Secure Cloud for AWS (IaaS)

Design Guide

June 2021

Contents

Abstract

This design guide aligns with the Cisco® Secure Cloud Architecture guide. The Secure Cloud Architecture guide explains the secure architecture for cloud applications, critical business flows; attack surfaces and corresponding security controls required for the cloud environment. This guide proposes a Cisco Validated Design (CVD) for security in a tiered application architecture. The solution proposed in this guide leverages Cisco security controls along with Cloud-Native security controls to achieve the desired security posture for applications in AWS.

Scope

This document illustrates the design and security aspects of an application hosted in AWS. Along with the design and security specifications, this document also delves into the details of implementation and validation steps for the proposed architecture.

This guide covers the following security controls.

- Cisco Tetration
- Cisco Advanced Malware Protection for Endpoints (AMP4E)
- Cisco Stealthwatch Cloud (SWC)
- Cisco Umbrella
- Cisco Firepower Next-Generation Firewalls Virtual (NGFWv)
- Cisco Adaptive Virtual Security Appliance (ASAv)
- Cisco Defense Orchestrator (CDO)
- AWS Web Application Firewall (WAF) and Shield service
- Radware Cloud Web Application Firewall (WAF) and DDOS prevention
- Cisco Duo Beyond
- Cisco SecureX threat response

For setting up the web application, we used the following AWS cloud components and services.

- AWS Virtual Private Networks (VPC)
- AWS Route Tables
- AWS Internet Gateway
- AWS Relational Database Service (RDS) Service
- AWS Auto Scale
- AWS Elastic Cloud Compute (EC2) Service
- Network Load Balancer
- Amazon Simple Storage Service (S3)
- Amazon Machine Image (AMI)
- AWS Route53

- AWS API Gateway
- AWS Lambda Service
- AWS Identity and Access Management (IAM)

SAFE Architecture - Introduction

As your data flows from an increasing number of devices to your data center or private/public cloud, you must understand your data flow, to be able to protect it. Cisco SAFE is an architectural approach that helps you visualize this transit of the data in terms of business flows, understand the attack surface associated with these flows and hence, devise appropriate capabilities to secure them. This framework provides complete guidance from the initial identification of business flows in a given architecture to securing it and then deploying and validating the solution.

These validated designs provide guidance that is complete with configuration steps that ensure secure deployments for your organization. Cisco Validated Designs (CVDs) for various SAFE PINs can be found at SAFE home page.

Cisco SAFE simplifies network security by providing solution guidance using the concept of 'Places in the Network' (PINs). This design guide is a recommended threat defense architecture for the Cloud PIN (see figure 1). Within the Cloud PIN, this design guide specifically covers the AWS cloud.

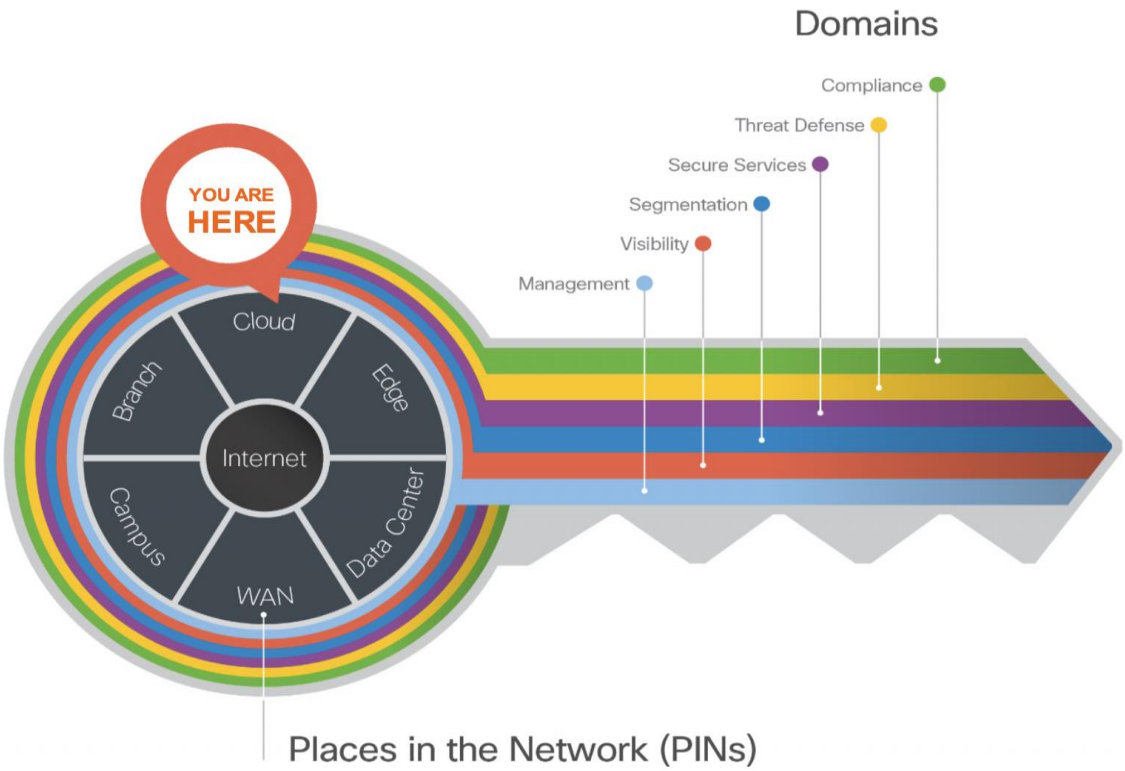


Figure 1.
Key to SAFE framework

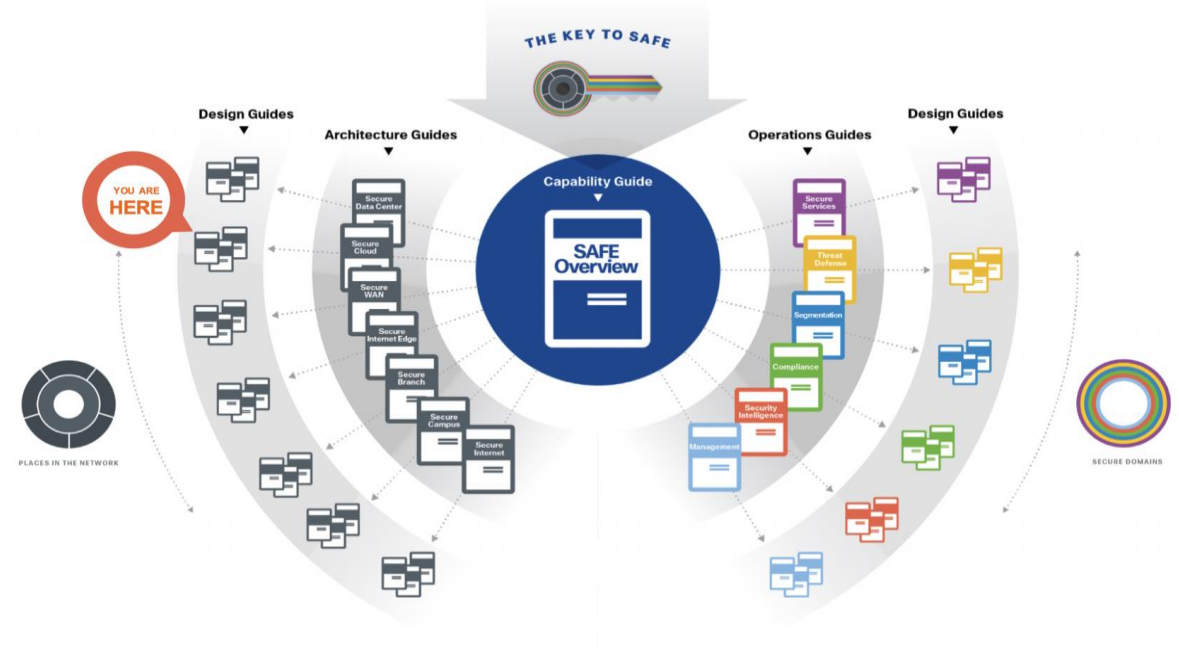


Figure 2.
SAFE Guidance Hierarchy

For more information on SAFE framework and architecture/design guides, check out the SAFE documentation (select architecture/design tab).

Cloud Business Flows

SAFE uses the concept of business flows to simplify the identification of threats. This enables the selection of very specific capabilities necessary to secure them.

This solution addresses the following business flows for a typical tiered web application hosted in AWS:

Figure 3.
Cloud business flows

- Customer browsing an e-commerce web application. The customer, sitting somewhere out on the Internet, browses the e-commerce web application hosted in the AWS cloud
- Application workloads downloading updates/patches from update servers outside the cloud (Internet). Application workloads sitting in the cloud need to reach out to various update servers to fetch the updates and patches at regular intervals
- Systems communicating east/west within the AWS cloud. For example- the frontend web servers will make HTTP requests to backend application engine or the application workloads will make API calls among themselves
- Application workloads transacting data with the database server within the cloud
- DevOps remotely accessing the management zone for workload management/update/patching purposes

Public Cloud Attack Surface

The secure cloud design protects systems by applying security controls to the attack surface found in the public cloud. The attack surface in public cloud spans the business flows used by humans, devices, and the network.

Threats include; rogue identity, DDoS, web vulnerabilities, infections, and advanced persistent threats allowing hackers the ability to take control of your devices and networks.

Considering the business flows elaborated in the last section (Figure 3), a deep dive into the attack surface for each of those business flows is shown below.

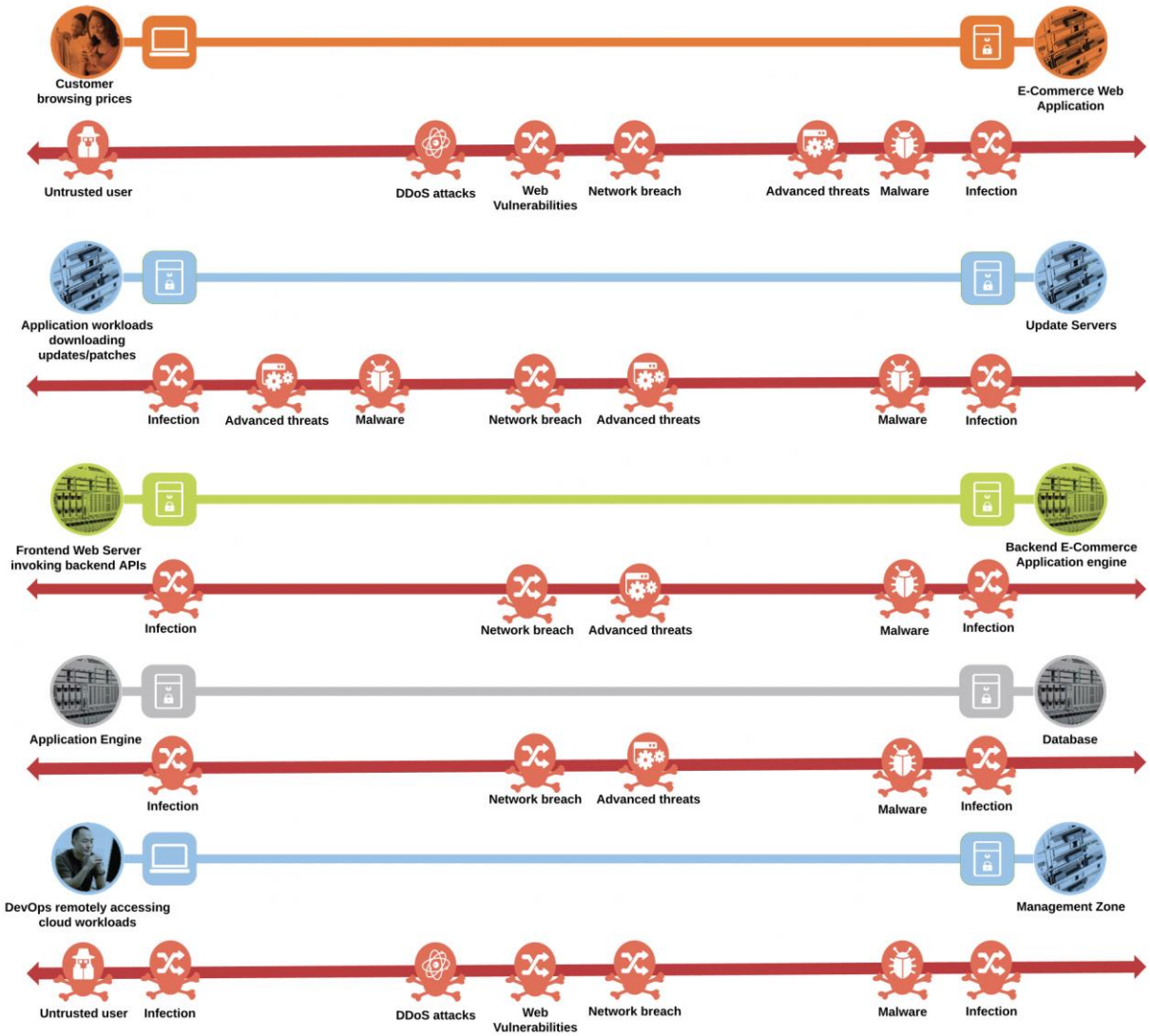


Figure 4.
Public cloud attack surface

- An untrusted/compromised user, out on the Internet, may try to exploit the cloud application or flood it with fake traffic to render it incapable of serving the genuine users

- The workloads need to communicate with update servers out on the untrusted public network. An attacker might compromise workloads to download malware to the application environment or upload crucial data to malicious servers
- Systems communicating east/west within the AWS cloud may spread the infection from one workload to another within the cloud, eventually compromising the whole application
- An attacker may compromise the application workloads to steal or corrupt data stored on the database servers
- A malicious user may try to gain the same privileged access as DevOps to compromise the complete application environment in AWS

Solution Overview

Cisco's security approach for the modern cloud applications allows companies to achieve:

- Improved resiliency to enable cloud availability and secure services
- Operational efficiency from automated provisioning and flexible, integrated security
- Advanced threat protection from Cisco TALOS – industry-leading threat intelligence to stay up to date, informed, and secure

What is our security approach?

Specific capabilities are necessary to protect the public cloud and build the appropriate layers of defense. These capabilities work together to create several layers of defense protecting the cloud applications. The top priorities or the three pillars that we keep in mind while designing the secure public cloud solutions are:

- **Visibility** - Complete visibility of users, devices, networks, applications, workloads, and processes
- **Segmentation** - Reduce the attack surface by preventing attackers from moving laterally, with consistent security policy enforcement, application access control and micro-segmentation
- **Threat Protection** - Stop the breach by deploying multi-layered threat sensors strategically in the public cloud to quickly detect, block, and dynamically respond to threats

Secure Cloud Business Flows

Developing a defense-in-depth architecture requires identifying existing threats and applying appropriate security capabilities to thwart them. Business flows and the corresponding attack surface and threat patterns that we defined earlier (Figures 3 and 4) are mapped to their corresponding security controls as below.

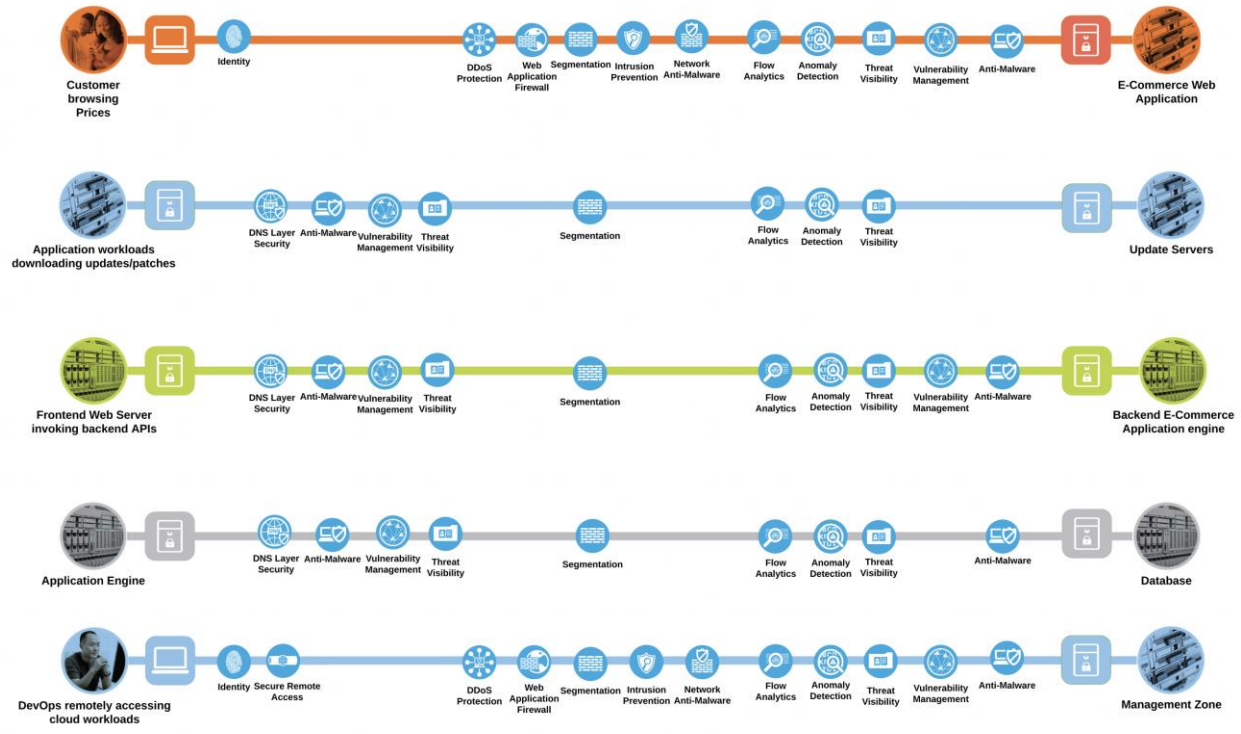


Figure 5.
Secure business flows

Cisco's Secure Architecture for AWS

The tiered application architecture has been a popular underlying principle for web application deployment for over a decade now and it remains equally relevant to date.

The multi-tier architecture provides a general framework to ensure decoupled and independently scalable application components. Each tier is separately developed, scaled, maintained and secured.

In the simplest tiered architecture form, the web applications would have the following layers:

Web tier: The end-user directly interacts with this layer. This tier has all the static web content.

Application tier: This tier is responsible for translating the user actions to application functionality. This tier carries the core application code components. For example, application code performing the read/write database operations.

Database tier: Storage tier or the database tier holds the data relevant to the application.

In this design, we are securing a tiered web application in the AWS cloud. We add various security capabilities and controls, that we established in the previous sections, to a tiered web application model to make it much more robust, secure and transparent in its security posture.

Secure Cloud Architecture

The Cisco Secure Cloud reference architecture solution includes all the security capabilities that we illustrated in previous sections.

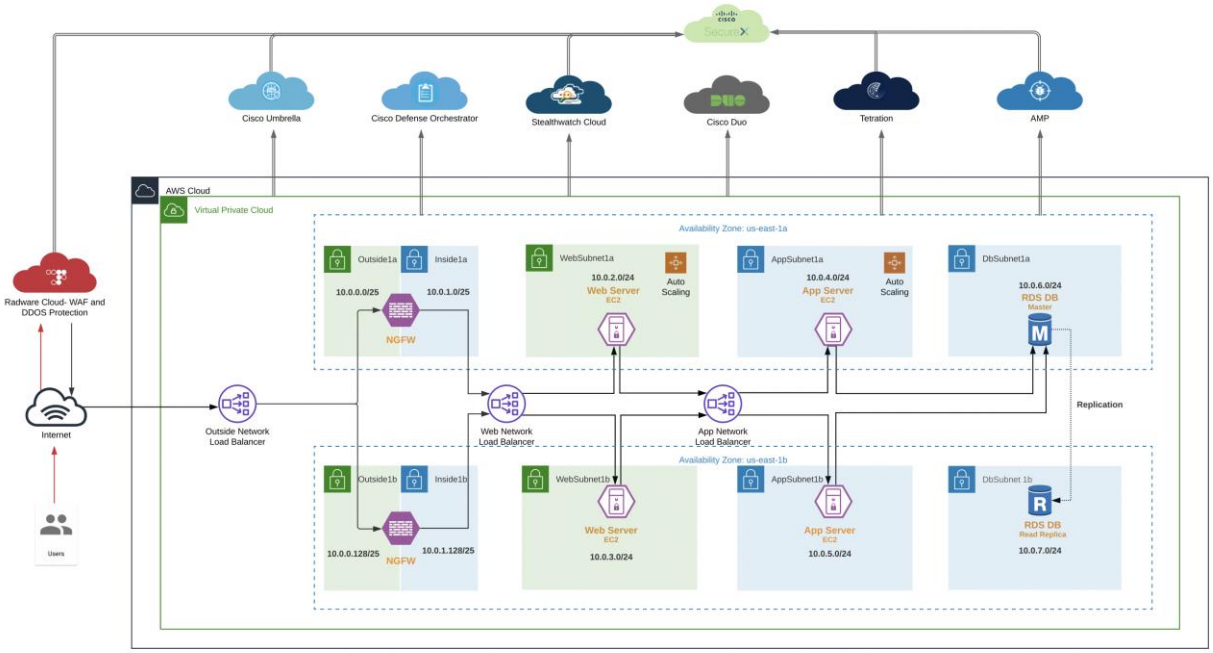


Figure 6.
Cisco Secure Cloud Reference Architecture

Business flows in Cisco's Reference Architecture

Considering the design above, all the threats, corresponding security capabilities and solutions required to attain those capabilities can be mapped as below.

Threat		Security Capability		Security Solutions
	Attackers or malicious users accessing restricted resources and information.		Identity based access	Cisco Duo – 2FA
	Massively scaled attacks that overwhelm services.		DDoS prevention	Radware DDoS prevention
	Attacks against poorly developed applications and web vulnerabilities.		Web Application Firewalls	Radware Web Application Firewall

Threat		Security Capability		Security Solutions
	Network breach causing unauthorized access and malformed packets between and within application in the cloud.		Segmentation	Cisco Firepower NGFWv Cisco Tetration
	Zero-day malware attacks and other forms of covert threats.		Threat visibility	Cisco Stealthwatch Cisco Tetration Cisco AMP4E Cisco Firepower NGFWv
	Attacks using worms, viruses, or other techniques.		Intrusion Prevention	Cisco Firepower NGFWv Cisco AMP4E
	Infections, attackers using a compromised workload to spread the damage.		Micro-segmentation	Cisco Tetration
	Malware distribution among workloads or between servers.		Anti-malware	Cisco Firepower NGFWv Cisco AMP4E
	Traffic, telemetry, and data exfiltration from successful attacks. Covert threats.		Flow Analytics	Cisco Stealthwatch Cisco Tetration Cisco Umbrella
	Exploiting privileged access to run shell code		Process Anomaly Detection & Forensics	Cisco Stealthwatch Cloud Cisco Tetration Cisco AMP4E
	Malware distribution across networks.		Network Anti-Malware	Cisco Firepower NGFWv
	Exploiting unpatched or outdated applications.		Vulnerability Assessment and Workload Inventory	Cisco Tetration
	Redirection of session to malicious domains.		DNS Layer Security	Cisco Umbrella – DNS Layer Security

Threat		Security Capability		Security Solutions
	Exposed services and data theft.		VPN Gateway or Concentrator	Cisco ASAv Cisco Firepower NGFWv

At this point, we have established the attack surface and, the capabilities and security solutions that we needed to secure the business flows mentioned previously.

- Customer browsing an e-commerce web application
  - Access to the web application is secured using Duo – Multi-Factor Authentication (MFA)
  - WAF and DDoS Services protect against web vulnerabilities and denial of service attacks. In this document, we will demonstrate Radware cloud WAF and DDoS service
  - Perimeter segmentation is done using next-generation firewalls (NGFW) to protect against any network level breaches. NGFWs also provide next-generation IPS and AMP capabilities along with stateful firewall, AVC (Application Visibility and Control) and URL filtering
  - Micro-segmentation of workloads is done using the Tetration policy enforcement agents. This would prevent any malware or malicious movement within the pool of workloads in a specific tier
  - Stealthwatch Cloud provides enhanced threat visibility into workload activity and the AWS cloud. It looks for any anomalous activity within the application environment. It also facilitates the flow analytics
  - Tetration agents allow us to gain a deep visibility into vulnerable packages and processes on the workloads that an attacker may leverage. It also provides a very robust network flow analytics for workload communications
  - AMP4E detects and quarantines any malware that may infect the workloads
- Workloads downloading updates/patches from update servers
  - Workloads are segmented into App and Web tier using Tetration Enforcement agents. No direct inbound public access is allowed to the App and Web servers, management access is allowed only from the management tier (also controlled via Tetration)
  - DNS layer security is achieved using Cisco Umbrella. This prevents any accidental or deliberate exposure to a malicious domain
  - Stealthwatch Cloud and Tetration provide enhanced threat visibility and flow analytics
  - AMP4E detects and quarantines any malware that may get downloaded to application workloads
- Systems communicating east/west within the AWS cloud
  - Workloads are micro-segmented using Tetration Enforcement agents. Web, App, Database and Inside tier has no direct inbound public access/addresses. Only Management and the Outside tier is allowed Public IP addressing, hence exposing them to untrusted public network/internet
  - Micro-segmentation within Web and App tier is done using the Tetration enforcement agents. This restricts any internal movement among the workloads
  - DNS layer security using Umbrella provides visibility into workload activity
  - Stealthwatch Cloud and Tetration provide enhanced threat visibility and flow analytics for this flow. They also look for any anomalous movement within the application environment or among the workloads within a tier. Tetration agents provide deep visibility into the workloads
  - AMP4E protects against malware spread

- Application engine transacting data with database server within the cloud
  - AWS Security Groups restrict access to the database. Only App tier is allowed to communicate with database tier
  - DNS layer security using Umbrella
  - Stealthwatch Cloud and Tetration provide enhanced threat visibility and flow analytics. They also look for any anomalous movement within the application environment or among the workloads within a tier. Tetration agents provide deep visibility into the workloads
  - AMP4E protects the application workloads against any malware infection
- DevOps remotely accessing the management zone for workload management/update/patching purposes
  - Anyconnect VPN mobility client is used to provide Secure Remote Access to the management tier. An ASA or NGFWv can be used for VPN termination. We tested a standalone ASAv for this design. Refer to the Secure Remote Worker design guide for detailed information on secure remote access designs and deployments
  - Management zone is segmented using Tetration enforcement agents. This provides the control knob for restricting access to workloads or the various other tiers
  - Stealthwatch Cloud and Tetration provide enhanced threat visibility and flow analytics. They also look for any anomalous movement or activity within the application environment or from the management tier. Tetration agents provide deep visibility into the workloads
  - AMP4E protects the jump servers and workloads against any malware infection

Security Integrations

Let's look at each of the security integrations in this secure design in more depth, we will start from the security controls on the workload itself and go all the way to the edge of our public cloud web application.

We start by looking at workload security using Tetration and Advanced Malware Protection, followed by an agentless deployment of Stealthwatch cloud for greater visibility into the AWS environment and workload activity. Then, we will look into Umbrella DNS layer security at the AWS VPC level.

Afterwards, we move to perimeter protection using Cisco Firepower NGFWv (policy orchestrated by Cisco Defense Orchestrator). We will also explore WAF and DDoS protection using Radware Cloud service.

Lastly, we will secure the access to our cloud application using Duo Multi-Factor Authentication.

To connect all these security controls to a single pane of glass, we will look at Cisco SecureX integrations.

Cisco Tetration

Tetration has a SaaS offering that provides the capability to do micro-segmentation in a highly flexible manner along with an in-depth visibility into the workloads.

Tetration offers visibility and enforcement agents that are installed on the workloads. The enforcement agents provide an additional capability to enforce policies.

Tetration can dynamically learn various ongoing changes in the cloud workload environment and enforce an adaptive micro-segmentation. The Tetration portal allows us to create workspaces and graphical views for applications and enforce security from the web application point of view unlike the traditional network perspective.

The Tetration platform supports multi-cloud and hybrid environments and hence, make the whole process of security operations seamless across the board.

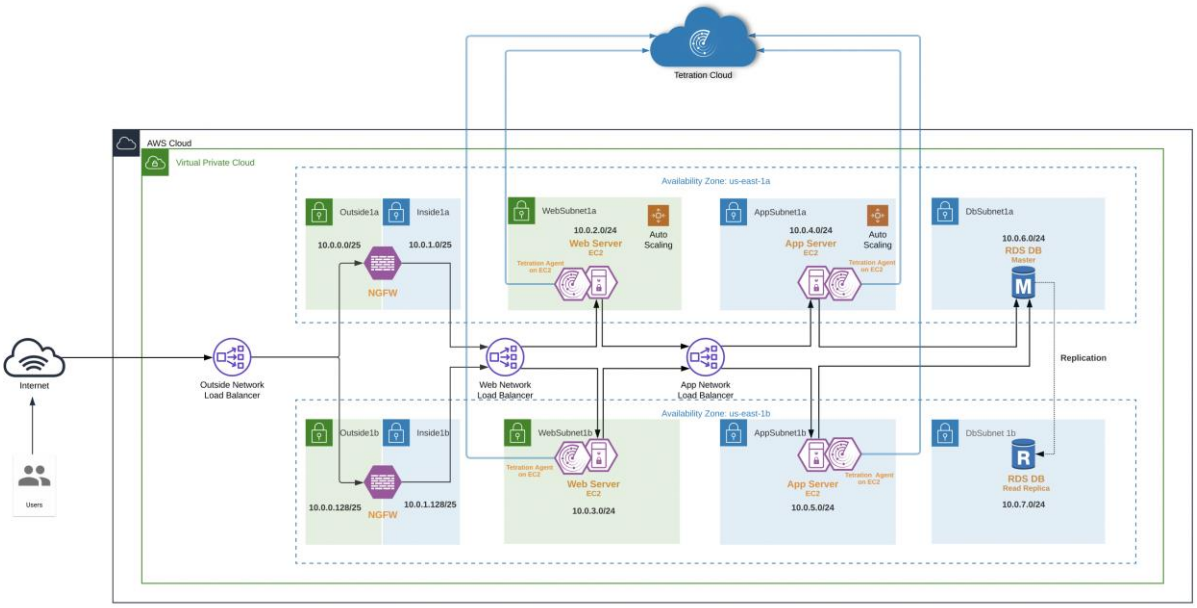


Figure 7.
Cisco Tetration

In this specific architecture, Web and Application tier has workloads in Auto Scaling Groups. To enable the auto-provisioning of Tetration agents, we used the User Data option provided for EC2 Instances. When the Auto Scaling Group deploys a new workload, the shell script will install the Tetration agent on it as part of the initialization process. Refer to the implementation section of this guide for more details.

Once the Tetration agent on the new workload is registered with the Tetration cloud (SaaS), it starts exporting the network flow and process information to the Tetration cloud engine for analysis. Tetration ensures Cisco's Zero Trust model by offering key features like:

- Policy enforcement (Micro-segmentation)
- Visibility into workload process activity
- Network flow visibility
- Software vulnerability reports

- Forensic analysis
- Behavior deviations

Based on all these features and more, the Tetration dashboard provides us with a very convenient and flexible scoring mechanism to monitor the security compliance of cloud applications. Tetration considers six parameters to calculate this score (Figure 8), and these parameters can be adjusted based on one's preference or requirements.

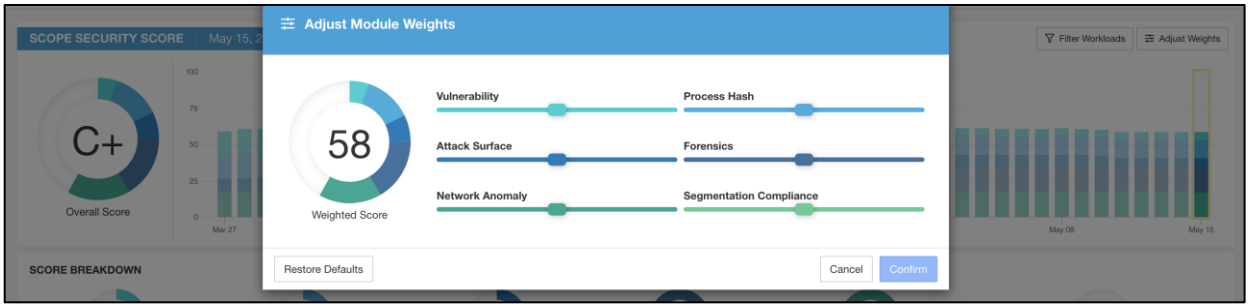


Figure 8.
Tetration Dashboard - Weighted Score

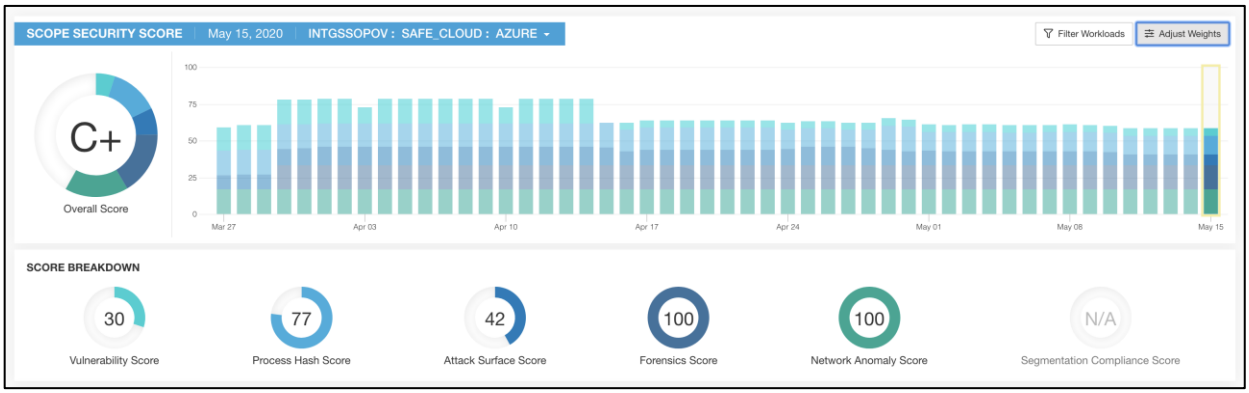


Figure 9.
Tetration Dashboard - Compliance Score Board

Refer to the Tetration documentation for more detailed information on cloud workload protection.

Cisco Advanced Malware Protection for Endpoints

The AMP4E agents installed on the cloud workloads provide us protection against zero-day attacks. Powered by Cisco TALOS, AMP4E not only relies on antivirus, but also uses machine learning and file reputation to block both file-based and file-less attacks. It also enables you to isolate the infected host before the malware is spread onto the others in the network. Advanced Malware Protection also supports taking forensic snapshots that help immensely with the security investigations.

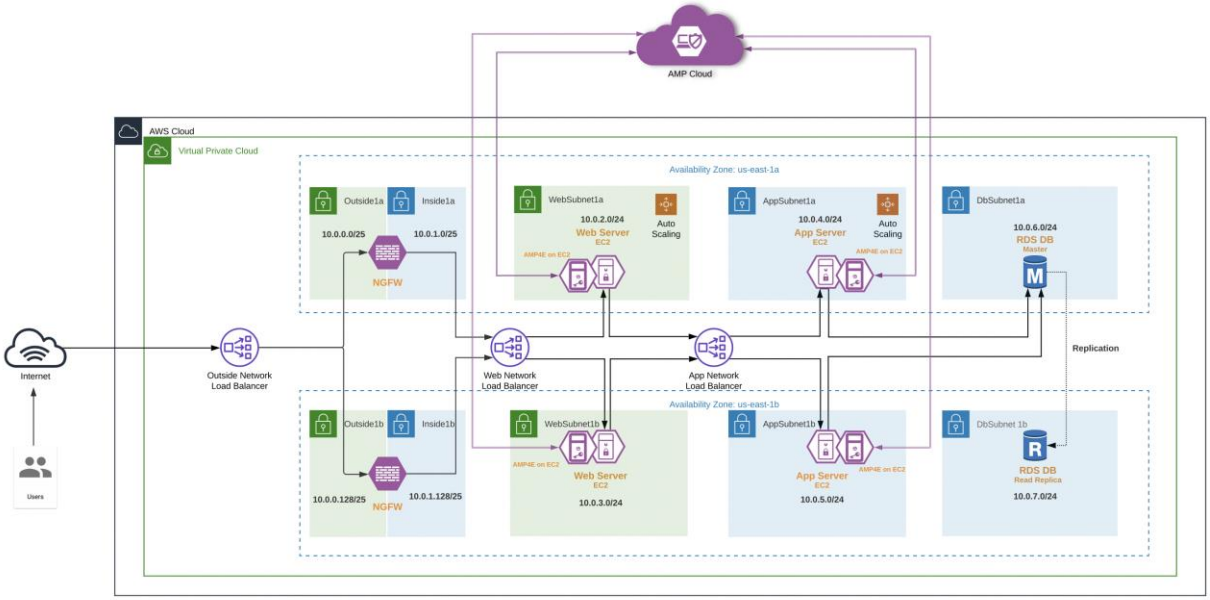


Figure 10.
Cisco Advanced Malware Protection for Endpoints

In this specific architecture, just like the Tetration agent, the web and application workloads in Auto Scaling Group are auto-provisioned with AMP4E agents using User Data option available under Auto Scaling Group configuration. When the Auto Scaling Group deploys a new workload, a shell script will install the AMP4E agent on the workload as part of the initialization process.

As soon as AMP4E agent on the new workload registers with the AMP cloud, the workload is continuously monitored and reported for any malicious activity. AMP's host isolation feature comes in very handy to contain any spread of malware in the cloud workloads.

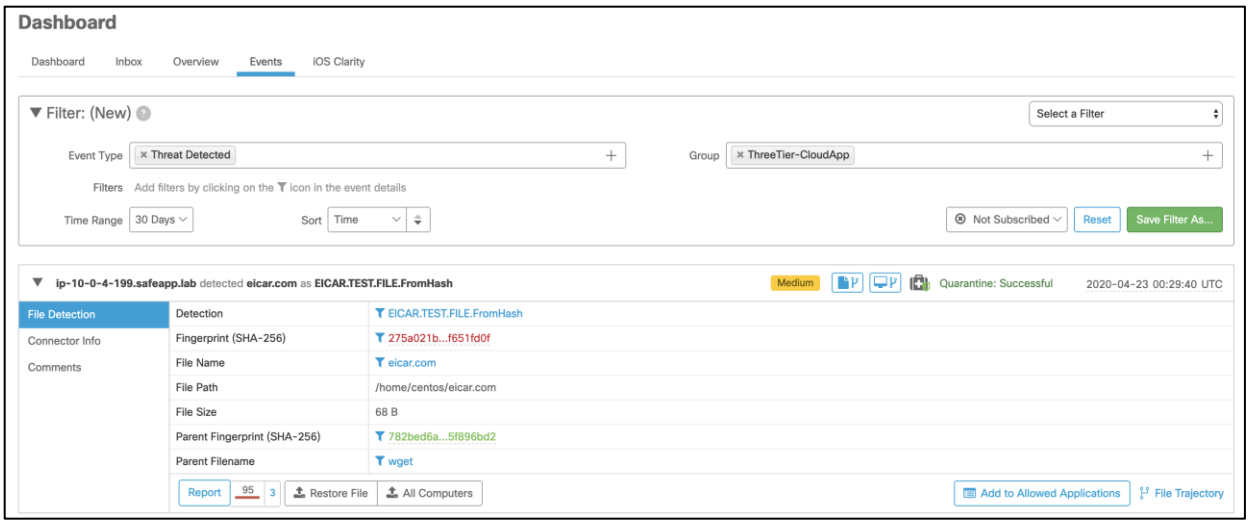


Figure 11.
AMP Dashboard – Threat monitoring

Cisco Stealthwatch Cloud

Stealthwatch Cloud (SWC) helps overcome the visibility challenge, especially in public cloud environments. It provides an agentless deployment in the AWS cloud.

Stealthwatch Cloud pulls the VPC flow logs from the designated S3 bucket. It learns the AWS environment and baselines the resources. VPC flow logs have the flow information associated with various AWS resources, even for those that are not strictly tied to a static IP address. SWC is capable of correlating the IPs and then tying them back to their origin AWS service. In other words, SWC performs dynamic entity modeling and organizes all the AWS resources based on the functions that they're performing. For example, the entity could be categorized as a firewall, an application server or a load balancer and so on. This type of resource profiling and modeling is extremely important to look for any suspicious activity within the cloud application environments.

In addition to VPC flow logs, Stealthwatch Cloud also consumes other telemetry sources like AWS IAM, CloudTrail, EC2, ElasticCache, Inspector, GuardDuty, RDS, S3, Auto Scale, Elastic Load Balancing service for additional context and alerting.

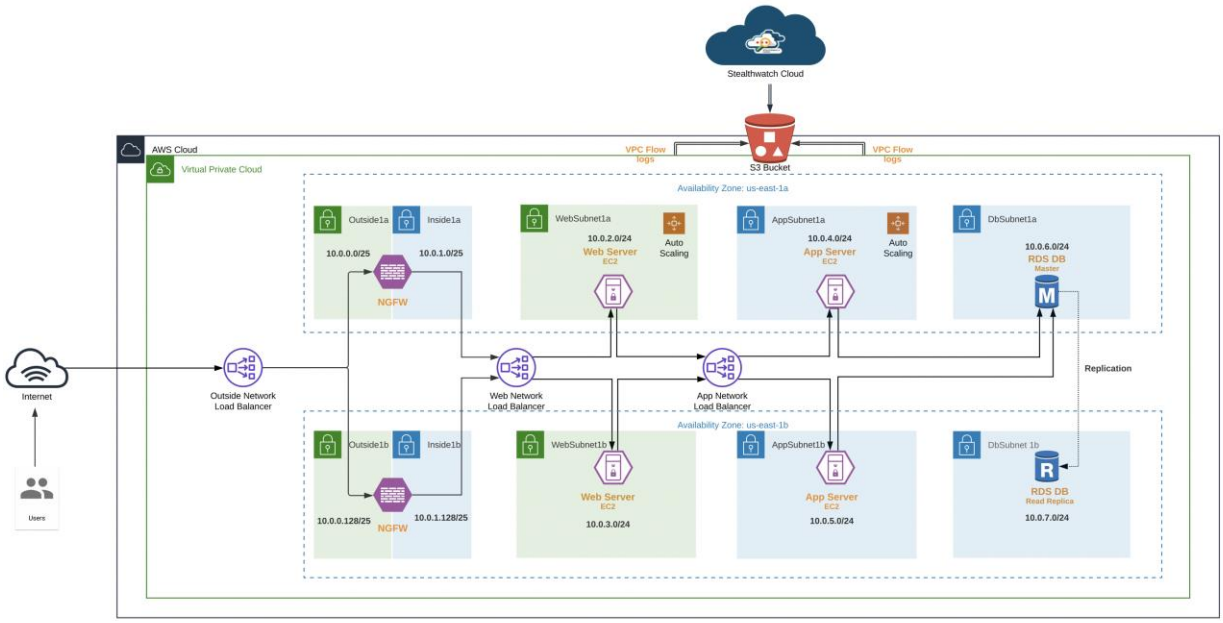


Figure 12.
Cisco Stealthwatch cloud

Once the Stealthwatch Cloud finishes identifying the entities, it baselines their behavior over a fixed period of time. As soon as the baselining is completed, any unexpected behavioral change of the entities and the way different cloud services communicate with each other is alerted on. This helps to maintain deep visibility into the cloud environment and hence, track and prevent any unauthorized transfer of data or resource access.

Some of the common Stealthwatch alerts related to the AWS services include:

- AWS API Watchlist IP Hit - This alert is triggered when an AWS API is accessed by an IP on a user-supplied watchlist.
- AWS Config Rule Violation – This alert uses the AWS Config Compliance observation and indicates that the resource is not compliant with configured AWS Config rules
- AWS Login Failures – This alert is triggered when a user tries and fails to log in to the AWS Console several times
- AWS Inspector Findings - Triggers when AWS Inspector reports a high severity event. This alert indicates that the resource is not complying with AWS best practices
- AWS Lambda Invocation Spike- Triggers when AWS Lambda function is invoked a record number of times. This alert may indicate a DoS attack.

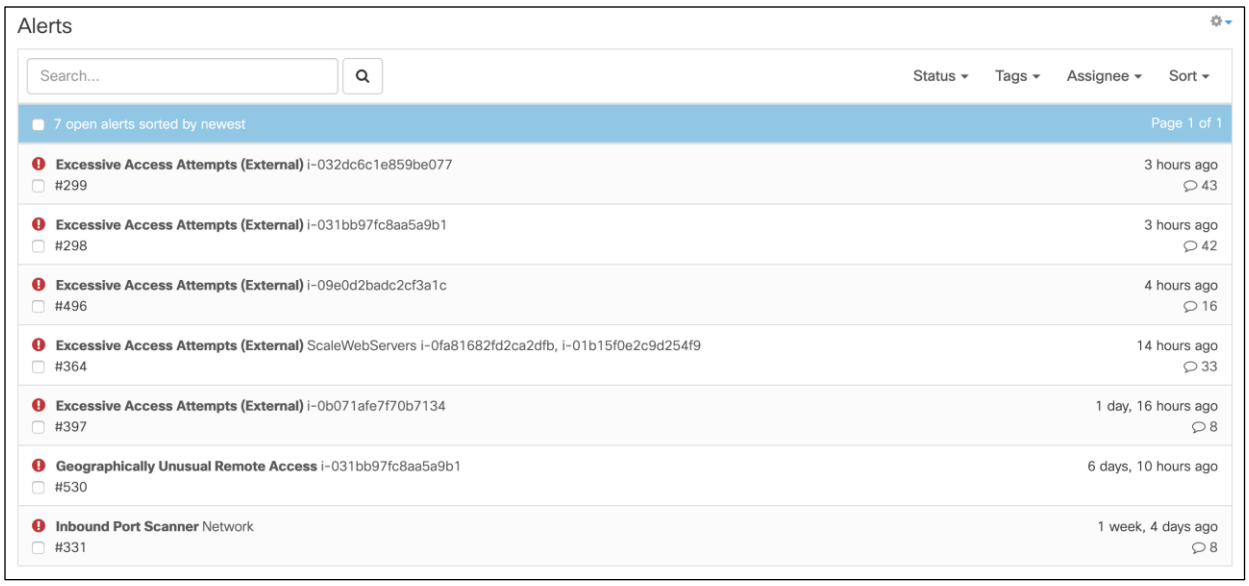


Figure 13.
Stealthwatch Cloud - Alerts

Cisco Umbrella

Cisco Umbrella offers flexible cloud-delivered security. It combines multiple security functions into one solution. Cisco Umbrella solutions provide DNS-layer security, secure web gateway, cloud-delivered firewall, cloud access security broker (CASB), and interactive threat intel. This document covers Umbrella DNS-layer protection for the workloads in the AWS Virtual Private Cloud (VPC).

The Umbrella DNS policies allow you to dictate block policy for a variety of pre-defined web categories. More details on web categories can be found in Umbrella documentation. It also gives you the flexibility to apply the policies to specific identities. For example, you could have one set of rules for your AWS cloud application and another set for a different site.

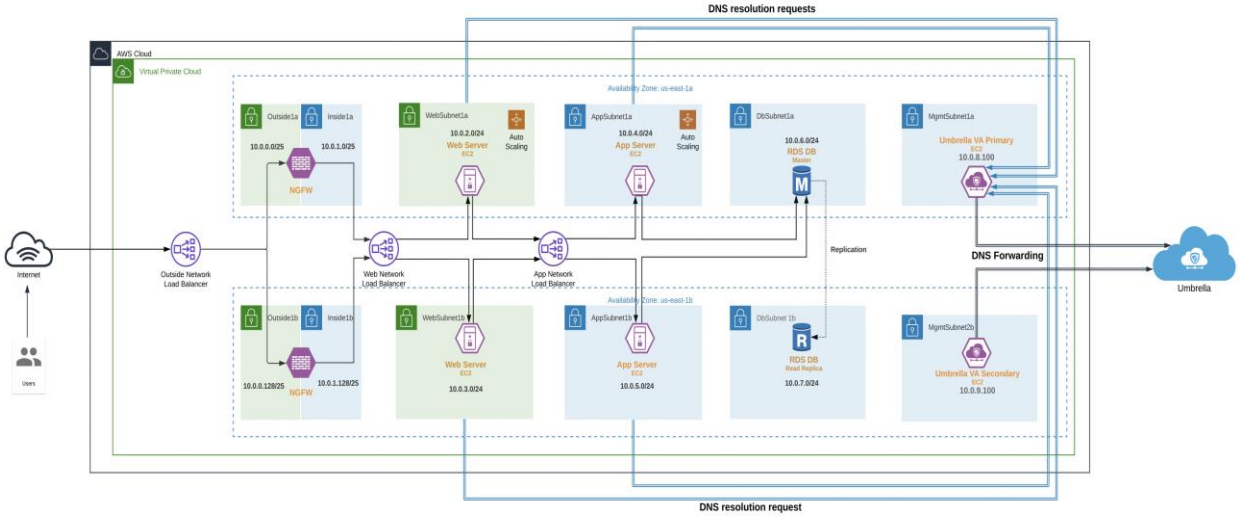


Figure 14.
Cisco Umbrella - DNS layer Security

We deploy Umbrella Virtual Appliances (VA) in the Management tier of the AWS VPC. These VAs act as DNS forwarders to Umbrella. The AWS VPC offers the option to configure custom DNS settings (DHCP Options Set), allowing us to point the cloud resources in a given VPC to Umbrella VAs instead of AWS local DNS. Every resource, that is launched into the VPC, will use these Umbrella DNS forwarders, to provide a control knob for the DNS layer security.

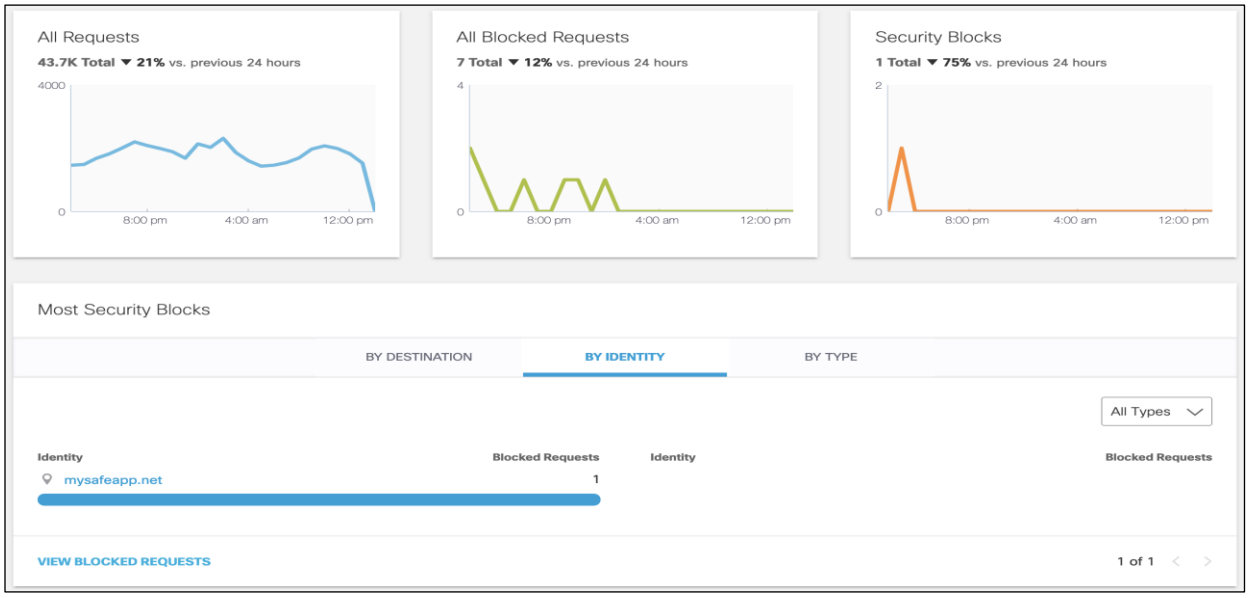


Figure 15.
Umbrella - DNS Traffic Monitoring

Next-Generation Firewall Virtual

Cisco® Firepower Next-Generation Firewall Virtual (NGFWv) appliance combines Cisco's network firewall with advanced next-gen IPS, URL filtering, AVC and malware detection (AMP) capabilities. In this design, we use NGFWv to secure the network perimeter from all sorts of threats from public Internet. This ensures that we have

security controls like filtering, intrusion prevention and malware detection right at the gateway to the cloud application.

To provide secure remote access to the workloads and database instance, we use Cisco ASAv as a VPN headend. Cisco ASAv offers secure remote access capabilities using Anyconnect VPN mobility client. You could also use NGFWv for this purpose. For detailed information on secure remote access deployments, refer to the Secure Remote Worker SAFE design guide.

Cisco Defense Orchestrator (CDO) is used for management and policy orchestration. CDO provides one security policy, faster deployment, and smart configuration management. It eliminates the time-consuming complexity of managing policies across multiple FTDs and ASAs.

If you choose to use AWS Security Groups alone for segmentation (without any next-generation firewalls), the reference design is shared in Appendix A of this document. Cisco also provides CloudFormation templates and scripts for deploying an auto-scaling group of FTDv appliances. This guide does not cover the auto scaling solution, please refer to the Cisco documentation for more details.

Note: The terms Next-Generation Firewall (NGFW) and Firepower Threat Defense (FTD) are used interchangeably throughout this guide. Both these terms refer to Cisco Firepower Next-Generation Firewalls in the context of this document. AWS marketplace offering is available under the name 'Cisco Firepower NGFW Virtual (NGFWv)'.

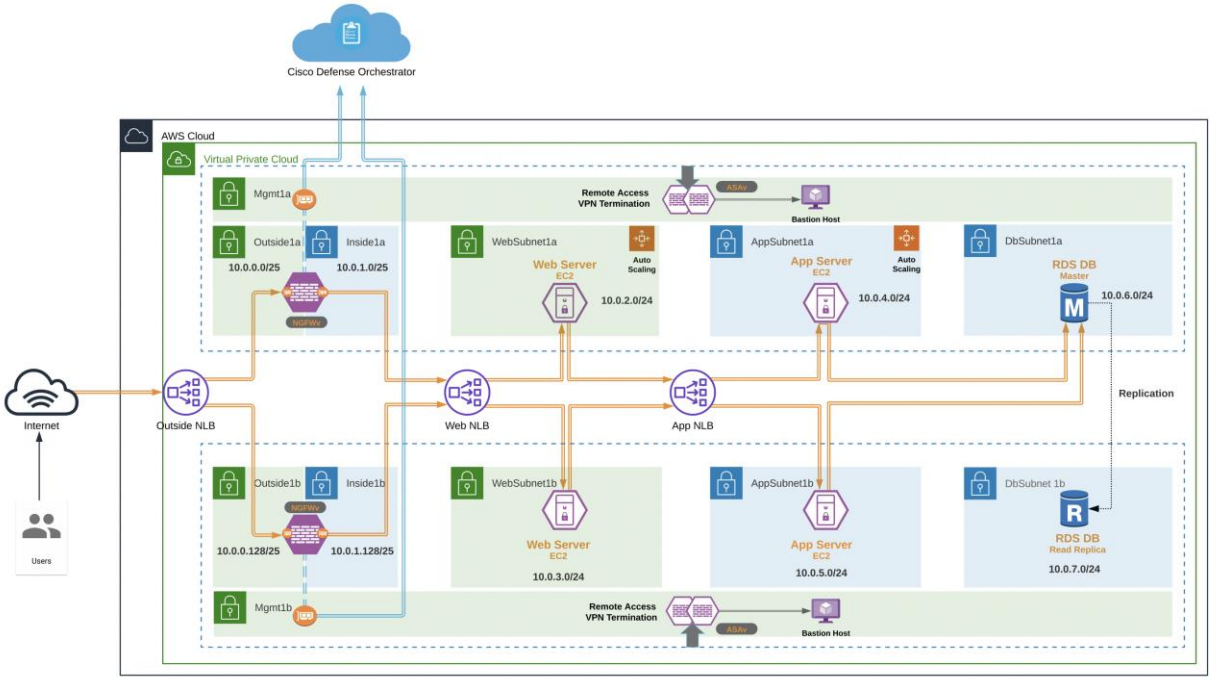


Figure 16.
NGFWv - Traffic flow from Internet User to application

User to application traffic flow

When the user out on the Internet tries to browse the cloud-hosted web application, it lands on Outside Network Load Balancer after being scanned by WAF and DDoS protection system for any malicious activity. The destination IP at this point is the public IP of the Outside load balancer. Outside load balancer sits in the Outside tier (segmented using AWS Security Group) and load balances traffic onto the pool of outside interfaces of

Firepower Threat Defense (FTD) appliances. The FTD appliance receives the request and then forwards the traffic to 'Web' Network Load Balancer, to be load-balanced on to the Auto-Scaled group of web servers.

Before the traffic leaves the inside interface of the FTDv (in the Inside tier), the source of the web request is translated (Network Address Translation) to FTD's inside interface IP and the destination is changed to 'Web' load balancer IP. The source IP is translated here to ensure traffic symmetry.

Web server receiving this incoming request, after being load-balanced by the Web load balancer, fetches the content from app workloads and returns the final response directly to the firewall which forwarded the initial request. At this point, firewall routes this response back to the end-user via the outside interface.

Web Application Firewall and DDoS Prevention

Public cloud has become a common place to host critical applications and make these applications available to end-users (internal or external). As a result, it is essential to ensure these applications receive the same level of protection from distributed denial of service (DDoS) and advanced web attacks that on-premises applications do.

Radware Cloud WAF service protects web applications from common web exploits. Radware's Cloud Security Services offer easy-to-deploy cloud-based security that can be integrated with any cloud environments to provide proactive, automated protection from advanced threats. The Cloud WAF service provides full coverage against OWASP top 10 attacks along with protection against 0-day web attacks. In addition to web traffic protection, DDoS component provide network flow monitoring to protect against the full breadth of DDoS attacks with real-time mitigation and no added latency in peacetime.

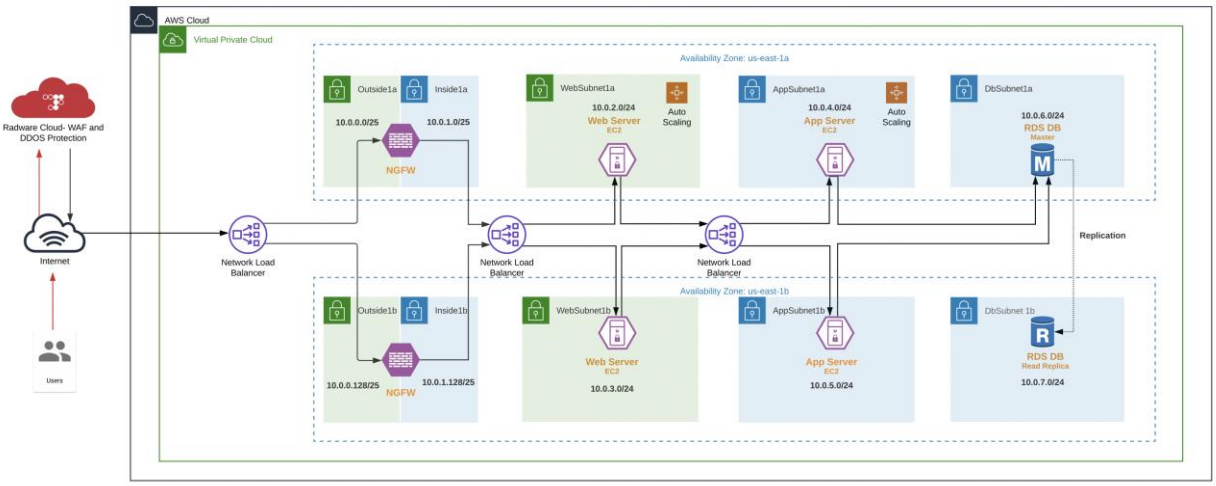


Figure 17.
Radware Cloud - WAF and DDoS Prevention

Deployment is hassle free, application's domain name points to the Radware Cloud service. Traffic is first routed to the Radware Cloud and scanned for any malicious activity. Post-inspection, the traffic is forwarded to the origin servers in the AWS cloud. Refer to the implementation section of this guide for more deployment level details.

Cisco Duo

Cisco Duo provides secure access to applications and data, no matter where the users are, on any device, and from anywhere. Cisco Duo's secure access solution creates trust in users, devices, and the applications they access. Cisco Duo provides the following functions:

- Multi-Factor Authentication: Verify the identity of all users with Duo's strong multi-factor authentication
- Single Sign-on: Seamless, single dashboard access to all applications
- Remote Access: Secure access to cloud and on-premises applications and servers, with or without VPN
- Device Trust: Check that user devices meet security standards before granting them access
- Adaptive Access Policies: Set policies to allow or block access attempts by a user or a device, based on contextual factors

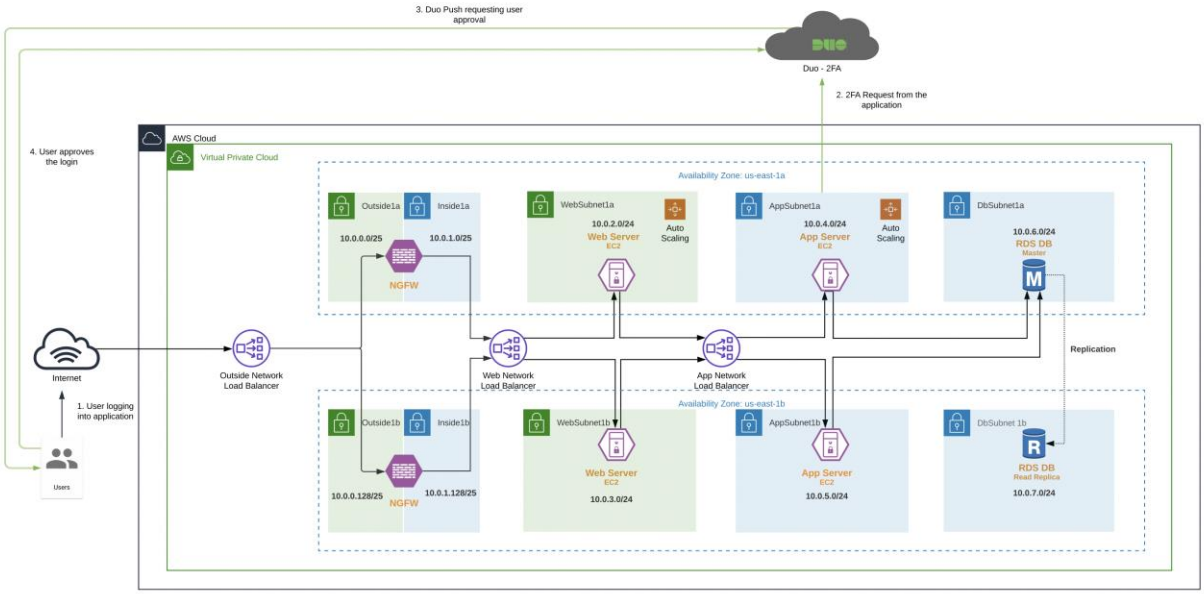


Figure 18.
Duo MFA Push

In this design, we used Duo's Multi-Factor Authentication (MFA) for our AWS cloud application. Multi-factor authentication from Duo protects the cloud applications by using a second source of validation, like a phone or token, to verify user identity before granting access. MFA not just allows you to build a zero-trust framework but is also essential for compliance purposes. Duo provides native integration for any application. Refer to the implementation section of this guide for more details.

Admins have several options when it comes to enrolling new users in Duo, such as self-enrollment, Active Directory sync, and OpenLDAP sync. Duo admin portal allows a highly convenient way to track any user activity.

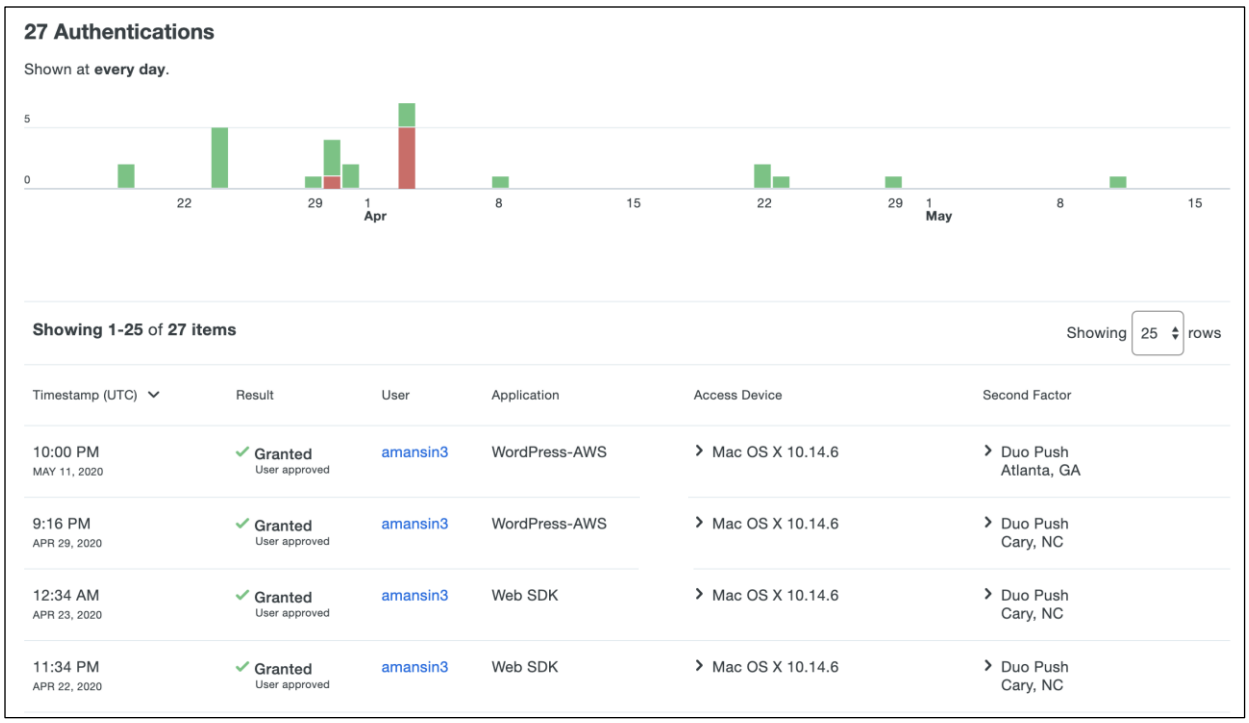


Figure 19.
Duo - User Activity

Cisco SecureX

Cisco SecureX leverages the integrated security architecture to accelerate investigations by automating and aggregating threat intelligence and data across your security infrastructure in one unified view. Some of the key features are:

- Aggregated threat intelligence: Integrates threat intelligence from Cisco TALOS and third-party sources to automatically research indicators of compromise (IOCs) and confirms threats quickly
- Automated enrichment: Automatically adds context from integrated Cisco Security products, so that you instantly know which of your systems was targeted and how
- Incident tracking: Provides the capability you need to collect and store key investigation information, and to manage and document your progress and findings
- Interactive visualizations - Shows your results on intuitive, configurable graphs for better situational awareness and quick conclusions
- Seamless drill down - Makes deeper investigations easy using integrated Cisco Security products. A single click takes you inside Cisco AMP for Endpoints
- Direct remediation - Lets you take corrective action directly from its interface. Block suspicious files, domains, and more without having to log in to another product

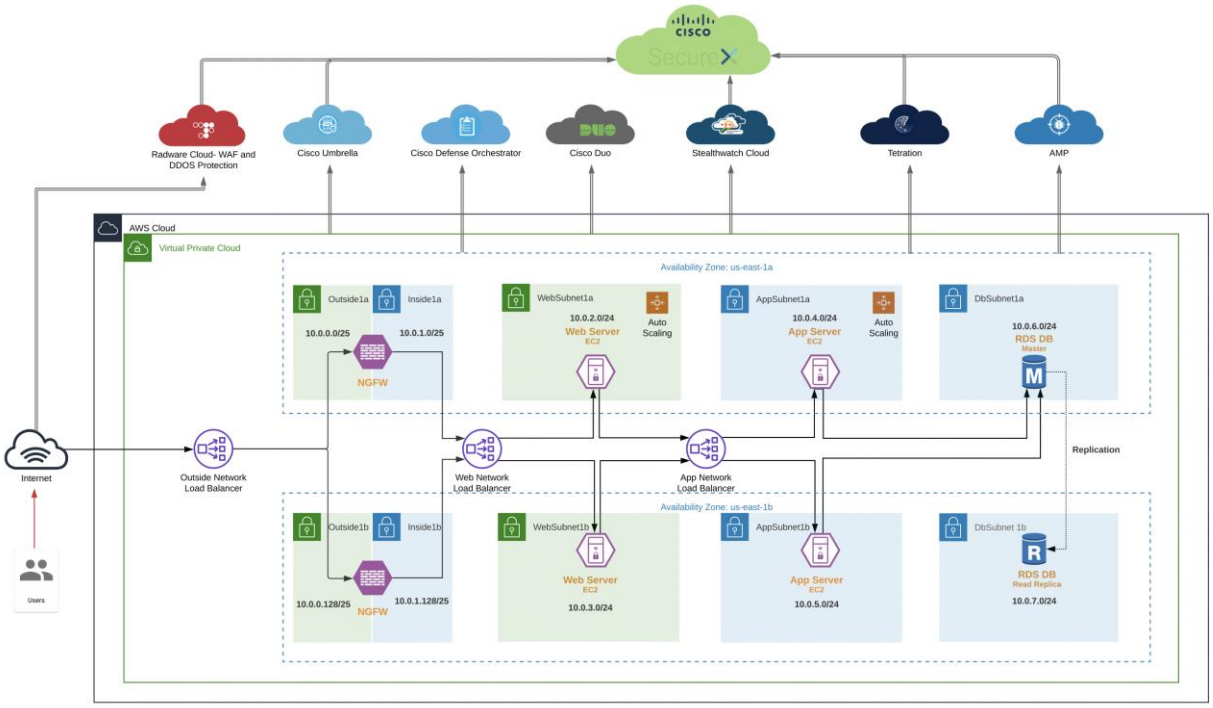


Figure 20.
Cisco SecureX threat response

In this architecture, we are receiving information from Stealthwatch Cloud, Umbrella, AMP and Tetration to provide threat intelligence, contextual approach, and threat hunting capabilities. Integrations for and Radware Cloud WAF and DDoS service are also available. Refer to the [Cisco SecureX]() documentation for more details on available Cisco and third-party integrations.

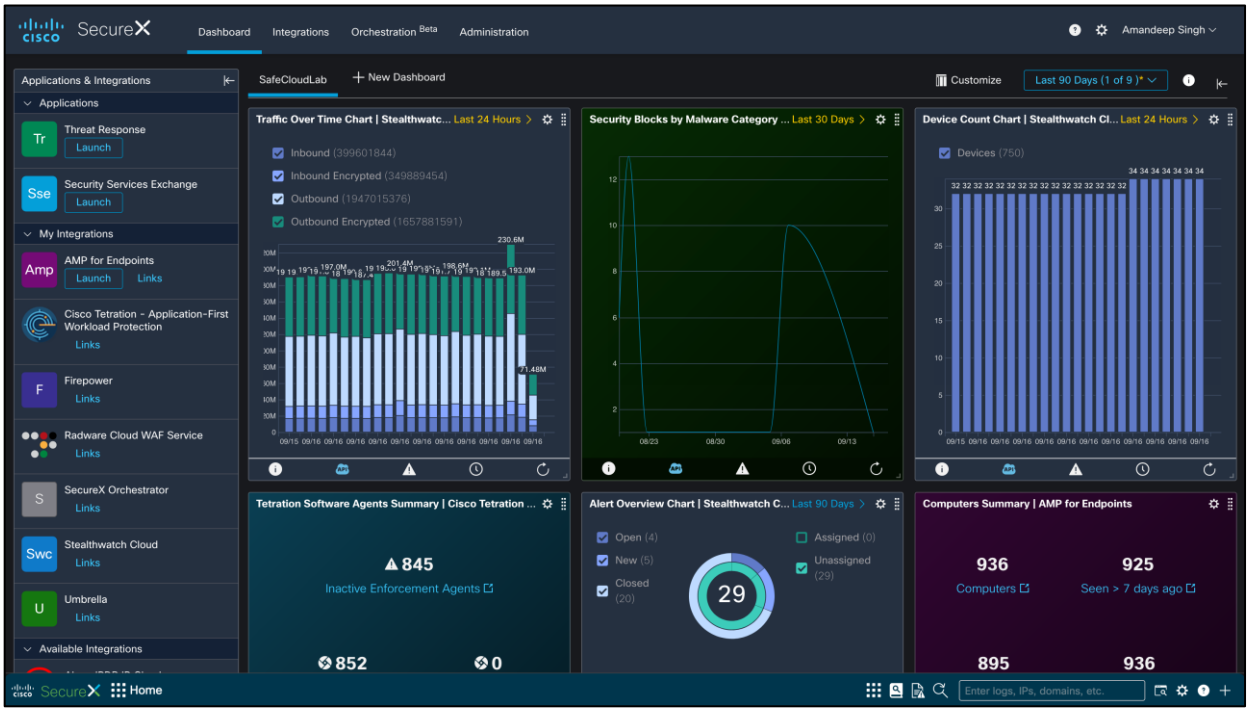


Figure 21.
Cisco SecureX Dashboard

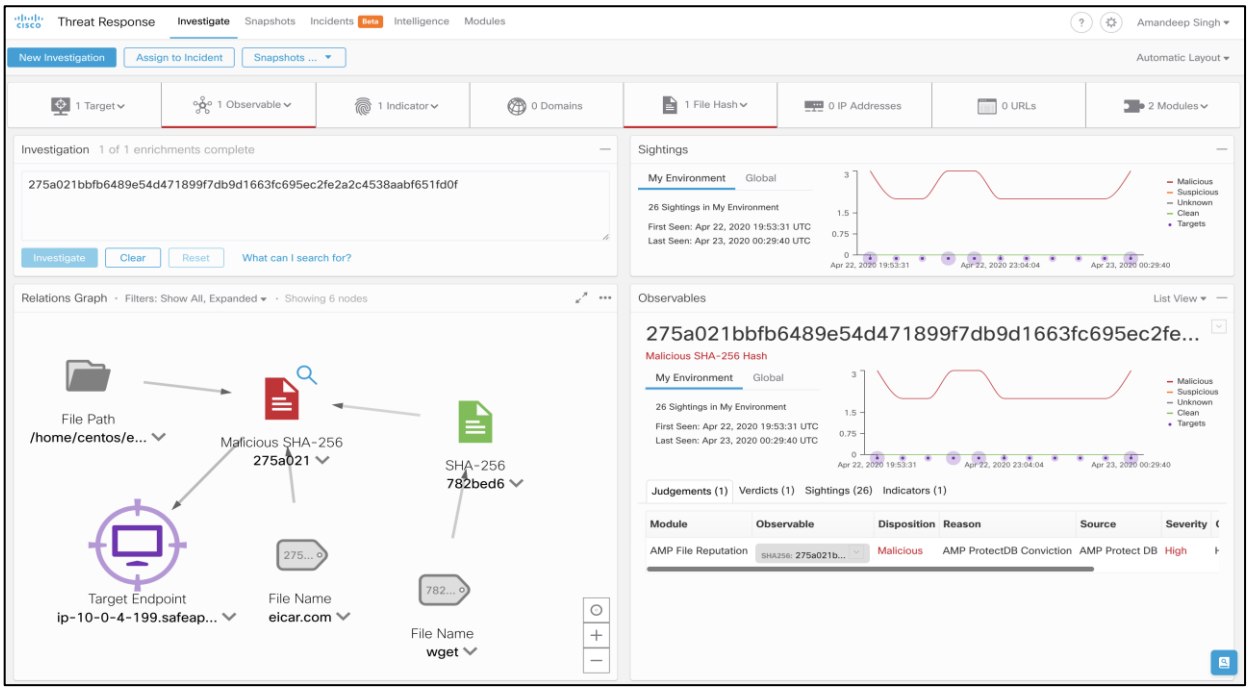


Figure 22.
Cisco SecureX threat response - Threat Hunting

Design Implementation

Now that we have established the design specifics of our tiered application in the AWS cloud, we will begin implementing and setting up the secure AWS application.

We will start by setting up the AWS VPC (Virtual Private Cloud) as per the tiered architecture specifications. We will then integrate the Stealthwatch Cloud and onboard the VPC to CDO for Security Group management. After that we will set up the Umbrella VAs in the management tier and update the DNS server settings for the VPC.

Once the AWS VPC and related integrations are finished, we will configure an RDS database instance and bring up the Auto Scaling Groups for the Application and Web workloads (with Tetration, AMP4E agents and Duo MFA plugin). We will then set up the Network Load Balancers for Web and Application Auto Scaling Groups. At this point we will have a fully functional application running in the AWS cloud.

In the last step, we will configure the firewalls, enable WAF and DDoS protection and then conclude our set up with Cisco SecureX integration.

Note: Cisco Tetration, AMP, Cisco SecureX threat response, Stealthwatch Cloud, Umbrella, Duo and CDO offer EU based locations for customers having to follow EU rules.

Deployment Overview:

- Set up the AWS VPC components
- Integrate Stealthwatch Cloud for VPC monitoring
- Onboard the AWS VPC to CDO for AWS Security Group management
- Set up Umbrella DNS Security
- Set up the AWS RDS database instance
- Set up the Auto Scaled Application and Web Workloads (Tetration, AMP4E agent and Duo MFA plugin installation) with App and Web NLBs
- Set up Cisco Firepower Next-Generation Firewalls with CDO onboarding
- Enable Radware cloud web application firewall and DDoS prevention service
- Set up Cisco SecureX

Note: Before you begin, make sure you have the appropriate privileges to create all the VPC components. Follow the AWS Documentation for more information on IAM service.

Set up the AWS VPC components

We will create a new AWS VPC and configure all the associated components that we need for our deployment in this section.

Implementation procedure:

Step 1. Create the VPC

Step 2. Set up the Subnets

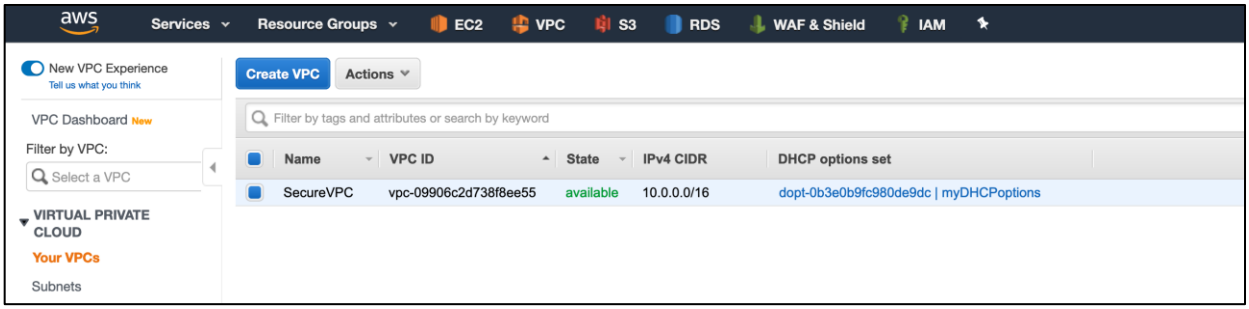
Step 3. Set up the Internet gateway

Step 4. Set up the NAT gateways

Step 5. Set up the Routing Tables

Step 6. Create the Security Groups

Step 1. Create the VPC - Log on to the AWS console and select the VPC service, click on Create VPC and fill in the required details. We chose the IPV4 CIDR block as 10.0.0.0/16.

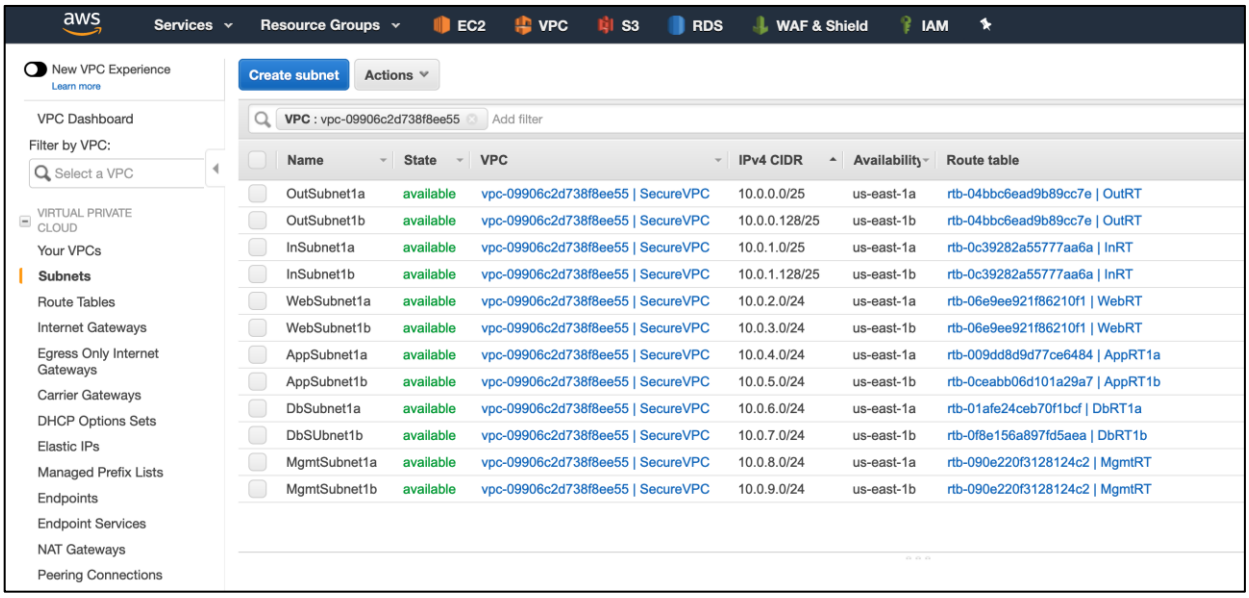


Follow the AWS documentation for more details on AWS VPCs.

Step 2. Set up the Subnets - Based on the tiered architecture, we defined two subnets for each tier - one for each AWS Availability Zone.

IPV4 CIDR Block	AWS Region	Tier
10.0.0.0/25	US-East-1a	OutSubnet1a
10.0.0.128/25	US-East-1b	OutSubnet1b
10.0.1.0/25	US-East-1a	InSubnet1a
10.0.1.128/25	US-East-1b	InSubnet1b
10.0.2.0/24	US-East-1a	WebSubnet1a
10.0.3.0/24	US-East-1b	WebSubnet1b
10.0.4.0/24	US-East-1a	AppSubnet1a
10.0.5.0/24	US-East-1b	AppSubnet1b
10.0.6.0/24	US-East-1a	DbSubnet1a
10.0.7.0/24	US-East-1b	DbSubnet1b
10.0.8.0/24	US-East-1a	MgmtSubnet1a
10.0.9.0/24	US-East-1b	MgmtSubnet1b

Go to **VPC Dashboard > Subnets** and create all these subnets and name them appropriately.



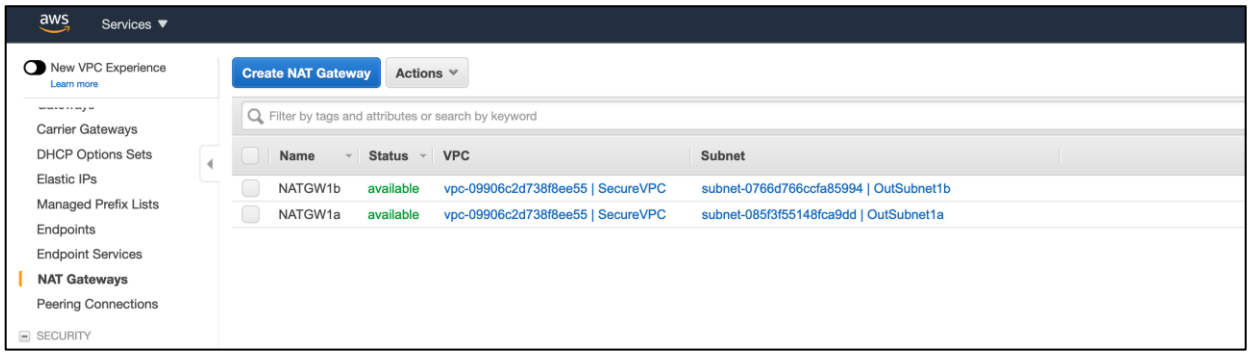
Step 3. **Set up the Internet gateway**– Navigate to **VPC Dashboard > Internet Gateways** to create an Internet Gateway for providing Internet access to Public resources in the VPC.

Note: We will use this Internet Gateway as the next hop for default routes in Firewall Route table (For Inside and Outside subnets) and the Management Route Table (For Management Subnets) respectively.

Follow the AWS documentation for more details on AWS Internet Gateway components.

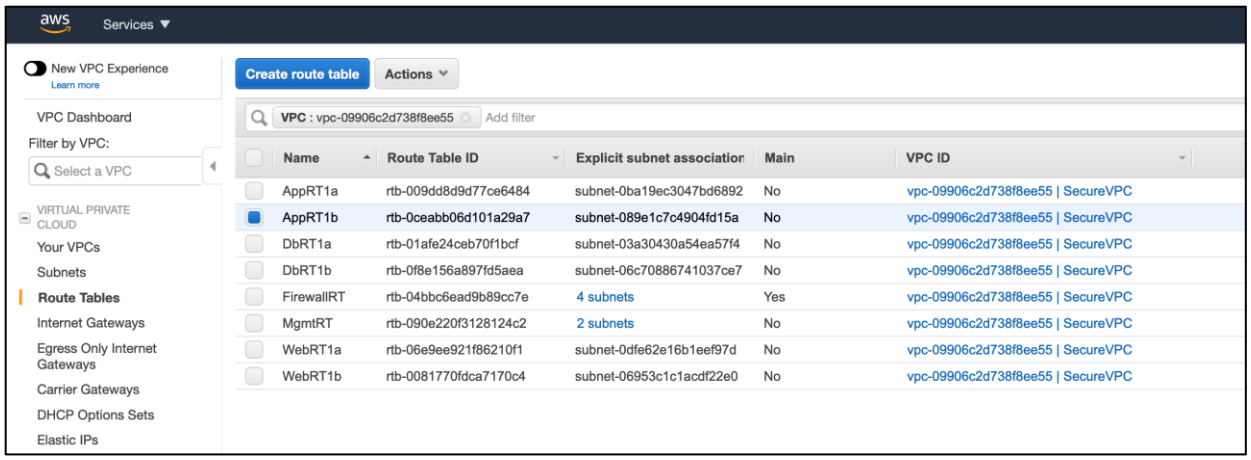
Step 4. **Set up the NAT gateways** - Navigate to **VPC Dashboard > NAT Gateway** to create NAT Gateways for providing Internet access to all resources in private subnets.

Note: We will use these NAT Gateways as the next hop for default routes in Web, App and Db Route tables.



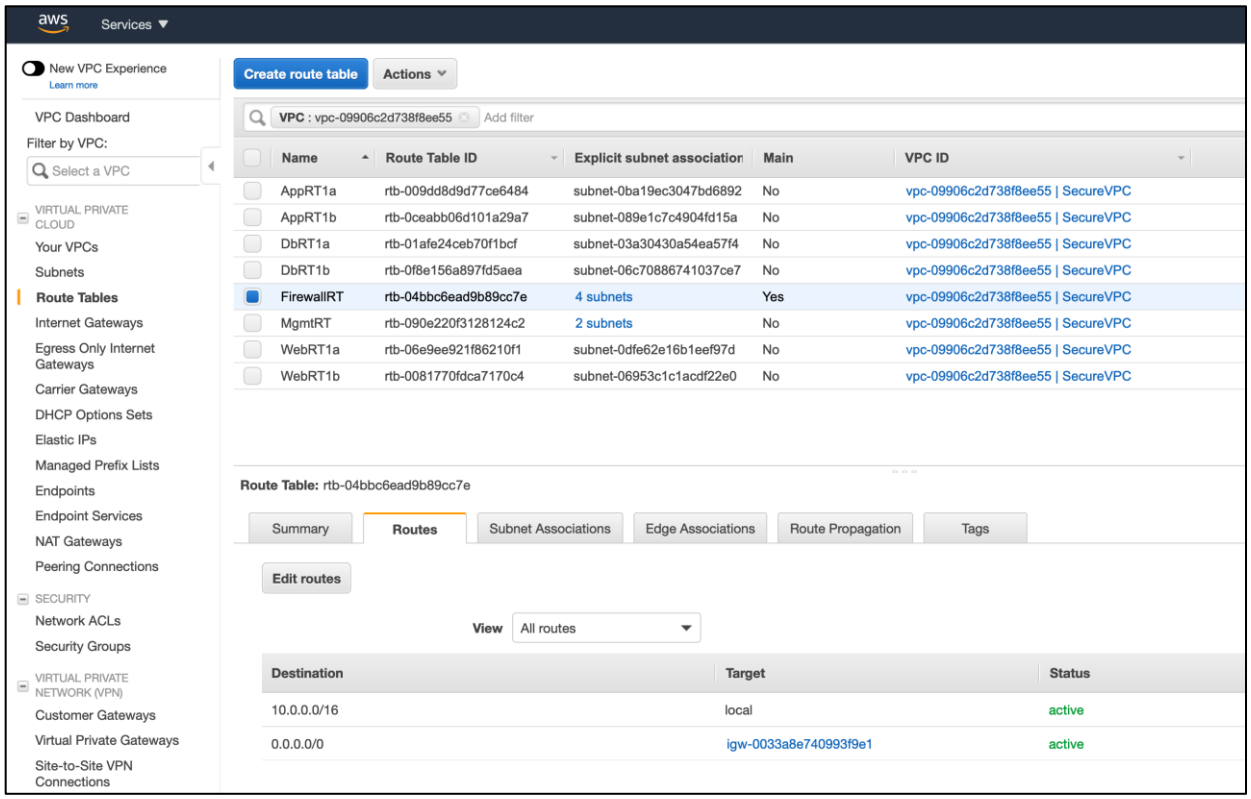
Step 5. **Set up the Routing Tables** - Go to VPC Dashboard > Route Tables and create the routing tables with subnet associations as per the table below.

Route Table Name	Subnets	Default Route
FirewallRT	OutSubnet1a, OutSubnet1b InSubnet1a, InSubnet1b	Internet Gateway
WebRT1a	WebSubnet1a	NAT Gateway1a
WebRT1b	WebSubnet1b	NAT Gateway1b
AppRT1a	AppSubnet1a	NAT Gateway1a
AppRT1b	AppSubnet1b	NAT Gateway1b
DbRT1a	DbSubnet1a	NAT Gateway1a
DbRT1a	DbSubnet1b	NAT Gateway1b
MgmtRT	MgmtSubnet1a, MgmtSubnt1b	Internet Gateway



For the Firewall and Management Route Tables, create the default route pointing to the Internet Gateway created previously. For the Web, App and Db Route Tables, create the default route pointing to the NAT Gateways created for each Availability Zone.

As per the AWS network design, we cannot load balance the outbound flows. If you decide to deploy a single firewall per availability zone then you can always use firewall inside interface ENI as next hop for default routes, to send outbound flows to firewalls. However, we recommend adding multiple firewalls in each availability zone to avoid any single point of failure.



Step 6. **Create the Security Groups** – Go to **'VPC Dashboard > Security Groups'**, set up a Security Group corresponding to each tier in the design. Set up the inbound access rules as per the application requirements. We used the following inbound rules.

webSG	
Port	**Reason**
TCP port 80	Allow HTTP access from the inside subnets to web servers
TCP port 22	Allow SSH access from mgmtSG

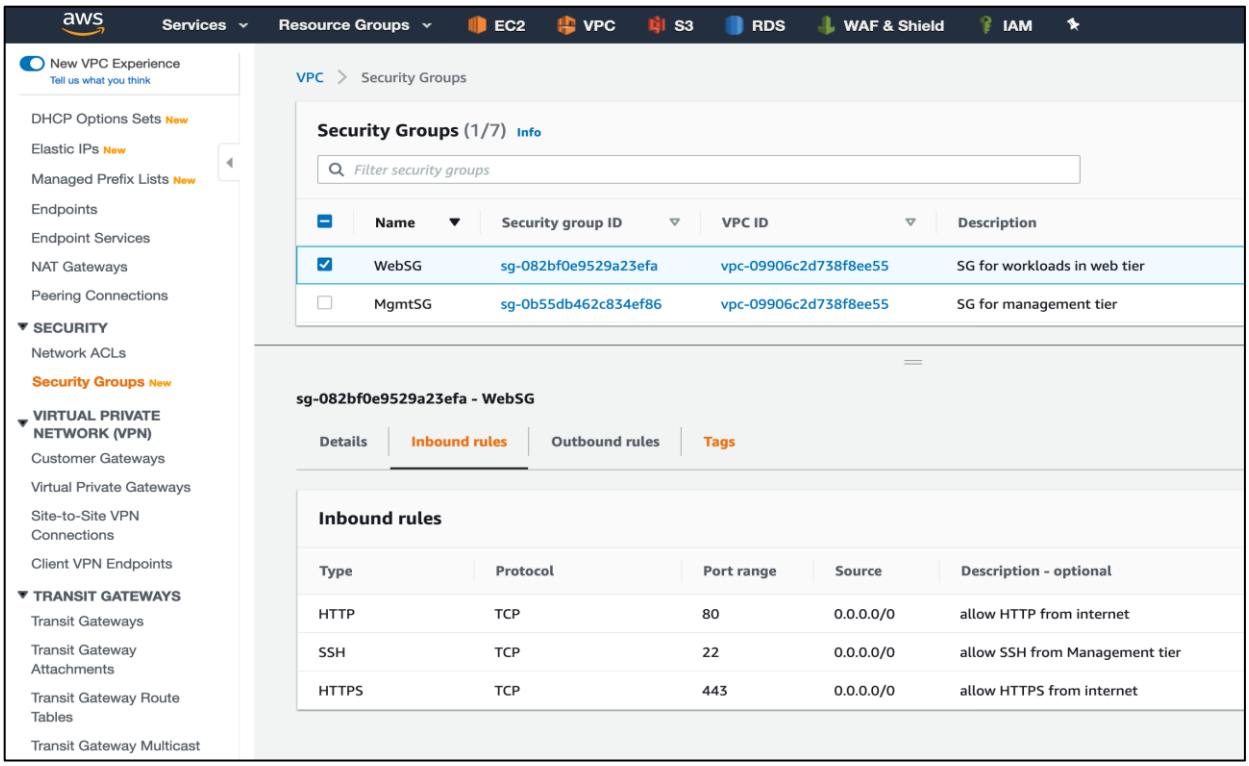
appSG	
Port	**Reason**
TCP port 80	Allow HTTP access from workloads within application tier subnets (allows load balancer health checks). Allow HTTP access from web tier subnets.
TCP port 22	Allow SSH access from mgmtSG

dbSG	
Port	**Reason**
TCP port 3306	Allow MYSQL/Aurora traffic from appSG

mgmtSG	
Port	**Reason**
TCP port 22	Allow SSH access from internet
UDP 53	Allow DNS traffic from appSG, webSG, dbSG and mgmtSG

firewallSG	
Port	**Reason**
All traffic	Allow all access from internet, we will control the traffic using firewall access lists.

Note: AWS Elastic Network Load Balancer (NLB) preserves the source IP of incoming connections from web tier workloads, hence we need to allow the source subnets specifically. We cannot use Security Groups to allow traffic from NLB, we must use subnets. Follow the AWS Documentation for more information on Security Group requirements for NLB.



Follow the AWS documentation for more details on AWS Security Groups.

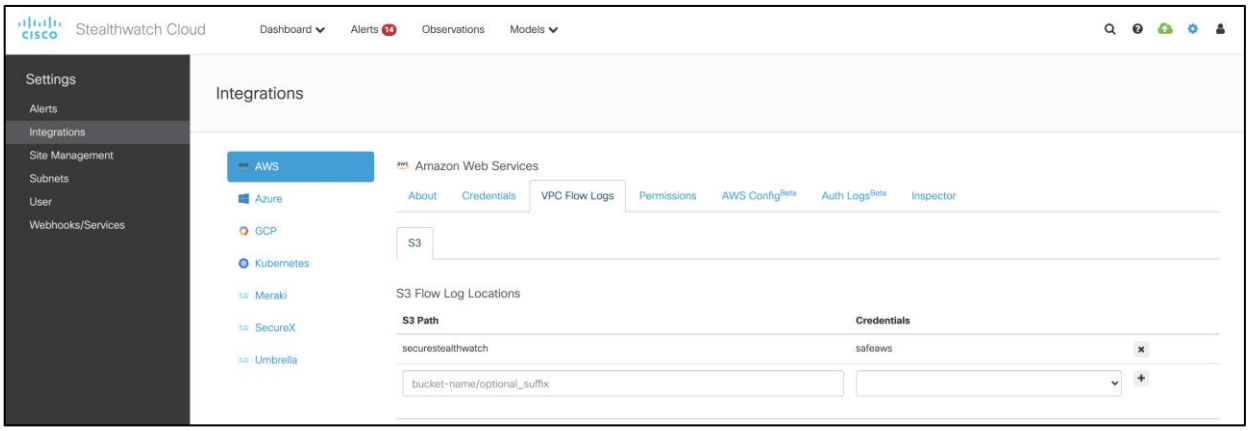
Integrating Stealthwatch Cloud

Implementation procedure:

Step 1. Set up the VPC flow logs and integrate Stealthwatch Cloud.

Step 1. Set up the VPC flow logs and integrate Stealthwatch Cloud – Follow the steps illustrated in Cisco Stealthwatch AWS Quick Start Guide to create the VPC flow logs and other required AWS resources for Stealthwatch cloud monitoring.

After the Stealthwatch cloud integration is done, click on the cloud icon on the top right hand side of the SWC portal and you should see an AWS sensor with a green check mark against it, indicating a successful integration.

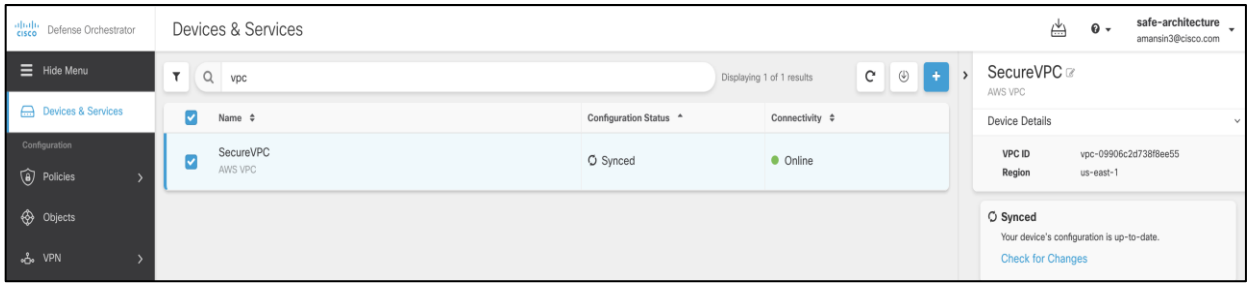


Onboard AWS VPC to Cisco Defense Orchestrator

Implementation procedure:

Step 1. Onboard the AWS VPC to Cisco Defense Orchestrator

Step 1. Onboard the AWS VPC to Cisco Defense Orchestrator – Follow the steps illustrated in CDO Documentation to onboard the AWS VPC. Once the onboarding is complete, CDO can be used to manage the AWS Security Groups.



Set up Umbrella DNS Security

Implementation procedure:

Step 1. Set up the Umbrella Virtual Appliance (VA) image

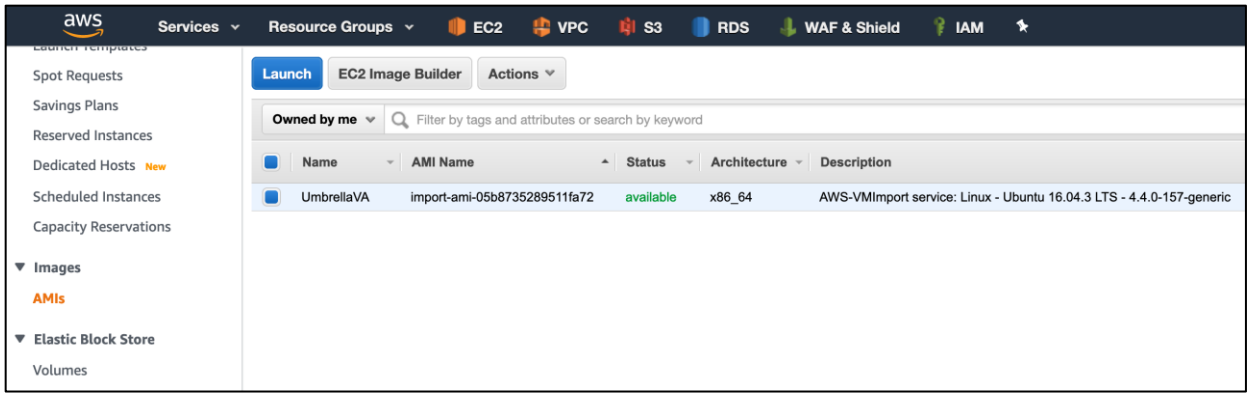
Step 2. Create the Umbrella VA instances

Step 3. Configure the local DNS on Umbrella VA instances

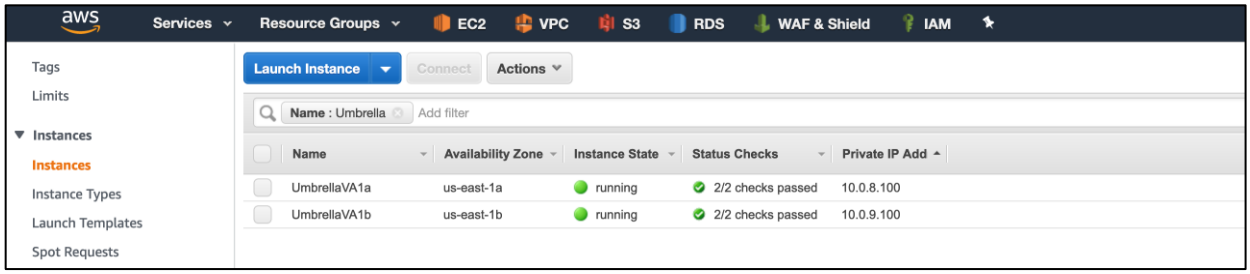
Step 4. Set up the policies to exempt internal domains

Step 5. Update the DHCP Options Set for VPC

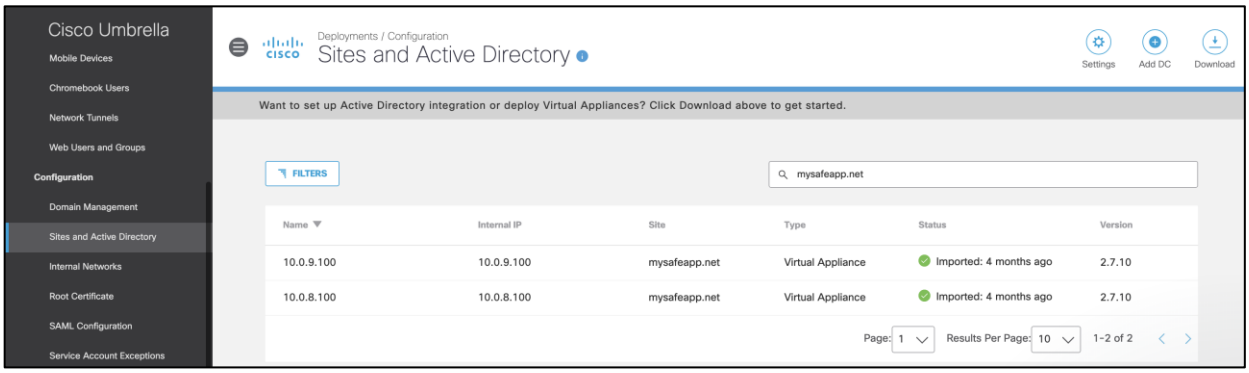
Step 1. Set up the Umbrella Virtual Appliance image – Follow the Umbrella documentation to deploy Virtual appliances (VA) in the AWS cloud. As per the documentation, create an AWS AMI and then use it to launch VA instances.



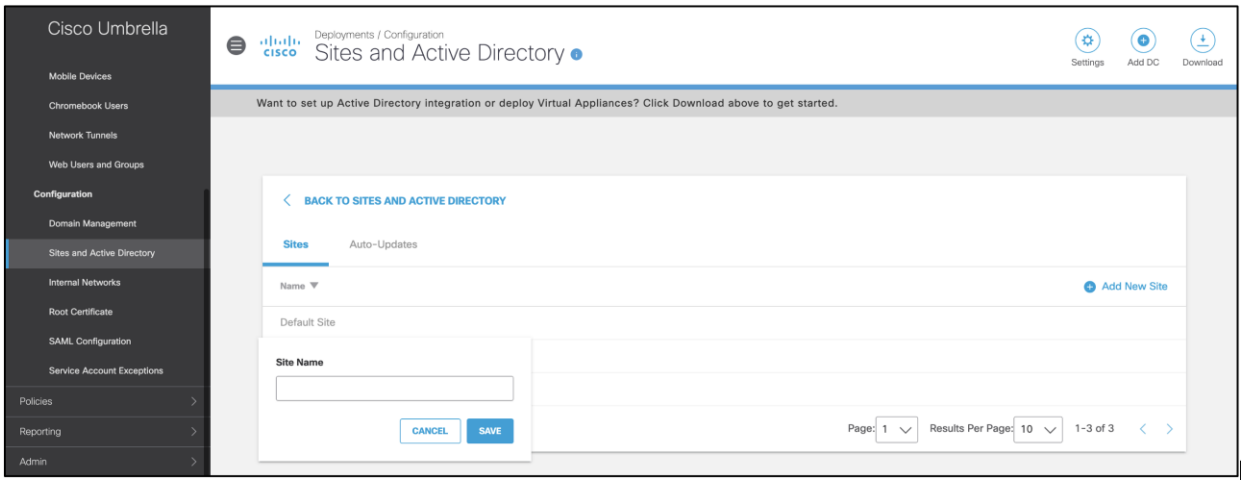
Step 2. Create the Umbrella VA instances – Create two VA instances using the AMI set up in Step1 and place these appliances in the management tier. We assign the static IP addresses 10.0.8.100 and 10.0.9.100 to these Umbrella Virtual Appliances. These VAs will act as DNS forwarders for the resources in our AWS application environment.



Once the appliances are fully up in AWS, login to the Umbrella portal and verify the green status under **Deployments > Configuration > Sites and Active Directory**.



Optionally, you can create and assign a site name for your AWS VAs. This site name can be used as an identity to configure specific policies for AWS Cloud. Click on Settings on the same page to add site name and then update the VA entries above.



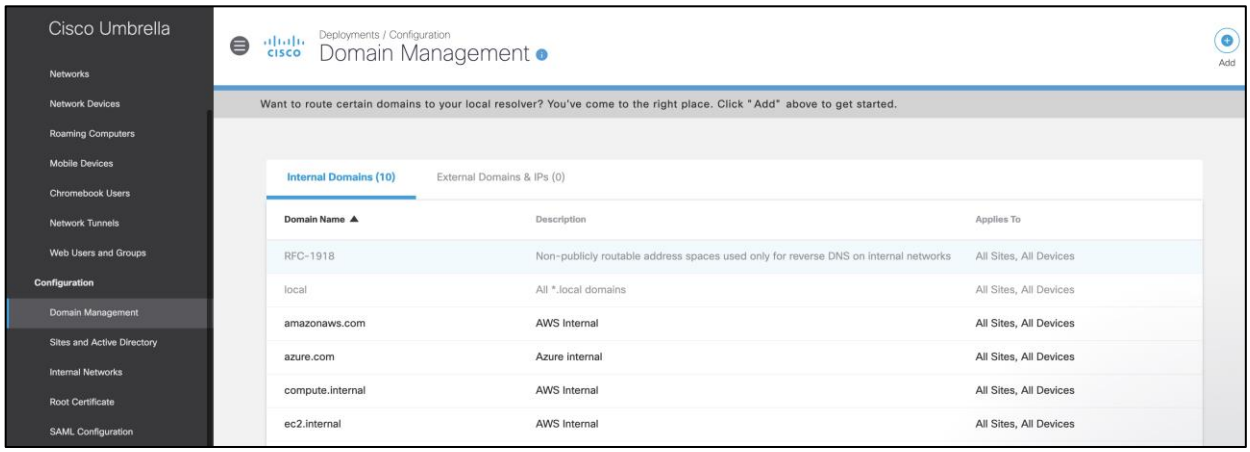
Step 3. **Configure the local DNS on Umbrella Virtual Appliances** – Follow the Umbrella documentation to configure local DNS on each VA. Based on the CIDR block chosen for lab VPC, the second IP address i.e. 10.0.0.2/24 is the local DNS. Set this IP as local DNS on both Umbrella VAs.

Note: We had set up Secure Remote Access to management tier using ASAv, we use the secure VPN connection to SSH into the VAs via a jump server hosted in the management tier. For more information on Secure Remote Access, refer to the Secure Remote Worker SAFE Design guide.

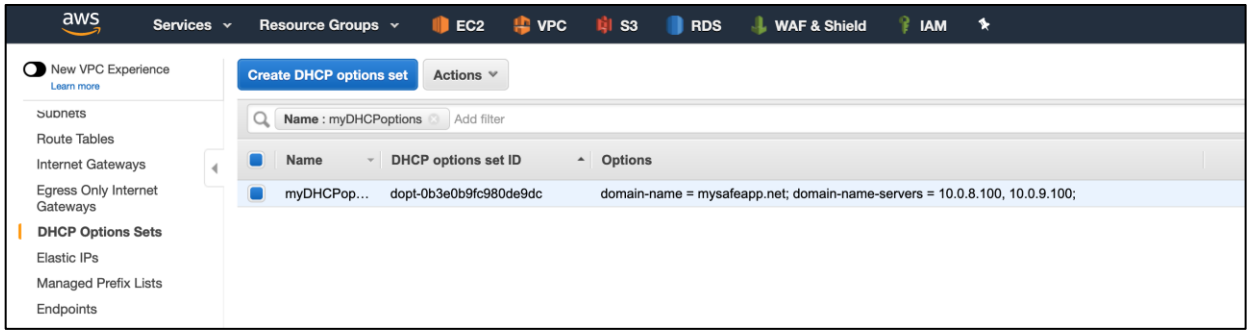
```
10.0.9.100   $
10.0.9.100   $ config va localdns 10.0.0.2
localdns: 10.0.0.2 -> 10.0.0.2
10.0.9.100   $
10.0.9.100   $ config va show
          Virtual Appliance Configuration

              Name:  10.0.9.100

              Local DNS -
                  ip address : 10.0.0.2

              Internal Domains Count: 4

              SSH access : enabled

              Primary Adapter   : eth0
                  MAC Adress    : 06:28:33:65:0b:18
                  IP Address    : 10.0.9.100
                  Netmask       : 255.255.255.0
                  Gateway       : 10.0.9.1


10.0.9.100   $
10.0.9.100   $
```

Step 4. **Set up policies to exempt internal domains** – Log on to the Umbrella portal, go to **Deployments > Configuration > Domain Management** and add the internal domains that

should be routed to the local AWS resolver. Based on your set up, the list of internal domains will vary.



Step 5. **Update the DHCP Options Set for the VPC** – Go to **VPC Dashboard > DHCP Options Sets** and create a new DHCP options set. Set the domain name servers to two IPs that we assigned to Umbrella VAs – 10.0.8.100 and 10.0.9.100.



Go to **VPC Dashboard > Your VPCs**, select the newly created VPC above and update the DHCP options set from the drop-down list. This will ensure that any instance deployed in this VPC is assigned the Umbrella VAs as DNS forwarders.

Follow the [AWS documentation]() for more details on DHCP Options Sets.

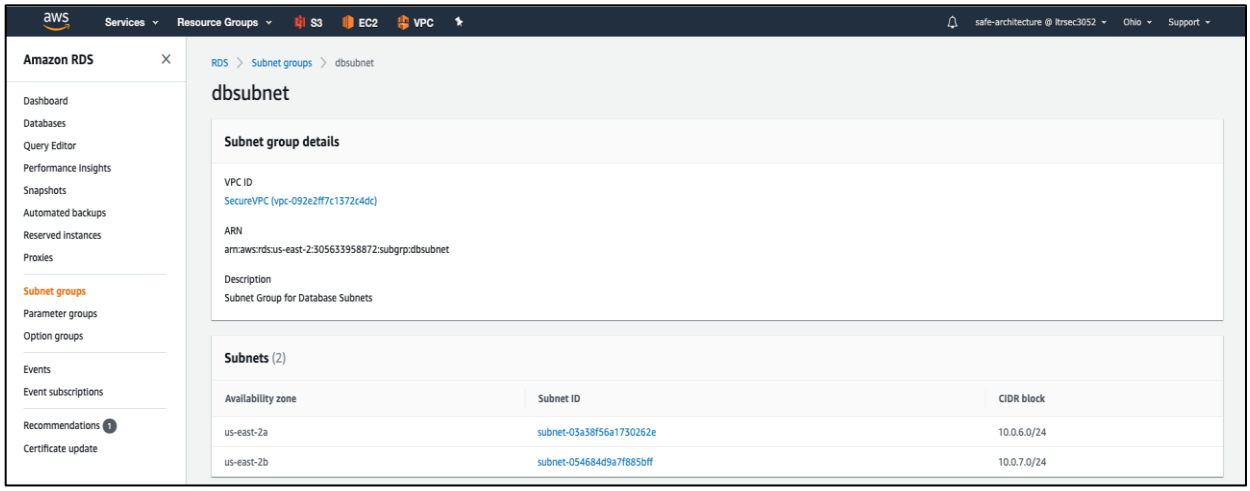
Setting up the RDS database

Implementation procedure:

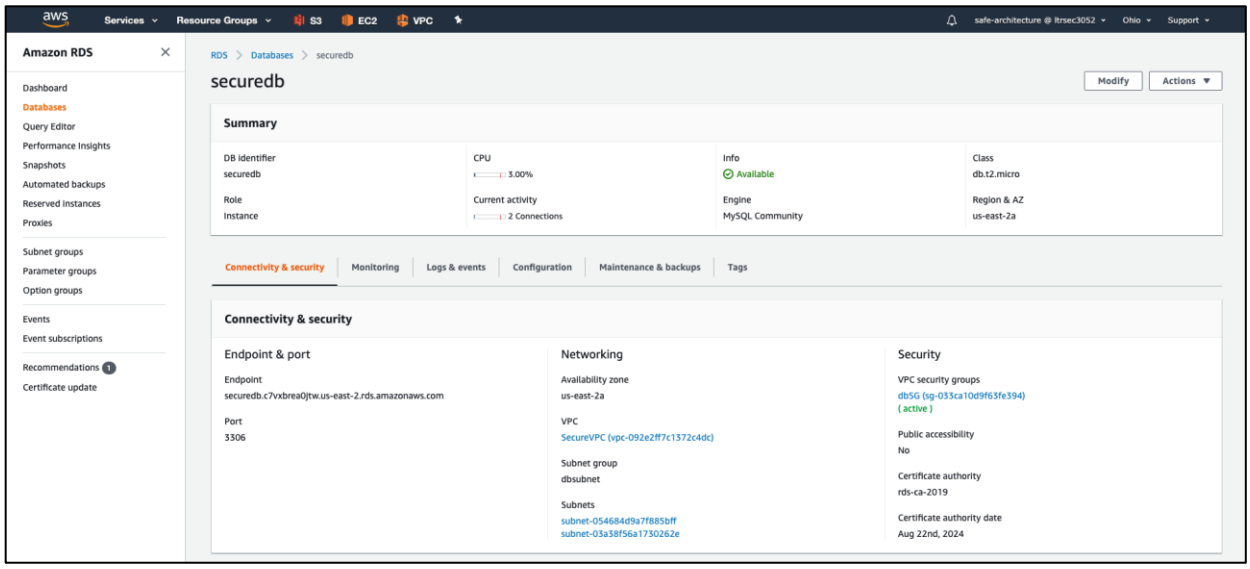
Step 1. **Define the database Subnet Groups**

Step 2. **Set up the RDS database instance**

Step 1. **Define the database Subnet Groups** – Go to **RDS > Subnet Groups** on the AWS console, create a **DB Subnet Group** and add to it the Database tier subnets defined in previous steps ('dbSubnet1a' and 'dbSubnet1b').



Step 2. **Set up the RDS database instance** – Set up the database instance as per your application requirements, follow the AWS documentation for further help. Use the Subnet Group defined in Step 1 above. The database instance is placed in Security Group 'dbSG'.



Make sure you note the username, password, endpoint hostname and port, we need these details to set up our cloud application later in this section.

Setting up the App and Web Load Balancers

Before we begin our implementation, please ensure you're familiar with these components. For information on AWS Elastic Load Balancing and health checks, check out the AWS documentation here. Follow the AWS documentation for detailed configuration steps for Network Load Balancer (NLB).

Per our tiered design, we will set up a 'Web' Network Load Balancer (NLB) for the Web Server workloads and an 'App' Network Load Balancer (NLB) for the Application workloads. We will create Target Groups for each NLB; the workloads register themselves with these Target Groups.

We will not register any instances to the Target Groups at this point but in the next section when we create the Auto Scaling Groups, we will integrate the Auto Scaling Groups with each of these blank Target Groups that we

create in this section. When the Auto Scaling process spins new instances, they will automatically register with these Target Groups.

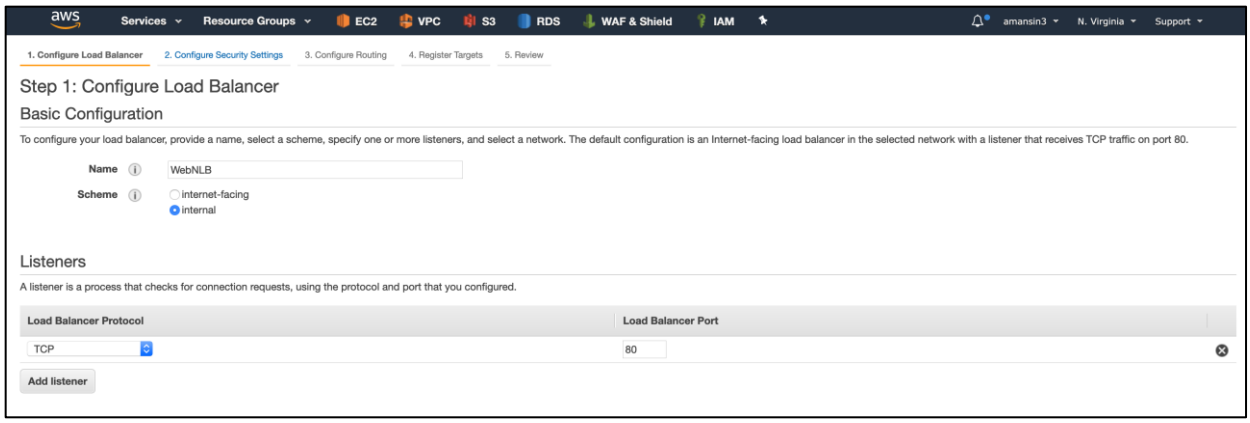
The Load Balancers will be configured to run health checks for each instance that is launched into the Target Groups. As soon as an instance is marked healthy, the Load Balancer starts load balancing traffic to it. When the instance becomes unhealthy, it is removed from the pool.

Implementation procedure:

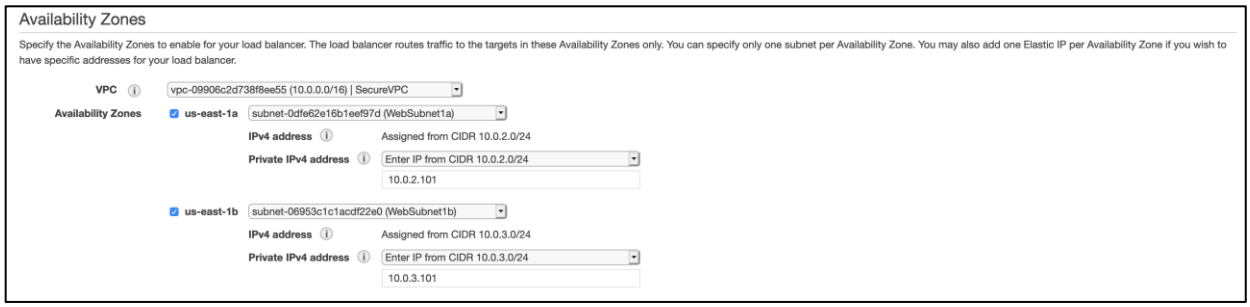
Step 1. Set up the Web Network Load Balancer

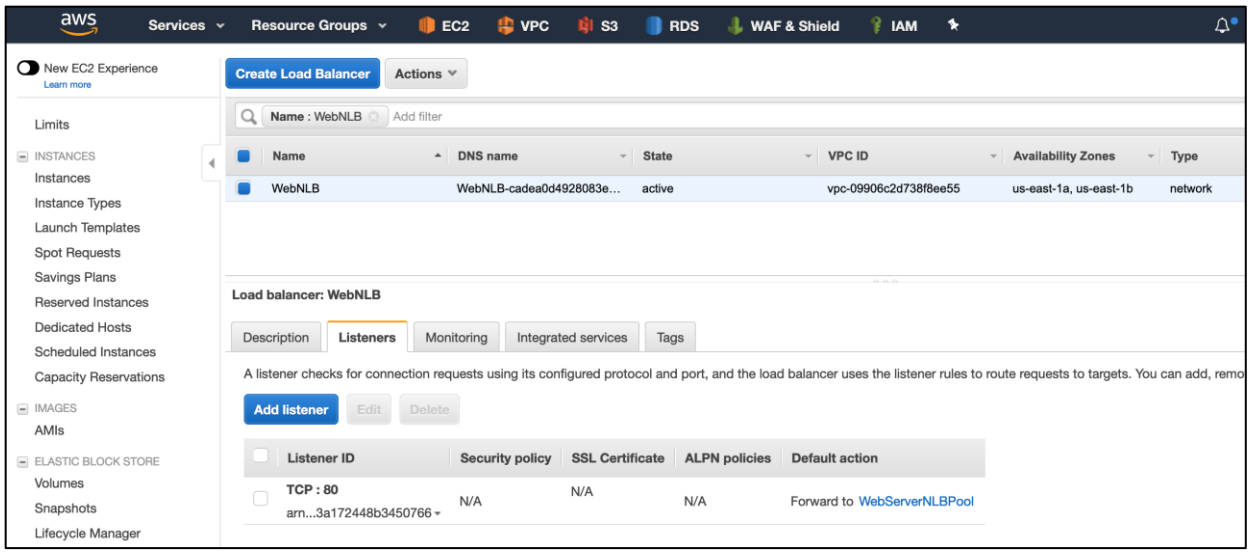
Step 2. Set up the App Network Load Balancer

Step 1. Set up the Web Network Load Balancer – The Web NLB is used to load balance web traffic coming from the Internet to the Auto Scaled pool of Web workloads. This load balancer is placed in the 'webSG' Security Group in the subnets – 'WebSubnet1a' and 'WebSubnet1b'. A new Target Group 'WebServerPool' is also created as part of load balancer configuration, this will be used later while setting up Auto Scaling Group for Web Servers.

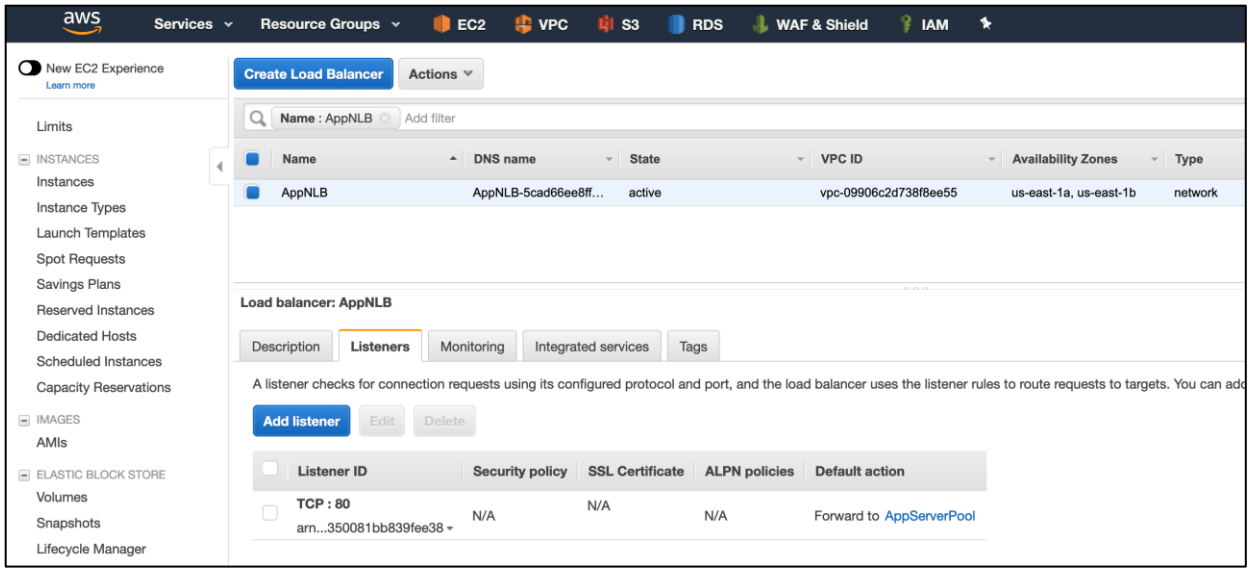


Assign static IP addresses in each availability zone. Make a note of these IP addresses, we will require these IPs while setting up the static NAT translations on FTD appliances in later section of this document.





Step 2. **Set up the App Network Load Balancer** – This Load Balancer is placed in the 'appSG' Security Group in the subnets – 'appSubnet1a' and 'appSubnet1b'. A new Target Group 'AppServerPool' was also created as part of Load Balancer configuration, this will be used later while setting up the Auto Scaling Group for Application workloads.



Setting up Web and App Auto Scaling groups

In this section, we will set up a pool of workloads for the web and application tier. We will use the AWS Auto Scaling Groups to achieve this. As part of workload initialization, we install Tetration and AMP4E agents on the web and app workloads along with other application-specific packages, including the Duo plugin.

Implementation procedure:

Step 1. **Host the configuration files in an S3 bucket**

Step 2. **Set up Launch Configurations**

Step 3. **Set up the Auto Scaling Groups**

Step 4. **Configure the Auto Scaling policies**

Step 1. **Host the configuration files in an S3 bucket** – We set up an AWS S3 bucket in US East region. We upload all the files and application code that we need for our application and web workloads into this bucket. The other option is to use golden images with all the required applications and packages pre-installed for app and web workloads. If you choose to host config files in an S3 bucket, ensure that you set appropriate access privileges for these files.

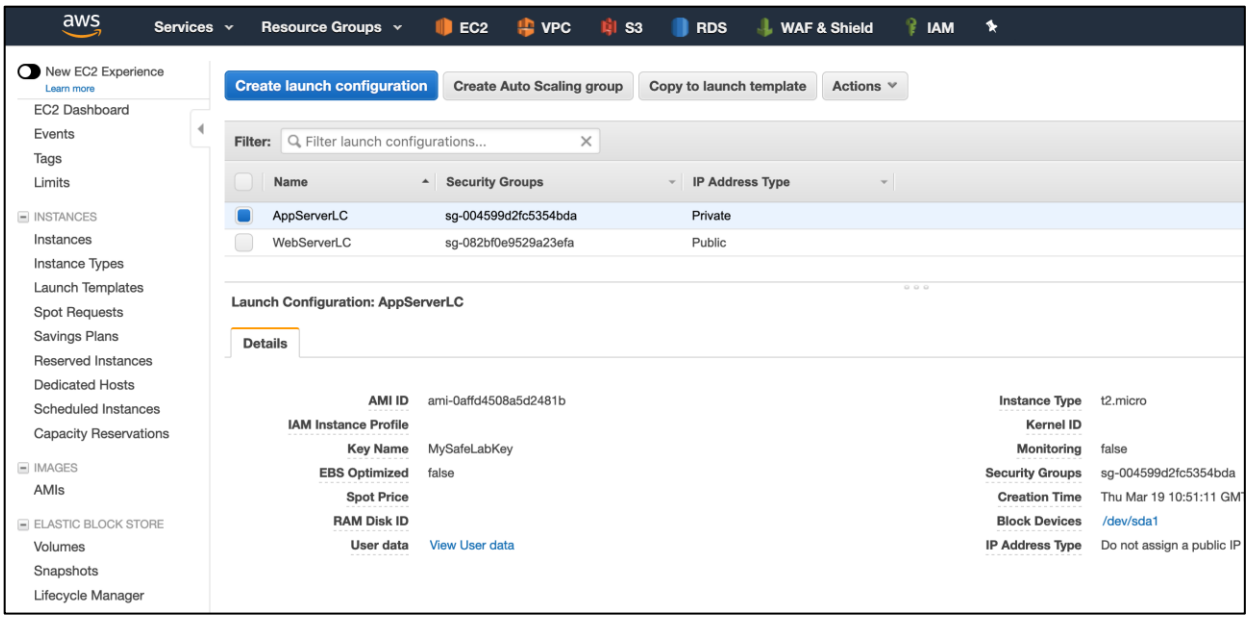
The S3 files include:

- Config/Code files for our App and Web workloads
  - App workloads – Modified Publicly available 'WordPress' blog code with database connection information (we recorded the database credentials while setting up RDS previously). Duo Plugin was also hosted in this S3 bucket.
  - Web workloads – The Web Server config file. This file has the FQDN address of App Network Load Balancer that we created in previous steps.
- AMP4E agent installer (AMP rpm and GPG). These were obtained from AMP cloud portal
- Tetration agent installer (Enforcement agent). These are downloaded from Tetration SaaS portal

Note: For Duo MFA for the cloud application, we used Duo WordPress plugin. However, if you choose to include the Duo integration in your native application, follow the DUO Web SDK documentation.

Step 2. **Set up the Launch Configurations** – Go to **EC2 Dashboard> Auto Scaling > Launch Configuration** and create launch configurations for web and application servers. For more information on creating Launch Configurations follow the AWS documentation.

- For the base image/operating system, we chose **CentOS**
- Under the **Advanced details** options, make sure not to assign any public IPs to the instances in the launch configuration
- Under the same **Advanced details** options, we use the **User Data** option to initialize the EC2 instances when they are launched into the auto scaling pool. For more details on **User Data** option, check out the AWS documentation. As part of this initialization process, we perform the following tasks:
  - Install packages (php, wget, unzip, lsof, httpd, ipset, nginx) on the workloads. Some of these are prerequisites for AMP4E and Tetration agents. Refer to the corresponding product documentation to understand these requirements
  - Download Code/Configuration files to the respective workloads from S3 bucket
  - Download the WordPress Duo plugin from S3 bucket to the application workloads
  - Download and install the Tetration enforcement agent to all the workloads
  - Download and install the AMP4E agent to all the workloads
- Use the Security Groups 'webSG' and 'appSG' for Web and Application Launch Configurations respectively



Below are sample User Data scripts that we used.

Web server initialization script:

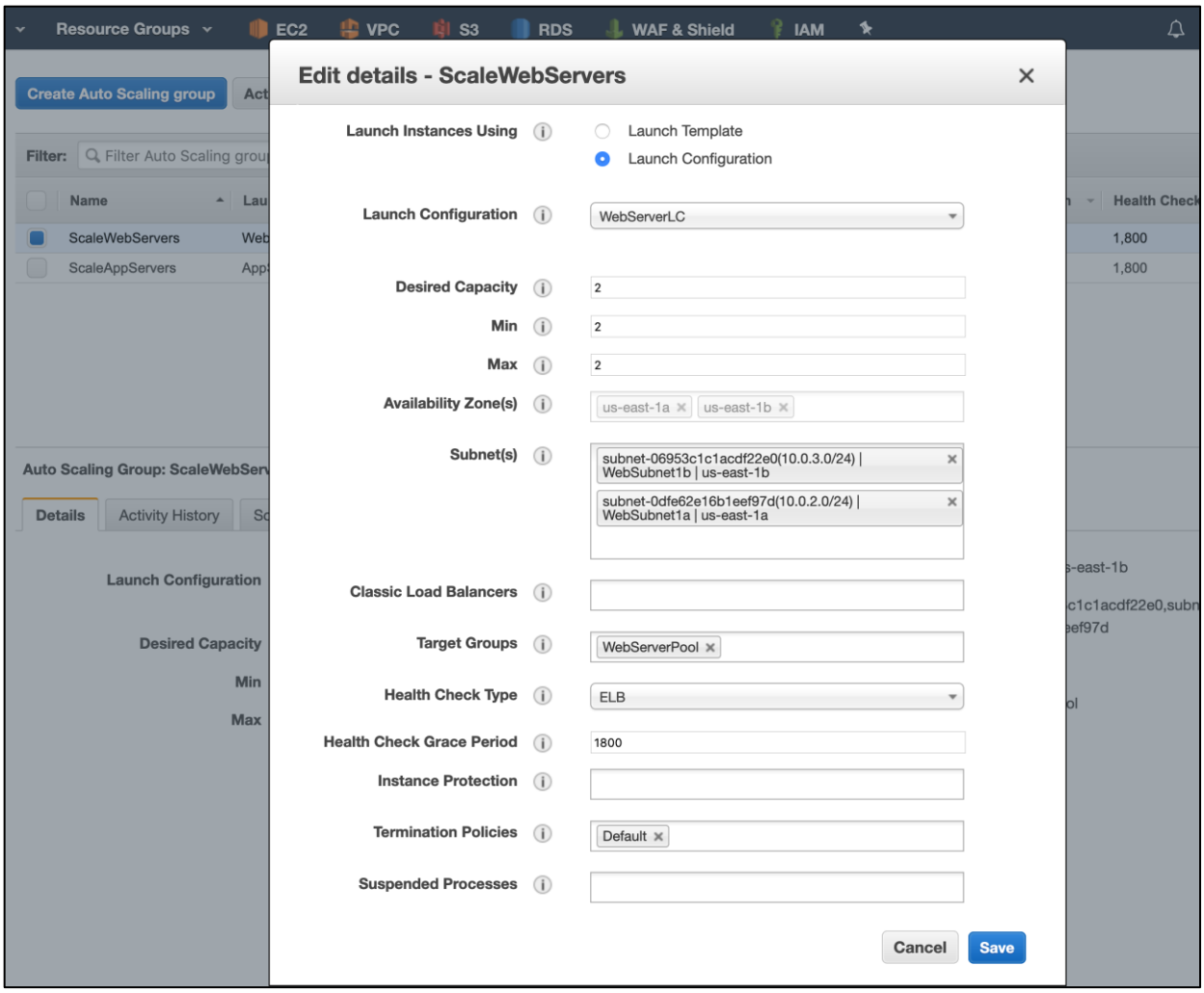
```
#!/bin/bash
sudo yum install -y wget
sudo yum install -y unzip
sudo yum install -y lsof//    lsof utility is required for enforcing tetration policies
sudo yum install -y ipset//   ipset utility is required for enforcing tetration policies
sudo yum install -y nginx//   Installing nginx

#Setting up the web server and updating it with hosted configuration file.
sudo mv nginx.conf nginx.conf.backup
sudo wget https://safelabfiles.s3.us-east-2.amazonaws.com/config/nginx.conf
sudo systemctl restart nginx
sudo systemctl enable nginx

#Downloading the Tetration enforcement agent from AWS S3 bucket and installing it.
sudo wget https:// safelabfiles.s3.us-east-
2.amazonaws.com/config/tetration_installer_intgssopov_enforcer_linux.sh
sudo chmod 755 tetration_installer_intgssopov_enforcer_linux.sh
sudo ./tetration_installer_intgssopov_enforcer_linux.sh --skip-pre-check

#Downloading the AMP4E agent hosted in an AWS S3 bucket and installing it.
sudo wget https:// safelabfiles.s3.us-east-2.amazonaws.com/config/cisco.gpg
sudo rpm --import ./cisco.gpg
sudo wget https:// safelabfiles.s3.us-east-2.amazonaws.com/config/AWS_rhel-centos-
7fireamplinux_connector.rpm
sudo yum install -y AWS_rhel-centos-7fireamplinux_connector.rpm
```

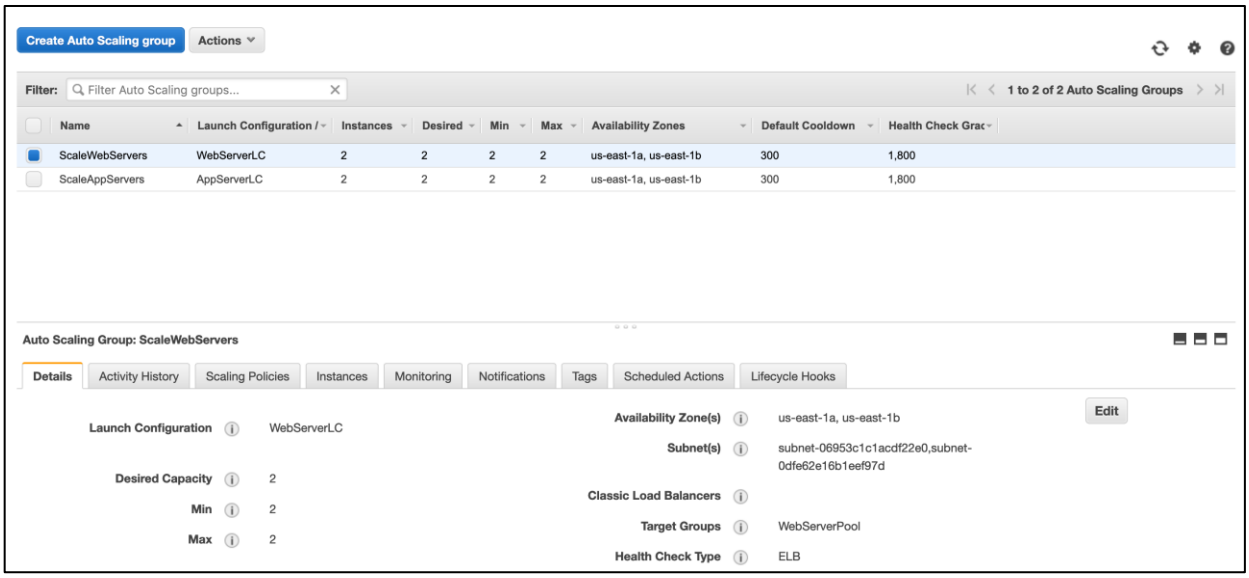
Application server initialization script:

```
#!/bin/bash
sudo yum install -y wget
sudo yum install -y unzip
sudo yum install -y lsof//    lsof utility is required for enforcing tetration policies
sudo yum install -y ipset//   ipset utility is required for enforcing tetration policies
sudo yum install -y httpd//   Installing httpd
#Setting up the HTTPD server and downloading the application code and Duo plugin hosted in AWS S3 bucket.
sudo systemctl start httpd
sudo systemctl enable httpd
sudo setsebool -P httpd_can_network_connect 1//Allow outbound connections from HTTPD daemon
sudo wget https://safelabfiles.s3.us-east-2.amazonaws.com/wordpresscodefile.zip -P /var/www/html/
sudo unzip /var/www/html/wordpresscodefile.zip
sudo wget https://downloads.wordpress.org/plugin/duo-wordpress.2.5.4.zip -P /var/www/html/wp-content/plugins
sudo unzip /var/www/html/wp-content/plugins/duo-wordpress.2.5.4.zip
sudo systemctl restart httpd

#Downloading the Tetration enforcement agent from AWS S3 bucket and installing it.
sudo wget https://safelabfiles.s3.us-east-2.amazonaws.com/config/tetration_installer_intgssopov_enforcer_linux.sh
sudo chmod 755 tetration_installer_intgssopov_enforcer_linux.sh
sudo ./tetration_installer_intgssopov_enforcer_linux.sh --skip-pre-check

#Downloading the AMP4E agent hosted in an AWS S3 bucket and installing it.
sudo wget https://safelabfiles.s3.us-east-2.amazonaws.com/config/cisco.gpg
sudo rpm --import ./cisco.gpg
sudo wget https://safelabfiles.s3.us-east-2.amazonaws.com/config/AWS_rhel-centos-7fireamplinux_connector.rpm
sudo yum install -y AWS_rhel-centos-7fireamplinux_connector.rpm
```

Step 3. **Set up the Auto Scaling Groups** - Create two Auto Scaling Groups using the Launch Configurations created in the previous step, one for the Web servers and another one for the Application servers. For more information on creation of Auto Scaling groups, follow the AWS documentation.

Once the Auto Scaling Groups are created, select each group and click on edit. In the edit menu, update the target groups and health check types. For Web Server Auto Scaling group, set the target group to 'WebServerPool' as created during the Web NLB set up. Also, update the health check type to **ELB** to integrate the Auto Scaling Group with Load Balancer. Repeat the same steps for the App Server Auto Scaling Group, use the target group 'AppServerPool' created during App NLB set up.

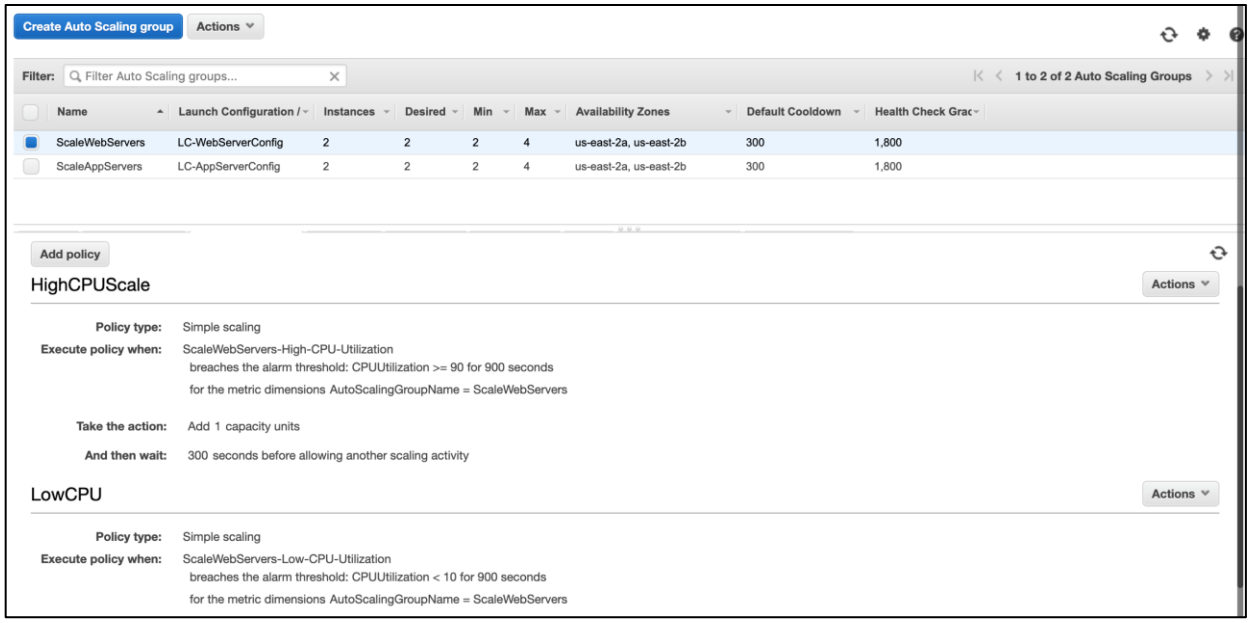




Step 4. **Configure the scaling policies** – On the Auto Scaling Groups page, select each group and click on Scaling policies tab to add the scaling policies. For testing purpose, we used Simple Scaling of adding or removing one instance when average CPU Utilization exceeds 90% or remains

below a minimum value of 10%. The desired state was set to two instances at a given point of time.

Click on **Add policy** and select **Create a simple scaling policy**. Fill in the policy name, alarm (you will need to create a new alarm) and action as per the requirement.



Follow the AWS documentation for more further details on scaling policies.

Setting up the Firepower Next-Generation Firewalls

In this section, we will set up a pair of Cisco Firepower Next-Generation Firewalls at the network perimeter and onboard them to Cisco Defense Orchestrator for management. As an alternate option, you could also use Cisco FMC (Firepower Management Center) available via the AWS marketplace (or an on-premise FMC) for management purposes.

Once the firewalls are set up, we will enable public access to the application via an 'Outside' Network Load Balancer.

Implementation procedure:

Step 1. Set up the AWS Environment for NGFWv

Step 2. Deploy NGFWv EC2 instances

Step 3. Onboard the NGFWv to CDO

Step 4. Configure interfaces, routes, NAT and access control on NGFWv

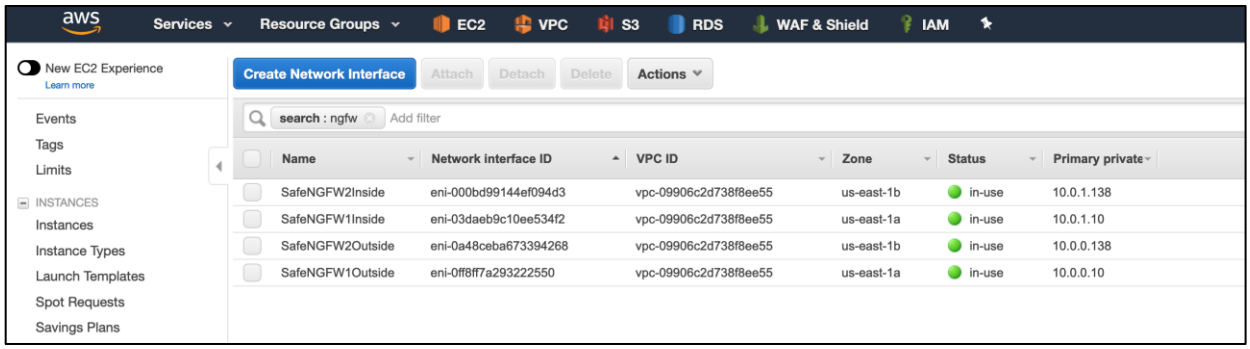
Step 5. Set up the Outside Network Load Balancer

Step 1. Set up the AWS Environment for NGFW- To deploy Firepower Threat Defense Virtual we need to set up the Network Interfaces for the appliance and allocate Elastic IPs to be assigned to the Management and Outside Interfaces. We use the IP addressing as defined in table below.

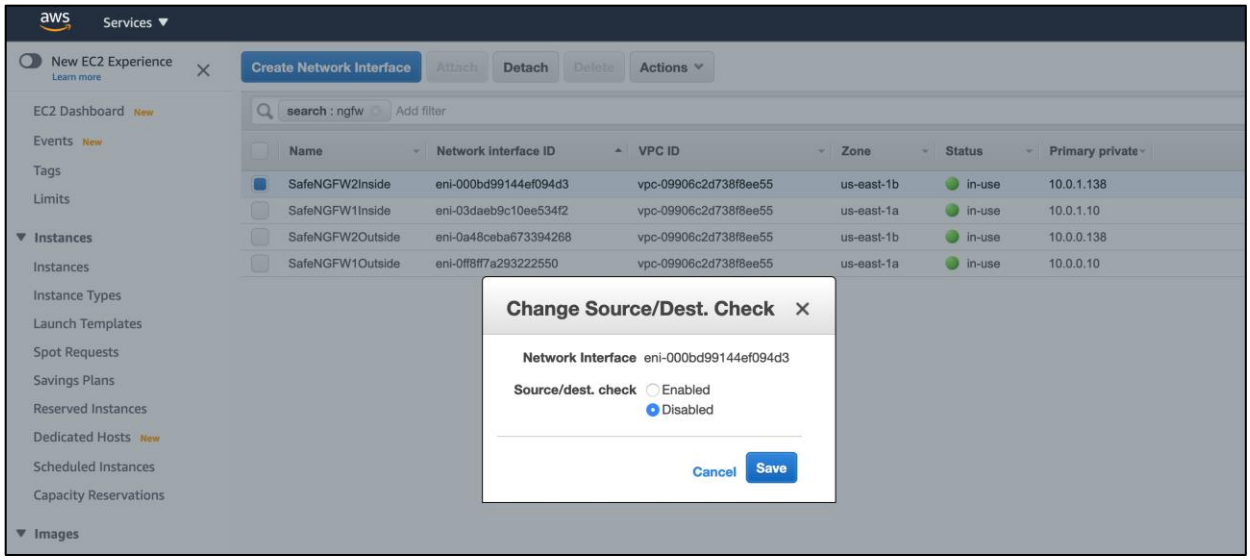
NGFWv	Interface Name	IPV4 Address	AWS NIC	Firepower Interface
Safengfw1	Management	10.0.8.200	NIC0	Management

NGFWv	Interface Name	IPV4 Address	AWS NIC	Firepower Interface
Safengfw1	Diagnostic	10.0.8.0/24 (DHCP)	NIC1	Diagnostic
Safengfw1	Outside	10.0.0.10	NIC2	Gig0/0
Safengfw1	Inside	10.0.1.10	NIC3	Gig0/1
Safengfw2	Management	10.0.9.200	NIC0	Management
Safengfw2	Diagnostic	10.0.9.0/24 (DHCP)	NIC1	Diagnostic
Safengfw2	Outside	10.0.0.138	NIC2	Gig0/0
Safengfw2	Inside	10.0.1.138	NIC3	Gig0/1

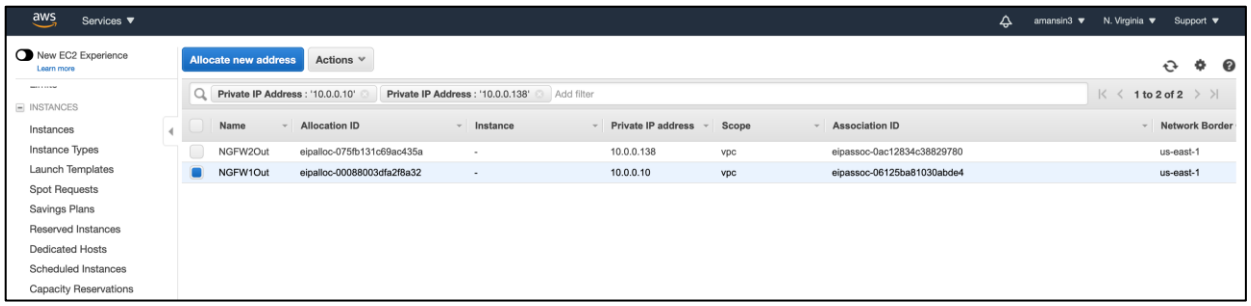
Navigate to **EC2 Dashboard > Network Interfaces** and **Create Network Interfaces** for inside and outside interfaces of each FTDv appliance.



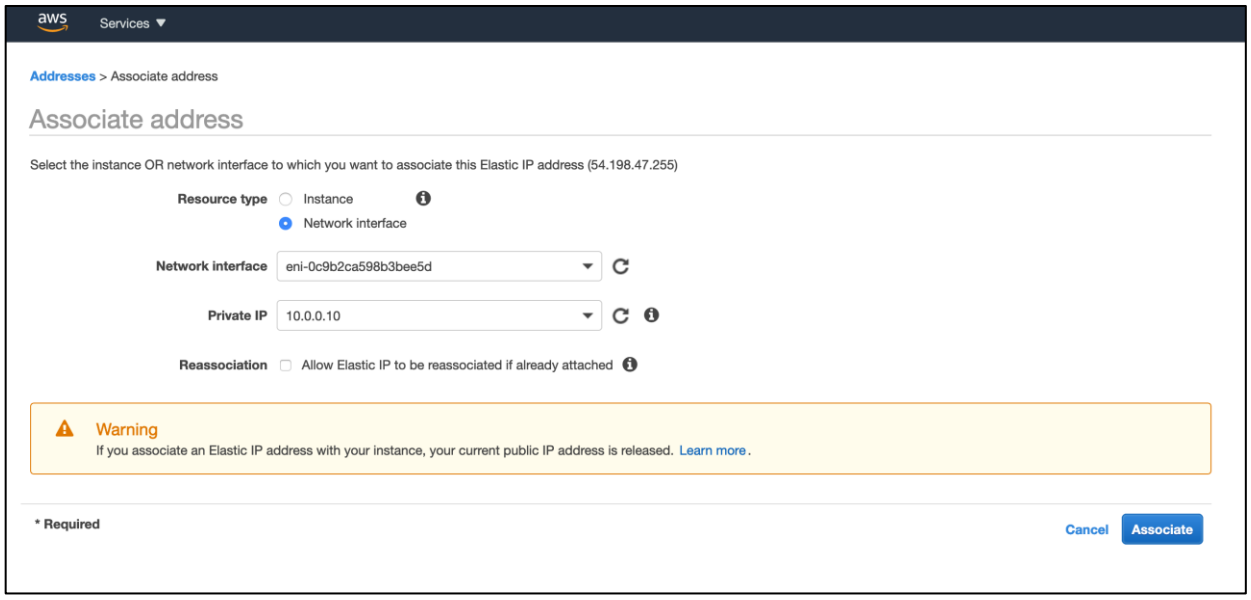
Select the network interfaces that were created and right click on Change Source/Dest. Check to disable it.



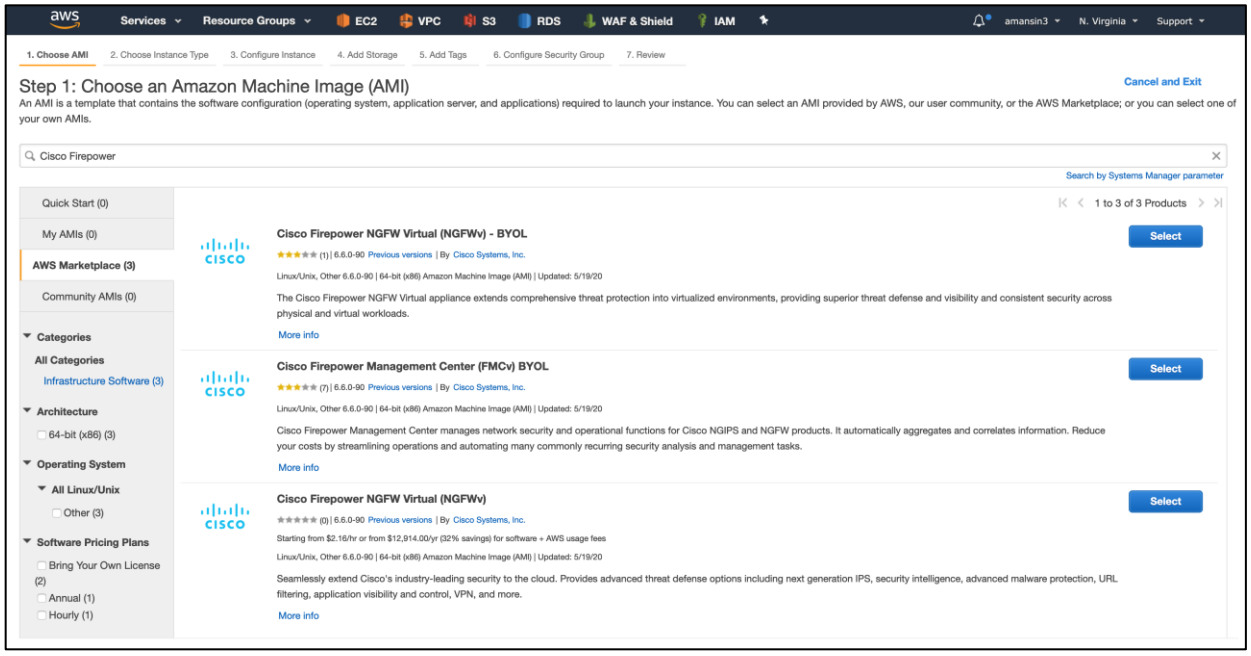
Navigate to EC2 Dashboard > Elastic IPs and click on Allocate New Address to allocate two elastic IPs.



Step 2. Select each of the newly assigned Elastic IP addresses and associate them with Outside Network interfaces created in previous step.



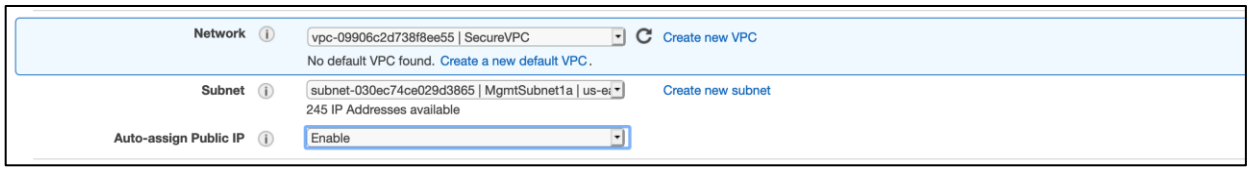
Step 2. **Deploy NGFWv EC2 instances**– Navigate to EC2 AWS console and click on launch and choose an AMI for Cisco Firepower NGFW virtual appliance. Choose the Instance Type.



Click **Next** to configure the Instance:

Change the **Network** to match your previously created VPC.

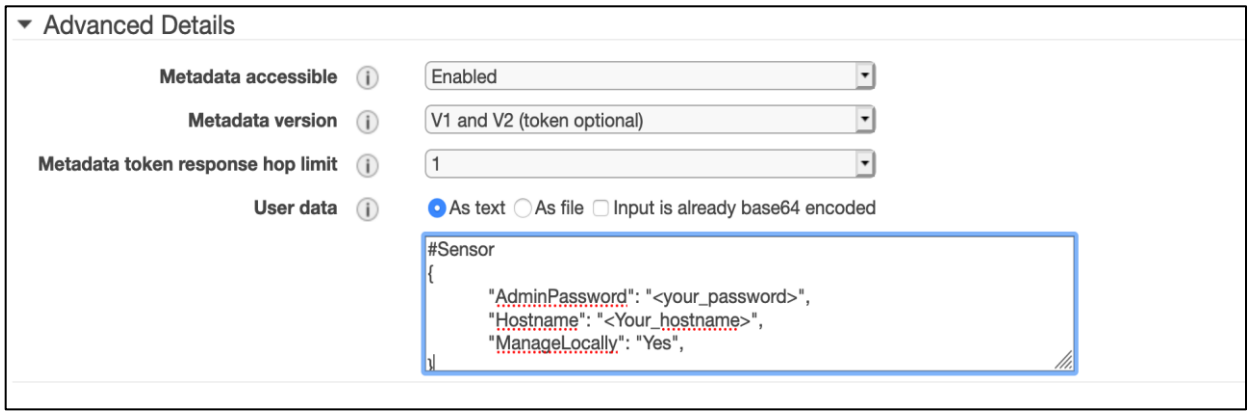
Change the **Subnet** to match your previously created management subnet. You can specify an IP address or use auto-generate.



Click the **Add Device** button under **Network interfaces** to add the **eth1** network interface.

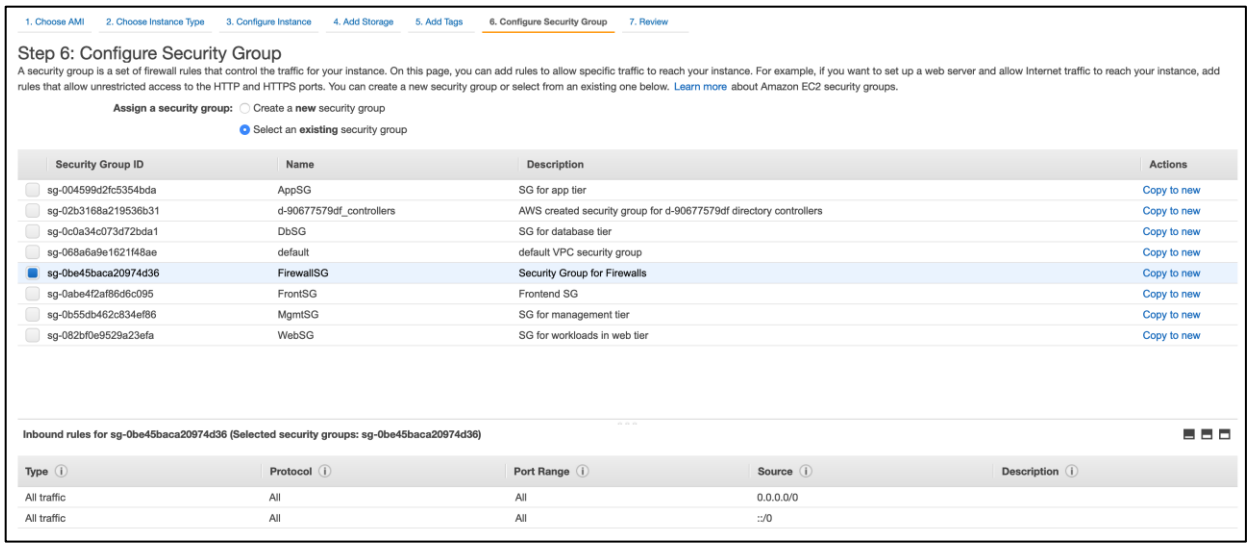
Change the **Subnet** to match your previously created management subnet that is used for eth0.

Under **Advanced Details**, add the default login information as **User data**.

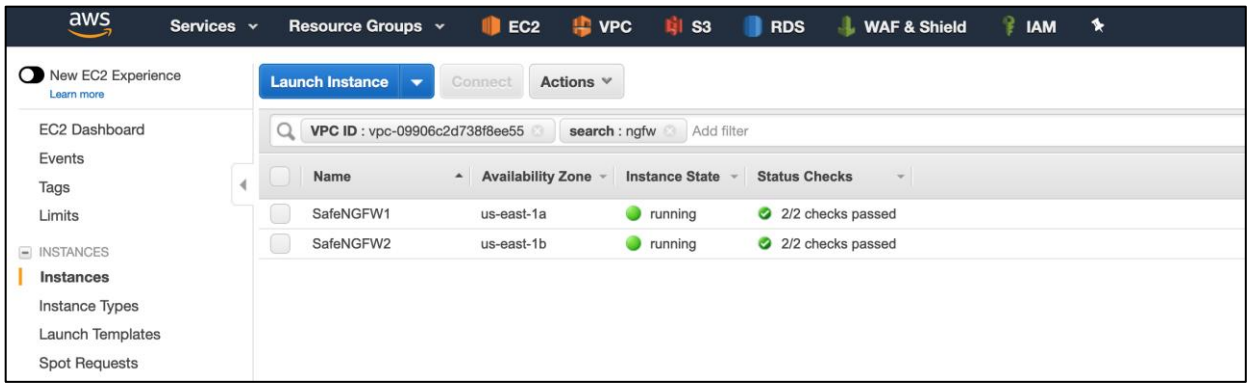


Click **Next** to **Add Storage**. You can accept the default or change the volume. Click **Next** again to add a **Tag**, this step is optional as well.

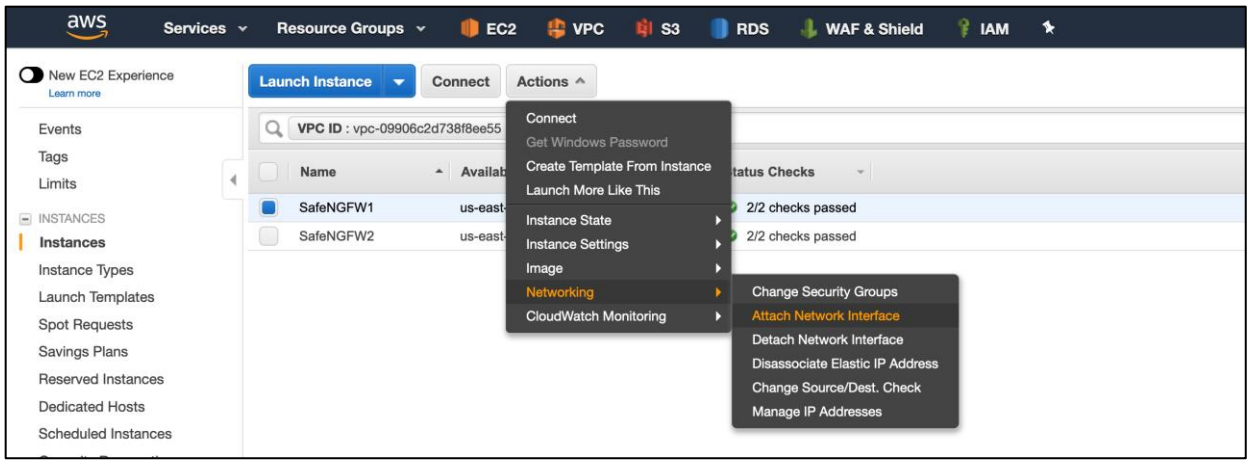
Lastly, Select **Next** again to **Configure Security Group**. Click **Select an existing Security Group** and choose the previously configured Firewall Security Group.

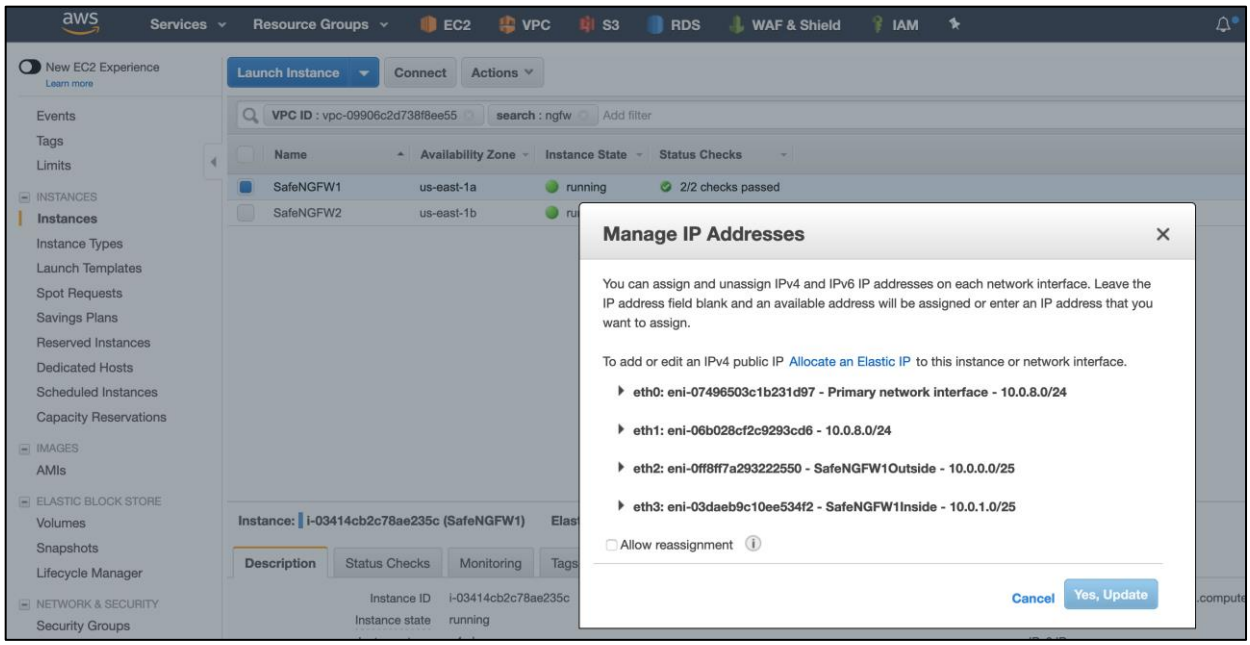


Click **Review and Launch**. Repeat the same steps to launch the second FTDv appliance, make sure you select management subnets corresponding to second Availability Zone.

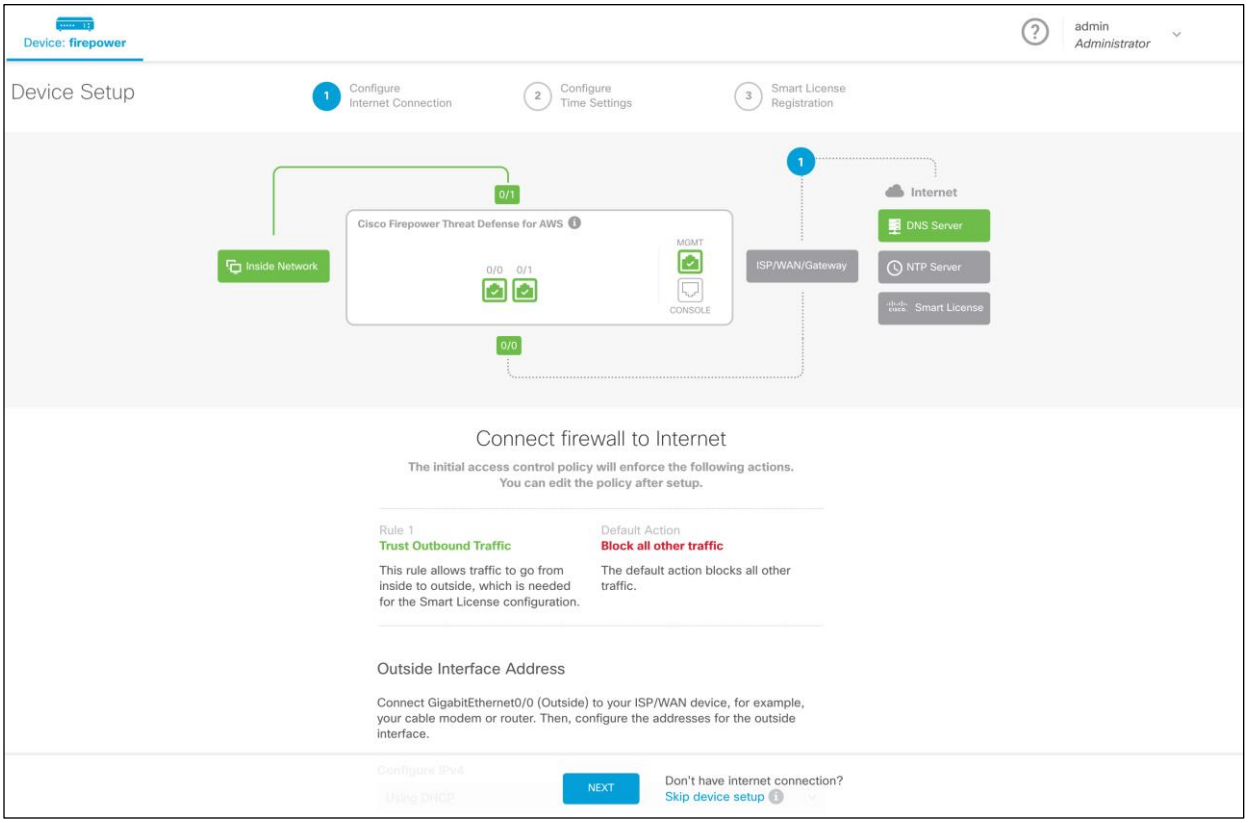


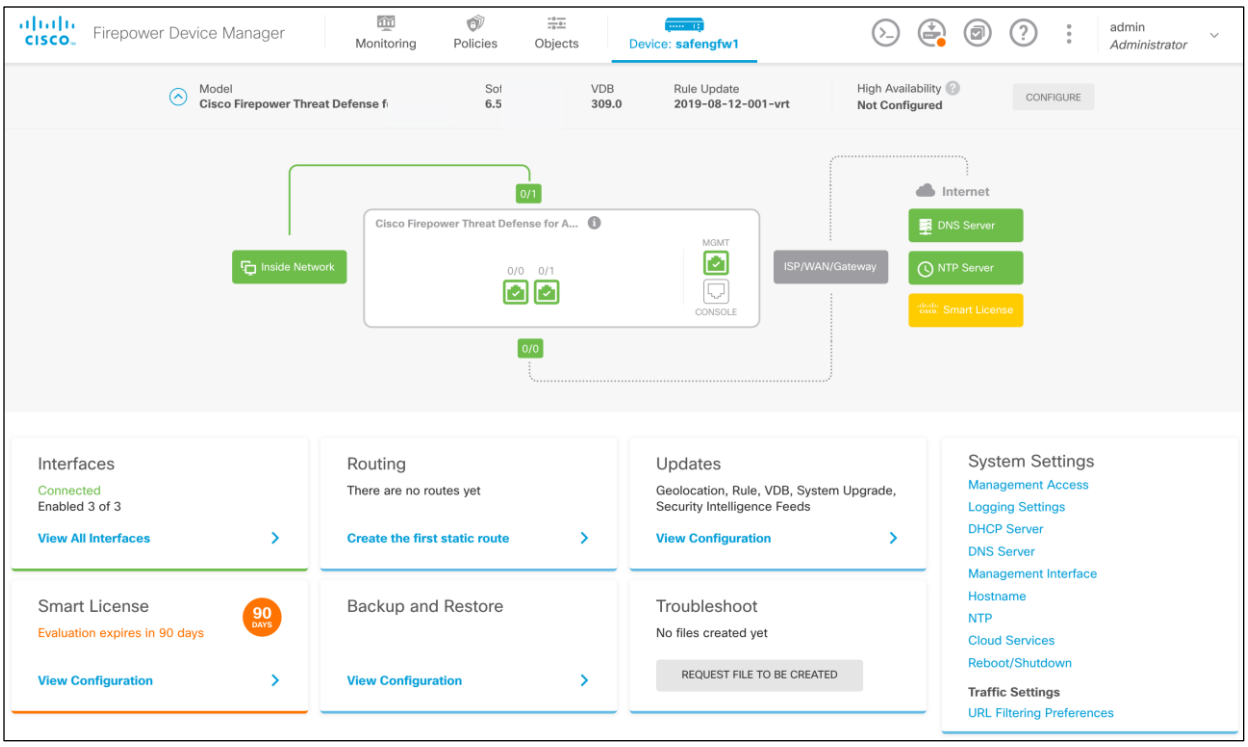
Select the newly configured FTDv appliance and click **Actions**, select **Networking > Attach Network Interface** to attach the outside and inside Network Interfaces created in Step 1.



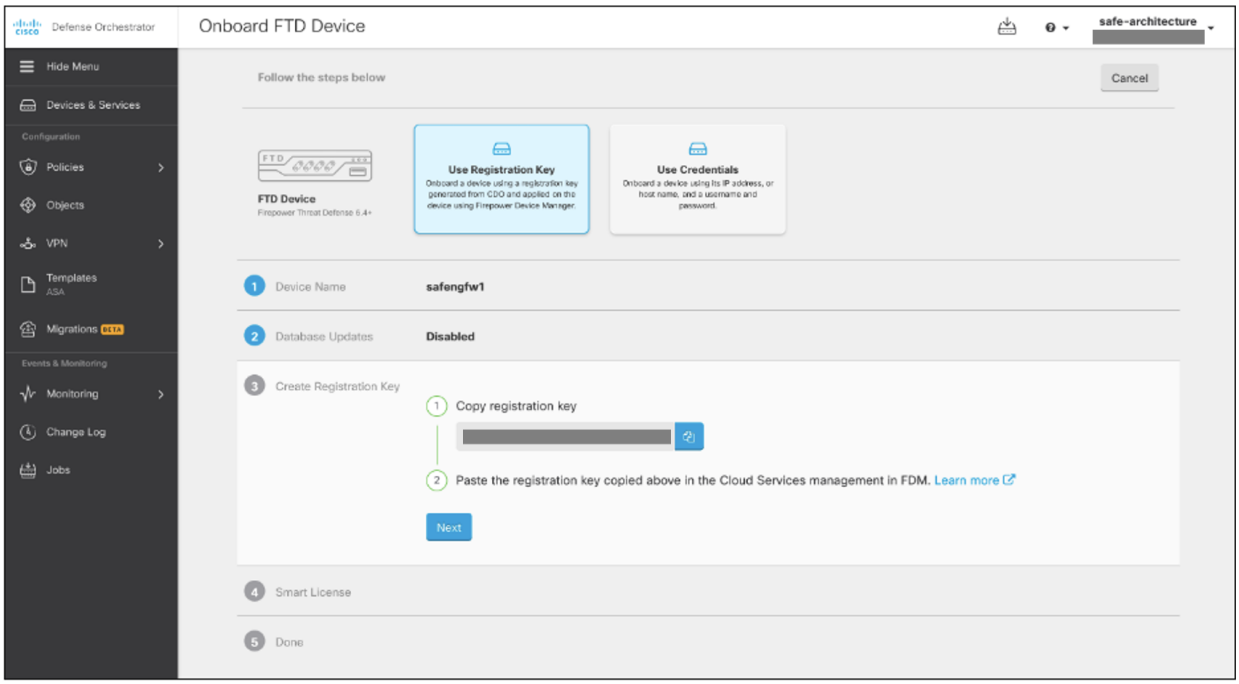


Step 3. **Onboard the NGFWv to CDO** – Access the Firepower Device Manager (FDM) using the management IP address. Click on **skip device set up** and acknowledge the 90-day trial license warning (we will configure smart licensing in subsequent steps). You will land at the FDM home page.

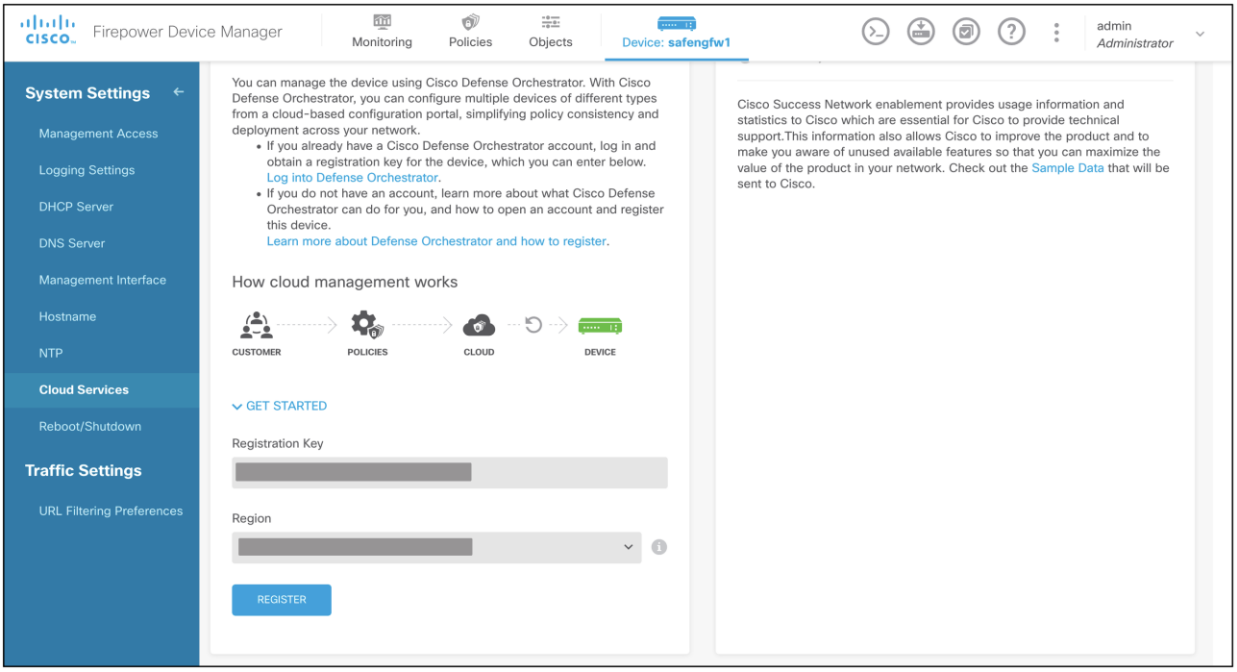




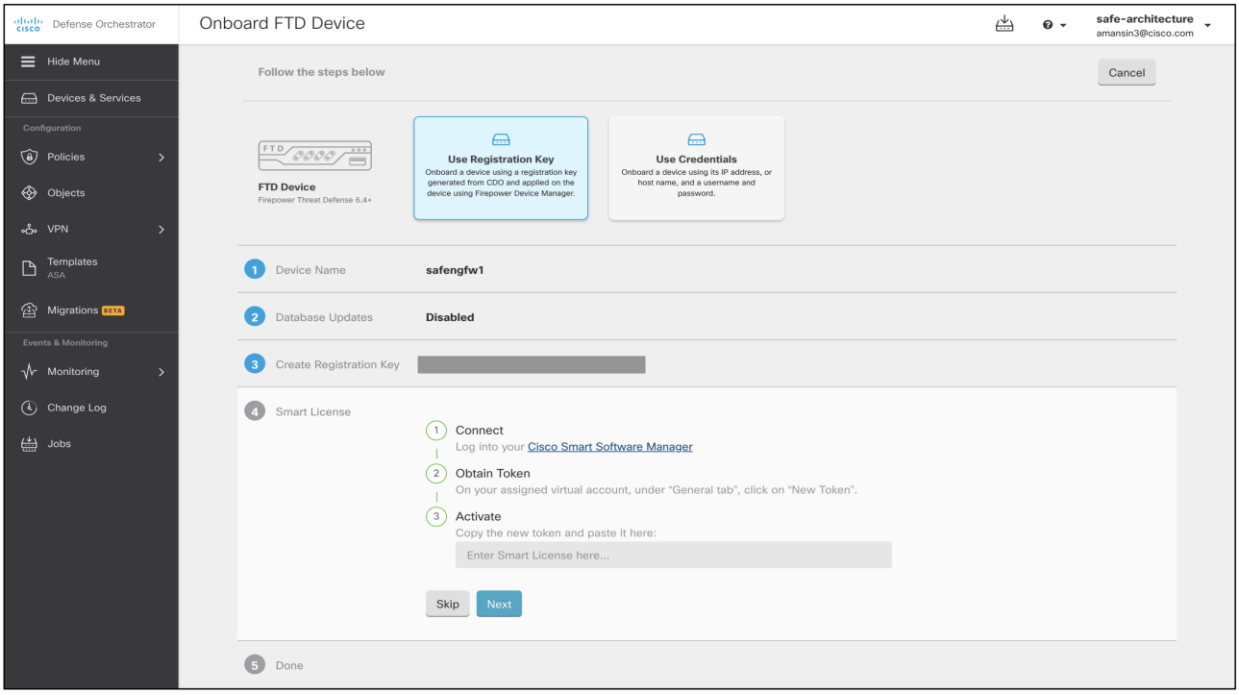
Log on to the CDO web portal and go to **Devices and Services** and click on plus button on the top right-hand side to onboard a Firepower Threat Defense (FTD) device. We use **Registration Key** option to onboard the FTD. Fill in the name of the FTD device and follow the wizard to copy the registration key.



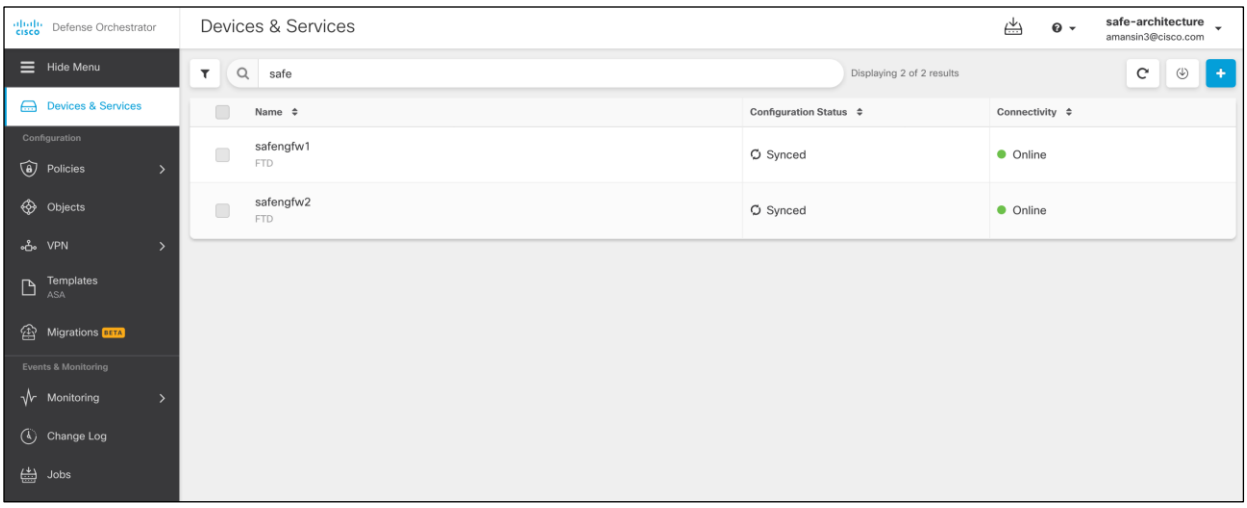
Go back to FDM portal and go to **Cloud Services** option under **System Settings**. Paste the **Registration Key**, specify the **Region** and click on **Register**.



Come back to the CDO portal, click on **Next** and log into your Cisco smart licensing manager and generate the token. Paste the token and click on **Next** to finish the onboarding process.



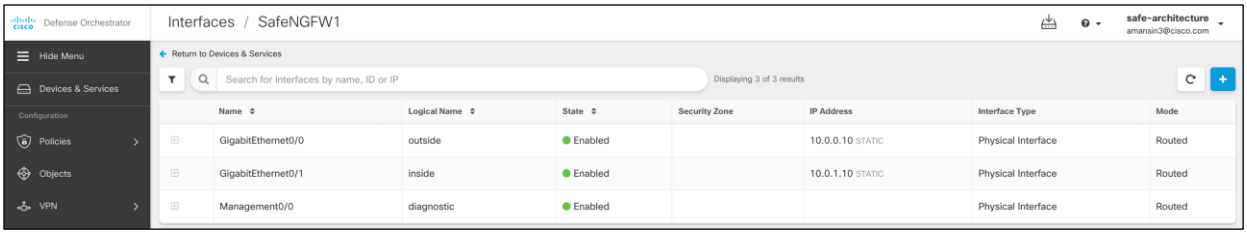
Repeat the same steps for the other firepower device, CDO will sync the configuration for both the devices. At this point we have successfully finished onboarding both the FTDs to CDO.



Step 4. Configure interfaces, routes, HTTPS health probes, NAT and access control on NGFWv – We need to configure the following four components:

- Network interfaces
- Static routes
- NAT rules
- Access rules

Click on the onboarded NGFWv appliance on CDO dashboard and then go to **Interfaces** option from the menu that appears on the right-hand side of the dashboard. Make sure Gig0/0 and Gig0/1 are assigned static IP addresses and names as defined in the table in Step 1.



Go back to the same menu and go to Routing option now, set the default route pointing to the gateway on outside the subnet- 10.0.0.1. Also, add the route for internal subnets (web, app and database subnet) pointing to the gateway on the inside subnet- 10.0.1.1. Lastly, add a route for AWS Metadata Server IP address for health probes, set the next hop as the inside subnet gateway.

Next, we set up three NAT rules. First is an optional dynamic PAT rule to allow outbound traffic to the Internet from the application workloads. You would need this rule if you decide to forward any outbound flows to the internet via NGFWv appliance. The translation would be as below.

```
Source: WorkloadIP:Port => OutsideFWInterfaceIP:Port
```

Second, a static NAT rule to expose the AWS application to the users on the internet accessing the application. We need the source translation to ensure that reply traffic is returned back to the same firewall's inside interface i.e. maintain the traffic symmetry. The destination is translated from outside interface IP of the FTD appliance in the specific availability zone to the static IP address of the Web NLB in same zone.

```
Source translation: InternetUserIP:Port => InsideFWInterfaceIP:Port
Destination translation: OutsideFTDIP:HTTP => WebLBIP:HTTP
```

In a similar manner, we pick a random health check port for FTD (example – TCP port 6612) and set up third NAT translation rule to forward the health check traffic to AWS metadata server. When Outside NLB sends TCP health probes on port 6612 to outside interface of FTD, the FTD would translate the source to Inside interface IP and destination to AWS metadata server on port 80 and forward the traffic.

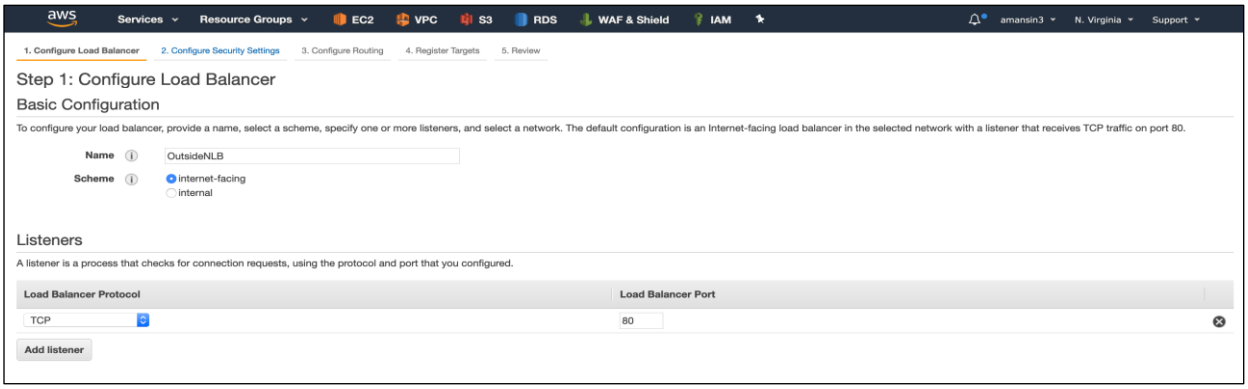
```
Source translation: InternetUserIP:Port => InsideFWInterfaceIP:Port
Destination translation: OutsideFTDIP:HealthCheckPort => AWSMetaDataServer:HTTP
```

Lastly, we configure access control policies to allow traffic from inside zone to outside zone. We also allow HTTP traffic incoming from the Internet users and health probes from outside NLB.

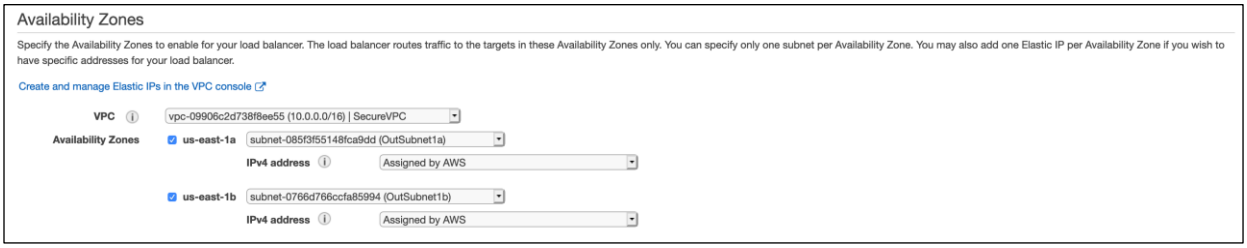
Go to Policies > FTD Rulesets on CDO portal and click on plus button to add a an FTD ruleset. Within this newly created ruleset, add the access rules and attach it to the newly onboarded FTD appliances.

Repeat the same set up for the second FTD device and then deploy all the changes.

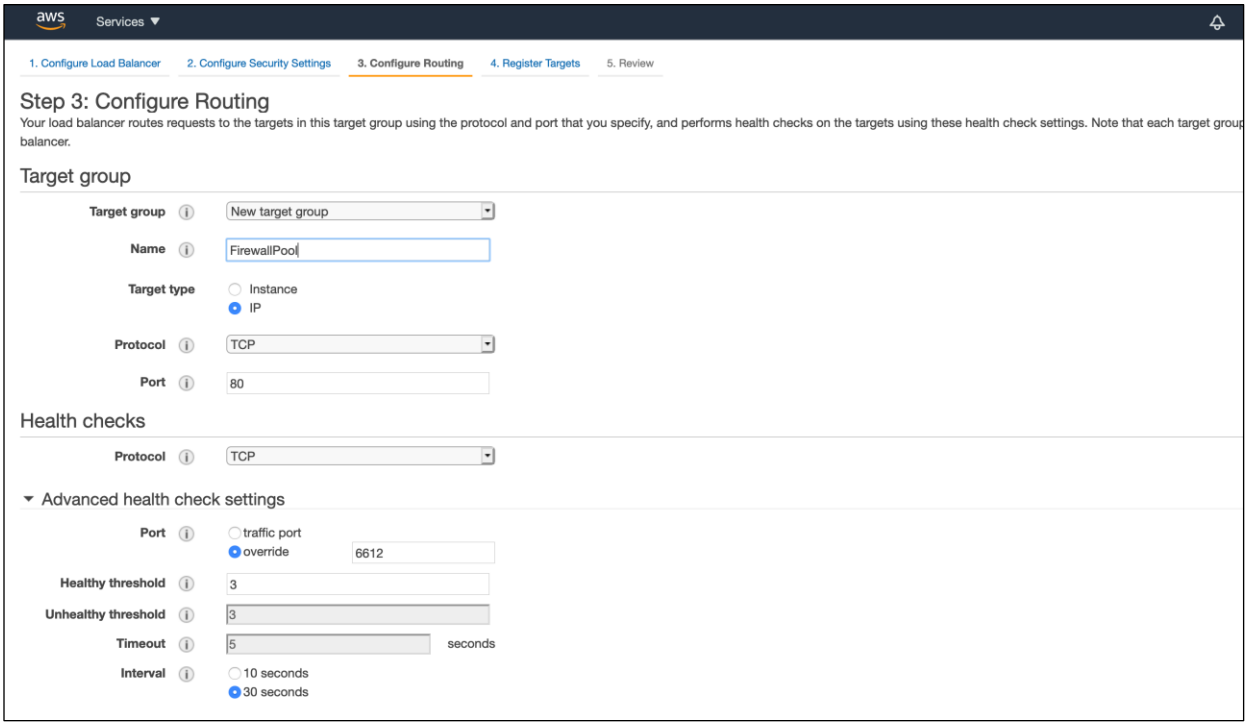
Step 5. **Set up the outside NLB –** In this step, we will set up the outside NLB with the Target Group for outside interfaces of the two FTDv appliances.



Select the two availability zones and corresponding Outside Subnets that we set up previously.

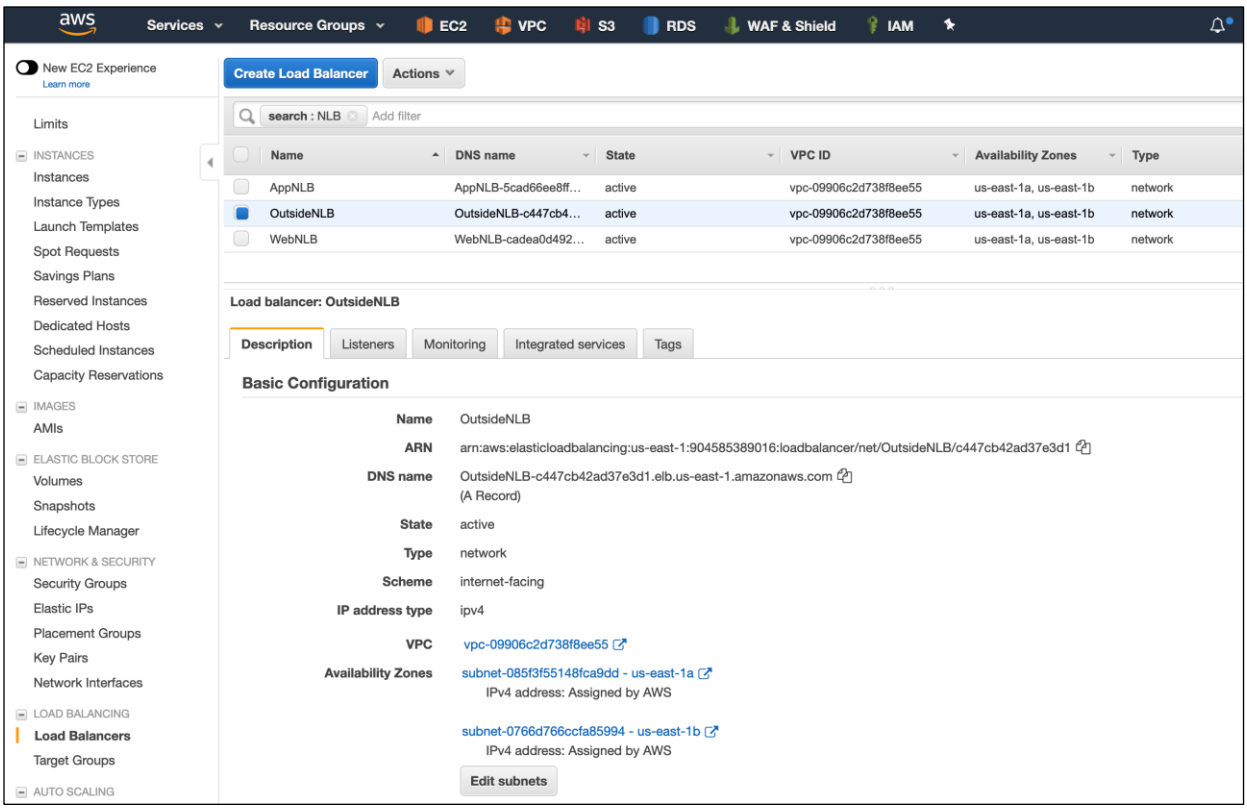


At Step 3, specify a **Name** for the **Target group** for FTD appliances. Select the **Target type** as IP and **Protocol** as TCP on Port 80. For **Health checks**, override the **Port** to 6612 (we had previously set up the FTD to redirect health probes on port 6612 to AWS metadata server). Click **Next** after making all the changes.

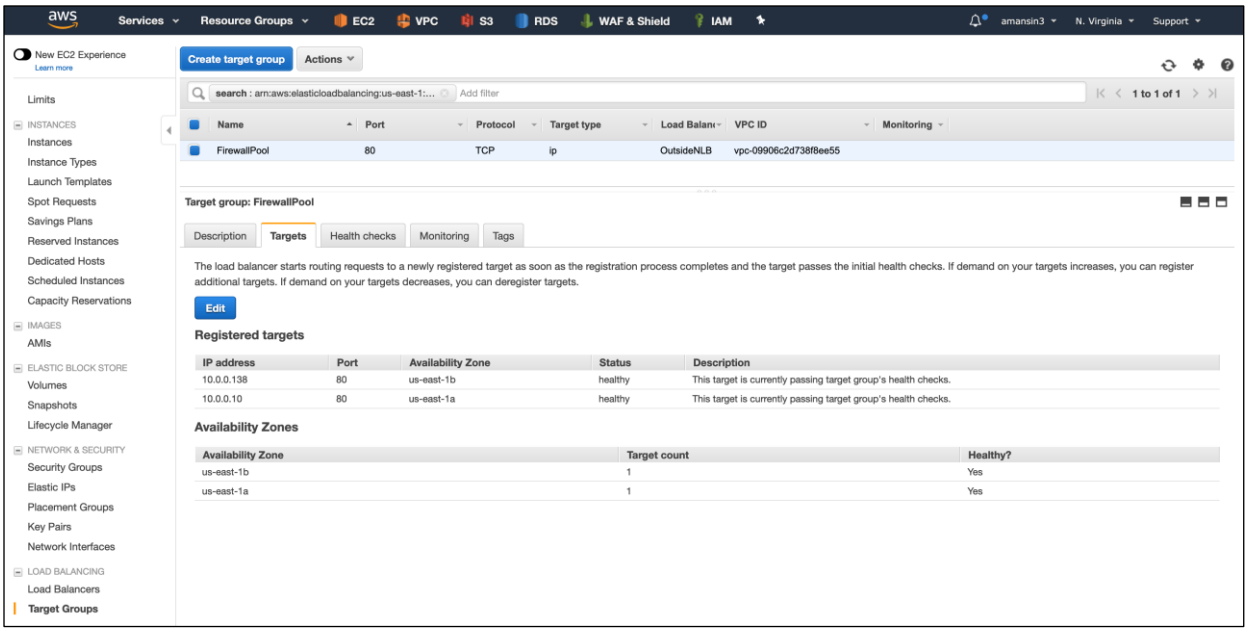


For target registration, add the Outside interface IP addresses.

Click **Next** and **Review and submit** the changes to finish Outside NLB creation.



After few mins, both the FTDs should register as targets and health status should move to healthy.



Enabling WAF and DDoS protection

At this point we have finished setting up a fully functional cloud application. We will now add the WAF and DDOS protection capabilities in our design.

Radware Cloud WAF and DDoS services are integrated by using CNAME DNS records. Deployment is independent of any specific Cloud service providers.

Implementation procedure:

Step 1. Onboard the application to Radware Cloud

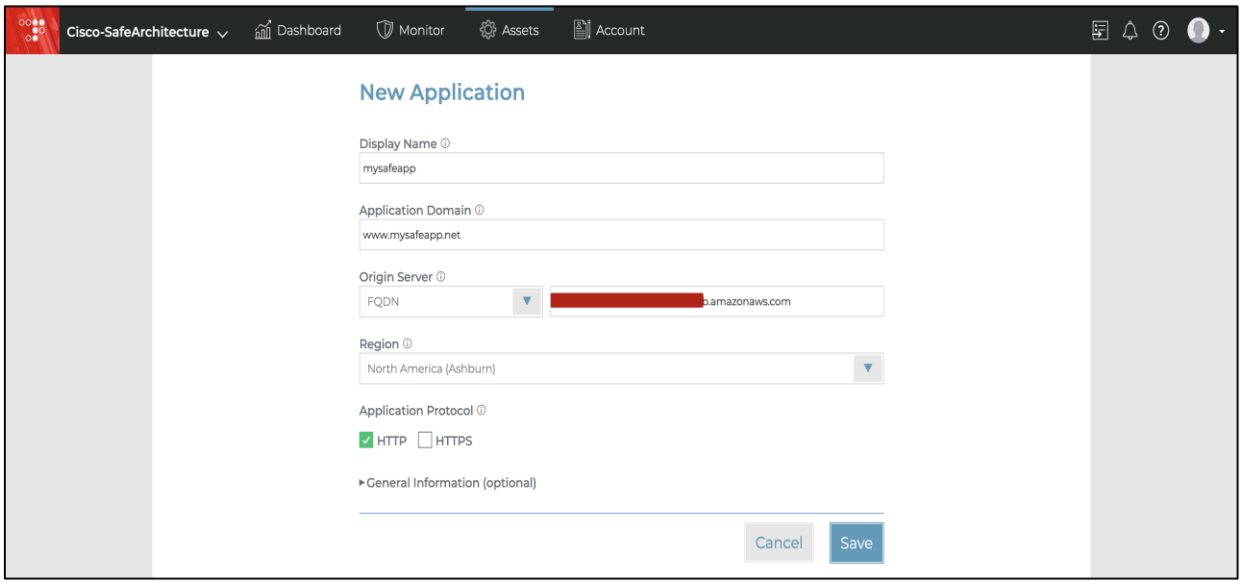
Step 2. Register a domain using AWS Route 53 service

Step 3. Update the DNS setting to point traffic to Radware Cloud

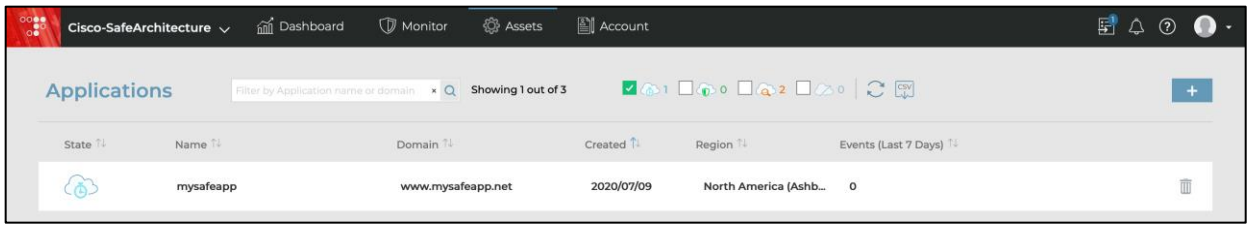
Step 4. Access the application

Step 1. **Onboard the application to Radware Cloud** - The first step to integrating Radware cloud service is to onboard the application onto the Radware cloud. On the Radware cloud portal, go to **Assets > Application** and click on the plus button on the upper right-hand side of the screen. Add the prompted details i.e. the **Application Domain Name** (www.mysafeapp.net), the **Origin Server** (in this case it would be the Outside network load balancer's FQDN) and the **Protocol** (HTTP). If your application is based on HTTPS protocol, you would need to add the certificate information as well.

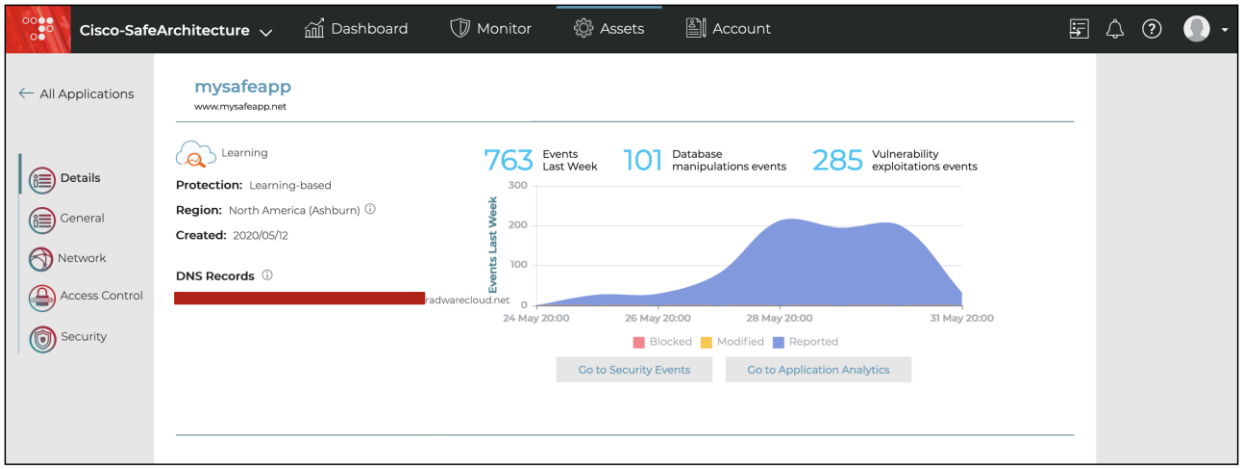
Note: As part of the onboarding process, each customer can choose between immediate and learning based protection. Immediate protection will enforce a predefined security policy, preventing known attacks. In order to cover both known and unknown attacks, Radware recommends using the learning-based protection method. During the first 2 weeks (duration can vary depending on traffic), Radware evaluates traffic patterns and can automatically update both negative and positive security models by refining signatures, creating exceptions and building the allowed file extension list per application, greatly reducing false positives.



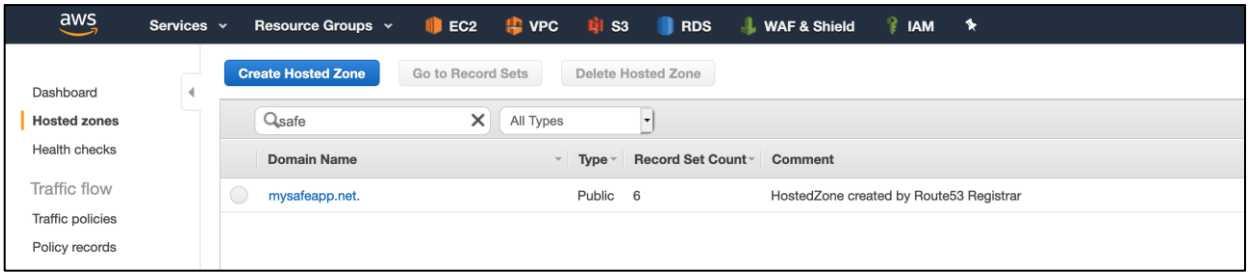
Once the details are saved, the application can be seen as below.



Click on the application and copy the allocated **CNAME**. We need to create a DNS record for our application with this Radware CNAME.

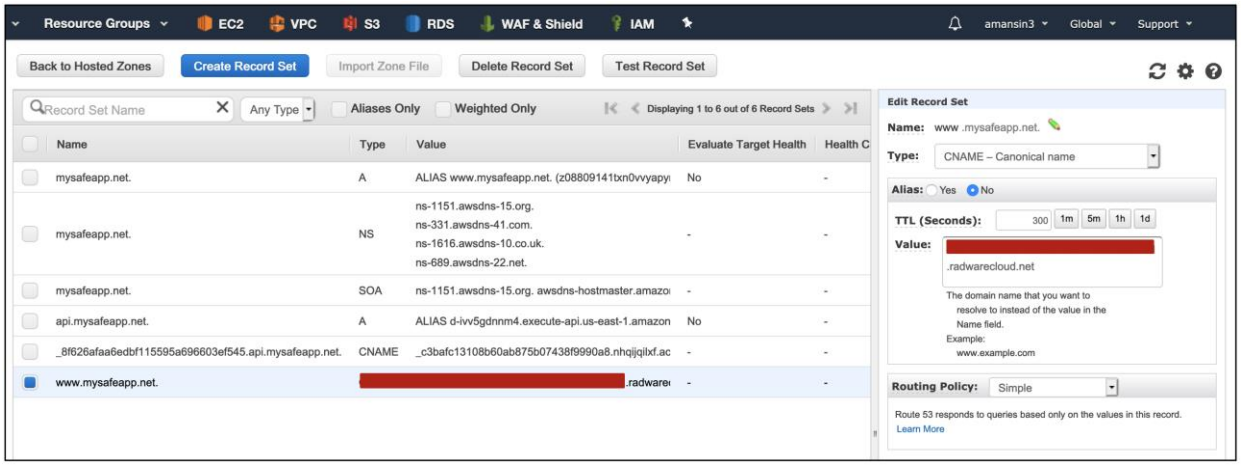


Step 2. **Register a domain using AWS Route 53 service –** On the **Route 53** dashboard on AWS console, click on **Create Hosted Zone** to register a domain for the application. The registration process might take anywhere from few minutes to few hours to complete.



Step 3. **Update the DNS setting to point traffic to Radware Cloud** – Once the domain registration is completed, go back to **Route 53** hosted domain and update the DNS record sets with Radware CNAME. After this change, it might take a few minutes for the DNS update to propagate. Once the DNS records are fully updated, the traffic will start getting redirected to Radware Cloud servers before it hits the 'origin server' in the AWS cloud.

Note: To eliminate direct origin attacks, Radware recommends configuring the perimeter firewall to only allow the Cloud WAF to access the application origin server directly. The service IP addresses can be requested from Radware Support Team. For more information, check out the [Radware Cloud WAF Quick Start guide](#) (login required).



Step 4. **Access the web application –** Go to this newly registered domain URL in the browser, you will be prompted to do the initial application setup, after the initial set up the application home page should load as below.

Integration with Cisco SecureX

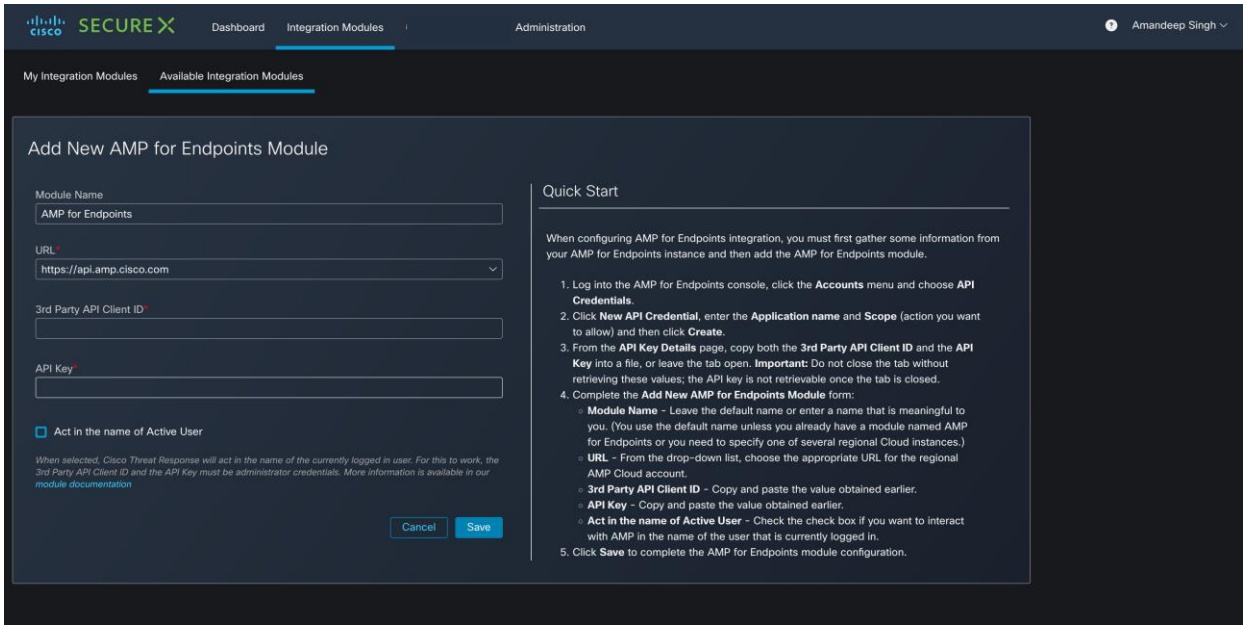
In this last deployment step, we will enable the Umbrella, Stealthwatch Cloud and Tetration and AMP4E modules in the SecureX portal to get a unified view into the AWS environment. We create API keys in the product portals and then configure those keys in the threat response dashboard.

Implementation procedure:

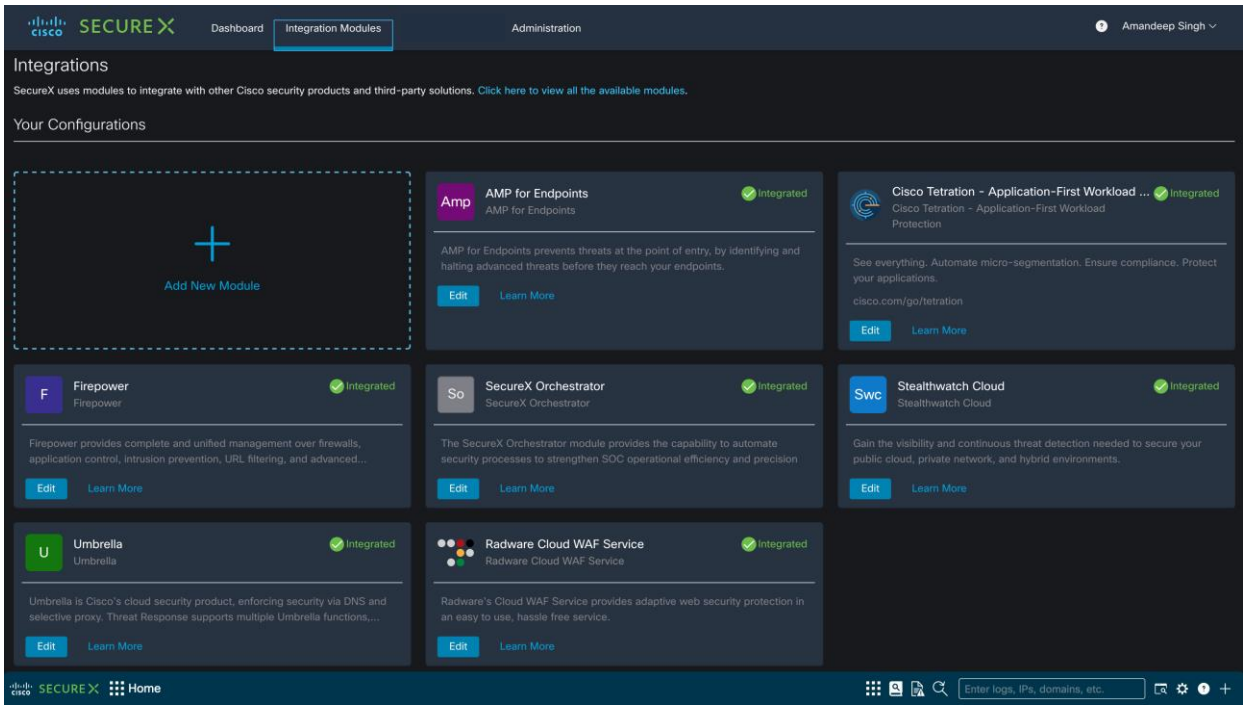
Step 1. **Add Integration modules**

Step 2. **Save the module**

Step 1. **Add Integration modules –** Log on to the SecureX dashboard and go to **Integration modules** tab, click on **Add a New Module** and select from the available modules. SecureX dashboard displays all the steps on API key generation and integration for each available module.



Step 2. **Save the module –** Click on **Save** to complete the integration. In a similar manner add all the remaining modules. After we have saved the module configurations, the modules will be listed under **Integration Modules** tab as below.



Validation Testing

Tetration

Validation procedure overview:

- Test Case 1 – Creating the workspace for AWS cloud application
- Test Case 2 – Using ADM to discover the policies for AWS workloads and setting up an app view

- Test Case 3 - Enforcing the policies on workloads
- Test Case 4 - Discovering the vulnerable packages on the AWS workloads

Test Case 1: Creating an application workspace for AWS cloud application

This test case involves defining annotations for the AWS environment. These annotated attributes are used later to segregate the tiers and segments within the AWS VPC and hence define a workspace for our tiered cloud application.

Validation procedure:

Step 1. Build an inventory

Step 2. Define scopes

Step 3. Create a workspace

Step 1. Build an inventory – Define the attributes that would help you segregate your tiered application workloads in the cloud and hence construct policies for them. We will use a combination of two different methods to add user annotations - 1) Upload a CSV file 2) Auto generate annotations using external AWS orchestration.

1.1: Based on the architecture of our tiered application (elaborated in the previous sections of this document), the following annotations were used (Table: AWS Cloud Inventory). Save this in a CSV file format.

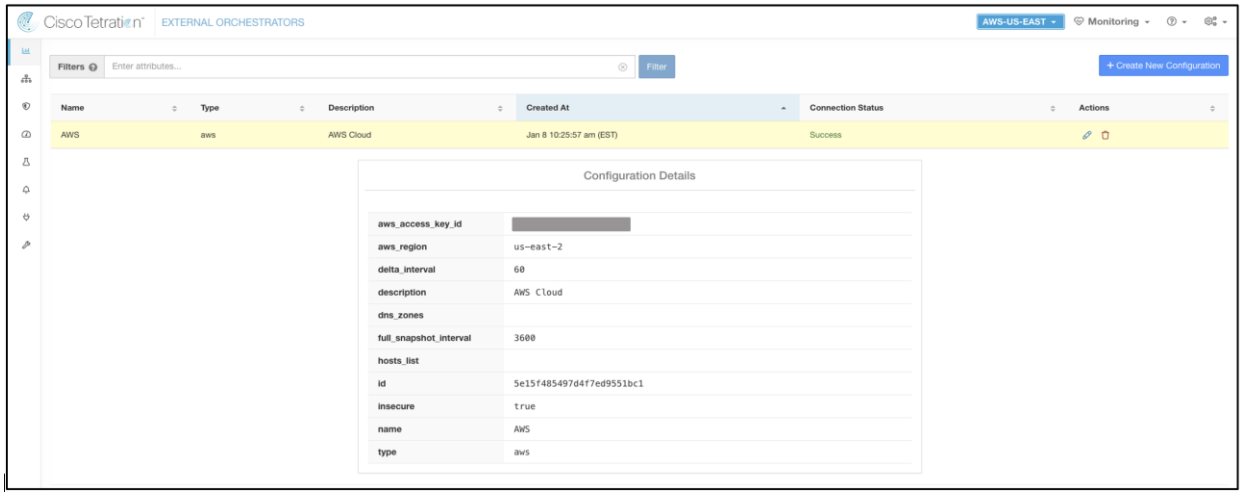
Table 1. AWS Cloud Inventory

IP	Application	Region	Tier	Type
10.0.2.0/24	Safe3tierApp	US-East-2	WebServer	AWS-Cloud
10.0.3.0/24	Safe3tierApp	US-East-2	WebServer	AWS-Cloud
10.0.4.0/24	Safe3tierApp	US-East-2	AppServer	AWS-Cloud
10.0.5.0/24	Safe3tierApp	US-East-2	AppServer	AWS-Cloud
10.0.6.0/24	Safe3tierApp	US-East-2	Database	AWS-Cloud
10.0.7.0/24	Safe3tierApp	US-East-2	Database	AWS-Cloud
10.0.8.0/24	Safe3tierApp	US-East-2	Management	AWS-Cloud
10.0.9.0/24	Safe3tierApp	US-East-2	Management	AWS-Cloud

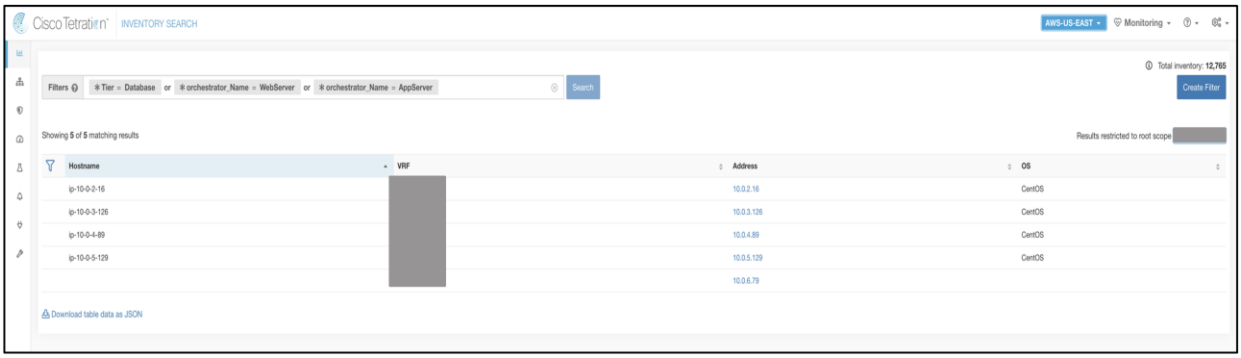
1.2: Now, log into the Tetration cloud portal and go to ‘Visibility > Inventory Upload’. Click on ‘Select File’ and ‘add’ the CSV file.

1.3: Go to ‘Visibility > External Orchestrator’. Click on ‘Create New Configuration’ and fill in the required details as shown below.

Note: - You will need to create an Access Key in your AWS account to be used in this configuration. Follow AWS documentation for more details on access key creation.

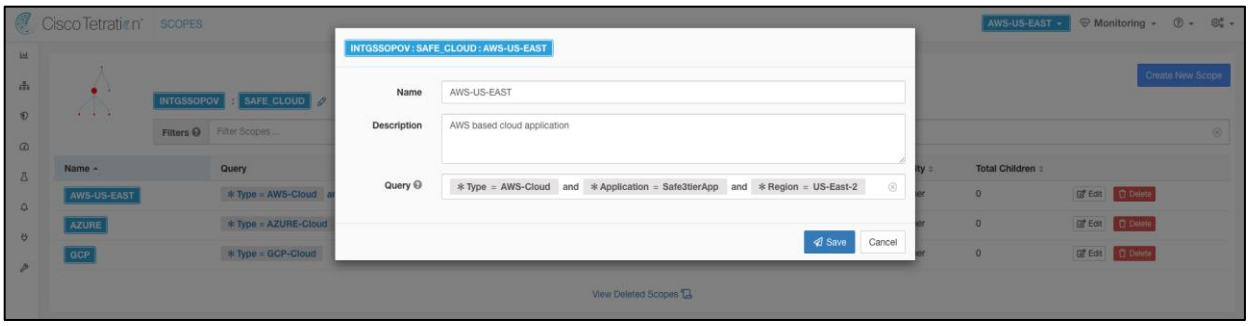


1.4: After a few minutes, you can go to ‘Visibility > Inventory Search’ and test the filters generated, based on annotations from the Step 1.2 and 1.3 above.



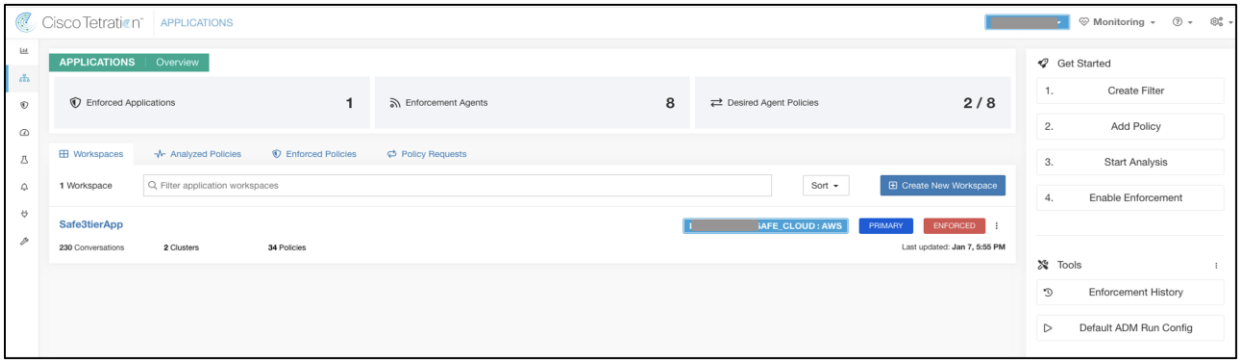
Step 2. **Define scopes –** We will define a scope to group together all the workloads in our tiered application in the AWS cloud. We will make use of the annotations/filters that we constructed in Step 1. We created the scope ‘AWS-US-EAST’, which includes all the workloads from our tiered app in ‘US-East’ region in AWS Cloud.

Click on the settings icon in the top right corner of the portal and then go to ‘Scopes’ option. Click on ‘Create New Scope’ and fill in the name of the Scope and a query as below.



Step 3. **Create a workspace –** Application workspaces are the containers for defining, analyzing and enforcing policies for a particular application. We will create a workspace for our tiered AWS cloud application in this step.

Click on ‘Segmentation’ and then click on ‘Create New Workspace’. Give the workspace a name and select the Scope that we created in Step 2.



At this point, we have successfully built the inventory, created a scope and defined a workspace for our tiered cloud application.

Test Case 2: Using ADM to discover the policies for AWS workloads and setting up an app view

This test case validates the use of 'ADM' to automatically discover the policies based on flow and other data received from workloads. We will refine the discovered workload clusters and update the inventory filters to eventually come up with a set of policies that can be enforced on our cloud workloads.

Validation procedure:

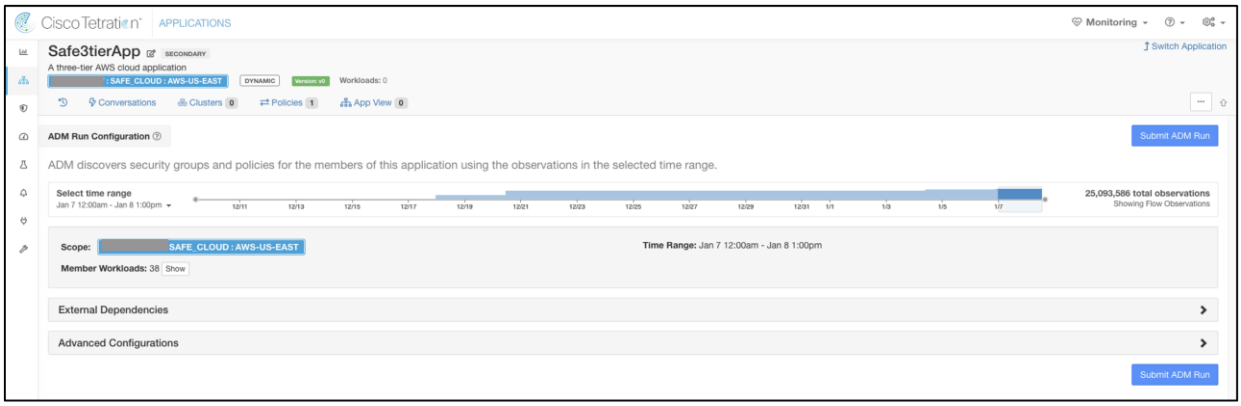
Step 1. Discover policies using ADM

Step 2. Refine inventory filters, clusters and policies

Step 3. Create the App View

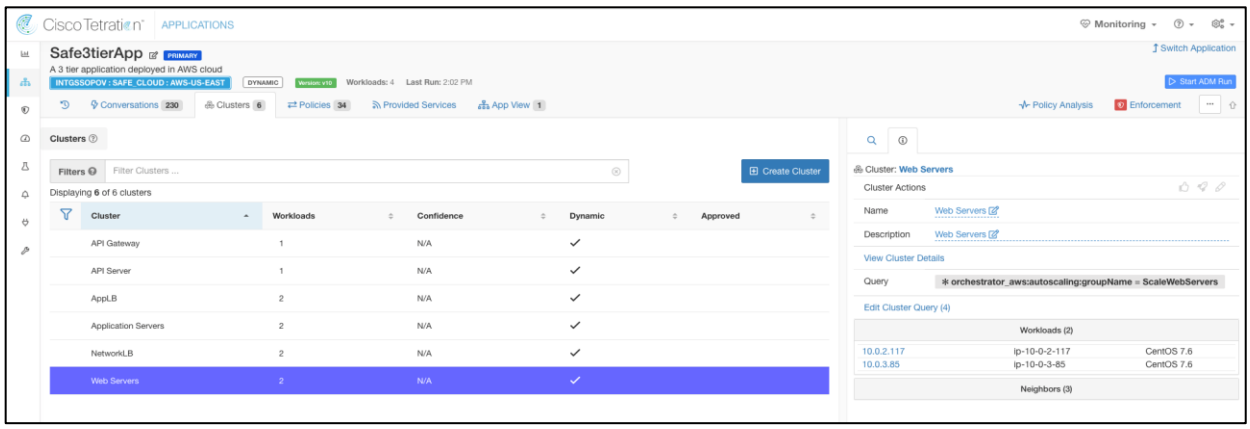
Step 1. Discover policies using ADM – Before running the ADM, ensure that all types of traffic flows are generated in the application environment. This would provide ADM the required data to generate an accurate policy set and hence ensure that we don't miss any critical but less common traffic flows.

Go to the newly created workspace and click on 'Start ADM Run' on the top right corner, select a suitable time range to ensure that you cover all the traffic flows.



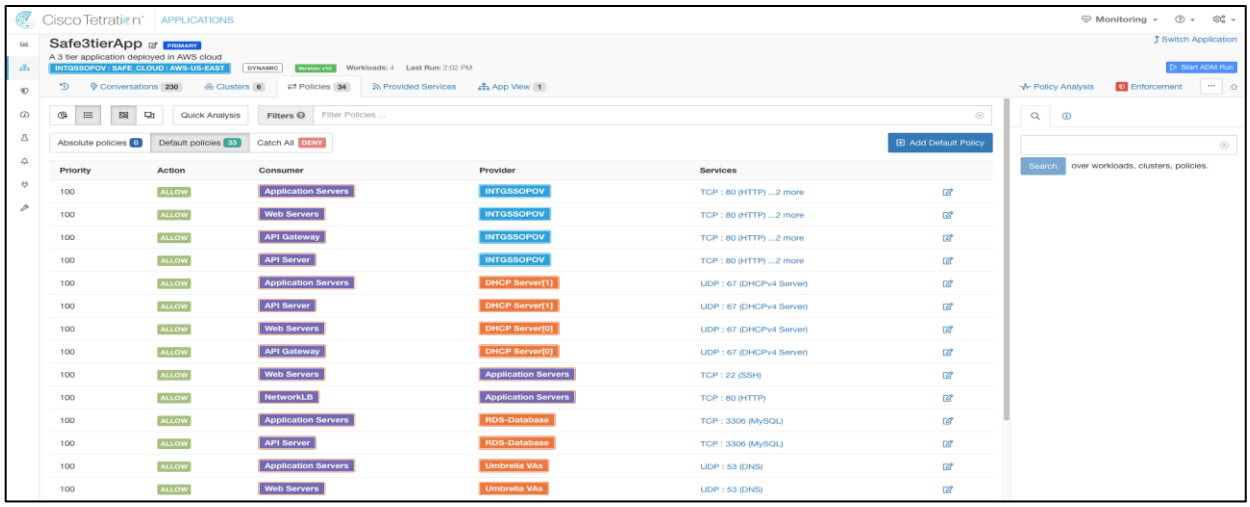
Step 2. Refine inventory filters, clusters and policies – Post the ADM run, policies and clusters would be generated. At this point, we manually update and customize all the cluster queries and approve them.

2.1: Go to 'Clusters' tab, click on any of the clusters, the panel on the right-hand side will show the cluster details like name, description, cluster query, workloads, neighbors. Update name and description to make it more intuitive, update the cluster query if need be. For example, we updated the cluster query for auto scaled workloads. We used previously defined annotation to dynamically identify the workloads in Auto Scaling groups for the web and application servers.

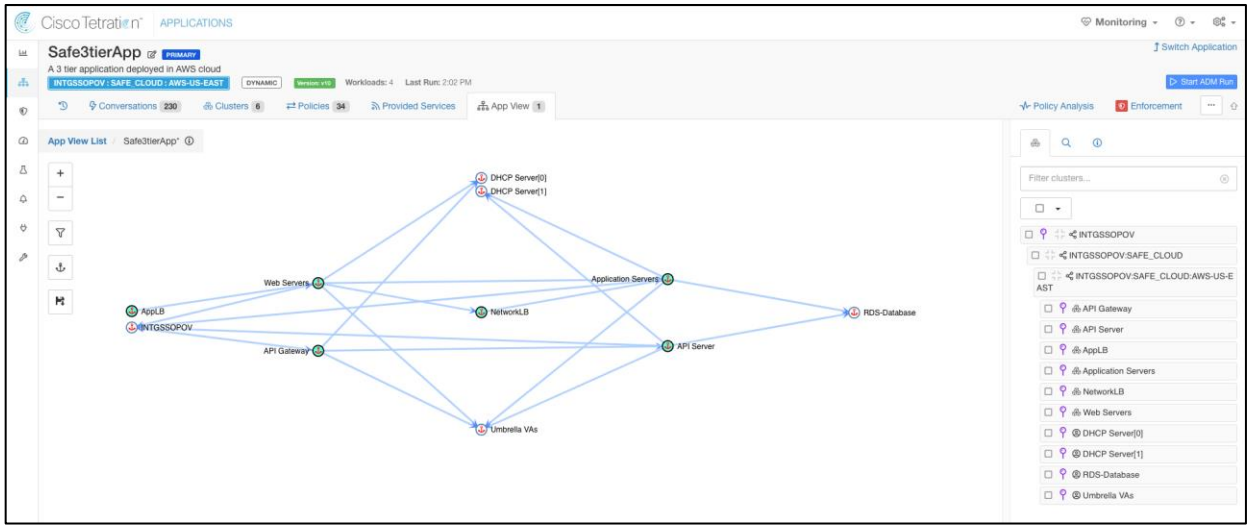


2.2: Click on ‘Policies’ tab, review the policies keeping the workload flows in mind. We considered the following flows for policies:

- User requests incoming to Web Servers via Application Load Balancer
- Traffic between the workloads
  - Web Servers to Network Load Balancer
  - Network Load Balancer to App Servers
  - App Servers to RDS Database instance
- Management tier to all the workloads
- Outbound internet access from all the workloads for updates/patches, DNS, DHCP, NTP



Step 3. **Create the App View –** Go to 'App view' tab and click on 'Create New App View'. Pin the workloads on the right-hand side panel to include them in your diagram. Double click on each pinned cluster on the view to automatically draw the traffic flows.



Test Case 3: Enforcing the policies on workloads.

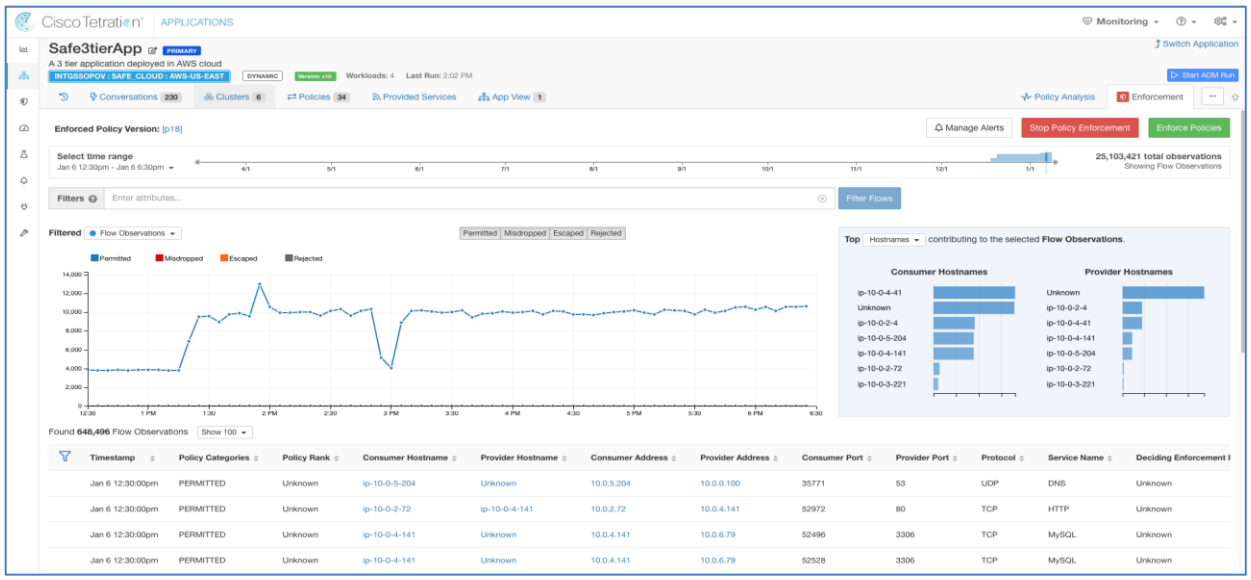
This test case focuses on enforcing the policy set that we formulated in Test Case 2. We will publish the policies and verify if those are enforced as expected.

Validation procedure:

Step 1. Publish the policies

Step 2. Verify policy enforcement on workloads

Step 1. **Publish the policies –** Select the 'Enforcement' tab on the Tetration portal within the application workspace and click on 'Enforce Policies'.



Step 2. **Verify policy enforcement on workloads** – Since we had CentOS based workloads, we monitored the '/usr/local/tet/log/tet-enforcer.log' to see if policies are successfully enforced. A simple ping or telnet test can also be used to verify the lockdown of ports and protocols.

```
I0107 04:25:25.275367   468 agent_controller.cpp:426] IPC is ready, start writing the message to IPC
I0107 04:25:25.275394   468 agent_controller.cpp:445] Done writing to shared memory
I0107 04:25:25.275499   471 agent_enforcer.cpp:813] Message from AgentController available, start processing

I0107 04:26:25.216424   468 ssl_client.cpp:410] Received message body length: 12


I0107 04:26:30.277519   468 ssl_client.cpp:510] Calling callback fn to process msg
I0107 04:26:30.277545   468 agent_controller.cpp:179] Start message processing from EFE
I0107 04:26:30.277563   468 agent_controller.cpp:222] Write the protobuf to AgentEnforcer
I0107 04:26:30.277696   472 agent_enforcer.cpp:966] Process EFE Message
I0107 04:27:05.342911   473 agent_enforcer.cpp:1283] Received Policy config version: 1545811665
I0107 04:27:05.344177   473 agent_enforcer.cpp:1392] Processing network policy config from EFE, version: 1545811665
I0107 04:27:05.344187   473 agent_enforcer.cpp:1396] Storing the policy and enforcing
I0107 04:27:05.344703   473 firewall_context.cpp:140] Policy has been validated, applying the policy
I0107 04:27:05.344712   473 firewall_context.cpp:169] Applying all firewall rules to the system firewall
I0107 04:27:05.714491   473 iptables_context.cpp:529] Staged rules have been committed
I0107 04:27:05.726775   473 agent_enforcer.cpp:1403] Policy config has been applied successfully, current version: 1545811665, highest version: 1545811665
```

Use the CLI command 'ipset list' to view the ipset firewall settings enforced by Tetration agent on the CentOS workloads.

```
Name: ta_520a2f879b4e9ab37a6e620e928b
Type: hash:net
Revision: 6
Header: family inet hashsize 1024 maxelem 65536
Size in memory: 504
References: 2
Number of entries: 2
Members:
10.0.3.196
10.0.2.57

Name: ta_58225b8365be52f7b49f8254d96e
Type: hash:net
Revision: 6
Header: family inet hashsize 1024 maxelem 65536
Size in memory: 440
References: 30
Number of entries: 1
Members:
10.0.2.16

Name: ta_9c21cc33b4bedbed655487444413
Type: hash:net
Revision: 6
Header: family inet hashsize 1024 maxelem 65536
Size in memory: 504
References: 6
Number of entries: 2
Members:
10.0.5.129
10.0.4.89

Name: ta_bdfa4e36735d84cc7cc983454a94
Type: hash:net
Revision: 6
Header: family inet hashsize 1024 maxelem 65536
Size in memory: 504
References: 8
Number of entries: 2
Members:
10.0.9.253
10.0.8.179
```

Test Case 4: Discovering the vulnerable packages on the AWS workloads.

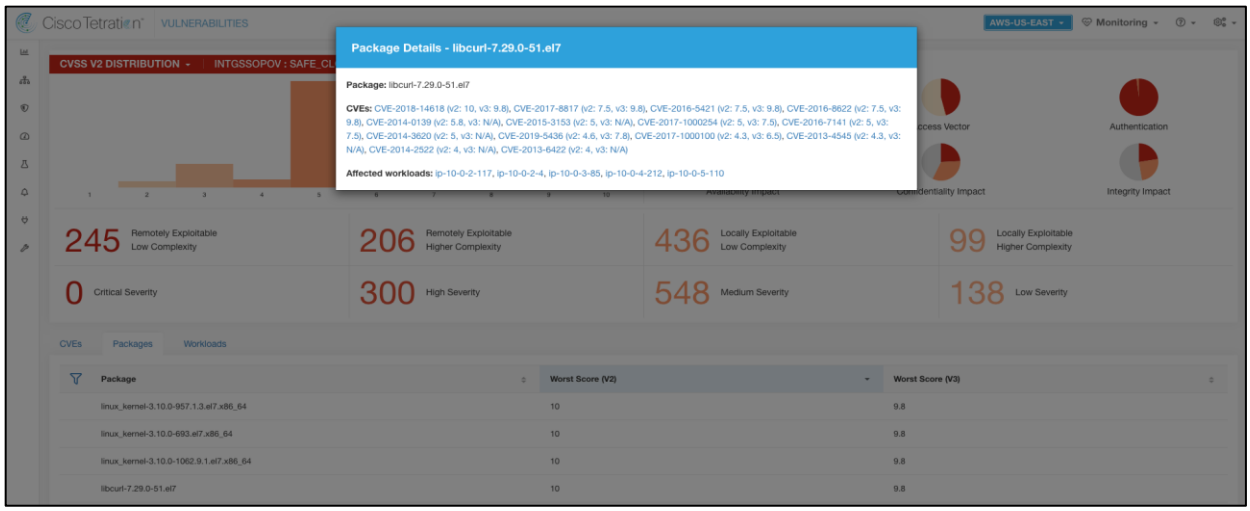
This test case looks for vulnerable packages/software installed on various workloads in the AWS. We identify a vulnerable package/software on our workloads, patch those and then rerun the report.

Validation procedure:

Step 1. Check the vulnerability report

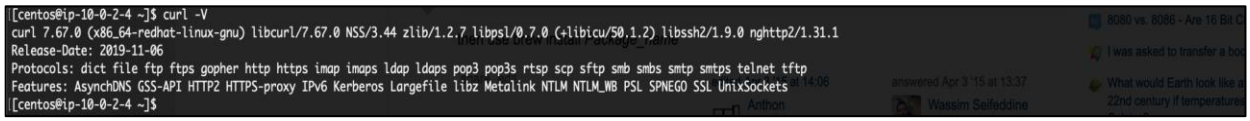
Step 2. Fix a vulnerability and rerun the report

Step 1. Check the vulnerability report – Go to 'Security > Vulnerabilities', click on 'Packages' tab to see all the vulnerable packages installed on various workloads in our three-tier application. For the sake of this test, let's consider 'libcurl-7.29.0-51.e17' as shown below.

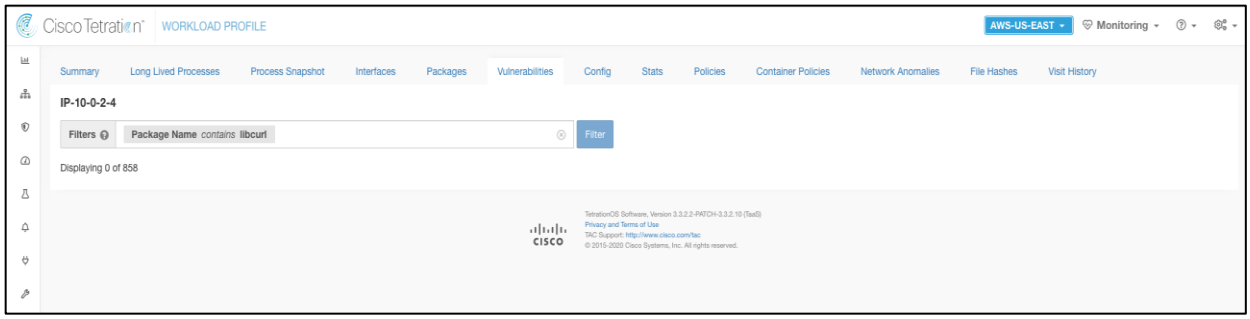


We see that the workload 'ip-10-0-2-4' is affected by this CVE. Logon to this workload and verify the libcurl package.

Step 2. **Fix the vulnerability and rerun the report –** We update the libcurl package from this workload to the latest version which has the fix to the CVEs listed in step.



Wait for a few minutes after the uninstall, go back to Tetration portal and check the vulnerability report again. We can see that none of the CVEs related to libcurl show up anymore.



Advanced Malware Protection for Endpoints

Test Case: Quarantine a suspicious file

This test case involves the detection of using AMP for endpoint 'simple custom detections' to quarantine a suspicious PDF file.

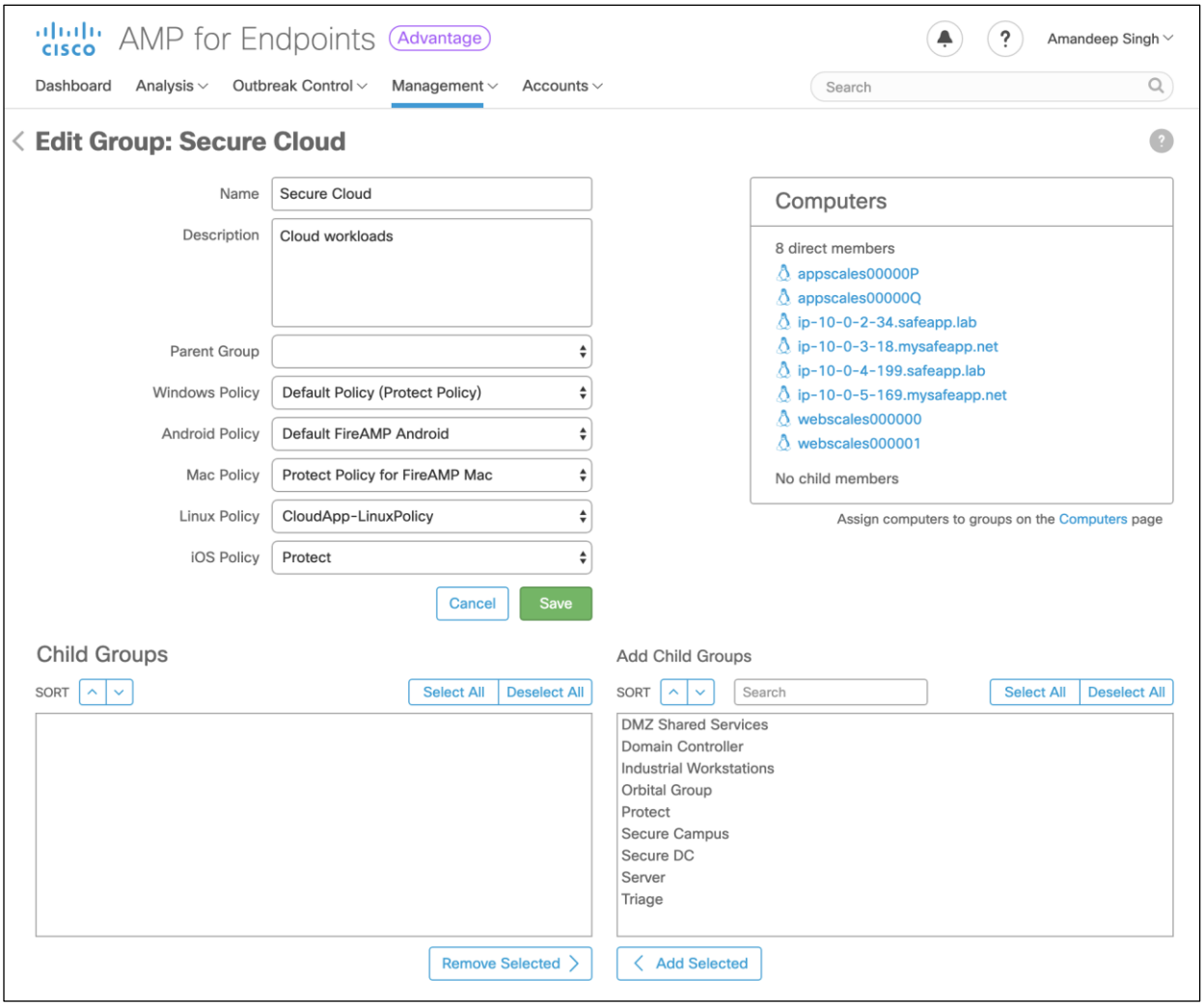
Validation procedure:

Step 1. Setting up AMP4E policy to quarantine a suspicious file

Step 2. Verifying the deletion of a suspicious file

Step 3. **Setting up AMP4E policy to quarantine a suspicious file –** For the validation purpose, we consider a 1 MB PDF file that we will block list using AMP 'Simple Custom Detections'. We will then try to download the same PDF file on a cloud workload and assert that our policy works as expected.

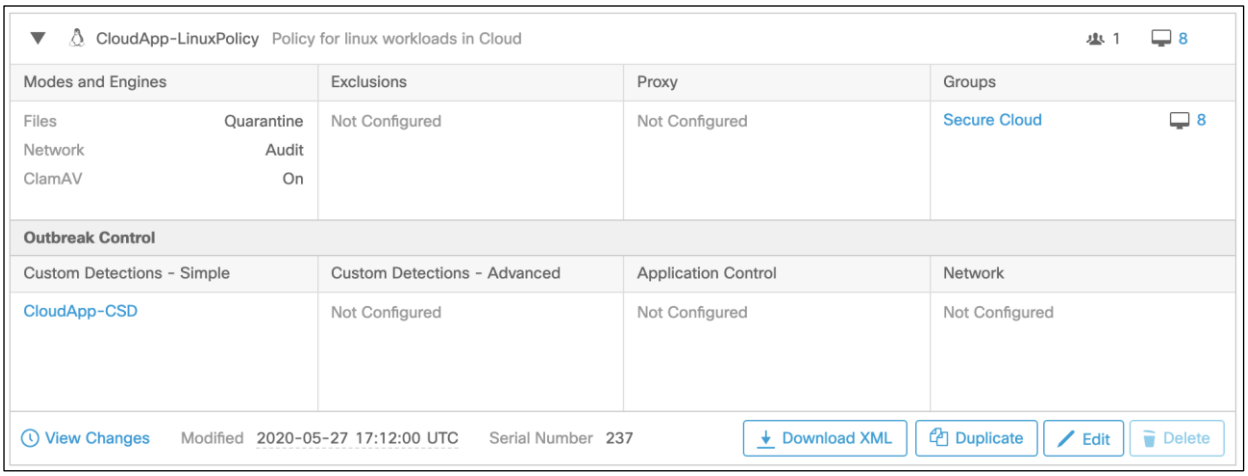
As per our initial AMP4E set up, we had configured the group 'Secure Cloud' (Management > Groups) for our workloads in the AWS cloud.



Note: During our implementation phase we had used the AMP4E agent tied to this specific group 'Secure Cloud', which we had created as part of the initial AMP4E set up (not elaborated in this guide, follow

AMP4E documentation for detailed steps on setting up AMP4E policies). All the workloads in AWS VPC register with AMP Cloud under this specific group.

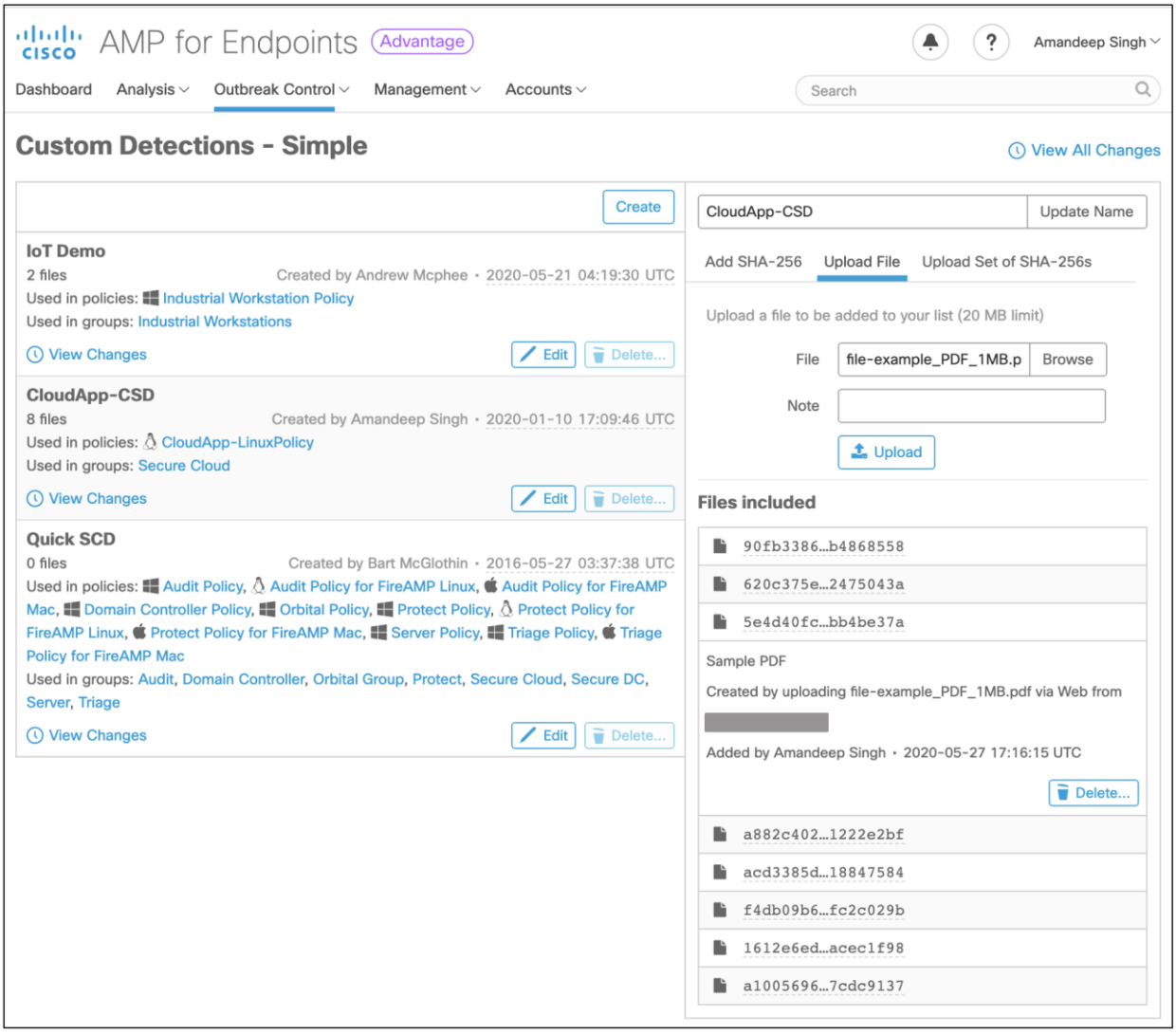
It can be seen in the snapshot above that we tied the specific group to Linux policy 'CloudApp-LinuxPolicy'. Go to 'Management > Policies' and select the specific Linux policy.



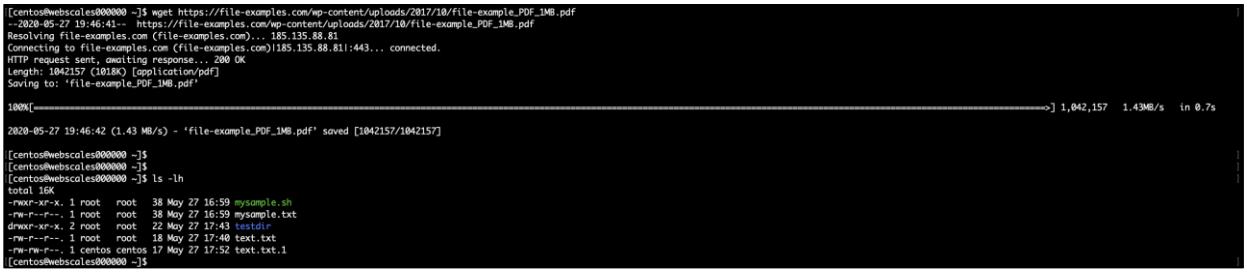
Note: We had preconfigured the Linux policy associated with AMP4E group 'Secure Cloud'. We also tied a new Simple Custom Detection 'CloudApp-CSD' to the Linux policy. If there was no initial config on AMP console, then you would see default policies here.

As we see in the snapshot, the Linux policy above is tied to Simple Custom Detections 'CloudApp-CSD' (Outbreak Control > Simple).

Go to 'Outbreak Control > Simple Custom Detections' and click on edit 'CloudApp-CSD' to upload the PDF file that we want to block list in the AWS cloud environment. Uploading the PDF file will add the SHA value to the SCD policy and quarantines the file associated with it from all the cloud workloads registered under the specific group.

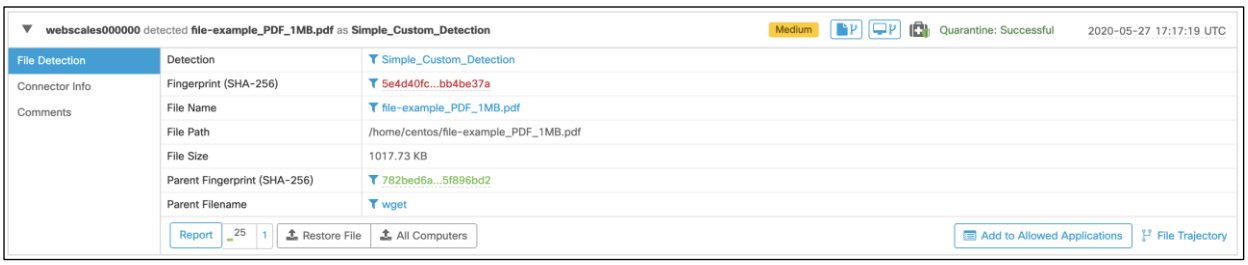


Step 4. **Verify the deletion of the suspicious program –** Log on to a cloud workload, we picked one of the web servers in Web Auto Scaling Group. We downloaded the PDF file that we block listed above. We can see that the file is immediately quarantined by the AMP agent on the workload.



```
[centos@webscales000000 ~]$ wget https://file-examples.com/wp-content/uploads/2017/10/file-example_PDF_1MB.pdf
--2020-05-27 19:46:41--  https://file-examples.com/wp-content/uploads/2017/10/file-example_PDF_1MB.pdf
Resolving file-examples.com (file-examples.com)... 185.135.88.81
Connecting to file-examples.com (file-examples.com)|185.135.88.81:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 1042157 (1018K) [application/pdf]
Saving to: 'file-example_PDF_1MB.pdf'

100%[===================================================================================================================================================>] 1,042,157   1.43MB/s   in 0.7s

2020-05-27 19:46:42 (1.43 MB/s) - 'file-example_PDF_1MB.pdf' saved [1042157/1042157]

[centos@webscales000000 ~]$
[centos@webscales000000 ~]$
[centos@webscales000000 ~]$ ls -lh
total 16K
-rwxr-xr-x. 1 root   root   38 May 27 16:59 mysample.sh
-rw-r--r--. 1 root   root   38 May 27 16:59 mysample.txt
drwxr-xr-x. 2 root   root   22 May 27 17:43 testdir
-rw-r--r--. 1 root   root   18 May 27 17:40 text.txt
-rw-rw-r--. 1 centos centos 17 May 27 17:52 text.txt.1
[centos@webscales000000 ~]$
```

We also confirm the quarantine event from the event logs on the AMP Cloud portal. Log on to the AMP Cloud portal and go to ‘Analysis > Event’, we see a ‘Quarantine successful’ event post our steps above.



Stealthwatch Cloud

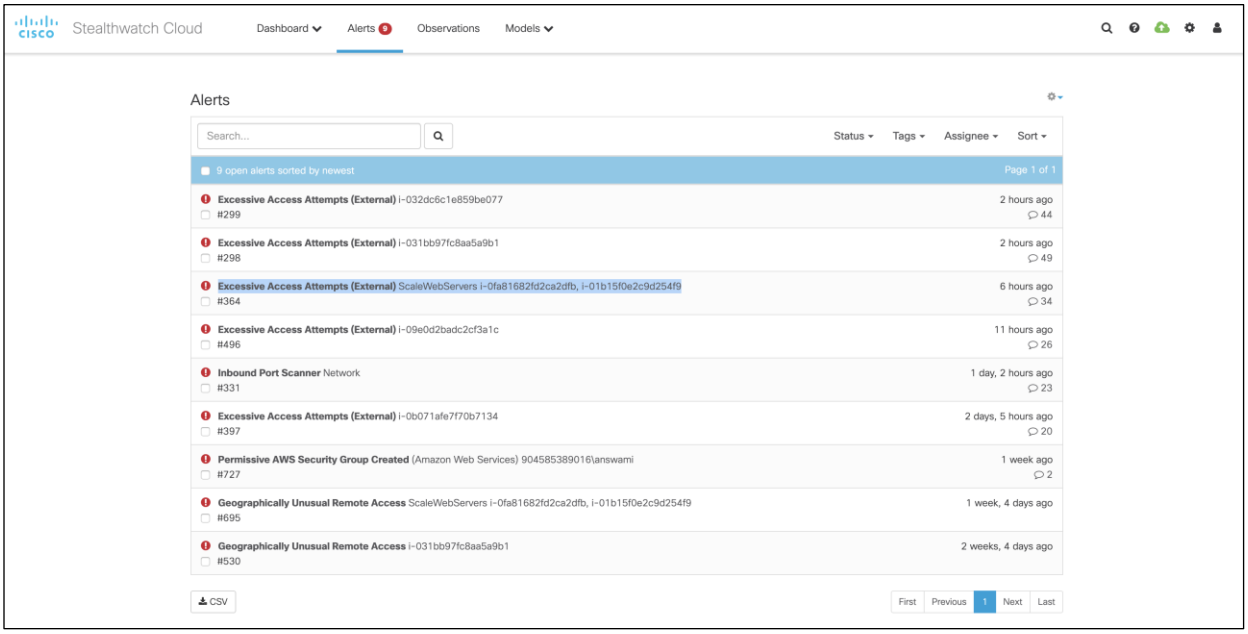
Test Case: Monitor suspicious activity

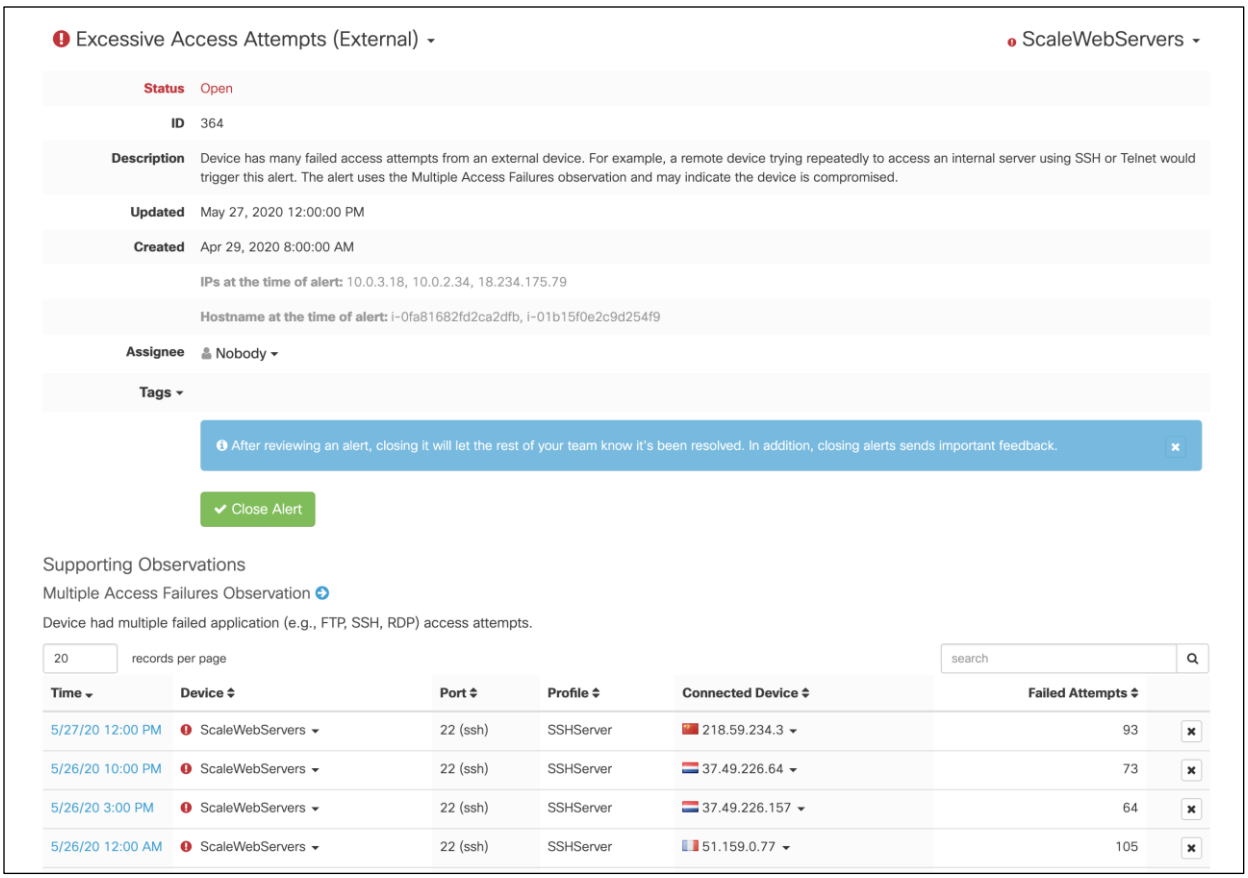
This test case involves using the Stealthwatch cloud to monitor the activity within the AWS cloud environment.

Validation procedure:

Step 1. Monitor suspicious activity in Stealthwatch Cloud

Step 1. **Monitor suspicious activity in Stealthwatch Cloud –** Login to the Stealthwatch cloud portal. Go to 'alerts', we see the alert 'Excessive Access Attempts' as shown below. This alert indicated that there were numerous attempts to get SSH access from an unexpected geo location, which is a suspicious behavior.





Cisco Umbrella

Test Case: DNS security

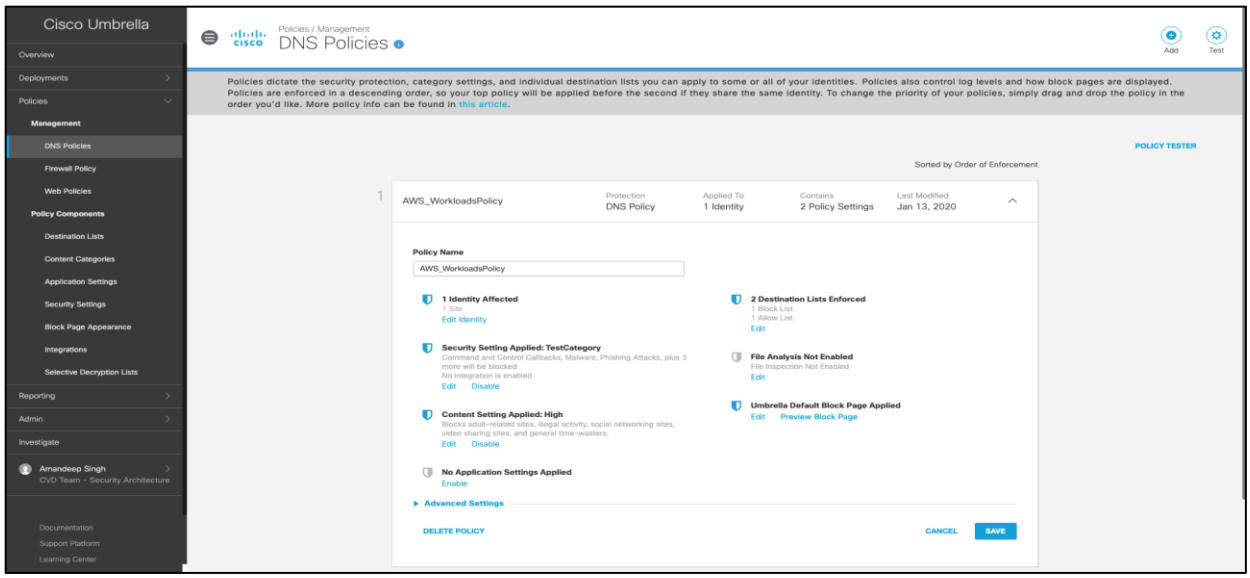
This test case involves adding DNS layer security to the AWS workloads. We created a DNS policy for our tiered application workloads to block malicious domains. To verify the blocks, we accessed a test domain 'examplemalwaredomain.com' and then confirmed the same from Umbrella reporting.

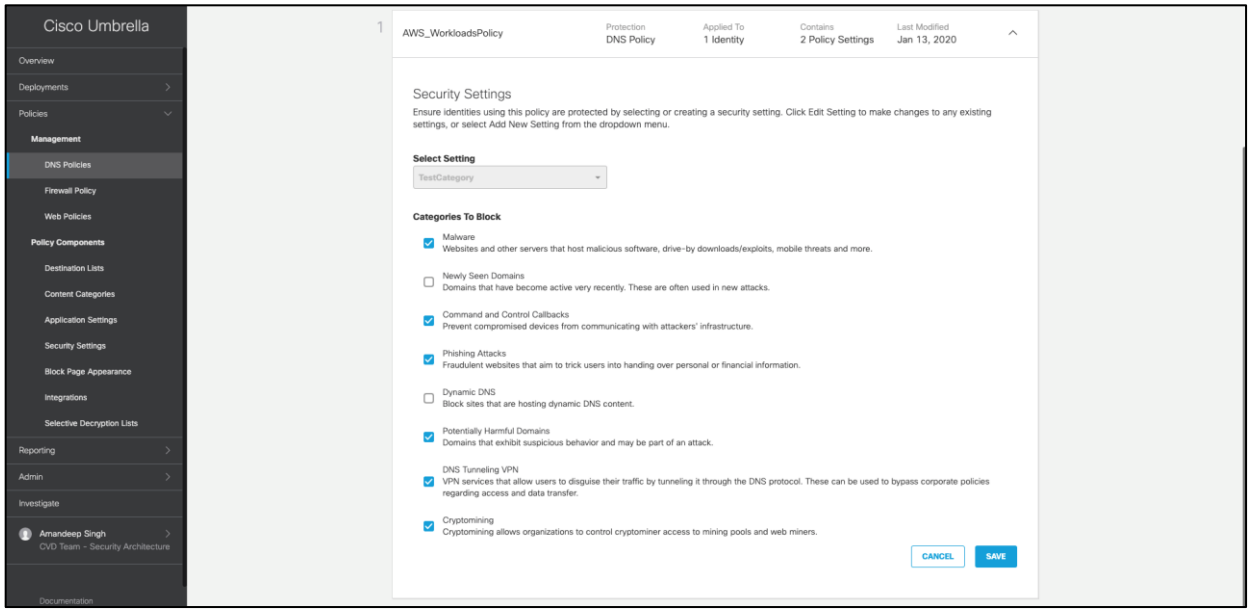
Validation procedure:

Step 1. Set up DNS policy for AWS workloads

Step 2. Confirm if malware domain is blocked

Step 1. Set up DNS policy for AWS workloads – Go to 'Policies > Management > DNS Policies', add a new policy and make sure 'Malware' is set to block under security settings. Save the change.

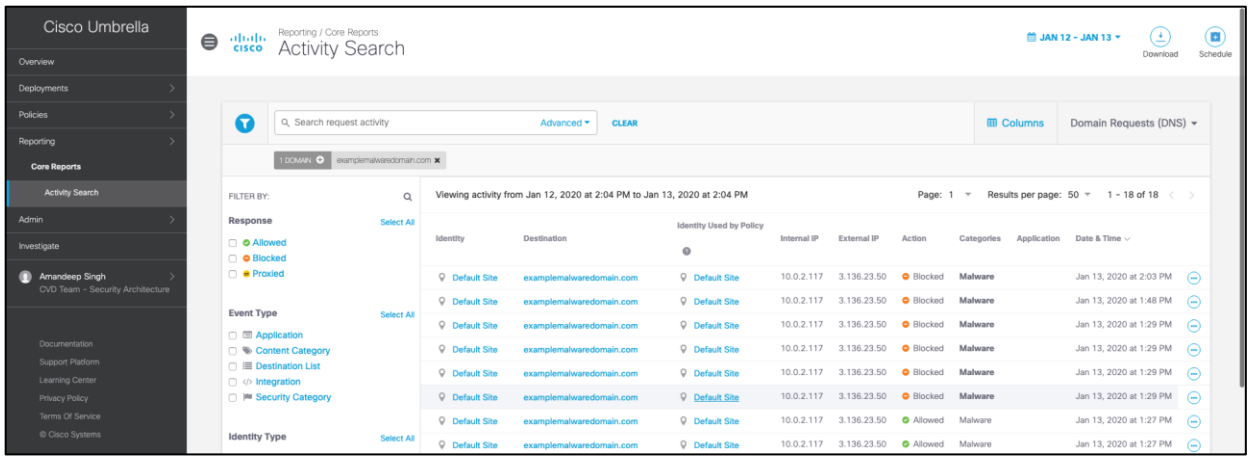




Step 2. **Confirm if malware domain is blocked –** Run 'nslookup' on a test malware domain as shown in snapshot below. Utility returns Umbrella block page IP address as below.

```
[centos@ip-10-0-2-117 ~]$
[centos@ip-10-0-2-117 ~]$ nslookup examplemalwaredomain.com
Server:         10.0.0.100
Address:        10.0.0.100#53

Non-authoritative answer:
Name:   examplemalwaredomain.com
Address: 146.112.61.107
Name:   examplemalwaredomain.com
Address: ::ffff:146.112.61.107

[centos@ip-10-0-2-117 ~]$
```

To further confirm the block action, select 'Reporting > Activity Search' and filter the accessed malware domain. Events show the action as 'Blocked'.



Cisco Defense Orchestrator

Test Case: Enforce Security Group policy using CDO

In this test case, we will try to lock down the outbound access for our cloud workloads for specific TCP ports. We will use CDO to manage the AWS Security Groups. We already onboarded the AWS VPC to the CDO in the implementation section of this document.

Validation procedure:

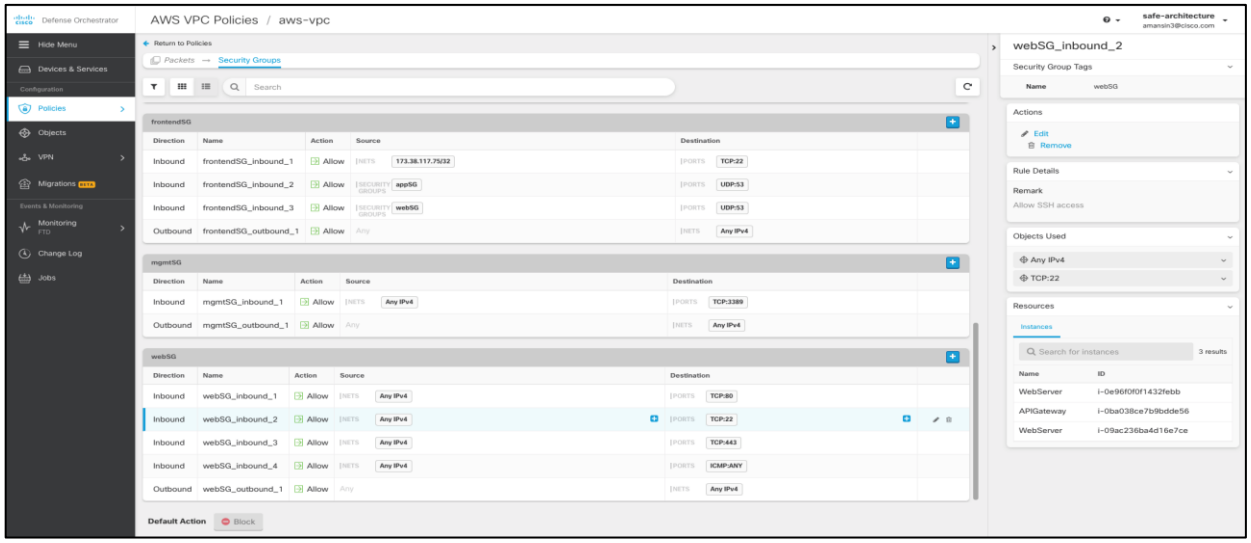
Step 1. Configure and enforce the access policy

Step 2. Verify the access block

Step 1. Configure and enforce the access policy - We log on to a Web Server workload and try to access a non-standard TCP port on a server on the Internet. We can see in the snapshot below that the Web Server workload is able to connect at this point.

```
[centos@webscales000000 ~]$
[centos@webscales000000 ~]$ telnet portquiz.net 666
Trying 52.47.209.216...
Connected to portquiz.net.
Escape character is '^]'.
^C
Connection closed by foreign host.
[centos@webscales000000 ~]$
[centos@webscales000000 ~]$
```

We want to block outbound access to such random TCP ports from our Web workloads. Log on to the CDO portal and go to 'Policies > AWS VPC policies', we can see that the 'WebSG' policy allows the Web workloads to access any destination on any port on the Internet.



We update the policy to lock it down to just the HTTP and HTTPS ports going out to the internet. After making the policy change, click on the notification on the top right-hand side of the portal to push the changes to the specific FTD devices.

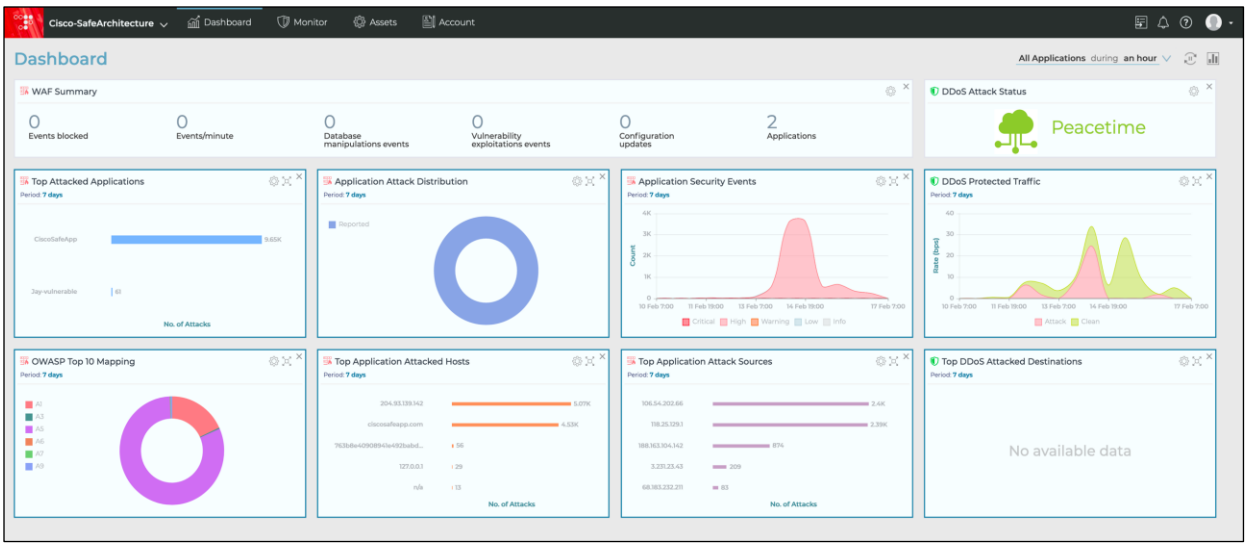
Step 2. **Verify the access block -** Now that we have updated the policy, we will try and attempt to verify the access. We SSH to a web server again and try to access websites on a random TCP port 666. We can see the connection timing out or getting blocked now. We can also see that an outbound access to a server on the internet on standard HTTP and HTTPS is still allowed.

```
[centos@webscales000000 ~]$
[centos@webscales000000 ~]$ telnet portquiz.net 666
Trying 52.47.209.216...
telnet: connect to address 52.47.209.216: Connection timed out
[centos@webscales000000 ~]$
[centos@webscales000000 ~]$
```

```
[centos@webscales000000 ~]$ telnet www.cisco.com 443
Trying 23.196.102.158...
Connected to www.cisco.com.
Escape character is '^]'.
^CConnection closed by foreign host.
[centos@webscales000000 ~]$
[centos@webscales000000 ~]$ telnet www.cisco.com 80
Trying 23.196.102.158...
Connected to www.cisco.com.
Escape character is '^]'.
^C^Z
^Z
Connection closed by foreign host.
[centos@webscales000000 ~]$
```

Radware Cloud WAF and DDoS Protection

Radware Cloud portal displays all the traffic statistics related to various onboarded applications. The dashboards are fully customizable based on the requirements.



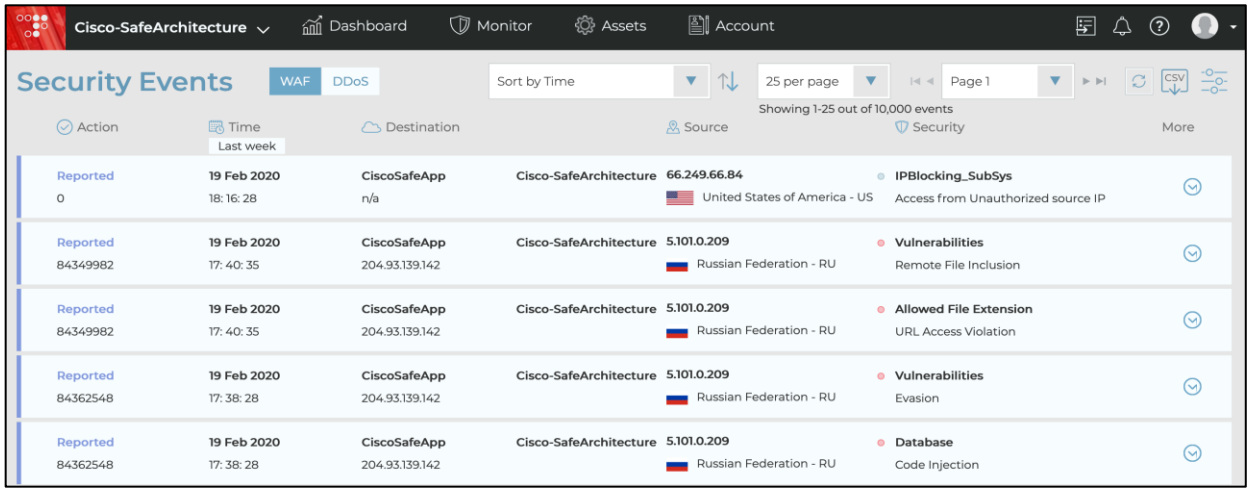
Test Case: Monitor Web and DDoS activity on Radware Cloud.

This test case involves monitoring the security events generated in the Radware Cloud portal.

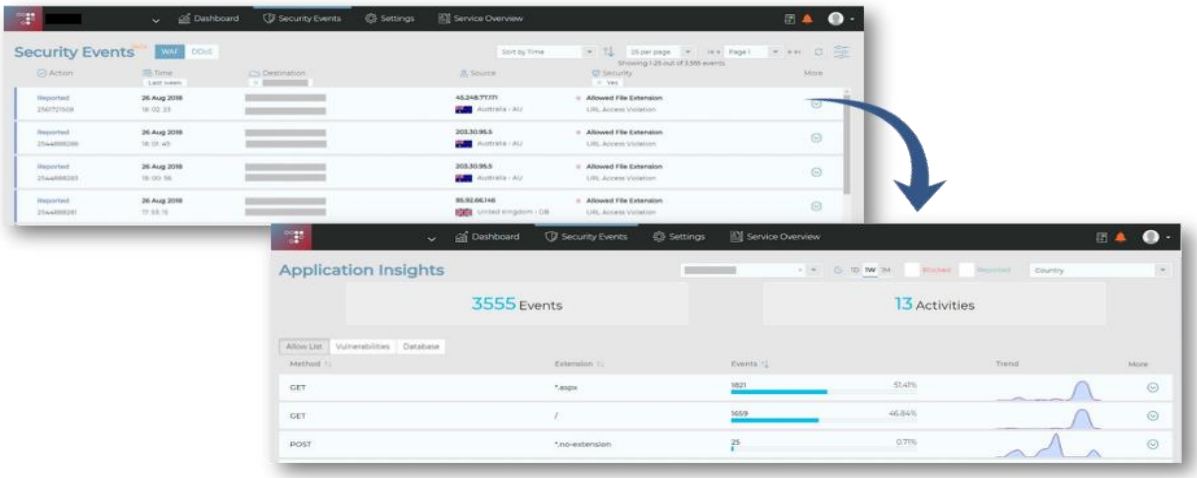
Validation procedure:

Step 1. Monitor Web and DDoS activity on Radware Cloud WAF and DDoS Portal.

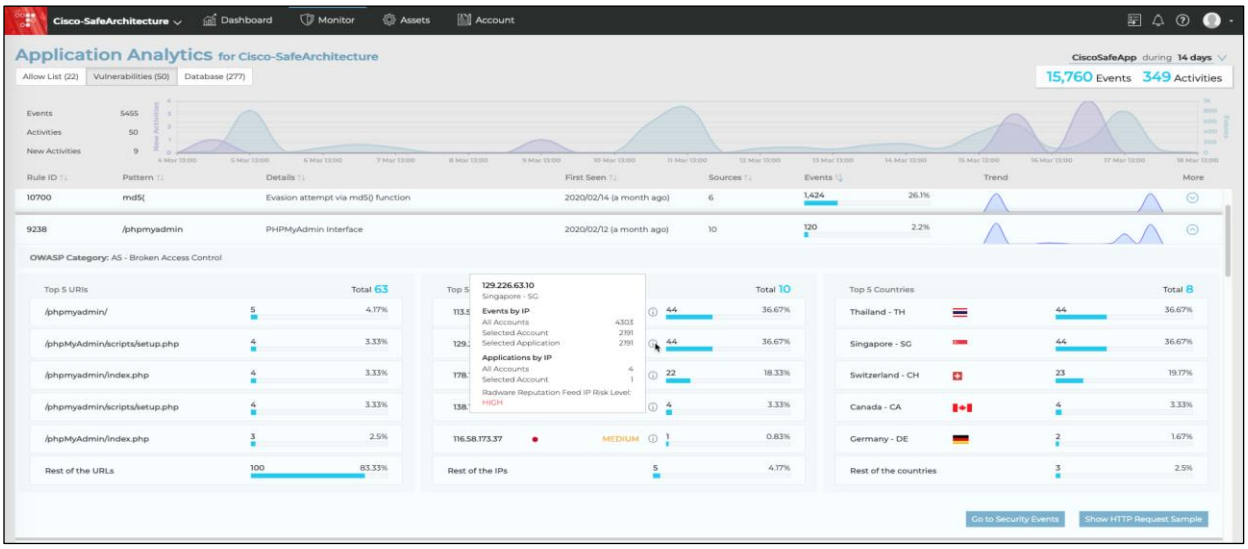
Step 1. **Monitor Web activity and DDoS activity on Radware Cloud –** On the Radware Cloud portal, go to **Monitor > Security Events** to see all the WAF and DDOS events generated from any malicious activity targeting your application.

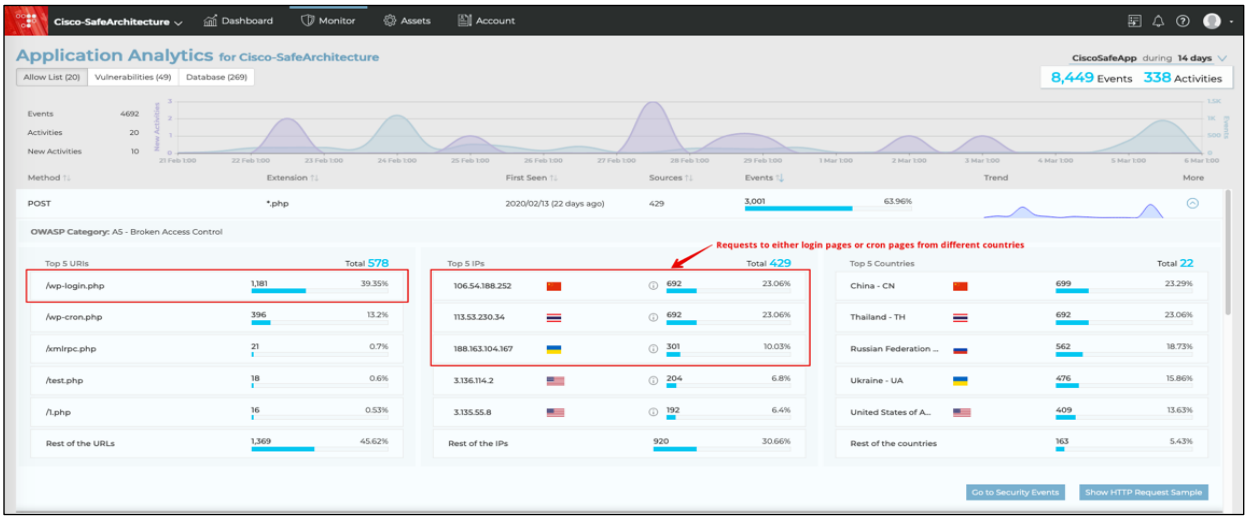


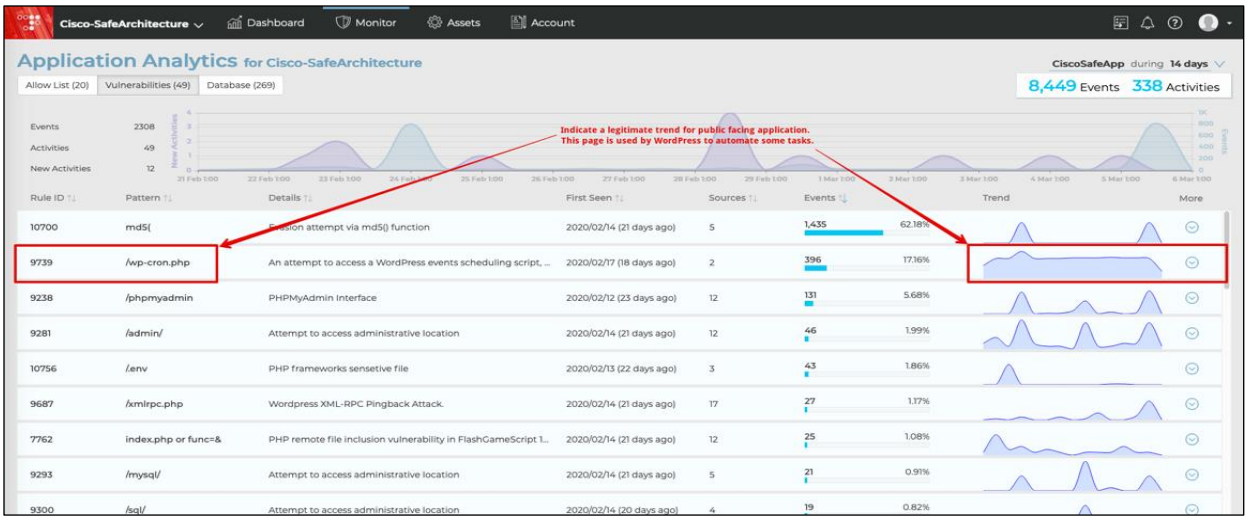
Radware's Application Analytics combines a large number of similar events and consolidating them into small, manageable sets of recurring activities. This helps to streamline response by providing additional context to security events needing attention.



In addition, the integrated ERT Active Attacker Feed will help you identify if listed requests are legitimate or not by identifying known attackers. As Illustrated below, we are able to gain intelligence if the IP's attempting to access the phpMyAdmin pages are from known malicious IPs (with risk level assessments).







Duo Beyond

Validation procedure overview:

- Test Case 1 – Set up the cloud application for Two-Factor Authentication (2FA)
- Test Case 2 – Monitor 2FA activity from Duo admin portal

Test Case 1: Set up the cloud application for Two-Factor Authentication (2FA)

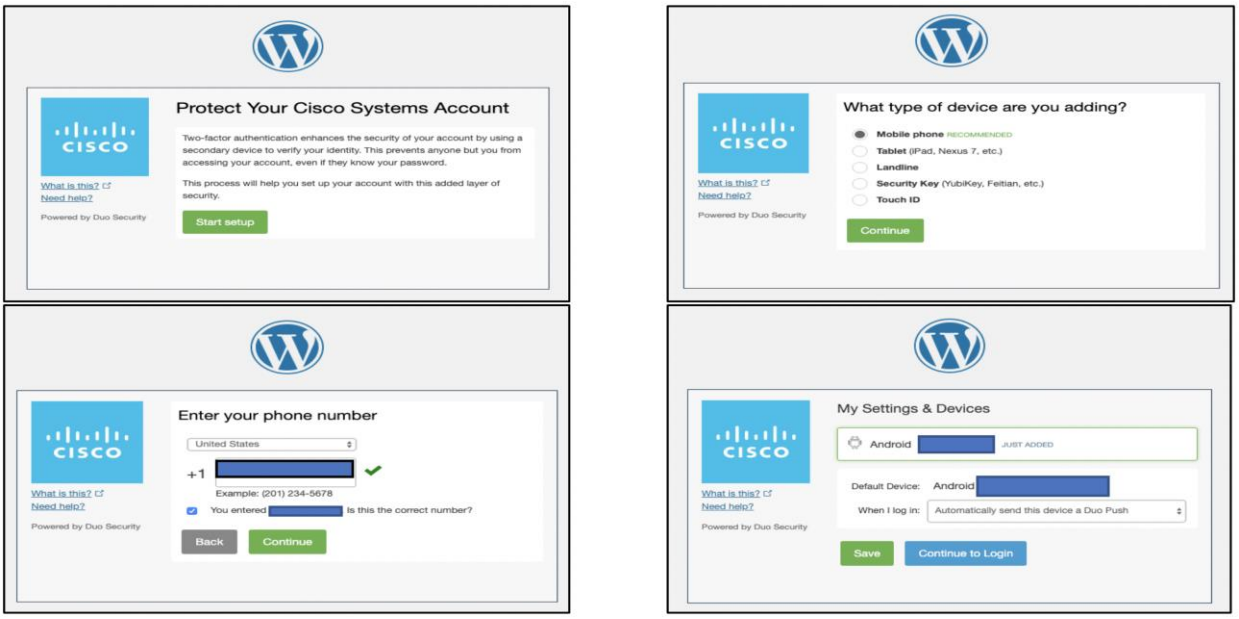
This test case involves logging into the application for the first time and activating the duo plugin. Previously, during the implementation phase, we had already downloaded the plugin to application workloads using AWS User Data option. Follow the Duo documentation (skip step 2 under 'Install and Configure the Plugin') to activate WordPress Duo plugin. After activating the plugin, log out and log in again. This time Duo will prompt the user to enroll their phone for 2FA. After successful enrollment, user gets the ability to approve subsequent login attempts.

Validation procedure:

Step 1. Set up Duo 2FA for a new user

Step 2. Log onto the cloud application

Step 1. Set up Duo 2FA for a new user – After the initial plugin activation, the Duo MFA kicks in and since this is the first authentication attempt, the user is prompted to enroll for MFA.



Step 2. **Log onto the cloud application -** After the enrollment, we continue to log onto the application, this time the user is presented with Duo authentication methods instead of 'setup'. Once the user approves the authentication request, they are allowed to login.

Test Case 2: Monitor 2FA activity from Duo admin portal

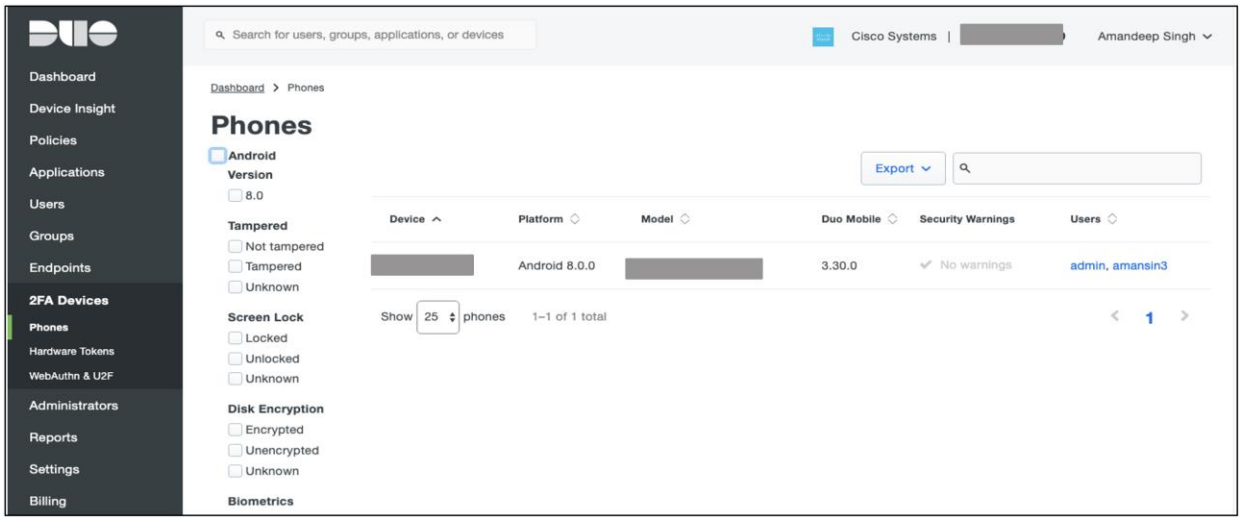
This test case involves monitoring the 2FA enrollment and login activity in the Duo admin portal.

Validation procedure:

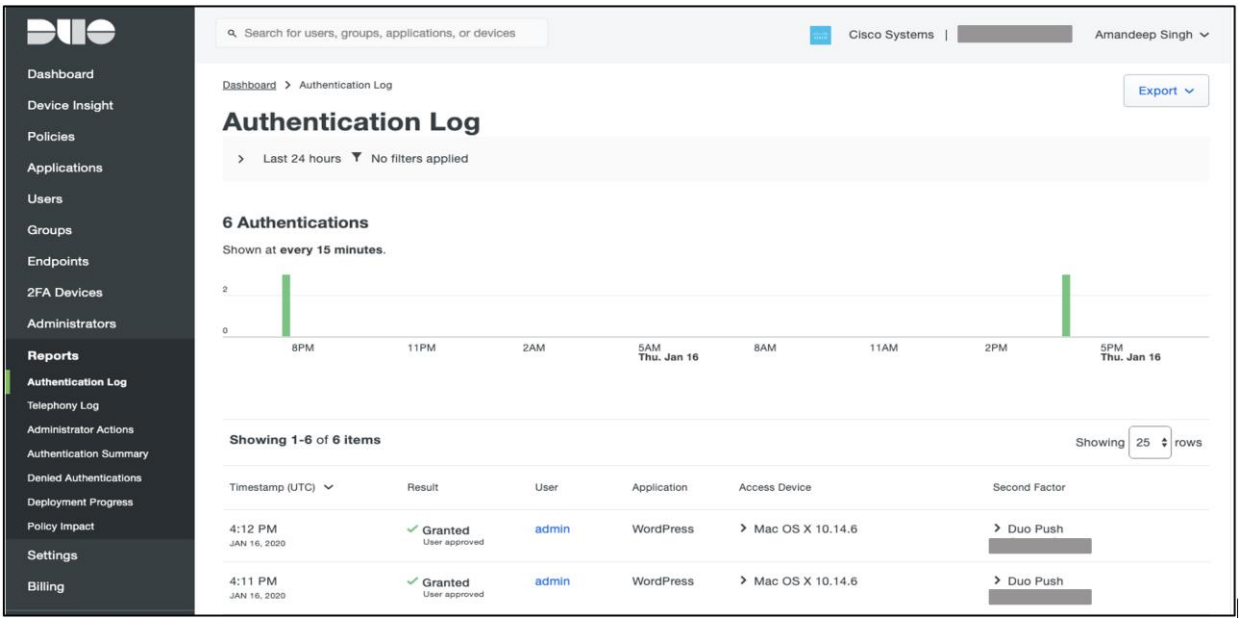
Step 1. **Verify the 2FA enrolled devices**

Step 2. **Track the user logins in authentication logs**

Step 3. **Verify the 2FA enrolled devices -** Logon to the Duo admin portal and select '2FA Devices', the portal shows the list of enrolled devices along with other details like platform, hardware model and usernames.



Step 2. **Track the user logins in authentication logs –** Go to 'Dashboard > Authentication log', to track user 2FA login activity as shown in the snapshot below.



Cisco SecureX Threat Response

Test Case: Track Malicious Activity on threat response

In this test case, we track the life cycle of the malicious PDF that we quarantined using AMP4E in previous steps. We will use the same SHA value and see what threat response offers in terms of visibility in our environment.

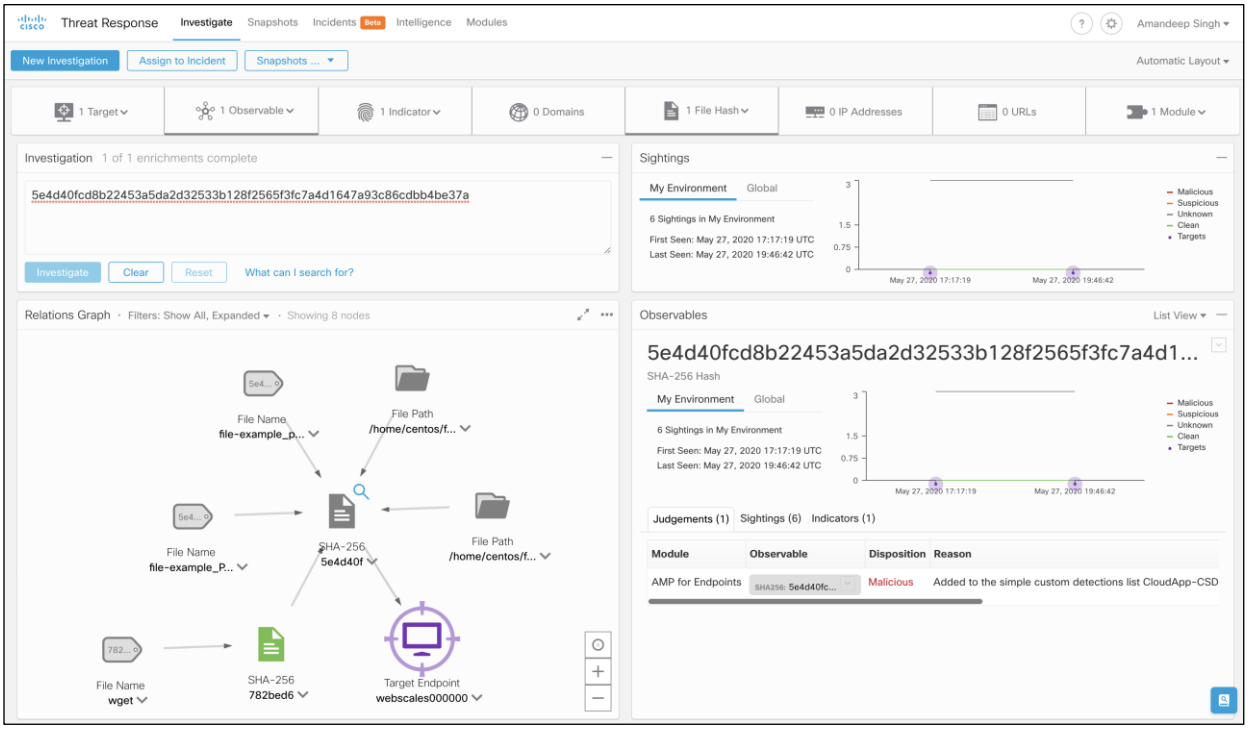
Implementation procedure:

Step 1. Investigate a malicious SHA value

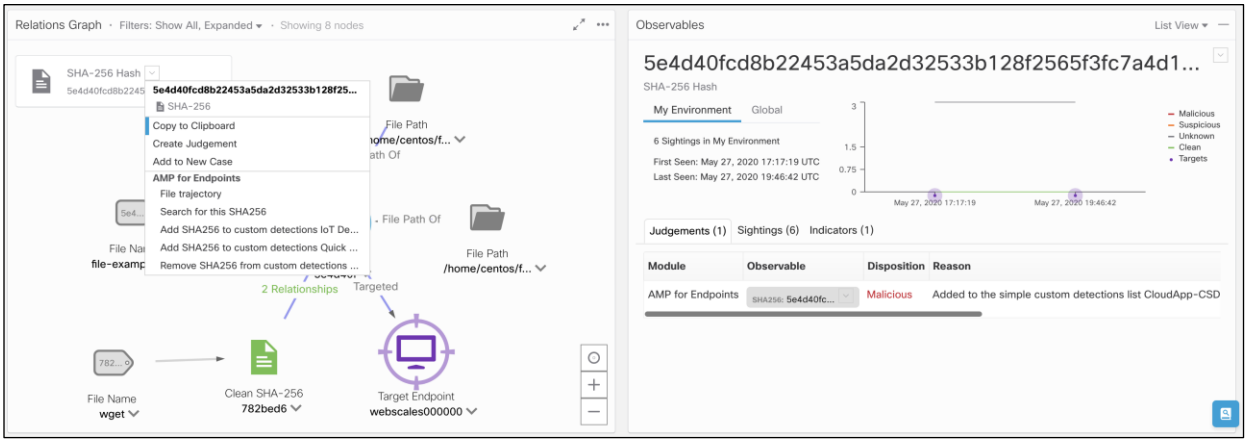
Step 2. Track the file trajectory

Step 1. **Investigate a malicious SHA value –** Log on to the threat response portal and select 'Investigate'. Add the SHA value in provided space and click on 'Investigate'. Threat response pulls all the information about the associated file and what workloads the specific file had interacted with. Under the 'Observables' section, we can see that AMP4E detected this SHA

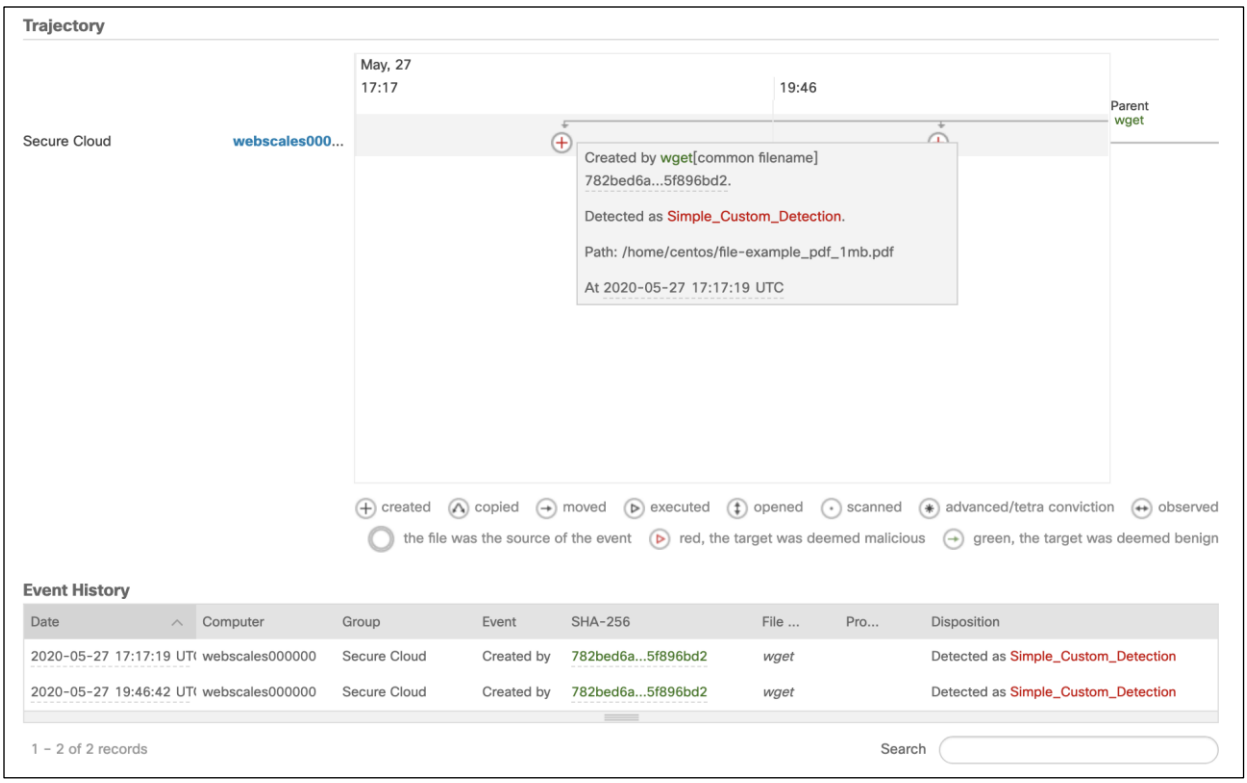
value as malicious based on our custom AMP policy, threat response displays the specific AMP4E policy name as well.



Step 2. **Track the file trajectory -** Click on the 'SHA-256 Hash' shown in the Relations Graph. Expand the drop-down menu and click on 'File trajectory'.



Clicking on 'File trajectory' should redirect you to AMP4E portal page which displays the trajectory of the malicious file on the specific workload. Clicking on a particular timestamp displays the related events. The event history shows all the events associated with the specific file.

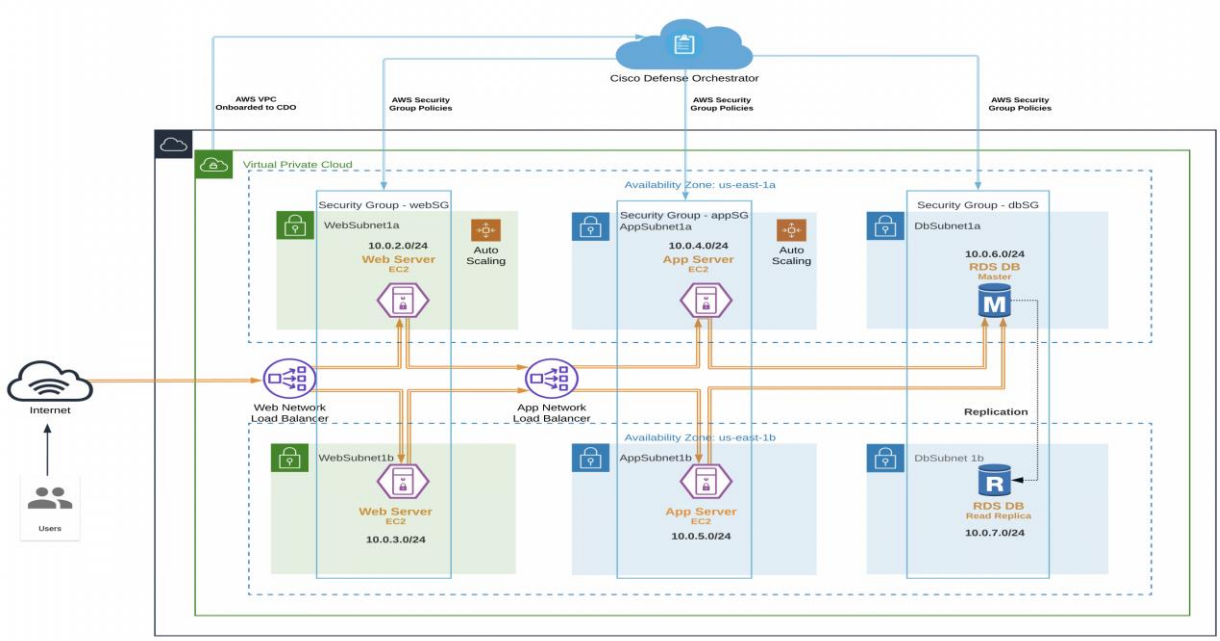


Appendix

Appendix A- AWS Security Groups with CDO

Cisco Defense Orchestrator (CDO) is used for management and policy orchestration. CDO provides one security policy, faster deployment, and smart configuration management. It eliminates the time-consuming complexity of managing policies not just across multiple FTDs and ASAs but also the AWS Security Groups. Cisco Defense Orchestrator helps to correct issues such as unused, duplicate, and inconsistent objects hence ensuring consistent policies for firewalls.

We can use CDO as a single pane to manage the AWS Security Groups, providing a centralized management solution across multiple AWS VPCs.



Appendix B- Acronyms Defined

ALB – Application Load Balancer

AMP4E – Advanced Malware Protection for Endpoints

AVC – Application Visibility and Control

CDO – Cisco Defense Orchestrator

CSD – Custom Simple Detection

CVD – Cisco Validated Design

ERT – Emergency Response Team

FQDN – Fully Qualified Domain Name

IOC – Indicators of Compromise

MFA – Multi-Factor Authentication

PaaS – Platform as a Service

PIN – Places in Network

SaaS – Software as a Service

SWC – Stealthwatch Cloud

VA – Virtual Appliance

VPC – Virtual Private Cloud

2FA – Two Factor Authentication

Appendix C- AWS CloudFormation Template

The AWS CloudFormation template used for the validation testing is located on the Cisco Security Validated Design GitHub. This template can be used to automate the deployment of the networking components,

database, application, and web servers. For more information on the full deployment using AWS CloudFormation, the readme in the GitHub repository goes over all the steps and how it works.

Appendix D- Software Versions

Product	Platform	Version
Tetration	Software agent	3.3.2.35-enforcer
AMP4E	Software agent	1.11.1.663
Stealthwatch Cloud	Cloud Offering	SaaS
Umbrella VAs	Appliance (EC2 Instance)	2.6.2
CDO	Cloud Offering	SaaS
Duo WordPress Plugin	Software Plugin	Version 2.5.5
Radware Cloud	Cloud Offering	SaaS
SecureX Threat Response	Cloud Offering	SaaS
Workloads	Linux	CentOS 7.7
RDS Database	MySQL database	mysql-5-7
Cisco NGFWv	OS	6.6.1.90

Appendix E- References

This section lists all the references.

- **Cisco SAFE:** https://www.cisco.com/c/en/us/solutions/enterprise/design-zone-security/landing_safe.html
- **AWS Three Tier Architecture**: https://d0.awsstatic.com/whitepapers/aws-web-hosting-best-practices.pdf
- **Cisco Tetration:** https://www.cisco.com/c/en/us/products/security/tetration/index.html
- **Cisco Stealthwatch Cloud:** https://www.cisco.com/c/en/us/products/security/stealthwatch-cloud/index.html
- **Cisco AMP for Endpoint:** https://www.cisco.com/c/en/us/products/security/amp-for-endpoints/index.html
- **Cisco Duo Beyond:** https://duo.com/docs/wordpress
- **Cisco Umbrella:** https://docs.umbrella.com/deployment-umbrella/docs/deploy-vas-in-amazon-web-services
- **Cisco Defense Orchestrator:** https://www.cisco.com/c/en/us/products/security/defense-orchestrator/index.html

- **Radware for AWS (WAF and DDoS):**
  https://www.radware.com/products/cloud-waf-service/
- **WordPress:**
  https://wordpress.org/download
- **NGINX:**
  https://www.nginx.com/resources/wiki/start/topics/recipes/wordpress/
- **AWS VPC:**
  https://aws.amazon.com/vpc
- **AWS Route Tables:**
  https://docs.aws.amazon.com/vpc/latest/userguide/VPC_Route_Tables.html
- **AWS Security Groups:**
  https://docs.aws.amazon.com/vpc/latest/userguide/VPC_SecurityGroups.html
- **AWS RDS Database for MySQL:**
  https://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/CHAP_MySQL.html
- **AWS Auto Scale:**
  https://docs.aws.amazon.com/autoscaling/ec2/userguide/AutoScalingGroup.html
- **AWS EC2 Instances:**
  https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/Instances.html
- **AWS Elastic Load Balancing (Application and Network):**
  https://aws.amazon.com/elasticloadbalancing/
- **AWS S3:**
  https://docs.aws.amazon.com/AmazonS3/latest/dev/Welcome.html
- **Amazon Machine Image:**
  https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/AMIs.html
- **AWS Route 53:**
  https://aws.amazon.com/route53/features/

Americas Headquarters
Cisco Systems, Inc.
San Jose, CA

Asia Pacific Headquarters
Cisco Systems (USA) Pte. Ltd.
Singapore

Europe Headquarters
Cisco Systems International BV Amsterdam,
The Netherlands

Cisco has more than 200 offices worldwide. Addresses, phone numbers, and fax numbers are listed on the Cisco Website at **https://www.cisco.com/go/offices**.