🔒 Guided AWS Lab: Simulate EC2 Compromise & Remediate with GuardDuty and Security Hub

🧰 AWS Services Used in This Lab

Category	Service Name	Purpose
Compute & Networking	Amazon EC2	Launch and simulate compromised and malicious instances
	Amazon VPC	Use default or custom networking and security groups
	Elastic IP	Assign fixed IP to simulate known threat IP
Security & Monitoring	Amazon GuardDuty	Detect malicious or unauthorized behavior
	AWS Security Hub	Aggregate, prioritize, and visualize GuardDuty findings
	AWS Identity & Access Management (IAM)	Create a role with permissions for Lambda
	AWS Key Pairs	Secure EC2 access via SSH
	AWS CloudTrail	Capture AWS account activity logs
	Amazon VPC Flow Logs	Monitor network traffic in VPC
	Amazon CloudWatch	View logs and create alarms for threat detection and automation
Storage	Amazon S3	Host a threat list for GuardDuty
Automation & Alerts	AWS Lambda	Automate instance stop based on findings
	Amazon EventBridge	Trigger Lambda from GuardDuty alerts
	Amazon SNS	Send email alerts for detection events
Other	AWS Billing Dashboard	Check usage and ensure resources are cleaned up

💰 AWS Free Tier & Billing Considerations

Before starting the lab, it's important to know which AWS services might lead to charges if not cleaned up after completion.

⚠️ Services That May Incur Charges (if left active)

AWS Service	Free Tier Coverage	Risk of Charges (After 4 Days)
EC2 (t2.micro)	✅ 750 hours/month (Linux only)	⚠️ No charge if within limit and usage is tracked carefully
Elastic IP (EIP)	❌ Only free when attached	⚠️ Charged if allocated but not associated or instance is stopped
S3	✅ 5 GB Standard storage/month	⚠️ Slight cost if >5GB or left unused long-term
Lambda	✅ 1M requests + 400K GB-sec	🚫 No charge unless usage exceeds free tier
SNS	✅ 1M publishes/month	🚫 No charge for simple email test use
GuardDuty	❌ Not included in Free Tier	⚠️ Charged after 30-day free trial
Security Hub	❌ Not included in Free Tier	⚠️ Charged after 30-day free trial
CloudTrail	✅ 1 trail for management events	🚫 No charge for default usage
VPC Flow Logs	❌ Not covered by Free Tier	⚠️ Charged based on volume of logs written to CloudWatch

🧮 Lab Scenario: 2 EC2 Instances Running Until Thursday

If a student launches **2 t2.micro EC2 Linux instances** on **Monday** and keeps them running until **Thursday** (~4 days), that equals **192 hours total (2 x 24 x 4)**.

- The Free Tier includes **750 hours/month**, so this usage is **still within limits**, assuming:
  - No other EC2 instances are running this month
  - Instances are terminated or stopped after Thursday

💡 Students should monitor their EC2 usage under **Billing** → **EC2 Usage Reports** to avoid exceeding the monthly limit.

✅ Safe Practices

- Terminate EC2 instances immediately after the lab
- Release Elastic IPs if not used
- Delete S3 buckets after uploading the threat list
- Disable GuardDuty and Security Hub after testing
- Monitor your account via the Billing Dashboard

> 💡 **Elastic IP Charging Reminder** A disassociated Elastic IP address remains allocated to your account until you explicitly release it. You are charged for all Elastic IP addresses in your account, regardless of whether they are associated or disassociated with an instance. For more information, see the Public IPv4 Address pricing page.

⚙️ Step 0: Set Your Region and AWS Console Access

1. Sign in to your **own AWS account** (not AWS Innovation Sandbox).
2. In the upper right, choose a region (e.g., N. Virginia (us-east-1)). **All resources must be created in this same region.**

🌐 Step 1: Launch EC2 Instances (Compromised & Malicious)

🔑 Create Key Pair

1. Go to the **EC2 Console** → Left menu → **Key Pairs**
2. Click **Create key pair**
3. Name: gd-lab-keypair
4. Key pair type: **RSA**
5. Private key format: **.pem**
6. Click **Create key pair**

> 💡 The file will download. Save it securely—you'll use it to SSH into the EC2 instance.

🔐 Create Security Group

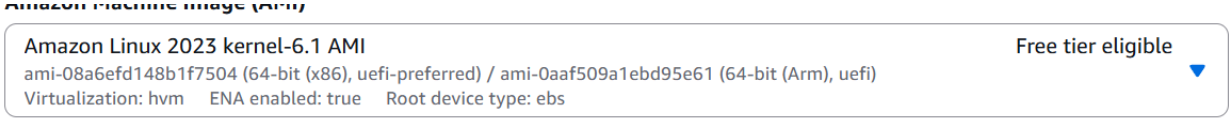
1. In EC2 Console → **Security Groups** → Click **Create security group**
2. Name: gd-lab-sg
3. Description: Security group for GuardDuty lab EC2s
4. VPC: Use your **default VPC** or a custom one
5. Inbound rules:
    - Type: **SSH** | Protocol: TCP | Port: **22** | Source: **My IP**
6. Outbound rules: Leave default (All traffic allowed)
7. Click **Create security group**

🖥️ Launch EC2-Compromised Instance

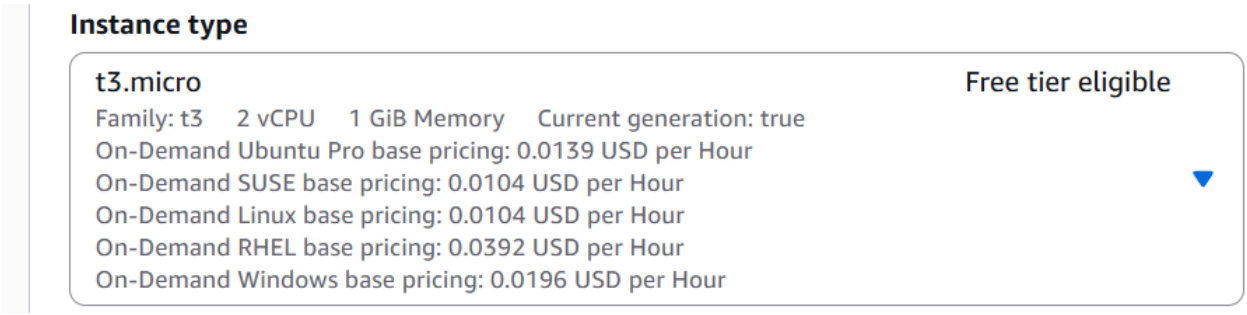
1. Go to **EC2 Console** → Click **Launch instance**
2. Name: EC2-Compromised
3. AMI: **Amazon Linux 2**
4. Instance type: t2.micro
5. Key pair: Select gd-lab-keypair
6. Network settings:
    - Select default VPC or lab VPC
    - Subnet: Any public subnet
    - Auto-assign Public IP: Enabled
    - Firewall (Security Group): Select gd-lab-sg
7. Click **Launch instance**

🖥️ Launch EC2-Malicious Instance

1. Go to **EC2 Console** → Click **Launch instance**
2. Name: EC2-Malicious
3. AMI: Amazon Linux 2023 Kernel-6.1 AMI



4. Instance type: t3.micro (use default)



5. Key pair: Select existing gd-lab-keypair
6. Network settings:
    - Same VPC as above
    - Choose a different subnet or same public subnet
    - Auto-assign Public IP: Enabled
    - Firewall (Security Group): Select existing gd-lab-sg
7. Click **Launch instance**

🔌 Allocate Elastic IP for EC2-Malicious

1. In EC2 Console → **Elastic IPs** → Allocate new address
2. Select the EIP → Click **Actions > Associate**

3. Associate to EC2-Malicious instance

📁 Step 2: Prepare the Threat List in S3

🔨 Create S3 Bucket

1. Go to **S3 Console**: https://s3.console.aws.amazon.com/s3/
2. Click **Create bucket**
3. Name: gd-threat-list-lab-yourname (use lowercase, no spaces)
4. Region: Same as your EC2 instances
5. Uncheck **Block all public access** (you'll use a bucket policy instead)
6. Click **Create bucket**

📝 Upload threatlist.txt

1. In a text editor, create a file with this content: **<EIP of EC2-Malicious>**
2. Save as threatlist.txt
3. Upload to the root of your S3 bucket

🔏 Set Bucket Policy to Allow GuardDuty Access

1. Go to **Permissions** tab of the bucket → Click **Edit** under Bucket Policy
2. Replace with the following JSON (update the bucket name):

```
{
 "Version": "2012-10-17",
 "Statement": [
  {
   "Sid": "AllowGuardDutyReadAccess",
   "Effect": "Allow",
   "Principal": {
    "Service": "guardduty.amazonaws.com"
   },
   "Action": "s3:GetObject",
   "Resource": "arn:aws:s3:::gd-threat-list-lab-yourname/threatlist.txt"
  }
 ]
}
```

3. Click **Save changes**

• Step 3: Enable GuardDuty and Configure Threat Intelligence

1. Go to **GuardDuty Console**: https://console.aws.amazon.com/guardduty/
2. Click **Enable GuardDuty**
3. In left menu, go to **Settings** → **List**
4. In the middle of the page, under Threat IP list, click **Add a threat ip list**

5. In the popup form, enter:
    - Name: e.g., CustomThreatList
    - Location: Paste your file's URL, for example: https://gd-threat-list-lab-emilied777.s3.us-east-1.amazonaws.com/threatlist.txt
    - List format: Plaintext
    - Click "GuardDuty server terms then click Add list
    - Select on CustomThreatList, click Actions then **Activate**
6. Click **Save settings**

🔶 Step 4: Enable Security Hub

1. Go to **Security Hub Console**: https://console.aws.amazon.com/securityhub/
2. Click **Enable Security Hub**
3. In **Settings > Integrations**, confirm **GuardDuty** is enabled

---

🚀 Step 5: Simulate Malicious Behavior

1. Go to **EC2 Console**, select EC2-Compromised → Click **Connect** → Select **SSH client**
2. Open terminal and run:

```
chmod 400 gd-lab-keypair.pem
ssh -i gd-lab-keypair.pem ec2-user@<Public-IP-of-EC2-Compromised>
```

3. Once connected, run:

```
curl http://<Elastic-IP-of-EC2-Malicious>
```

💥 This simulates communication with a known threat IP

📊 Step 6: View Findings in GuardDuty and Security Hub

1. Go to **GuardDuty Console** → Findings
    - Look for findings like Backdoor:EC2/DenialOfService.TCP
2. Go to **Security Hub** → **Findings**
    - Verify same GuardDuty finding appears here

🎙️ Step 7: Create SNS Topic for Notifications

1. Go to **SNS Console**: https://console.aws.amazon.com/sns/
2. Click **Create topic**
    - Type: **Standard**
    - Name: gd-lab-alerts
3. Click **Create topic**

4. Under Subscriptions → Click **Create subscription**
    - Protocol: **Email**
    - Endpoint: your email address (use a temporary email address platform)
        - Example Temp Email Platforms:
            - https://maildrop.cc/
            - https://internxt.com/temporary-email
            - https://temp-mail.io/en
            - http://temp-mail.org
5. Open your email inbox and confirm the subscription

🔧 Step 8: Automate Response with Lambda

Create IAM Role

1. Go to **IAM Console** → Roles → Create role
2. Service: **Lambda**
3. Attach policies:
    - AmazonEC2FullAccess
    - AWSLambdaBasicExecutionRole
4. Role name: gd-lab-lambda-role
5. Click **Create role`

Create Lambda Function

1. Go to **Lambda Console** → Click **Create function**
2. Name: gd-stop-compromised-instance
3. Runtime: Python 3.12
4. Choose existing role: gd-lab-lambda-role
5. Click **Create function**
6. Replace code with:

```
import boto3

def lambda_handler(event, context):
    ec2 = boto3.client('ec2')
    instance_id = event['detail']['resource']['instanceDetails']['instanceId']
    ec2.stop_instances(InstanceIds=[instance_id])
    print(f"Stopped instance: {instance_id}")
```

7. Click **Deploy**

🌟 Step 9: Create EventBridge Rule to Trigger Lambda

1. Go to **EventBridge Console** → Click **Create rule**
2. Name: gd-guardduty-rule
3. Event source: **AWS services**
4. Service Name: GuardDuty, Event Type: GuardDuty Finding
5. Target: Lambda function → Select gd-stop-compromised-instance
6. Click **Create rule**

📊 Step 10: Enable Logging and Monitoring (CloudTrail, VPC Flow Logs, CloudWatch)

🔍 10.1 Enable CloudTrail (if not already enabled)

1. Go to the **CloudTrail Console**
2. Click **Create trail** (or use existing default trail)
3. Name: `gd-lab-cloudtrail`
4. Choose **Apply trail to all regions**
5. For storage location:
    - Create or choose an existing **S3 bucket**
6. Enable **management events** (read/write)
7. Click **Create trail**

✅ CloudTrail is free for management events using 1 trail.

🌐 10.2 Enable VPC Flow Logs

1. Go to **VPC Console**
2. On the left, choose **Your VPCs**
3. Select the VPC associated with your EC2 instances
4. Go to the **Flow Logs** tab → Click **Create Flow Log**
5. Filter: **All traffic**
6. Destination: **Send to CloudWatch Logs**
7. Create a new **IAM Role** (name suggestion: `gd-vpc-flow-role`)
8. Log group: create one (e.g., `/gd/lab/vpcflow`)
9. Click **Create Flow Log**

⚠️ VPC Flow Logs may incur charges if high volumes of data are logged.

📈 10.3 Configure CloudWatch Metrics & Alarms (Optional)

1. Go to **CloudWatch Console**
2. Navigate to **Log groups** → Select the log group created earlier
3. Click **Create Metric Filter** (optional)
4. Set up a filter pattern and assign a name
5. Optionally create an alarm to alert via **SNS**

> ✅ Logging and monitoring provide visibility for auditing and incident detection ⚠️ Remember to delete log groups if you want to avoid long-term storage costs

🧹 Cleanup Checklist (Step-by-Step)

To avoid AWS charges, follow these steps to remove all resources used in this lab:

🔻 EC2 Instances and Elastic IP

1. Go to **EC2 Console** → Instances → Terminate both:
   - EC2-Compromised
   - EC2-Malicious
2. In **Elastic IPs**, select the one associated → **Disassociate** → then **Release Elastic IP**

🪣 S3 Bucket

1. Go to **S3 Console**
2. Empty the bucket gd-threat-list-lab-yourname
3. Then select the bucket → **Delete bucket**

📣 SNS Topic

1. Go to **SNS Console**
2. Delete the topic gd-lab-alerts
3. Also delete any email subscriptions

🧠 Lambda Function & IAM Role

1. Go to **Lambda Console** → Delete gd-stop-compromised-instance
2. Go to **IAM Console** → Roles → Delete gd-lab-lambda-role

📅 EventBridge Rule

1. Go to **EventBridge Console** → Rules → Delete gd-guardduty-rule

🔐 Key Pair and Security Group

1. In **EC2 Console** → Key Pairs → Delete gd-lab-keypair

2. In **Security Groups** → Delete gd-lab-sg (only if not attached elsewhere)

🛡️ GuardDuty & Security Hub (Optional)

1. In **GuardDuty Console**, click **Settings** → **Disable GuardDuty**
2. In **Security Hub Console**, go to **Settings** → **Disable Security Hub**

✅ Finally, visit the **Billing Dashboard** to confirm no active resources are running.

📜 CloudTrail

- If a new trail was created, **delete it**
- (Optional) Keep if using for other AWS activities

📄 VPC Flow Logs

- Go to **VPC Console** → **Flow Logs tab**
- Delete the **Flow Log** associated with the lab VPC
- Delete the associated **IAM role** and **CloudWatch Log Group**

📈 CloudWatch Logs

- Go to **CloudWatch Console** → **Log Groups**
- Delete:
  - Log group for VPC Flow Logs (e.g., `/gd/lab/vpcflow`)
  - Any log groups or metric filters you created