lambda function to remediate. Lambda is serverless compute and does not monitor the environment; CloudTrail captures API calls but gives no resource inventory; Fargate is serverless compute for containers and does not monitor.

5. **Which AWS services are event-driven response services, and how does event-driven response work?**

   Answer: Lambda, Amazon SNS, and Step Functions. A detective mechanism invokes a responsive mechanism to remediate automatically — for example, EventBridge monitors for cloudtrail:StopLogging, invokes a Lambda function that collects the event details, sends an SNS notification to an analyst, and can invoke another Lambda to restart logging.

6. **How is CloudFormation used during forensic response to a compromised instance?**

   Answer: A CloudFormation template creates a brand-new, isolated environment — a new forensic VPC containing an investigation instance with prebuilt tools attached to a copy of the snapshot volumes — so the team can deep-dive, reproduce the issue, apply a fix, and test without touching production.

7. **List the industry best practices for handling an incident.**

   Answer: Identify key personnel, external resources, and tooling; automate containment capabilities; develop incident response plans and runbooks; pre-provision access and pre-deploy tools into AWS; and run incident response game days, using lessons learned as a feedback loop to improve the process.