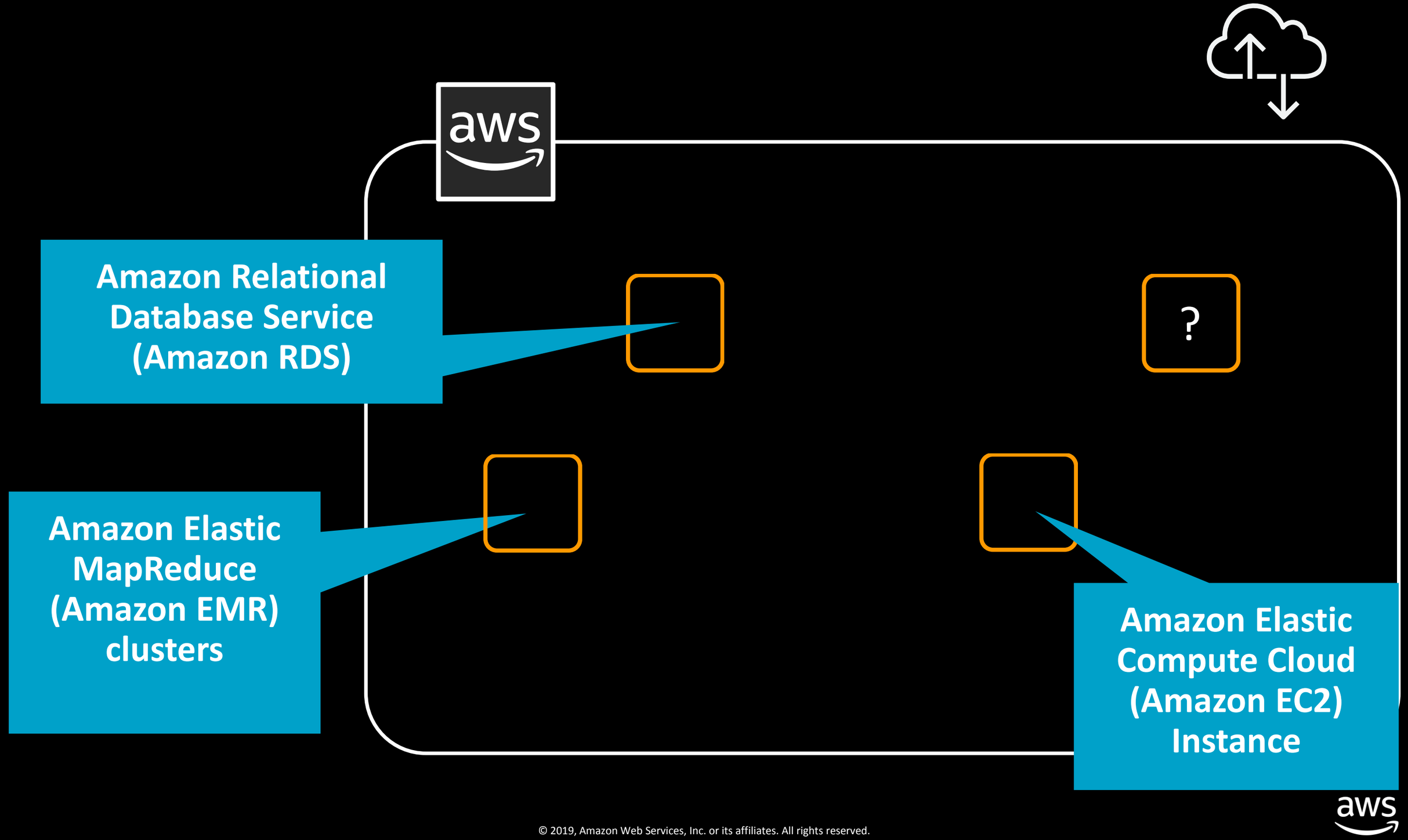
Webinar

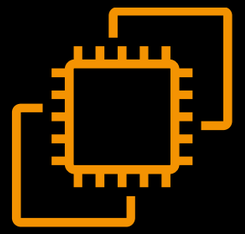
Simplify Traffic Monitoring and Visibility with Amazon VPC Traffic Mirroring

Anoop Dawani
Sr. Product Manager
AWS

aws

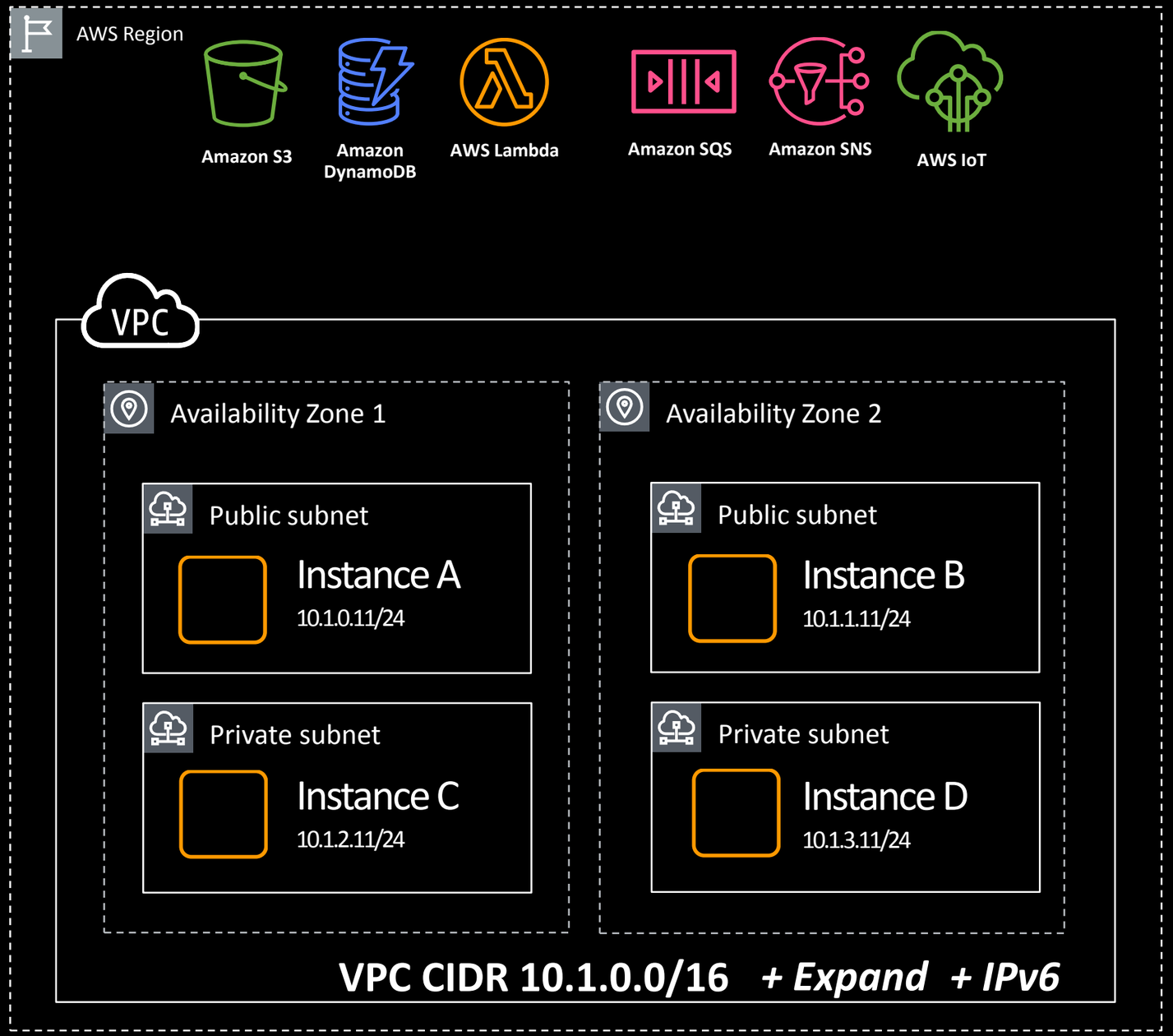


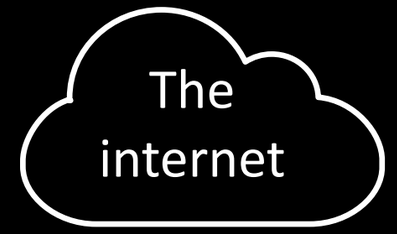
Let’s take a closer look

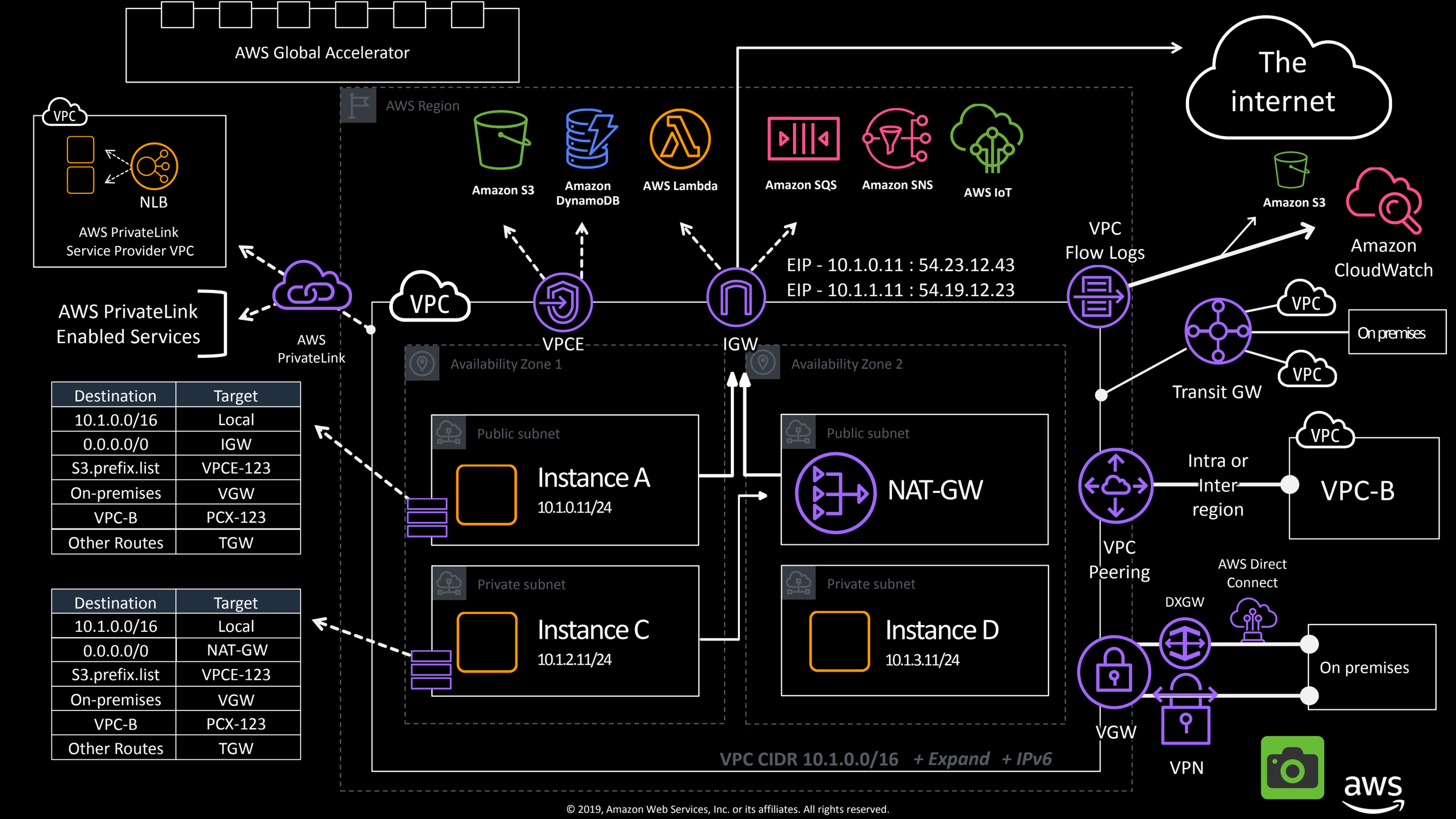


Amazon EC2

Amazon VPC



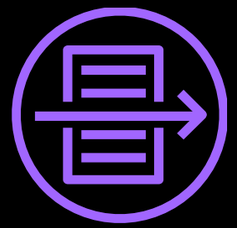




Destination	Target
10.1.0.0/16	Local
0.0.0.0/0	IGW
S3.prefix.list	VPCE-123
On-premises	VGW
VPC-B	PCX-123
Other Routes	TGW

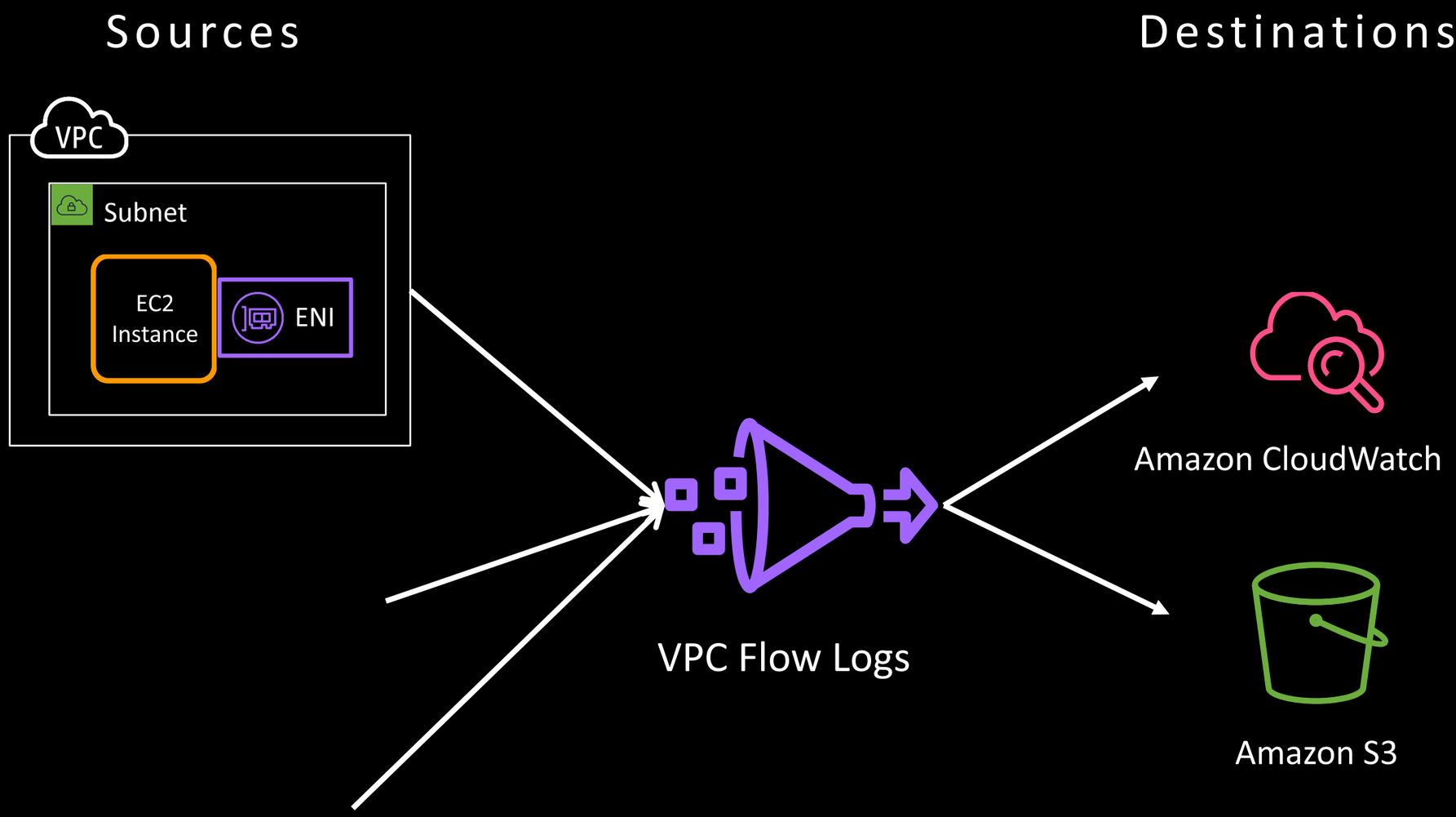
Destination	Target
10.1.0.0/16	Local
0.0.0.0/0	NAT-GW
S3.prefix.list	VPCE-123
On-premises	VGW
VPC-B	PCX-123
Other Routes	TGW

How do I get visibility into my AWS network traffic?



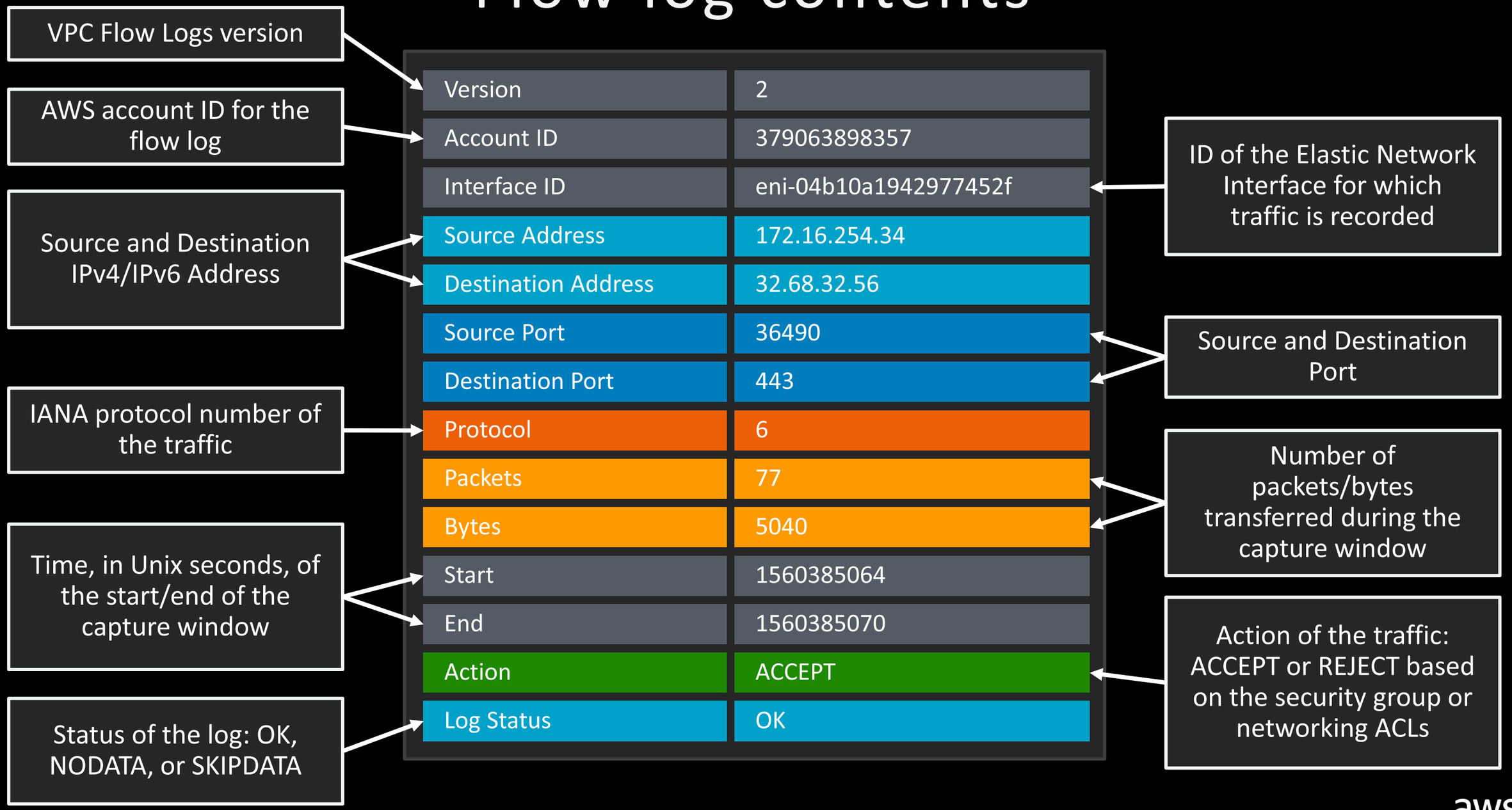
Amazon VPC Flow Logs

Life of a VPC Flow Log



What does a flow log look like?

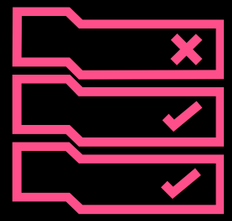
Flow log contents



VPC Flow Logs are a building block

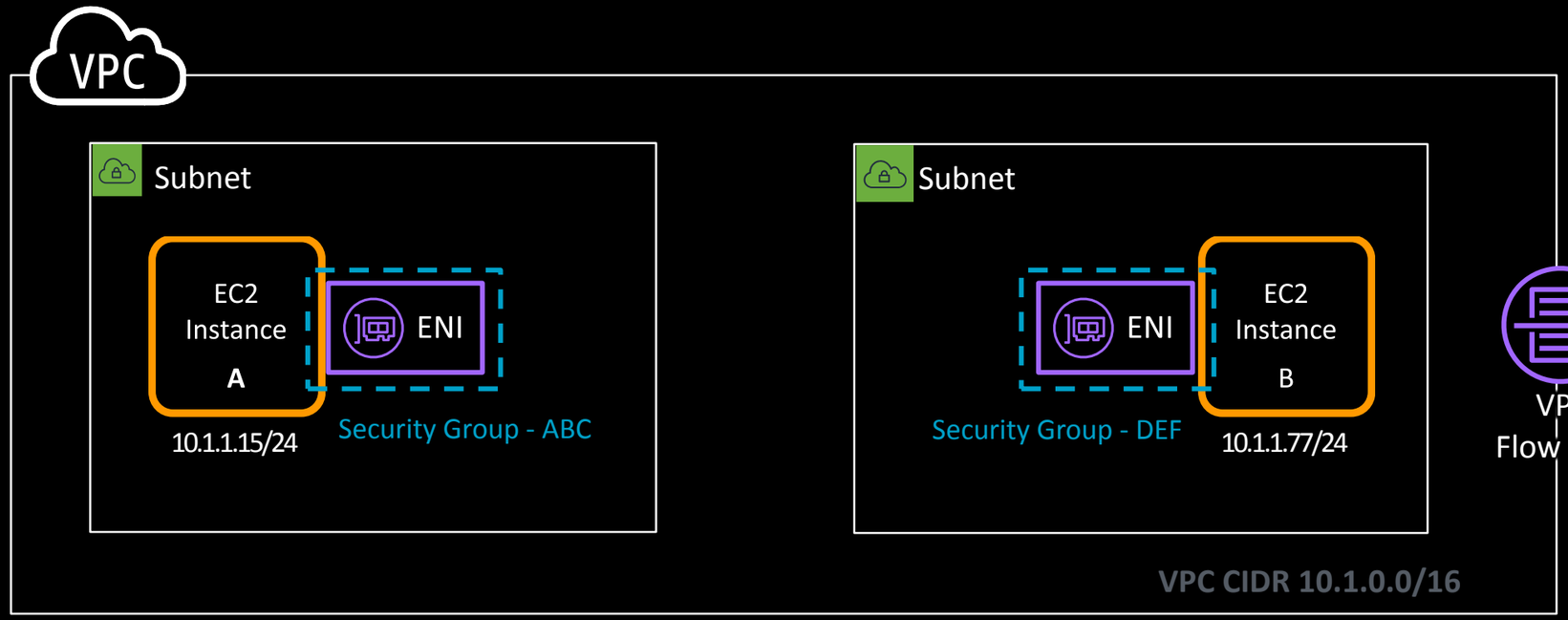
VPC Flow Logs use cases

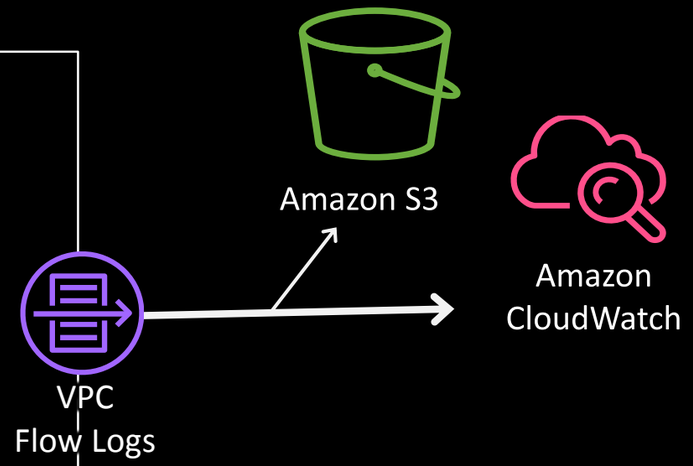
Capture IP traffic going to and from network interfaces in your VPC



Troubleshoot Security Group and Network ACL Rules

Let’s dive into a simple example: Amazon EC2 instance communication within a VPC





Do we have full connectivity?

Security Group	Inbound Rules
SG-ABC	Receive All Traffic from Security Group SG-DEF
	Outbound Rules
	Send All Traffic to Any Destination

Security Group	Inbound Rules
SG-DEF	Receive All Traffic from Security Group SG-ABC
	Outbound Rules
	Send All Traffic to Any Destination

What are we doing here?

Allowing all traffic between the instances

What did that flow log look like?

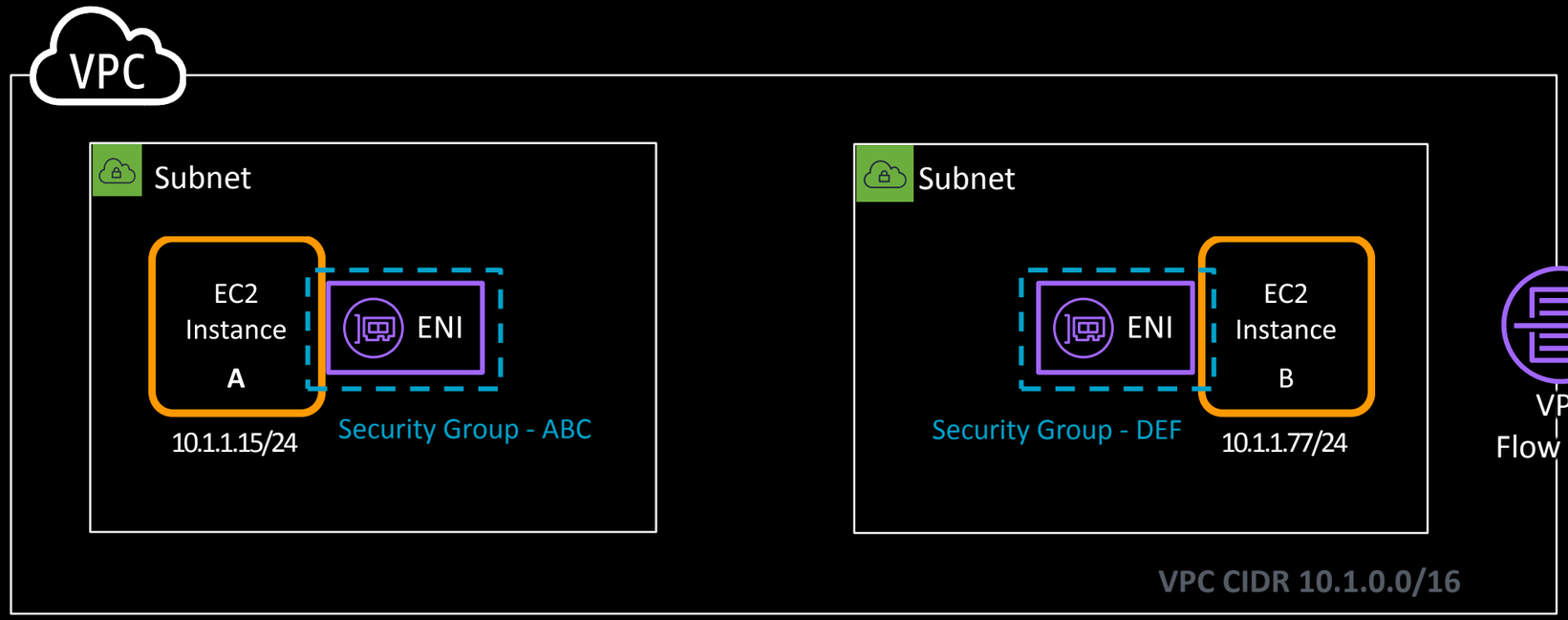
Traffic Flow – Instance A → B

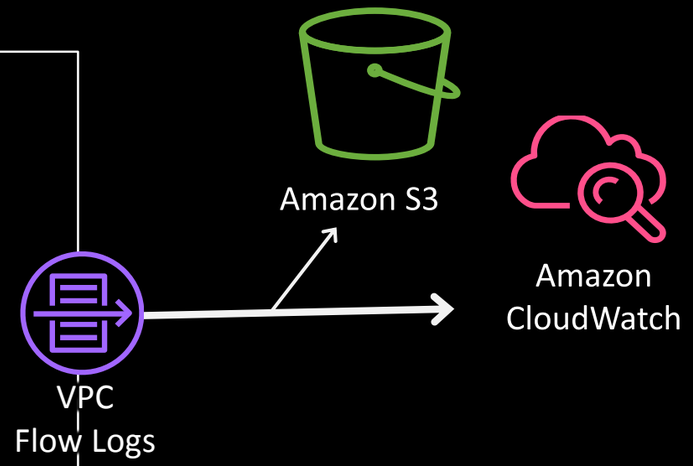
Version	…	ENI	SRC Addr	DES Addr	SRC Port	DES Port	…	Action
2		ENI-04b10a…	10.1.1.15	10.1.1.77	32512	443		ACCEPT

Traffic Flow – Instance B → A

Version	…	ENI	SRC Addr	DES Addr	SRC Port	DES Port	…	Action
2		ENI-17faab…	10.1.1.77	10.1.1.15	443	32512		ACCEPT

…. more flows





Do we have full connectivity?

X

Security Group	Inbound Rules
SG-ABC	Receive All Traffic from 10.5.1.0/24
	Outbound Rules
	Send All Traffic to Any Destination

Security Group	Inbound Rules
SG-DEF	Receive All Traffic from Security Group SG-ABC
	Outbound Rules
	Send All Traffic to Any Destination

What are we doing here?

Instance A can talk to B, but B can't talk to A

What did that flow log look like?

Traffic Flow – Instance A → B

Version	…	ENI	SRC Addr	DES Addr	SRC Port	DES Port	…	Action
2		ENI-04b10a…	10.1.1.15	10.1.1.77	32512	443		ACCEPT

Traffic Flow – Instance B → A

Version	…	ENI	SRC Addr	DES Addr	SRC Port	DES Port	…	Action
2		ENI-17faab…	10.1.1.77	10.1.1.15	443	32512		REJECT

…. more flows

VPC Flow Logs destinations

Sending VPC Flow Logs to …

Amazon CloudWatch

Sending VPC Flow Logs to …

Amazon S3

Analyzing VPC Flow Logs with …

Amazon Athena

https://amzn.to/2Fy4hq5

aws

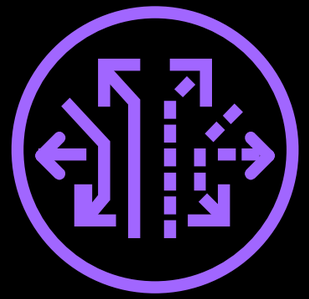
Enriched customer VPC Flow Logs

Watch the full video here
https://bit.ly/2L7BAnt

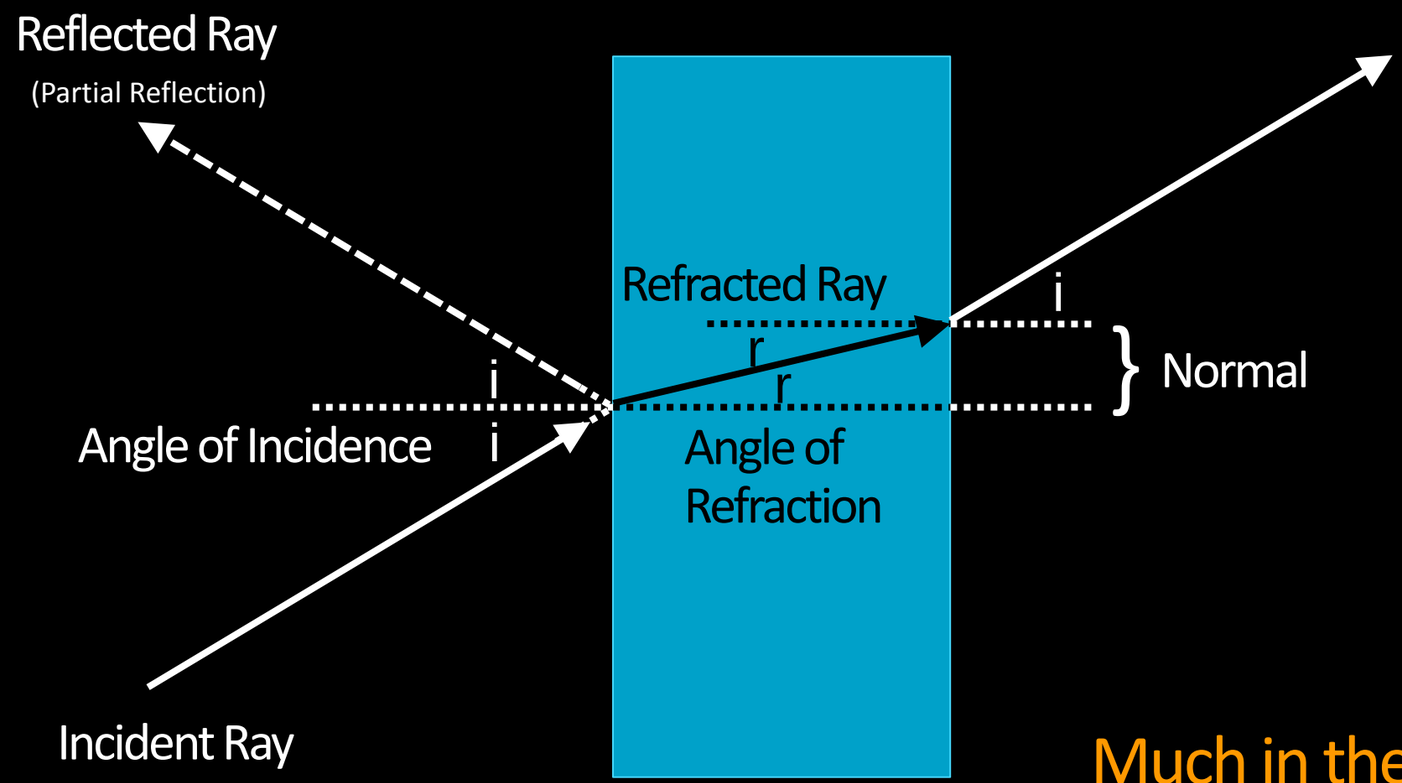
Netflix & Amazon Kinesis Streams Case Study

https://amzn.to/2KgxKJ1

Amazon VPC Traffic Mirroring

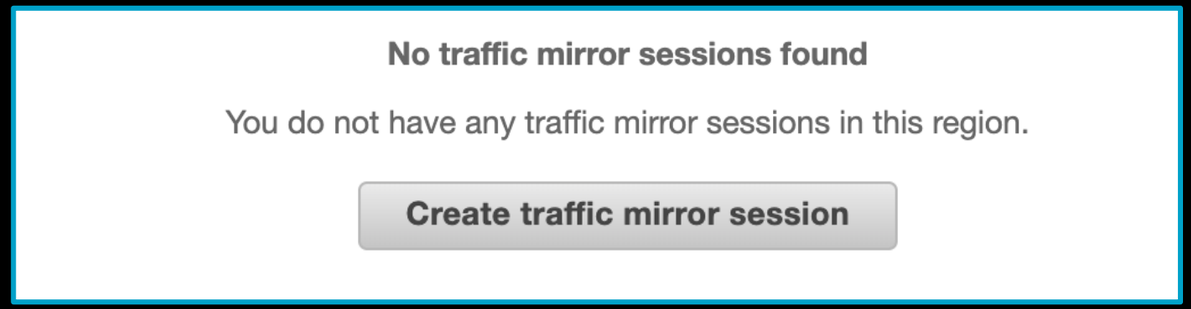


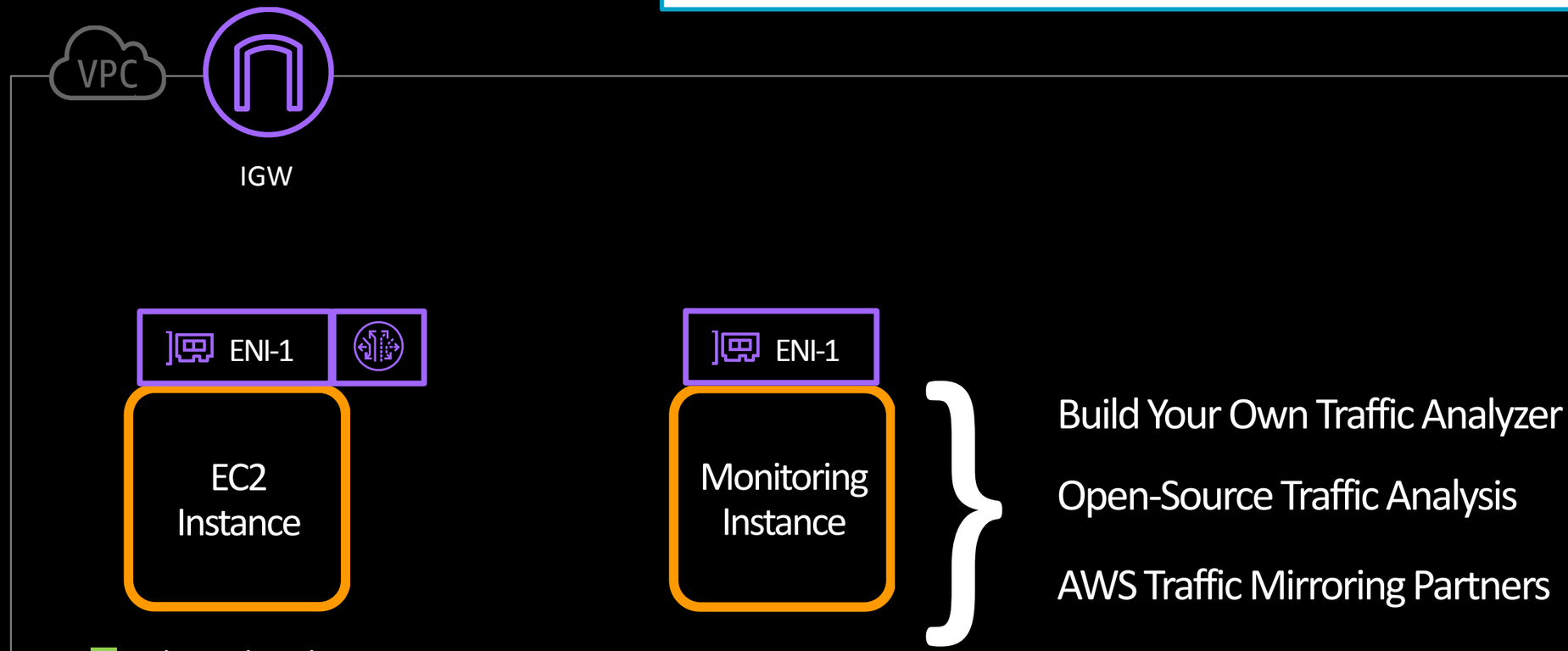
What is Amazon VPC Traffic Mirroring?



Much in the same way…

What is Amazon VPC Traffic Mirroring?



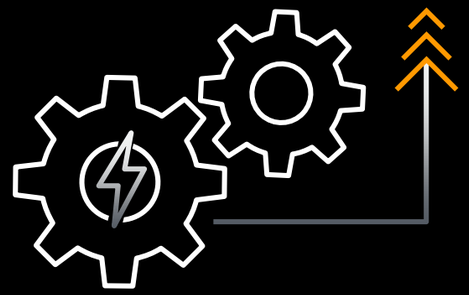


Why have we built VPC Traffic Mirroring?

Customers wanted...

Detection of network and security anomalies

VPC Traffic Mirroring will allow customers to extract traffic of interest from any workload in a VPC and send it to the right tools to detect and respond faster to attacks often missed by traditional log-centric tools



Visibility and Troubleshooting

Through using traffic mirroring, customers can analyze specific traffic patterns to identify any vulnerable “blind spots” or “choke points” between application tiers and/or Amazon EC2 instances

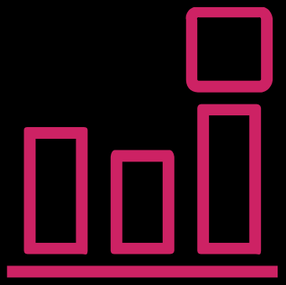
What are some of the key benefits of VPC Traffic Mirroring?

Simplified native operation

Instead of using an agent to have mirroring capability, you now have VPC Traffic Mirroring, natively…

Improved security posture

Through allowing packet capture at the elastic network interface level



Wide range of monitoring options

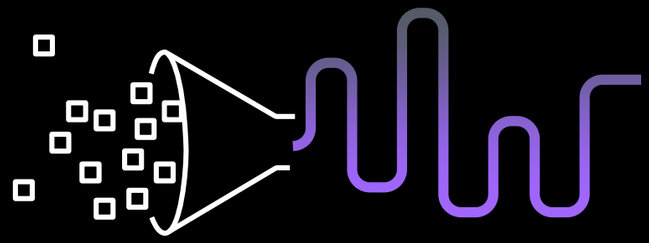
Integrating with multiple tools and partners, VPC Traffic Mirroring allows you to mix and match options

aws

VPC Traffic Mirroring: Three components

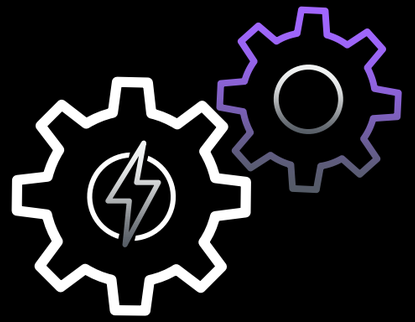
Targets

The destination for mirrored traffic



Filters

A set of rules that define the traffic that is copied in a Traffic Mirror session

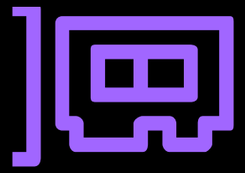


Sessions

An entity that describes traffic mirroring from a source to a target using filters

VPC Traffic Mirror targets

The destination for mirrored traffic



Elastic network interfaces

Network load balancers

VPC Traffic Mirror targets

The destination for mirrored traffic

A Traffic Mirror target can be used in more than one Traffic Mirror session

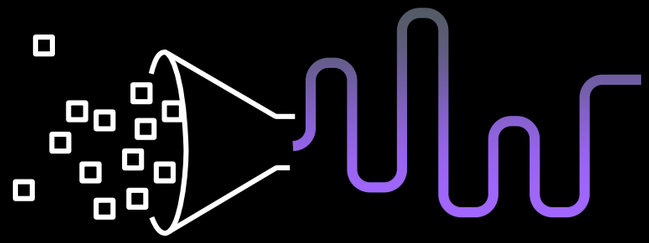
VPC Traffic Mirror targets

The destination for mirrored traffic

Security group associated with target: Allow VXLAN traffic (UDP port 4789) from the Traffic Mirror source

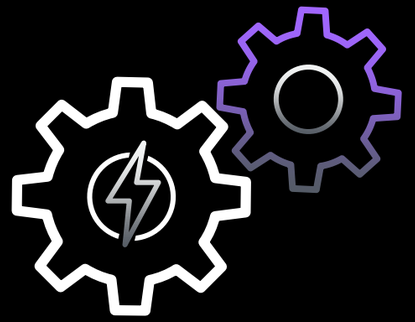
Targets

The destination for mirrored traffic



Filters

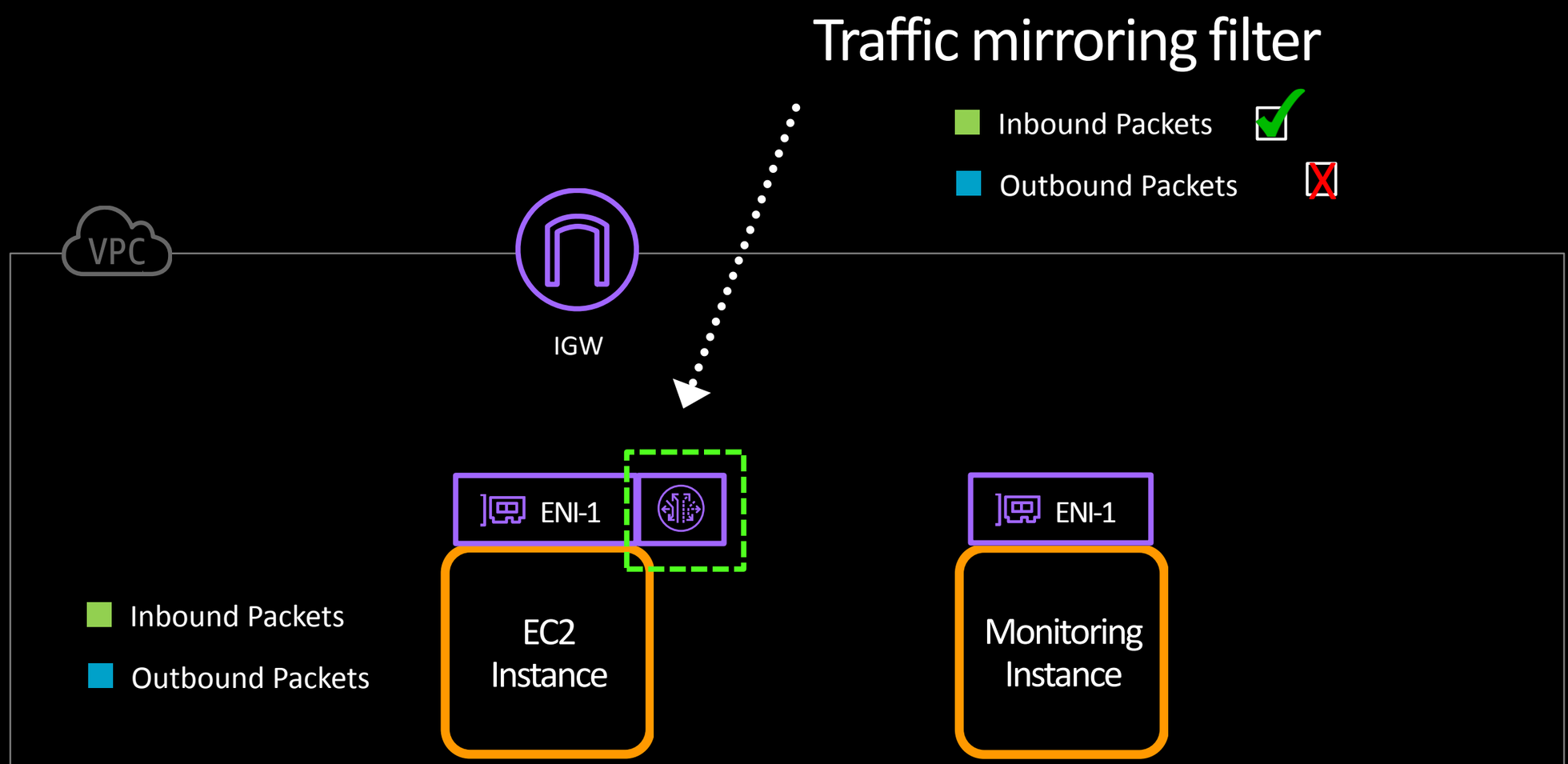
A set of rules that defines the traffic that is copied in a Traffic Mirror session



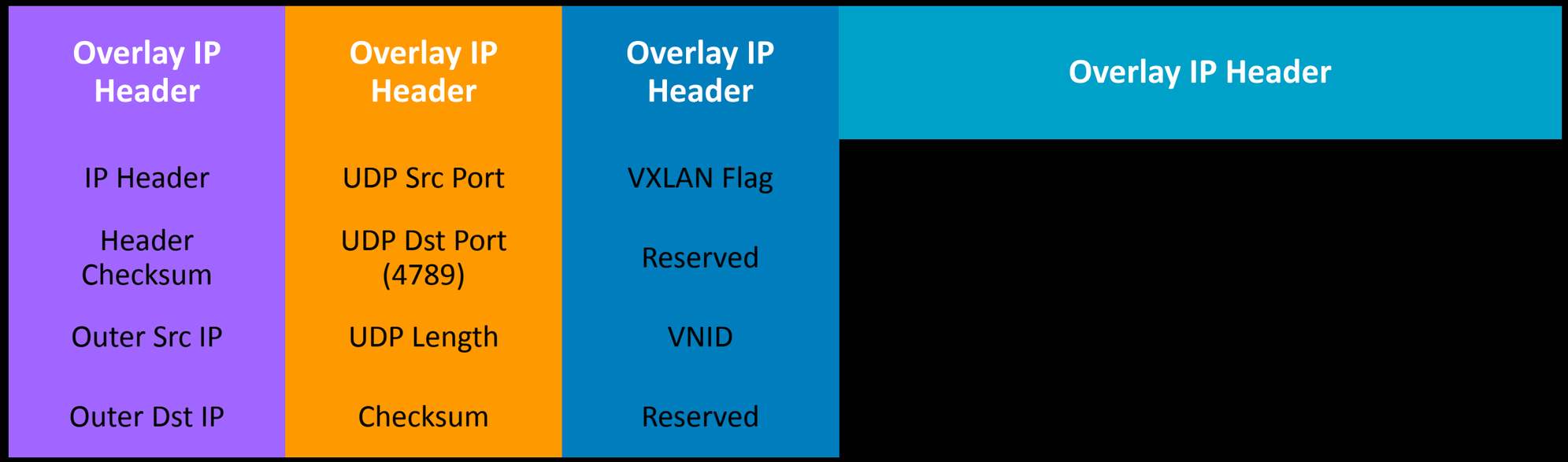
Sessions

An entity that describes traffic mirroring from a source to a target using filters

Traffic mirroring: Traffic filtering



Traffic mirroring: Packet format...



Note: Any packet over 8,946 bytes will be truncated

Traffic mirroring: Traffic filtering

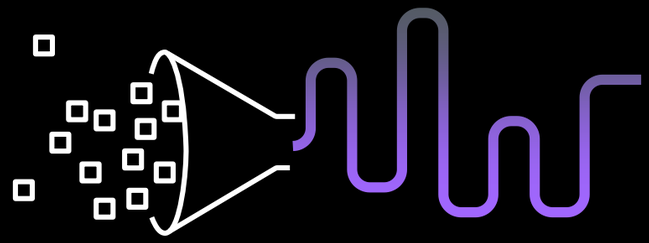
Type	Flow Direction	Protocol	Source IP	Source Port	Dest IP	Dest Port	Truncation
Option	Ingress	TCP	Source IP	Single	Source IP	Single	Y/N
	Egress	UDP	List of CIDRs	Range	List of CIDRs	Range	Sample (Bytes)
	Both	Any					

Traffic mirroring: Traffic filtering

Type	Flow Direction	Protocol	Source IP	Source Port	Dest IP	Dest Port	Truncation
Option	Ingress	TCP	Source IP	Any	My server IP	80	No
	Egress	UDP	0.0.0.0/0	Range	List of CIDRs	Range	Sample (Bytes)
	Both	Any					

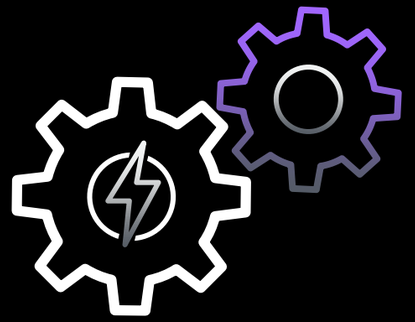
Targets

The destination for mirrored traffic



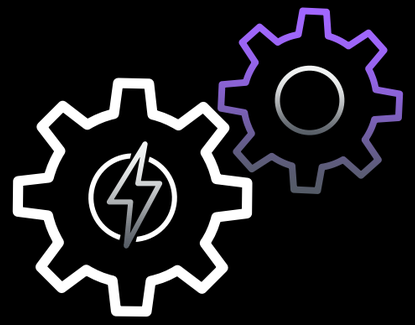
Filters

A set of rules that define the traffic that is copied in a Traffic Mirror session



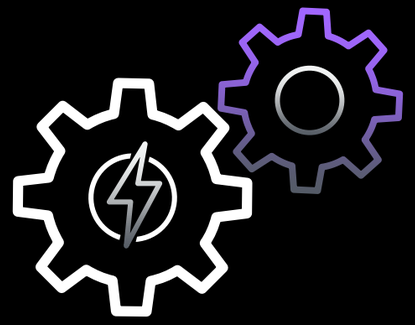
Sessions

An entity that describes traffic mirroring from a source to a target using filters



VPC Traffic Mirror sessions

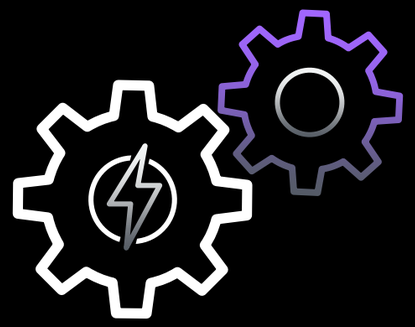
A Traffic Mirror session establishes a relationship between a Traffic Mirror source and a Traffic Mirror target



VPC Traffic Mirror sessions

A Traffic Mirror session has three components:

Source	Target	Filter



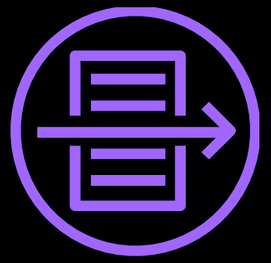
VPC Traffic Mirror sessions

Each Traffic Mirror source can support up to three sessions

Note: Session number determines priority, with the lowest ID given the highest priority—a packet can be mirrored only once

More advanced topics:

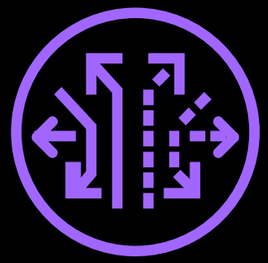
VPC Flow Logs vs. VPC Traffic Mirroring



VPC Flow Logs

- Logs of network flows
- Destination: Amazon S3 or Amazon CloudWatch Logs
- Each record captures the network flow for a specific 5-tuple, for a specific capture window

VS.

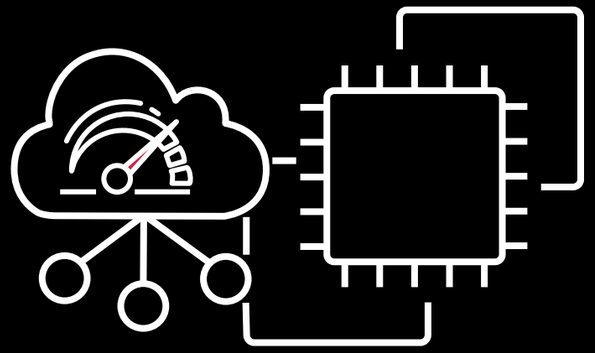


VPC Traffic Mirroring

- Real network packets with the ability to truncate
- Destination: Another ENI or NLB
- Real network packets

aws

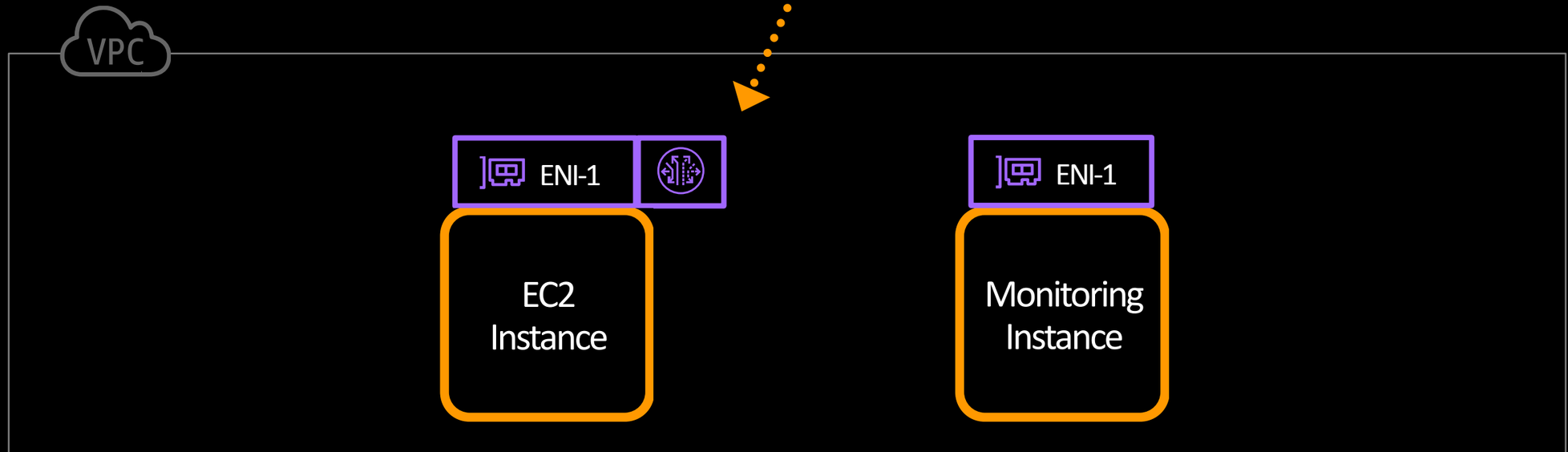
How does traffic mirroring affect the performance of my EC2 instance?

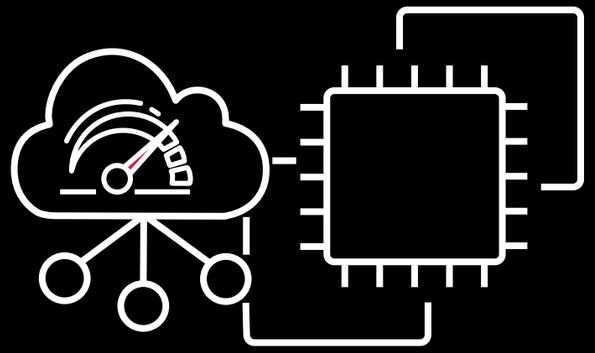


VPC Traffic Mirroring

and EC2 network performance

Instance traffic and mirror traffic **both** count toward your overall instance's performance



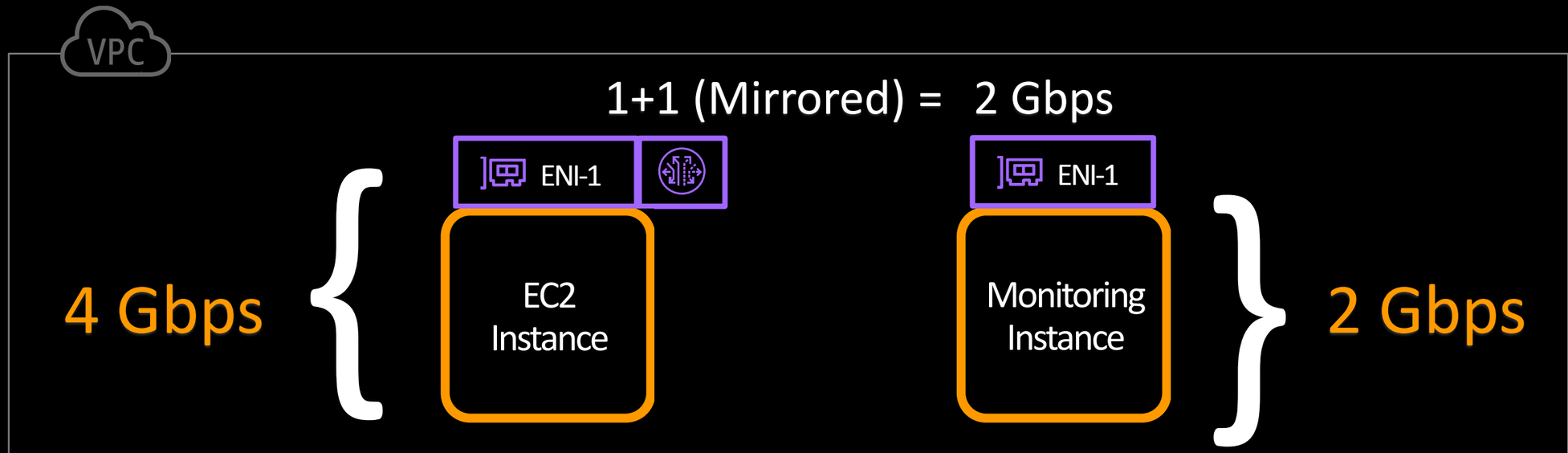


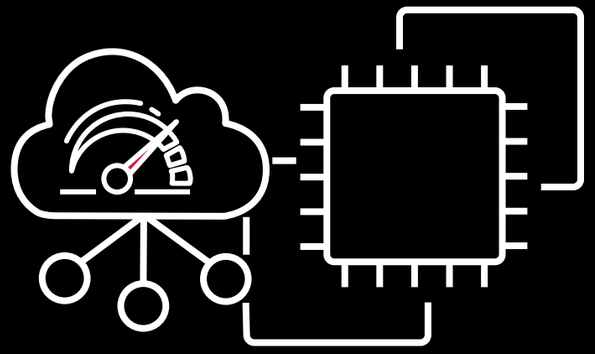
VPC Traffic Mirroring

and EC2 network performance

1 Gbps inbound

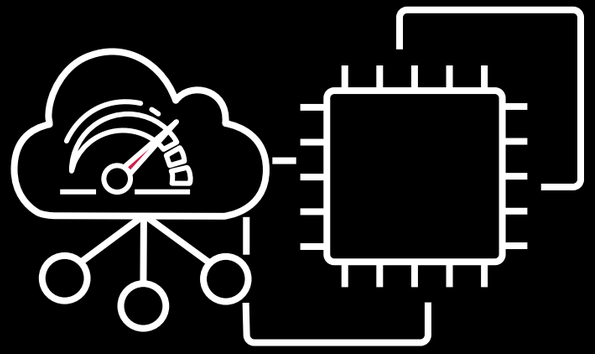
1 Gbps outbound





VPC Traffic Mirroring
and EC2 network performance

Instance right-sizing of sources and targets
is an important consideration

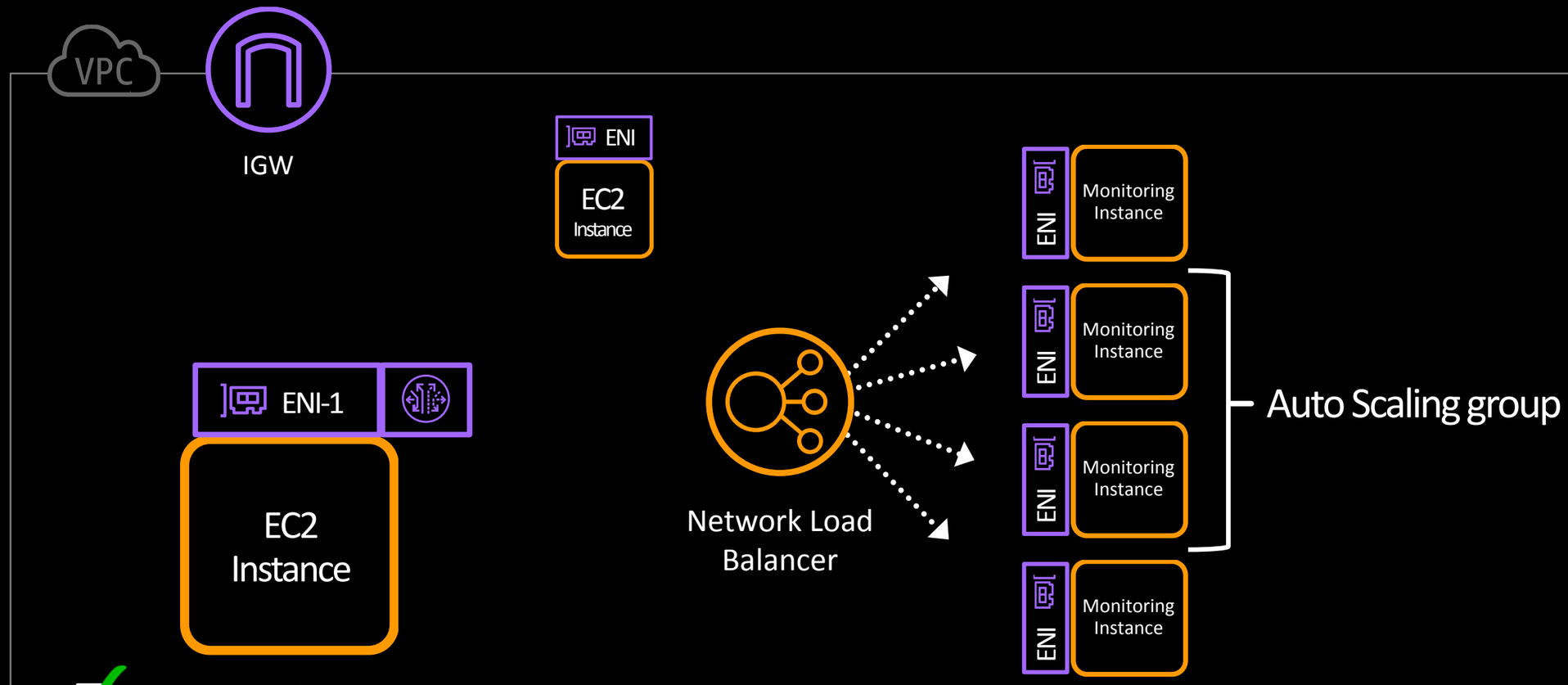


VPC Traffic Mirroring
and EC2 network performance

Note: Production traffic has a higher priority than mirrored traffic when there is traffic congestion

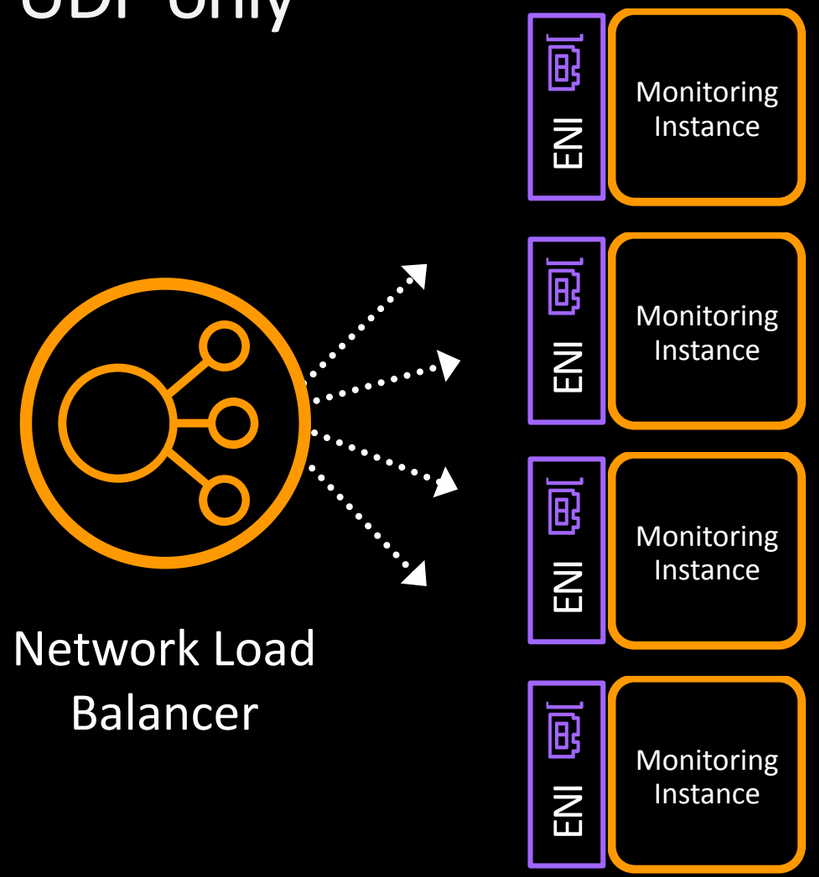
Let’s talk more about scale…

VPC Traffic Mirroring: AWS Auto Scaling



Network Load Balancer: Flow hashing

Traffic Mirroring is UDP only



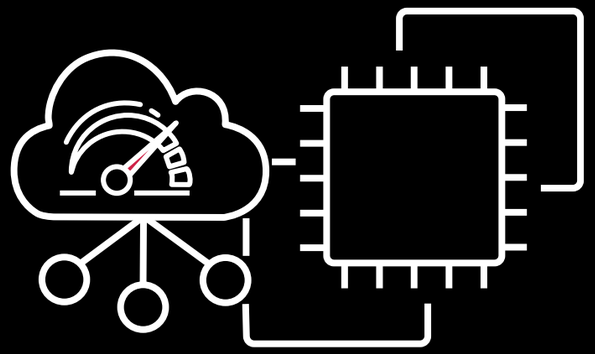
Note: There must be UDP listeners on port 4789.

Network Load Balancer: Flow hashing

For **UDP traffic**, the load balancer selects a target using a flow hash algorithm based on:

- A UDP flow has the same source and destination, so it is consistently routed to a single target throughout its lifetime
- Different UDP flows have different **source ports**, so they can be routed to different targets

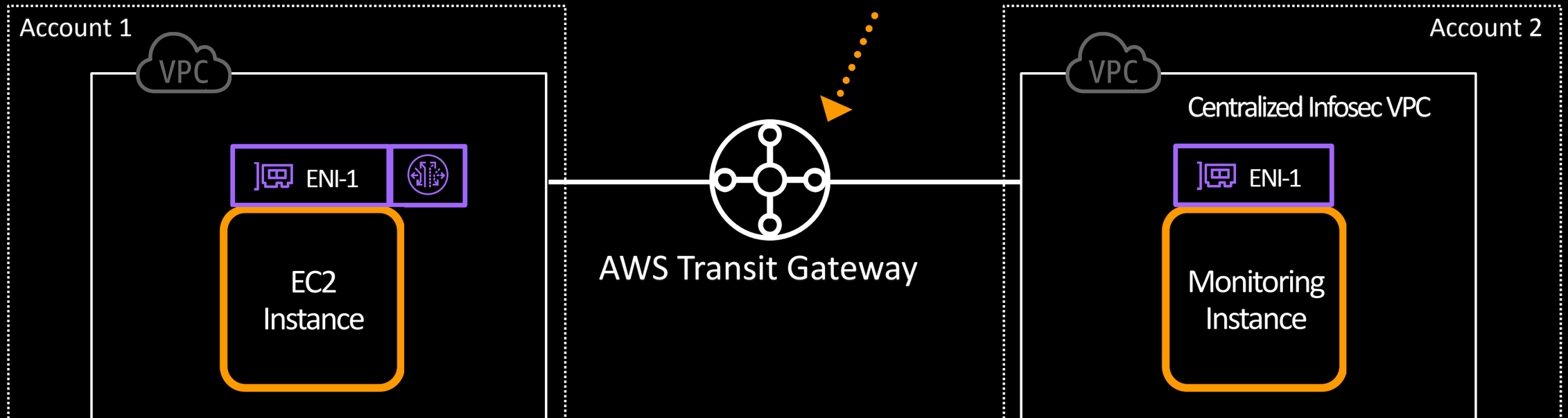
Decentralizing your traffic mirroring deployment



VPC Traffic Mirroring

Decentralized

With routing connectivity, Traffic Mirror sources and destinations can be decentralized



Let’s check it out in the console

AWS Management Console

AWS services

Find Services

You can enter names, keywords or acronyms.

Example: Relational Database Service, database, RDS

Recently visited services

- VPC
- S3
- EC2
- Direct Connect

All services

Build a solution

Get started with simple wizards and automated workflows.

Access resources on the go

Access the Management Console using the AWS Console Mobile App. Learn more

Explore AWS

Open Distro for Elasticsearch

A 100% open-source, community driven distribution of Elasticsearch with enterprise-grade security and alerting features Learn more

Data Lake Storage

Build your data lake on the most secure, durable, and scalable storage. Learn more

VPC Dashboard

Filter by VPC:

Select a VPC

Virtual Private Cloud

Your VPCs

Subnets

Route Tables

Internet Gateways

Egress Only Internet Gateways

Mirror Sessions

Endpoints

Endpoint Services

NAT Gateways

Peering Connections

Security

Launch VPC Wizard | Launch EC2 Instances

Note: Your Instances will launch in the US East (N. Virginia) region.

Resources by Region

Refresh Resources

You are using the following Amazon VPC resources

VPCs	N. Virginia 5	NAT Gateways	N. Virginia 0
Subnets	N. Virginia 14	VPC Peering Connections	N. Virginia 1
Route Tables	N. Virginia 8	Network ACLs	N. Virginia 5
Internet Gateways	N. Virginia 5	Security Groups	N. Virginia 40
Egress-only Internet Gateways	N. Virginia 0	Customer Gateways	N. Virginia 5
DHCP options sets	N. Virginia 1	Virtual Private Gateways	N. Virginia 5

(All: See all regions)

Service Health

Current Status	Details
Amazon EC2 - US East (N. Virginia)	Service is operating normally

View complete service health details

Account Attributes

Resource ID length management

Additional Information

VPC Documentation

All VPC Resources

Forums

Report an Issue

Site-to-Site VPN Connections

Amazon VPC enables you to use your own isolated resources within the AWS cloud, and then connect those resources directly to your own datacenter using industry-standard encrypted IPsec VPN connections.

Create VPN Connection

Virtual Private Gateways

Site-to-Site VPN Connections

Client VPN Endpoints

Transit Gateways

Transit Gateways

Transit Gateway Attachments

Transit Gateway Route Tables

Traffic Mirroring

Mirror Sessions

Mirror Targets

Mirror Filters

Traffic mirror sessions

Actions | Create traffic mirror session

1

Name	Session ID	Description	Source	Target	Session number	Filter	Owner

No traffic mirror sessions found

You do not have any traffic mirror sessions in this region.

Create traffic mirror session

Primary network interface
eni-2003bd0e

Network Interface for Transit Gateway Attachment tgw-attach-0389f25d2a1d76612
eni-0dce5735551cef549

Primary network interface
eni-0da63cb578d3fe980

Primary network interface
eni-01edacf1243d44d26

Primary network interface
eni-0d851b8589ebb10c2

Primary network interface
eni-65f26879

test rax
eni-fc449232

Primary network interface
eni-00a14e97c2c48f460

eni-20003bd0e

Only network interfaces of type "interface" are allowed.

Mirror target

A network interface, or a network load balancer that is the destination for mirrored traffic.

tmt-0cd45c2defe98a55b

Create target

traffic-mirroring-target
tmt-0cd45c2defe98a55b
network-interface eni-00a14e97c2c48f460

Additional settings

Set priority, packet length, etc ...

Session number

The order sessions for the same resource are evaluated

1

Create traffic mirror session

Session settings

Set description, source, and target

Name tag - *optional*

traffic-mirroring-session

Description - *optional*

traffic-mirroring-session

Mirror source

The resource that you want to monitor.

eni-20003bd0e

Only network interfaces of type "interface" are allowed.

Mirror target

A network interface, or a network load balancer that is the destination for mirrored traffic.

tmt-0cd45c2defe98a55b

Create target

Additional settings

Set priority, packet length, etc ...

Session number

The order sessions for the same resource are evaluated

1

Additional settings

Set priority, packet length, etc .

Session number

The order sessions for the same

1

Number between 1 and 32766

VNI - *optional*

The unique VXLAN network ide

Number between 0 and 167772

Packet length - *optional*

The number of bytes in each pa

If not specified, the entire packe

Filter

Determines what traffic gets mi

Select mirror filter

Create filter

Tags - *optional*

Apply tags to your resources to

A tag consists of a case-sensiti

Add tag

Modify traffic mirror filter rule

Inbound rule

Rule number

100

Rule action

Select the action to take.

accept

Protocol

Select the IP traffic protocol.

TCP (6)

Source port range - *optional*

Enter the source IP address port range.

Destination port range - *optional*

Enter the destination IP address port range.

8080

Source CIDR block

Enter the source CIDR block.

0.0.0.0/0

Destination CIDR block

Enter the destination CIDR block.

0.0.0.0/0

Description - *optional*

Describe the rule.

Cancel Create

Virtual Private Gateways

Site-to-Site VPN Connections

Client VPN Endpoints

Transit Gateways

Transit Gateways

Transit Gateway Route Tables

Traffic Mirroring

Mirror Sessions

Mirror Targets

Mirror Filters

Traffic mirror sessions

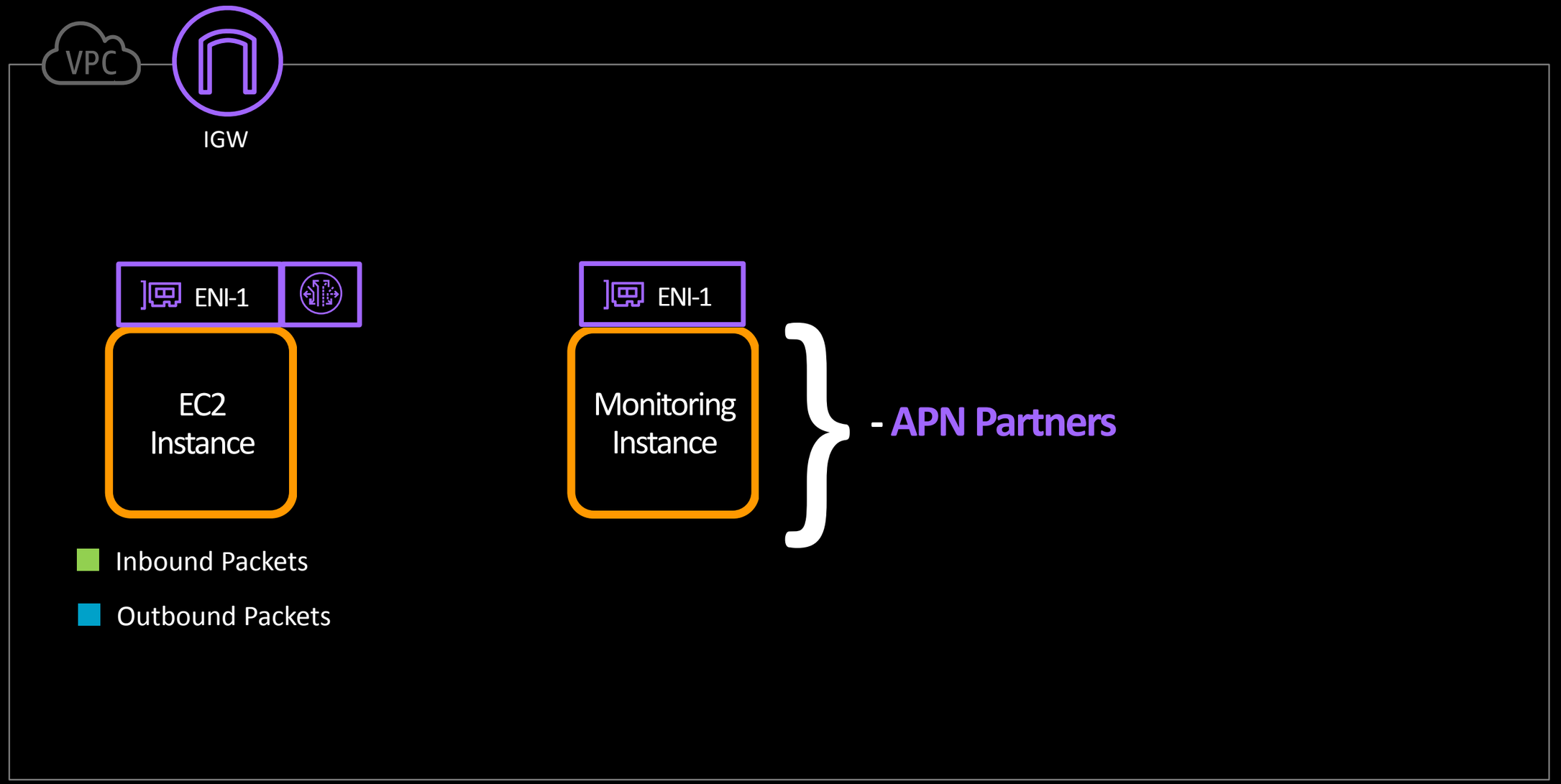
Actions | Create traffic mirror session

Name	Session ID	Description	Source	Target	Session number	Filter	Owner
traffic-mirroring-session	tms-033093c9c3c74cac8	traffic-mirroring-session	eni-058fb2a77a6501061	tmt-0cd45c2defe98a55b	1	tmf-04811c2d05c564045	450545058648

traffic-mirroring-session tms-033093c9c3c74cac8 traffic-mirroring-session eni-058fb2a77a6501061 tmt-0cd45c2defe98a55b 1 tmf-04811c2d05c564045 450545058648

What do you actually ***do*** with the packets?

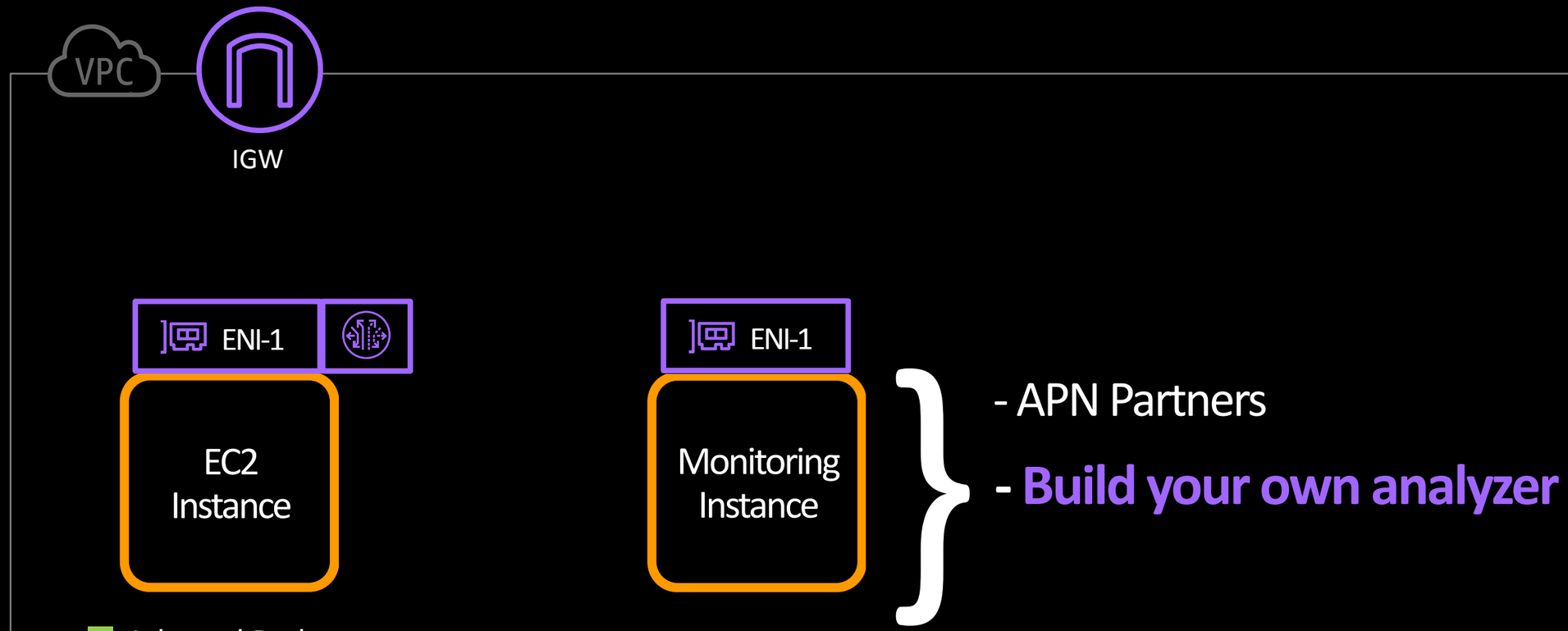
What do you actually *do* with the packets?



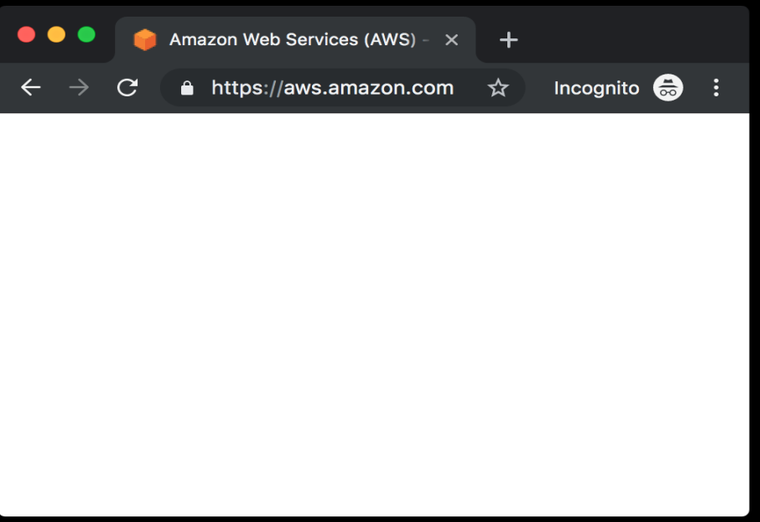
aws

aws

What do you actually *do* with the packets?



What do you actually *do* with the packets?



User

Make an HTTP request to http://3.2.5.4:8080

Capture HTTP Requests both in/out of an EC2 Instance

VPC

ENI-1

EC2 Instance

ENI-1

Monitoring Instance

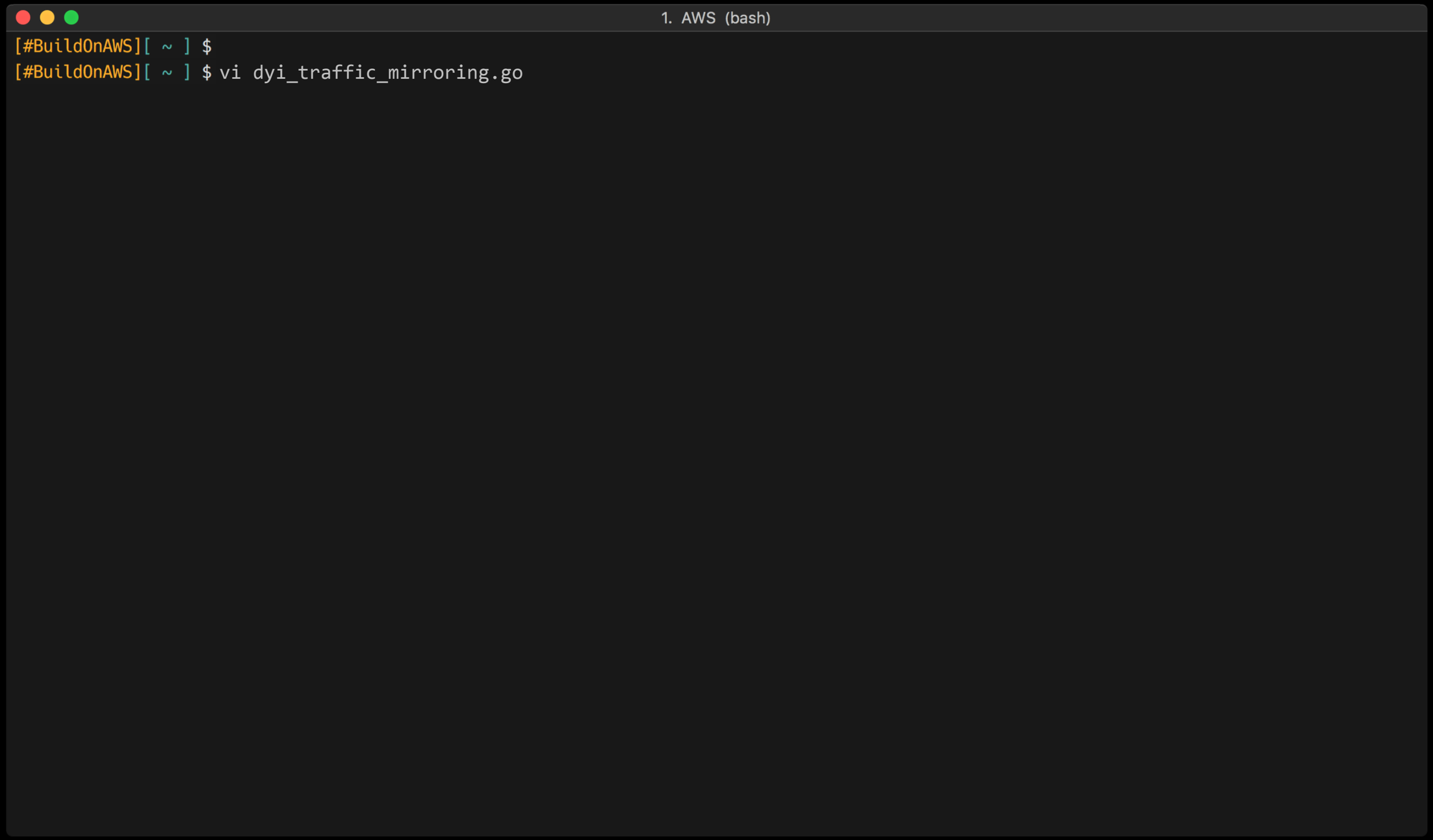
- APN Partners
- **Build your own analyzer**

Inbound Packets

Outbound Packets

aws

Let’s build the analyzer



1. AWS (bash)

```
1  package main
2
3  import (
4          "fmt"
5          "github.com/google/gopacket"
6          "github.com/google/gopacket/layers"
7          "github.com/google/gopacket/pcap"
8          "strings"
9  )
10
11 func main() {
12         // Agent code ...
13 }
14
15
16
17
18
19
20
21
22
23
24
```

"fmt" allows you to print messages to the screen

"gopacket" provides packet decoding for the Go language

"layers" provides decoding layers for many common protocols

"pcap" allows you to read packets off the wire or from PCAP files

"strings" is a helper library to deal with strings

This is where we will write our code to capture packets that were mirrored to this server

What are we doing here?

Importing libraries needed for the PCAP capture

aws

Let’s define the analyzer code

1. AWS (bash)

```
1  package main
2
3  import (
4          // imported packages ...
5  )
6
7  func main() {
8          // Bind to an EC2 Elastic Network Interface
9          handle, err := pcap.OpenLive("ens5", 9001, true, pcap.BlockForever)
10         if err != nil {
11             panic(err)
12         }
13         // Filter for UDP Packets
14         // VPC Traffic Mirroring sends mirrored traffic as encapsulated UDP packets
15         filter := "udp"
16         if err := handle.SetBPFFilter(filter); err != nil {
17             panic(err)
18         }
19         // Create a packet source from the ENI with an link type of Ethernet
20         packetSource := gopacket.NewPacketSource(handle, handle.LinkType())
21         // but wait, there's more ...
22 }
23
```

What are we doing here?

Starting a PCAP capture on an EC2 elastic network interface

aws

Let’s grab the mirrored packet

1. AWS (bash)

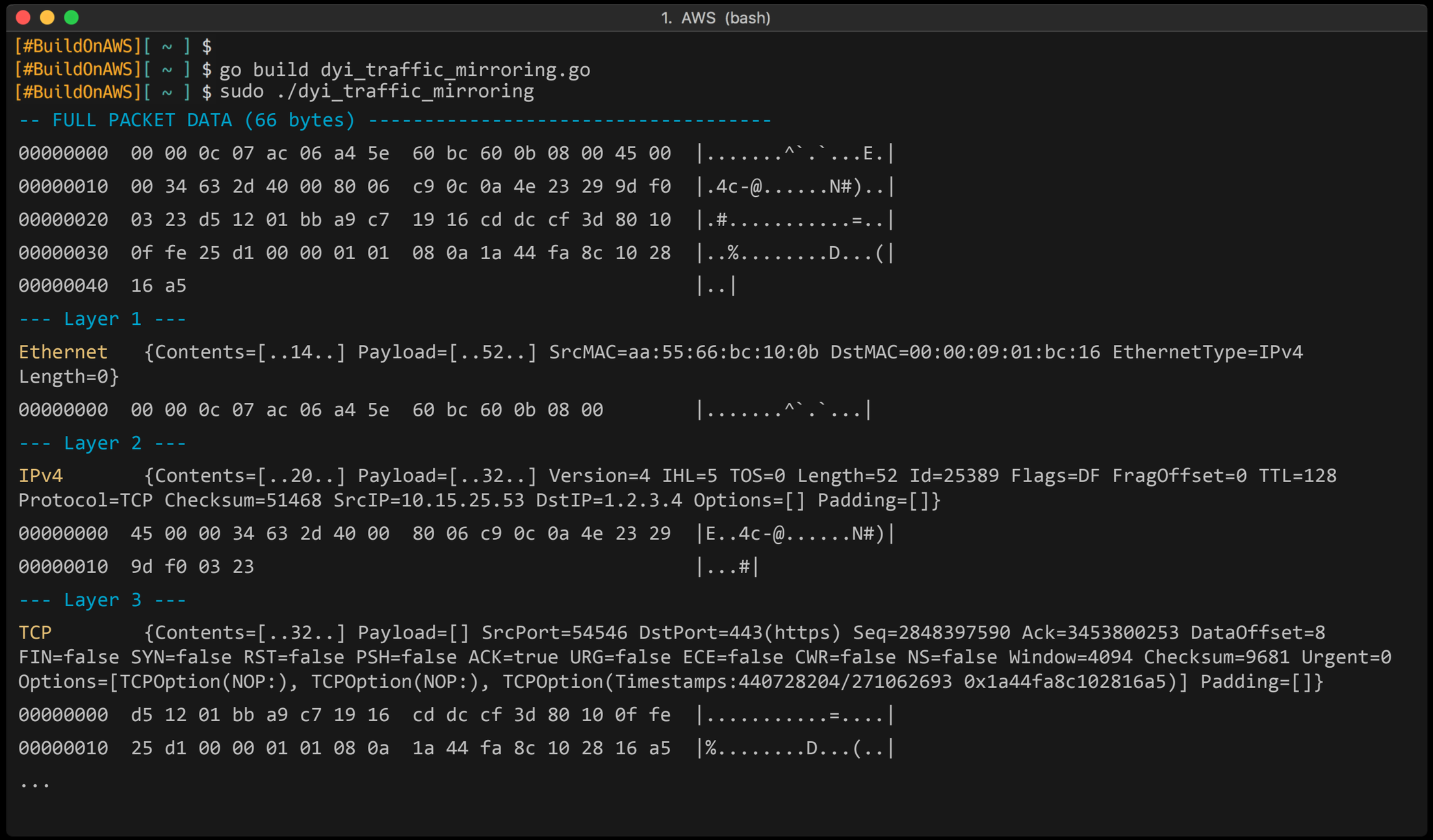
```
22    // Grab every packet in/out of the Elastic Network Interface (ENI)
23    for overlayPacket := range packetSource.Packets() {
24        // Check if VXLAN header is inside the packet
25        vxlanLayer := overlayPacket.Layer(layers.LayerTypeVXLAN)
26        if vxlanLayer != nil {
27            // Grab the VXLAN packet
28            vxlanPacket, _ := vxlanLayer.(*layers.VXLAN)
29            // Create a new packet with the mirrored Ethernet Frame that is inside the VXLAN packet
30            packet := gopacket.NewPacket(vxlanPacket.LayerPayload(), layers.LayerTypeEthernet, gopacket.Default)
31            // Print out the contents of the mirrored packet
32            fmt.Println(packet.Dump())
33        }
34    }
35
36
37
38
39
40
41
42
43
44
```

What are we doing here?

Grabbing the mirrored source Ethernet frame from an VXLAN IP packet

aws

Let’s build it and see what output we get



Let’s capture HTTP requests

1. AWS (bash)

```
22    // Grab every packet in/out of the Elastic Network Interface (ENI)
23    for overlayPacket := range packetSource.Packets() {
24        // Check if VXLAN header is inside the packet
25        vxlanLayer := overlayPacket.Layer(layers.LayerTypeVXLAN)
26        if vxlanLayer != nil {
27            // Grab the VXLAN packet
28            vxlanPacket, _ := vxlanLayer.(*layers.VXLAN)
29            // Create a new packet with the mirrored Ethernet Frame that is inside the VXLAN packet
30            packet := gopacket.NewPacket(vxlanPacket.LayerPayload(), layers.LayerTypeEthernet, gopacket.Default)
31            // Check if contents in the Application Layer exists in the mirrored Packet
32            applicationLayer := packet.ApplicationLayer()
33            if applicationLayer != nil {
34                // Check if the Application Payload contains HTTP requests
35                if strings.Contains(string(applicationLayer.Payload()), "HTTP") {
36                    // Print out the SRC IP and DEST IP
37                    netFlow := packet.NetworkLayer().NetworkFlow()
38                    src, dst := netFlow.Endpoints()
39                    fmt.Printf("[SRC HOST IP: %s] [DEST HOST IP: %s]\n", src, dst)
40                    // Print out the HTTP Request
41                    fmt.Printf("%s\n", applicationLayer.Payload())
42                }
43            }
44        }
45    }
```

What are we doing here?

Grabbing the HTTP request from the application payload

aws

Let’s build it and see what output we get

Traffic Mirroring Destination

1. AWS (bash)

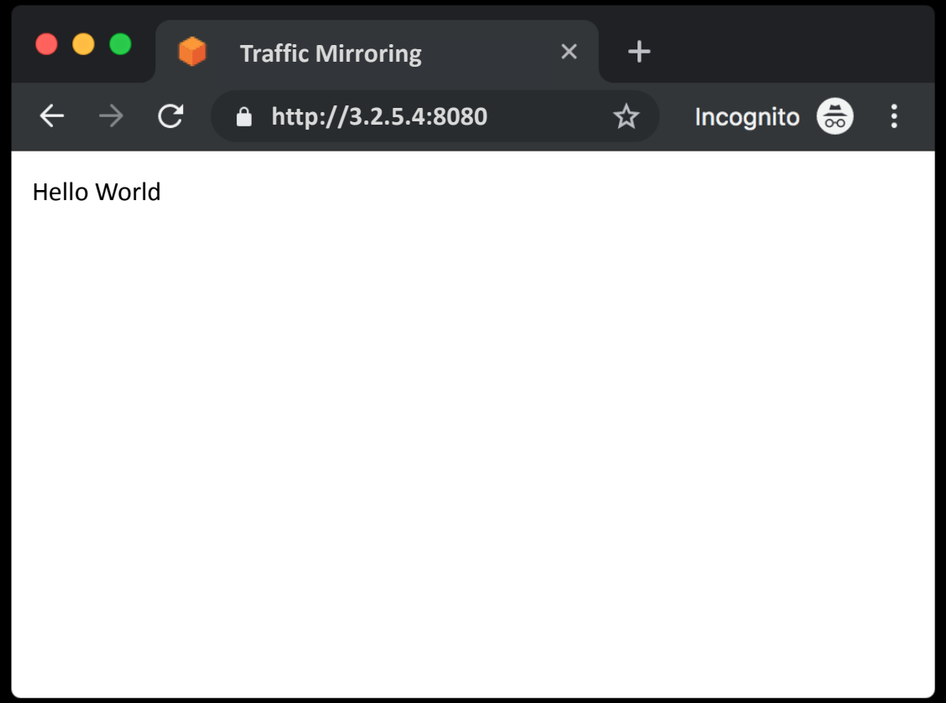
```
[#BuildOnAWS][ ~ ] $ go build dyi_traffic_mirroring.go
[#BuildOnAWS][ ~ ] $ sudo ./dyi_traffic_mirroring
[SRC HOST IP: 5.4.3.2]
[DEST HOST IP: 10.0.0.46]
GET / HTTP/1.1
Host: 3.2.5.4:8080
User-Agent: curl/7.54.0
Accept: */*

[SRC HOST IP: 10.0.0.46]
[DEST HOST IP: 5.4.3.2]
HTTP/1.0 200 OK

[SRC HOST IP: 10.0.0.46]
[DEST HOST IP: 5.4.3.2]
Server: SimpleHTTP/0.6 Python/2.7.12
Date: Tues, 25 Jun 2019 01:37:11 GMT
Content-type: text/html
Content-Length: 76
Last-Modified: Wed, 19 Jun 2019 00:10:07 GMT

<html>
    <head><title>Traffic Mirroring</title></head>
    <body><p>Hello World</p></body>
</html>
```

Traffic Mirroring Source

1. AWS (bash)

```
[#BuildOnAWS][ ~ ] $ python –m SimpleHTTPServer 8080
Serving HTTP on 0.0.0.0 port 8080
5.4.3.2 - - [21/Jun/2019 20:31:32] "GET /index.html
HTTP/1.1" 200 -
```

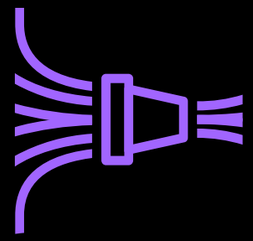
Client Accessing The Web Server



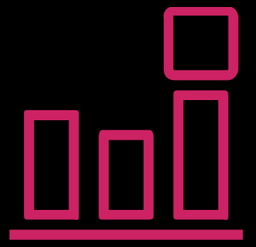
Done! Now what can I do from here?

Send the captured traffic to…

Amazon S3



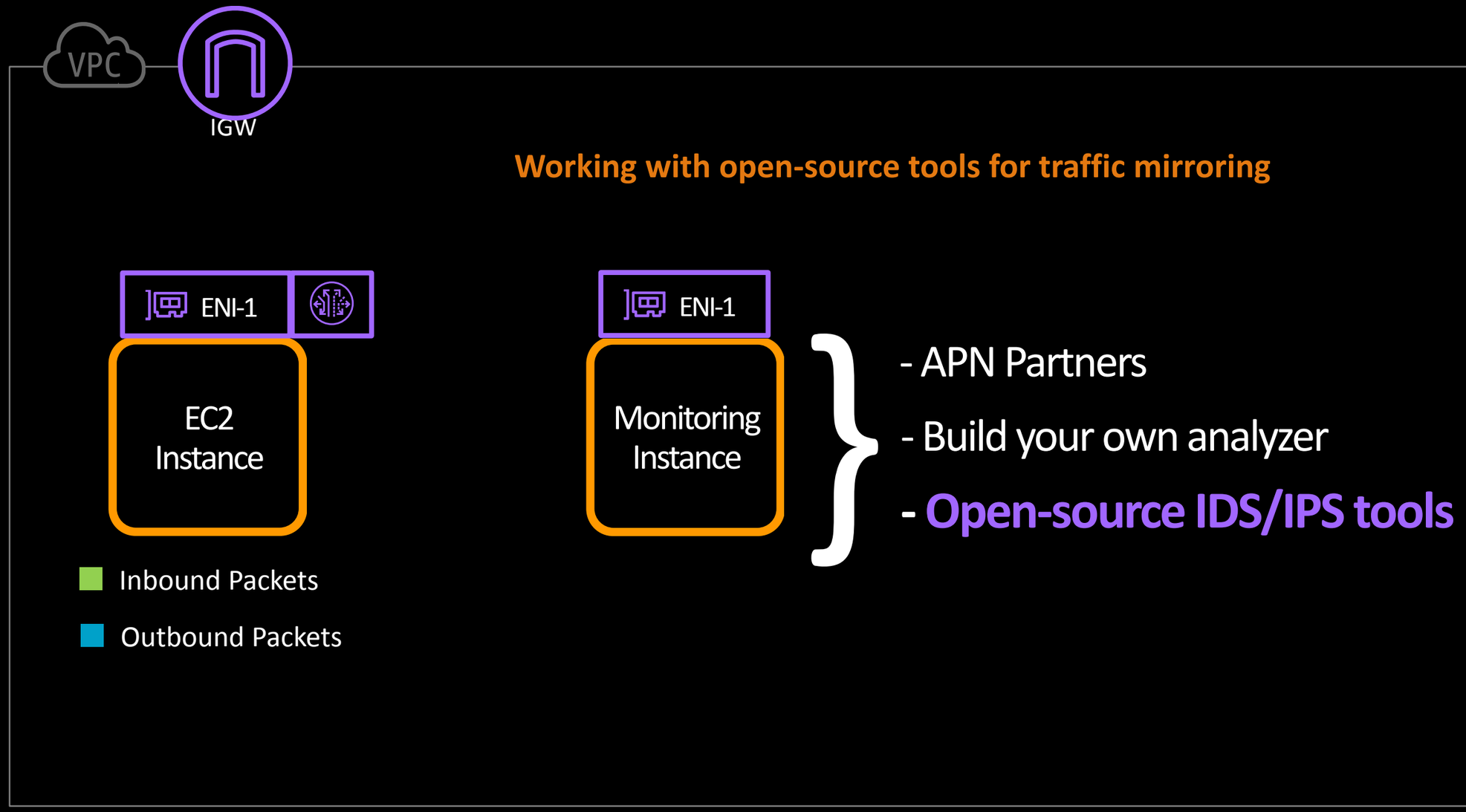
Amazon Kinesis

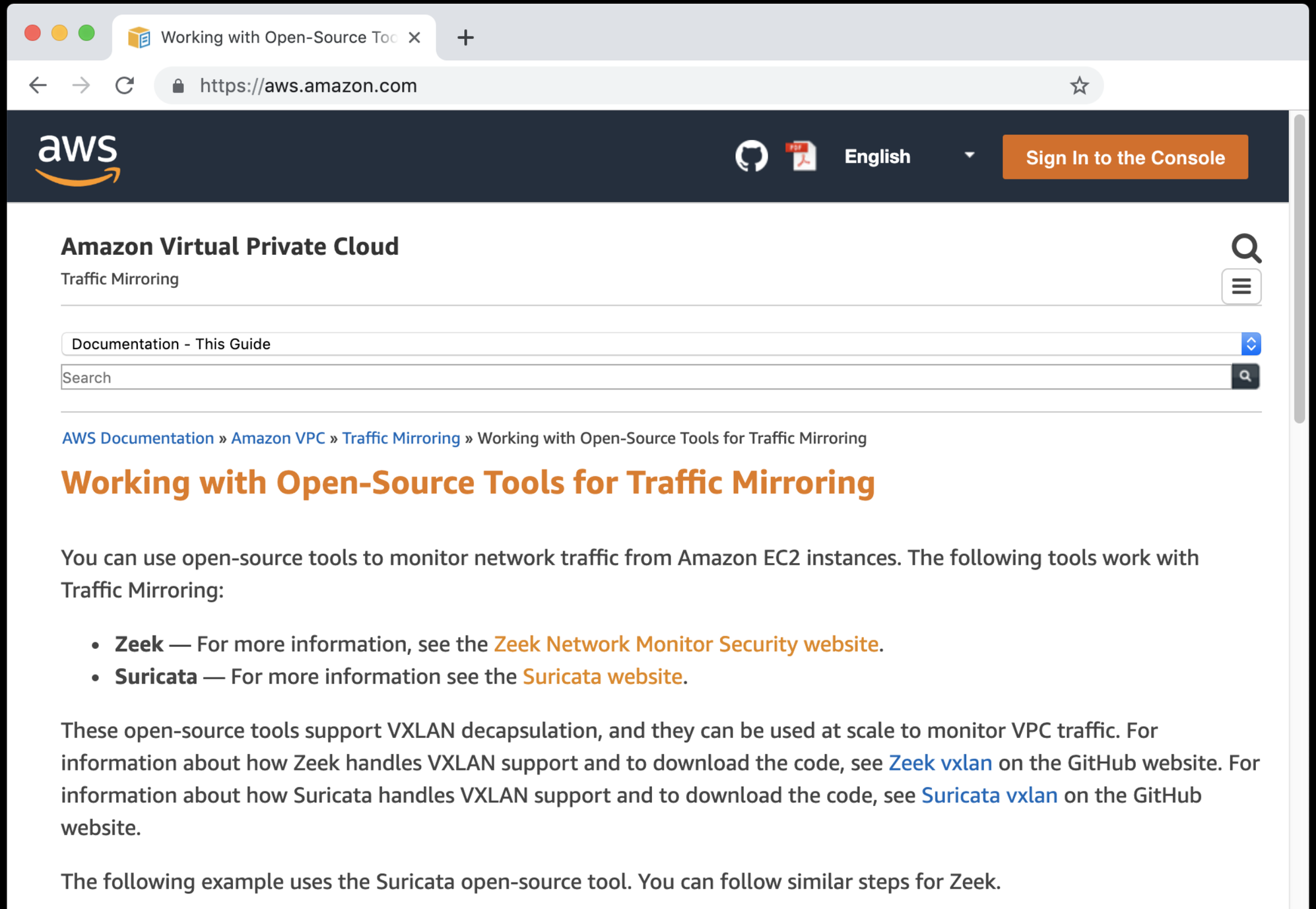


Amazon CloudWatch

aws

What do you actually *do* with the packets?



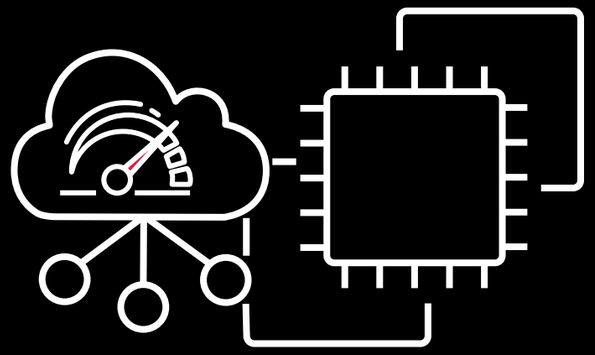

Working with Open-Source Too
https://aws.amazon.com
aws
English
Sign In to the Console
Amazon Virtual Private Cloud
Traffic Mirroring
Documentation - This Guide
Search
AWS Documentation » Amazon VPC » Traffic Mirroring » Working with Open-Source Tools for Traffic Mirroring
Working with Open-Source Tools for Traffic Mirroring
You can use open-source tools to monitor network traffic from Amazon EC2 instances. The following tools work with Traffic Mirroring:
Zeek — For more information, see the Zeek Network Monitor Security website.
Suricata — For more information see the Suricata website.
These open-source tools support VXLAN decapsulation, and they can be used at scale to monitor VPC traffic. For information about how Zeek handles VXLAN support and to download the code, see Zeek vxlan on the GitHub website. For information about how Suricata handles VXLAN support and to download the code, see Suricata vxlan on the GitHub website.
The following example uses the Suricata open-source tool. You can follow similar steps for Zeek.

Working with Open-Source Tools for Traffic Mirroring

https://amzn.to/2J6F07r

aws

Any other considerations?



VPC Traffic Mirroring

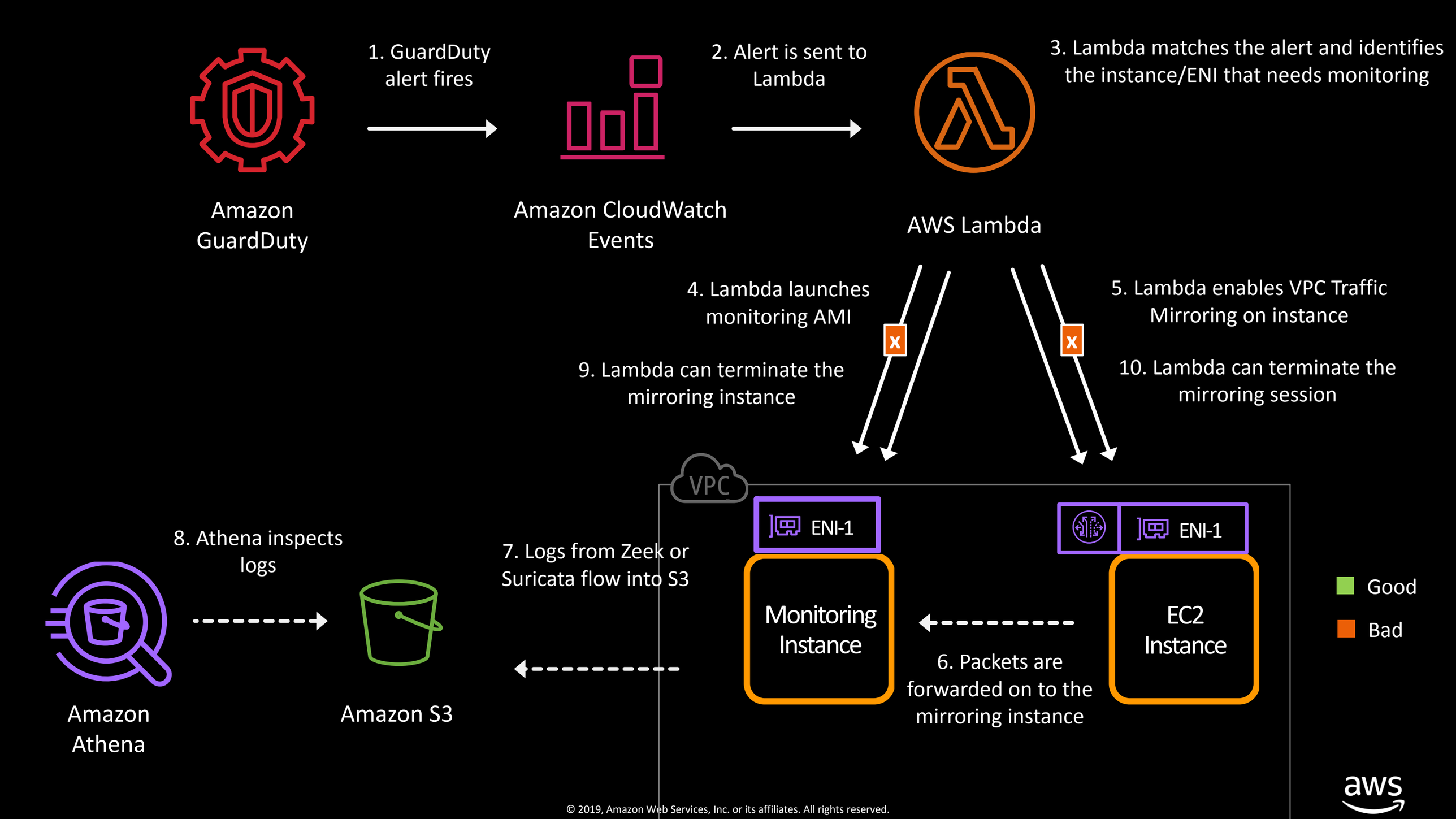
Other Considerations

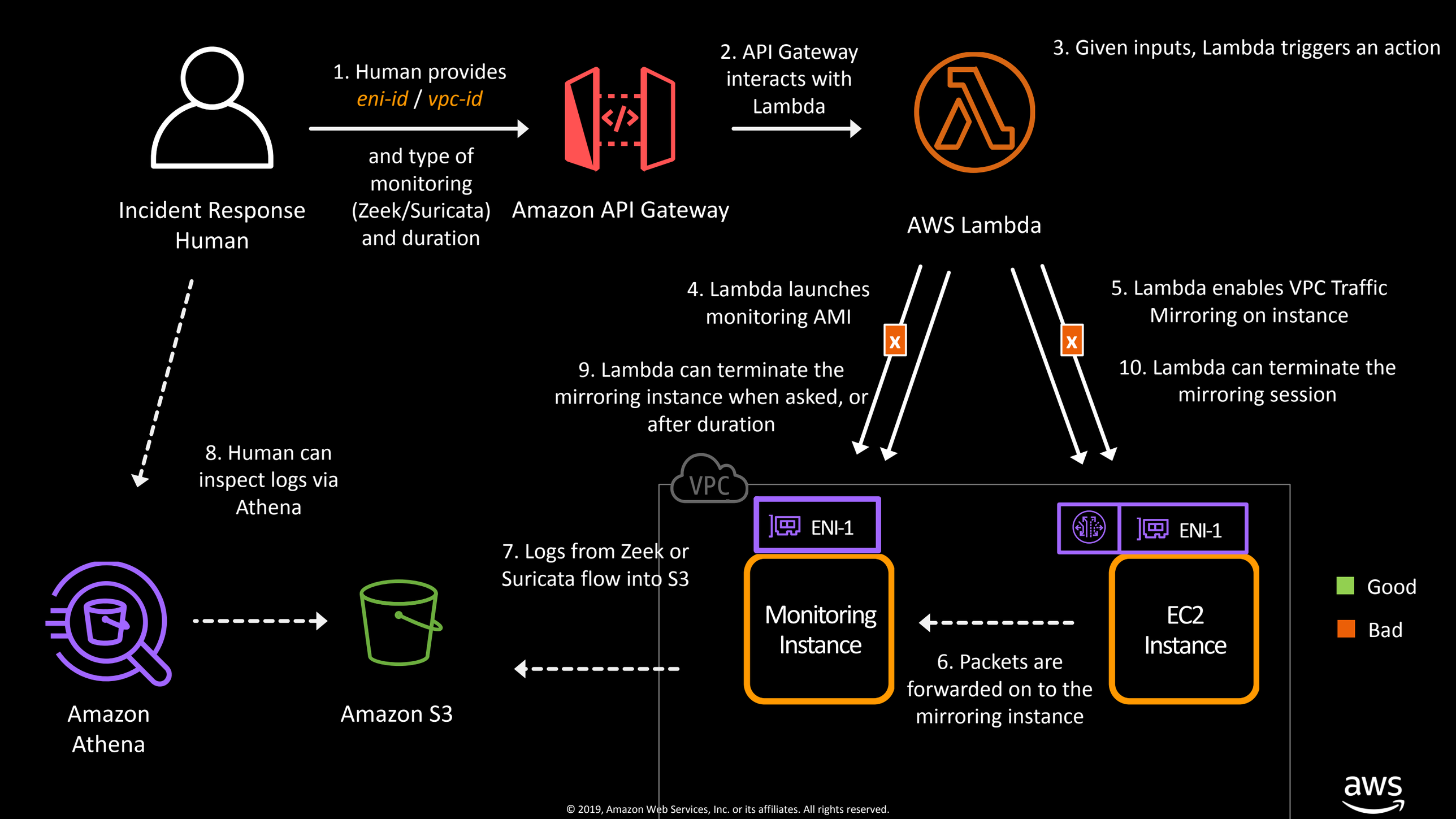
1. Mirrored traffic is not subject to the egress security group
2. Flow Logs do not capture mirrored traffic
3. Packets will be mirrored only if they pass the inbound security group or ACL

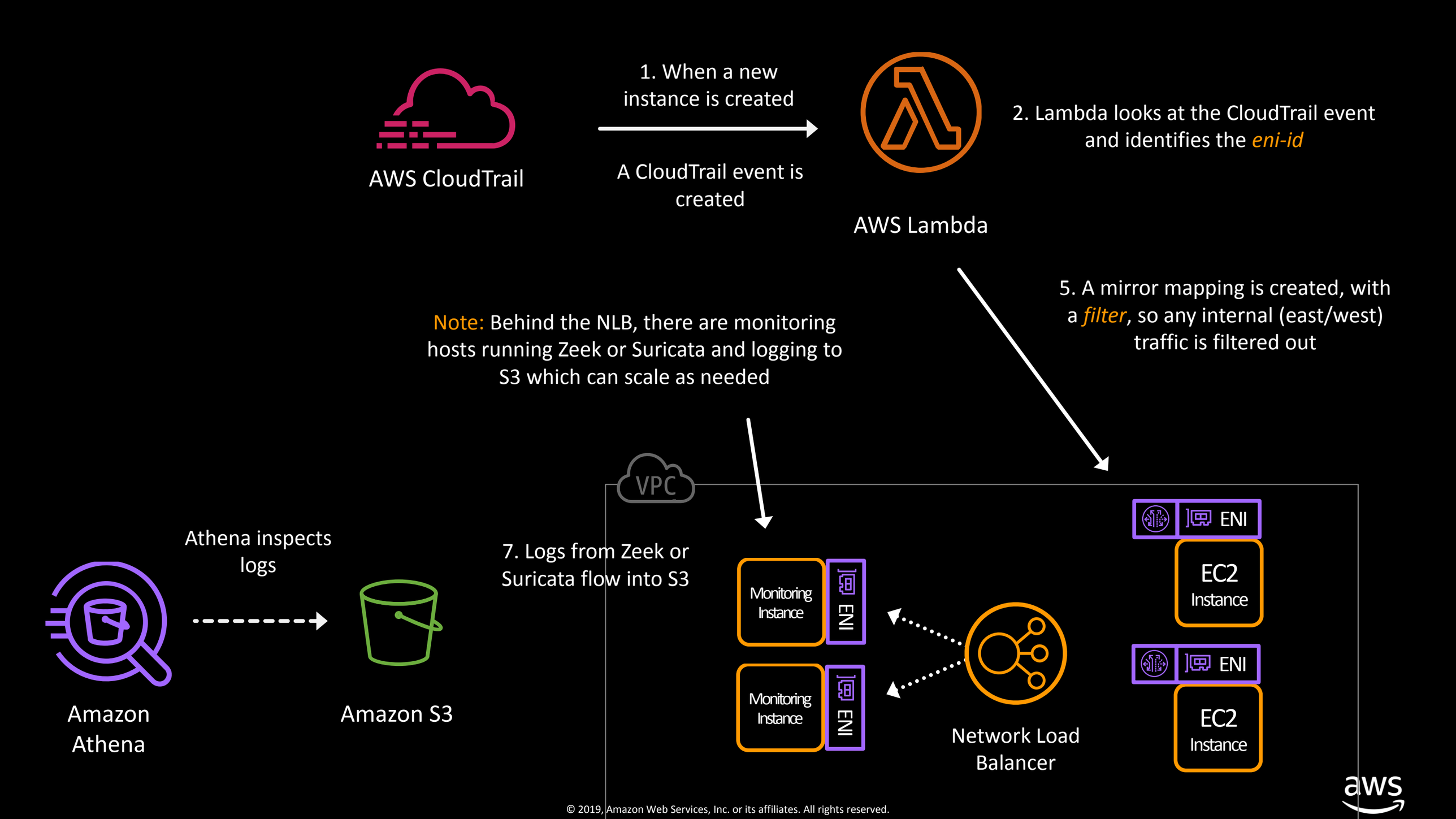
How Customers are usingTraffic Mirroring

New opportunities

- Continuous monitoring for sensitive workloads
- Deep and quick visibility for acquisitions
- On-demand access for incident responders







Thank you!

Anoop Dawani
Anoopda@amazon.com