Threater Enforce in Google Cloud

November 02, 2023

1. Overview

This document provides end-to-end guidance for a typical deployment of Threater Enforce software leveraging a **Centralized Network Appliance** in Google Cloud, 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://storage.googleapis.com/threaterproduction/docs/Threater%20Enforce%20in%20Google%20Cloud.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 Google Cloud.
- 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 Google Cloud principles, including VPCs, firewalls, 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 Google Cloud deployment prerequisites are also important:

- An existing Google Cloud account with Google Cloud console access.
- Sufficient permission to perform deployments via GCP Deployment Manager.

Later in this document we will be deploying an example template using the gCloud CLI. If you don't already have this utility installed then now would be a good time to do so. The Google Cloud UI console cannot be used to perform the deployment. GCP gCloud Installation instructions can be found here:

https://cloud.google.com/sdk/docs/install

IMPORTANT: Keep in mind that the tools and instructions presented here only demonstrate one possible use-case of Threater Enforce in GCP; they are not a "turn-key" or "one size fits all" solution for securing your GCP 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
- **Enforcer:** 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:

- **ASN:** Autonomous System Number
- **BYOL:** Bring Your Own License
- **DHCP:** Dynamic Host Configuration Protocol
- **DPDK:** Data Plane Development Kit
- **EULA:** End User License Agreement
- **GCP:** Google Cloud Platform
- **GDPR:** General Data Protection Regulation
- **GVNIC:** Google Virtual Network Interface Card
- **HA:** High Availability
- **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:** 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 Enforcer protection point! That's the beauty of Threater Enforce running both in GCP 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 GCP (Centralized Network Appliance)

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 GCP image with Threater Enforce software pre-installed and a **BYOL subscription model.**
- Allows Threater patented technology to protect your GCP 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 Enforcer instances regardless of whether they are deployed on-premise, in the cloud (such as GCP), 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 GCP 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 GCP

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, in GCP we can leverage load balancers and next-hop routing features to route all inbound and outbound traffic via one or more Enforcer instances.

7. Integration with the Threater Portal

Our Threater Enforce software in GCP 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 GCP-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 GCP.

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.

8. Example Network Deployment

8.1. Choice of GCP Deployment Architecture

Within Google Cloud, there are several centralized network appliance architectures that support third-party appliances like an Enforcer. Of the currently supported GCP architectures we only support the **Load Balancer as the Next Hop** option. The reasons for choosing this option are:

- It allows the Threater Enforce to operate in a HA arrangement
- Supports the use of unmanaged instance groups

- Symmetric hashing and flow stickiness are supported (with use of two GCP internal load balancers)
- The Threater Enforce software can inspect inbound and outbound traffic unaltered
- It requires less complex routing

More information on GCP Centralized Network Appliances can be found here:

https://cloud.google.com/architecture/architecture-centralized-network-appliances-on-google-cloud

Note that deploying an Enforcer instance within a GCP managed instance group is not supported at this time. Only deployment into an unmanaged instance group is currently supported.

8.2. Important Google Cloud contraints

Before continuing further it is worth pointing out some constraints imposed by GCP that all third-party appliances operating in this mode must observe. These constraints have impacted the Threater Enforce software design requirements and will likely need to be considered in your GCP deployment as well.

8.2.1. Limitations on use of Multiple Network Interfaces

Google states that: ***"when using multiple network interfaces from an instance, each interface must attach to a subnet that is in a different VPC network."*** Since the Threater Enforce software requires an administration, inside, and outside interface, all GCP deployments utilizing Threater Enforce will be required to have at least three VPC networks. Further information on this limitation can be found here:

https://cloud.google.com/vpc/docs/create-use-multiple-interfaces

8.2.2. Regional Backend Service Network Interface Requirement

For regional service-based external load balancers: ***"Each backend service distributes traffic to the first network interface (nic0) of backend VMs."*** This means that the "outside" interface **MUST** be the first network interface device in the Enforcer's list of network devices. To accommodate this within our software we have elected to move the administration interface to nic2 such that nic0, nic1, and nic2 are outside, inside, and administration respectively. Unfortunately, this can have some undesirable side-effects because GCP is only capable of configuring host instances with a default route via DHCP on nic0. Therefore, the Threater Enforce software configures some policy-based routing to ensure that admin traffic uses the intended interface.

For more information reference the section "Regional backend services" in the following link:

https://cloud.google.com/load-balancing/docs/network/networklb-backend-service

8.2.3. External Load Balancer Forwarding Rules

Google Cloud external load balancer forwarding rules forward traffic to backend instances unmodified; no source or destination NAT operations are performed. Since neither GCP load balancers nor the Threater Enforce software perform NAT, there will be no default public/private IP address association. Thus, unless some custom routing is configured to map the inbound traffic to a specific instance, all inbound traffic will be dropped by GCP. To address this limitation Google suggests: ***"adding an entry in the local routing table to route traffic that's destined for the load balancer's IP address to the network interface controller (NIC)."*** Further information on this approach can be found here:

https://cloud.google.com/load-balancing/docs/network/udp-with-network-load-balancing

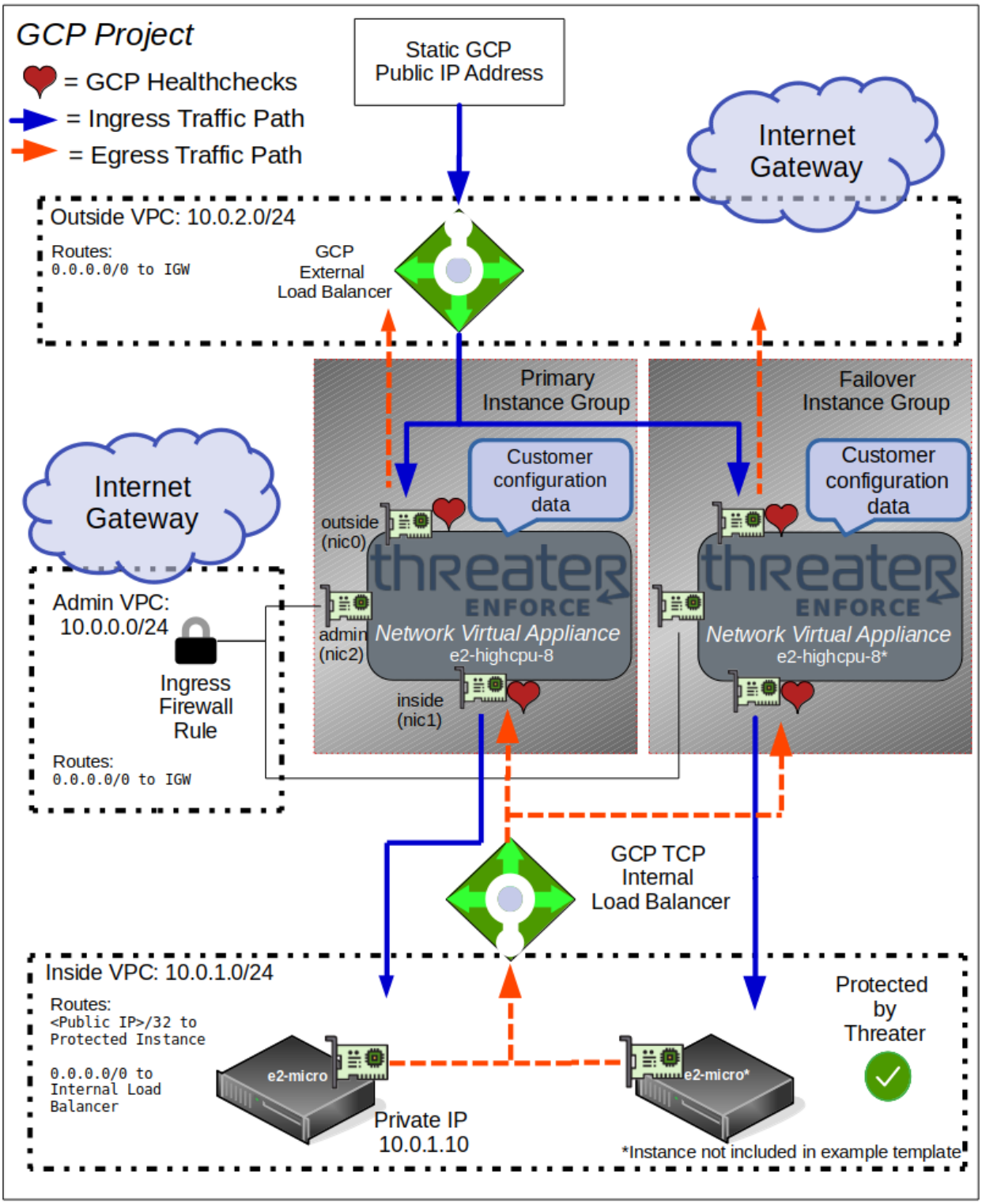
8.2.4. Network Interface Receive and Transmit Queues

Google Cloud network interface queue availability is highly dependent on the number vCPUs assigned to the instance. For systems that require more than one queue per interface (as does Threater Enforce) considerations must be taken to ensure that the deployed GCP VM instance type can deliver adequate queues per interface. More information on queue allocation is available here:

https://cloud.google.com/compute/docs/network-bandwidth#rx-tx

8.3. Example Network Deployment Diagrams

The following diagram shows a rather typical, generic Enforcer deployment into Google Cloud. You can decide to scale this "load balancer as a next hop" approach as your needs dictate, but in all deployments, the Enforcer(s) will need to be "enclosed" between internal and external load balancers as a backend service. Thus, the architecture will always look similar this:


GCP Project
= GCP Healthchecks
= Ingress Traffic Path
= Egress Traffic Path
Static GCP Public IP Address
Internet Gateway
Outside VPC: 10.0.2.0/24
Routes:
0.0.0.0/0 to IGW
GCP External Load Balancer
Primary Instance Group
Failover Instance Group
Customer configuration data
Customer configuration data
Internet Gateway
outside (nic0)
admin (nic2)
inside (nic1)
threater ENFORCE
Network Virtual Appliance
e2-highcpu-8
threater ENFORCE
Network Virtual Appliance
e2-highcpu-8*
Admin VPC: 10.0.0.0/24
Ingress Firewall Rule
Routes:
0.0.0.0/0 to IGW
GCP TCP Internal Load Balancer
Inside VPC: 10.0.1.0/24
Routes:
<Public IP>/32 to Protected Instance
0.0.0.0/0 to Internal Load Balancer
e2-micro
Private IP 10.0.1.10
e2-micro*
Protected by Threater
*Instance not included in example template

8.4. Load Balancers

The GCP next-hop network architecture requires one or more third-party appliances to be deployed within an instance group that is shared among multiple GCP load balancers. In this arrangement a minimum of two load balancers are required so that next hop routes can be implemented for both inbound and outbound traffic. The example deployment we provide implements this approach using one External GCP load balancer to handle inbound traffic and one Internal GCP Load Balancer to handle outbound traffic. Both load balancers share the same instance group that features one Enforcer instance.

Optionally, a second failover instance group can be added to create a redundant service as shown in the diagram. This instance group will receive traffic only if GCP determines that the Enforcer in the primary instance group is unhealthy.

Note that we have allocated only a single GCP (static) public facing IPv4 address in our example. However, if your requirements dictate protecting more than one public facing IPv4 address you can simply allocate the new static IPv4 address, configure an external load balancer/forwarding rule to use it, and set the load balancer backend to use the EXISTING instance group. In short, you can provide protection for multiple public IP addresses with one Enforcer instance group; you do not need a separate set of Enforcer resources for each public IP address you wish to protect with Threater.

Another alternative implementation (not presented in this document) would be to replace the external load balancer with an internal load balancer and leverage the symmetric 5-tuple hashing capabilities offered by GCP internal load balancers. This would allow distribution of flows among two or more Threater Enforce instances (while maintaining flow stickiness) and eliminate the need for a dedicated idle failover instance group when HA is required.

A detailed discussion regarding GCP load balancers is beyond the scope of this document. However, the following link provided by Google describes in more detail what the example we have provided has implemented:

https://cloud.google.com/load-balancing/docs/internal/ilb-next-hop-overview

8.5. Threater Enforce Network Interfaces and VPC Routing

Once the Enforcer is deployed, it will include three network interfaces: the standard administration interface, an inside interface, and an outside interface.

Your VPC subnetworks can have any network address range they need as long as they observe GCP requirements. For our example use case, we've chosen the address ranges shown in the above diagram. If your scenario warrants modifying these, you will need to adjust accordingly any local address references in the startup script metadata of the protected instances.

The Threater Enforce software only supports the use of the Google GVNIC network adapter type. Do not use any other adapter type for any of the Enforcer interfaces.

8.5.1. Administration Interface and Routing

As mentioned previously the Threater Enforce software automatically selects nic2 as it's administration interface upon startup. As such, this interface will need a routable connection to the Internet so that the device can be configured and Threater backend services can be reached. In our example we have simply assigned a GCP static public-facing IP address to this interface so that the Threater Enforce software can reach the Internet with minimal configuration.

The provided example also locks down access via a proper GCP firewall rule so that administration access is available only to individuals and systems who should have access. At a minimum you will need to open HTTP (port 443) for inbound connections so that the Enforcer can be managed via its user interface.

Alternatively, if your IT staff is knowledgeable about advanced GCP 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 GCP directly for assistance and training as needed.

Be aware that any changes to the Enforcer routing tables that reset or modify the default route may "lock-out" admin access to the system. Therefore, once the Enforcer is deployed, its route tables should not be modified manually nor should route changes be induced indirectly via the Google Guest Agent.

8.5.2. Inside Interface and Routing

The Enforcer inside interface will connect to the inside subnet and serve one or more protected instances. Inbound traffic that is not blocked by the Enforcer will be routed according to the route tables of the inside VPC. As noted earlier in this document, inbound traffic arriving at the inside subnet from the Enforcer will retain its public source and destination IP addresses. Thus, to properly route inbound traffic to the protected instance the example deployment configures the following routing:

- Adds the external load balancer's front-end IP address to the local routing table of the protected instance (added via the protected instance startup script).
- Adds a route to the inside VPC route table to deliver all traffic destined to the external load balancer's public IP address to the protected instance.

In addition to handling inbound traffic (connections initiated externally) the example also configures the protected instance and the inside VPC network to support outbound connections initiated by the protected instance. Without these additions, the protected instance would use the GCP provided default route and forward Internet traffic to the default gateway where it would be dropped. To support outbound connections via the Enforcer the following additional routes are configured:

- Add entries to the POSTROUTING chain of the netfilter nat table on the protected instance to source NAT all outbound traffic to the IP address of the load balancer front end.
- Adds a default route (0.0.0.0/0) to the inside VPC route table to deliver all Internet traffic to the inside interface of the Threater Enforce. This new default route will override the existing GCP default route through the use of a lower route metric.

The above customizations to the routing tables of the protected instance and the inside VPC network allow the protected instance to support both inbound and outbound connections.

8.5.3. Outside VPC Interface and Routing

The routing requirements of the outside VPC network are quite simple. In fact, the load balancer forwarding rules ensure that all inbound traffic is routed to the Enforcer outside interface while the existing GCP default route ensures that all outbound traffic leaving the outside interface is routed to the VPC default Internet Gateway. Therefore, there are no custom routes required in the example deployment we provide.

8.5.4. Transmit and Receive Queue Requirements

The Threater Enforce software requires an extra transmit and receive queue on the inside and outside interfaces to handle healthcheck messages properly. Thus, the required minimum queue counts per interface are as follows.

- Outside Interface: (nic0): 2
- Inside Interface: (nic1): 2
- Admin Interface: (nic2): 1

8.6. Health Probes

8.6.1. Health Probe Path and Protocol

Google Cloud will issue health probe checks on behalf of deployed load balancers and on configurable intervals to ensure that all Enforcer instances in the backend instance groups are operating correctly and can accept new flows. Note that these probe messages are sent only to

the inside and outside interfaces -- not the admin interface. When the Threater Enforce software is operating normally (ready to accept flows), it will respond as healthy and the load balancer will be free to send it new flows for inspection. Important: The Threater Enforce software only responds to HTTP (port 80) health probe requests at the path:

/api/v1/healthcheck/gwlb

Threater Enforce software will not respond to any other health probe configuration so it is absolutely imperative that the health checks are configured correctly in your deployment. Ensure that all load balancer backend services are reporting healthy before attempting to evaluate or test traffic flows via the Enforcer.

8.6.2. Health Probe Firewall Requirements

Google Cloud healthcheck messages are blocked by the default VPC network firewall rules. Therefore, it is the responsibility of the end user to allow these connections when needed. It is important to recognize that healthchecks will be issued by GCP for **ALL** internal and external load balancers. Further, healthchecks can arrive from a multitude of source IP ranges that depend on the type of load balancer. The example deployment we provide clearly demonstrates how to configure healthcheck firewall rules for the Enforcer inside and outside interfaces. The following link explains in more detail how to configure healthchecks in Google Cloud:

https://cloud.google.com/load-balancing/docs/health-check-concepts?hl=en

8.7. Traffic Ingress Firewall Configurations

The provided example creates two GCP Ingress Firewall rules to support traffic flow through the Enforcer:

- outside-firewall: Limits external traffic that can enter the outside VPC network. By default this rule limits access to one public IPv4 address of your choosing to keep things secure and locked-down. However, once the Enforcer is deployed and configured you will probably want to modify this rule to meet your specific needs. The use of 0.0.0.0/0 would be reasonable here to allow the Enforcer to evaluate all inbound traffic.
- inside-firewall: Allows traffic with non-private IPv4 source addresses to enter the inside network. This rule can be enhanced as-needed but by default it leaves access open for simplicity.

8.8. Serial Port Access

Google Cloud offers serial port access to virtual machines that choose to enable it. In our example we have gone ahead and enabled serial port access for you. If for some reason

connectivity is lost to either the Enforcer or the protected instance you can easily connect and resolve issues using the "Connect to Serial Console " feature in the Google Cloud console. The serial port login credentials are different from other default Threater Enforce credentials and can be found in the template startup script metadata for each virtual machine.

9. 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 GCP Deployment Manager template we provide in a subsequent section, deployment will take anywhere from 15 to 30 minutes, total.

10. Obtaining the Enforcer Image

We have publicly shared the latest image of the Enforcer at the following Google Cloud URI:

https://www.googleapis.com/compute/v1/projects/pubdocs/global/images/enforce

Any GCP authenticated user should have permission to access it and freely deploy it. The example deployment that we provide references this URI so that would be a good place to see how it is referenced. There should be no need to download or manually transfer the image.

IMPORTANT: **Please note that the image we share with you will have all of the Threater Enforce software pre-installed and will be ready for final configuration. However, it does not include the required BYOL software subscription component:**

- It includes only the GCP 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 **e2-highcpu-8** instance type. When spun up in region **us-east1 (South Carolina)**, it costs **$0.199/hr** (on-demand pricing as of October 2023), which is a 24x7 annualized cost of **$1745.40/yr** Threater, does not see any of that money - that goes entirely to Google Cloud.
- Bandwidth charges are also billed separately to your account by GCP, and that money too goes entirely to Google Cloud.
- 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 Enforcer 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 Enforcer instance. The subscriptions must be obtained directly from Threater and attached to each deployed Enforcer using the Threater portal.

11. Automated GCP Deployment Manager Templates

Now that we have described the deployment in detail and have access to the public VM Image we are ready to deploy a working example.

11.1. Why Deployment Manager Templates?

Our deployment method opts for the use of YAML format GCP Deployment Manager templates to perform the deployment. While it is certainly possible to deploy entirely via the gCloud CLI or the UI console, deployment using GCP templates is much more seamless and is by far the easiest way to build the example. This is because the templates manage all resource dependencies and you don't need to be an expert with gCloud or GCP console to use them.

You can learn more about GCP Deployment Manager templates here:

https://cloud.google.com/deployment-manager/docs/configuration/templates/create-basic-template

Note that the template only deploys one Enforcer instance in the primary unmanaged instance group and one protected instance attached to the inside VPC. This is to keep things simpler and also to keep GCP infrastructure costs down. Adding either more protected instances or a failover Enforcer instance to the templates is perfectly fine though.

11.2. The Deployment Manager Template

The Deployment Manager Template can be downloaded from this link:

https://storage.googleapis.com/threaterproduction/templates/enforcer-gcp-template-create-project.jinja

The contents of the file are a standard GCP YAML deployment configuration format with some jinja templating. The inclusion of jinja is a supported GCP feature that allows for parameterizing of the configuration so that modifying the template file is not necessary. Therefore, no changes to the file contents should be necessary and it can be used "as-is". The required deployment parameters are as follows:

- region -- GCP deployment region for various deployment resources

- zone -- zone where VM instance will be deployed (must be in selected region)
- myip -- Your public IPv4 address that will connect to the Enforcer instance (for firewall rule)

11.2.1. Deploy

Once we have a local copy of the template, we are ready to perform the deployment as follows using the gCloud CLI:

First, ensure default project is configured:

gcloud config set project <gcp_project_name>

Then to perform the deployment:

**gcloud deployment-manager deployments create <deployment name> **
**--template enforcer-gcp-template-create-project.jinja **
--properties region:<target region>,zone:<target zone>,myip:<your public IPv4 address>

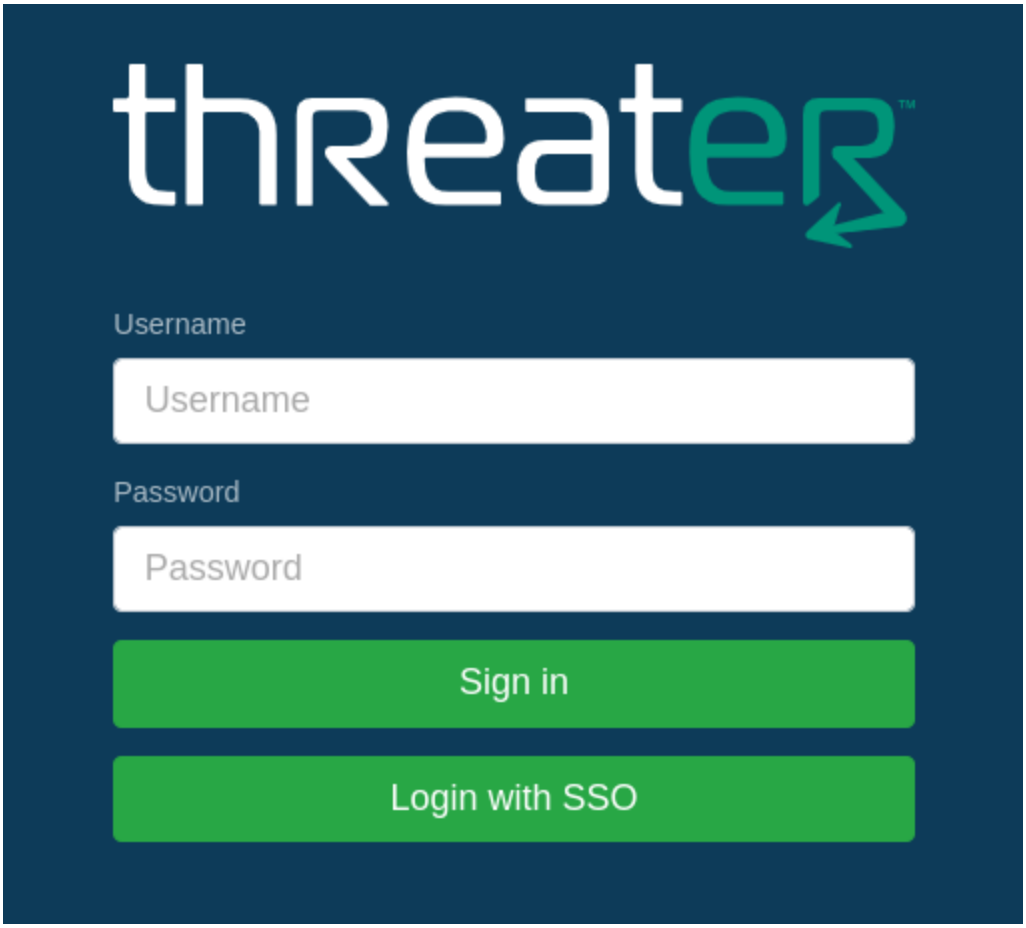
You can choose any "deployment name" you prefer that meets GCP requirements. The entire deployment will be built for you in just a few minutes.

11.2.2. Verify Threater Enforce Login

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

Note that we have utilized firewall 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 Enforcer deployment was successful!

12. License and Configuration

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

12.1. 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 Enforcer 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 GCP firewall 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.

12.2. Register with Threater portal

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

https://portal.threater.com

After logging in, select Subscriptions in the left-hand navigation, and assign an unassigned subscription to the new Enforcer instance we just created above.

Within a few minutes of assigning your BYOL license, your new Enforcer 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 Enforcer instance can largely be managed from the portal, in exactly the same way you manage your on-premise installations.

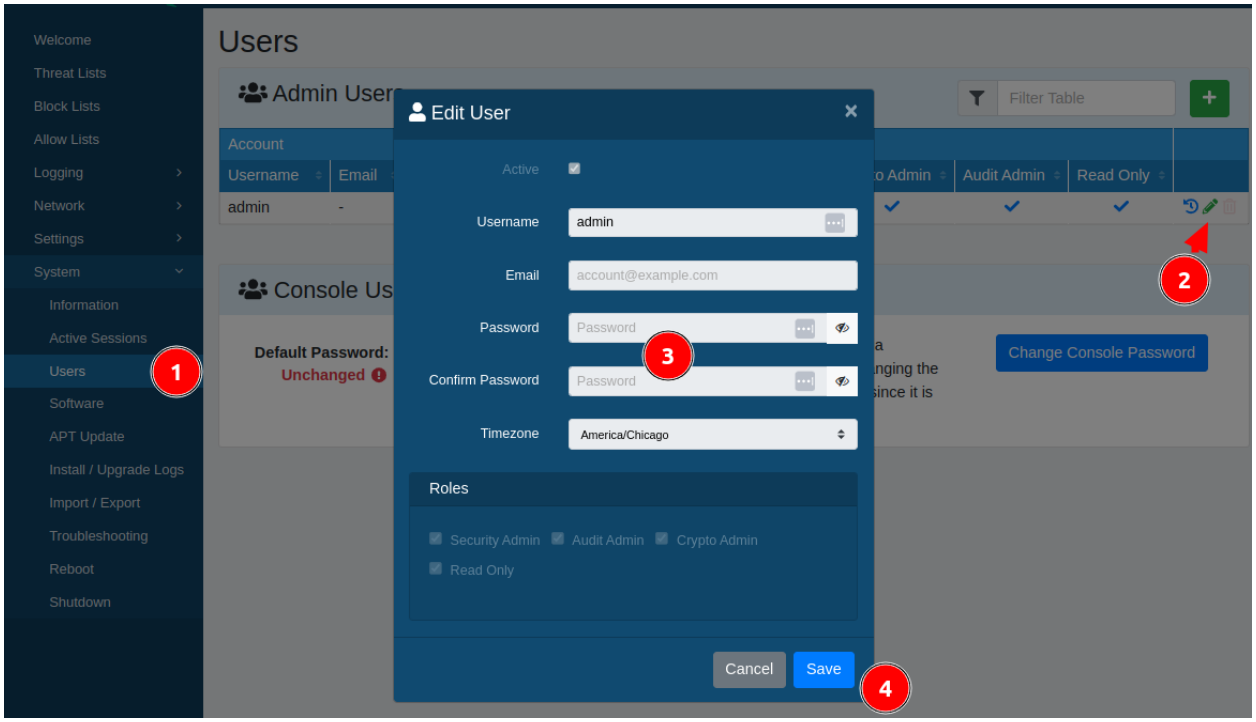
Note that if you do not fully assign a BYOL license and configure your system as described in the contents above, then your Enforcer instance will afford you no protection whatsoever, nor will it log any information. It will simply pass ALL traffic in both directions and log nothing. Only after applying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

12.3. 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 firewall 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

13. Test out the Deployment

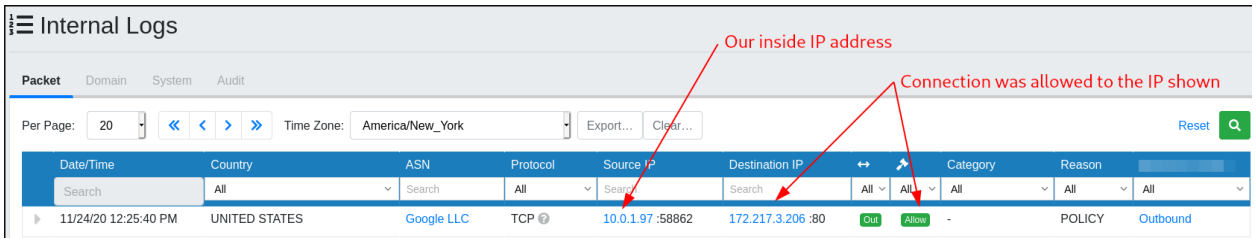
Now that everything is configured we can log into the Enforcer's UI and investigate the Internal Logs to see the connection activity in more detail. To generate a logged connection let's first login via ssh to the protected e2-micro instance and issue an outbound HTTP request to google.com:

$ curl google.com

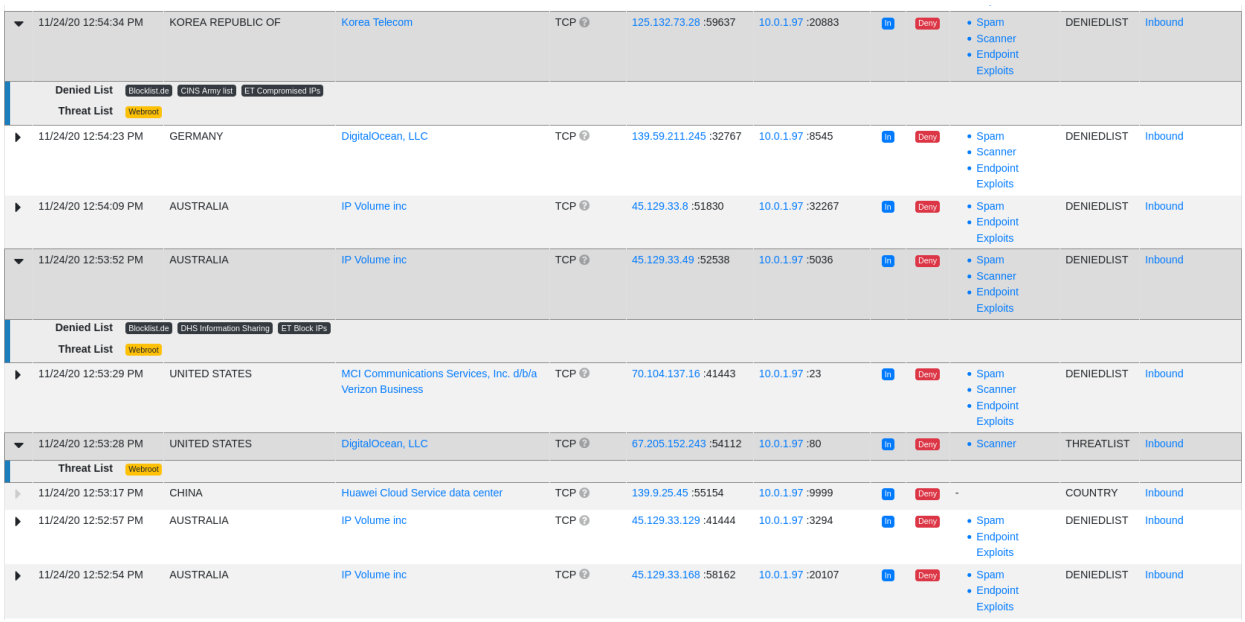
The request should return 301: The document has moved response (which is what we expect in this case).

In the Enforcer's 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:



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 GCP firewall policies. But you can never lock things completely down via GCP firewalls 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.

14. 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 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 Enforcer 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 Enforcer deployments on its own. Instead, just like our on-premise deployments, the Threater Enforce software (even when running in GCP) 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.

15. 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 Enforcer unabated. Unfortunately, in GCP, neither of these can happen as there is no direct hardware bypass capability or access in GCP network infrastructure.

However, by deploying Threater Enforce using the GCP failover architecture we have proposed in the provided example it is possible to realize HA capabilities similar to that of on-prem hardware. Please reference Google Cloud literature on "high availability" to learn how to most effectively deploy redundant HA services.

16. Monitoring Health

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

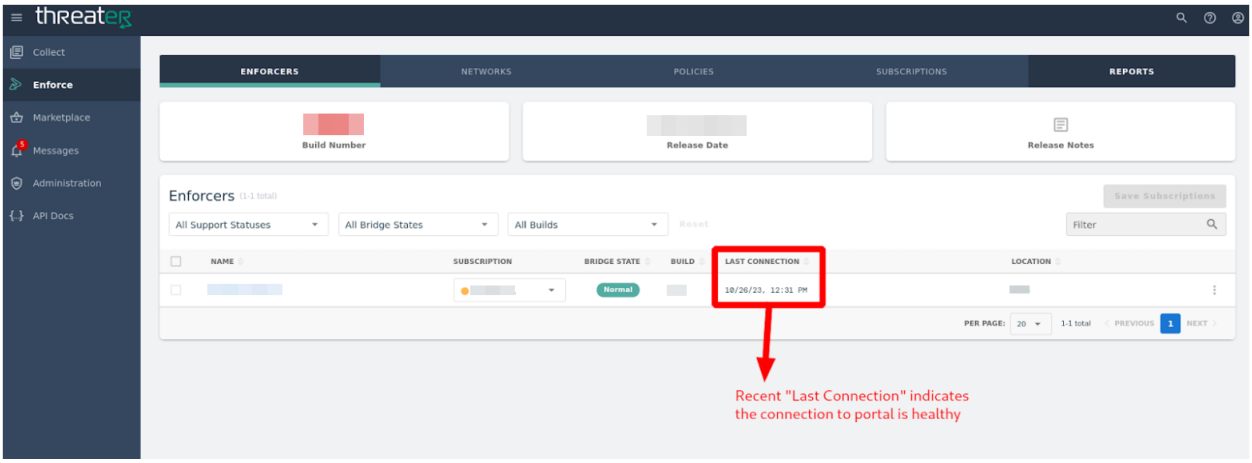
- via the Threater portal
- via Enforcer’s exportable RFC-compliant system logs

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.

16.1. 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 Enforcer 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:



16.2. 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.

17. Backup And Recovery

Threater Enforce deployments in GCP, just like our on-premise deployments, have a comprehensive built-in ability to backup the entirety of the configuration. The backup can be downloaded as a JSON file. It can then be re-imported later for recovery purposes.

Once logged into the Enforcer's UI as previously described in this document, you can use the following flow to backup and restore the configuration:

The referenced numerical steps become:

1.	Select System > Import / Export.
2.	Select Export.
3.	Select either the Copy button to copy the configuration contents to the clipboard, or the Export button to export it to a persistent JSON file.
4.	When you are ready to restore a backup configuration to a particular system, choose the Import tab and point it at your previously exported configuration file. That's it!

Note that when importing a backup configuration for restore, it is instantaneously applied. There is no wait time or need to reboot.

This technique is also useful if you find that you had a need to fully terminate the current instance (or if it suffered a hardware failure - failures occur sometimes in the cloud too!), as opposed to stopping it for an eventual restart. When you relaunch a previously terminated instance from GCP Marketplace, it will come up as a brand new instance with no configuration. You could of course reconfigure everything manually as described earlier in this document, but if

you had a backup JSON configuration, it is trivial to reload it. After importing the configuration, the Threater Enforce software will then be back in the exact same configuration state as when the backup was made, and will immediately start functioning just as you left it.

Note that one of the nice benefits of our architecture is that the only thing to be concerned about for backup and recovery with our architecture is this configuration itself! Furthermore, since it is a simple JSON file, it is trivial to maintain in any system of your choosing, following whatever best practices that your corporation employs. There is very little post-deployment Threater Enforce configuration since most of the functionality is managed exclusively by the Threater portal which means that it isn't necessary to worry about routine, automated configuration backups.

18. 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.

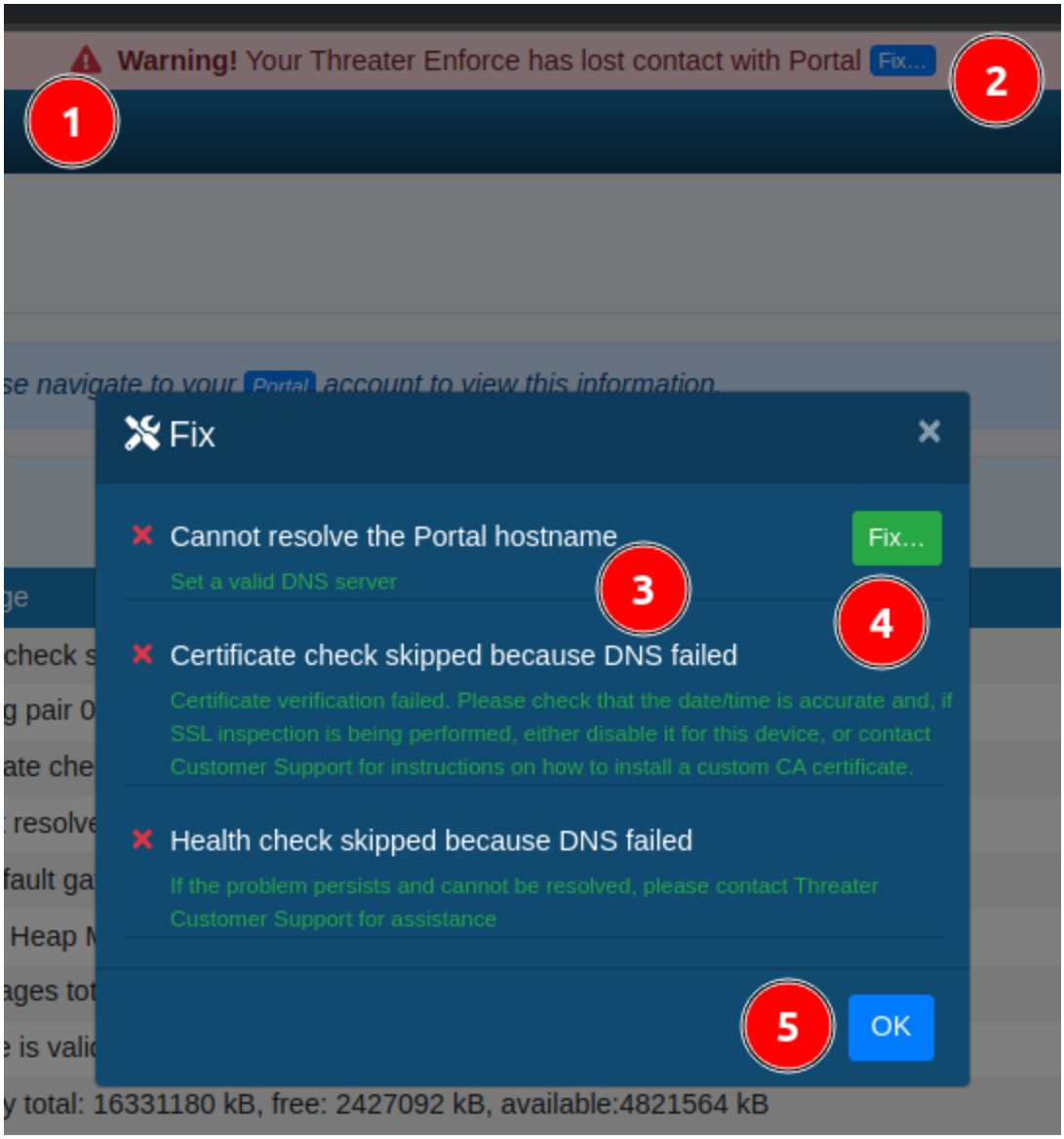
This is particularly beneficial for customers with both on-premise and cloud deployments - they can upgrade any or all of them centrally with no procedural differences whatsoever, all from the Threater portal. In such cases, Threater Enforce software does not need to be directly accessed by the end user 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 GCP public image offering to match. That ensures that any new Threater customers deploying on GCP 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.

19. 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.

20. 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 completely deleted by deleting the associated Deployment Manager Template(s) from the GCP console. There is no need to remove each resource individually as GCP takes care of that for us.

21. Summary

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

- Presented a network diagram that demonstrates how Threater Enforce works as a GCP Centralized Network Appliance.
- Used a GCP Deployment Manager template to realize a fully functional deployment based on the proposed network diagram.
- Applied a Threater Enforce BYOL subscription via the Threater portal
- Evaluated Threater Enforce allowing traffic considered "trustworthy"
- Investigated log examples of 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 GCP 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