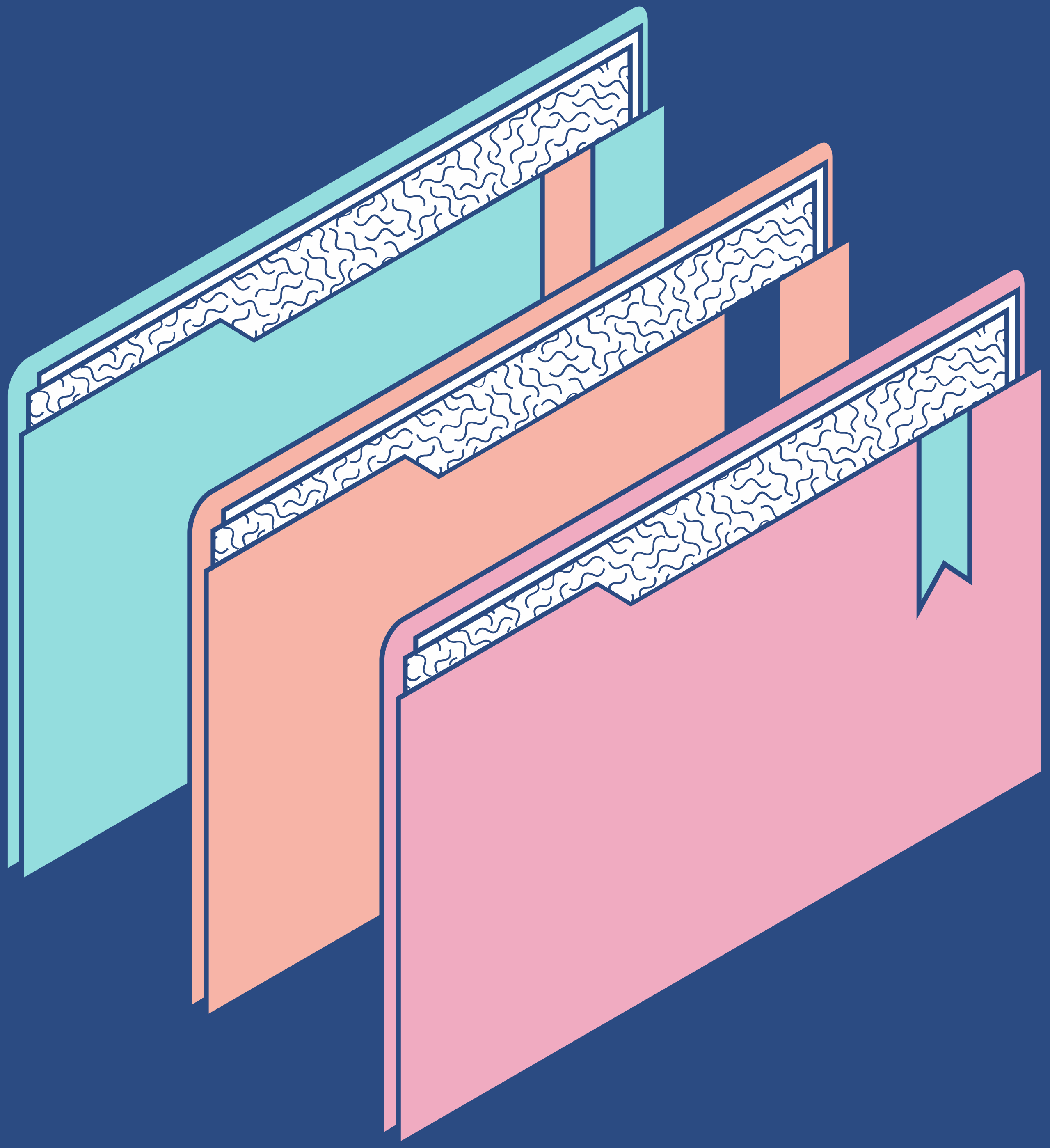
TKH - PHASE 2 - TEAM 1

# Building a Multi-Subnet Cybersecurity Training Lab on AWS

**Cyborgs:** Emmanuel Apiteu, Abraham DeLaCruz, KJ McDaniels, Opeyemi Olaleye, Alawi Rashed

**Project Manager:** Yonisbel Soto

**Team Lead:** Zedd Chisholm



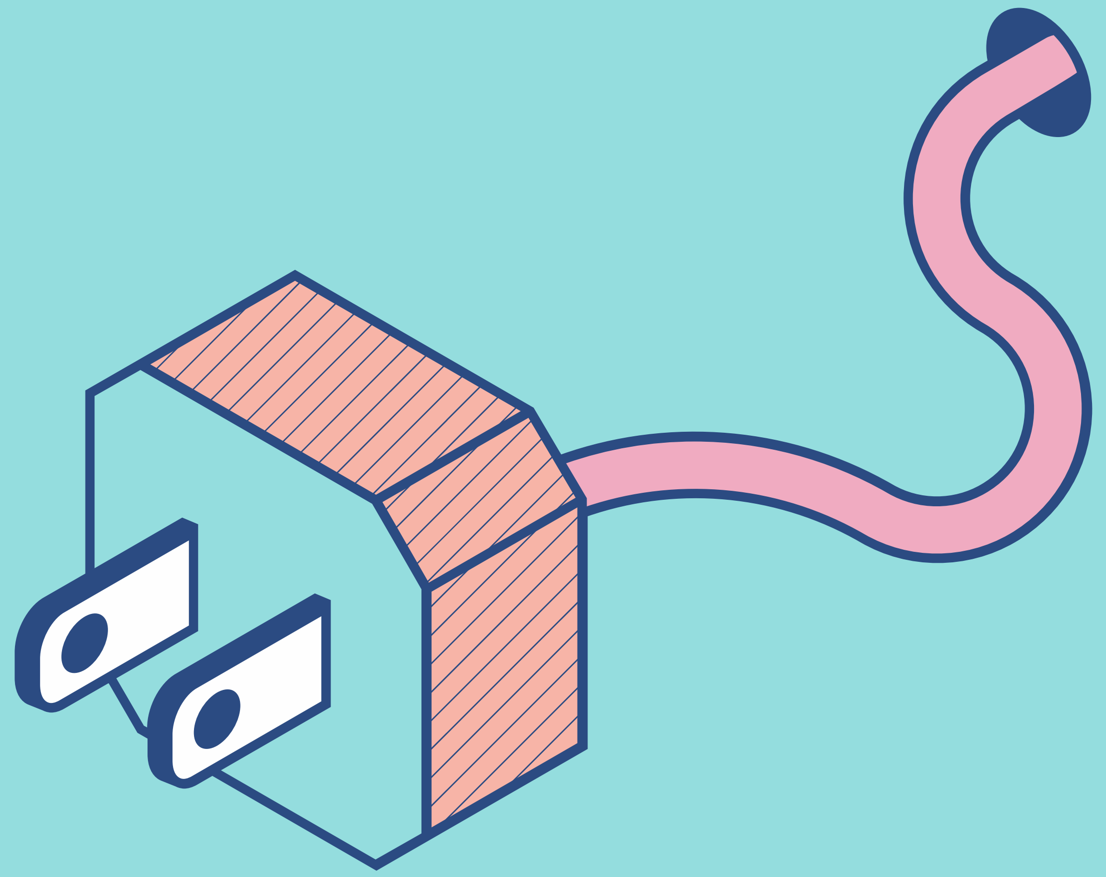
# Agenda

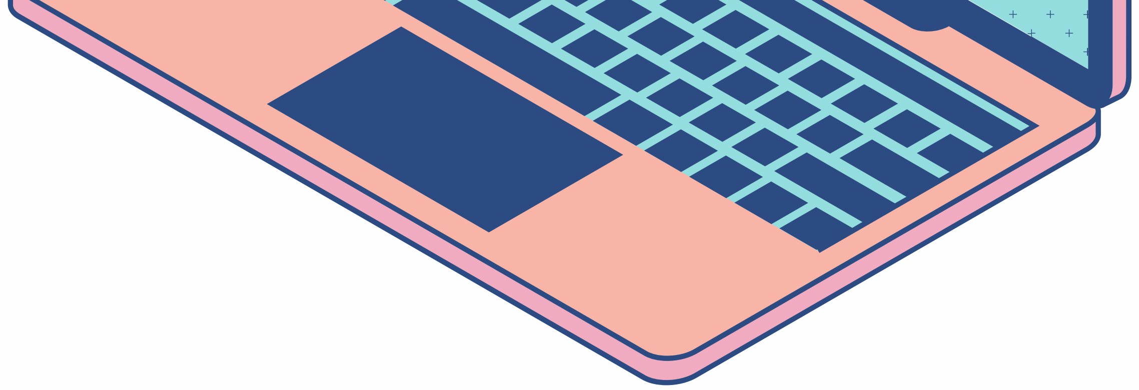
KEY TOPICS DISCUSSED IN THIS PRESENTATION

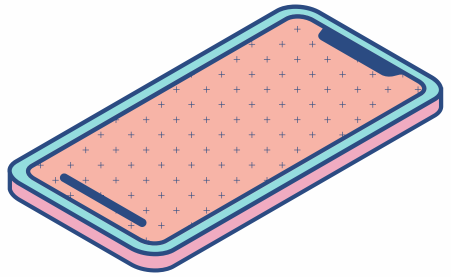
- Introduction and Purpose
- Virtual Lab vs Cloud Lab
- Tools
- Setup Steps
- Demo
- Ways to Elevate: Future Improvements
- Bastion Host Example
- Resources

# Purpose/Intro to Project

Our project's aim was to construct a secure, multi-subnet environment using AWS services to demonstrate advanced network segmentation and security monitoring capabilities, effectively creating a training ground for future cybersecurity professionals. This lab showcases the integration of native AWS tools, including GuardDuty, WAF, Inspector, and CloudWatch, to provide a comprehensive look at threat detection, system vulnerability assessments, and real-time security analytics.







VS

## Traditional Lab Setup (Local PC VMs)

- If you computer goes down your VMs go down
- Limited by your computer resources
- You have to do all the work all the time

## AWS Lab Setup

- High Availability and Reliability
- Scalability and Elasticity
- Managed Services and Advanced Features

# AWS Tools

| | | |
|---|---|---|
| Amazon EC2 | **Provides on-demand, scalable virtual servers (instances) in the cloud** | - Pre-configure instances with vulnerable software or misconfigurations to practice security hardening |
| Amazon VPC | **Creates logically isolated network segments within the AWS cloud** | - Enables defining subnets for different application tiers (e.g., web servers, database) to enforce network segmentation best practices |
| aws<br>Security Group | **Acts as stateful firewalls attached to EC2 instances, controlling inbound and outbound traffic** | - Enables implementing least privilege principles by restricting access to specific ports and IP addresses. |
| NACL | **Stateless firewalls that define allowed inbound and outbound traffic at the subnet level** | - Network equivalent of security groups.<br>- Provide rule-based tool for controlling traffic<br>- Separate rules for inbound/outbound traffic |

# AWS Security Tools

Amazon
CloudWatch

| | |
|---|---|
| **THREAT DETECTION SERVICE** | - Analyzes logs from CloudTrail, VPC Flow Logs, and DNS logs.<br>- Uses threat intelligence to identify security issues.<br>- Continuously monitors for suspicious activity.<br>- Can trigger automated responses. |
| **LOG AGGREGATION & ANALYSIS SERVICE** | - Centralized repository for logs.<br>-Enables filtering and searching of logs.<br>-Provides visualization tools.<br>-Integrates with CloudWatch metrics & alarms. |
| **MONITORING & ALERTING SERVICE** | -Monitors metrics from AWS resources (i.e., CPU usage, networks)<br>-Creates alarms triggered by surpassing threshold<br>-Integrates with CloudWatch Logs<br>-Offers different notification options (SNS, email) |
| **VULNERABILTY SCANNING SERVICE** | -Scans Amazon EC2 instances for vulnerabilities<br>-Uses pre-defined rules to identify potential security weaknesses<br>-Provides detailed reports on discovered vulnerabilites<br>-Integrates with other AWS security services (i.e., GuardDuty) |
| **PROTECTION AGAINST WEB APPLICATION ATTACKS** | -Filters and monitors incoming HTTP/S traffic -Blocks malicious requests based on pre-defined rules<br>-Protects web applications from common attacks<br>-Integrates with CloudWatch for monitoring and logging |

# How to set up an AWS lab environment

1 STEP
Create Network Diagram

2 STEP
VPC and Subnet Creation

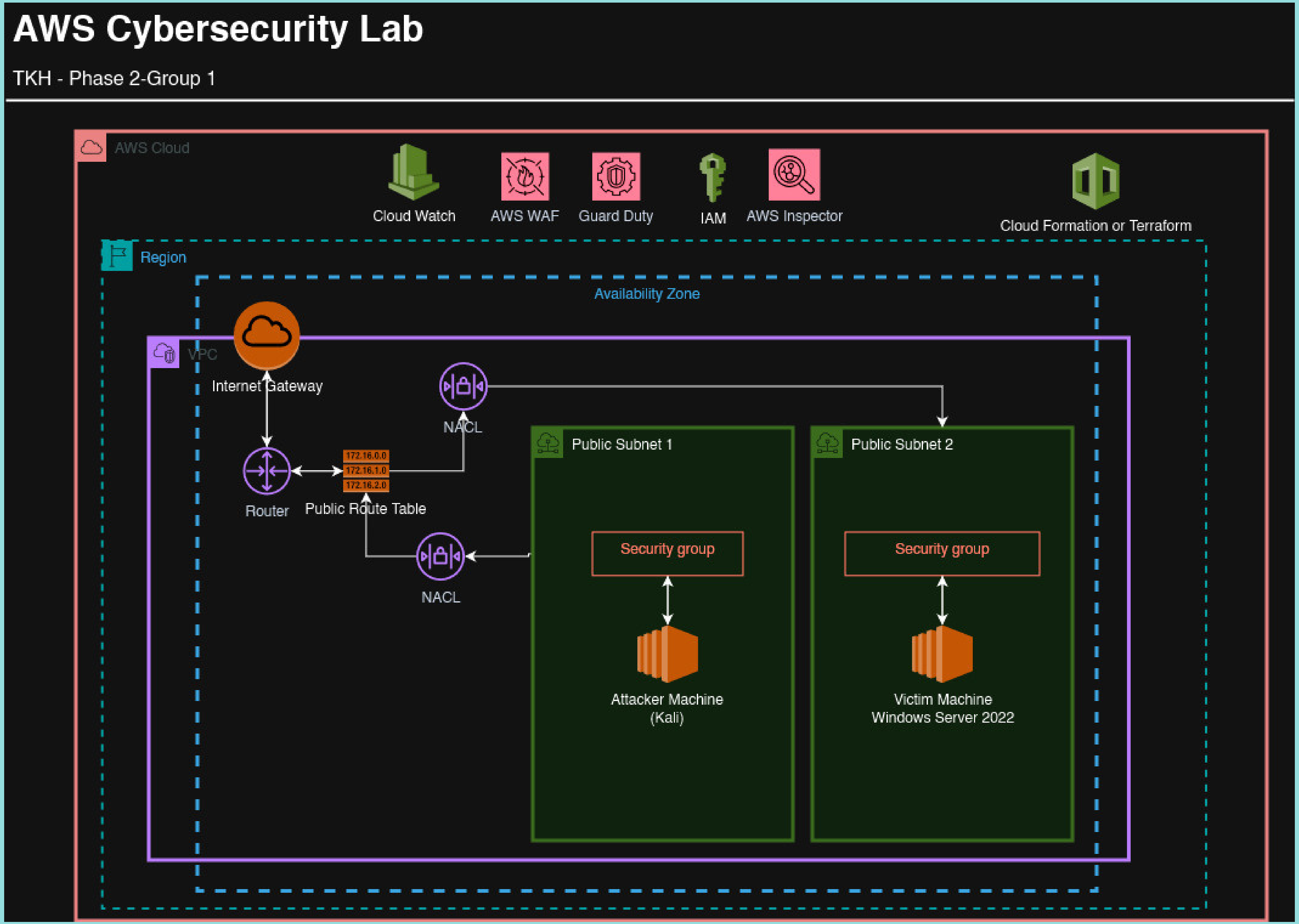
3 STEP
Internet Gateway and Route Tables

4 STEP
Security Group and Network Access Control List (ACL) Creation

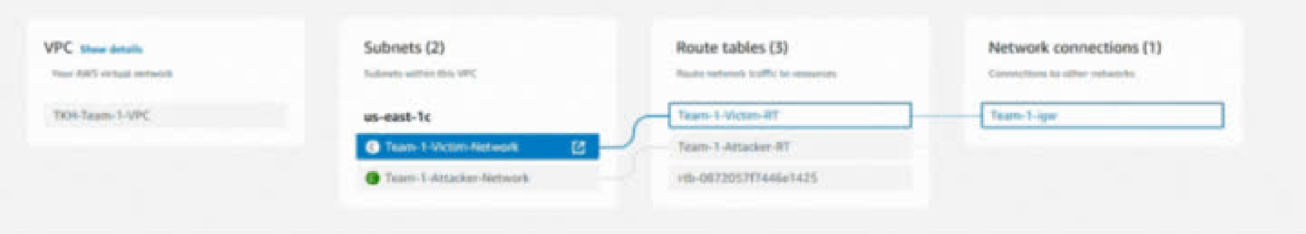
5 STEP
Launching Instances

# Network Diagram

This network topology outlines our AWS Cybersecurity Lab with two public subnets containing an Kali Linux 'Attacker' machine and a 'Victim' Windows Server. Security groups protect both, while AWS services like GuardDuty and AWS Inspector ensure continuous security monitoring.



# VPC and Subnet Creation



## Virtual Private Cloud (VPC):

To create a VPC, specify a CIDR (Classless Inter-Domain Routing) block for the VPC's IP address range. You can also configure additional settings such as DNS resolution, DHCP options, and IPv6 support.

## Subnet Creation

When creating subnets, specify a CIDR block within the VPC's address range. Additionally, define the availability zone in which the subnet will reside. Create subnets across multiple availability zones to ensure redundancy and fault tolerance for your applications.

# Internet Gateway and Route Tables

## Internet Gateway

It essentially serves as a gateway between your VPC and the public internet. Without an Internet Gateway, resources within your VPC are isolated and cannot communicate directly with the internet

## Route Tables

A Route Table is a set of rules, called routes, that are used to determine where network traffic from your VPC should be directed. Each subnet in your VPC must be associated with a route table, which controls the routing for the subnet

# Security Group and Network Access Control List (NACL) Creation

## Security Groups

Security Groups act as a virtual firewall for your AWS instances to control inbound and outbound traffic Security Groups are stateful, meaning if you allow inbound traffic, the return traffic is automatically allowed regardless of outbound rules.

## Network Access Control Lists (NACLs)

Network Access Control Lists (NACLs) are another layer of defense for controlling traffic at the subnet level within your VPC. : Unlike security groups, which are associated with individual instances, Network NACLs are associated with subnets. They control traffic entering and leaving the subnet.

# Launching Instances

## Function

Launching instances in AWS allows you to create virtual servers, known as EC2 instances, within your chosen region and VPC.

## Process

To launch an instance, you select an Amazon Machine Image (AMI) which serves as a template for the virtual server's operating system and pre-installed software

## Access

Once launched, you can access your instances remotely using SSH (for Linux-based instances) or Remote Desktop Protocol (RDP) for Windows instances.

# Configuring AWS Security Services

1

STEP

AWS Inspector

2

STEP

AWS GuardDuty

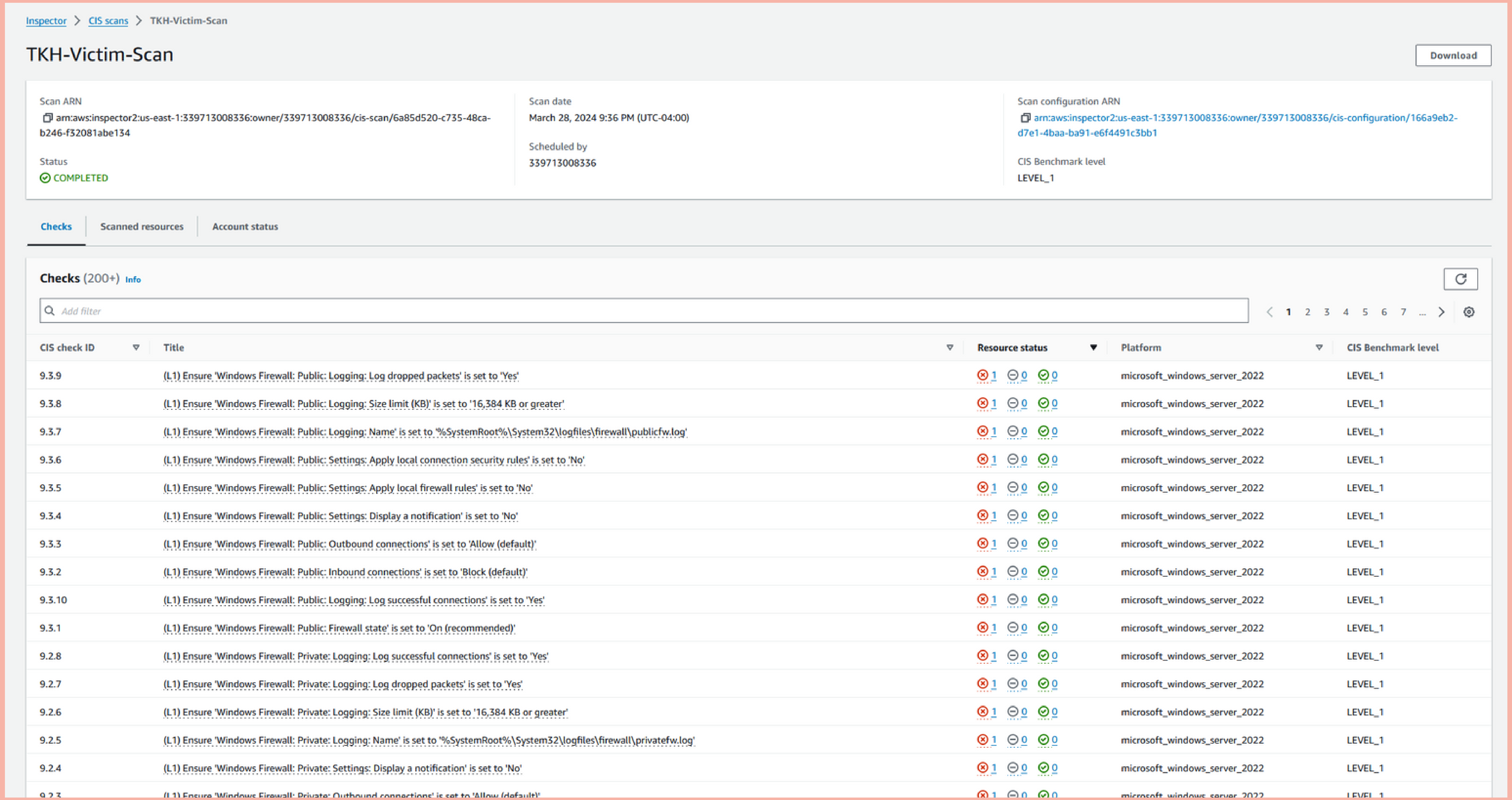
3

STEP

AWS CloudWatch

# AWS Inspector

Our initial scan resulted in 200+ vulnerabilities discovered on out Windows Server 2022 system.



## TKH-Victim-Scan

Scan ARN
arn:aws:inspector2:us-east-1:339713008336:owner/339713008336/cis-scan/6a85d520-c735-48ca-b246-f32081abe134

Status
✓ COMPLETED

Scan date
March 28, 2024 9:36 PM (UTC-04:00)

Scheduled by
339713008336

Scan configuration ARN
arn:aws:inspector2:us-east-1:339713008336:owner/339713008336/cis-configuration/166a9eb2-d7e1-4baa-ba91-e6f4491c3bb1

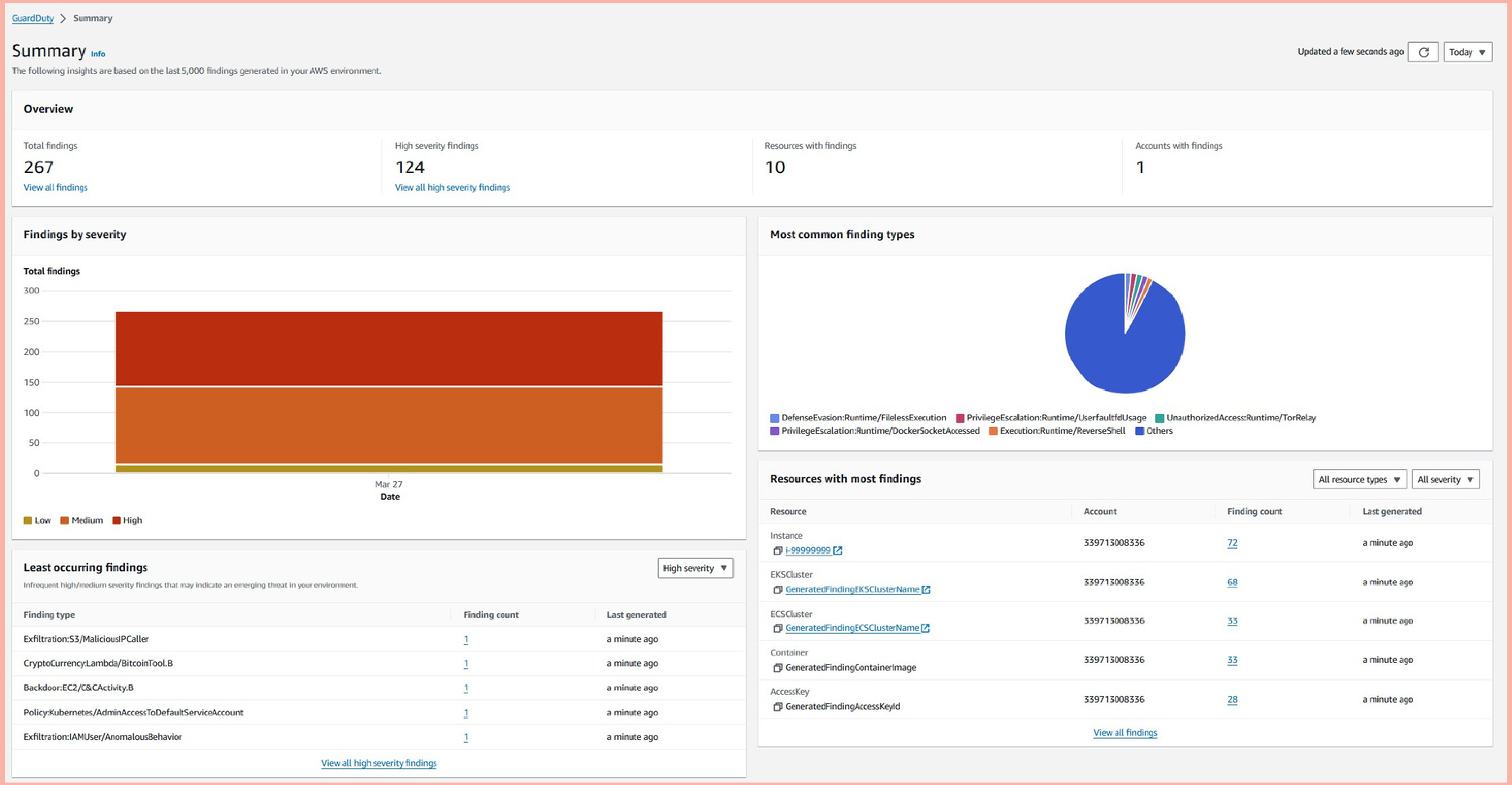
CIS Benchmark level
LEVEL_1

**Checks** (200+) Info

| CIS check ID | Title | Resource status | Platform | CIS Benchmark level |
|---|---|---|---|---|
| 9.3.9 | (L1) Ensure 'Windows Firewall: Public: Logging: Log dropped packets' is set to 'Yes' | ⊗ 1 ⊖ 0 ✓ 0 | microsoft_windows_server_2022 | LEVEL_1 |
| 9.3.8 | (L1) Ensure 'Windows Firewall: Public: Logging: Size limit (KB)' is set to '16,384 KB or greater' | ⊗ 1 ⊖ 0 ✓ 0 | microsoft_windows_server_2022 | LEVEL_1 |
| 9.3.7 | (L1) Ensure 'Windows Firewall: Public: Logging: Name' is set to '%SystemRoot%\System32\logfiles\firewall\publicfw.log' | ⊗ 1 ⊖ 0 ✓ 0 | microsoft_windows_server_2022 | LEVEL_1 |
| 9.3.6 | (L1) Ensure 'Windows Firewall: Public: Settings: Apply local connection security rules' is set to 'No' | ⊗ 1 ⊖ 0 ✓ 0 | microsoft_windows_server_2022 | LEVEL_1 |
| 9.3.5 | (L1) Ensure 'Windows Firewall: Public: Settings: Apply local firewall rules' is set to 'No' | ⊗ 1 ⊖ 0 ✓ 0 | microsoft_windows_server_2022 | LEVEL_1 |
| 9.3.4 | (L1) Ensure 'Windows Firewall: Public: Settings: Display a notification' is set to 'No' | ⊗ 1 ⊖ 0 ✓ 0 | microsoft_windows_server_2022 | LEVEL_1 |
| 9.3.3 | (L1) Ensure 'Windows Firewall: Public: Outbound connections' is set to 'Allow (default)' | ⊗ 1 ⊖ 0 ✓ 0 | microsoft_windows_server_2022 | LEVEL_1 |
| 9.3.2 | (L1) Ensure 'Windows Firewall: Public: Inbound connections' is set to 'Block (default)' | ⊗ 1 ⊖ 0 ✓ 0 | microsoft_windows_server_2022 | LEVEL_1 |
| 9.3.10 | (L1) Ensure 'Windows Firewall: Public: Logging: Log successful connections' is set to 'Yes' | ⊗ 1 ⊖ 0 ✓ 0 | microsoft_windows_server_2022 | LEVEL_1 |
| 9.3.1 | (L1) Ensure 'Windows Firewall: Public: Firewall state' is set to 'On (recommended)' | ⊗ 1 ⊖ 0 ✓ 0 | microsoft_windows_server_2022 | LEVEL_1 |
| 9.2.8 | (L1) Ensure 'Windows Firewall: Private: Logging: Log successful connections' is set to 'Yes' | ⊗ 1 ⊖ 0 ✓ 0 | microsoft_windows_server_2022 | LEVEL_1 |
| 9.2.7 | (L1) Ensure 'Windows Firewall: Private: Logging: Log dropped packets' is set to 'Yes' | ⊗ 1 ⊖ 0 ✓ 0 | microsoft_windows_server_2022 | LEVEL_1 |
| 9.2.6 | (L1) Ensure 'Windows Firewall: Private: Logging: Size limit (KB)' is set to '16,384 KB or greater' | ⊗ 1 ⊖ 0 ✓ 0 | microsoft_windows_server_2022 | LEVEL_1 |
| 9.2.5 | (L1) Ensure 'Windows Firewall: Private: Logging: Name' is set to '%SystemRoot%\System32\logfiles\firewall\privatefw.log' | ⊗ 1 ⊖ 0 ✓ 0 | microsoft_windows_server_2022 | LEVEL_1 |
| 9.2.4 | (L1) Ensure 'Windows Firewall: Private: Settings: Display a notification' is set to 'No' | ⊗ 1 ⊖ 0 ✓ 0 | microsoft_windows_server_2022 | LEVEL_1 |
| 9.2.3 | (L1) Ensure 'Windows Firewall: Private: Outbound connections' is set to 'Allow (default)' | ⊗ 1 ⊖ 0 ✓ 0 | microsoft_windows_server_2022 | LEVEL_1 |

# AWS GuardDuty

GuardDuty is a threat detection service that continuously monitors for malicious activity and unauthorized behavior



GuardDuty > Summary
Info
Updated a few seconds ago
Today ▼
High severity ▼
All resource types ▼
All severity ▼
300
250
200
150
100
50
0
Mar 27
Date
🟨 Low
🟧 Medium
🟥 High
DefenseEvasion:Runtime/FilelessExecution
PrivilegeEscalation:Runtime/UserfaultfdUsage
UnauthorizedAccess:Runtime/TorRelay
PrivilegeEscalation:Runtime/DockerSocketAccessed
Execution:Runtime/ReverseShell
Others

## Summary

The following insights are based on the last 5,000 findings generated in your AWS environment.

### Overview

| Total findings | High severity findings | Resources with findings | Accounts with findings |
|---|---|---|---|
| 267 | 124 | 10 | 1 |
| View all findings | View all high severity findings | | |

### Findings by severity

Total findings

### Most common finding types

### Least occurring findings

Infrequent high/medium severity findings that may indicate an emerging threat in your environment.

| Finding type | Finding count | Last generated |
|---|---|---|
| Exfiltration:S3/MaliciousIPCaller | 1 | a minute ago |
| CryptoCurrency:Lambda/BitcoinTool.B | 1 | a minute ago |
| Backdoor:EC2/C&CActivity.B | 1 | a minute ago |
| Policy:Kubernetes/AdminAccessToDefaultServiceAccount | 1 | a minute ago |
| Exfiltration:IAMUser/AnomalousBehavior | 1 | a minute ago |

View all high severity findings

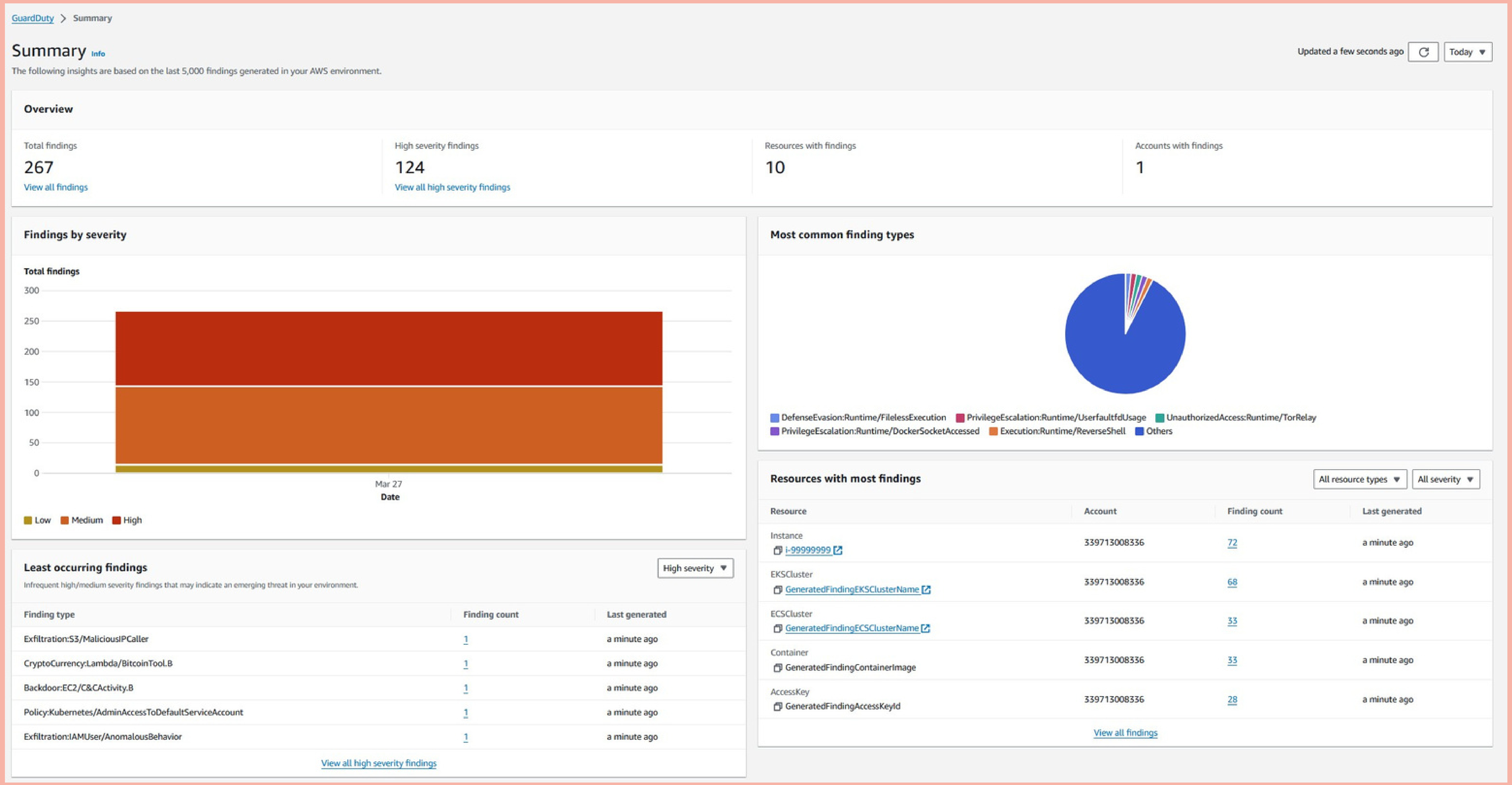
### Resources with most findings

| Resource | Account | Finding count | Last generated |
|---|---|---|---|
| Instance<br>i-99999999 | 339713008336 | 72 | a minute ago |
| EKSCluster<br>GeneratedFindingEKSClusterName | 339713008336 | 68 | a minute ago |
| ECSCluster<br>GeneratedFindingECSClusterName | 339713008336 | 33 | a minute ago |
| Container<br>GeneratedFindingContainerImage | 339713008336 | 33 | a minute ago |
| AccessKey<br>GeneratedFindingAccessKeyId | 339713008336 | 28 | a minute ago |

View all findings

# AWS CloudWatch

CloudWatch is a centralized repository for logs that enables filtering and searching of logs and provides visualization tools.



GuardDuty > Summary
Info
Updated a few seconds ago
Today ▼

Summary

The following insights are based on the last 5,000 findings generated in your AWS environment.

Overview

Total findings
267
View all findings

High severity findings
124
View all high severity findings

Resources with findings
10

Accounts with findings
1

Findings by severity

Total findings
300
250
200
150
100
50
0
Mar 27
Date
🟨 Low
🟧 Medium
🟥 High

Most common finding types

🟦 DefenseEvasion:Runtime/FilelessExecution
🟥 PrivilegeEscalation:Runtime/UserfaultfdUsage
🟩 UnauthorizedAccess:Runtime/TorRelay
🟪 PrivilegeEscalation:Runtime/DockerSocketAccessed
🟧 Execution:Runtime/ReverseShell
🟦 Others

Least occurring findings

Infrequent high/medium severity findings that may indicate an emerging threat in your environment.

| Finding type | Finding count | Last generated |
|---|---|---|
| Exfiltration:S3/MaliciousIPCaller | 1 | a minute ago |
| CryptoCurrency:Lambda/BitcoinTool.B | 1 | a minute ago |
| Backdoor:EC2/C&CActivity.B | 1 | a minute ago |
| Policy:Kubernetes/AdminAccessToDefaultServiceAccount | 1 | a minute ago |
| Exfiltration:IAMUser/AnomalousBehavior | 1 | a minute ago |

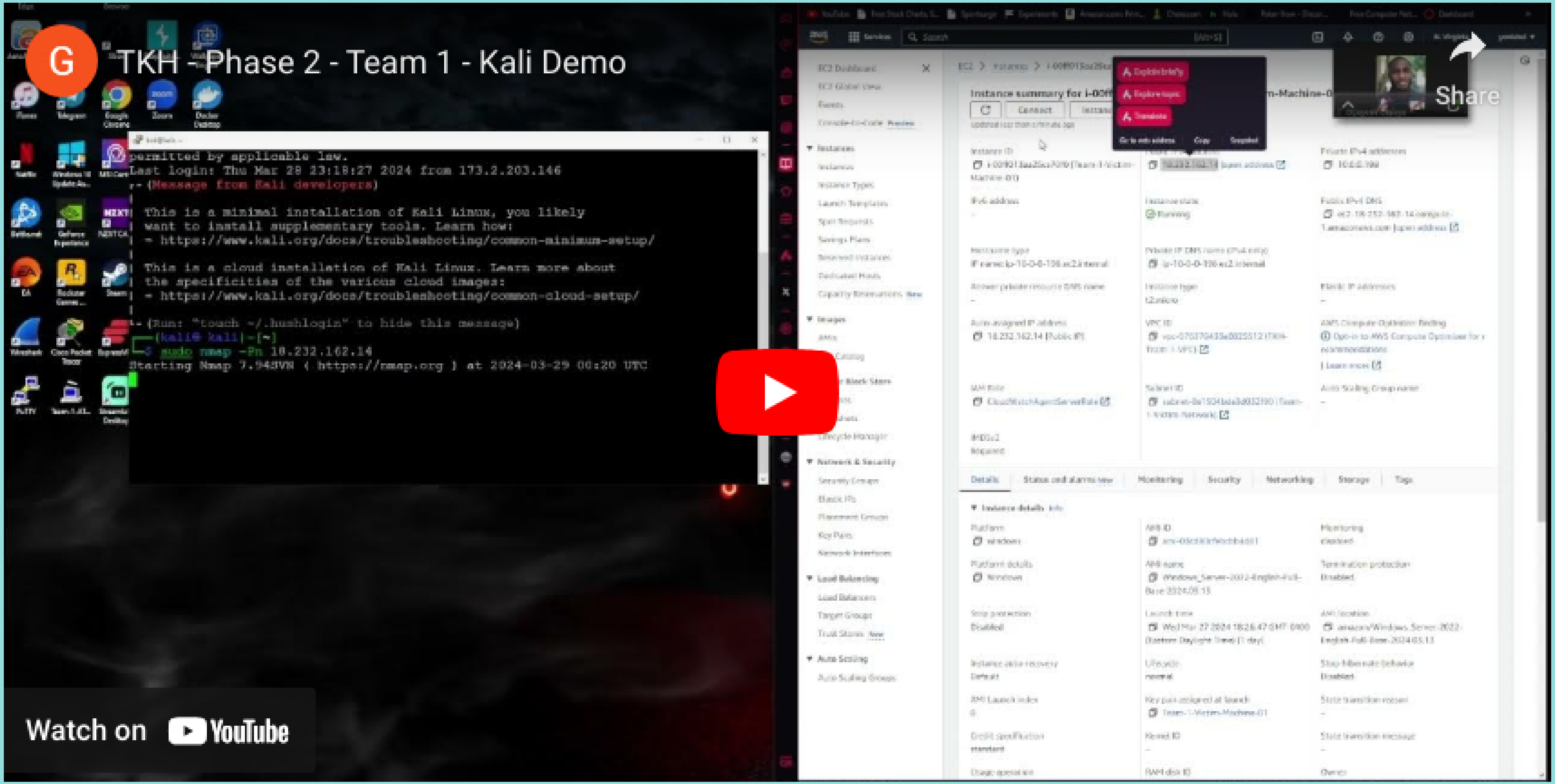
View all high severity findings

Resources with most findings

| Resource | Account | Finding count | Last generated |
|---|---|---|---|
| Instance<br>i-99999999 | 339713008336 | 72 | a minute ago |
| EKSCluster<br>GeneratedFindingEKSClusterName | 339713008336 | 68 | a minute ago |
| ECSCluster<br>GeneratedFindingECSClusterName | 339713008336 | 33 | a minute ago |
| Container<br>GeneratedFindingContainerImage | 339713008336 | 33 | a minute ago |
| AccessKey<br>GeneratedFindingAccessKeyId | 339713008336 | 28 | a minute ago |

View all findings

# ATTACK DEMO

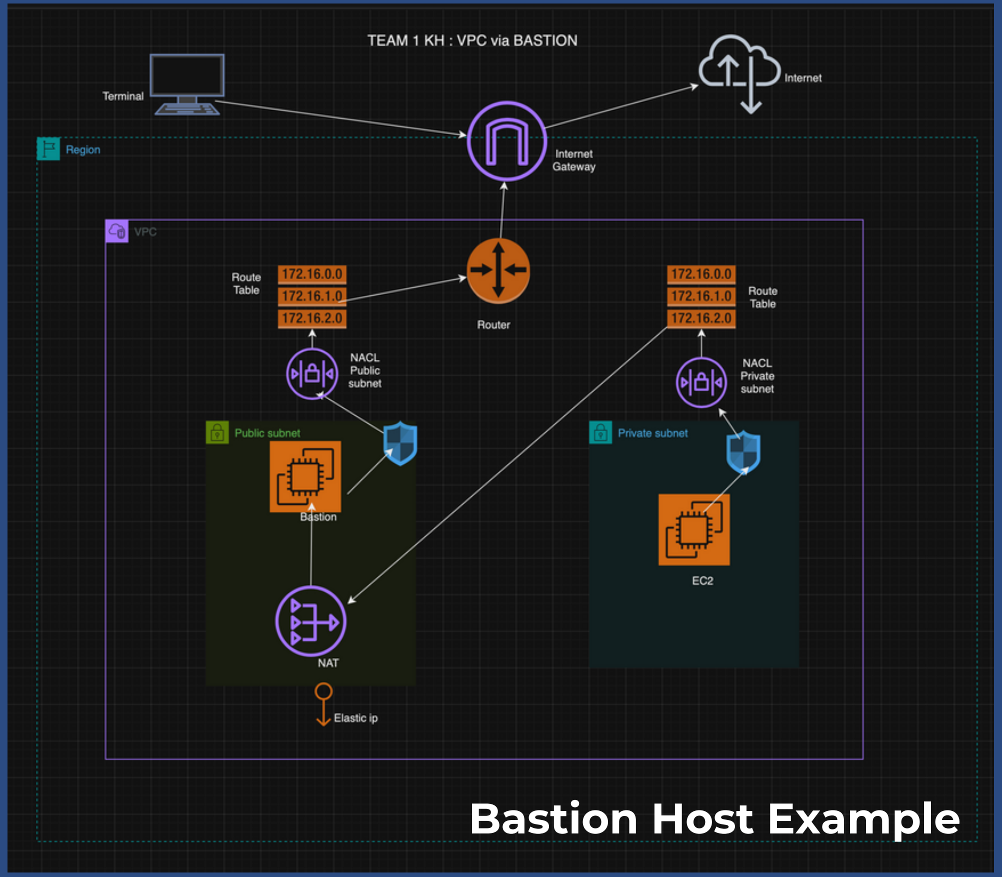


```
permitted by applicable law.
Last login: Thu Mar 28 23:18:27 2024 from 173.2.203.146
┌─(Message from Kali developers)
│
│ This is a minimal installation of Kali Linux, you likely
│ want to install supplementary tools. Learn how:
│ ⇒ https://www.kali.org/docs/troubleshooting/common-minimum-setup/
│
│ This is a cloud installation of Kali Linux. Learn more about
│ the specificities of the various cloud images:
│ ⇒ https://www.kali.org/docs/troubleshooting/common-cloud-setup/
│
└─(Run: "touch ~/.hushlogin" to hide this message)
┌──(kali㉿kali)-[~]
└─$ sudo nmap -Pn 10.232.162.14
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-03-29 00:20 UTC
```

Running nmap from Attack (Kali) to Victim (Windows Server)

# Ways to Elevate

Some things that could be done to improve the lab.



Bastion Host Example

## Adding a Private Subnet

Enhances network segmentation and provides a controlled environment for sensitive operations away from public access.

## Adding a Bastion Host

Enhances network segmentation and provides a controlled environment for sensitive operations away from public access.

## Using additional Managed Tools

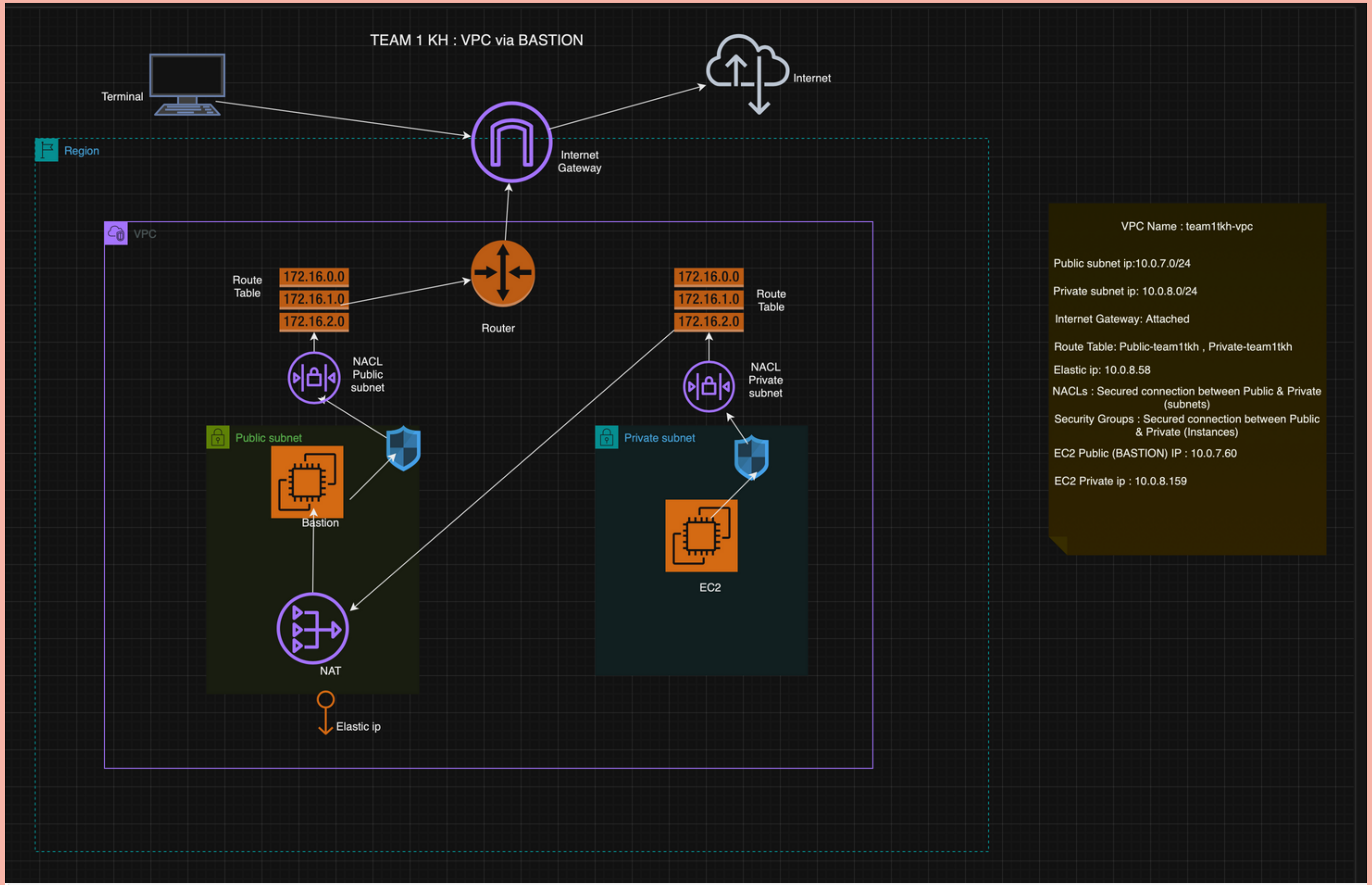
Incorporate advanced AWS services such as AWS Systems Manager for streamlined management and automation.

## Integrating Advanced Security Features

Utilize services like AWS Shield for DDoS protection and AWS Key Management Service for secure key storage.

# Bastion Host Exmaple Steps

1. Create a VPC
2. Create Public and Private subnets
3. Create Route Table
4. Create NAT Gateway
5. Create Internet Gateway
6. Launch 2 EC2 Instances, one in Public and another in Private subnets
7. Commands to proceed :
8. cd Downloads, ls, chmod 400 team1tkhkp.pem, ssh -i team1tkhkp ec2-user@<public ip> Now for private keypair, cat team1tkhkp.pem, create a nano file.
9. Consider public machine as Bastion host and log into Private instance
10. Test by SSH from public into Private instance by using keypair (public) use NANO to transcript Private keypair, then SSH from Private instance to Public instance. Ping to confirm connection.
11. Set Security Groups



VPC Name : team1tkh-vpc

Public subnet ip:10.0.7.0/24

Private subnet ip: 10.0.8.0/24

Internet Gateway: Attached

Route Table: Public-team1tkh , Private-team1tkh

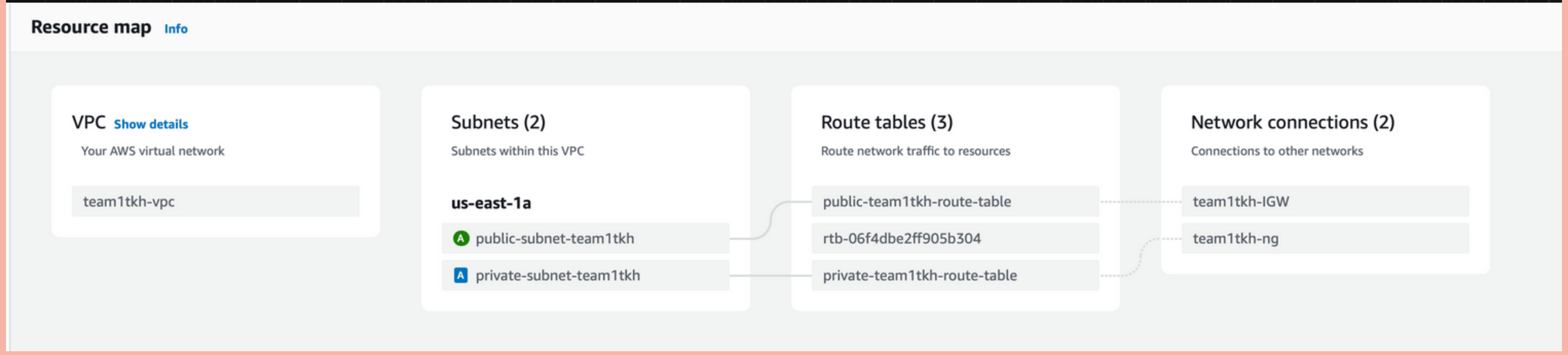
Elastic ip: 10.0.8.58

NACLs : Secured connection between Public & Private (subnets)

Security Groups : Secured connection between Public & Private (Instances)

EC2 Public (BASTION) IP : 10.0.7.60

EC2 Private ip : 10.0.8.159



Resource map Info

VPC Show details
Your AWS virtual network
team1tkh-vpc

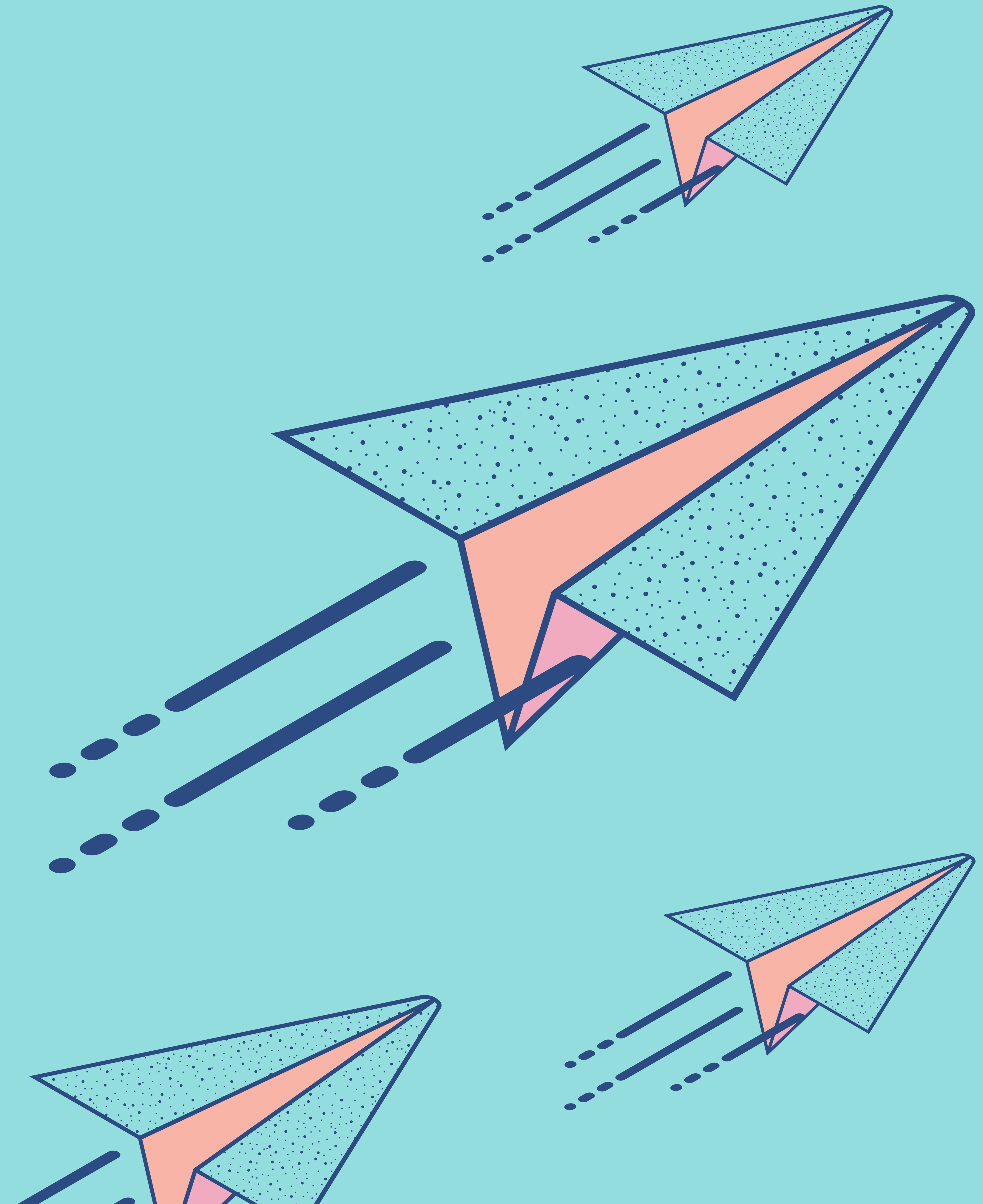
Subnets (2)
Subnets within this VPC
us-east-1a
A public-subnet-team1tkh
A private-subnet-team1tkh

Route tables (3)
Route network traffic to resources
public-team1tkh-route-table
rtb-06f4dbe2ff905b304
private-team1tkh-route-table

Network connections (2)
Connections to other networks
team1tkh-IGW
team1tkh-ng

# THANK YOU

Are there any questions? Send them our way.



Team GitHub Repo

# Resource Page

Our complete list of resources can be found on our GitHub

Configuring your Subnet

How to Install Metaploit on Kali Linux

AWS EC2 IP Addressing

**Introducing Amazon CloudWatch Internet Monitor (AWS Documentation)**

Quicker CloudWatch Logs for EC2 Tutorial Setup

How to Enable AWS GuardDuty

# Resource Page

Our complete list of resources can be found on our GitHub

Linux Bastion Hosts on AWS Partner Solution Deployment Guide

VPC log Endpoints

Getting Started with Guard Duty

EC2 SSH Error

Team Repo