Vectra NDR for Cloud

Reference Architectures

Version: Feb 20, 2025

Table of Contents

Overview

Vectra's NDR for Cloud (NDRC), provides a unique opportunity to provide greater abilities to capture network traffic throughout public cloud platforms. The purpose of this guide is to provide the reader with a clear understanding of the design and deployment best practices for component placement. Each Cloud environment, whether it be AWS, Azure or GCP will share common design and deployment best practices. In each case where applicable we will also share the nuances of each environment. A table of resources is available below to help you find additional material.

KB Article Link or Index Category	Description
Product Documentation Index	Vectra's main index that tracks formal product documentation. Use the search box to find more KB articles.
NDR for Cloud Gigamon Deployment Guide for Azure	Vectra's deployment guide for NDRC for Azure
NDR for Cloud Gigamon Deployment Guide for AWS	Vectra's deployment guide for NDRC for AWS
GigaVUE 6.9 Documentation Publicly available Gigamon documentation	This online documentation provides the complete GigaVUE 6.9 documentation set in a single, searchable interface. This site is easier to navigate and is best for interactive use.
GigaVUE 6.9 Guides Publicly available Gigamon documentation	Downloadable versions of GigaVUE 6.9 documentation.

Use Cases

In each deployment environment, capturing network traffic has its own set of unique issues. Here are just a few examples:d

Amazon Web Services (AWS)

AWS provides a mature native ability to capture network traffic and duplicate this traffic toward another security capture device, such as our virtual sensors.

Having such native capabilities is very positive, as there are no third-party components to deploy nor maintain. However, the issue is financial in a large deployment. The pricing is based by each ENI (Elastic Network Interface) and each ENI is priced at $0.018 per hour within European ($0.015 for US) regions. That equates to $12.96 per ENI per month. When multiply that figure by the total number of Port Mirrors required, the costs soon add up.

Also, not all EC2 instances are supported. There are still a few of the common tiny and medium sized EC2 instances that are not able to take advantage of the VPC Traffic Mirroring capability.

Microsoft Azure

Azure has no native capabilities in this environment, so the third-party solutions are the only available methods to capture network traffic from virtual instances.

Google Cloud Platform (GCP)

Like AWS, GCP has its own native capabilities with regards to network traffic capture. Although the difference here is GCP charges per GB traversing the network load balancer. In a high data volume environment, it can be difficult to predict the costs.

Native Functionality Pricing Considerations

AWS

As stated previously, AWS provided has its own packet forwarding capability in VPC Traffic Mirroring. This capability functions by configuring AWS to first filter the traffic desired, then where you want the traffic to go and then from where the traffic will be mirrored from.

This is a fairly simple process, however there are a few limitations. This is not natively an automated process however; lambda scripts can be obtained to provide automated traffic mirror creation. Without, each traffic mirror must be configured independently. You will pay for each mirror configured, whether the EC2 instance is running or not. So careful inventory of your cloud resources and policies are necessary.

Another limitation to AWS Traffic Mirroring is that not all EC2 instances are supported. For those running unsupported instances, there is no native capability to mirror traffic.

Lastly there is the cost. At $12.96 per month per ENI, a typical environment of 500 EC2 instances would incur costs of $6,480 per month.

Google Cloud Platform (GCP)

GCP also provides its own native port mirroring capability in Packet Mirroring. This capability is priced a little differently, in that it is priced based on inbound data processed by load balancer for Packet Mirroring. Currently priced at $0.008 per GB, it can be difficult to forecast charges and budget appropriately.

Vectra NDR for Cloud Solution Components

NDR for Cloud is a bundle of Vectra's NDR and Gigamon's Cloud Suite products. All NDR for Cloud deployments will typically begin with deployment of Vectra NDR or NDR for Cloud will be an add-on to an existing Vectra NDR deployment to bring visibility to cloud deployed resources. For deployment of Vectra NDR, please see the Product Documentation Index. This product sets about overcoming the limitations of native traffic mirroring capabilities at a predictable price point for customers. The various Gigamon components of the overall solution are described below.

Fabric Manager (FM)

A web-based fabric management interface that provides a single pane of glass visibility and management of the traffic that forms the GigaVUE Cloud Suite. You will require one of these per deployment. In a large enterprise where you may have multiple Accounts or Tenants in multiple regions isolated from one another, a Fabric Manager per region or account where there is no interconnectivity between them will be necessary.

The Fabric Manager manages the configuration of the V Series Nodes and UCT-V controllers in your deployment. The Fabric Manager and UCT-V agents are deployed manually. All the other components are deployed as part of policy creation and operation through the Fabric Manager.

V Series Nodes

These virtual appliances accept the captured traffic from the UCT-V agents, process and distribute the traffic to the Vectra vSensors. The traffic can be processed in different intelligent ways (license dependant). In large enterprise deployments an example could be to send UCT-V traffic from a specified VNet to a specific vSensor within a centralized group of vSensors. This is to maintain the frequently used hub-spoke model for ease of ongoing operations and maintain session state between sensors.

Once in production and the appropriate traffic policies are in place. The Fabric Manager will monitor the number of UCT-V agents deployed and auto-scale the number of V Series appliances that are required to handle the load. By default, one V Series Node per 100 UCT-V's monitored will be deployed.

UCT-V Controller

The UCT-V Controller (previously known as G-vTAP Controller) holds the filtering policies for the UCT-Vs (agents) and orchestrates the flow of mirrored traffic to the V Series Nodes. The Fabric Manager creates and pushes the polices to the UCT-V Controller. The UCT-V agent will then communicate back to the controller for the latest filtering policy to apply. A UCT-V Controller can only manage UCT-Vs that have the same version.

Gigamon UCT-V (Agent)

These agents are deployed to each virtual instance (VM) you wish to monitor. Once deployed they run as a background service or process and create an additional virtual capture interface. The agent will then apply the filtering policy and forward the captured traffic to the V Series Node via the new capture interface.

Agents are available for Microsoft Windows and various Linux distributions.

Please see Supported Operating Systems for UCT-V for more detail.

Operating System	Supported Versions
Ubuntu / Debian	Versions 16.04 through 24.04
CentOS	Versions 7.5 through 9.0
RHEL	Versions 7.5 through 9.4
Windows Server	Versions 2012 through 2022
Rocky OS	Versions 8.4 and above

For large cloud-based infrastructures, deployment of the UCT-V agent will be an important consideration in the success of the deployment.

Where appropriate, the UCT-V agent can be deployed in several methods: manually, Active Directory group policy with both Terraform and Ansible scripts. The scripts will be made available from Vectra support.

Solution Deployment Recommendations

Let's take a deeper look into the possible deployment scenarios. Each case will share some similarities but will highlight key points of interest of each environment.

Azure Deployment

Prerequisites

When considering deployment within Azure, it is important to note the permissions required. First of all, there is a choice of two methods of authentication and integration between the Gigamon and Azure. The first method is Managed Identities, and the second Application ID. Managed Identities are the recommended option.

The next step is to understand the Permissions and Privileges required for the chosen authentication and integration method. The link above describes the permissions required and granted for a successful deployment. You will notice the solution requires the ability to Read, Write, and Deleted instances within the deployed VNet. For autoscaling to function properly this level of permissions is necessary. To minimize the risks associated with this, we would recommend creating a separate Resources Group and subsequent subnet for the Gigamon components. This way, the impact of the permissions are negligible.

Azure Deployment Example (Simple)

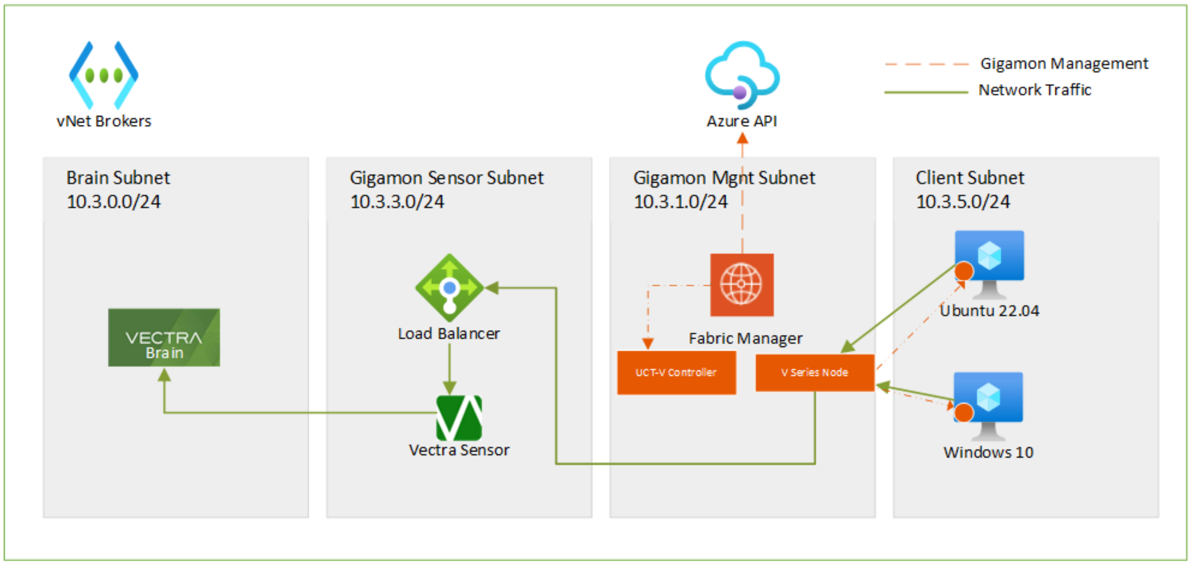


Figure 1 - Azure Example Deployment (simple)

In this example the Vectra Sensor and Azure Load balancer are placed in their own Resource Group and subnet. However, they can just as easily be placed within the same subnet along with the Gigamon solution.

As for the Vectra Brain, this can be located in the most desirable location, cloud or local data center.

Spanning Multiple Networks (large enterprise)

Larger environments can also be accommodated from a single deployment. Each subsequent VNet will be peered to the VNet and subnet containing the deployment. If the appropriate read permissions and peering exists between the VNets, a multi-VNet or multi-subscription environment will function correctly.

Within Azure, there are transit costs between VNets that need to be considered. These costs can become significant as they are based on the volume of traffic traversing the VNets. In this scenario the deployment will require some changes.

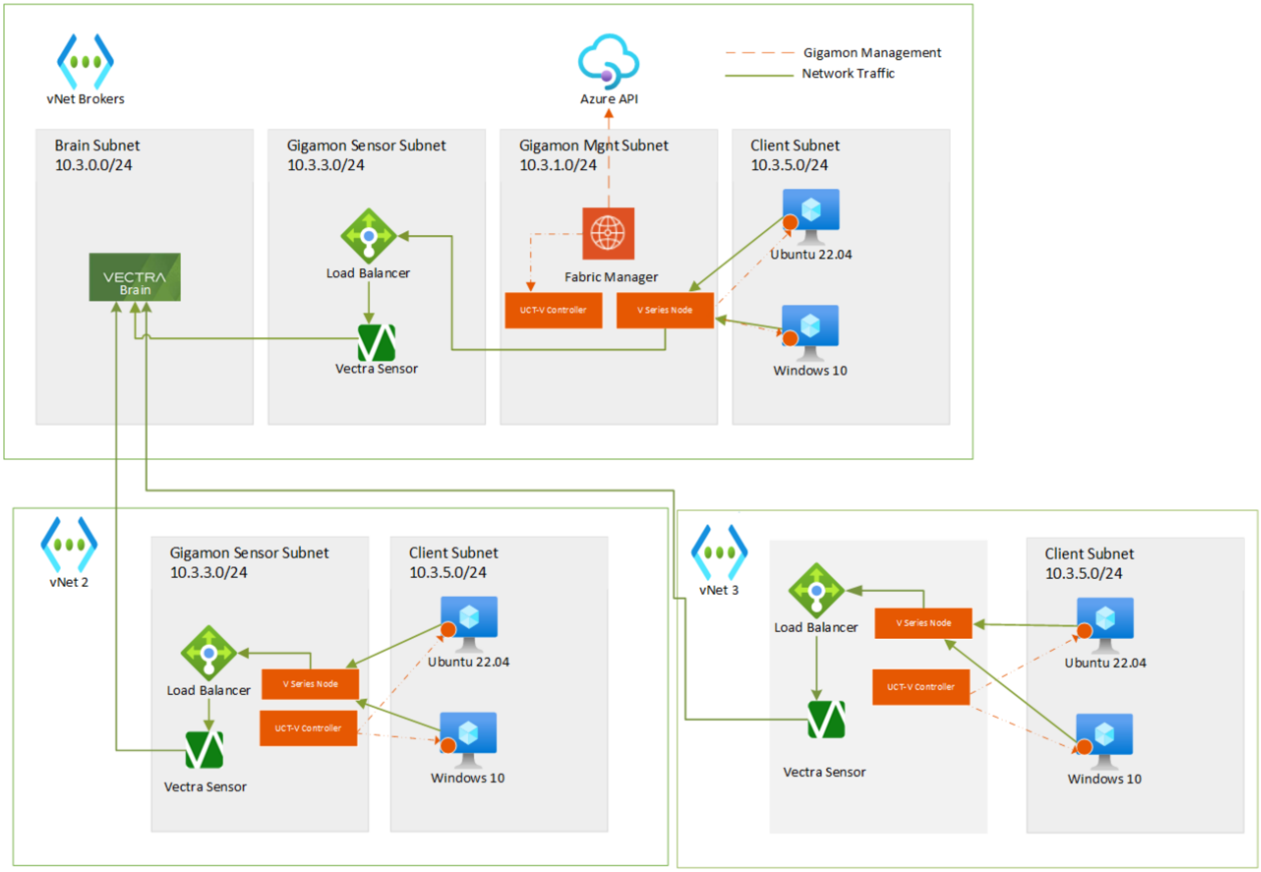


Figure 2 - Enterprise Distributed Model

In the above example, the escalating costs incurred during VNet traversal is greatly reduced by placing a Vectra Sensor, V Series Node and UCT-V Controller in each VNet of each Region. The only data that leaves the Region or VNet is metadata from the Vectra Sensor, created by the captured traffic. This data is composed of around 1% of the total traffic seen thus greatly reducing the traffic traversal costs.

Amazon Web Services Deployment

Prerequisites

Deciding the method of authentication and integration with AWS is the first step. The methods to consider are:

- **Identity and Access Management (IAM) role**— If the Fabric Manager is running in AWS, it is recommended to use an IAM role. By using an IAM role, you can securely make API requests from the Fabric Manager instance. Create an IAM role and ensure that the permissions and policies listed in Permissions and Privileges are associated to the role and ensure that you are using Customer Managed Policies or Inline Policies.
- **Access Keys**—If Fabric Manager is configured in the enterprise data center, then you must use the access keys or basic credentials to connect to the VPC. Basic credentials allow full access to all the resources in your AWS account. An access key consists of an access key ID and a secret access key.

AWS Deployment Example

Figure 3 provides an example of a muti-VPC environment in which we have created to two designated VPC's for the NDR for Cloud solution. The permissions and components are spanning both VPC's and ensure the Gigamon components can scale appropriately. There are no issues with the Vectra Sensors being deployed in either of the two VPC's.

In relation to costs, in above example the entire environment consists of a single account and all VPC's are peered within the same Availability Zone. This scenario will incur no VPC transfer costs.

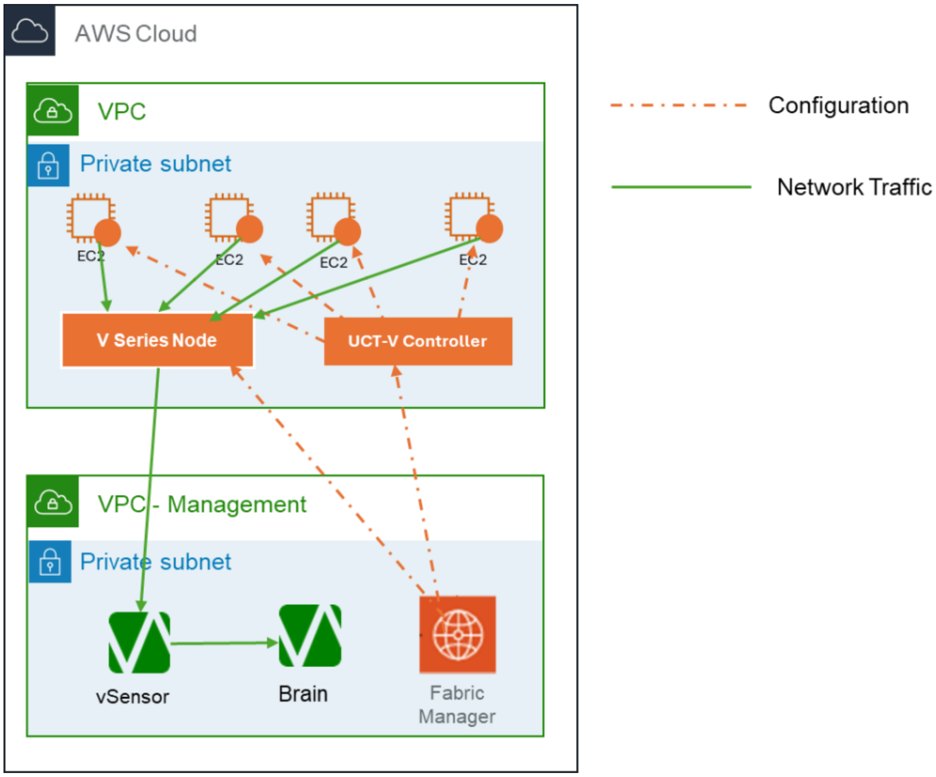


Figure 3 - AWS Recommended Example Deployment

Google Cloud Platform

Prerequisites

It is important to note that GCP does not support multiple vNIC’s of a multihomed instance being deployed in the same VPC. Per Google, they must be homed in separate VPC’s.

The NDR for Cloud solution has two multihomed components within the solution: Gigamon’s V Series Nodes and any Vectra vSensors.

GCP Deployment Example

To deploy NDR for Cloud in GCP, we must consider the requirement of only one vNIC to be present in any VPC. To accommodate this case additional VPC’s will be added to the deployment.

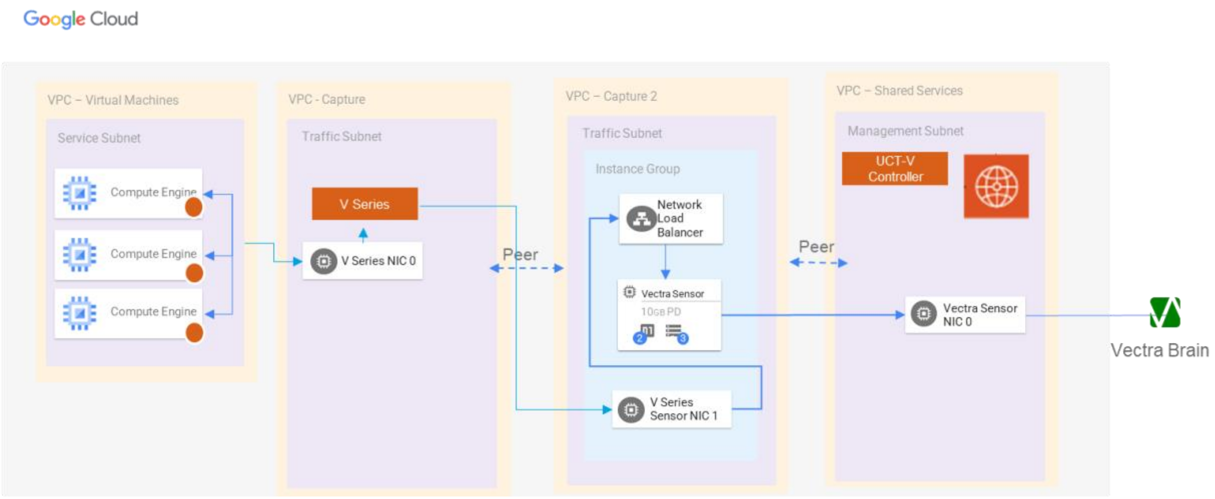


Figure 4 - GCP Recommended Example Deployment

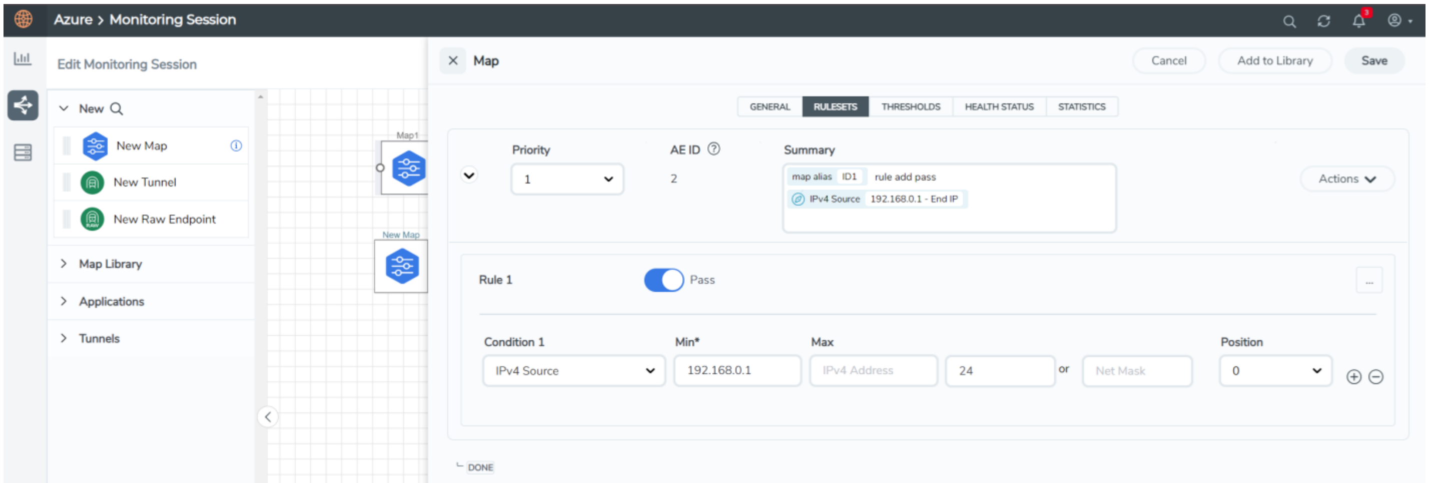
You will notice in the example above how the network interfaces are separated into different VPC’s, for each appliance creating a chain effect.

The compute engines / instances are deployed with the UCT-V agent which will then mirror the traffic to the V Series Node(s). The V Series Node(s) will then forward that traffic out via its second network interface to the network load balancer in front the vSensor appliance.

High Traffic Environments

In high traffic or larger enterprise deployments, it may become necessary to deploy additional Vectra Sensors to accommodate the higher traffic volumes and split the traffic between the Vectra Sensors. When this becomes necessary, you will need to implement additional paps within the Gigamon Monitoring Sessions.

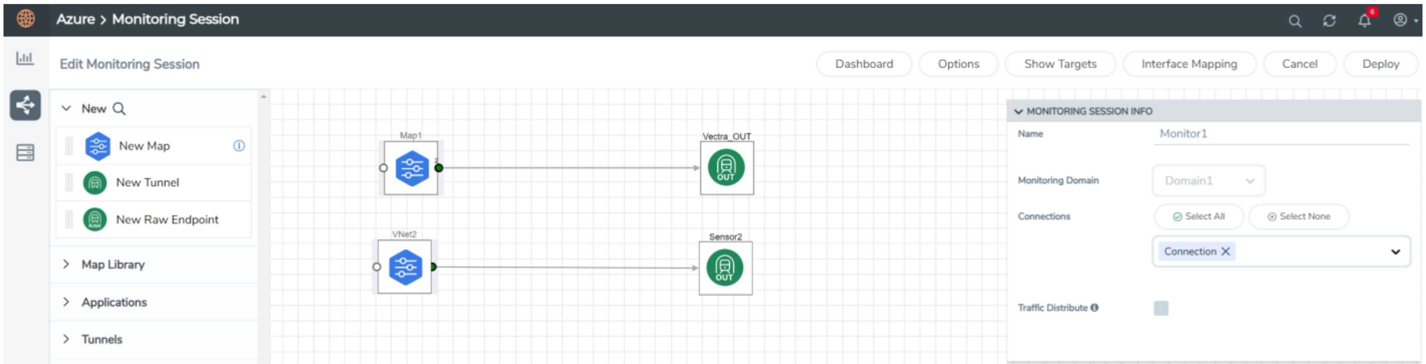
It is recommended to separate each map by the either the VNet or subnet containing the VM's to monitor. An example of such a configuration is given below.



Priority: 1
AE ID: 2
Summary: map alias ID1 rule add pass; IPv4 Source 192.168.0.1 - End IP
Rule 1: Pass
Condition 1: IPv4 Source
Min*: 192.168.0.1
Max: IPv4 Address / 24 or Net Mask
Position: 0

Figure 5 - Additional Session Map

For each Vectra Sensor, an additional map is required. These maps can be considered as traffic filtering policies. With these maps you configure which traffic is to be captured.



MONITORING SESSION INFO
Name: Monitor1
Monitoring Domain: Domain1
Connections: Connection
Traffic Distribute: □

Figure 6 - Final Multiple Sensor Session Configuration

Worldwide Support Contact Information

- Support portal: https://support.vectra.ai/
- Email: support@vectra.ai (preferred contact method)
- Additional information: https://www.vectra.ai/support