Threater Enforce in AWS - Gateway Load Balancer

October 21, 2025

1. Overview

This document provides end-to-end guidance for a typical deployment of Threater Enforce software leveraging a **Gateway Load Balancer** in AWS, including a working example configuration.

2. Prerequisites

It is possible that you may be reading an old version of this document. We recommend that you check the following publicly available link to make sure you have the most recent copy of this document:

https://threater-marketplace.s3.amazonaws.com/Threater+Enforce+in+AWS+-+GWLB.pdf

This document will be most useful to readers who are IT and/or cyber security professionals with:

- A solid understanding of computer networking.
- At least a cursory understanding of AWS.
- At least a cursory familiarity of Threater Enforce and the Threater Portal.

Additionally, although it is not explicitly required, readers who have a deep working knowledge of standard AWS principles, including VPCs, security groups, subnets, routing, Internet gateways, and instance management will gain the most benefit from this document. This is because cloud deployments can feel overwhelming at first for IT personnel who have never managed cloud services. There is much to learn!

The following AWS deployment prerequisites are also important:

- An existing AWS account with AWS console access.
- Sufficient permission to perform deployments via AWS CloudFormation.

IMPORTANT: Keep in mind that the tools and instructions presented here only demonstrate one possible use-case of Threater Enforce in AWS; they are not a "turn-key" or "one size fits all" solution for securing your AWS infrastructure. The goal of this document is to provide you with the information and tools necessary so that you can integrate Threater Enforce into the specific needs of your cloud security stack.

In the following document it is important to recognize some of the subtle differences in terminology:

- **Enforce:** Threater provided **software only** component of deployment
- **Enforce Instance:** Deployed **instance** running Enforce software (including host and operating system); can be virtual or on-premises

Lastly, it is worth mentioning some abbreviations you may encounter in this document:

- **AMI:** Amazon Machine Image
- **ARN:** Amazon Resource Names
- **ASN:** Autonomous System Number
- **BYOL:** Bring Your Own License
- **DHCP:** Dynamic Host Configuration Protocol
- **DPDK:** Data Plane Development Kit
- **EULA:** End User License Agreement
- **GDPR:** General Data Protection Regulation
- **GENEVE:** Generic Network Virtualization Encapsulation
- **GWLB:** Gateway Load Balancer
- **HA:** High Availability
- **IAM:** Identity and Access Management
- **IOC:** Indicator of Compromise
- **JSON:** JavaScript Object Notation
- **NAT:** Network Address Translation
- **RFC:** Request For Comments (Internet Specification Document)
- **SIEM:** Security Information and Event Management
- **SLA:** Service Level Agreement
- **TLS:** Transport Layer Security
- **UI:** User Interface
- **VPC:** Amazon Virtual Private Cloud
- **VPN:** Virtual Private Network

3. Note About Customer Data Retention and Privacy

Before going further it is important to mention that customer configuration data is retained and managed by the deployed Threater Enforce software. This includes information such as local administrator usernames and passwords, as well as detailed connection logging information. It also includes non-customer-specific information, such as standard out-of-the-box threat feeds and related threat intelligence.

Any and all "customer" specific attributed information is transmitted nowhere else, ever, until and unless the customer decides to do so. For example, it is common for advanced customers to choose to export our RFC-compliant logs data to any number of third-party SIEM tools of their choosing. Common connectivity that we see customers use include connectors to systems like Splunk, IBM QRadar, and Gravwell. For those who are interested, we talk more about such configurations later in this document when we briefly discuss software configuration, akin to what our customers have come to expect from our on-premise deployments.

And, of course, the protected instances should always be considered by the end customer as points of presence for customer data. After all, you're installing Threater Enforce for a reason, and one primary reason is protecting your data stored/used in any number of instances sitting behind an Enforce deployment! That's the beauty of Threater Enforce running both in AWS and on-premise: they can protect anything, anywhere, seamlessly, operating effectively as a bump-in-the-wire.

We take great pride in collecting as little information as possible about our customers and even when we do have reason to collect customer-attributable information (for example, detailed access or logging information), it is transmitted nowhere at all without the customer taking action. Our customers decide where their data goes, always, without fail. This has allowed us to do very well in places in the world where privacy is at a premium, such as the European Union (GDPR, etc.), where we have many customers.

4. Threater Enforce in AWS (Gateway Load Balancer)

Threater Enforce is the only active defense cybersecurity platform that fully automates the enforcement, deployment, and analysis of cyber intelligence at a massive scale. As the foundational layer of an active defense strategy, our patented solution blocks known threats from ever reaching your networks. Threater Enforce utilizes immense volumes of cyber intelligence from over 50 renowned security vendors to provide unparalleled visibility over the threat landscape resulting in a more efficient and effective security posture. Security teams at companies of all sizes use Threater Enforce to deploy active security, gain real-time network visibility into threats and policy violations, ensure their network is protected, and reduce manual work.

Threater Enforce:

- Deploys as a standard AWS image with Threater Enforce software pre-installed and a **BYOL subscription model.**
- Allows Threater patented technology to protect your AWS infrastructure by allowing and/or blocking incoming and/or outgoing packets in real-time, based on policy and list configurations.

Configuration is managed entirely in the cloud via the Threater Portal which is hosted at **https://portal.threater.com** and provides centralized management of all Enforce instances regardless of whether they are deployed on-premise, in the cloud (such as AWS), or both. The platform features always-on control and synchronization of geo-IP data, ASNs, allowed lists, denied lists, threat lists, policies, and more in real-time. Threater Enforce provides best-in-class protection with no measurable impact to network performance, regardless of the number of IOCs that you are protected against. Any changes in configuration of any type, including list contents and policies, are always propagated in real time to all Threater Enforce software installations, whether they are on-premise or in the cloud.

5. BYOL Support and Pricing

Our **BYOL** Threater Enforce customers with active subscriptions are able to receive various levels of support. As our support plans can vary over time and in some cases from subscription type to subscription type, we do not embed tier descriptions or SLAs and the like in this deployment document. Instead, we refer the reader to our online corporate website documentation, with a good starting point being the following link:

https://support.threater.com

Our software subscription pricing is identical for both our on-premise and cloud deployments. For detailed pricing information for standard BYOL subscriptions, please see our support link above. Existing on-premise customers looking to leverage our solutions in AWS can of course contact their existing Threater Sales Representative for more complex configurations. Also, it is trivial from within our Portal to move BYOL subscriptions from existing on-premise devices to new cloud instances.

6. Threater On-Premise vs. Threater in AWS

Traditional on-premise Threater Enforce software installations operate as a layer 2 bump-on-the-wire. Anything arriving at the "inside" port will be propagated to the "outside" port and vice versa, unless the packet is blocked due to your configured policy and list configurations.

In the cloud, things are a bit different, since you don’t have full control of the underlying networking like you do in on-premise environments. Fortunately, The AWS Gateway Load Balancer together with a virtual appliance running Threater Enforce software can be leveraged to provide the same level of protection as on-premise deployments. Using the Gateway Load Balancer AWS resource, Threater Enforce can efficiently examine ALL VPC ingress and egress traffic by adding a VPC Endpoint and a few routes to any existing production VPC.

7. Supported Host Architectures

Threater Enforce can be deployed onto either x86 or ARM based virtual machines within AWS. The Enforce software will function the same regardless of the underlying host architecture. The decision to choose one architecture over the other can be based simply on availability and lower cost.

8. Integration with the Threater Portal

Our Threater Enforce software in AWS interacts with the Threater Portal in exactly the same way that our on-premise deployments do, with no exceptions.

All of our Threater Enforce deployment paradigms leverage exactly the same codebase.This is very different from offerings from other security vendors in the Marketplace who had to create unique form-feature-function products between their on-premise and cloud offerings. Because of our patented architecture, we didn’t have to do that. It’s exactly the same.

This was achievable for us since our Threater Enforce on-premise and AWS-based design is based on a Linux software stack leveraging a bump-in-the-wire network architecture which further leverages DPDK, which deploys beautifully on both standard on-premise equipment as well as into AWS.

Our customers use the same Threater portal for management whether they are deployed on-premise, or in the cloud, or any combination thereof, without having to make any distinction between on-premise or cloud-based deployments.

9. Example Network Deployment Diagrams

This section outlines a simple AWS Gateway Load Balancer deployment that features two consumer virtual machines each exposing interfaces with public IP addresses that are protected by a pair of Enforce instances. Note that the following diagrams are simply adaptations of the example found in the AWS ***Gateway Load Balancer Getting Started Guide*** located here:

https://docs.aws.amazon.com/elasticloadbalancing/latest/gateway/getting-started.html

We have expanded the proposed solution provided by AWS to demonstrate where and how an Enforce instance integrates with the AWS Gateway Load Balancer. We have attempted to follow the naming conventions (VPCs and subnets) in the example provided by AWS as closely as possible with the intention of leveraging that existing documentation in our examples.

In AWS the Gateway Load Balancer is assigned its own VPC and paired with an appliance target group that contains one or more "security appliances." In our example the security appliance is of course an Enforce instance. AWS terms this VPC the "Service Provider VPC." The role of this VPC is to provide protection to other "consumer" VPCs. For security purposes the Service Provider VPC should not contain other resources that are not specifically related to Gateway Load Balancer/Enforce instance operation. One possible exception here would be SIEM resources to collect syslog export information.

In order to exercise the Service Provider VPC we will need at least one other VPC to create Internet traffic flows that we can configure for inspection by the Service Provider VPC and the Enforce instance. However, any existing VPC could also be configured to route ingress and egress traffic through a Service Provider VPC. You can use the "minimal" VPC example we provide, make your own VPC from scratch, or use an existing VPC you have available.

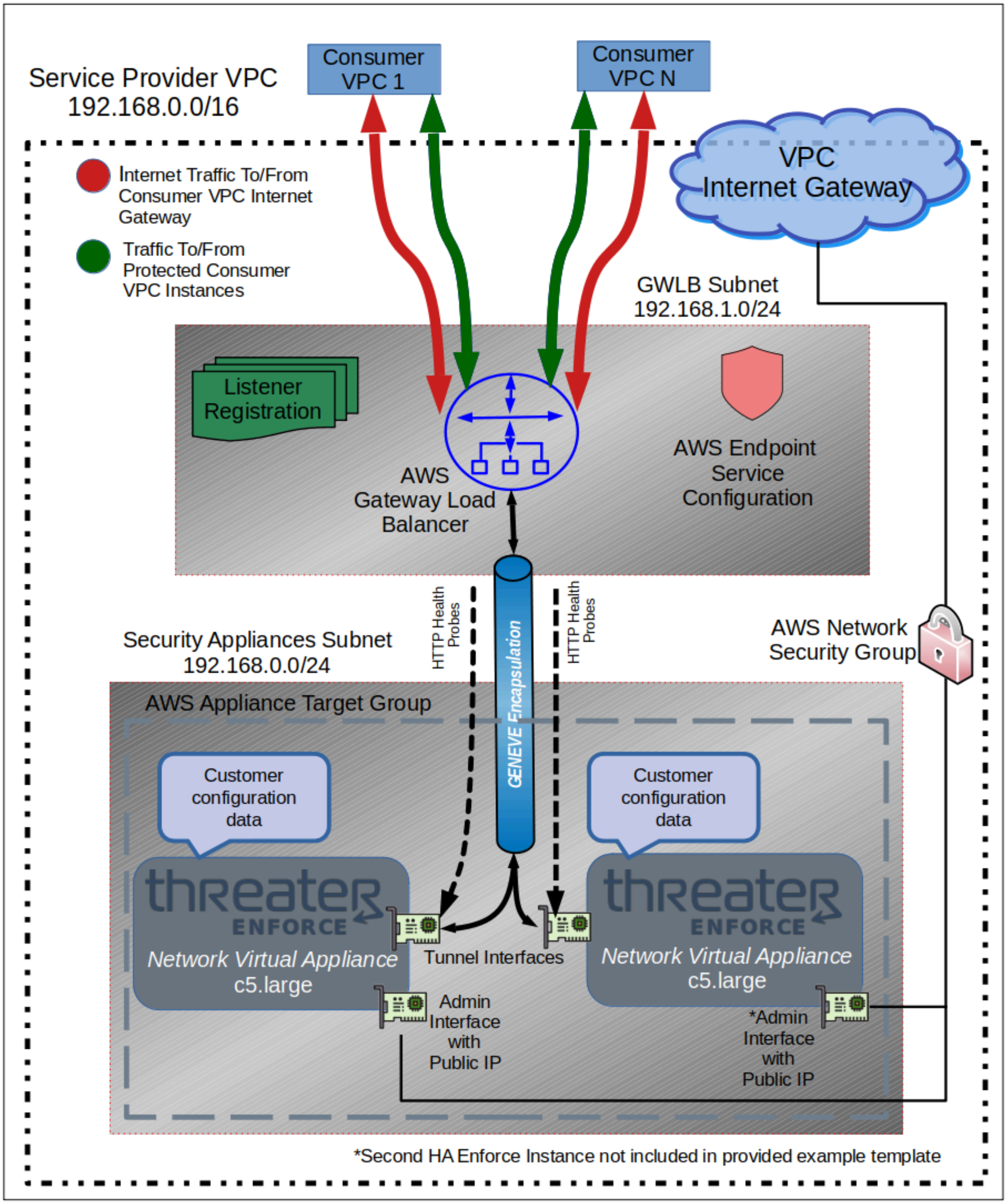
The following sections describe the two VPCs that we have implemented using CloudFormation templates It should be noted that AWS terms VPCs that utilize services of a Provider VPC "Service Consumer VPCs." Thus, we will be referring to these VPCs quite often:

- Service Provider VPC
- Service Consumer VPC

9.1. Service Provider VPC

The following diagram focuses on the Service Provider VPC component of the example. The VPC is assigned a private address space of 192.168.0.0/16. You are free to choose any valid AWS VPC address space you like. In addition, there are two subnets assigned to the VPC: a "GWLB Subnet" and a "Security Appliances" subnet. In most cases your production Service

Provider VPC will closely resemble the example we have provided. The following sections describe each VPC component in more detail.



9.1.1. Gateway Load Balancer

In the provided example the Gateway Load Balancer is deployed into the GWLB Subnet with address space 192.168.1.0/24. In order to make the Gateway Load Balancer useful, we need to register some backend appliance instance targets and also set up some permissions.

9.1.1.1. Enforce Instance Target Registration

When traffic from other VPCs arrives at the Gateway Load Balancer for inspection, there must be a list of devices prepared to handle traffic flows (in our case this is an appliance target group of one or more Enforce instances). Therefore, we must make the Gateway Load Balancer aware of these appliances that are available to it.

In AWS a unique appliance target can be defined by:

- The AWS instance ID
- Use of port overrides
- The private IP of an instance's interface
- AWS Lambda functions

Given that Threater Enforce software in Gateway Load Balancer mode must operate with two network interfaces (admin and tunnel) we must specify the target by IP address so that the Gateway Load Balancer forwards traffic for inspection to the tunnel interface and NOT to the admin interface of the Enforce instance. **Therefore, we must register Enforce instance application targets by interface IP address.** None of the other target types should be used when registering Enforce instance appliance targets.

9.1.1.2. Endpoint Service Configuration

The AWS Gateway Load Balancer and Enforce instance operate inside of their own VPC as an AWS "Endpoint Service." The services provided by the VPC could potentially be offered to any other VPC so by default all access is disabled. Thus, we need to give access to only those resources that should have permission to use the VPC and the Enforce instance.

In our example we keep things simple and add the account "root ARN" to the list of allowed principals for the endpoint service of the Gateway Load Balancer. This will open access to all requests from other VPCs owned by our account. Additionally, we disable the "acceptance required" option of the service configuration to make deployment as seamless as possible. These are configuration options you will want to evaluate carefully in a production deployment.

9.1.2. Threater Enforce Interfaces

9.1.2.3. Tunnel Interface

GENEVE encapsulated traffic from the Gateway Load Balancer arrives for inspection at the Enforce instance's "tunnel" interface. Note that the Gateway Load Balancer ensures flow stickiness by routing both forward and reverse flow packets to the same Enforce instance in cases where two or more Enforce instances are in use.

From the perspective of an end-user, the GENEVE encapsulation is mostly irrelevant as it only applies to traffic that flows between the Gateway Load Balancer and the Enforce instance. The Enforce software is equipped to handle the encapsulation "out-of-the-box" so there is nothing that needs to be configured on the instance itself. When the Threater Enforce boots, it will recognize that it is paired with an AWS Gateway Load Balancer and take care of managing the encapsulation for you.

More information on how Threater Enforce integrates with the Gateway Load Balancer can be found here:

https://aws.amazon.com/blogs/networking-and-content-delivery/integrate-your-custom-logic-or-appliance-with-aws-gateway-load-balancer/

Note that the Enforce instance "tunnel" interface handles BOTH inbound and outbound traffic. With an AWS Gateway Load Balancer, there is no concept of a physical inside/outside bridging pairs as exist with on-prem deployments; nor does the GENEVE header include information regarding packet direction. Therefore, packet direction is determined by examining the source IP address of each packet (without encapsulation) to determine if the packet came from the Internet Gateway (public IP) or a protected instance (private IP).

In our example we apply a strict inbound security group policy to the Tunnel interface such that only those services that are applicable to Gateway Load Balancer/Threater Enforce interoperation are permitted. The permitted services are:

- UDP on port 6081 outbound (GENEVE tunnel traffic)
- TCP on port 80 (Health Probes from Gateway Load Balancer to Threater Enforce)

9.1.2.4. Admin Interface

The Threater Enforce standard administration interface is the default interface that the instance receives when it is deployed. In our example, we’ve specified our appliance admin target subnet as 192.168.0.0/24 so the Enforce instance will be assigned a private IP in this address range.

You can choose to assign your Threater Enforce admin connection a public-facing IP on AWS. We do that in our example configuration, and as such we will make absolutely sure to lock down access via a proper AWS security group definition so that administration access is available only to individuals and systems who should have access. However, at a minimum you will need to open HTTP (port 443) for inbound connections so that the Enforce instance can be managed via its user interface.

Alternatively, if your IT staff is knowledgeable about advanced AWS configurations, it is certainly wise to disavow a public IP altogether, and use a properly configured VPN to access your VPC's administration subnet. Those and other more advanced configurations are beyond the scope of this document, and if they are of interest to you, we recommend that you contact AWS directly for assistance and training as needed.

9.1.2.5. Health Probes

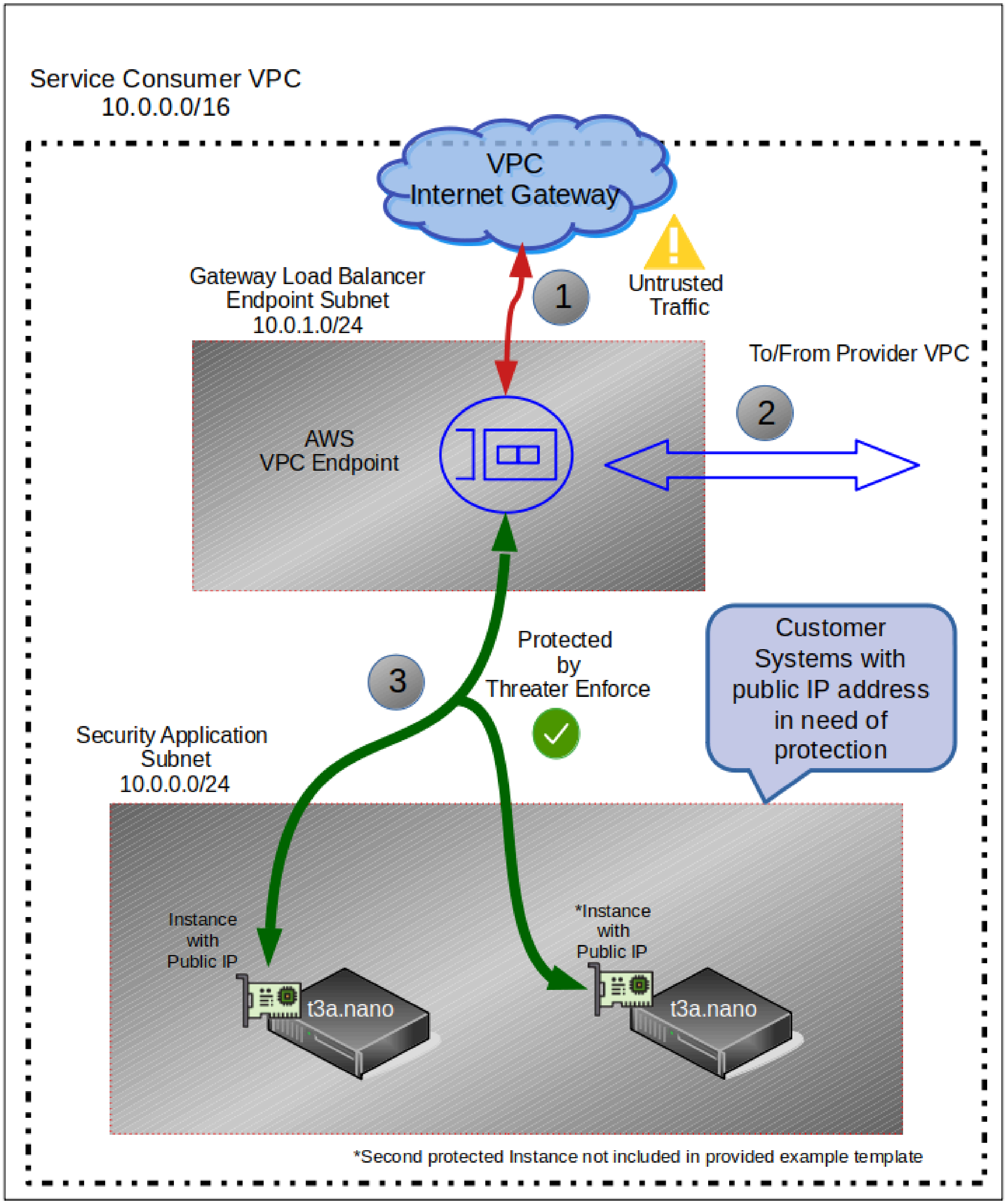
The Gateway Load Balancer will also issue health probe checks on configurable intervals to ensure that associated Enforce instances are operating correctly and can accept new flows. Note that these probe messages are sent to the "tunnel" interface -- not the admin interface. When the Enforce software is operating normally (ready to accept traffic), it will respond as healthy and the Gateway Load Balancer will be free to send it new flows for inspection. Important: The Enforce software only responds to HTTP (port 80) health probe requests at the path:

/api/v1/healthcheck/gwlb

Threater Enforce will not respond to any other health probe configuration so it is absolutely imperative that the health check is configured correctly in AWS. If health probes are not properly configured then connections to protected instances will likely be unreachable as the Gateway Load Balancer will determine that there is no appliance available to handle new connections.

9.2. Service Consumer VPC

The diagram below describes the Service Consumer VPC component of the example. This VPC contains the resources you intend to protect with a Gateway Load Balancer and Enforce instance. Thus, your production consumer VPC will certainly be different from this simplified example we present here. In fact, if you are looking to protect an existing VPC with a Gateway Load Balancer enabled Threater Enforce then your consumer VPC will of course already exist. In that case this example will still be valuable because it demonstrates how to integrate the required resources and routing to your current production VPC to get up and running with a Gateway Load Balancer enabled Threater Enforce.



For this example we have selected a VPC network address space of 10.0.0.0/16 and two subnets: a VPC Endpoint subnet and a subnet for the instances we are protecting with Threater Enforce. You can have as many “application” subnets as you like but the VPC Endpoint must have its own subnet (this allows more specific routing for traffic to and from the VPC endpoint). If your scenario warrants modifying this address space, you must of course adjust all other private IP mappings for the various associated subnets so that they fall within the constraints of the network you choose to deploy.

Without the VPC Endpoint and routes configured, traffic flows freely from instances in the Security Application Subnet and the Internet. Instances in the application subnet are unprotected. In order to route all inbound and outbound traffic via the Enforce instance(s) in the Service Provider VPC we need to make the following additions to the Service Consumer VPC:

- Add a VPC Endpoint subnet and add a VPC endpoint to it.
- Add a route to the Internet Gateway so that all inbound traffic is routed to the VPC endpoint upon entering the VPC.
- Add a route to the application servers subnet so that outbound Internet traffic is routed to the VPC endpoint and NOT directly to the Internet Gateway.
- Add a route to the VPC endpoint subnet so that outbound traffic returning from inspection is routed to the Internet Gateway.
- Setup Permissions on the Gateway Balancer Endpoint

Once the proper routing is in place, inbound traffic should traverse the path in the order of the bubbles labeled 1, 2, and 3. Outbound traffic simply takes the same path but in reverse. Once all of these components are in place, all inbound and outbound traffic is forced to take a detour for inspection by the Threater Enforce in the Service Provider VPC.

The AWS Gateway Load Balancer Getting Started Guide offers more information regarding the VPC endpoint, routes, and service permissions. That guide can be found here:

https://docs.aws.amazon.com/elasticloadbalancing/latest/gateway/getting-started.html

10. Deployment Time Guidance

The deployment time will likely vary from customer to customer, depending on your level of expertise and familiarity with the cloud, and depending on whether you are building out everything from scratch.

Generally, by using the AWS CloudFormation template we provide in a subsequent section, deployment will take anywhere from 15 to 30 minutes, total.

11. Marketplace Image

In subsequent steps we will be deploying the Threater Enforce AMI from AWS Marketplace. Before doing that we must subscribe to the AMI and also record its AMI ID (this is required for entry into the CloudFormation template we will use later). Note that there is a different AMI ID for each supported region and for each host architecture (x86 vs ARM). When you subscribe to an AWS Marketplace product like our Threater Enforce software, your subscribing AWS account is effectively granted an internal AWS permission to be able to deploy the associated AMI identifier for the AWS Marketplace listing for deployment on AWS infrastructure. For more information about AWS Marketplace subscriptions and the steps and procedure you use to deploy services from within it, see the following link:

https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/launch-marketplace-console.html

The direct link for the AMI with Threater Enforce software pre-installed in the AWS Marketplace BYOL listing is:

https://aws.amazon.com/marketplace/pp/B08RMJGDD9

We currently support Enforce Marketplace deployments in the following regions:

- Asia Pacific (Melbourne) ap-southeast-4
- Asia Pacific (Mumbai) ap-south-1
- Asia Pacific (Sydney) ap-southeast-2
- Asia Pacific (Tokyo) ap-northeast-1
- Canada (Calgary) ca-west-1
- Canada (Central) ca-central-1
- Europe (Ireland) eu-west-1
- US East (N. Virginia) us-east-1
- US East (Ohio) us-east-2
- US West (N. California) us-west-1
- US West (Oregon) us-west-2

If your to-be-protected infrastructure is in a different region, please contact Threater directly for assistance so that we can add the AMI to your required region(s).

IMPORTANT: **Please note that the AWS Marketplace BYOL subscription does not include the required BYOL software subscription component:**

- It includes only the AWS infrastructure/hardware component of the instance deployment. For example, at the time this document was last updated, the recommended deployment for the Threater Enforce is a **c5.large (2 core / 4GB RAM)** instance type. When spun up in region **US East (N. Virginia)**, it costs **$0.085/hr** (on-demand pricing as of July 2025), which is a 24x7 annualized cost of **$744.60/yr** Threater, does not see any of that money - that goes entirely to AWS.

- Bandwidth charges are also billed separately to your account by AWS, and that money too goes entirely to AWS.
- The Threater Enforce subscription must be purchased separately and directly from Threater as a separate software subscription, which leverages an identical software subscription pricing model that we use for on-premise deployments.
- Without the BYOL subscription attached via the Threater Portal, the Enforce instance that you deploy will stay in a special **allow all** mode and will just blindly forward packets without performing any packet protection or logging. This means that even without a BYOL subscription, you will still be able to complete the example configuration, but the result will be that all traffic is allowed to pass in both directions. None of the traffic will be logged, and none of the traffic that we would have detected as malicious will be blocked until you install a valid subscription for each Enforce instance. The subscriptions must be obtained directly from Threater and attached to each deployed Enforce using the Threater Portal.

12. Automated AWS CloudFormation Templates

Now that we have described both VPCs in detail and are subscribed to the proper AMI we are ready to deploy working examples.

12.1. Why CloudFormation Templates?

Our deployment method opts for the use of JSON formatted AWS CloudFormation templates to perform the deployment. While it is certainly possible to deploy entirely via the AWS CLI or AWS console, deployment via CloudFormation templates is much more seamless and is by far the easiest way to build the example(s). This is because the templates manage all resource dependencies and you don't need to be an expert with the CLI or console to use them.

Any CloudFormation templates provided here are under the MIT license and you are fully within your rights to modify it in any way you may require. AWS CloudFormation templates are simple JSON documents and can be modified with any text editor.

You can learn more about AWS CloudFormation templates here:

https://aws.amazon.com/cloudformation/

Note that the templates only deploy one Enforce instance in the Service Provider VPC and one protected instance in the Service Consumer VPC. This is to keep things simpler and also to keep AWS infrastructure costs down. Adding either more protected instances or more Enforce instances to the templates is perfectly fine though.

We will be deploying the Service Provider VPC first.

12.2. Service Provider VPC Deployment

12.2.1. Service Provider VPC CloudFormation Template

The Service Provider VPC CloudFormation Template can be downloaded from this link:

https://threater-marketplace.s3.amazonaws.com/enforce-aws-cloudformation-gwlb-create-provider-vpc.json

12.2.2. Update CloudFormation Template Parameters

After the CloudFormation Template is downloaded, we will need to edit it and fill in some parameters that are unique to your AWS account. The parameters that require update are labeled with "CHANGEME" and the "description" field of each parameter thoroughly explains what needs to be filled in. Therefore, the purpose of each parameter is not discussed here. Note that all of the template parameters that need to be configured are at the top of the file and can be easily identified. All parameters pre-filled with "CHANGEME" must be populated correctly; they cannot be left as-is or empty.

12.2.3. Deploy Service Provider VPC via AWS CloudFormation

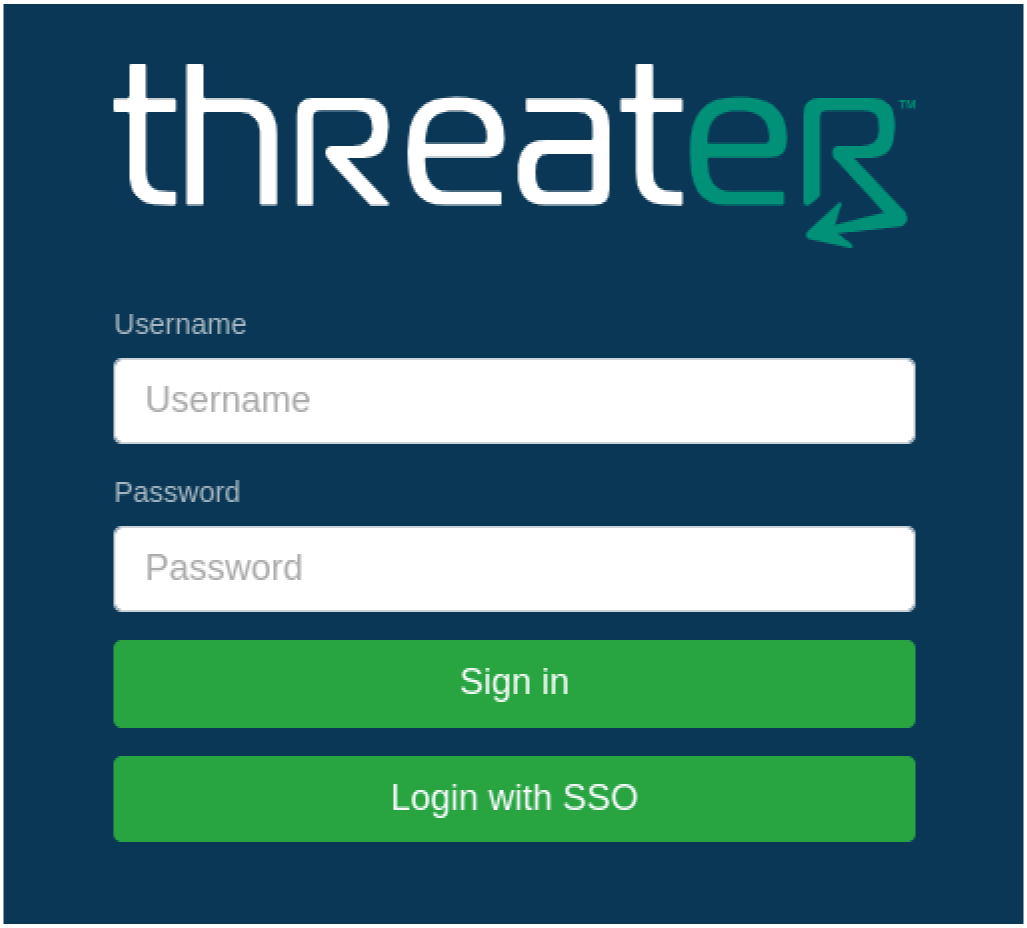
Once the template parameters are updated, we are ready to deploy the template. To deploy the template from the AWS console simply select: CloudFormation->Stacks->Create Stack and choose: "Upload a template file," then select the updated template file from your local disk, and simply follow the steps to complete the VPC deployment (you can name the CloudFormation stack anything you like). The entire VPC will be built for you in just a few minutes. It's really that easy!

12.2.4. Verify Threater Enforce Deployment

After AWS finishes building out all of the resources, we can verify that the Enforce instance is running by pointing a web browser to the public IP assigned to the Enforce instance's admin interface.

Note that we have utilized security group configurations in the deployed template to safely lock-down initial administration access solely to the public-facing IP address of the system you’re currently using. That means that generally you must access the Threater Enforce UI for initial configuration from that same system. If you are behind a NAT point at your current location, then be aware that any other system that NATs to the same public-facing IP will be able to connect as well (if they have the proper credentials to connect, of course).

The following login page should render in your browser after navigating the security warnings:



If the login page renders correctly, the Enforce deployment was successful!

12.3. Service Consumer VPC Deployment

12.3.1. Service Consumer VPC CloudFormation Template

The Service Consumer CloudFormation Template can be downloaded from this link:

https://threater-marketplace.s3.amazonaws.com/enforce-aws-cloudformation-gwlb-create-consumer-vpc.json

12.3.2. Template Parameters

The Service Consumer VPC CloudFormation Template is deployed in the same way as Service Provider VPC above with one important exception: configuration of the Gateway Load Balancer Endpoint parameter. This setting simply tells the VPC Endpoint in the Service Consumer VPC where to forward traffic and is effectively the glue that ties the VPC resources together. This value can be found under the Service Provider VPC->Endpoint services->vpce-svc-xxxxxxxxxxxxxxxxx->Service name. Simply copy this value into the parameter named "GatewayLoadBalancerEndpointServiceName." in the template. Update the remainder of the parameters listed at the top of the file and deploy the template.

After the deployment is complete, you should be able to connect via ssh to the public IP address of the protected t3a.nano instance deployed in the Service Provider VPC. The t3a.nano instance does not use a reserved AWS elastic IP so you will need to retrieve its value from the AWS console.

A successful ssh connection into the t3a.nano instance indicates that traffic is now flowing through the unlicensed Threater Enforce in the Service Provider VPC.

13. License and Configuration

To finish the deployment, you **MUST NOW** configure your instance before it will function properly.

13.1. Initial Threater Enforce Configuration

A typical initial Threater Enforce configuration flow is as follows:

1.	Login with the default administration user: admin and default password: admin
2.	Accept the displayed terms of service / end user license agreements (a one-time operation).
3.	On the next screen (another one-time operation), provide your login credentials for your pre-existing Threater Portal account. This is a key step that will register this Enforce instance with your Threater account. Once authenticated, the screen will close and you’ll be presented with the standard Threater Enforce "Welcome" page.
4.	Select System > Users to set up any required local users. At minimum, you should change the default admin password immediately. As with any production system and especially security controls, it is never wise to leave the default login credentials in place. Make sure you choose a strong password that you can remember or use a

	secure commercial password storage solution.
5.	Check your Network > Admin Interface and make any changes needed. In most scenarios no changes will be required here.
6.	Set DNS via Network > Admin Interface > DNS. By default the system will utilize Google's DNS for the primary and Cloudflare's for the secondary, but you can change these if you prefer others.
7.	Generally you'll stick with your AWS security group configurations for access, but you can decide on any extra admin access subnets if needed via Network > Access.
8.	Enter a unique hostname via Settings > General. Uniqueness is important so that if you later decide to leverage our powerful syslog export feature, individual installations will be able to be uniquely identified by their hostname.
9.	Set your desired time zone via Settings > Date & Time.
10.	If desired, set an NTP server of your choosing via Settings > Date & Time > NTP Servers. By default we configure Google's time services via time.google.com, but you can change this to whatever you'd like.

13.2. Register with Threater Portal

Now that the basic networking configuration for your Enforce instance is complete, open up a separate browser tab and connect to the Threater Portal platform at:

https://portal.threater.com

After logging in, navigate to the list of Enforce instances and associate an unassigned subscription with the new Enforce instance we just created above.

Within a few minutes of assigning your BYOL license, your new Enforce instance will start communicating with Threater's central systems to synchronize policies and lists. If you're an existing Threater customer, then this process is seamless. If you're a new Threater customer, we recommend you consult our available documentation on configuring and using our Portal.

In no time, you'll be up and running, with active protection in place for all network traffic flowing between your protected instances and the Internet, just like you're used to when running our solution entirely on-premise.

From that point forward, your lists and policies associated with your Enforce instance can largely be managed from the Portal, in exactly the same way you manage your on-premise installations.

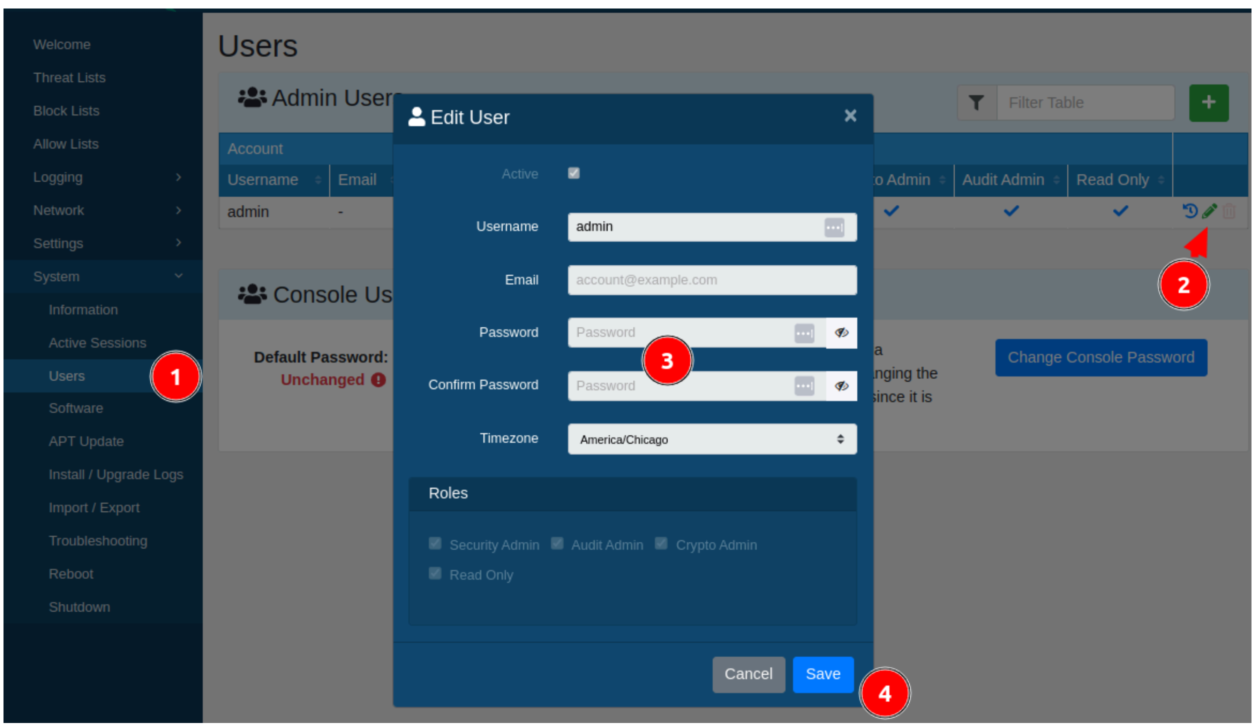
Note that if you do not fully assign a BYOL license and configure your system as described in the contents above, then your Enforce instance will afford you no protection whatsoever, nor will it log any information. It will simply pass ALL traffic in both directions and log nothing. Only after applying your BYOL license and configuring the system properly will bidirectional traffic be fully protected alongside comprehensive logging.

13.3. UI Access: Password Rotation

Best practice (as described earlier in this document) ensures that administration access should be locked down to specific known IPs through the security group configurations described, but it is still a good idea to make sure you change your system passwords regularly.

And on that note, we strongly urge customers to rotate your local UI passwords.

Like most modern systems, changing your password is as simple as logging into the UI, navigating to the user configuration, and then modifying the password. The flow graphic is:



The steps are straightforward:

1. Select System > Users
2. Click the green pencil edit icon
3. Enter and confirm the desired password when prompted

4. Click Save

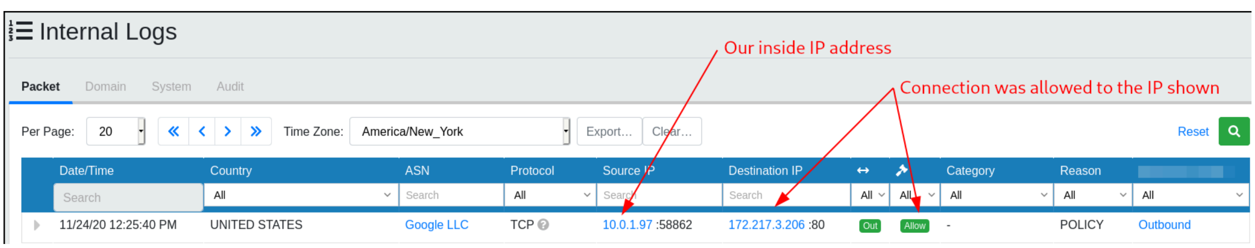
14. Test out the Deployment

Now that everything is configured we can log into the Enforce UI and investigate the Internal Logs to see the connection activity in more detail. To generate a logged connection let's first login via ssh to the protected t3a.nano instance and issue an outbound HTTP request to google.com:

$ curl google.com

The request should return 301: The document has moved response (which is what we expect in this case).

In the Enforce UI we can now investigate the Internal Logs to see the connection activity in more detail. As we might expect, we can see that the destination IP was indeed registered to a Google ASN, and the IP maps to a known location in the US. We see that our protected IP address that attempted to reach it was 10.0.1.97 (you will likely see a different address when you try it, as the addresses assigned to your protected instances will likely differ). We see that the TCP connection was allowed from source port 58862 (this port number will also likely differ in your case as it is generally randomly assigned by the test instance's operating system) to destination port 80 on the Google server, as it passed all relevant outbound policy criteria.



An obvious question is what might it have looked like if access to a known malicious site had attempted activity to or from that protected IP address? The short answer is that if it's known-malicious and on any threat or denied list attached to your policies of record, then it's going to be blocked. Here's a live example of an extraordinary amount of blocked, known-malicious traffic that we witnessed after manually opening up the outside subnet to all IPs and port access:

11/24/20 12:54:34 PM	KOREA REPUBLIC OF	Korea Telecom	TCP	125.132.73.28 :59637	10.0.1.97 :20883	In	Deny	• Spam • Scanner • Endpoint Exploits	DENIEDLIST	Inbound
Denied List Blocklist.de, CINS Army list, ET Compromised IPs **Threat List** Webroot										
11/24/20 12:54:23 PM	GERMANY	DigitalOcean, LLC	TCP	139.59.211.245 :32767	10.0.1.97 :8545	In	Deny	• Spam • Scanner • Endpoint Exploits	DENIEDLIST	Inbound
11/24/20 12:54:09 PM	AUSTRALIA	IP Volume inc	TCP	45.129.33.8 :51830	10.0.1.97 :32267	In	Deny	• Spam • Endpoint Exploits	DENIEDLIST	Inbound
11/24/20 12:53:52 PM	AUSTRALIA	IP Volume inc	TCP	45.129.33.49 :52538	10.0.1.97 :5036	In	Deny	• Spam • Scanner • Endpoint Exploits	DENIEDLIST	Inbound
Denied List Blocklist.de, DHS Information Sharing, ET Block IPs **Threat List** Webroot										
11/24/20 12:53:29 PM	UNITED STATES	MCI Communications Services, Inc. d/b/a Verizon Business	TCP	70.104.137.16 :41443	10.0.1.97 :23	In	Deny	• Spam • Scanner • Endpoint Exploits	DENIEDLIST	Inbound
11/24/20 12:53:28 PM	UNITED STATES	DigitalOcean, LLC	TCP	67.205.152.243 :54112	10.0.1.97 :80	In	Deny	• Scanner	THREATLIST	Inbound
Threat List Webroot										
11/24/20 12:53:17 PM	CHINA	Huawei Cloud Service data center	TCP	139.9.25.45 :55154	10.0.1.97 :9999	In	Deny	-	COUNTRY	Inbound
11/24/20 12:52:57 PM	AUSTRALIA	IP Volume inc	TCP	45.129.33.129 :41444	10.0.1.97 :3294	In	Deny	• Spam • Endpoint Exploits	DENIEDLIST	Inbound
11/24/20 12:52:54 PM	AUSTRALIA	IP Volume inc	TCP	45.129.33.168 :58162	10.0.1.97 :20107	In	Deny	• Spam • Endpoint Exploits	DENIEDLIST	Inbound

As you can see, the Threater Enforce software seamlessly blocks a ton of very bad stuff, leveraging our best-in-class third party threat list and denied list integrations. We've got several that come out-of-the-box, and a plethora of integrations for other proprietary threat intelligence feeds. For a full list of all of our supported integrations please visit our website.

You can see in the partial screenshot above that the malicious behavior we encountered was detected stemming from multiple countries, with multiple attributed threat and deny lists - attributable to a variety of proprietary and open source third party feed information. One of the great benefits of our patented platform is regardless of how much threat intelligence is used or how many blocks are being enforced, there will be no decrease in your network performance. That is, whether you are protecting against one or tens of millions of threats, both our on-premise and cloud-based Threater Enforce software will continue to operate at line rates with no additional latency.

Granted, this extraordinary amount of nefarious activity shown above can be at least somewhat mitigated through the use of proper AWS security group policies. But you can never lock things completely down via AWS security groups only. And as you can see by the nefarious activity above and specifically the targeted port numbers, very little of it is attributable to "web sources" via ports 80 and 443 - that is, web application firewalls and the like are not sufficient protection by themselves.

For example, if you're a typical business, you will have some number of public facing access points and/or open ports, and that's where you have risk. That's where you are vulnerable, especially when services (such as VPN services) must be kept open to large swaths of the world for large multi-national organizations or companies who do business with them. Those holes are what the bad guys will attempt to exploit. Just like they're trying to do here against our simple little test instance. And that's where Threater shines.

15. Data Encryption and Secure Communications

As was likely evident when you went through the deployment steps, there are no specific data encryption considerations for the deployment. Everything is automated within the Threater provided marketplace image and centrally managed by HTTPS connections which ensures on-the-fly standards-based public/private key sessions, as negotiated by an end user's browser.

Just like on-premise deployments, our cloud deployments encrypt all data transfer between the Threater Enforce software and the Threater Portal, utilizing HTTPS transactions via port 443. As such, standard techniques leveraging internally managed public/private keypairs are used, with standards-based negotiation for any and all access.

An example would be when the Enforce instance communicates to the Portal to determine if there is any new real-time threat intelligence information ready to be retrieved.

Note that the architecture is a 100% pull architecture (vs. a push architecture) with respect to the Threater Enforce software. That is, our Threater Portal has no way to directly reach Enforce deployments on its own. Instead, just like our on-premise deployments, the Threater Enforce software (even when running in AWS) always initiates secure communications to the Threater Portal at which point the Portal can provide new information, and never the other way around. Specifically, these communications are:

- Feed data, statistics and related metadata, and health checks are sent over TLS.
- All threat intelligence and related feed data is sent over TLS, encrypted and signed.
- All software updates scheduled by the end customer from the Threater Portal and subsequently delivered and transferred over TLS.
- All TLS connections to the Threater Portal are verified by certificates.
- And last but not least, each and every such communication uses TLS by way of HTTPS on port 443, always, without exception. Although often not applicable in the cloud, this fact is often quite useful for our on-premise customers when they choose to deploy us on the near-side of an existing on-premise next-generation firewall, as generally port 443 is already open, so no further external network configuration is typically required.

16. Maximizing Uptime

Our on-premise solutions are sometimes deployed by our customers as HA pairs. In addition, most of our on-premise hardware includes NICs that can be manually or automatically placed into a physical bypass mode, where even in the event of power failure or some other catastrophic hardware failure, traffic will pass through the system housing of the Enforce instance unabated. Unfortunately, in AWS, neither of these can happen as there is no direct hardware bypass capability or access in AWS network infrastructure.

To solve this problem AWS has designed its Gateway Load Balancers with HA capabilities in mind as they can: "distribute incoming traffic across your Amazon EC2 instances in a single Availability Zone or multiple Availability Zones." Thus, by intelligently distributing multiple Threater Enforce resources across multiple availability zones it is possible to design a service with high availability that is perhaps superior to that of on-prem hardware bypass.

Thus, it is indeed possible to protect against hardware faults in the cloud. However, creating HA services in AWS is beyond the scope of this document so please reference AWS documentation for further details on this topic. This link is a good place to start to learn more about AWS best practices for HA scenarios leveraging GWLB:

https://aws.amazon.com/elasticloadbalancing/gateway-load-balancer/

17. Managing Idle Flow Timeouts

By default, the Gateway Load Balancer will remove TCP flows from its internal flow table after 350 seconds of flow inactivity. This limitation often has repercussions for services that transit the Gateway Load Balancer. Thus, services that allow for idle durations longer than the Gateway Load Balancer default idle timeout will be problematic unless adjustments are made. When the Gateway Load Balancer TCP default idle timeouts are insufficient, either the Gateway Load Balancer TCP idle timeout can be increased or the underlying service can be configured with more frequent keepalives.

For UDP flows the same considerations apply except that the Gateway Load Balancer idle timeout is configured for 120 seconds and cannot be changed. More information on idle timeouts and adjustments can be found in the following link:

https://docs.aws.amazon.com/elasticloadbalancing/latest/gateway/gateway-load-balancers.html#idle-timeout

18. Monitoring Health

Monitoring health is simple. There are two recommended ways to check health:

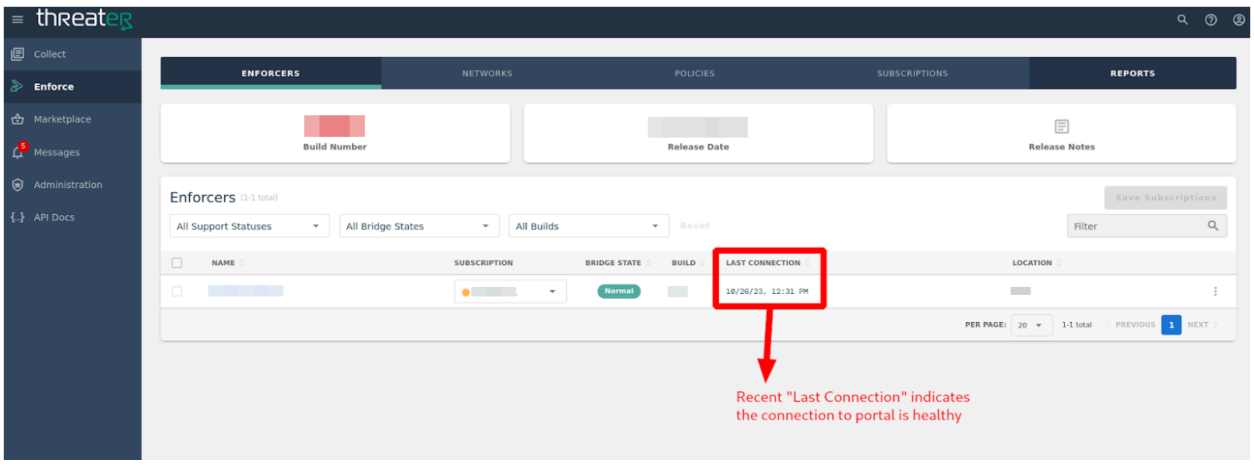
- via the Threater Portal
- via Enforce’s exportable RFC-compliant system logs

Both are valuable, and the use of one does not preclude the use of the other. In fact, it is our strong recommendation that customers leverage both paradigms, which we discuss in more detail below.

18.1. Checking Health via the Portal

This is the recommended way to verify health, as you can log in exclusively to the Threater Portal to view information about your Threater Enforce instances - whether on-premise, in the cloud, or both. This way you don't have to concern yourself with individual asset logins.

Threater Enforce software automatically and routinely connects to the Threater portal. This connection information can be found in your Portal views:



18.2. Checking Health via the Syslog Export Capability

Although leveraging the Portal's health monitoring tools are useful, we also highly recommend that you connect the Threater Enforce software's powerful RFC-compliant syslog export capability to one or more syslog sinks of your choosing. Not only does this allow you to monitor health, but it also allows you to monitor all detailed low level activity for all connections allowed and denied, which can be very important when analyzing network and security behavior. Additionally, it provides a means for you to backup critical historical security/log information routinely, in real-time.

Targeted systems for the logs exports could be a SIEM such as Splunk or IBM QRadar, or analysis tools like Gravwell, or even open-source syslog-ng. Configuring one or more of these types of exports will allow you to see real-time low level detail of the system activity, to include monitoring of the underlying software components. Detailed documentation on our RFC-compliant syslog export capability is available here:

https://threater-marketplace.s3.amazonaws.com/Threater+-+Syslog+Export.pdf

The nice thing about using a third party tool such as a SIEM is that you can leverage that tool to alert you and potentially take action on any behavior that you'd like. For example, many of our customers prefer to leverage SIEM and SIEM-like tools to initiate HA failover scenarios, as part of standard security stack best-practices.

19. Software Patches and Upgrades

Our on-premise and cloud-based software uses the exact same codebase. As such, whenever we release software, it is always, without fail, released for both.

One very nice benefit of our architecture is that the software patch and upgrade process is entirely managed by the Threater Portal.

This means that once a subscription is attached and Threater Enforce is securely communicating to our Portal, you are able to schedule software patches and upgrades directly from the Portal itself. You can even elect to do an immediate unscheduled software upgrade. In each case, the Portal will instruct the Threater Enforce software to download and install updates at a configurable time.

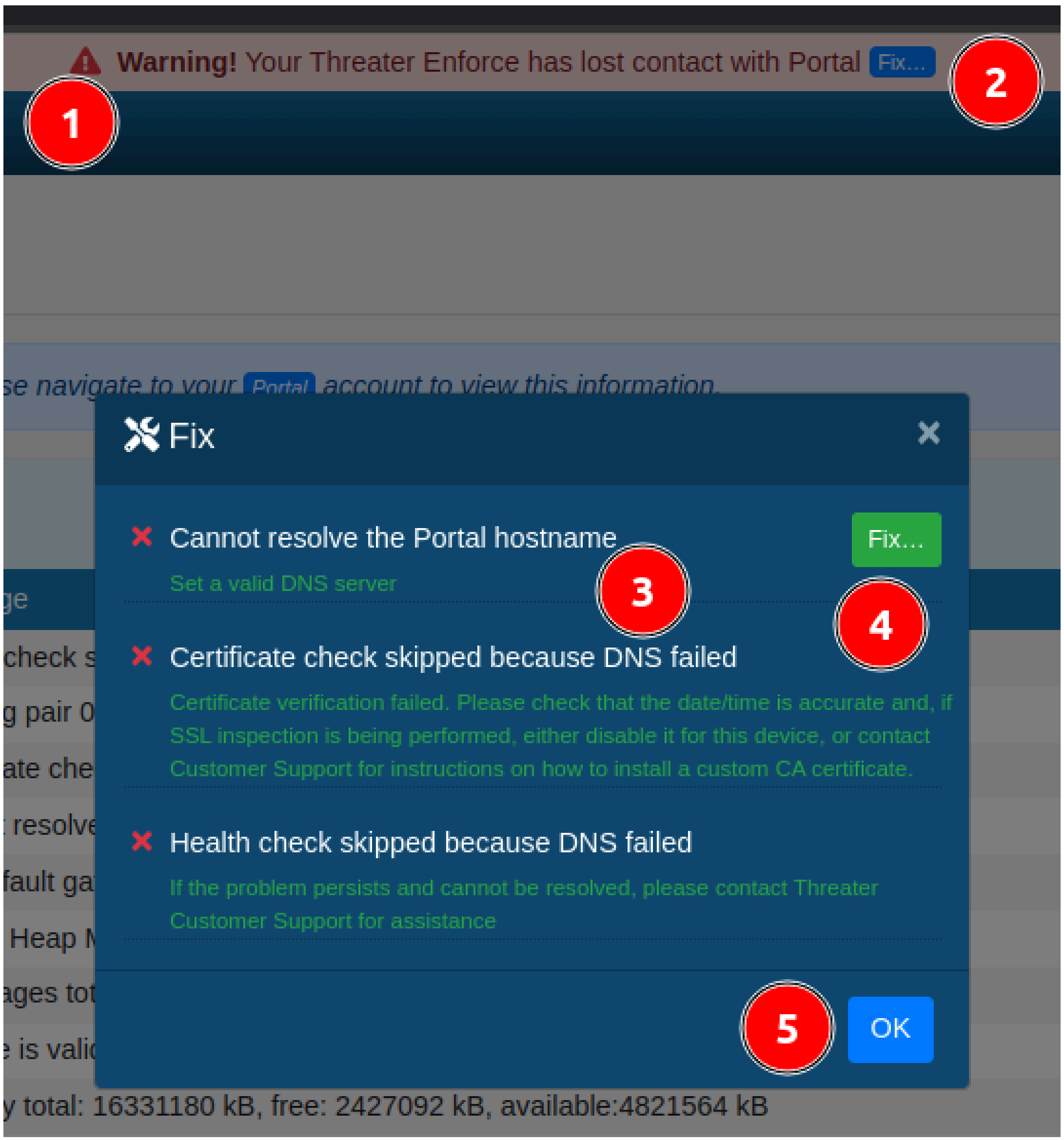
This is particularly beneficial for customers with both on-premise and cloud deployments - they can upgrade any or all of them centrally with no procedural differences whatsoever, all from the Threater Portal. In such cases, Threater Enforce software does not need to be directly accessed by the end user at all during the upgrade process. It's all handled for you, automatically. You just decide when you want the upgrade to occur. We generally recommend doing it in a short maintenance window. In general, it takes no more than about 5 to 10 minutes start-to-finish for a software upgrade to complete.

Additionally, when we release a new software version, we also update the AWS public AMI offering to match. That ensures that any new Threater customers deploying on AWS for the first time will always have the "latest-and-greatest" software available to them, just like they would get if they ordered a new on-premise system.

20. Handling Faults

We've gone the extra mile to make critical fault identification and recovery trivial, with straightforward instructions and "Fix It" style guides directly in the UI. We accomplish this with an in-your-face banner that will dock to the top of the UI whenever such an anomaly arises, enabling rapid remediation. This applies for absolutely everything that can go wrong relating to the deployment.

Here's an example where a user misconfigured their DNS causing a failure to connect to our Portal. That's bad, since it means that the Threater Enforce software wouldn't be able to retrieve updated threat intelligence. As you can see below, the system intelligently detects this and other catastrophic faults, and actually tells you about them so you don't have to guess. And then it helps you fix it, with a dialog similar to what is shown below:



The flow here becomes:

1.	The warning bar clearly tells you something is wrong.
2.	The blue "Fix" button takes you to the fix modal.
3.	The modal specifically tells you what it figured out - that it can't connect to the Threater Portal, and it's likely because of a DNS configuration problem.
4.	You click the green Fix button and it will take you to the appropriate configuration screen for fixing.
5.	And finally, you'll click OK.

Assuming you've fixed it properly with the guided fault remediation flows, the warning will disappear within seconds.

That same flow is the handling and remediation flow for all fault conditions with remediation pathways. The goal of these remediations is to quickly get a user back up and running whenever anything critical is detected.

Additionally, as mentioned in multiple places elsewhere in this deployment document, it is highly beneficial to export our RFC-compliant syslog data to one or more target systems, such as a SIEM, so that you have the ability to do detailed low-level fault analysis at your discretion, or perform historical analysis as needed. This is also highly desirable since SIEMs and related tools can be easily configured to alert on any number of criteria by way of things like email, SMS, and so on.

21. Next Steps and Cleaning Up

If you used this guide to construct a live, production deployment that you wish to use moving forward to deploy protected instances, then you can leave everything configured as-is. If not, any resources that we have just created can be cleanly deleted by using AWS CloudFormation deletion features. There is no need to remove each resource individually as AWS takes care of that for us.

22. Summary

We have now finished a complete example configuration within the confines of AWS entirely from scratch. We have:

- Presented a diagram that demonstrates how Threater Enforce is deployed in AWS Gateway Load Balancer mode.
- Deployed a functional VPC to demonstrate the capabilities of Threater Enforce
- Applied a Threater Enforce BYOL subscription via the Threater Portal
- Investigated log examples of the Threater Enforce allowing trustworthy content while blocking malicious content

Our existing customers already know how simple, smart, and scalable our patented technology is for their on-premise Threater Enforce deployments. And now, with Threater Enforce protecting native AWS infrastructure, we are pleased to offer the same simple, smart, and scalable capabilities providing a robust layered security architecture. Everywhere.

Learn more about us by visiting our website at:

https://threater.com